HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Khắc Minh

NGHIÊN CỨU, TRIỂN KHAI PHƯƠNG PHÁP XÁC THỰC KHÔNG MẬT KHẨU CHO ỨNG DỤNG WEB

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2026

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Khắc Minh

NGHIÊN CỨU, TRIỂN KHAI PHƯƠNG PHÁP XÁC THỰC KHÔNG MẬT KHẨU CHO ỨNG DỤNG WEB

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. ĐINH TRƯỜNG DUY

HÀ NỘI - 2026

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả đề án



Nguyễn Khắc Minh

LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới thầy TS. Đinh Trường Duy, người đã trực tiếp hướng dẫn, tận tình chỉ bảo và đưa ra những góp ý quý báu trong suốt quá trình em thực hiện đề tài thực tập. Sự quan tâm, định hướng và những kinh nghiệm thực tiễn mà thầy chia sẻ đã giúp em có thêm nhiều kiến thức bổ ích để hoàn thiện đề tài của mình.

Em xin chân thành cảm ơn các thầy, cô trong Học viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy, cô thuộc Khoa Đào tạo Sau đại học, đã luôn tận tâm trong công tác giảng dạy, truyền đạt kiến thức và tạo điều kiện thuận lợi cho em trong quá trình học tập, nghiên cứu tại Học viện.

Bên cạnh đó, em cũng xin gửi lời cảm ơn tới các anh, chị, bạn bè và đồng nghiệp đã luôn quan tâm, hỗ trợ, động viên em trong thời gian thực hiện đề tài. Những ý kiến đóng góp, sự chia sẻ và giúp đỡ của mọi người là nguồn động lực quan trọng giúp em hoàn thành tốt hơn công việc của mình.

Những kiến thức, kỹ năng và kinh nghiệm tích lũy được sẽ là nền tảng quan trọng, giúp em tự tin hơn trên con đường phát triển nghề nghiệp trong tương lai.

Em xin chân thành cảm ơn!

Hà Nội, ngày 29 tháng 06 năm 2026

Học viên thực hiện

Nguyễn Khắc Minh

MỤC LỤC

LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
DANH MỤC CÁC THUẬT NGỮ VÀ CÁC CHỮ VIẾT TẮT vi
DANH MỤC CÁC BẢNG vii
DANH MỤC CÁC HÌNH vii
MỞ ĐẦU 1
CHƯƠNG I. CƠ SỞ LÝ LUẬN 5
1.1 Tổng quan về xác thực và phân quyền người dùng trong ứng dụng web 5
1.1.1 Tìm hiểu về quá trình xác thực 5
1.1.2 Tìm hiểu về phân quyền 5
1.1.3 Vai trò của xác thực trong an toàn ứng dụng web 6
1.2 Giới thiệu về xác thực bằng mật khẩu 7
1.2.1 Khái niệm về xác thực dựa trên mật khẩu 7
1.2.2 Cơ chế hoạt động của xác thực dựa trên mật khẩu 8
1.2.3 Các rủi ro bảo mật và bất cập về trải nghiệm người dùng trong thực tế triển khai phương thức xác thực bằng mật khẩu 9
1.2.4 Một số hình thức tấn công phổ biến vào mật khẩu 11
1.3 Giới thiệu xác thực không mật khẩu 14
1.3.1 Cơ chế hoạt động của xác thực không mật khẩu 14
1.3.2 Lý do sử dụng các phương pháp xác thực không dùng mật khẩu 15
1.3.3 Các phương pháp không dùng mật khẩu xác thực phổ biến hiện nay 16
1.4 Tổng quan về mô hình zero trust 21
1.4.1 Kiến trúc tổng quan của zero trust 21
1.4.2 Các rủi ro thực tế cho thấy sự cấp thiết trong việc ứng dụng Zero Trust 23
1.4.3 Lợi ích đem đến của Zero trust 24
1.5 So sánh hình thức đăng nhập bằng mật khẩu và các giải pháp không sử dụng mật khẩu 25
1.6 Kết luận chương I 26

CHƯƠNG II. PHÂN TÍCH YÊU CẦU VÀ ĐỀ XUẤT CÁC GIẢI PHÁP XÁC THỰC KHÔNG MẬT KHẨU 27
2.1 Phân tích yêu cầu về giải pháp xác thực không mật khẩu 27
2.1.1 Yêu cầu với giải pháp xác thực không mật khẩu 27
2.1.2 Yêu cầu về khả năng tích hợp với ứng dụng web 28
2.1.3 Yêu cầu về bảo mật và khả năng mở rộng 29
2.2 Đề xuất và định hướng kiến trúc, công nghệ 31
2.2.1 Định hướng kiến trúc client server 31
2.2.2 Định hướng công nghệ 32
2.2.3 Định hướng tích hợp 35
2.3 Kết luận chương II 41
CHƯƠNG III. THIẾT KẾ, XÂY DỰNG MÔ HÌNH THỬ NGHIỆM VÀ ĐÁNH GIÁ GIẢI PHÁP 41
3.1 Thiết kế kiến trúc mô hình thử nghiệm 41
3.1.1 Thiết kế nghiệp vụ các chức năng 41
3.1.2 Tổng quát các thành phần chính (Back-end, front-end) 45
3.1.3 Mối quan hệ giữa ứng dụng web, mô-đun xác thực và các thư viện hỗ trợ xác thực: 46
3.1.4 Đảm bảo kiến trúc đáp ứng yêu cầu về bảo mật và khả năng mở rộng 47
3.2 Xây dựng và thực nghiệm giải pháp 48
3.2.1 Xây dựng giao diện người dùng 48
3.2.2 Xây dựng cơ sở dữ liệu 53
3.2.3 Xây dựng back-end cho các phương pháp không dùng mật khẩu xác thực 55
3.2.4 Các kịch bản để kiểm thử cơ chế xác thực 64
3.2.5 Thực hiện mô phỏng các kịch bản: 65
3.3 Đánh giá giải pháp: 68
3.3.1 Đánh giá tính an toàn 68
3.3.2 Tính thuận tiện 70
3.3.3 Tính ổn định 70

3.3.4 Khả năng tích hợp 71
3.3.5 Đánh giá ưu điểm và nhược điểm khi triển khai các giải pháp không dùng mật khẩu xác thực: 71
3.4 Kết luận chương III 74
KẾT LUẬN 75
DANH MỤC TÀI LIỆU THAM KHẢO 77

DANH MỤC CÁC THUẬT NGỮ VÀ CÁC CHỮ VIẾT TẮT

Từ viết tắt, thuật ngữ	Tiếng Anh	Tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
Back-end	Back-end	Phần xử lý máy chủ
Brute-force	Brute-force	Tấn công vét cạn
Client	Client	Máy khách
Keylogger	Keylogger	Phần mềm hoặc thiết bị ghi lại thao tác bàn phím
Front end	Front end	Phần giao diện phía người dùng
HTTPS	Hypertext Transfer Protocol Secure	Giao thức truyền tải siêu văn bản an toàn
ID	Identifier	Mã định danh
JSESSIONID	Java Session Identifier	Mã định danh phiên đăng nhập của Java
LTS	Long-Term Support	Hỗ trợ dài hạn
MFA	Multi-Factor Authentication	Xác thực đa yếu tố
MVC	Model-View-Controller	Mô hình – Giao diện – Bộ điều khiển
OAuth2	Open Authorization 2.0	Giao thức ủy quyền mở phiên bản 2.0
OTP	One-Time Password	Mật khẩu dùng 1 lần
OWASP	Open Worldwide Application Security Project	Dự án Bảo mật Ứng dụng Toàn cầu Mở)
Passkey	Passkey	Mã khóa, mã truy cập
Phishing	Phishing	Tấn công giả mạo
PIN	Personal Identification Number	Mã số định danh cá nhân
QR	Quick Response Code	Mã phản hồi nhanh
RFC 6238	Request for Comments 6238	Tài liệu kỹ thuật về thuật toán mật khẩu dùng một lần dựa trên thời gian
Server	Server	Máy chủ
SMTP	Simple Mail Transfer Protocol	Giao thức truyền thư điện tử đơn giản
TOTP	Time-Based One-Time Password	Mật khẩu dùng một lần dựa trên thời gian
URL	Uniform Resource Locator	Địa chỉ định danh duy nhất

Use case	Use case	Trường hợp sử dụng
User Agent	User Agent	Tác nhân người dùng
Zero Trust	Zero Trust	Mô hình bảo mật không tin cậy mặc định
WebAuthn	Web Authentication	Tiêu chuẩn web và API do World Wide Web Consortium (W3C) phát triển
W3C	World Wide Web Consortium	Hiệp hội Mạng lưới Toàn cầu

DANH MỤC CÁC BẢNG

Bảng 1.1 So sánh giữa xác thực và phân quyền 6
Bảng 1.2 So sánh đăng nhập bằng mật khẩu và không sử dụng mật khẩu 25
Bảng 3.1 Bảng mô tả use case các chức năng cơ bản trong ứng dụng 42
Bảng 3.2 Bảng mô tả use case các chức năng xác thực không mật khẩu 43
Bảng 3.3 Danh sách bảng và vai trò của các bảng trong cơ sở dữ liệu 53
Bảng 3.4 Quan hệ giữa các thực thể 54
Bảng 3.5 Tổng kết ưu và hạn chế các phương pháp không dùng 71
mật khẩu xác thực 71

DANH MỤC CÁC HÌNH

Hình 1.1 Minh họa quá trình đăng ký và xác thực dựa trên mật khẩu 9
Hình 1.2 Tổng quát mô hình zero trust 22
Hình 2.1 Tổng quan kiến trúc ứng dụng Java Spring Boot theo mô hình MVC kết hợp các lớp xử lý 31
Hình 2.2 Minh họa quá trình duy trì phiên làm việc giữa client và server 34
Hình 2.3 Đoạn code xử lý quá trình gửi mail 38
Hình 2.4 Cấu hình các thông số liên quan đến đăng nhập bằng tài khoản Google.. 39
Hình 2.5 Lấy địa chỉ email từ đối tượng WebAuthnAuthentication 41
Hình 3.1 Tổng quát luồng xử lý giữa Back-end, front-end 45
Hình 3.2 Giao diện người dùng đăng ký tài khoản 48

Hình 3.3 Giao diện đặt lại mật khẩu 49
Hình 3.4 Giao diện chức năng chỉnh sửa thông tin tài khoản 49
Hình 3.5 Giao diện chức năng xem danh sách nhân viên 49
Hình 3.6 Giao diện chức năng đăng nhập thông qua ứng dụng Google 50
Hình 3.7 Giao diện chức năng đăng nhập bằng mã xác thực OTP 50
Hình 3.8 Giao diện chức năng đăng nhập bằng liên kết 51
Hình 3.9 Liên kết xác thực được gửi đến địa chỉ email 51
Hình 3.10 Giao diện chức năng quét mã QR 51
Hình 3.11 Giao diện chức năng đăng bằng mã xác thực TOTP 52
Hình 3.12 Giao diện chức năng thêm passkey trên iPad 52
Hình 3.13 Giao diện chức năng đăng nhập bằng passkey 53
Hình 3.14 Luồng xử lý xác thực người dùng trong spring security 55
Hình 3.15 Luồng xử lý giữa authentication provider và các service 56
Hình 3.16 Luồng xử lý kết quả xác thực 57
Hình 3.17 Đoạn mã lấy thông tin nhà cung cấp và tài khoản từ ứng dụng thứ ba... 58
Hình 3.18 Đoạn mã lấy thông tin người dùng sau khi xác thực bằng Passkey 59
Hình 3.19 Quá trình đăng ký thông tin xác thực bằng passkey 60
Hình 3.20 Quá trình đăng nhập bằng cách sử dụng passkey 61
Hình 3.21 Luồng xử lý quá trình kiểm tra phiên đăng nhập 63
Hình 3.22 Tài khoản bị khóa nếu người dùng xác thực sai quá 5 lần 66
Hình 3.23 Hệ thống điều hướng sang trang cảnh báo đăng nhập bất thường 66
Hình 3.24 Mã xác thực OTP đã được sử dụng 67
Hình 3.25 Liên kết đăng nhập đã sử dụng hoặc đã hết hạn 67
Hình 3.26 Giao diện hiển thị thời gian chờ gửi lại mã OTP 68
Hình 3.27 Back-end chặn việc gửi OTP 68

MỞ ĐẦU

- Lý do chọn đề tài

Trong bối cảnh công nghệ số ngày càng phát triển, nhiều hoạt động trong đời sống, sản xuất và kinh doanh đã được triển khai trên các nền tảng web. Các hệ thống này ngày càng lưu trữ và xử lý nhiều dữ liệu quan trọng của người dùng. Vì vậy, việc xác minh chính xác danh tính của người truy cập giữ vai trò quan trọng trong bảo vệ tài khoản, hạn chế truy cập trái phép và bảo đảm an toàn thông tin. Mặc dù xác thực bằng mật khẩu vẫn được áp dụng rộng rãi, nhưng cách xác thực này dần cho thấy những bất cập liên quan đến mức độ an toàn và trải nghiệm của người sử dụng.

Xét trong quá trình sử dụng thực tế, phương thức xác thực bằng mật khẩu vẫn tiềm ẩn nguy cơ bị kẻ xấu khai thác bằng nhiều phương thức khác nhau như dò quét mật khẩu, tấn công lừa đảo, rò rỉ dữ liệu hay lợi dụng thói quen thiết lập mật khẩu có độ phức tạp thấp. Bên cạnh đó, việc người dùng phải ghi nhớ nhiều mật khẩu khác nhau khiến người dùng gặp nhiều bất tiện trong quản lý tài khoản. Những vấn đề trên cho ta thấy việc sử dụng mật khẩu đang tồn đọng các rủi ro bảo mật phát sinh trong ứng dụng web.

Từ các vấn đề trên, đề tài mang tên nghiên cứu, triển khai phương pháp xác thực không mật khẩu cho ứng dụng web được tác giả lựa chọn để tìm hiểu các mô hình xác thực không mật khẩu đang có hiện nay, triển khai thử nghiệm trên ứng dụng web cụ thể. Các phương pháp này hứa hẹn sẽ nâng cao mức độ bảo mật, khắc phục một số hạn chế của cơ chế xác thực bằng mật khẩu và mang lại quá trình đăng nhập thuận tiện hơn cho người dùng. Qua đó tác giả muốn áp dụng các giải pháp không mật khẩu và đưa ra những ưu điểm cũng như hạn chế. Tóm lại, trong đề án tác giả muốn hướng đến việc làm rõ tính khả thi và những lợi ích của xác thực không mật khẩu, qua đó tạo nguồn tham khảo cho quá trình triển khai giải pháp này trong thực tế.

- Tổng quan về vấn đề nghiên cứu

Xác thực người dùng là một thành phần có ý nghĩa lớn đối với công tác bảo đảm an ninh cho dữ liệu trong hệ thống. Hiện nay như chúng ta thấy phần lớn hệ thống sử dụng phương thức xác thực dựa trên tên đăng nhập và mật khẩu. Tuy vậy, trước sự phát triển của nhiều hình thức tấn công trên không gian mạng, phương thức này đã cho thấy những bất cập về khả năng bảo vệ và sự thuận tiện đối với người dùng.

Thực tế cho thấy xác thực bằng mật khẩu truyền thống tiềm ẩn nhiều rủi ro như người dùng đặt mật khẩu yếu, tái sử dụng mật khẩu ở nhiều hệ thống, hoặc bị tấn công lừa đảo, dò quét và rò rỉ thông tin xác thực. Bên cạnh đó, việc người dùng phải cố gắng ghi nhớ nhiều mật khẩu từ nhiều ứng dụng khác nhau cũng gây bất tiện, làm giảm trải nghiệm người dùng.

Từ các vấn đề trên, xu hướng thay thế mật khẩu bằng những hình thức xác thực khác ngày càng được chú trọng. Có thể kể đến như người dùng sử dụng đăng nhập bằng mã xác thực OTP, liên kết đăng nhập, mã xác thực TOTP, đăng nhập thông qua ứng dụng thứ ba và đăng nhập bằng passkey. Những phương pháp này được kỳ vọng giúp giảm phụ thuộc vào mật khẩu, nâng cao mức độ an toàn và đơn giản hóa quá trình đăng nhập.

Tuy nhiên, việc triển khai xác thực không mật khẩu trong thực tế vẫn còn vướng mắc với các vấn đề còn đang hiện hữu. Nhiều hệ thống chưa được thiết kế sẵn để tích hợp các cơ chế xác thực mới; quá trình chuyển đổi có thể đòi hỏi thay đổi về kiến trúc, chi phí triển khai và cách thức vận hành.

Từ thực tế đó, đề án tập trung nghiên cứu, triển khai thử nghiệm các giải pháp trên ứng dụng web và đưa ra một số kịch bản ví dụ để từ đó đánh giá tổng quan đề án dựa trên các tiêu chí như tính an toàn, tính thuận tiện, tính ổn định và khả năng tích hợp.

- Mục đích nghiên cứu

Tác giả tìm hiểu các cơ sở lý thuyết qua đó làm định hướng để áp dụng các phương thức xác thực không sử dụng mật khẩu cho ứng dụng web.

Nghiên cứu cách hoạt động, ưu điểm, hạn chế và khả năng triển khai của các giải pháp xác minh danh tính không dựa trên mật khẩu.

Xây dựng ứng dụng web và triển khai tích hợp một số phương pháp xác thực không dùng mật khẩu.

Chọn ra một số kịch bản thử nghiệm và đưa ra kết quả. Cuối cùng đánh giá tổng quan đề án bao gồm như tính an toàn, tính thuận tiện, tính tổn định và khả năng tích hợp các phương pháp không dùng mật khẩu xác thực.

Phân tích các khó khăn, hạn chế và thách thức kỹ thuật trong quá trình triển khai các phương pháp xác thực không mật khẩu.

- Phạm vi và đối tượng nghiên cứu của đề án

Tác giả lựa chọn các giải pháp đăng nhập không yêu cầu mật khẩu làm đối tượng nghiên cứu chính, bao gồm mô hình xác thực, quy trình đăng ký và đăng nhập, cơ chế xử lý thông tin xác thực

Đề án được thực hiện trong phạm vi là trên một ứng dụng web, mô phỏng một số chức năng đăng ký, đăng nhập và quản lý xác thực thường gặp trong các ứng dụng web hiện nay. Các phương pháp không dựa trên mật khẩu để xác thực được triển khai để xem xét mức độ phù hợp của giải pháp khi triển khai trong điều kiện sử dụng thực tế.

Đề án được tác giả triển khai trong thời gian thực hiện chương trình đào tạo trình độ thạc sĩ.

- Phương pháp nghiên cứu

Quá trình tác giả nghiên cứu bắt đầu từ việc tìm kiếm, chọn lọc và xem xét nội dung từ các công trình nghiên cứu, tài liệu chuyên ngành và nguồn thông tin có liên quan đến đến xác thực người dùng và các phương pháp không dựa trên mật khẩu để xác thực trên ứng dụng web.

Phân tích và so sánh bao gồm phân tích đặc điểm, đưa ra những ưu điểm và những nhược điểm hiện hữu của một số phương pháp xác thực không mật khẩu tiêu biểu, so sánh với phương pháp dùng mật khẩu truyền thống.

Triển khai thực nghiệm bao gồm xây dựng ứng dụng web thử nghiệm, tích hợp các cơ chế xác thực không mật khẩu.

Sau khi đã hoàn thành xây dựng ứng dụng web, tác giả chọn ra một số kịch bản thử nghiệm và qua đó đánh giá tổng quan ứng dụng web dưa theo một số tiêu chí như mức độ an toàn, tính thuận tiện, tính ổn định và khả năng tích hợp. Đồng thời đưa ra các hạn chế cần cải thiện.

- Ý nghĩa đóng góp của đề án

Về mặt lý luận, đề án làm rõ cơ sở lý thuyết về các phương pháp xác thực không dùng mật khẩu, đồng thời đưa ra cả ưu điểm lẫn nhược điểm đang có của từng phương pháp và so sánh với phương pháp đăng nhập sử dụng mật khẩu truyền thống.

Về mặt thực tiễn, đề án xây dựng và triển khai một ứng dụng web thử nghiệm tích hợp một số phương pháp không dùng mật khẩu để xác thực, qua đó đánh giá khả năng áp dụng, mức độ an toàn, tính ổn định và trải nghiệm người dùng trong điều kiện triển khai thực tế.

Xét trên phương diện thực tiễn, những kết quả thu được từ đề án có thể hỗ trợ quá trình xây dựng và lựa chọn cơ chế đăng nhập phù hợp cho ứng dụng web, nhất là những ứng dụng được sử dụng trong tổ chức và doanh nghiệp

CHƯƠNG I. CƠ SỞ LÝ LUẬN

1.1 Tổng quan về xác thực và phân quyền người dùng trong ứng dụng web

Xác thực và phân quyền là hai cơ chế quan trọng trong bảo mật ứng dụng web. Quá trình kiểm tra danh tính nhằm nhận diện chủ thể đang sử dụng hệ thống, còn cơ chế cấp quyền quy định các chức năng và dữ liệu mà chủ thể đó có thể khai thác. Hai cơ chế này thường được kết hợp để kiểm soát truy cập và bảo vệ dữ liệu trong hệ thống.

1.1.1 Tìm hiểu về quá trình xác thực

Xác thực là quá trình hệ thống kiểm tra thông tin nhận diện nhằm xác định chủ thể đang gửi yêu cầu truy cập. Lấy ví dụ như người dùng cung cấp thông tin cho việc xác thực, chẳng hạn như nhập mật khẩu, nhập mã xác thực OTP, và sử dụng các phương pháp đăng nhập sinh trắc học. Tiếp theo, hệ thống đối chiếu giữa thông tin người dùng lúc xác thực với dữ liệu trên hệ thống. Kết quả xác thực được coi là thành công khi thông tin đối chiếu là trùng khớp.

1.1.2 Tìm hiểu về phân quyền

Cơ chế cấp quyền được thực hiện sau bước kiểm tra danh tính nhằm quy định tài khoản có thể sử dụng chức năng nào và khai thác những dữ liệu nào. Nói một cách dễ hiểu, sau khi hệ thống biết người dùng là ai, dựa trên quyền được gán, hệ thống giới hạn những dữ liệu người dùng có thể tiếp cận và các thao tác mà họ được phép thực hiện. Thông qua cơ chế này, hệ thống kiểm soát những chức năng mà mỗi tài khoản có thể thực hiện theo vai trò và quyền hạn tương ứng.

Việc kiểm tra xác thực danh tính và việc phân quyền truy cập là hai bước độc lập trong cơ chế bảo vệ hệ thống. Xác thực dùng để xác minh người dùng có đúng là người mà họ khai báo hay không, Trong khi đó, cơ chế phân quyền quy định phạm vi dữ liệu và chức năng mà người dùng có thể sử dụng trong hệ thống. [1]

Bảng 1.1 So sánh giữa xác thực và phân quyền

Xác thực	Phân quyền
Là việc kiểm tra thông tin định danh để hệ thống xác định yêu cầu đăng nhập có xuất phát từ người sở hữu tài khoản hay không.	Là việc dựa vào phạm vi quyền của tài khoản để giới hạn các chức năng và dữ liệu mà chủ tài khoản đó có thể sử dụng.
Mục đích là để xác minh người đang đăng nhập có đúng là chủ sở hữu của tài khoản đã khai báo hay không.	Mục đích là xác định những chức năng người dùng được phép truy cập.
Xác thực thường được thực hiện thông qua mật khẩu, mã xác thực OTP, liên kết đăng nhập, các phương pháp sử dụng thông tin sinh trắc học của người dùng như nhận diện khuôn mặt hoặc dấu vân tay.	Phân quyền có thể được thực hiện dựa trên vai trò người dùng, quyền hạn.
Xác thực là bước diễn ra trước nhằm xác định người dùng là ai.	Phân quyền được thực hiện sau bước xác thực để quyết định người dùng được phép làm gì.

1.1.3 Vai trò của xác thực trong an toàn ứng dụng web

Tăng cường mức độ an toàn cho phiên đăng nhập: Thông tin xác thực sẽ được kiểm tra trước khi mở quyền sử dụng các dữ liệu và chức năng trong ứng dụng web. Nhờ đó, hệ thống có thể hạn chế truy cập trái phép và giảm nguy cơ từ các kiểu tấn công như tấn công lừa đảo, dò mật khẩu, tấn công vét cạn, hoặc chiếm đoạt phiên đăng nhập.

Bảo vệ dữ liệu: Kết quả kiểm tra thông tin xác thực là cơ sở để hệ thống nhận diện người dùng và áp dụng các quyền tương ứng. Qua đó hạn chế các hành vi xem, sửa đổi hoặc xóa thông tin khi chưa được phép, từ đó góp phần bảo đảm bí mật, dữ liệu được đảm bảo toàn vẹn và an toàn.

Đáp ứng các quy định pháp lý: Trong quá trình vận hành, nhiều hệ thống phải đáp ứng các chính sách về quyền riêng tư, bảo mật dữ liệu người dùng và những quy định pháp lý có liên quan. Việc áp dụng cơ chế xác thực phù hợp giúp nâng cao khả năng kiểm soát truy cập, qua đó góp phần hệ thống đáp ứng tốt các quy định đã được đề ra.

Tăng cường trách nhiệm của người dùng: Khi mỗi hoạt động trong hệ thống được gắn với một danh tính đã được xác minh, việc truy vết và xác định chủ thể thực hiện sẽ trở nên rõ ràng hơn. Điều này giúp nâng cao trách nhiệm của người dùng, đồng thời hỗ trợ công tác giám sát, kiểm tra và xử lý khi có sự cố xảy ra.

Nâng cao mức độ tin cậy đối với hệ thống: Các phương thức xác thực đủ mạnh không chỉ góp phần tăng cường an toàn mà còn tạo dựng niềm tin cho người dùng khi sử dụng dịch vụ. Khi danh tính được xác minh chặt chẽ, người dùng và các bên liên quan sẽ có mức độ yên tâm cao hơn đối với khả năng bảo vệ tài khoản, dữ liệu và giao dịch của hệ thống.

Phòng ngừa gian lận và giả mạo: Xác thực góp phần giảm thiểu các hành vi giả mạo danh tính, chiếm đoạt tài khoản và gian lận trong môi trường trực tuyến. Việc yêu cầu xác minh danh tính trước khi thực hiện các thao tác hoặc giao dịch quan trọng giúp hạn chế đáng kể nguy cơ lạm dụng trái phép.

1.2 Giới thiệu về xác thực bằng mật khẩu

1.2.1 Khái niệm về xác thực dựa trên mật khẩu

Trong cơ chế đăng nhập này, danh tính của người dùng được hệ thống xác thự thông qua tên tài khoản và mật khẩu. Khi yêu cầu đăng nhập cùng dữ liệu xác thực được gửi đến hệ thống, hệ thống kiểm tra mật khẩu do người dùng cung cấp bằng cách đối chiếu với mật khẩu đã lưu trên hệ thống. Khi kết quả đối chiếu phù hợp, người dùng được xác thực thành công.

Mật khẩu là một chuỗi ký tự bí mật do người dùng sử dụng để chứng minh danh tính khi truy cập vào thiết bị, website, ứng dụng hoặc các hệ thống thông tin.

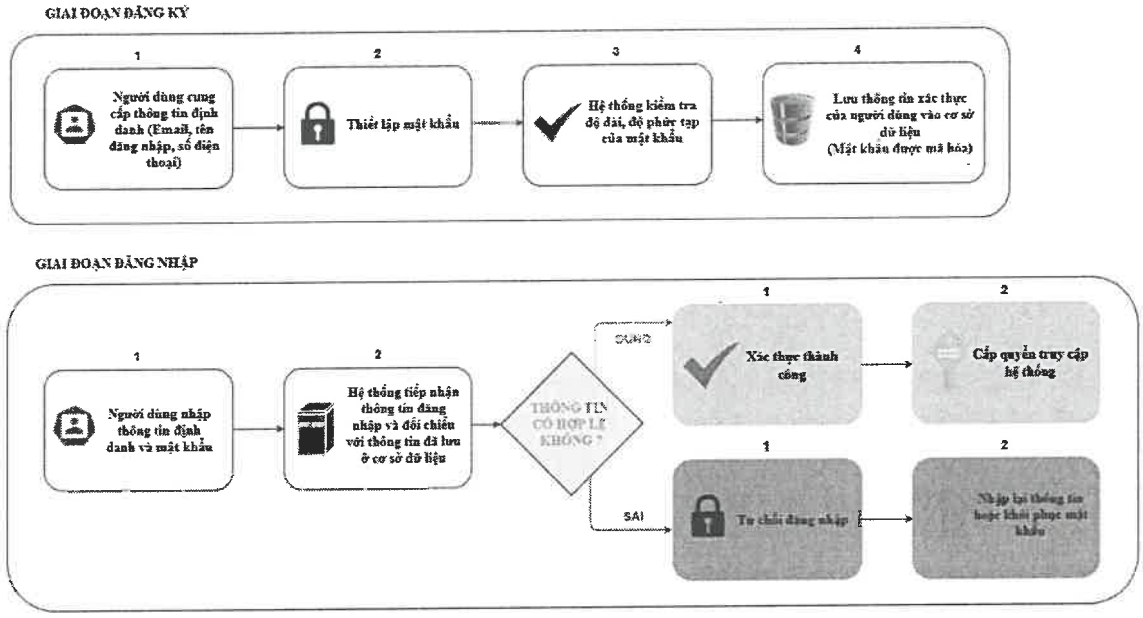
Trong môi trường số, mật khẩu có thể được xem như một chìa khóa giúp mở quyền truy cập, đồng thời là một trong những cơ chế xác thực cơ bản và phổ biến nhất hiện nay.

Về bản chất, mật khẩu đóng vai trò là công cụ xác thực nhằm giúp hệ thống kiểm tra người đang yêu cầu truy cập có đúng là chủ thể hợp lệ hay không. Cơ chế này tạo ra tuyến phòng vệ ban đầu nhằm hạn chế những yêu cầu truy cập không hợp lệ.

1.2.2 Cơ chế hoạt động của xác thực dựa trên mật khẩu

Xác thực dựa trên mật khẩu bắt đầu từ giai đoạn đăng ký tài khoản, khi người dùng cung cấp thông tin định danh, ví dụ như địa chỉ email và thiết lập mật khẩu tương ứng. Để tăng mức độ an toàn, hệ thống đặt ra một số yêu cầu khi đặt mật khẩu, chẳng hạn về độ dài hoặc mức độ phức tạp. Sau khi được tạo xong tài khoản, thông tin xác thực của người dùng sẽ được lưu trữ trong cơ sở dữ liệu nhằm mục đích đối chiếu khi có quá trình xác thực sau này.

Đến với quá trình đăng nhập, người dùng điền thông tin định danh cùng mật khẩu. Hệ thống tiếp nhận các thông tin này, thực hiện kiểm tra và đối chiếu với dữ liệu đang lưu trữ trong hệ thống. Khi dữ liệu kiểm tra hợp lệ, hệ thống thiết lập phiên đăng nhập và tài khoản được phép sử dụng các chức năng phù hợp với quyền đã được thiết lập. Trường hợp thông tin cung cấp không phù hợp với dữ liệu hiện có, yêu cầu đăng nhập sẽ bị từ chối. Khi không thể đăng nhập, người dùng có thể sử dụng chức năng khôi phục mật khẩu để lấy lại quyền truy cập vào tài khoản.



Hình 1.1 Minh họa quá trình đăng ký và xác thực dựa trên mật khẩu

1.2.3 Các rủi ro bảo mật và bất cập về trải nghiệm người dùng trong thực tế triển khai phương thức xác thực bằng mật khẩu

a) Các rủi ro bảo mật của phương thức xác thực bằng mật khẩu

Chính sách mật khẩu chưa đủ mạnh: Nếu hệ thống cho phép người dùng thiết lập các mật khẩu ngắn, đơn giản hoặc thuộc nhóm mật khẩu phổ biến, mức độ an toàn của cơ chế xác thực sẽ bị ảnh hưởng đáng kể. Những mật khẩu như vậy thường dễ bị đoán ra hoặc bị phá bằng các công cụ thử mật khẩu tự động. Đây là một trong những yếu tố thường dẫn đến nguy cơ tài khoản bị truy cập trái phép.

Thói quen tái sử dụng mật khẩu của người dùng: Trong thực tế, nhiều người dùng có xu hướng sử dụng cùng một mật khẩu cho nhiều hệ thống khác nhau nhằm dễ ghi nhớ. Tuy nhiên, điều này làm gia tăng nguy cơ mất an toàn trên diện rộng. Một số báo cáo thực tế cho thấy người dùng vẫn có xu hướng sử dụng các mật khẩu đơn giản và dễ đoán. Theo số liệu trong báo cáo, các mật khẩu phổ biến như 123456, admin, 12345678, password và Password vẫn xuất hiện với số lượng lớn trong danh sách mật khẩu bị đánh cắp. Cụ thể, mật khẩu 123456 có khoảng 3,7 triệu kết quả trùng khớp, admin có khoảng 1,9 triệu, 12345678 có khoảng 1,5 triệu, trong khi

password và Password lần lượt có khoảng 558.000 và 474.000 kết quả. Điều này cho thấy việc người dùng lựa chọn mật khẩu yếu vẫn là một vấn đề phổ biến, làm gia tăng nguy cơ bị dò đoán mật khẩu. [2]

Xác thực đa yếu tố chưa được áp dụng: Cách tiếp cận sử dụng đồng thời nhiều phương thức kiểm tra danh tính giúp ngăn chặn những yêu cầu truy cập không hợp lệ so với phương thức chỉ yêu cầu duy nhất mật khẩu. Một nghiên cứu của Microsoft cho thấy hơn 99,99% tài khoản bật MFA vẫn được bảo vệ trong thời gian khảo sát; đồng thời, MFA giúp giảm 99,22% nguy cơ tài khoản bị xâm phạm và giảm 98,56% rủi ro đối với các tài khoản có thông tin đăng nhập bị rò rỉ. Do đó, nếu hệ thống không áp dụng MFA, tài khoản người dùng sẽ dễ bị chiếm quyền hơn khi mật khẩu bị lộ hoặc bị khai thác qua các cách tấn công như tấn công giả mạo, vét cạn hoặc chiếm đoạt phiên đăng nhập. [3]

Không có cơ chế kiểm soát số lần đăng nhập thất bại: Một số hệ thống chưa giới hạn số lần nhập sai thông tin đăng nhập hoặc chưa áp dụng biện pháp tạm khóa tài khoản sau nhiều lần thất bại liên tiếp. Điều này tạo điều kiện lý tưởng cho những cuộc tấn công vét cạn hoặc các hình thức dò quét thông tin xác thực có thể tiếp tục kéo dài trước khi hệ thống kịp thời phát hiện và xử lý.

Việc duy trì và kiểm soát phiên người dùng chưa được bảo vệ đầy đủ: Khi danh tính người truy cập đã được kiểm chứng, một phiên làm việc sẽ được khởi tạo để duy trì trạng thái đăng nhập. Tuy nhiên, nếu phiên làm việc không được hủy đúng cách sau khi đăng xuất, Đối tượng tấn công có khả năng lợi dụng phiên đăng nhập để truy cập trái phép vào hệ thống. Khi đó, dù không biết mật khẩu, đối tượng tấn công vẫn có thể giả mạo người dùng hợp lệ để truy cập hệ thống.

Thông tin xác thực bị lộ trong quá trình truyền hoặc lưu trữ: Việc bảo vệ thông tin đăng nhập không chỉ dừng lại ở bước kiểm tra mật khẩu mà còn liên quan chặt chẽ đến cách dữ liệu được truyền và lưu trữ. Nếu mật khẩu được gửi qua mạng mà không có kênh truyền an toàn, hoặc được lưu trong hệ thống dưới dạng không được bảo vệ thích hợp, nguy cơ bị đánh cắp sẽ tăng lên rõ rệt.

b) Bất cập về trải nghiệm người dùng trong thực tế triển khai phương thức xác thực bằng mật khẩu

Bất tiện trong quá trình sử dụng và chi phí khôi phục tài khoản cao: Trong thực tế, người dùng thường phải quản lý quá nhiều mật khẩu dùng cho nhiều tài khoản đăng ký cho các ứng dụng. Việc ghi nhớ đồng thời nhiều mật khẩu, đặc biệt là các mật khẩu có độ phức tạp cao, dễ gây khó khăn và gây cho người dùng sự bất tiên khi sử dụng. Điều này khiến người dùng có xu hướng chọn những cách xử lý kém an toàn hơn, chẳng hạn như dùng lại cùng một mật khẩu cho nhiều tài khoản hoặc lưu trữ mật khẩu theo cách không bảo mật.

Phát sinh thêm chi phí vận hành, đặc biệt là các yêu cầu trợ giúp phát sinh khi người dùng không nhớ hoặc cần thiết lập lại mật khẩu: Việc tiếp nhận và xử lý các trường hợp khôi phục mật khẩu làm gia tăng khối lượng công việc cho bộ phận kỹ thuật, đồng thời khiến người dùng phải chờ đợi trước khi có thể tiếp tục truy cập hệ thống. Sự gián đoạn này có thể làm giảm mức độ hài lòng của khách hàng trong quá trình sử dụng dịch vụ. Do đó, việc giảm sự phụ thuộc vào mật khẩu, chẳng hạn thông qua các phương thức xác thực không mật khẩu, giúp người dùng tự thực hiện các thao tác cần thiết, qua đó giảm số trường hợp cần đến sự can thiệp của nhân viên hỗ trợ. [3]

Quy trình khôi phục tài khoản hoặc đặt lại mật khẩu được thiết kế quá nhiều bước, mất quá nhiều thời gian: Điều này làm gián đoạn quá trình sử dụng dịch vụ, khiến người dùng không thể tiếp tục thao tác đúng thời điểm họ có nhu cầu. Khi phải chờ đợi quá lâu hoặc gặp quá nhiều trở ngại, người dùng có thể dừng truy cập, không hoàn thành phiên làm việc đang thực hiện hoặc thậm chí từ bỏ việc sử dụng dịch vụ.

1.2.4 Một số hình thức tấn công phổ biến vào mật khẩu

a) Tấn công vét cạn

Đây là hình thức kẻ tấn công sử dụng nhiều tổ hợp tên đăng nhập và mật khẩu khác nhau để đăng nhập cho đến khi đăng nhập thành công mới dừng thì thôi.

Cách thức tấn công là kẻ tấn công sử dụng công cụ tự động để gửi liên tục các yêu cầu đăng nhập với nhiều mật khẩu khác nhau. Nếu hệ thống không có các cơ chế như giới hạn số lần đăng nhập thất bại, không khóa quyền truy cập tài khoản hoặc không phát hiện hành vi bất thường, quá trình dò mật khẩu diễn ra liên tục và kéo dài. Hình thức này hiệu quả hơn khi người dùng đặt mật khẩu đơn giản và dễ đoán.

Cách thứ hai là kẻ xấu chiếm được quyền kiểm soát cơ sở dữ liệu chứa mật khẩu đã được băm, chúng có thể tạo ra nhiều mật khẩu giả định, các mật khẩu này được đưa qua thuật toán băm tương ứng để tạo ra chuỗi đầu ra. Chuỗi này tiếp tục được đối chiếu với dữ liệu băm thu được trước đó; khi kết quả giống nhau, mật khẩu ban đầu có thể được suy ra. Hình thức này cho thấy việc lưu trữ mật khẩu bằng thuật toán băm sẽ làm tăng nguy cơ bị dò ra mật khẩu gốc.

b) Tấn công dò tìm từ điển

Đây là cách tấn công theo cách kẻ xấu dò mật khẩu, cụ thể là tập hợp những danh sách những mật khẩu thường được người dùng lựa chọn, từ ngữ thông dụng hoặc các mẫu mật khẩu thường gặp như abc123, 12345678

Cách thức tấn công là thay vì thử toàn bộ tổ hợp ký tự như tấn công vét cạn, kẻ tấn công tập trung vào những mật khẩu có xác suất được người dùng sử dụng cao, chẳng hạn như 123456, password, ngày sinh, tên cá nhân hoặc các chuỗi đơn giản. Danh sách này có thể được lấy từ các bộ mật khẩu bị rò rỉ hoặc được xây dựng dựa trên thói quen đặt mật khẩu của người dùng. Vì vậy, tấn công từ điển thường nhanh và hiệu quả hơn tấn công vét cạn trong nhiều trường hợp thực tế. [4]

c) Tấn công giả mạo

Trong hình thức này, đối tượng xấu xây dựng nội dung hoặc giao diện có vẻ đáng tin cậy nhằm đánh lừa người sử dụng. các nội dung giả mạo như email, tin nhắn, đường dẫn hoặc trang đăng nhập có giao diện giống với trang thật. Mục đích của hình thức này là khiến người dùng tin rằng họ đang truy cập vào hệ thống thật, từ đó tự

nhập và cung cấp các thông tin nhạy cảm như tên đăng nhập, mật khẩu, mã OTP hoặc thông tin tài khoản khác cho kẻ tấn công.

Quá trình tấn công bắt đầu bằng việc kẻ tấn công thu thập thông tin về người dùng, chẳng hạn như email, tổ chức mà người dùng đang sử dụng, dịch vụ liên quan hoặc các thông tin công khai trên internet. Dựa trên những thông tin này, kẻ tấn công lựa chọn phương thức tấn công phù hợp, ví dụ như gửi email giả mạo, tạo trang đăng nhập giả hoặc gửi liên kết độc hại.

Ở giai đoạn chuẩn bị, kẻ tấn công xây dựng nội dung giả mạo sao cho giống với trang thật, bao gồm logo, màu sắc, bố cục giao diện, tên thương hiệu hoặc cách trình bày quen thuộc. Sau đó, cuộc tấn công được thực hiện bằng cách gửi thông báo, email hoặc đường dẫn giả mạo đến người dùng, thường kèm theo nội dung mang tính thúc ép như yêu cầu xác minh tài khoản, cập nhật thông tin hoặc cảnh báo tài khoản sẽ bị khóa.

Nếu người dùng tin rằng thông báo là hợp lệ và nhập thông tin vào trang giả mạo, các dữ liệu này sẽ được gửi trực tiếp cho kẻ tấn công. Sau khi thu thập được thông tin, kẻ tấn công có thể sử dụng chúng để đăng nhập trái phép, chiếm đoạt tài khoản, rút tiền, tống tiền hoặc tiếp tục thực hiện các hành vi gian lận khác. [5]

d) Tấn công chiếm đoạt phiên đăng nhập

Tấn công chiếm đoạt phiên đăng nhập là cách tấn công mà kẻ xấu chiếm đoạt mã định danh phiên đăng nhập (session ID). Kẻ tấn công có thể chuyển yêu cầu đến máy chủ và được hệ thống xử lý nếu thông tin phiên đi kèm vẫn còn hợp lệ, hệ thống sẽ xem yêu cầu đó như được gửi từ người dùng đã đăng nhập, từ đó kẻ xấu truy cập được vào tài khoản khi thậm chí không phải nhập mật khẩu.

Cách thức tấn công cụ thể là trong tấn công chiếm đoạt phiên đăng nhập, kẻ tấn công không chỉ đánh cắp thông tin phiên mà còn can thiệp trực tiếp vào phiên làm việc đang diễn ra giữa người dùng và máy chủ. Sau khi người dùng đăng nhập thành công, máy chủ tạo một mã phiên nhằm ghi nhận trạng thái đăng nhập. Trong các lần

tương tác tiếp theo, mã này được gửi kèm theo yêu cầu để máy chủ nhận biết phiên đăng nhập tương ứng để máy chủ nhận biết rằng yêu cầu đến từ phiên đăng nhập đã được xác thực.

Kẻ tấn công có thể theo dõi quá trình trao đổi dữ liệu giữa người dùng và máy chủ nhằm thu thập các thông tin liên quan đến phiên đăng nhập. Nếu lấy được mã định danh phiên đăng nhập, kẻ tấn công có thể sử dụng các thông tin này để tạo yêu cầu gửi đến máy chủ dưới danh nghĩa người dùng thật. [6]

1.3 Giới thiệu xác thực không mật khẩu

Phương pháp không dùng mật khẩu khi đăng nhập cho phép hệ thống nhận diện tài khoản mà không yêu cầu xác thực không sử dụng mật khẩu. Thay vào đó, hệ thống sử dụng các yếu tố xác thực khác như mã dùng một lần, liên kết đăng nhập, dữ liệu sinh trắc học. Mục tiêu của phương thức này là giảm sự phụ thuộc vào mật khẩu, từ đó hạn chế các rủi ro như mật khẩu yếu, tái sử dụng mật khẩu, phishing hoặc rò rỉ thông tin đăng nhập.

1.3.1 Cơ chế hoạt động của xác thực không mật khẩu

Trước hết, người dùng nhập email hoặc tên tài khoản để hệ thống xác định danh tính cần đăng nhập. Tiếp theo, quá trình xác minh được thực hiện bằng phương thức tương ứng, chẳng hạn như cung cấp mã OTP được gửi qua email, nhập mã TOTP từ ứng dụng xác thực hoặc đăng nhập bằng passkey thông qua vân tay, khuôn mặt hay mã PIN trên thiết bị.

Đối chiếu thông tin người dùng: Các dữ liệu được gửi từ bước trước sẽ được hệ thống kiểm tra. Nếu kết quả hợp lệ, phiên đăng nhập được thiết lập và tài khoản được sử dụng các chức năng tương ứng.

Ví dụ với phương thức xác thực bằng mã OTP, hệ thống xác minh mã xác thực 6 số. Với liên kết đăng nhập, hệ thống xác minh mã đăng nhập dựa trên các điều kiện như thời hạn sử dụng, trạng thái đã sử dụng hay chưa. Nếu mã xác thực chưa được sử dụng hoặc thời gian hiệu lực vẫn còn, quá trình xác thực diễn ra thành công.

Đối với phương thức đăng nhập thông qua ứng dụng thứ ba, ví dụ: Khi lựa chọn đăng nhập với tài khoản Google, người truy cập được đưa đến giao diện đăng nhập của ứng dụng Google. Sau khi hoàn tất bước kiểm tra danh tính, Google chuyển kết quả xác nhận về địa chỉ phản hồi của ứng dụng để hệ thống tiếp tục xử lý và thiết lập phiên đăng nhập. Cụ thể, hệ thống sử dụng mã xác thực để trao đổi lấy mã truy cập. Dựa vào mã truy cập, hệ thống lấy ra được các dữ liệu định danh cần thiết như địa chỉ email, họ và tên và đối chiếu với tài khoản đang có trong cơ sở dữ liệu. Nếu quá trình đối chiếu và kiểm tra hợp lệ, quá trình đăng nhập thành công. [7]

Đối với phương thức đăng nhập bằng passkey, cơ chế xác thực được thực hiện dựa trên một cặp khóa gồm cặp khóa bí mật và khóa công khai. Trong quá trình đăng ký, thiết bị hoặc bộ xác thực của người dùng tạo ra một cặp khóa bất đối xứng này. Trong khi khóa công khai được gửi đến và lưu trữ trên máy chủ thì khóa bí mật được lưu giữ an toàn trên thiết bị và không được gửi đến máy chủ. Khi đăng nhập, máy chủ tạo một dữ liệu thử thách và gửi đến thiết bị của người dùng. Người dùng xác minh danh tính bằng vân tay, khuôn mặt, mã PIN để cho phép thiết bị sử dụng khóa bí mật ký dữ liệu xác thực. Sau đó, khóa công khai tương ứng được máy chủ dùng để xác minh tính hợp lệ của chữ ký. Nếu kết quả kiểm tra hợp lệ, người dùng được xác thực thành công. [8]

1.3.2 Lý do sử dụng các phương pháp xác thực không dùng mật khẩu

Đầu tiên là tính bảo mật, xác thực không mật khẩu giúp giảm thiểu các rủi ro phổ biến liên quan đến mật khẩu như bị đánh cắp, bị dò đoán hoặc bị lợi dụng qua các hình thức lừa đảo đăng nhập. Khi không còn phụ thuộc vào mật khẩu tĩnh, hệ thống cũng hạn chế được đáng kể nguy cơ rò rỉ thông tin xác thực từ phía người dùng.

Một báo cáo điều tra được trích dẫn trong nghiên cứu cho thấy 81% các vụ vi phạm dữ liệu có liên quan đến mật khẩu không an toàn hoặc mật khẩu đã bị xâm phạm. Bên cạnh đó, sự phát triển mạnh mẽ của các dịch vụ trực tuyến khiến người dùng ngày càng phải quản lý nhiều tài khoản ứng dụng cùng với nhiều mật khẩu hơn, từ đó làm gia tăng gánh nặng trong việc ghi nhớ và bảo vệ thông tin đăng nhập. [9]

Thứ hai, phương thức này còn cải thiện trải nghiệm sử dụng. Việc sử dụng một phương thức nhận diện thống nhất giúp giảm gánh nặng ghi nhớ thông tin truy cập cho từng tài khoản, từ đó giảm áp lực trong quá trình đăng nhập và hạn chế tình trạng quên mật khẩu. Theo khảo sát của Keeper Security, có khoảng 76% người tham gia cho biết họ cố gắng ghi nhớ hoặc ghi lại mật khẩu ở một nơi nào đó; trong đó 60% người trong số họ phải đặt lại mật khẩu trong khoảng 60 ngày; và 33% người dùng cần từ 3 đến 4 lần thử mới có thể nhớ lại mật khẩu đã quên. Những số liệu này cho thấy mật khẩu không chỉ tồn tại rủi ro về bảo mật mà còn tạo ra gánh nặng đáng kể đối với trải nghiệm người dùng. [10]

Thứ ba, xác thực không mật khẩu góp phần đơn giản hóa quá trình quản lý tài khoản trong hệ thống. Khi không còn phụ thuộc vào mật khẩu, người dùng sẽ ít gặp các vấn đề phổ biến như quên mật khẩu, nhập sai nhiều lần hoặc phải thực hiện đặt lại mật khẩu. Điều này giúp giảm bớt các thao tác hỗ trợ lặp lại trong quá trình vận hành, đồng thời hạn chế tình trạng gián đoạn truy cập của người dùng. Về phía đơn vị quản trị hệ thống, việc giảm số lượng yêu cầu liên quan đến mật khẩu có thể góp phần tối ưu nguồn lực hỗ trợ kỹ thuật và nâng cao hiệu quả quản lý tài khoản.

1.3.3 Các phương pháp không dùng mật khẩu xác thực phổ biến hiện nay

a) Đăng nhập bằng mã xác thực OTP

Đây là một mã xác thực tạm thời được hệ thống tạo ra khi người dùng thực hiện đăng nhập, xác nhận giao dịch hoặc gửi yêu cầu xác minh danh tính. Mã này thường gồm các chữ số ngẫu nhiên, với độ dài và định dạng có thể thay đổi tùy theo mức độ bảo mật mà hệ thống yêu cầu.

Sau khi được tạo, mã xác thực sẽ được gửi qua email hoặc các ứng dụng nhắn tin. Người dùng sau đó nhập lại mã này trên ứng dụng hoặc website để hoàn tất bước kiểm tra danh tính. Khác với mật khẩu truyền thống có thể sử dụng nhiều lần, mã xác thực chỉ có thời gian sống ngắn hoặc cho một lần xác thực duy nhất. [1]

Ưu điểm:

- Tăng cường bảo mật nhờ bổ sung thêm một lớp xác thực ngoài mật khẩu.
- Giảm rủi ro từ việc tái sử dụng mật khẩu, nhất là khi thông tin đăng nhập bị rò rỉ ở hệ thống khác.
- Hỗ trợ xác minh và khôi phục tài khoản khá thuận tiện, dễ áp dụng trong nhiều tình huống.
- Dễ tiếp cận với người dùng vì đây là phương thức đã quen thuộc và được sử dụng rộng rãi.

Nhược điểm:

- Phụ thuộc vào độ an toàn của kênh truyền mã như SMS, email hoặc ứng dụng nhắn tin.
- Làm tăng số bước thao tác trong quá trình đăng nhập hoặc xác minh, có thể gây gián đoạn trải nghiệm.
- Phát sinh chi phí vận hành và hỗ trợ do phải gửi mã, tích hợp hệ thống và xử lý sự cố liên quan.
- Bị giới hạn về khả năng tiếp cận trong một số trường hợp, chẳng hạn khi người dùng mất kết nối, đổi thiết bị hoặc không nhận được mã.

b) Đăng nhập bằng mã xác thực TOTP

Cách hoạt động dựa trên việc tạo ra mã xác thực có hiệu lực trong thời gian ngắn từ hai yếu tố chính, gồm khóa bí mật dùng chung và thời gian hiện tại. Trong đó, khóa bí mật sẽ được thiết lập trước giữa máy chủ và ứng dụng xác thực của người dùng trong quá trình đăng ký. Khi người dùng đăng nhập, cả hai phía sẽ đồng thời sử dụng cùng một khóa bí mật, cùng thuật toán và cùng khoảng thời gian xác định, thường là 30 giây, để sinh ra mã xác thực. Nếu mã người dùng nhập vào trùng với mã mà máy chủ tạo ra tại thời điểm đó, hệ thống sẽ chấp nhận yêu cầu xác thực. [3]

Ưu điểm:

- Tăng cường bảo mật nhờ bổ sung thêm một lớp xác thực ngoài mật khẩu tĩnh.

- Phù hợp với mô hình xác thực đa yếu tố, giúp giảm nguy cơ truy cập trái phép vào tài khoản.
- Có thể hoạt động ngoại tuyến, không phụ thuộc vào kết nối mạng hay việc gửi mã qua tin nhắn.
- Không cần sử dụng số điện thoại, từ đó hạn chế việc phụ thuộc vào thông tin cá nhân của người dùng.
- Thao tác xác thực thường nhanh và ổn định hơn so với việc chờ nhận mã xác thực qua tin nhắn.

Nhược điểm:

- Phụ thuộc vào thiết bị hoặc ứng dụng xác thực mà người dùng đã đăng ký trước đó.
- Việc thiết lập ban đầu có thể mất thời gian, đặc biệt khi người dùng sử dụng nhiều tài khoản hoặc nhiều dịch vụ.
- Đòi hỏi khóa bí mật phải được quản lý chặt chẽ ở cả phía người dùng và hệ thống nhằm hạn chế nguy cơ bị lộ hoặc sử dụng trái phép.

c) Sử dụng liên kết đăng nhập

Với cơ chế này, hệ thống gửi một đường dẫn có hiệu lực trong thời gian giới hạn để người dùng truy cập và hoàn tất quá trình đăng nhập mà không phải nhập mật khẩu. Sau khi tiếp nhận yêu cầu truy cập tài khoản, ứng dụng tạo một đường dẫn liên kết xác thực có chứa mã xác thực. Mỗi mã xác thực chỉ có hiệu lực cho một lần sử dụng và hết hạn sau thời gian ngắn nhằm hạn chế nguy cơ bị lạm dụng.

Khi người dùng nhấn vào liên kết đăng nhập, hệ thống sẽ kiểm tra tính hợp lệ của liên kết và thực hiện xác thực. Cụ thể là đối chiếu mã bên trong liên kết với mã xác nhận đã lưu trước đó trên hệ thống. Sau khi xác minh yêu cầu là hợp lệ, ứng dụng ghi nhận danh tính tài khoản và thiết lập phiên đăng nhập. [1]

Ưu điểm:

- Đơn giản hóa trải nghiệm đăng nhập, do người dùng không cần ghi nhớ và tạo mật khẩu riêng cho từng ứng dụng.
- Giảm khối lượng công việc liên quan đến quản lý mật khẩu cho hệ thống, vì không cần lưu trữ và bảo vệ mật khẩu theo cách truyền thống.
- Giảm nguy cơ mật khẩu bị lộ, do phương thức này không phụ thuộc vào mật khẩu tĩnh.

Nhược điểm:

- Phụ thuộc vào dịch vụ gửi và nhận email. Nếu email chứa liên kết được gửi chậm, thất bại do kết nối Internet hoặc bị chuyển vào thư rác, quá trình đăng nhập sẽ bị gián đoạn.
- Nếu tài khoản email bị kẻ xấu quyền truy cập trái phép thì kẻ đó có thể sử dụng liên kết xác thực để đăng nhập trái phép vào hệ thống.
- Liên kết xác thực chứa mã thông báo trong URL nên có nguy cơ bị lộ qua lịch sử trình duyệt, nhật ký hệ thống, thao tác chuyển tiếp email hoặc việc chia sẻ liên kết không an toàn.

d) Xác thực bằng passkey

Thay vì xác thực bằng mật khẩu, cách xác thực này sử dụng cơ chế khóa công khai và khóa bí mật; trong đó khóa bí mật được giữ trên thiết bị người dùng, còn khóa công khai được dịch vụ lưu để kiểm tra. Trong quá trình đăng nhập bằng Passkey, người dùng xác minh danh tính ngay trên thiết bị thông qua vân tay, nhận diện khuôn mặt hoặc mã mở khóa. Sau khi được người dùng chấp thuận, thiết bị sử dụng khóa riêng để tạo dữ liệu xác thực gửi đến hệ thống. Máy chủ sử dụng khóa công khai đã lưu để xác minh chữ ký trong dữ liệu xác thực do thiết bị gửi về. Nhờ đó, người dùng không phải nhập mật khẩu mà chỉ cần xác nhận danh tính trên thiết bị. Cách tiếp cận này góp phần hạn chế các rủi ro liên quan đến mật khẩu, đồng thời đơn giản hóa thao tác đăng nhập. [11]

Ưu điểm:

- Tăng cường bảo mật nhờ cơ chế xác thực dựa trên khóa mật mã và thiết bị người dùng.
- Cho phép người dùng xác nhận danh tính bằng mã PIN, dấu vân tay hoặc dữ liệu khuôn mặt.
- Chống lừa đảo giả mạo tốt hơn do quá trình xác thực gắn với đúng ứng dụng hoặc website hợp lệ.
- Cải thiện trải nghiệm người dùng nhờ thao tác đăng nhập nhanh, đơn giản và không cần ghi nhớ mật khẩu.

Nhược điểm:

- Phụ thuộc vào thiết bị, nên việc mất hoặc hỏng thiết bị có thể ảnh hưởng đến khả năng truy cập tài khoản.
- Mức độ tương thích của passkey còn phụ thuộc vào thiết bị, hệ điều hành và trình duyệt mà người dùng đang sử dụng.
- Tâm lý ngại thay đổi từ mật khẩu sang passkey có thể ảnh hưởng đến mức độ tiếp nhận ban đầu.

e) Xác thực bằng ứng dụng bên thứ ba:

Xác thực thông qua ứng dụng bên thứ ba cho phép người dùng sử dụng tài khoản đã có tại một nhà cung cấp định danh, chẳng hạn như Google, để truy cập vào hệ thống mà không cần tạo và ghi nhớ thêm mật khẩu mới. Trong quá trình này, người dùng thực hiện đăng nhập trực tiếp tại nhà cung cấp, còn ứng dụng chỉ tiếp nhận kết quả xác thực và những thông tin định danh đã được người dùng cho phép chia sẻ. Sau khi quá trình cấp quyền hoàn tất, ứng dụng sử dụng mã truy cập được nhà cung cấp phát hành để lấy các thông tin cần thiết và hoàn tất đăng nhập mà không phải tiếp nhận mật khẩu của người dùng. [12]

Ưu điểm:

- Không cần tạo và ghi nhớ mật khẩu mới.

- Ứng dụng không trực tiếp tiếp nhận và quản lý mật khẩu của ứng dụng bên thứ ba.

Nhược điểm:

- Nếu ứng dụng thứ ba bị gián đoạn hoặc ngừng hoạt động, người dùng không đăng nhập được vào ứng dụng
- Ứng dụng có thể không nhận được đầy đủ thông tin người dùng.

1.4 Tổng quan về mô hình zero trust

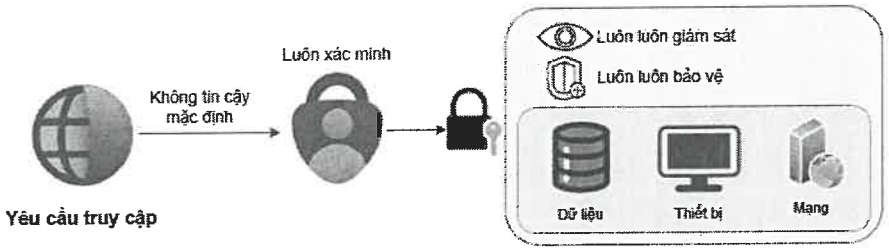
Zero trust yêu cầu liên tục xác minh người dùng, thiết bị và yêu cầu truy cập, không cấp sự tin cậy mặc định cho bất kỳ đối tượng nào. Theo cách tiếp cận này, việc bảo vệ hệ thống thực hiện thông qua xác thực, sau đó kiểm soát truy cập, liên tục giám sát và đánh giá rủi ro. Các thành phần này sẽ kiểm tra trước khi cho phép người dùng truy cập và sử dụng tài nguyên của hệ thống. [13]

Trong phạm vi đề án, cách tiếp cận Zero trust của tác giả được thể hiện bắt đầu từ việc hệ thống yêu cầu xác minh danh tính bằng các phương pháp xác thực không dùng mật khẩu trước khi cấp quyền truy cập. Sau quá trình xác thực, hệ thống ghi nhận vị trí địa lý, địa chỉ ip và User-Agent để tiếp tục giám sát phiên đăng nhập và phát hiện thay đổi bất thường. Bên cạnh đó, cơ chế phân quyền, giới hạn thời gian trong phiên đăng nhập cũng là yếu tố tác giả áp dụng. Trên đây là một số nguyên tắc của Zero Trust nhưng tác giả chưa triển khai đầy đủ toàn bộ mô hình. [13]

1.4.1 Kiến trúc tổng quan của zero trust

Zero trust thay đổi cách bảo vệ hệ thống bằng việc yêu cầu mọi yêu cầu truy cập đều phải được kiểm tra, thay vì mặc định tin cậy người dùng, thiết bị hoặc phiên đăng nhập đã được xác thực trước đó. Trong cách tiếp cận bảo mật thông thường, hệ thống thường tập trung kiểm soát quyền truy cập ở lớp bên ngoài, chẳng hạn như tường lửa, vpn. Khi người dùng hoặc thiết bị đã vượt qua lớp kiểm soát này và vào được mạng nội bộ, chúng thường được xem là có mức độ tin cậy cao hơn. Nhưng có

một nhược điểm của cách tiếp cận này là phát sinh rủi ro nếu tài khoản, thiết bị hoặc một thành phần bên trong hệ thống bị xâm nhập.



Hình 1.2 Tổng quát mô hình zero trust

Khác với cách tiếp cận trên, zero trust yêu cầu từng đối tượng và từng lần truy cập phải được xác minh trước khi cấp quyền, không phụ thuộc vào việc yêu cầu đó được gửi từ mạng nội bộ hay từ môi trường bên ngoài. Trong kiến trúc zero trust, từng yêu cầu muốn được truy cập trước hết được gửi đến lớp kiểm soát chính sách truy cập. Lớp này thực hiện việc tổng hợp dữ liệu và xem xét các giá trị thu được như danh tính người dùng, quyền truy cập, thiết bị sử dụng, dấu vân tay thiết bị, địa chỉ ip và nâng cao hơn là ngữ cảnh đăng nhập, hành vi thao tác của người dùng trong hệ thống. Dựa trên các thông tin đó, hệ thống đưa ra quyết định cấp quyền truy cập hoặc ngăn chặn yêu cầu đối với tài nguyên.

Nói cách khác, người dùng không được cấp quyền chỉ vì đã đăng nhập thành công hoặc đang ở trong mạng nội bộ. Quyền truy cập cần được kiểm tra theo từng yêu cầu và có thể thay đổi tùy theo mức độ tin cậy tại thời điểm truy cập. Ví dụ, nếu người dùng đăng nhập từ thiết bị quen thuộc và địa chỉ ip đáng tin cậy, hệ thống có thể cho phép truy cập bình thường. Ngược lại, nếu phát hiện thiết bị lạ, vị trí bất thường hoặc hành vi đăng nhập có rủi ro, hệ thống có thể yêu cầu xác minh bổ sung hoặc từ chối truy cập. Đây chính là tinh thần của nguyên tắc "không tin tưởng mặc định, luôn xác minh" trong mô hình zero trust. [14]

1.4.2 Các rủi ro thực tế cho thấy sự cấp thiết trong việc ứng dụng Zero Trust

Một ví dụ điển hình là sự cố rò rỉ dữ liệu của Equifax vào năm 2017. Vào ngày 7/9/2017, Equifax công bố rằng hệ thống của công ty đã bị tấn công, làm lộ dữ liệu cá nhân của khoảng 143 triệu người tiêu dùng. Tuy nhiên, sự cố này chỉ được phát hiện vào ngày 29/7/2017, trong khi kẻ tấn công đã âm thầm truy cập vào hệ thống và lấy được dữ liệu người dùng từ nhiều tháng trước đó. Trường hợp này cho thấy một vấn đề phổ biến trong nhiều sự cố bảo mật: hệ thống có thể bị xâm nhập trong thời gian dài trước khi đội ngũ an ninh phát hiện ra bất thường. [15]

Một trường hợp khác là mã độc tống tiền Wanna cry và các biến thể của nó, từng ảnh hưởng đến nhiều tổ chức tại hơn 150 quốc gia. Trong đó, thành phố Atlanta của Hoa Kỳ đã bị gián đoạn hoạt động trong hơn một tuần và phải chi khoảng 17 triệu usd cho việc khắc phục, phục hồi hệ thống. Sự cố này cho thấy khi mã độc có thể lan truyền giữa các hệ thống quan trọng, Hậu quả của sự cố không chỉ là mất mát dữ liệu mà còn có thể ảnh hưởng đến hoạt động vận hành. Mô hình zero trust có thể hỗ trợ hạn chế rủi ro này thông qua việc phân tách tài nguyên, kiểm soát truy cập chặt chẽ và giảm khả năng mã độc di chuyển tự do trong hệ thống. [16]

Các sự cố rò rỉ dữ liệu không chỉ gây thiệt hại về mặt kỹ thuật, mà còn để lại hậu quả nghiêm trọng đối với hoạt động kinh doanh và uy tín của tổ chức. Trong thực tế, nhiều lãnh đạo cấp cao tại các tổ chức lớn như Bangladesh Bank, Sony hoặc Target đã phải từ chức sau các sự cố bảo mật. Trường hợp Văn phòng Quản lý Nhân sự Hoa Kỳ cũng cho thấy mức độ nghiêm trọng của vấn đề, khi hơn 20 triệu hồ sơ liên quan đến nhân viên chính phủ, nhà thầu và các đối tượng khác bị lộ, dẫn đến việc người đứng đầu cơ quan này từ chức. [17]

Bên cạnh tổn thất về nhân sự và uy tín, chi phí khắc phục các sự cố này cũng rất lớn. Chẳng hạn, riêng thỏa thuận dàn xếp giữa Equifax và Ủy ban Thương mại Liên bang Hoa Kỳ đã có giá trị khoảng 575 triệu USD. Những ví dụ này cho thấy việc bảo vệ dữ liệu nhạy cảm và ngăn chặn dữ liệu bị đưa ra ngoài bởi tác nhân độc hại là yêu cầu rất quan trọng. Theo định hướng zero trust, hệ thống cần kiểm soát chặt

chẽ quyền truy cập, áp dụng nguyên tắc cấp quyền tối thiểu và giám sát liên tục để giảm nguy cơ rò rỉ dữ liệu quan trọng, bao gồm cả tài sản trí tuệ và dữ liệu ảnh hưởng đến doanh thu tương lai của tổ chức. [17]

1.4.3 Lợi ích đem đến của Zero trust

Zero Trust mang lại nhiều lợi ích, góp phần cải thiện mức độ an toàn hệ thống. Trước hết, mô hình này giúp tăng khả năng giám sát và phát hiện rủi ro trong hệ thống thông qua việc luôn luôn theo dõi người dùng, thiết bị, phiên đăng nhập và các hành vi bất thường. Nhờ đó, hệ thống có thể phát hiện sớm các dấu hiệu xâm nhập hoặc điểm yếu bảo mật cần được xử lý.

Thứ hai, zero trust góp phần hạn chế sự lan truyền của mã độc trong hệ thống. Do quyền truy cập được kiểm soát theo từng tài nguyên và không mặc định tin tưởng các thành phần bên trong mạng, kẻ tấn công khó có thể di chuyển tự do từ khu vực này sang khu vực khác nếu đã xâm nhập thành công một phần hệ thống. [18]

Thứ ba, zero trust giúp tăng cường khả năng phối hợp trong quản lý bảo mật. Khi các chính sách truy cập, nhật ký hoạt động và dữ liệu giám sát được quản lý rõ ràng hơn, các bộ phận liên quan có thể xác định trách nhiệm và xử lý sự cố hiệu quả hơn, thay vì đổ lỗi qua lại giữa các nhóm vận hành. [18]

Thứ tư, mô hình này giúp hệ thống hiểu rõ hơn về dữ liệu và tài nguyên cần bảo vệ. Việc kiểm soát truy cập theo danh tính, thiết bị và ngữ cảnh giúp tổ chức nhận biết dữ liệu nào đang được truy cập, ai truy cập và truy cập trong điều kiện nào.

Cuối cùng, zero trust giúp hạn chế nguy cơ rò rỉ dữ liệu nhạy cảm. Bằng cách áp dụng xác thực chặt chẽ, cấp quyền tối thiểu và giám sát liên tục, hệ thống có thể giảm khả năng dữ liệu quan trọng bị truy cập hoặc đưa ra ngoài bởi các tác nhân độc hại.

1.5 So sánh hình thức đăng nhập bằng mật khẩu và các giải pháp không sử dụng mật khẩu

Bảng 1.2 So sánh đăng nhập bằng mật khẩu và không sử dụng mật khẩu

	Đăng nhập sử dụng mật khẩu	**Đăng nhập không sử dụng mật khẩu**
Bảo mật	Dễ bị lộ, tái sử dụng, dò đoán hoặc bị khai thác thông qua tấn công giả mạo. Ngoài ra, mật khẩu có thể bị thu thập bởi các phần mềm độc hại như keylogger khi người dùng nhập thông tin đăng nhập. Hiệu quả bảo vệ tài khoản sẽ giảm nếu người dùng lựa chọn mật khẩu đơn giản hoặc dễ bị suy đoán.	Giảm sự phụ thuộc vào mật khẩu, giảm nguy cơ bảo mật do mật khẩu yếu hoặc bị lộ mật khẩu.
Trải nghiệm người dùng	Việc ghi nhớ mật khẩu, thậm chí mật khẩu quá phức tạp khiến người dùng dễ quên, nhập sai hoặc phải thực hiện đặt lại mật khẩu.	Người dùng sử dụng ngay mã xác thực dùng một lần, liên kết đăng nhập hoặc sử dụng cách xác thực bằng những cơ chế mở khóa phổ biến trên thiết bị, chẳng hạn quét vân tay hoặc nhận diện khuôn mặt.
Chi phí vận hành	Phát sinh chi phí liên quan đến việc quản lý mật	Hạn chế số lượng yêu cầu hỗ trợ về khôi phục hoặc

	khẩu, hỗ trợ đặt lại mật khẩu, xử lý tài khoản bị khóa hoặc khôi phục tài khoản.	thay đổi thông tin đăng nhập nhờ không còn sử dụng mật khẩu làm phương thức xác thực.
Khả năng triển khai	Dễ triển khai, quen thuộc với người dùng và phù hợp với hầu hết hệ thống hiện nay.	Khả năng triển khai và mức độ phức tạp khác nhau tùy theo từng phương pháp.
Tính tương thích	Giải pháp có khả năng hoạt động trên nhiều nền tảng, trình duyệt và loại thiết bị khác nhau.	Phụ thuộc vào thiết bị, trình duyệt, ứng dụng xác thực hoặc ứng dụng bên thứ ba có hỗ trợ hay không.

1.6 Kết luận chương I

Chương I đã trình bày cơ sở lý luận về xác thực người dùng trong ứng dụng web, bao gồm khái niệm xác thực, phân quyền và vai trò của xác thực trong việc bảo vệ hệ thống. Chương cũng làm rõ phương thức xác thực bằng mật khẩu, cơ chế hoạt động, các rủi ro bảo mật và những bất cập về trải nghiệm người dùng trong thực tế triển khai.

Từ những hạn chế của mật khẩu truyền thống, chương đã giới thiệu xác thực không mật khẩu như xác thực bằng mã xác thực OTP, mã xác thực TOTP, liên kết đăng nhập và đăng nhập bằng passkey. Các phương thức này được phân tích và so sánh với xác thực bằng mật khẩu trên các khía cạnh như bảo mật, trải nghiệm người dùng, chi phí vận hành và khả năng triển khai. Bên cạnh đó, mô hình zero trust cũng được đề cập như một định hướng bảo mật phù hợp với yêu cầu xác minh liên tục và kiểm soát truy cập theo ngữ cảnh.

CHƯƠNG II. PHÂN TÍCH YÊU CẦU VÀ ĐỀ XUẤT CÁC GIẢI PHÁP XÁC THỰC KHÔNG MẬT KHẨU

2.1 Phân tích yêu cầu về giải pháp xác thực không mật khẩu

2.1.1 Yêu cầu với giải pháp xác thực không mật khẩu

- Yêu cầu về phần cứng

Trong phạm vi đề án, Trong quá trình thử nghiệm, tác giả chạy ứng dụng trên máy tính cá nhân và sử dụng thiết bị này như một máy chủ cục bộ và cần cài đặt hệ quản trị cơ sở dữ liệu Mysql, biên dịch mã nguồn java và khởi chạy ứng dụng web. Đồng thời hỗ trợ tích hợp các thư viện thông qua công cụ quản lý thư viện như Maven, phục vụ việc xây dựng các phương thức xác thực không mật khẩu.

Về phía máy khách, Để sử dụng chức năng này, thiết bị của người dùng phải hỗ trợ phương thức mở khóa như nhập mã PIN hoặc bằng sinh trắc học, bao gồm quét khuôn mặt hoặc vân tay.

- Yêu cầu về phần mềm

Trong phạm vi đề án, tác giả xây dựng ứng dụng web và kiểm thử trên máy tính cá nhân. Bao gồm biên dịch mã nguồn Java, chạy ứng dụng web và sử dụng Mysql làm cơ sở dữ liệu. Ứng dụng được tác giả phát triển trên Spring boot phiên bản 4.0.5, Java 25 và hệ điều hành windows 11.

Bên cạnh đó, Maven được sử dụng để quản lý các thư viện phụ thuộc thông qua tệp pom.xml. Đồng thời, maven hỗ trợ biên dịch mã nguồn, đóng gói ứng dụng thành file jar để khởi chạy ứng dụng web spring boot.

Đối với hệ quản trị cơ sở dữ liệu, tác giả cài đặt Mysql phiên bản 8.4.9 thuộc nhánh LTS. Tác giả lựa chọn phiên bản LTS nhằm ưu tiên sự ổn định, hỗ trợ lâu dài.

Các phương pháp đăng nhập khác như xác thực bằng mã OTP hoặc liên kết đăng nhập yêu cầu thiết bị người dùng cần có kết nối mạng Internet ổn định để đảm bảo quá trình gửi mã OTP và liên kết đăng nhập không bị gián đoạn.

Phương pháp đăng nhập bằng mã xác thực TOTP yêu cầu người dùng cài đặt ứng dụng xác thực như Google authenticator để lấy mã code gồm 6 chữ số ngẫu nhiên, tự động thay đổi sau 30 giây.

Phương pháp đăng nhập bằng ứng dụng bên thứ ba, ví dụ như Google thì yêu cầu người dùng phải có sẵn tài khoản Google nói riêng và trên các ứng dụng bên thứ ba nói chung để thực hiện xác thực và cung cấp cho ứng dụng chính những thông tin cần thiết để nhận diện tài khoản người dùng và hoàn tất quá trình đăng nhập.

2.1.2 Yêu cầu về khả năng tích hợp với ứng dụng web

Đối với phương pháp xác thực bằng mã OTP và liên kết đăng nhập, phía back-end cần bao gồm các service đảm nhiệm việc tạo mã OTP hoặc tạo liên kết đăng nhập tương ứng với từng tài khoản. Các thông tin xác thực này sau đó được kiểm tra tính hợp lệ khi người dùng nhập mã OTP hoặc truy cập liên kết đăng nhập.

Bên cạnh đó, cần có service gửi email được tích hợp với dịch vụ gửi thư bên ngoài thông qua gmail SMTP để gửi mã xác thực OTP hoặc liên kết đăng nhập đến địa chỉ email.

Đối với phương thức xác thực TOTP, server cần sinh khóa bí mật riêng cho từng người dùng và tạo ra mã QR dựa vào khóa bí mật đó để người dùng quét bằng ứng dụng xác thực như google authenticator. Sau khi thiết lập thành công, server lưu trữ khóa bí mật của người dùng và sử dụng khóa này để kiểm tra mã xác thực TOTP khi người dùng thực hiện đăng nhập.

Với phương pháp đăng nhập bằng mã xác thực TOTP, phía back-end cần xây dựng service đảm nhiệm việc tạo mã QR dựa trên khóa bí mật, đây là khóa bí mật đi kèm với mỗi tài khoản người dùng khi đăng ký mới. Mục đích là để người dùng quét mã QR này trên ứng dụng Google authenticator để nhận mã xác thực phía máy khách.

Sau đó, cần có một service có nhiệm vụ tạo mã xác thực phía back-end và kiểm tra mã đó khi người dùng đăng nhập.

Với phương pháp xác thực bằng passkey, phía back-end cần tích hợp thư viện spring-security-webauthn để hỗ trợ xử lý quá trình đăng ký và xác thực passkey. Trong quá trình đăng ký, phía front-end gọi API để tạo thông tin xác thực mới cho tài khoản người dùng và gửi dữ liệu đăng ký đến back-end. Trong quá trình đăng nhập, front-end tiếp tục gọi API để gửi dữ liệu xác thực đến back-end. Sau khi xác thực thành công, phía back-end tiếp nhận đối tượng xác thực và lấy thông tin người dùng để phục vụ quá trình đăng nhập vào ứng dụng. [19]

Với phương pháp xác thực bằng ứng dụng bên thứ ba như đăng nhập bằng tài khoản Google, phía back-end cần cấu hình các thông tin cần thiết như client ID, client secret, redirect URI và các endpoint liên quan để tích hợp với nhà cung cấp xác thực bên thứ ba, ví dụ như Google. Sau khi xác thực thành công, back-end tiếp nhận thông tin người dùng do nhà cung cấp trả về để phục vụ quá trình đăng nhập vào ứng dụng. [20]

2.1.3 Yêu cầu về bảo mật và khả năng mở rộng

a) Tính bảo mật:

- Giai đoạn xác thực:

Đối với phương pháp xác thực bằng liên kết đăng nhập và mã OTP, mã xác thực của cả hai phương cần được chuyển thành giá trị băm rồi mới được lưu trực tiếp vào cơ sở dữ liệu nhằm hạn chế rủi ro lộ thông tin xác thực. Mã xác thực cần có thời gian sử dụng giới hạn, chỉ được phép sử dụng duy nhất một lần và không được sử dụng lại sau khi người dùng xác thực thành công, nhằm phòng chống nguy cơ tấn công phát lại.

Với phương pháp đăng nhập bằng mã xác thực TOTP, khóa bí mật được sinh ngẫu nhiên nhằm tạo ra giá trị có độ ngẫu nhiên cao và khó dự đoán. Khóa bí mật này được sử dụng để tạo mã QR, giúp người dùng quét và liên kết tài khoản với ứng dụng

xác thực như Google authenticator. Sau khi liên kết thành công, khóa bí mật được lấy ra để kiểm tra mã TOTP trong các lần đăng nhập sau và không được hiển thị lại cho phía máy khác (client) nhằm hạn chế nguy cơ lộ khóa bí mật và ngăn chặn việc kẻ tấn công tự tạo mã TOTP hợp lệ.

Với phương pháp xác thực bằng ứng dụng bên thứ ba như đăng nhập bằng tài khoản Google, các thông tin cấu hình nhạy cảm như client ID, client secret cần được lưu trữ an toàn ở phía back-end và không được hiển thị cho phía máy khách hoặc đưa trực tiếp lên mã nguồn công khai. Bên cạnh đó, redirect URI cần được cấu hình chính xác theo địa chỉ của ứng dụng nhằm hạn chế nguy cơ chuyển hướng không hợp lệ hoặc bị lợi dụng tấn công open redirect. Ứng dụng cũng chỉ nên yêu cầu các phạm vi quyền truy cập cần thiết, chẳng hạn như email và họ tên, để giảm thiểu rủi ro lộ lọt thông tin người dùng. [21]

Với phương pháp đăng nhập bằng passkey, ứng dụng web cần được chạy trong ngữ cảnh an toàn như HTTPS hoặc localhost. Quá trình xác thực sử dụng các dữ liệu như challenge, credential và chữ ký được tạo bởi khóa riêng trên thiết bị của người dùng. Đồng thời, passkey được ràng buộc với Relying Party ID (RP ID) tương ứng với tên miền của ứng dụng, qua đó ngăn passkey được sử dụng trên các tên miền giả mạo hoặc không hợp lệ. [22]

Với hai hình thức xác minh dựa trên mã dùng một lần là OTP và TOTP, ứng dụng cần có cơ chế giới hạn số lần xác thực thất bại và tạm khóa tài khoản khi vượt quá số lần cho phép. Cơ chế này phát hiện dấu hiệu bất thường như ngăn chặn hành vi thử sai mã xác thực nhiều lần, hạn chế nguy cơ tấn công brute force.

- Giao đoạn quản lý phiên đăng nhập:

Khi quá trình kiểm tra thông tin đăng nhập hoàn tất và cho kết quả hợp lệ, các thông tin định danh của phiên đăng nhập như địa chỉ IP và chuỗi user-agent được dùng để phát hiện dấu hiệu truy cập bất thường. Bên cạnh đó, mỗi phiên đăng nhập cần có thời gian hiệu lực nhất định; khi phiên hết hạn, người dùng phải đăng nhập lại

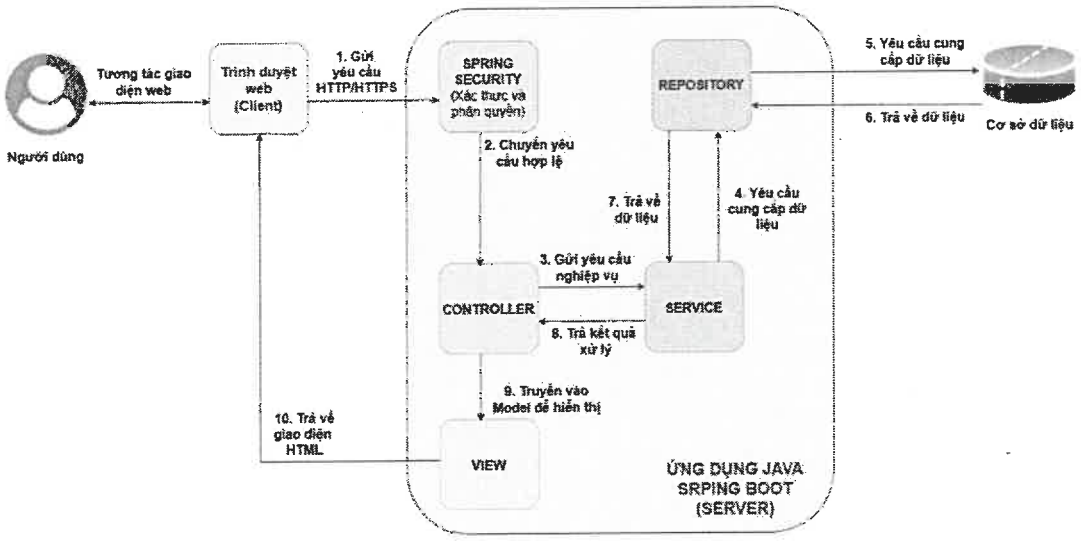
nhằm hạn chế nguy cơ phiên đăng nhập bị lợi dụng. Các cơ chế này góp phần thể hiện nguyên tắc của zero trust, trong đó phiên đăng nhập không được tin tưởng tuyệt đối sau khi xác thực thành công mà cần tiếp tục được giới hạn thời gian, giám sát và kiểm tra theo ngữ cảnh truy cập. [13]

b) Khả năng mở rộng:

Với phương pháp xác thực bằng mã xác thực OTP, mã xác thực TOTP và liên kết đăng nhập, tác giả xây dựng các service riêng để xử lý việc sinh mã, tạo liên kết, lưu trữ và kiểm tra thông tin xác thực. Với xác thực bằng ứng dụng bên thứ ba, tác giả cài đặt thư viên spring-security-oauth2-client để hỗ trợ đăng nhập thông qua các nhà cung cấp xác thực bên thứ ba. Còn phương pháp đăng nhập bằng passkey, tác giả cài đặt thư viện spring-security-webauthn để hỗ trợ quá trình đăng ký và xác thực passkey. Việc tách riêng phần xử lý của từng phương thức xác thực giúp tác giả có thể mở rộng thực hiện các phương thực xác thực khác mà không gây xung đột mã nguồn, từ đó quá trình kiểm thử, sửa lỗi được thực hiện thuận tiện hơn.

2.2 Đề xuất và định hướng kiến trúc, công nghệ

2.2.1 Định hướng kiến trúc client server



Hình 2.1 Tổng quan kiến trúc ứng dụng Java Spring Boot theo mô hình MVC kết hợp các lớp xử lý

Trong phạm vi đề án, ứng dụng được định hướng xây dựng kiến trúc client và server. Trong kiến trúc này, trình duyệt đảm nhiệm việc trình bày nội dung cho người truy cập và trao đổi dữ liệu với máy chủ bằng các giao thức phổ biến hiện nay, ví dụ như: HTTP hoặc HTTPS. Phía máy chủ, cụ thể là back-end được xây dựng bằng java spring boot, tiếp nhận yêu cầu, kiểm tra danh tính và áp dụng quyền truy cập tương ứng thông qua spring security, sau đó chuyển yêu cầu hợp lệ đến controller để điều phối luồng xử lý.

Bên trong ứng dụng, mã nguồn được tổ chức theo mô hình nhiều lớp kết hợp MVC. Các yêu cầu từ phía người dùng trước tiên được Controller ghi nhận, sau đó chuyển sang Service để thực hiện logic nghiệp vụ của ứng dụng. Khi cần tương tác với dữ liệu, Service sử dụng Repository để tìm kiếm, bổ sung hoặc cập nhật các bản ghi. Tiếp theo, view nhận dữ liệu từ controller gửi sang để hiển thị giao diện và trả về trang HTML cho trình duyệt người dùng.

2.2.2 Định hướng công nghệ

a) Định hướng công nghệ cho front-end:

Ứng dụng web trong đề án được xây dựng bằng HTML, CSS, JavaScript và Thymeleaf. HTML được dùng để tạo nên bố cục và các thành phần hiển thị của giao diện, CSS được dùng để thiết lập bố cục và hình thức hiển thị của các thành phần trên trang. Trong khi đó, JavaScript thực hiện các thao tác tại trình duyệt, bao gồm đối chiếu giá trị người dùng nhập và trao đổi dữ liệu với back-end.

Thymeleaf được tích hợp với Spring MVC để hiển thị dữ liệu từ phía server trên các trang giao diện. Thymeleaf cho phép controller truyền dữ liệu trực tiếp sang các trang giao diện HTML để hiển thị giao diện động. Cách tiếp cận này giúp đề án không cần phát triển một ứng dụng front-end độc lập, qua đó tập trung hơn vào cơ chế xác thực không mật khẩu và xử lý nghiệp vụ phía server.

b) Định hướng công nghệ cho cho back-end:

Phần xử lý phía back-end được phát triển bằng Java Spring Boot. Spring boot hỗ trợ cấu hình, tổ chức mã nguồn rõ ràng theo mô hình MVC và dễ dàng tích hợp với các thành phần như spring security, thymeleaf và mysql.

Bên cạnh đó, Spring Security đảm nhiệm xác thực, phân quyền, quản lý phiên đăng nhập và tích hợp các phương thức đăng nhập không mật khẩu. Việc sử dụng spring security giúp đề án tận dụng các cơ chế bảo mật có sẵn của spring security, thay vì phải tự xây dựng toàn bộ các chức năng này từ đầu, qua đó hạn chế mức độ phức tạp trong khâu triển khai, đồng thời hỗ trợ quá trình duy trì và chỉnh sửa hệ thống sửa lỗi nếu có lỗi phát sinh.

c) Ứng dụng Spring security trong xử lý đăng nhập không mật khẩu:

Spring Security hỗ trợ tổ chức và xử lý các phương thức xác thực không mật khẩu bao gồm các thành phần và chức năng sau: [23]

Authentication filter: Là bộ lọc trong Spring security có nhiệm vụ tiếp nhận các yêu cầu đăng nhập từ người dùng. Bộ lọc này trích xuất thông tin xác thực từ request, chẳng hạn như email, mật khẩu, mã xác thực OTP hoặc mã xác thực TOTP, mã xác thực từ liên kết đăng nhập. Từ những dữ liệu này, bộ lọc tạo đối tượng Authentication và chuyển đến Authentication manager để tiếp tục quá trình xác thực.

Authentication manager: Trong Spring Security, Authentication manager tiếp nhận đối tượng Authentication do Authentication filter tạo ra và thực hiện điều phối quá trình xác thực. Đối tượng này được chuyển đến Authentication provider phù hợp để xác thực thông tin đăng nhập.

Nếu quá trình xác thực thành công, một đối tượng Authentication được trả về, bao gồm: Principal cho thông tin đăng nhập, credentials chứa thông tin xác thực, authorities thể hiện các vai trò hoặc quyền hạn và details bao gồm các thông chi tiết về tài khoản. Trong trường hợp xác thực không thành công, Spring security sẽ phát sinh ngoại lệ thông báo xác thực thất bại.

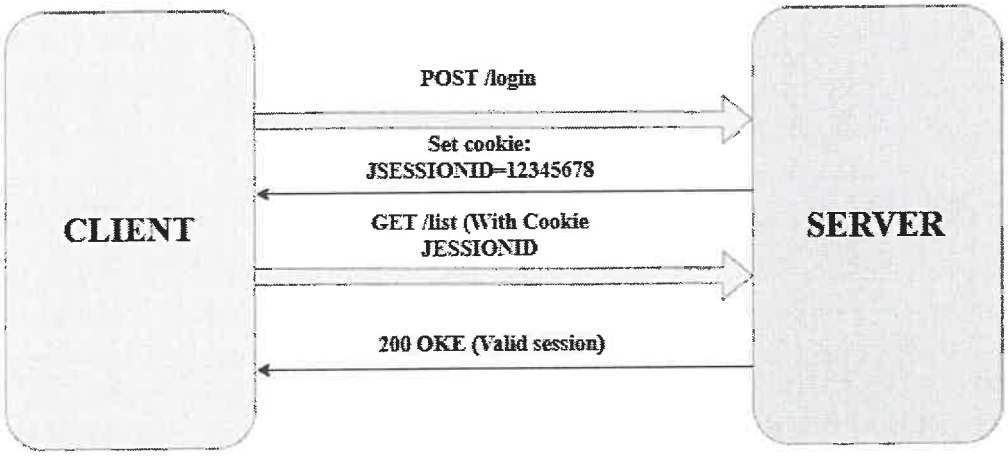
Authentication provider: là thành phần thực hiện việc kiểm tra thông tin xác thực sau khi yêu cầu xác thực từ người dùng gửi lên. Trong spring security, thông tin xác thực sẽ được đóng gói thành một đối tượng Authentication, sau đó authentication manager chuyển đối tượng này đến Authentication provider phù hợp để xử lý.

Tùy theo phương thức xác thực, Authentication provider có nhiệm vụ kiểm tra thông tin xác thực tương ứng như kiểm tra mã xác thực OTP, mã xác thực TOTP hoặc mã xác thực từ liên kết đăng nhập.

Truy xuất thông tin xác thức thông qua đối tượng Authentication:

Đối tượng Authentication bao gồm các thông tin bao gồm danh tính, trạng thái xác thực và các quyền truy cập được cập. Trong đó, principal biểu diễn đối tượng chứa các dữ liệu cần thiết như địa chỉ email, mật khẩu, trạng thái tài khoản và danh sách quyền.

Duy trì phiên đăng nhập trong ứng dụng web:



Hình 2.2 Minh họa quá trình duy trì phiên làm việc giữa client và server

Để máy chủ nhận diện đúng phiên làm việc của người dùng, trình duyệt sử dụng cookie JSESSIONID. Người dùng sau khi đăng nhập phía trình duyệt lưu JSESSIONID và gửi kèm giá trị này trong các request tiếp theo. Nhờ đó, máy chủ có thể xác định lại phiên làm việc của người dùng. Trong phạm vi đề án, khi người dùng đăng xuất, phiên đăng nhập sẽ bị hủy, cookie JSESSIONID được xóa và thông tin xác

thực hiện tại cũng được loại bỏ, từ đó kết thúc trạng thái đăng nhập của người dùng. [24]

2.2.3 Định hướng tích hợp

a) Lý do chọn các phương pháp xác thực không mật khẩu và phạm vi áp dụng:

- Phương pháp xác thực bằng mã OTP:

Lý do sử dụng vì phù hợp với mục tiêu xây dựng chức năng đăng nhập không mật khẩu trong đề án. Thay vì sử dụng mật khẩu cố định, phía server sẽ sinh ra mã xác thực OTP dùng một lần để người dùng xác thực khi đăng nhập. Đăng nhập bằng mã xác thực OTP hỗ trợ bảo vệ tài khoản tốt hơn khi người dùng thực hiện đăng nhập. Mỗi mã xác thực OTP chỉ có thời gian sống ngắn và chỉ được sử dụng một lần, từ đó hạn chế rủi ro mã bị sử dụng lại hoặc bị lợi dụng sau khi hết hạn.

Phạm vi áp dụng trong đề án bao gồm việc khi người dùng yêu cầu đăng nhập bằng mã OTP, server sẽ sinh mã xác thực gồm 6 chữ số, lưu thông tin mã kèm thời gian hết hạn và trạng thái sử dụng. Người dùng nhập mã xác thực OTP trên giao diện xác thực. Server sẽ kiểm tra mã xác thực bao gồm: Thời gian hết hạn và trạng thái sử dụng của mã. Trường hợp mã mà người dùng nhập khớp với mã trên cơ sở dữ liệu thì hệ thống coi xác thực là hợp lệ. Nếu mã không đúng, hết hạn hoặc đã được sử dụng, server sẽ từ chối đăng nhập.

- Phương pháp xác thực bằng liên kết đăng nhập:

Lý do tác giả sử dụng phương pháp này là thay vì yêu cầu người dùng sử dụng mật khẩu hoặc mã xác thực thủ công, server sẽ tạo một liên kết đăng nhập tạm thời để người dùng sử dụng. Liên kết đăng nhập giúp đơn giản hóa thao tác đăng nhập, vì người dùng chỉ cần nhấn vào liên kết. Liên kết này có chứa mã xác thực tạm thời, có thời hạn sử dụng ngắn và chỉ phục vụ cho quá trình xác thực đăng nhập, từ đó hạn chế rủi ro bị lợi dụng sau khi hết hiệu lực.

Trong phạm vi đề án, tác giả áp dụng bao gồm khi người dùng xác thực bằng liên kết đăng nhập, server tạo mã xác thực và lưu kèm thời gian hết hạn và gửi đến địa chỉ email. Khi người dùng nhấn vào liên kết, server sau đó đối chiếu mã xác thực này với mã xác thực đã lưu trong cơ sở dữ liệu, cụ thể là đối chiếu mã xác thực, cộng với kiểm tra thời gian hết hạn và trạng thái sử dụng. Nếu mã xác thực hợp lệ, người dùng được xác thực thành công và truy cập vào ứng dụng. Nếu mã xác thực không hợp lệ, hết hạn hoặc đã được sử dụng, server sẽ từ chối đăng nhập.

- Phương pháp đăng nhập bằng mã xác thực TOTP:

Lý do tác giả lựa chọn phương pháp này vì đây là hình thức xác thực không mật khẩu có tính bảo mật cao vì Mã TOTP được tạo mới sau mỗi khoảng thời gian xác định và chỉ có thời gian sống ngắn.

Phạm vi áp dụng trong đề án bao gồm trước khi xác thực bằng mã TOTP, người dùng cần quét mã QR để lấy mã TOTP trong ứng dụng Google authenticator. Ở giai đoạn đăng nhập, người dùng nhập mã TOTP từ app Google authenticator vào giao diện xác thực. Server sẽ dựa trên khóa bí mật đã lưu và thời gian hiện tại để kiểm tra mã người dùng nhập. Nếu mã TOTP trùng khớp với giá trị do hệ thống tính toán trong khoảng thời gian hiện tại, quá trình đăng nhập được chấp nhận. Nếu mã không đúng hoặc không còn hợp lệ tại thời điểm kiểm tra, server sẽ từ chối đăng nhập.

- Phương pháp xác thực bằng ứng dụng bên thứ ba:

Lý do tác giả sử dụng phương pháp này vì giúp người dùng đăng nhập nhanh chóng thông qua các tài khoản phổ biến như Google. Thay vì phải tạo và ghi nhớ mật khẩu riêng lẻ, người dùng xác thực bằng ứng dụng bên thứ ba giúp không chỉ đơn giản hóa việc đăng nhập mà còn giảm thao tác nhập thông tin thủ công và tận dụng cơ chế xác thực sẵn có của tài khoản ứng dụng bên thứ ba. Điều này phù hợp với ứng dụng web vì Do phần lớn người dùng đã sở hữu tài khoản Google, quá trình đăng nhập có thể được thực hiện nhanh chóng và thuận tiện hơn.

Phạm vi áp dụng trong đề án bao gồm ứng dụng web sẽ chuyển hướng người dùng đến trang đăng nhập của nhà cung cấp tương ứng. Khi quá trình kiểm tra thông tin đăng nhập hoàn tất và hợp lệ, nhà cung cấp sẽ trả về thông tin cần thiết như email, họ tên và thông tin định danh tài khoản. Thông tin này được máy chủ tra cứu trong cơ sở dữ liệu nhằm kiểm tra sự tồn tại của tài khoản. Trường hợp người dùng đăng nhập lần đầu, hệ thống tạo tài khoản mới từ những thông tin được nhà cung cấp trả về, sau đó tạo phiên đăng nhập và cấp quyền cho người dùng truy cập bên trong hệ thống

- Phương pháp đăng nhập bằng Passkey:

Lý do tác giả sử dụng phương pháp này xuất phát từ việc người dùng có thể xác thực bằng các phương thức xác thực có sẵn trên thiết bị như vân tay, khuôn mặt, mã PIN hoặc khóa bảo mật. Trong phạm vi đề án, passkey phù hợp với mục tiêu xây dựng cơ chế đăng nhập không mật khẩu, giúp hạn chế rủi ro liên quan đến mật khẩu như quên mật khẩu, mật khẩu yếu hoặc bị đánh cắp. Bên cạnh đó, Passkey còn có ưu điểm là khóa xác thực được lưu an toàn trên thiết bị của người dùng, quá trình đăng nhập nhanh chóng và khó bị giả mạo hơn so với các phương thức nhập mã hoặc mật khẩu truyền thống. Ngoài ra, Passkey còn giúp cải thiện trải nghiệm người dùng vì quá trình đăng nhập nhanh và ít thao tác hơn.

Phạm vi áp dụng trong đề án bao gồm trước khi có thể xác thực bằng Passkey, người dùng cần đăng ký Passkey cho tài khoản của mình thông qua thiết bị cá nhân. Khi người dùng đăng nhập sẽ cần xác thực trên thiết bị thông qua các hình thức như nhận diện dấu vân tay, dữ liệu khuôn mặt hoặc truyền thống như mã PIN, sau đó server kiểm tra kết quả xác thực được gửi về. Nếu thông tin xác thực hợp lệ và khớp với tài khoản đã đăng ký, quá trình xác thực thành công và hệ thống cho phép truy cập. Trường hợp xác thực không hợp lệ, server sẽ từ chối đăng nhập.

b) Tích hợp các thư viện hỗ trợ xác thực:

Tích hợp dịch vụ gửi mail để gửi mã xác thực OTP và liên kết đăng nhập: Trong phạm vi đề án, dịch vụ gửi email được tích hợp nhằm hỗ trợ quá trình xác thực không mật khẩu, cụ thể là gửi mã xác thực OTP và liên kết đăng nhập đến người dùng. Đóng vai trò là kênh trung gian để gửi thông tin xác thực do server tạo ra.

Để sử dụng chức năng gửi mail, tác giả cài đặt thư viện spring-boot-starter-mail, thư viện này cung cấp các thành phần cần thiết để Spring Boot hỗ trợ gửi email, trong đó có đối tượng JavaMailSender được sử dụng trong quá trình gửi thư.

Chức năng gửi email được triển khai thông qua lớp Email service. Lớp này có nhiệm vụ gửi email xác thực. Tác giả không tự xây dựng máy chủ SMTP riêng mà kết nối đến dịch vụ SMTP của Gmail.

```
@Service
@RequiredArgsConstructor
public class EmailServiceImpl implements EmailService {

    private final JavaMailSender mailSender;

    @Override
    public void sendOtp(String toEmail, String otp) {
        SimpleMailMessage message = new SimpleMailMessage();
        message.setTo(toEmail);
        message.setSubject("Your login code");
        message.setText("Your login code is: " + otp);

        mailSender.send(message);
    }

    @Override
    public void sendUserTokenLink(String toEmail, String link, UserTokenType userTokenType) {
        SimpleMailMessage message = new SimpleMailMessage();
        message.setTo(toEmail);
        message.setSubject(userTokenType.getSubject());
        message.setText(userTokenType.getMessage() + link);

        mailSender.send(message);
    }
```

Hình 2.3 Đoạn code xử lý quá trình gửi mail

Trong lớp EmailServiceImpl, phương thức sendOtp() có nhiệm vụ tạo nội dung email chứa mã OTP và gửi đến địa chỉ email của người dùng. Bên cạnh đó, phương thức sendUserTokenLink() được sử dụng để gửi liên kết đăng nhập.

Các thông tin cấu hình gửi mail được khai báo trong file application.properties, bao gồm các tham số cần thiết gồm tên máy chủ cung cấp dịch vụ thư điện tử, số cổng giao tiếp và thông tin xác thực của tài khoản thực hiện gửi thư. Trong đề án, máy chủ SMTP được cấu hình là smtp.gmail.com với cổng 587, đồng thời bật cơ chế xác thực SMTP. [25]

Cách tích hợp này giúp tác giả tận dụng dịch vụ gửi mail có sẵn của bên thứ ba, giảm độ phức tạp trong triển khai và vẫn có khả năng thay thế bằng các dịch vụ gửi email khác trong tương lai.

Tích hợp xác thực người dùng bằng ứng dụng bên thứ ba: Trong đề án, để sử dụng chức năng đăng nhập bằng ứng dụng thứ ba, tác giả đã cài đặt thư viện spring-boot-starter-oauth2-client. Thư viện này cung cấp các thành phần cần thiết để Spring Security hỗ trợ đăng nhập thông qua nhà cung cấp bên thứ ba như Google.

Các tham số được khai báo trong file application.properties cho từng nhà cung cấp đăng nhập bên thứ ba như Google. Trong đó, client-id và client-secret là thông tin định danh ứng dụng do từng nhà cung cấp cấp cho project. Ví dụ với Google, thuộc tính scope=profile,email được sử dụng để yêu cầu quyền truy cập thông tin hồ sơ cơ bản như họ và tên và địa chỉ email.

```
# Google OAuth2 Client Configuration
spring.security.oauth2.client.registration.google.client-id=
spring.security.oauth2.client.registration.google.client-secret=
spring.security.oauth2.client.registration.google.scope=profile,email
spring.security.oauth2.client.registration.google.redirect-uri={baseUrl}/login/oauth2/code/google
spring.security.oauth2.client.registration.google.authorization-grant-type=authorization_code
```

Hình 2.4 Cấu hình các thông số liên quan đến đăng nhập bằng tài khoản Google

Tại trang đăng nhập, khi người dùng lựa chọn đăng nhập với tài khoản Google, trình duyệt gửi yêu cầu đến đường dẫn /oauth2/authorization/google. Sau khi tiếp nhận yêu cầu, Spring Security khởi tạo quy trình xác thực OAuth2 và chuyển hướng người dùng đến trang xác thực của Google để đăng nhập và cấp quyền truy cập thông tin cần thiết.

Trong quá trình đăng nhập bằng Google, Spring security thực hiện việc chuyển người dùng đến trang xác thực của ứng dụng thứ ba, tiếp nhận kết quả tại địa chỉ nhận phản hồi, sử dụng mã ủy quyền để lấy thông tin tài khoản và thiết lập phiên đăng nhập cho ứng dụng, do đó tác giả không cần tự viết toàn bộ quá trình thủ công. Khi việc xác thực thông tin đăng nhập kết thúc thành công, Spring security cung cấp thông tin người dùng dưới dạng đối tượng OAuth2User. Từ đối tượng chứa kết quả xác thực, ứng dụng đọc các dữ liệu cần thiết như địa chỉ email, tên hiển thị và mã định danh do Google cấp. Dựa trên những dữ liệu này, hệ thống tiến hành tra cứu tài khoản đã được lưu trước đó. Trường hợp chưa tìm thấy bản ghi phù hợp, một tài khoản mới sẽ được khởi tạo bằng thông tin nhận từ Google. Nếu tài khoản đã tồn tại, hệ thống sử dụng bản ghi tương ứng để tiếp tục xử lý. Cuối cùng, trạng thái đăng nhập được thiết lập để người dùng truy cập các chức năng của ứng dụng.

Tóm lại, cách tích hợp này giúp tác giả tận dụng cơ chế xác thực sẵn có của Google, làm cho quá trình triển khai trở nên bớt phức tạp hơn và giúp mở rộng để tiếp nhận phương thức đăng nhập từ những ứng dụng bên thứ ba khác.

Tích hợp xác thực bằng passkey: Trong đề án, tác giả định hướng tích hợp cơ chế xác thực Passkey bắt đầu từ việc cài đặt thư viện spring-security-webauthn. Thư viện này hỗ trợ triển khai đăng nhập bằng passkey bằng cách sử dụng API để thực hiện quá trình đăng ký, xác thực người dùng.

Với cách tích hợp này, server đóng vai trò là Relying Party (RP), chịu trách nhiệm phối hợp quá trình xác thực với trình duyệt và thiết bị của người dùng. Phía client sử dụng API để tương tác với thiết bị xác thực như vân tay, khuôn mặt, mã PIN hoặc khóa bảo mật phần cứng. Khóa bí mật không được gửi về server mà được lưu an toàn trên thiết bị người dùng; server chỉ lưu các thông tin cần thiết như Public Key và Credential ID, sau đó sử dụng endpoint WebAuthn do Spring Security hỗ trợ để kiểm tra dữ liệu xác thực trong các lần đăng nhập sau.

Một ưu điểm của việc sử dụng spring-security-webauthn là Back-end không viết toàn bộ logic xử lý như tạo challenge, kiểm tra phản hồi xác thực. Sau khi quá

trình xác thực hoàn tất, Spring Security biểu diễn kết quả thu được thông qua đối tượng Authentication. Từ đối tượng này, tác giả sử dụng phương thức getName() để lấy ra địa chỉ email. [19]

```
if (authentication instanceof WebAuthnAuthentication) {
    String email = authentication.getName();

    User user = userRepository.findByEmail(email)
            .orElseThrow(() -> new RuntimeException(ErrorMessage.USER_NOT_FOUND.getMessage()));
```

Hình 2.5 Lấy địa chỉ email từ đối tượng WebAuthnAuthentication

2.3 Kết luận chương II

Kết luận qua chương II tác giả đã phân tích các yêu cầu của giải pháp xác thực không mật khẩu, bao gồm yêu cầu về phần cứng, phần mềm, khả năng tích hợp với ứng dụng web, khả năng mở rộng và vừa đáp ứng khả năng bảo mật. Những cơ sở này là rất quan trọng để tác giả có thể định hướng thiết kế phù hợp cho hệ thống. Bên cạnh đó, chương đã đề xuất định hướng kiến trúc theo mô hình client–server, kết hợp với các công nghệ web phổ biến và định hướng cách tích hợp vào phạm vi đề án. Tổng kết, tác giả đã cho biết chương II trình bày cơ sở nền tảng phân tích và định hướng kỹ thuật cho việc triển khai giải pháp không dùng mật khẩu xác thực. Đây là cơ sở để Chương III tiếp tục trình bày quá trình xây dựng, tích hợp và đánh giá giải pháp trong môi trường ứng dụng web.

CHƯƠNG III. THIẾT KẾ, XÂY DỰNG MÔ HÌNH THỬ NGHIỆM VÀ ĐÁNH GIÁ GIẢI PHÁP

3.1 Thiết kế kiến trúc mô hình thử nghiệm

3.1.1 Thiết kế nghiệp vụ các chức năng

a) Các chức năng cơ bản trong ứng dụng:

Bảng 3.1 Bảng mô tả use case các chức năng cơ bản trong ứng dụng

Chức năng	**Tác nhân, điều kiện**	**Luồng sự kiện chính và ngoại lệ**
Đăng ký tài khoản	Người dùng, địa chỉ email được cung cấp chưa dùng để đăng ký với bất kỳ tài khoản nào.	Người dùng điền các thông tin cho việc đăng ký như: họ và tên, email, số điện thoại và mật khẩu; dữ liệu gửi lên được kiểm tra, nếu hợp lệ sẽ tạo tài khoản có quyền là guest, lưu thông tin và gửi liên kết xác minh. Ngoại lệ: dữ liệu sai, email trùng hoặc mật khẩu xác nhận không khớp.
Đặt lại mật khẩu	Tài khoản đã được đăng ký và hiện không có phiên đăng nhập đang hoạt động.	Người dùng cung cấp địa chỉ email để hệ thống gửi liên kết đăng nhập. Sau đó đặt mật khẩu mới, hệ thống cập nhật địa chỉ email mới. Ngoại lệ: Liên kết xác thực hết hạn, đã sử dụng, mật khẩu không đúng hoặc xác nhận mật khẩu không trùng với ô mật khẩu.
Cập nhật thông tin tài khoản	Tài khoản đã được xác minh, đã đăng nhập và có	Người dùng cập nhật địa chỉ email, họ tên và số

	quyền là user hoặc admin.	điện thoại trên giao diện, sau đó chọn nút lưu để cập nhật thông tin cá nhân. Ngoại lệ: Hệ thống trả về thông báo dữ liệu bị sai và không cập nhật dữ liệu.
Xem danh sách nhân viên	Phiên đăng nhập đang hoạt động và tài khoản có vai trò quản trị viên (admin)	Hiển thị các thông tin nhân viên như họ và tên, email, số điện thoại, trạng thái khóa và trạng thái xác minh của nhân viên. Ngoại lệ: Tài khoản có quyền là user hoặc guest thì quyền truy cập bị từ chối.

b) Các chức năng xác thực không mật khẩu:

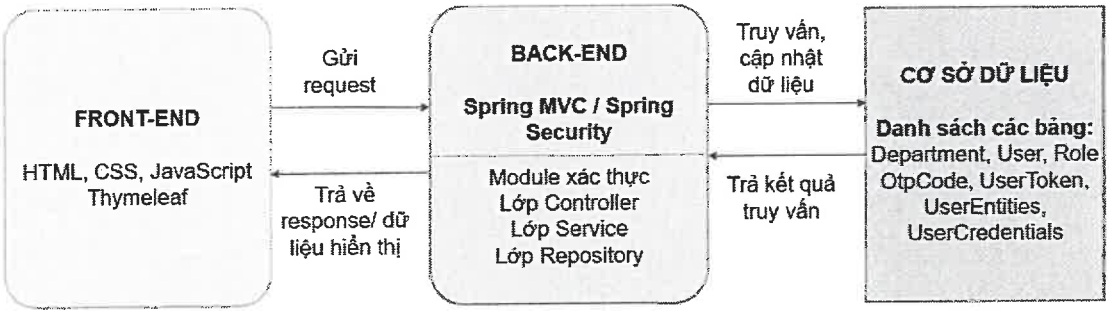
Bảng 3.2 Bảng mô tả use case các chức năng xác thực không mật khẩu

Chức năng	**Tác nhân, điều kiện**	**Luồng sự kiện chính và ngoại lệ.**
Đăng nhập bằng ứng dụng bên thứ ba.	Người dùng có tài khoản Google.	Chọn ứng dụng bên thứ ba và đăng nhập. Sau đó hệ thống nhận kết quả, đối chiếu hoặc tạo tài khoản rồi đăng nhập. Ngoại lệ: người dùng từ chối cấp quyền hoặc xác thực bằng ứng dụng bên thứ ba không thành công.

Đăng nhập bằng mã xác thực OTP	Người dùng đã có tài khoản.	Người dùng điền email để hệ thống gửi mã OTP, sau đó xác thực bằng địa chỉ email và mã OTP để xác thực. Ngoại lệ: Mã OTP bị sai, hết hạn hoặc đã sử dụng; nhập sai quá 5 lần thì khóa tài khoản 30 phút.
Đăng nhập bằng liên kết xác thực	Người dùng đã đăng ký tài khoản	Người dùng điền email để hệ thống gửi liên kết xác thực. Sau đó người dùng mở liên kết để đăng nhập. Sau đó hệ thống kiểm tra mã xác thực hợp lệ hay không. Ngoại lệ: Liên kết hết hạn sau một phút hoặc đã được sử dụng.
Quét mã QR để lấy xác thực mã TOTP và đăng nhập bằng mã xác thực TOTP	Tài khoản của người dùng đã được gán quyền là user hoặc admin. Người dùng đã xác thực vào đã vào bên trong ứng dụng.	Người dùng quét mã QR trên ứng dụng xác thực để tạo mã TOTP. Sau đó nhập email và mã TOTP để đăng nhập. Ngoại lệ: Tài khoản có quyền là guest có quyền quét mã QR để lấy mã TOTP. Khi đăng nhập,

		nhập sai mã quá 5 lần thì bị khóa 30 phút.
Quản lý passkey và đăng nhập bằng passkey.	Tài khoản của người dùng đã được gán quyền là user hoặc admin. Người dùng đã xác thực vào đã vào bên trong ứng dụng.	Người dùng thêm và xóa Passkey bằng sinh trắc học hoặc nhập mã PIN. Khi đăng nhập, trình duyệt tạo dữ liệu xác thực, người dùng xác thực và gửi lên cho server kiểm tra. Ngoại lệ: Người dùng hủy quá trình xác thực hoặc xác thực không hợp lệ.

3.1.2 Tổng quát các thành phần chính (Back-end, front-end)



Hình 3.1 Tổng quát luồng xử lý giữa Back-end, front-end

Hình trên mô tả mối quan hệ tổng quát giữa Front-end, back-end và cơ sở dữ liệu. Trong đó, Front-end là thành phần giao tiếp trực tiếp với người dùng, cung cấp giao diện tương tác và tiếp nhận dữ liệu đầu vào từ người dùng. Cụ thể đầu tiên Front-end gửi request đến Back-end kèm biểu mẫu. Trong quá trình back-end tiếp nhận yêu cầu và xử lý, back-end có thể đọc hoặc ghi thông tin tại cơ sở dữ liệu. Sau khi Back-end hoàn tất việc xử lý yêu cầu, phản hồi được gửi đến Front-end. Dựa trên nội dung

nhận được, Front-end sẽ thông báo kết quả, thay đổi trạng thái giao diện hoặc chuyển người dùng đến trang tương ứng.

Giao diện phía người dùng được phát triển với HTML, CSS và JavaScript. và Thymeleaf, có nhiệm vụ hiển thị giao diện, tiếp nhận thao tác của người dùng và gửi dữ liệu đăng ký, đăng nhập đến Back-end.

Back-end chịu trách nhiệm xử lý logic nghiệp vụ, bao gồm quá trình xác thực người dùng. Trong Back-end bao gồm các bộ phận xử lý sau:

- Controller: Tiếp nhận request từ front-end và điều phối yêu cầu đến các service tương ứng.
- Service: Đảm nhiệm việc xử lý các chức năng nghiệp vụ như đăng ký tài khoản, đăng nhập bằng các phương pháp không mật khẩu, quên mật khẩu, chỉnh sửa thông tin cá nhân, quản lý nhân viên và phòng ban, phân quyền.
- Repository: Spring Data JPA đảm nhiệm việc đọc, thêm, cập nhật và xóa dữ liệu của ứng dụng, bao gồm tìm kiếm, tạo mới, cập nhật và loại bỏ các bản ghi.
- Mô-đun xác thực: Xử lý các phương thức xác thực bằng bằng mã OTP, liên kết đăng nhập, mã xác thực TOTP, đăng nhập bằng tài khoản ứng dụng bên thứ ba (ví dụ như Google) và đăng nhập bằng Passkey.
- Cơ sở dữ liệu: Thao tác truy vấn với dữ liệu và gửi lại dữ liệu truy vấn cho Back-end, cụ thể là trả kết quả cho lớp Repository.

3.1.3 Mối quan hệ giữa ứng dụng web, mô-đun xác thực và các thư viện hỗ trợ xác thực:

Trong ứng dụng web bao gồm phần Front-end đảm nhiệm giao diện tương tác và phần Back-end xử lý nghiệp vụ. ở phía front-end chịu trách nhiệm hiển thị giao diện, và gửi các yêu cầu đăng ký, đăng nhập đến Back-end. Back-end tiếp nhận và chuyển các yêu cầu liên quan đến đăng nhập tới mô-đun xác thực để kiểm tra thông tin

Bên trong Back-end, mô-đun xác thực chịu trách nhiệm xử lý các phương thức đăng nhập không mật khẩu. Đối với OTP, TOTP và liên kết đăng nhập, tác giả xây dựng filter và provider riêng cho từng phương thức. Filter tiếp nhận thông tin từ yêu cầu đăng nhập và chuyển đến provider tương ứng để xác thực. Sau đó, provider gọi service để thực hiện các nghiệp vụ như tạo và kiểm tra mã OTP, tạo và xác minh mã TOTP, hoặc tạo và kiểm tra mã xác thực của liên kết đăng nhập.

Đối với phương thức đăng nhập bằng ứng dụng bên thứ ba, thư viện spring-boot-starter-oauth2-client được tích hợp vào back-end nhằm hỗ trợ quy trình đăng nhập qua ứng dụng bên ngoài như Google. Thông qua thư viện này, quá trình đăng nhập bắt đầu từ việc chuyển hướng người dùng đến Google để đăng nhập và cấp quyền, tiếp nhận mã ủy quyền được trả về, trao đổi mã để lấy thông tin định danh và ghi nhận kết quả xác thực. Sau đó, back-end sử dụng thông tin nhận được để đối chiếu thông tin tài khoản với thông tin đang được lưu trữ; nếu tài khoản chưa tồn tại, hệ thống tạo mới trước khi hoàn tất quá trình đăng nhập.

Đối với phương thức đăng nhập bằng passkey, thư viện spring-security-webauthn giúp hỗ trợ xử lý quá trình xác thực. Từ quá trình thực hiện việc tạo dữ liệu yêu cầu xác thực, quản lý challenge cho đến tiếp nhận phản hồi từ trình duyệt, kiểm tra thông tin xác thực và xác định tài khoản tương ứng. [19]

Để triển khai chức năng gửi email, tác giả tích hợp thư viện spring-boot-starter-mail và cấu hình máy chủ SMTP của Gmail. Thư viện này hỗ trợ Back-end tạo và gửi thư điện tử đến địa chỉ của người dùng. Cụ thể là gửi mã OTP hoặc liên kết đăng nhập đến địa chỉ email.

3.1.4 Đảm bảo kiến trúc đáp ứng yêu cầu về bảo mật và khả năng mở rộng

a) Về bảo mật:

Trong phạm vi đề án, tác giả sử dụng Spring Security để kiểm soát quá trình xác thực và truy cập. Cụ thể các yêu cầu gửi lên back-end được điều hướng đến Spring

Security Filter Chain trước khi chuyển đến controller. Đối với các phương pháp đăng nhập, yêu cầu xác thực được chuyển từ filter đến provider và đến service tương ứng để kiểm tra xác thực. Hệ thống chỉ thiết lập phiên đăng nhập khi thông tin xác thực hợp lệ và từ chối các yêu cầu không đáp ứng điều kiện. Cách tổ chức này giúp kiểm soát và thống nhất luồng xác thực, hạn chế truy cập trái phép. [23]

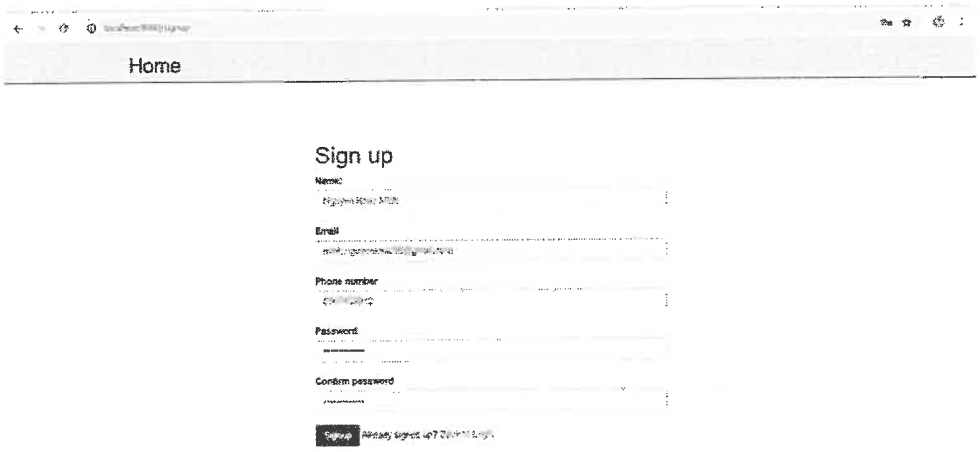
b) Về khả năng mở rộng:

Ứng dụng được tổ chức theo mô hình phân tầng gồm controller, service và repository. Riêng các phương thức không sử dụng mật khẩu khi xác thực được xây dựng thành những thành phần xử lý riêng gọi là mô-đun xác thực. Cách tổ chức này cũng giúp việc bảo trì và mở rộng thêm các phương thức đăng nhập mới thuận tiện hơn.

3.2 Xây dựng và thực nghiệm giải pháp

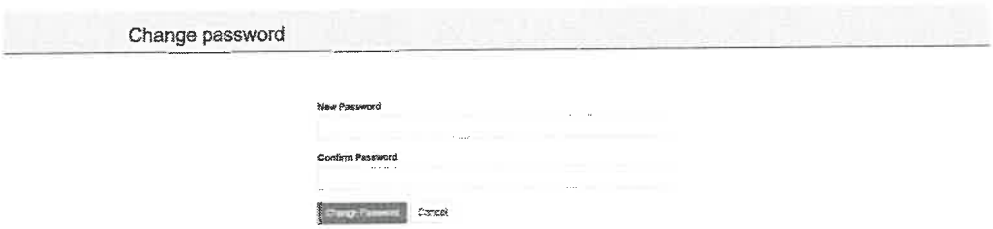
3.2.1 Xây dựng giao diện người dùng

a) Giao diện đăng ký tạo tài khoản mới:



Hình 3.2 Giao diện người dùng đăng ký tài khoản

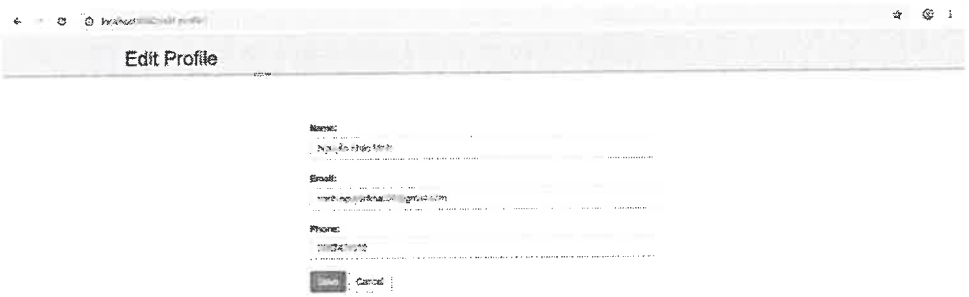
b) Giao diện chức năng đặt lại mật khẩu:



Hình 3.3 Giao diện đặt lại mật khẩu

Nếu giá trị ở ô xác nhận mật khẩu không trùng với giá trị ở mật khẩu mới, server trả về lỗi mật khẩu không trùng khớp.

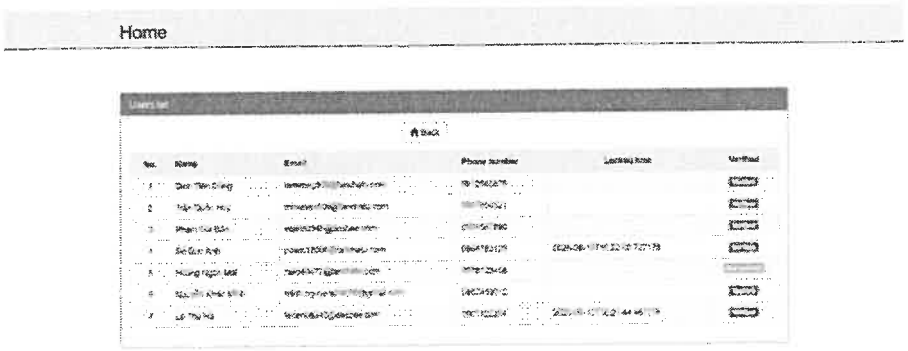
c) Giao diện chức năng chỉnh sửa thông tin tài khoản:



Hình 3.4 Giao diện chức năng chỉnh sửa thông tin tài khoản

Chức năng này mô tả người dùng thao tác chỉnh sửa các thông tin như họ tên tên, email và số điện thoại.

d) Giao diện chức năng xem danh sách nhân viên:

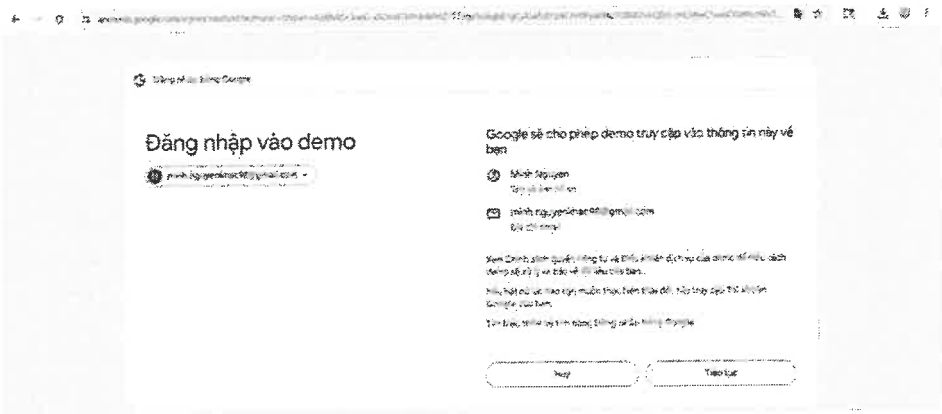


Hình 3.5 Giao diện chức năng xem danh sách nhân viên

Các dữ liệu về tài khoản nhân viên được hiển thị để quản trị viên kiểm tra bao gồm họ tên, email, số điện thoại, thời gian tạm khóa (nếu tài khoản bị khóa) và trạng thái xác thực (Chưa xác thực hoặc đã xác thực)

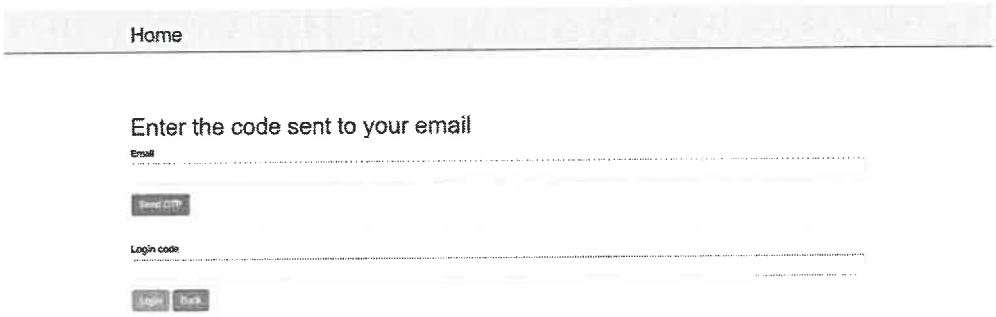
e) Giao diện chức năng đăng nhập bằng tài khoản ứng dụng bên thứ ba:

Xác thực thông qua Google: Sau khi người dùng kết thúc đăng nhập tại Google và cấp quyền cho ứng dụng nhận được một số thông tin trong phạm vi đã cho phép, chẳng hạn như tên hiển thị và email.



Hình 3.6 Giao diện chức năng đăng nhập thông qua ứng dụng Google

f) Giao diện chức năng đăng nhập bằng mã xác thực OTP:



Hình 3.7 Giao diện chức năng đăng nhập bằng mã xác thực OTP

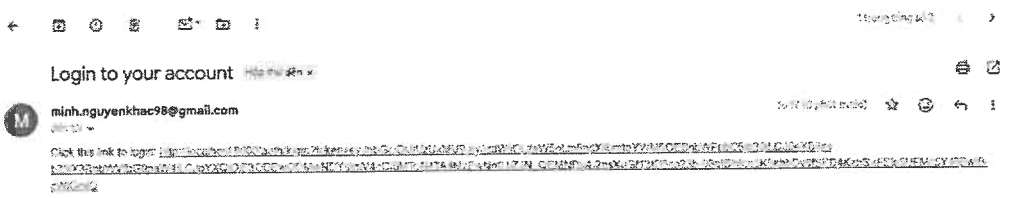
Người dùng điền địa chỉ email để hệ thống gửi xác mã xác thực về địa chỉ email đó.

g) Giao diện chức năng đăng nhập bằng liên kết:



Hình 3.8 Giao diện chức năng đăng nhập bằng liên kết

Người dùng nhập địa chỉ email. Sau đó, server sẽ gửi liên kết xác thực về địa chỉ email.



Hình 3.9 Liên kết xác thực được gửi đến địa chỉ email

Liên kết được gửi tới địa chỉ email và có hiệu lực trong vòng 1 phút. Nếu quá 1 phút, server trả về thông báo lỗi liên kết đã hết hiệu lực.

Trường hợp người dùng sử dụng lại liên kết để đăng nhập, server trả về thông báo lỗi liên kết không hợp lệ.

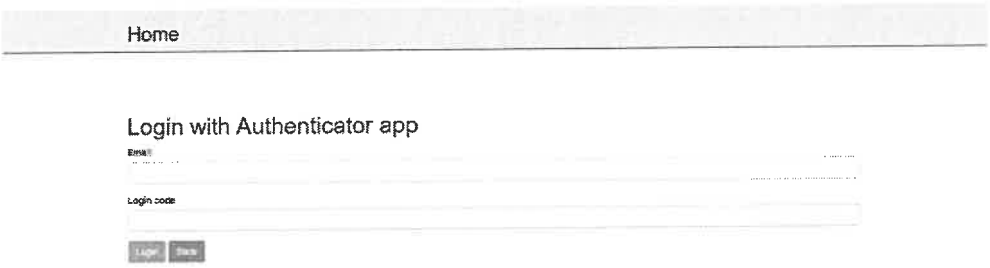
h) Giao diện chức năng quét mã QR để lấy mã xác thực TOTP:



Hình 3.10 Giao diện chức năng quét mã QR

Người dùng sử dụng ứng dụng Google authenticator trên điện thoại thông mình để quét mã QR và lấy mã đăng nhập.

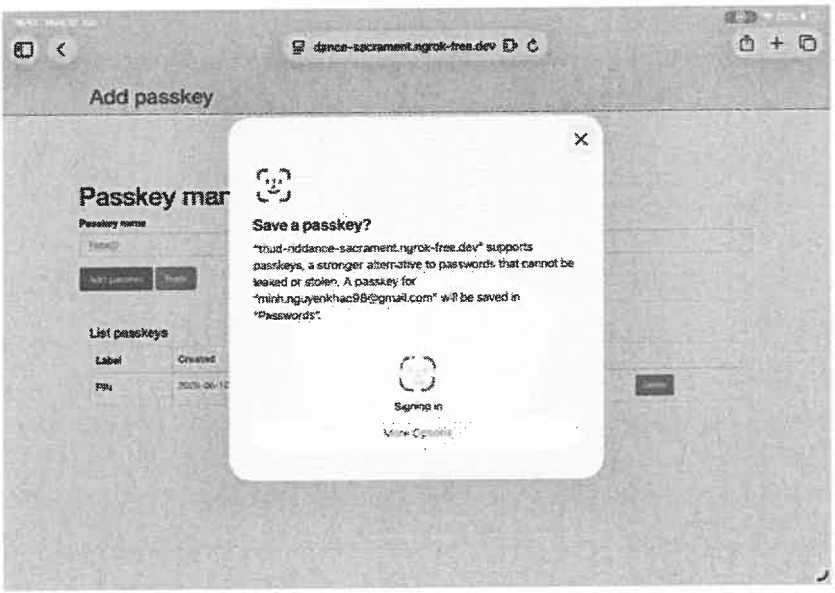
i) Giao diện chức năng đăng nhập bằng mã xác thực TOTP:



Hình 3.11 Giao diện chức năng đăng bằng mã xác thực TOTP

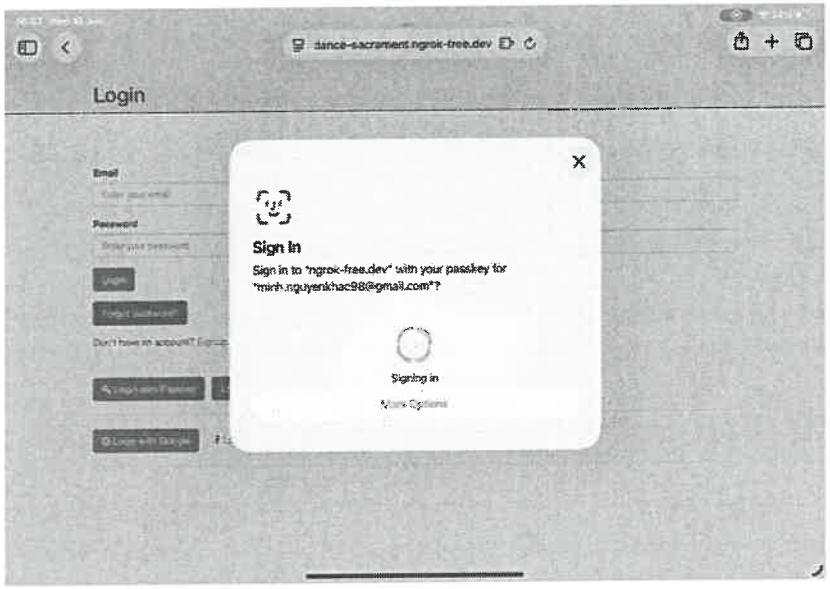
Người dùng mở Google Authenticator để lấy mã TOTP hiện hành, sau đó nhập mã này cùng địa chỉ email vào biểu mẫu đăng nhập.

k) Giao diện chức năng thêm và xóa passkey:



Hình 3.12 Giao diện chức năng thêm passkey trên iPad

l) Giao diện chức năng đăng nhập bằng passkey:



Hình 3.13 Giao diện chức năng đăng nhập bằng passkey

3.2.2 Xây dựng cơ sở dữ liệu

Bảng 3.3 Danh sách bảng và vai trò của các bảng trong cơ sở dữ liệu

Bảng	**Vai trò**
User	Bao gồm dữ liệu của tài khoản người dùng (Ví dụ: Họ tên, email, số điện thoại và mật khẩu)
User token	Lưu mã xác thực dùng cho các nghiệp vụ như xác nhận tài khoản, đặt lại mật khẩu, xác thực bằng liên kết đăng nhập
Otp code	Lưu mã xác thực OTP, thời gian hết hạn và trạng thái sử dụng của mã xác thực.
Role	Lưu danh sách role (Bao gồm: Admin, user, guest) để phân quyền truy cập.
User roles	Bảng user_roles đóng vai trò liên kết dữ liệu giữa bảng user và bảng role, dùng để xác định quyền hạn của từng tài khoản người dùng.

User credentials	Lưu các thông tin cần thiết của Passkey, chẳng hạn credential ID, khóa công khai và tài khoản sở hữu credential, phục vụ việc kiểm tra dữ liệu xác thực trong quá trình đăng nhập.
User entities	Lưu thông tin định danh liên quan đến người dùng phục vụ quá trình đăng ký và xác thực bằng Passkey.

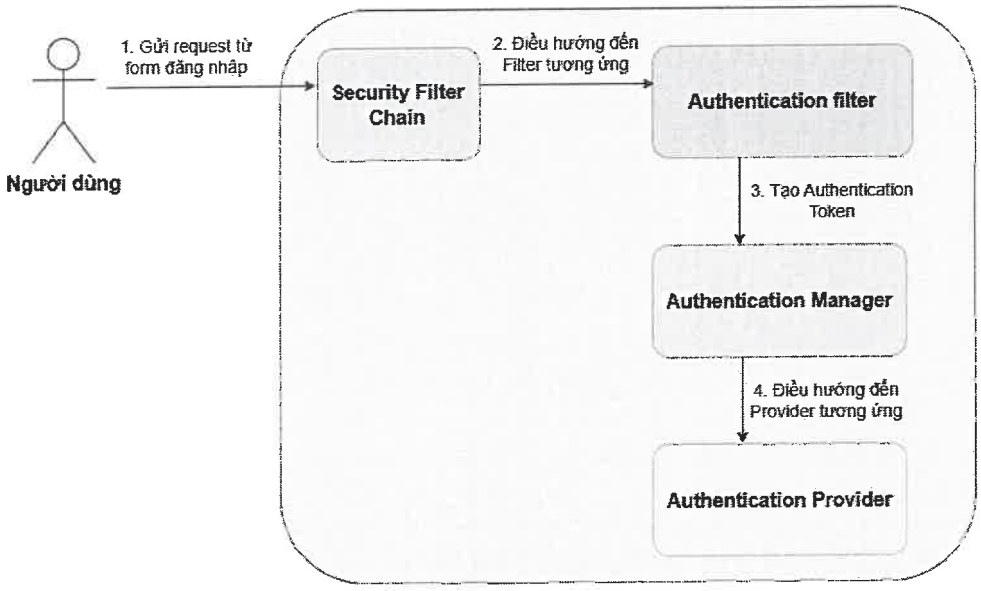
Bảng 3.4 Quan hệ giữa các thực thể

Liên hệ giữa các thực thể	**Kiểu quan hệ**	**Mô tả**
User – User token	1 – n	Một tài khoản có nhiều mã xác thực dùng cho các nghiệp vụ như: xác nhận tài khoản, đặt lại mật khẩu hoặc xác thực bằng liên kết đăng nhập.
User – Otp code	1 – n	Một tài khoản có nhiều mã OTP dùng để đăng nhập.
User - Role	n – n	Mỗi tài khoản được cấp nhiều quyền và một quyền cũng có thể được áp dụng cho nhiều tài khoản khác nhau. Quan hệ n-n được biểu diễn thông qua bảng trung gian user_roles.
User entities – User credentials	1 - n	Một định danh người dùng có nhiều credential Passkey, tương ứng với việc người dùng có thể đăng ký nhiều Passkey.

User – User entities	1 - 1	Một tài khoản có một thông tin định danh Passkey, dùng để phục vụ quá trình đăng ký và xác thực bằng Passkey.

3.2.3 Xây dựng back-end cho các phương pháp không dùng mật khẩu xác thực

a) Xây dựng luồng đăng nhập cho xác thực mã OTP, mã xác thực TOTP và liên kết đăng nhập:



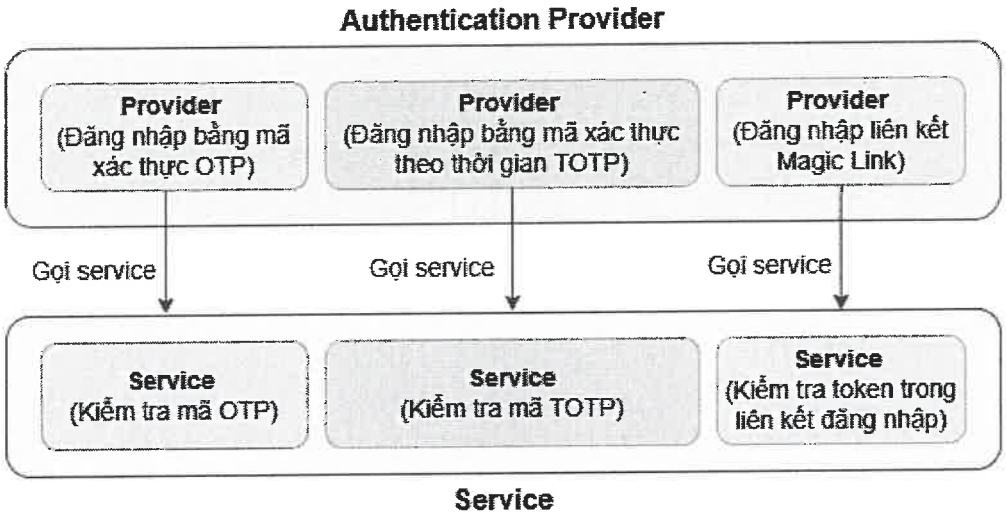
Hình 3.14 Luồng xử lý xác thực người dùng trong spring security

- Xây dựng luồng xác thực với Authentication filter và Authentication provider:

Trong cấu hình SecurityFilterChain.java, các filter của các phương pháp xác thực bằng mã OTP, mã TOTP, liên kết đăng nhập được tạo riêng và thêm vào Spring Security Filter Chain. Mỗi filter có nhiệm vụ xử lý yêu cầu đăng nhập của phương thức xác thực tương ứng.

Khi yêu cầu được gửi đến URL xác thực đã cấu hình, filter tương ứng sẽ trích xuất thông tin từ yêu cầu xác thực, tạo đối tượng Authentication phù hợp và chuyển đối tượng này đến Authentication manager. Sau đó, Authentication manager chuyển yêu cầu xác thực đến Authentication provider tương ứng để kiểm tra và trả về kết quả xác thực.

- Authentication provider gọi đến service tương ứng để kiểm tra thông tin xác thực:



Hình 3.15 Luồng xử lý giữa authentication provider và các service

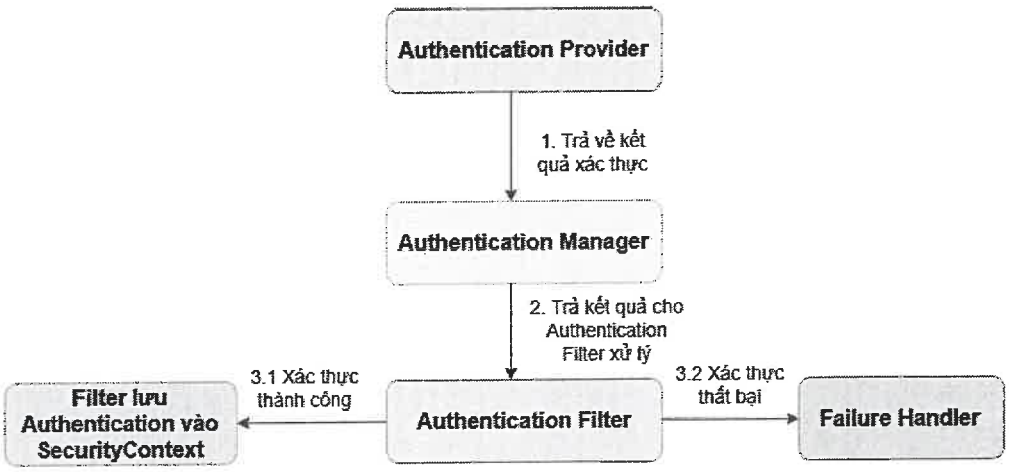
Các Service được gọi từ Authentication provider có nhiệm vụ kiểm tra thông tin xác thực tương ứng với từng phương thức đăng nhập:

Service kiểm tra mã OTP: Service sẽ kiểm tra mã OTP có phải là định dạng số hay không, mã OTP đã được sử dụng hay đã hết hiệu lực hay chưa, và mã OTP có khớp với mã OTP đang tồn tại ở trong cơ sở dữ liệu không.

Service kiểm tra mã TOTP: Theo tiêu chuẩn RFC 6238, mã TOTP được tạo dựa trên khóa bí mật của người dùng và thời gian hiện tại của máy chủ. Thời gian được chia thành các khoảng 30 giây; trong mỗi khoảng, hệ thống kết hợp giá trị thời gian tương ứng với khóa bí mật để tạo mã TOTP gồm 6 chữ số. Sau đó, mã do hệ thống tạo ra được so sánh với mã người dùng nhập để xác định tính hợp lệ. [26]

Service kiểm tra mã xác thực trong liên kết đăng nhập, đối chiếu mã xác thực với mã trên cơ sở dữ liệu, kiểm tra mã xác thực còn hạn hoặc đã được sử dụng hay chưa. Sau đó, mã xác thực được dùng để lấy thông tin như địa chỉ email.

- Xử lý kết quả xác thực trong authentication filter:



Hình 3.16 Luồng xử lý kết quả xác thực

Sau khi authentication provider kiểm tra thông tin xác thực, kết quả được trả về cho authentication manager và chuyển lại cho authentication filter. Nếu xác thực thành công, filter nhận đối tượng authentication đã xác thực, lưu vào security context, tạo thông tin phiên đăng nhập. Nếu quá trình xác thực diễn ra không thành công, request sẽ được chuyển đến authentication failure handler tương ứng để trả về thông báo xác thực thất bại.

b) Tích hợp chức năng đăng nhập bằng ứng dụng thứ ba:

Đối với chức năng đăng nhập bằng ứng dụng thứ ba, tác giả lựa chọn cài đặt thư viện spring-boot-starter-security-oauth2-client của Spring Security. Việc bổ sung thư viện này giúp ứng dụng thiết lập liên kết với ứng dụng thứ ba.

Endpoint /oauth2/authorization/google là endpoint mặc định do spring security oauth2 client cung cấp. Khi người dùng chọn đăng nhập bằng Google, Spring

security sẽ tự động chuyển hướng đến trang đăng nhập của ứng dụng thứ ba để xác thực tài khoản.

Thông tin oauth2 client như client-id, client-secret, phạm vi quyền truy cập được khai báo trong file application.properties. Các thông tin này giúp Spring Security xác định nhà cung cấp đăng nhập, gửi yêu cầu đăng nhập và xử lý phản hồi do ứng dụng bên thứ ba chuyển về. [20]

Spring Security hỗ trợ xử lý mã ủy quyền, trao đổi mã để nhận mã truy cập và lấy thông tin người dùng từ nhà cung cấp. Dựa trên thông tin nhận được, ứng dụng xác định người dùng và tạo phiên đăng nhập mà không cần tự viết toàn bộ quy trình xử lý.

```
if (principal instanceof OAuth2User oauthUser) {
    OAuth2AuthenticationToken token = (OAuth2AuthenticationToken) authentication;
    String provider = token.getAuthorizedClientRegistrationId();
    String providerId = oauthUser.getName();
    AuthProvider authProviderEnum = AuthProvider.from(provider);

    User user = userRepository.findByProviderAndProviderId(authProviderEnum, providerId);

    if (user == null) {
        throw new RuntimeException(ErrorMessage.USER_NOT_FOUND.getMessage());
    }
```

Hình 3.17 Đoạn mã mô tả quá trình lấy thông tin nhà cung cấp và thông tin tài khoản thông qua cách xác thực bằng ứng dụng thứ ba

Thông tin người dùng đã đăng nhập được lấy từ đối tượng authentication. Nếu principal là oauth2 user, thì thông tin user được lấy dựa theo tên nhà cung cấp đăng nhập (provider) và mã định danh tài khoản (providerId)

c) Tích hợp chức năng thêm passkey và đăng nhập bằng passkey:

Đối với phương thức đăng nhập bằng Passkey, tác giả cài đặt thư viện spring-security-webauthn để hỗ trợ xử lý các bước chính trong quá trình đăng ký và xác thực Passkey. Chẳng hạn như tạo challenge, xử lý credential và kiểm tra kết quả xác thực. Vì vậy, tác giả không cần tự xây dựng Filter và Provider riêng như các phương thức

xác thực mã OTP, TOTP hay liên kết đăng nhập. Trong SecurityConfig, tác giả khai báo rpId và allowedOrigins để xác định url website được phép sử dụng Passkey:

.webAuthn(webAuthn -> webAuthn

.rpId(APP_URL)

.allowedOrigins(APP_FULL_URL)

rpId là định danh của website sử dụng Passkey, allowedOrigins là địa chỉ được phép thực hiện các chức năng đăng ký và đăng nhập bằng passkey. Website cần sử dụng HTTPS để bảo vệ dữ liệu trao đổi trong quá trình đăng ký và xác thực bằng Passkey. AllowedOrigins để khai báo các nguồn được phép gửi phản hồi đến server. Server sẽ đối chiếu giá trị origin trong phản hồi với danh sách đã cấu hình và từ chối yêu cầu nếu không khớp, qua đó hạn chế việc một website giả mạo thực hiện quy trình đăng ký hoặc xác thực Passkey cho ứng dụng. [27]

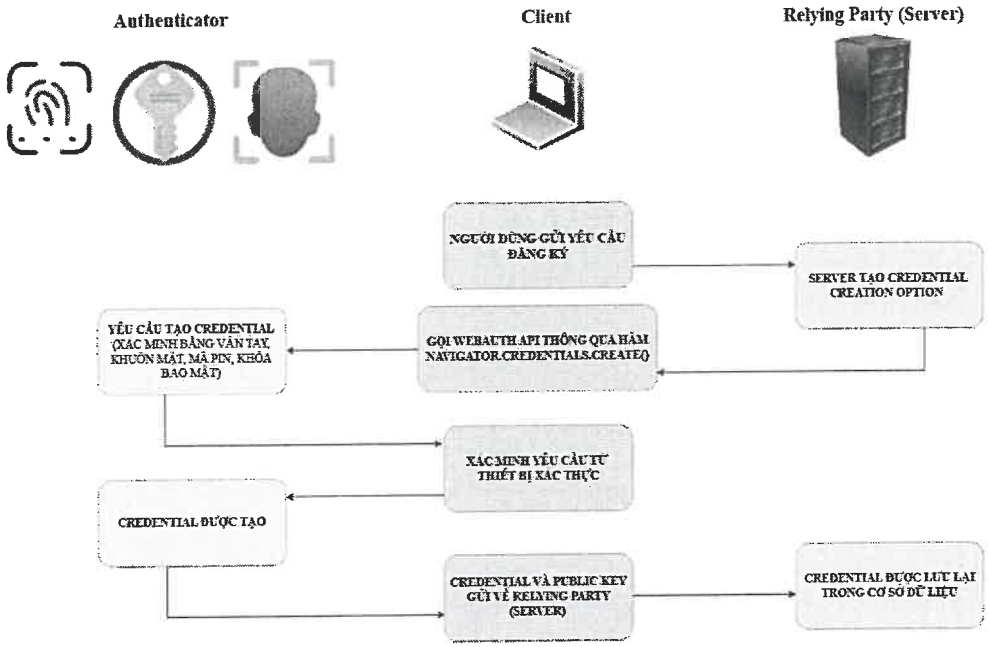
Ở back-end, tác giả không tự triển khai thủ công toàn bộ quy trình xác thực bằng Passkey. Sau khi xác thực thành công thì kết quả xác thực được biểu diễn bằng đối tượng authentication. Đoạn mã bên dưới cho biết lấy được địa chỉ email tài khoản từ đối tượng đó:

```
if (authentication instanceof WebAuthnAuthentication) {
    String email = authentication.getName();

    User user = userRepository.findByEmail(email)
            .orElseThrow(() -> new RuntimeException(ErrorMessage.USER_NOT_FOUND.getMessage()));
```

Hình 3.18 Đoạn mã lấy thông tin người dùng sau khi xác thực bằng Passkey

- Quá trình người dùng đăng ký thông tin xác thực bằng passkey:

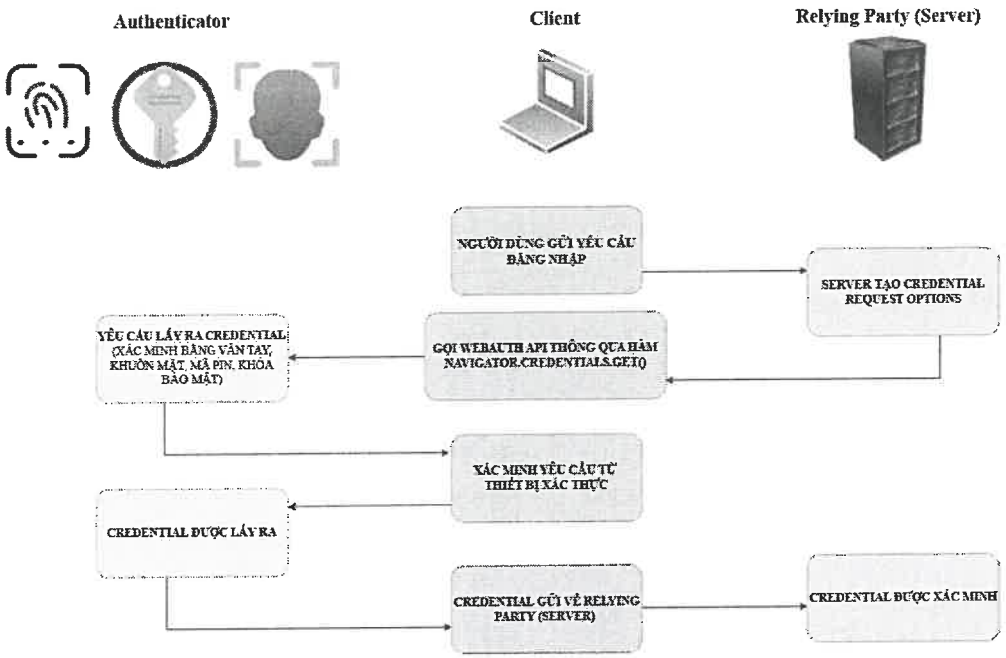


Hình 3.19 Quá trình đăng ký thông tin xác thực bằng passkey

Khi người dùng đăng ký Passkey, phía client sẽ gửi request lên server. Server lúc này tạo credential creation options bao gồm các thông tin như challenge, relying party. Trong đó quan trọng nhất là challenge — một chuỗi ngẫu nhiên được tạo riêng cho phiên đăng ký hiện tại. Challenge giúp đảm bảo yêu cầu phiên đăng ký là mới và không thể tái sử dụng. Server gửi các tùy chọn này về client, sau đó phía client gọi API qua hàm: navigator.credentials.create() để yêu cầu bộ phận xác thực tạo credential mới cho người dùng.

Bộ phận xác thực hay còn gọi authenticator là trình quản lý Passkey và các phương thức bảo mật như mã PIN, nhận diện vân tay và gương mặt. Khi kết quả kiểm tra cho thấy thông tin đăng nhập hợp lệ, bộ phận xác thực sẽ tạo cặp khóa bao gồm: Khóa bí mật được lưu trên bộ phận xác thực, còn khóa công khai và credential ID được gửi về server, server lưu vào cơ sở dữ liệu để sử dụng cho các lần xác thực sau. [22]

- Quá trình xác thực người dùng bằng Passkey:



Hình 3.20 Quá trình đăng nhập bằng cách sử dụng passkey

Client gửi yêu cầu request đăng nhập đến server. Server tạo credential request options, trong đó bao gồm challenge mới cho phiên đăng nhập hiện tại và danh sách credential ID đã được đăng ký trước đó, sau đó gửi các thông tin này về client.

Client gọi API qua hàm: navigator.credentials.get() Hàm này yêu cầu bộ phận xác thực tìm credential đã được đăng ký trước đó. Người dùng xác nhận danh tính bằng vân tay, khuôn mặt, mã PIN hoặc khóa bảo mật. Nếu xác nhận thành công, bộ phận xác thực sử dụng khóa bí mật tương ứng với credential đã đăng ký để ký challenge do server cung cấp.

Kết quả xác thực được client gửi lại cho server. Server sử dụng khóa công khai đã lưu trong cơ sở dữ liệu để kiểm tra chữ ký và challenge. Nếu chữ ký và challenge hợp lệ, server xác nhận người dùng đăng nhập thành công. [22]

d) Tích hợp dịch vụ gửi mã OTP và liên kết đăng nhập qua email:

Để gửi email xác thực, trong back-end tác giả đã cài đặt thư viện spring-boot-starter-mail của java spring boot. Thông tin sau đây được thêm trong file application.properties, bao gồm máy chủ mail smtp, cổng kết nối, tài khoản gửi mail và mật khẩu ứng dụng. Đồng thời, tác giả xây dựng Email service sử dụng có nhiệm vụ gửi email đến địa chỉ của người dùng. Service này được dùng trong hai trường hợp chính: gửi mã OTP đăng nhập và gửi liên kết đăng nhập. Trong đó, phương thức sendOtp() tạo và gửi email chứa mã xác thực OTP, còn sendUserTokenLink() tạo và gửi email chứa liên kết đăng nhập.

e) Kiểm soát phiên đăng nhập người dùng dựa trên cách tiếp cận Zero trust:

- Kiểm soát quyền truy cập:

Trong hệ thống, tài khoản được gán một trong ba vai trò gồm Admin, user và guest. Quá trình xác thực sau khi thành công, quyền của tài khoản được kiểm tra trước khi hệ thống xử lý yêu cầu truy cập đến từng chức năng. Ví dụ: Tài khoản guest không được phép truy cập bất kỳ chức năng nào nếu xác thực tài khoản, hay tài khoản user và guest không có quyền xem danh sách nhân viên ngoại trừ quản trị viên. Nếu vai trò hiện tại không có quyền sử dụng chức năng được yêu cầu, hệ thống sẽ từ chối truy cập. Cách triển khai này giúp giới hạn phạm vi sử dụng của từng nhóm người dùng và hỗ trợ nguyên tắc chỉ cấp quyền truy cập cần thiết trong cách tiếp cận Zero Trust.

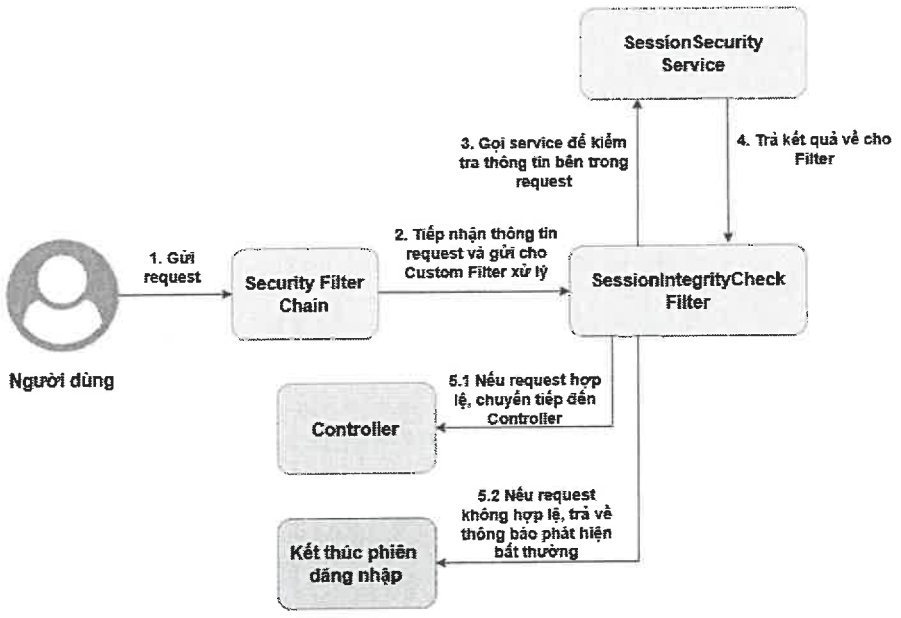
- Giới hạn thời gian hiệu lực của phiên đăng nhập:

Theo STIG Viewer, một công cụ tra cứu các STIG (Security Technical Implementation Guide) được phát triển bởi DISA (Cơ quan Hệ thống Thông tin Quốc phòng Mỹ), việc để phiên đăng nhập tồn tại vô thời hạn là một rủi ro bảo mật vì kẻ tấn công có thể lợi dụng phiên đăng nhập đã được xác thực để truy cập ứng dụng như người dùng hợp lệ. [28]

Mỗi phiên đăng nhập được gán một thời gian hiệu lực nhất định thông qua thuộc tính LOGIN_TIME lưu trong session. Khi một request đi qua security filter chain trong spring security sẽ được kiểm tra thời điểm đăng nhập ban đầu và so sánh với thời gian hiện tại. Khi phiên vượt quá thời gian cho phép, phiên đăng nhập sẽ kết thúc và người dùng phải xác thực lại. Chức năng này giúp đảm bảo mỗi phiên đăng nhập chỉ duy trì hiệu lực trong một khoảng thời gian cố định. Cơ chế này giúp quyền truy cập không được duy trì vô thời hạn, phù hợp với nguyên tắc xác minh lại của Zero Trust.

- Kiểm tra tính toàn vẹn của phiên đăng nhập:

Sau khi người dùng đăng nhập thành công, server tạo một bản ghi trạng thái tại thời điểm ngay sau khi đăng nhập để lưu lại thông tin, bao gồm địa chỉ IP, chuỗi User-Agent và vị trí địa lý được xác định từ IP như quốc gia, thành phố. Mục đích: Dùng để đối chiếu và phát hiện các thay đổi bất thường trong một phiên đăng nhập.



Hình 3.21 Luồng xử lý quá trình kiểm tra phiên đăng nhập

User-agent được sử dụng như một tín hiệu hỗ trợ trong quá trình đánh giá tính toàn vẹn của phiên đăng nhập. User-agent có thể cung cấp một số thông tin về môi

trường truy cập của người dùng, chẳng hạn như trình duyệt, phiên bản trình duyệt, hệ điều hành và loại thiết bị.

Một nghiên cứu về browser fingerprinting cho thấy fingerprint được hình thành từ nhiều thông tin của trình duyệt và thiết bị, trong đó user-agent là một thuộc tính có khả năng đóng góp lớn vào việc nhận diện môi trường truy cập. [29]

Tuy nhiên, do user-agent được gửi từ phía client nên thông tin này có thể thay đổi hoặc bị giả mạo, vì vậy không thể được xem là một định danh duy nhất hoặc một yếu tố xác thực độc lập. Vì vậy, tác giả không sử dụng User-Agent riêng lẻ để xác thực phiên đăng nhập, mà chỉ xem đây là một thuộc tính bổ trợ, được kết hợp với các thông tin khác như địa chỉ IP và vị trí địa lý nhằm phát hiện những thay đổi bất thường trong môi trường truy cập của phiên đăng nhập.

Khi request được gửi lên server, nó sẽ đi qua security filter chain trước khi đến Controller và sẽ đối chiếu thông tin hiện tại trong request với bản ghi trạng thái đã lưu tại thời điểm đăng nhập, bao gồm địa chỉ IP, chuỗi user-agent và vị trí địa lý tương ứng với IP hiện tại.

Nếu các thông tin này vẫn hợp lệ, request được tiếp tục chuyển đến Controller để xử lý. Ngược lại, nếu phát hiện sự thay đổi bất thường, server sẽ kết thúc phiên đăng nhập hiện tại và chuyển người dùng đến trang cảnh báo, yêu cầu xác thực lại trước khi tiếp tục sử dụng ứng dụng.

3.2.4 Các kịch bản để kiểm thử cơ chế xác thực

a) Kịch bản 1: Mô phỏng hành vi thử mã liên tiếp

Mục tiêu của kịch bản này là để xem tính an toàn trong trường hợp các yêu cầu xác thực thất bại liên tiếp đối với phương thức đăng nhập bằng mã OTP và TOTP.

Cách thực hiện cụ thể là nhập liên tiếp các mã OTP hoặc TOTP không chính xác. Trong vòng một phút, số yêu cầu xác thực được thực hiện vượt quá 5 lần. Sau đó, các yêu cầu tiếp tục được gửi để kiểm tra phản ứng của cơ chế giới hạn số lần thử.

b) Kịch bản 2: Mô phỏng việc tái sử dụng mã xác thực

Mục tiêu của kịch bản này đánh giá việc ngăn chặn sử dụng lại mã OTP hoặc liên kết đăng nhập sau khi giá trị xác thực đã được sử dụng, hết hạn hoặc bị thay thế.

Cách thực hiện là một mã OTP hoặc liên kết đăng nhập còn hiệu lực được sử dụng để thực hiện xác thực. Sau đó, giá trị vừa sử dụng được gửi lại trong một yêu cầu đăng nhập mới. Kịch bản cũng kiểm tra trường hợp sử dụng mã đã hết thời gian hiệu lực và trường hợp yêu cầu cấp mã OTP mới nhưng vẫn tiếp tục sử dụng mã cũ.

c) Kịch bản 3: Mô phỏng hành vi gửi mã OTP và liên kết đăng nhập liên tiếp

Mục tiêu của kịch bản được thực hiện nhằm đánh giá khả năng ngăn chặn việc yêu cầu gửi mã OTP hoặc liên kết đăng nhập nhiều lần trong thời gian ngắn.

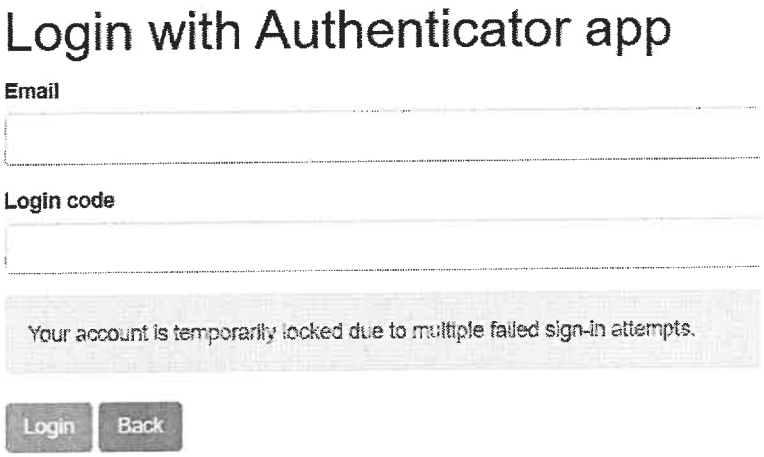
Sau khi hệ thống gửi email thành công, giao diện hiển thị bộ đếm ngược 60 giây. Trong khoảng thời gian này, chức năng gửi lại chưa được phép sử dụng. Khi tải lại trang, thời gian còn lại vẫn tiếp tục được duy trì thay vì bắt đầu lại từ đầu. Người kiểm thử cũng gửi yêu cầu trực tiếp đến back-end để bỏ qua giới hạn trên giao diện. Tuy nhiên, nếu chưa đủ 60 giây tính từ lần gửi gần nhất, yêu cầu vẫn bị từ chối và hệ thống trả về ngoại lệ. Chức năng gửi lại chỉ hoạt động khi thời gian chờ đã kết thúc.

3.2.5 Thực hiện mô phỏng các kịch bản:

a) Kịch bản 1: Mô phỏng hành vi thử mã liên tiếp

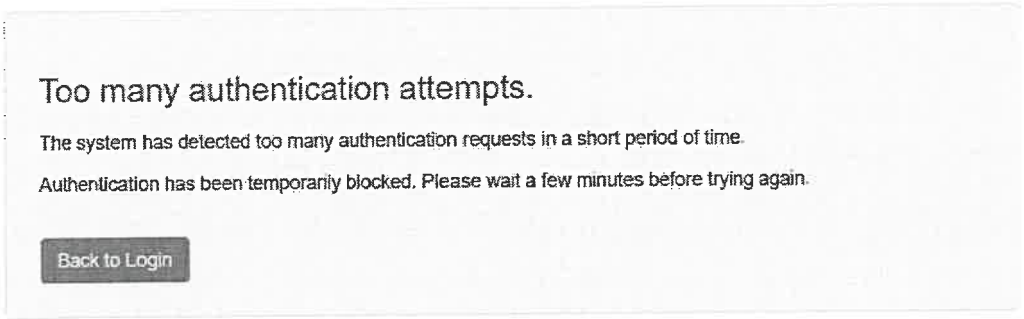
Người thực hiện liên tục cung cấp mã OTP hoặc TOTP không chính xác cho cùng một tài khoản. Sau lần xác thực sai thứ năm, tài khoản bị tạm khóa trong 30 phút và mọi yêu cầu đăng nhập tiếp theo đều bị từ chối, kể cả khi mã OTP được nhập

đúng. Khi thời gian khóa kết thúc, tài khoản được phép xác thực trở lại. Tuy nhiên, nếu lần đăng nhập đầu tiên sau khi mở khóa vẫn thất bại, hệ thống tiếp tục khóa tài khoản thêm 30 phút mà không đặt lại ngưỡng 5 lần thử.



Hình 3.22 Tài khoản bị khóa nếu người dùng xác thực sai quá 5 lần

Khi một tài khoản thực hiện quá 5 lần nhập mã xác thực trong thời gian chưa đến 1 phút, hệ thống sẽ chuyển người dùng đến trang cảnh báo về hoạt động đăng nhập bất thường. Ngưỡng này được lựa chọn vì tần suất gửi yêu cầu cao trong khoảng thời gian ngắn có thể là dấu hiệu của hành vi dò thử mã xác thực liên tục.



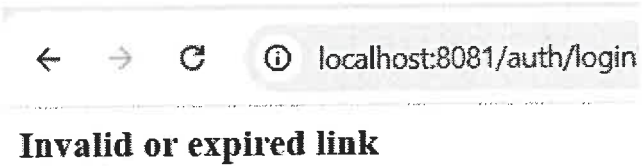
Hình 3.23 Hệ thống điều hướng sang trang cảnh báo đăng nhập bất thường

b) Kịch bản 2: Mô phỏng việc tái sử dụng mã xác thực

Đối với mã OTP và liên kết đăng nhập, người thực hiện sử dụng lại giá trị xác thực đã được dùng trước đó. Do mã hoặc liên kết chỉ có hiệu lực trong một lần xác thực, hệ thống từ chối đăng nhập và thông báo giá trị xác thực không còn hợp lệ.



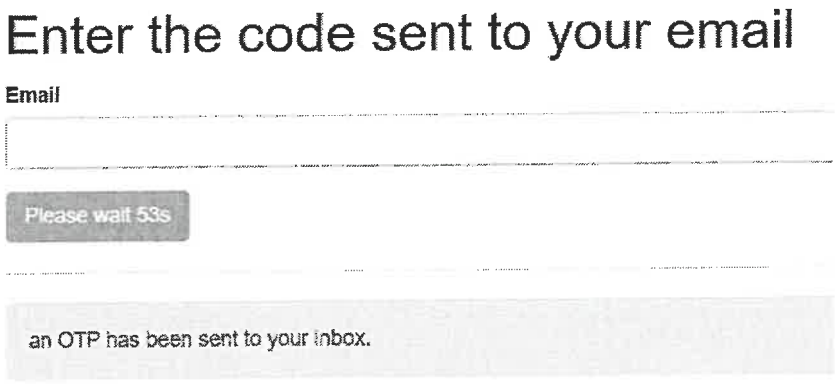
Hình 3.24 Mã xác thực OTP đã được sử dụng



Hình 3.25 Liên kết đăng nhập đã sử dụng hoặc đã hết hạn

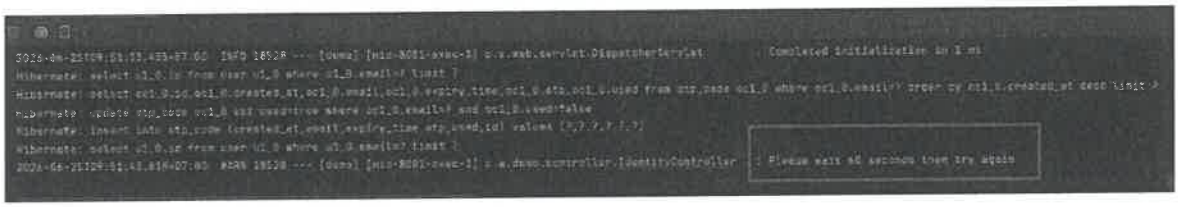
c) Kịch bản 3: Mô phỏng hành vi gửi mã OTP và liên kết đăng nhập liên tiếp

Trường hợp thứ nhất, người dùng nhập địa chỉ email để yêu cầu gửi mã OTP hoặc liên kết đăng nhập, lúc này giao diện hiển thị thời gian đếm ngược 60s để tránh việc gửi liên tiếp.



Hình 3.26 Giao diện hiển thị thời gian chờ gửi lại mã OTP

Trường hợp hai, tác giả gửi ngay yêu cầu gửi mã OTP và liên kết đăng nhập trong back-end để kiểm chứng việc giới hạn việc gửi liên tiếp được kiểm tra không chỉ ở front-end mà cả ở back-end



Hình 3.27 Back-end chặn việc gửi OTP

3.3 Đánh giá giải pháp:

3.3.1 Đánh giá tính an toàn

So với xác thực bằng mật khẩu truyền thống, các phương thức xác thực không mật khẩu giúp loại bỏ việc người dùng phải ghi nhớ và nhập mật khẩu, từ đó hạn chế các rủi ro như đặt mật khẩu yếu, sử dụng lại mật khẩu trên nhiều hệ thống hoặc mật khẩu bị đánh cắp. Tuy nhiên, các phương pháp xác thực như xác thực bằng mã OTP, mã TOTP và liên kết đăng nhập vẫn có nguy cơ kẻ xấu lợi dụng nếu mã xác thực không được quản lý an toàn. Vì vậy, các khuyến nghị của OWASP được lựa chọn làm cơ sở để xây dựng những cơ chế kiểm soát cho hệ thống.

Ở kịch bản đầu tiên, để phòng chống việc dò mã thì hệ thống giới hạn số lần đăng nhập thất bại đối với mã xác thực OTP và mã xác thực TOTP, khi người dùng nhập sai quá 5 lần trong một phút, các yêu cầu xác thực tiếp theo sẽ bị từ chối và hệ

thống sẽ khóa quyền truy cập tài khoản. Trường hợp nếu nhập sai tiếp hệ thống hiển thị cảnh báo quá số lần thử. Kết quả kiểm thử cho thấy sau khi vượt quá giới hạn, người dùng không thể tiếp tục gửi liên tục các mã khác để xác thực. Biện pháp này làm giảm khả năng thử nhiều giá trị cho đến khi tìm được mã đúng, phù hợp với khuyến nghị của OWASP về việc áp dụng giới hạn nghiêm ngặt đối với số lần thử OTP. [30]

Kịch bản kiểm thử thứ hai là nếu mã xác thực OTP và mã xác thực của liên kết đăng nhập được thiết lập thời gian tồn tại ngắn và chỉ được sử dụng một lần thì sau khi xác thực thành công, mã được coi là đã sử dụng và không còn khả năng sử dụng lại. Khi người dùng gửi lại OTP, hệ thống tạo mã mới và thay thế mã cũ. Trong quá trình kiểm thử, OTP hoặc liên kết đăng nhập đã sử dụng, đã hết hạn hoặc đã bị thay thế không thể tiếp tục dùng để đăng nhập. Cơ chế này phù hợp với khuyến nghị của OWASP về việc OTP phải có thời gian tồn tại ngắn, chỉ được sử dụng một lần, bị vô hiệu hóa sau khi xác thực thành công và được thay thế khi người dùng yêu cầu gửi lại. [30]

Với kịch bản thứ ba, mỗi địa chỉ email phải chờ 60 giây mới được yêu cầu gửi lại mã OTP hoặc liên kết đăng nhập. Điều kiện này được kiểm tra tại cả front-end và back-end. Cơ chế trên có tác dụng giảm các yêu cầu lặp lại trong thời gian ngắn nhưng chưa xử lý đầy đủ tình huống có hàng trăm hoặc hàng nghìn yêu cầu được tạo ra đồng thời. Theo khuyến nghị của OWASP, hệ thống cần kiểm soát tần suất cấp mã để tránh việc chức năng này bị lợi dụng nhằm tạo ra số lượng lớn email xác thực. Trong đề án, Hệ thống cần bổ sung việc thống kê số lượng yêu cầu trong một khoảng thời gian ngắn và từ chối xử lý khi vượt quá giới hạn quy định. Đây là nội dung cần tiếp tục hoàn thiện để tăng khả năng kiểm soát việc cấp mã với số lượng lớn. [31]

Để giảm thiểu rủi ro lộ thông tin trong cơ sở dữ liệu, hệ thống không lưu trực tiếp mã OTP và mã xác thực trong liên kết đăng nhập dưới dạng nguyên bản mà được chuyển thành giá trị băm trước khi ghi vào cơ sở dữ liệu. Nhờ đó, dữ liệu thu được khi cơ sở dữ liệu bị truy cập trái phép không thể được sử dụng trực tiếp để đăng nhập.

Biện pháp này phù hợp với khuyến nghị của OWASP về việc không ghi OTP vào nhật ký và không lưu OTP lâu dài dưới dạng nguyên bản. [30]

3.3.2 Tính thuận tiện

Hệ thống cung cấp nhiều phương thức đăng nhập không cần mật khẩu gồm đăng nhập bằng passkey, mã OTP, mã TOTP, liên kết đăng nhập và đăng nhập bằng ứng dụng thứ ba. Với đăng nhập bằng passkey, Việc đăng nhập được hoàn tất thông qua cơ chế mở khóa của thiết bị, chẳng hạn quét vân tay, nhận diện khuôn mặt hoặc nhập mã PIN.

Với đăng nhập bằng ứng dụng bên thứ ba, người dùng không cần điền mật khẩu cho ứng dụng web mà mật khẩu chính là từ ứng dụng bên thứ ba. Vì vậy phương pháp này thuận tiện với nhiều người dùng đang sử dụng tài khoản ứng dụng thứ ba như Google.

Đối với đăng nhập bằng mã xác thực OTP và liên kết đăng nhập, người dùng không phải ghi nhớ hay nhập bất kỳ mật khẩu mà chỉ cần kiểm tra hộp thư để lấy và nhập mã hoặc liên kết đăng nhập.

Trong khi với đăng nhập bằng mã xác thực TOTP yêu cầu cài đặt ứng dụng xác thực và quét mã QR trong lần thiết lập đầu tiên. Sau đó khi đăng nhập cần nhập địa chỉ email và mã TOTP hiển thị trên ứng dụng xác thực.

3.3.3 Tính ổn định

Kết quả thu được sau khi xây dựng và thử nghiệm cho thấy hệ thống vận hành ổn định trong phạm vi của đề án. Các phương pháp không sử dụng mật khẩu khi xác thực thực hiện đúng quy trình đã thiết kế, bao gồm tiếp nhận yêu cầu đăng nhập, xác thực thông tin từ yêu cầu đăng nhập, từ chối dữ liệu nếu không hợp lệ hoặc tạo phiên đăng nhập nếu xác thực thành công. Sau khi tích hợp các phương thức này, những chức năng khác của ứng dụng như đăng ký tài khoản, đặt lại mật khẩu, xem danh sách nhân viên và phân quyền hoạt động ổn định trong quá trình tác giả kiểm thử.

3.3.4 Khả năng tích hợp

Các phương thức xác thực không mật khẩu được tích hợp vào ứng dụng dựa trên cơ chế xác thực sẵn có của Spring Security. Đối với đăng nhập bằng mã xác thực OTP, mã TOTP và liên kết đăng nhập, hệ thống bổ sung các thành phần Filter và Provider để tiếp nhận yêu cầu và kiểm tra mã xác thực. Sau khi xác thực thành công, Spring Security thực hiện tạo phiên đăng nhập cho người dùng.

Đối với Passkey và đăng nhập bằng ứng dụng bên thứ ba, các thư viện của Spring Security hỗ trợ quy trình xác thực. Vì vậy, hệ thống chủ yếu bổ sung thông tin cấu hình và xử lý thông tin người dùng thay vì phải tự xây dựng lại toàn bộ luồng đăng nhập. Qua đó, việc tích hợp không làm tăng khối lượng mã nguồn và độ phức tạp, đồng thời thuận lợi cho việc sửa lỗi.

3.3.5 Đánh giá ưu điểm và nhược điểm khi triển khai các giải pháp không dùng mật khẩu xác thực:

Việc áp dụng các phương thức đăng nhập không dùng mật khẩu trong ứng dụng cho thấy mỗi phương thức đều có những lợi thế và vấn đề cần tiếp tục hoàn thiện, được trình bày dưới đây:

Bảng 3.5 Tổng kết ưu và hạn chế các phương pháp không dùng mật khẩu xác thực

	Ưu điểm	**Hạn chế**
Đăng nhập bằng Passkey	Quá trình xác thực sử dụng cơ chế mở khóa đã được thiết lập trên thiết bị, ví dụ: xác thực vân tay, khuôn mặt hoặc nhập mã PIN. Nhờ đó, phương pháp này giúp giảm rủi ro liên quan đến việc sử	Phương pháp này vẫn tồn tại rủi ro khi tác giả chưa xử lý trường hợp nhiều người cùng sử dụng chung một thiết bị và cùng có khả năng mở khóa thiết bị.

	dụng mật khẩu truyền thống.	Nếu nhiều passkey được lưu trên cùng thiết bị, người có quyền mở khóa thiết bị có thể sử dụng passkey để truy cập vào tài khoản không thuộc về mình.
Đăng nhập bằng ứng dụng bên thứ ba	Ứng dụng web không trực tiếp xử lý mật khẩu của người dùng, mà sử dụng cơ chế xác thực do các nhà cung cấp bên thứ ba, ví dụ: Google. Người dùng sau khi đăng nhập và đồng ý cấp quyền, ứng dụng sẽ nhận kết quả xác thực và trả về thông tin cơ bản như email hoặc tên người dùng để tạo phiên đăng nhập trong hệ thống.	Do phụ thuộc vào nhà cung cấp bên thứ ba, tác giả chưa xử lý được trường hợp thông tin người dùng trả về không đầy đủ, ví dụ như thiếu địa chỉ email, ảnh đại diện và số điện thoại.
Đăng nhập bằng mã xác thực OTP	Thay vì cố gắng ghi nhớ và nhập mật khẩu trực tiếp, mã xác thực được gửi đến và người dùng kiểm tra hộp thư để lấy mã đăng nhập. Sau đó điền mã xác thực và truy cập vào hệ thống.	Phương pháp này tồn tại rủi ro khi tác giả chưa xử lý tốt trường hợp người khác xác định được email của người dùng, sau đó liên tục thực hiện các lần

		xác minh thất bại nhằm khiến tài khoản bị khóa. Hệ thống đã có cơ chế giới hạn số lượng request gửi xác thực. Tuy nhiên, kết quả kiểm thử mới chỉ thực hiện trong phạm vi nhỏ, nên chưa đủ để khẳng định hệ thống có thể chống chịu tốt trước lượng request rất lớn hoặc tấn công gửi request hàng loạt.
Đăng nhập bằng mã xác thực TOTP	Phương pháp này giúp giảm sự phụ thuộc vào việc gửi mã xác thực qua email trong mỗi lần đăng nhập. Mã xác thực được tạo trực tiếp trên ứng dụng xác thực của người dùng và tự động thay đổi theo thời gian, giúp quá trình đăng nhập diễn ra thuận tiện hơn.	Hệ thống chưa hỗ trợ thay đổi khóa bí mật. Do mã QR được tạo từ khóa bí mật này nên khi khóa không thay đổi, mã QR cũng cố định. Nếu mã QR bị lộ, người khác có thể tạo mã xác thực giống người dùng.

Đăng nhập bằng liên kết đăng nhập	Liên kết đăng nhập giúp người dùng đăng nhập mà không cần ghi nhớ mật khẩu hoặc nhập mã xác thực thủ công. Thay vào đó, hệ thống gửi một liên kết đăng nhập đến email và người dùng đăng nhập bằng cách bấm vào liên kết đó.	Việc đăng nhập sẽ khó khăn nếu người dùng nhận email trên máy tính nhưng lại muốn đăng nhập bằng điện thoại hoặc máy tính bảng nên phải chuyển sang thiết bị đang sử dụng, khiến thao tác trở nên phức tạp hơn. Hệ thống đã có cơ chế giới hạn request gửi liên kết xác thực. Tuy nhiên, kết quả kiểm thử mới chỉ thực hiện trong phạm vi nhỏ, nên chưa đủ để khẳng định hệ thống có thể chống chịu tốt trước lượng request rất lớn hoặc tấn công gửi request hàng loạt.

3.4 Kết luận chương III

Với chương III tác giả đã mô tả các bước xây dựng mô hình thử nghiệm, triển khai và đánh giá các phương pháp không dùng mật khẩu xác thực. Kết quả đánh giá

cho thấy từng giải pháp có thế mạnh riêng. Bên cạnh đó, giải pháp vẫn còn một số hạn chế cần được cải thiện. Kết quả đánh giá cho thấy các giải pháp này góp phần đơn giản hóa thao tác và giúp quá trình đăng nhập trở nên thuận tiện hơn và giảm sự phụ thuộc vào mật khẩu truyền thống. Dù vậy, kết quả triển khai cho thấy giải pháp vẫn còn một số điểm cần hoàn thiện về phương diện an toàn. Từ các nội dung đã phân tích, chương tiếp theo sẽ tổng hợp nhận định, kết quả và định hướng phát triển cho đề án.

KẾT LUẬN

- Tóm tắt các kết quả đạt được

Thông qua đề tài "Nghiên cứu, triển khai phương pháp xác thực không mật khẩu trên ứng dụng web", tác giả tiến hành tổng hợp cơ sở lý thuyết về xác thực người dùng, những hạn chế của mật khẩu truyền thống và các phương thức xác thực không mật khẩu gồm đăng nhập bằng mã xác thực OTP, mã xác thực TOTP, liên kết đăng nhập, đăng nhập bằng passkey và đăng nhập bằng ứng dụng bên thứ ba. Các phương thức này được phân tích về cơ chế hoạt động, ưu điểm, hạn chế và so sánh với mật khẩu truyền thống theo các tiêu chí như tính bảo mật, trải nghiệm người dùng, chi phí vận hành, khả năng triển khai và mức độ tương thích.

Tác giả đã tìm hiểu các yêu cầu, thiết kế kiến trúc và xây dựng ứng dụng web bằng Java Spring MVC. Tác giả đã triển khai các phương thức xác thực không mật khẩu, đồng thời tích hợp các chức năng cơ bản như tạo mới tài khoản và cập nhật hồ sơ người dùng, phân quyền và quản lý nhân viên. Đặc biệt với các phương pháp không dùng mật khẩu xác thực, tác giả xây dựng mô-đun xác thực được tổ chức tách biệt với các thành phần xử lý nghiệp vụ khác trong hệ thống, qua đó hỗ trợ duy trì, chỉnh sửa và bổ sung các phương pháp không dùng mật khẩu xác thực khác.

Bên cạnh các chức năng đăng nhập, ứng dụng được bổ sung các cơ chế bảo vệ như kiểm soát số lượng yêu cầu gửi mã hoặc liên kết, giới hạn số lần xác thực thất bại, khóa tài khoản tạm thời. Đề tài cũng áp dụng một phần nguyên tắc Zero Trust

trong quản lý phiên bằng cách theo dõi địa chỉ IP, vị trí địa lý và user agent nhằm ghi nhận các thay đổi bất thường trong quá trình sử dụng. Và cuối cùng là đánh giá tổng quan theo các tiêu chí về tính an toàn, tính thuận tiện, tính ổn định và khả năng tích hợp. Từ đó, đề án xác định những ưu điểm và hạn chế của giải pháp cần cải thiện.

- Hướng phát triển tiếp theo

Mặc dù đề tài đã triển khai và kiểm chứng một số phương thức xác thực không sử dụng mật khẩu trên ứng dụng web, song vẫn tồn tại một số điểm chưa hoàn thiện và cần được nghiên cứu, điều chỉnh thêm trước khi triển khai ở quy mô thực tế.

Trước hết, tác giả cần tiếp tục cải thiện các vấn đề sau: Đối với phương pháp đăng nhập bằng mã xác thực TOTP, hệ thống cần hỗ trợ cấp lại khóa bí mật khi mã QR có nguy cơ bị người khác biết. Đối với đăng nhập bằng passkey, hệ thống cần nghiên cứu thêm cách xử lý trường hợp nhiều người dùng chung một thiết bị hoặc cùng có khả năng mở khóa thiết bị. Đặc biệt, đối với phương pháp đăng nhập bằng mã OTP và liên kết đăng nhập, hệ thống cần được kiểm thử với số lượng request lớn hơn để đánh giá khả năng chịu tải, khả năng giới hạn request và khả năng duy trì hoạt động ổn định khi có nhiều yêu cầu gửi mã hoặc gửi liên kết xác thực cùng lúc. Việc kiểm thử này giúp đánh giá rõ hơn mức độ hiệu quả của hệ thống trước các hành vi gửi yêu cầu xác thực hàng loạt, vượt ngoài phạm vi thử nghiệm hiện tại của đề án.

Ngoài ra, tác giả định hướng phát triển tính năng đăng nhập dựa trên ngữ cảnh nhằm xác định mức độ an toàn của từng phiên đăng nhập. Cụ thể, các yếu tố được hệ thống tiếp nhận và xử lý nhằm phục vụ việc đánh giá như địa chỉ IP, vị trí địa lý, user agent và dấu vân tay trình duyệt để tính điểm rủi ro cho mỗi lần đăng nhập. Dựa trên điểm rủi ro này, hệ thống sẽ xử lý và đưa ra một trong các phương pháp xác thực phù hợp với từng ngữ cảnh đăng nhập, qua đó những phát hiện bất thường được phát hiện kịp thời ngăn chặn các yêu cầu truy cập không hợp lệ.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] V. Parmar, H. Sanghvi, R. Patel, A. Pandya (2022). "A Comprehensive Study on Passwordless Authentication", *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, India.

[2] Specops Software (2025). *Specops breached password report 2025*, 15 trang

[3] Caria, Davide (2025), "Transition to Passwordless Technologies: A Comprehensive Analysis and Real-World Implementation", *Politecnico di Torino*, Italy.

[4] M. Raza, M. Iqbal, M. Sharif, W. Haider (2012), "A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication", *World Applied Sciences Journal 19,* no. 4, pp. 439–444.

[5] Z. Alkhalil, C. Hewage, L. Nawaf, I. Khan (2021), "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy", *Frontiers in Computer Science 3,* no. 563060.

[6] Kama, Parves (2016), "State of the Art Survey on Session Hijacking", *Global Journal of Computer Science and Technology: E Network, Web & Security 16,* no. 1, pp. 1 - 12.

[7] https://auth0.com/docs/get-started/authentication-and-authorization-flow, truy cập ngày 20 tháng 5 năm 2026.

[8] Jyotia, S. Rajput, N. Tiwaric (2025), "Beyond Passwords: Trends and Innovationsin Password less Authentication", *Proceedings of the International Conference on Innovative Computing & Communication (ICICC 2024)*, India.

[9] Verizon (2017). *2017 Data Breach Investigations Report,* Verizon.

[10] Keepersecurity (2017). *2017 Consumer Mobile Security App Use*, Keepersecurity.

[11] https://fidoalliance.org/passkeys, truy cập ngày 22 tháng 5 năm 2026.

[12] https://auth0.com/intro-to-iam/what-is-oauth-2, truy cập ngày 25 tháng 5 năm 2026.

[13] A. Mustafa, Z. Huma (2024), "Zero Trust Security in Web Applications: Implementing Secure Authentication and Access Control", *Pioneer Research Journal of Computing Science 1*, no. 3, pp. 71 - 79.

[14] S. Mushtaq, M. Mohsin, M. Mushtaq (2025), "A Systematic Literature Review on the Implementation and Challenges of Zero Trust Architecture Across Domains", *Sensors 25*, no. 19, pp. 6118.

[15] T. S. Bernard, T. Hsu, N. Perlroth, R. Lieber (2017), "Equifax Says Cyberattack May Have Affected 143 Milion in the U.S", *The New York Times*.

[16] Freed, Benjamin (2019), "One Year After Atlanta's Ransomware Attack, the City Says It's Transforming Its Technology", *StateScoop*.

[17] E. Perez, W. Bruer (2015), "OPM Director Katherine Archuleta Steps Down", *CNN*.

[18] J. Pollard, D. Holmes, C. Cunningham, J. Blankenship, M. Cyr, P. Dostie (2019), "The Eight Business and Security Benefits of Zero Trust", *Forrester Research*.

[19] https://docs.spring.io/spring-security/reference/servlet/authentication/passkeys.html, truy cập ngày 22 tháng 5 năm 2026.

[20] https://docs.spring.io/spring-security/reference/servlet/oauth2/, truy cập ngày 22 tháng 5 năm 2026.

[21] Knivets, Dmytro (2026). Open Redirect, *OWASP Foundation*.

[22] World Wide Web Consortium (2021). Web Authentication: An API for Accessing Public Key Credentials – Level 2, *World Wide Web Consortium*

[23] https://docs.spring.io/spring-security/reference/servlet/authentication/architecture.html, truy cập ngày 23 tháng 5 năm 2026.

[24] https://docs.spring.io/spring-security/reference/servlet/authentication/session-management.html, truy cập ngày 23 tháng 5 năm 2026.

[25] https://docs.spring.io/spring-framework/reference/integration/email.html, truy cập ngày 24 tháng 5 năm 2026.

[26] D. M'Raihi, S. Machani, M. Pei, J. Rydell (2011), "TOTP: Time-Based One-Time Password Algorithm", *Internet Engineering Task Force (IETF)*.

[27] https://docs.dfns.co/guides/developers/webauthn-configuration, truy cập ngày 25 tháng 5 năm 2026

[28] Defense Information Systems Agency (2025). The web server must set an absolute session timeout value of eight hours or less, *STIG Viewer*.

[29] A. Gómez-Boix, P. Laperdrix, B. Baudry (2018), "Hiding in the Crowd: An Analysis of the Effectiveness of Browser Fingerprinting at Large Scale", *Proceedings of the 2018 World Wide Web Conference*, France.

[30] https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html, truy cập ngày 28 tháng 5 năn 2026.

[31] https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/11-Testing_Multi-Factor_Authentication, truy cập ngày 27 tháng 5 năm 2026.

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm Turnitin một cách trung thực và đạt kết quả mức độ tương đồng 18% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà Nội, ngày 29 tháng 6 năm 2026

HỌC VIÊN CAO HỌC



Nguyễn Khắc Minh

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ
BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 1949/QĐ-HV ngày 07 tháng 07 năm 2026 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi...9...giờ...00...phút, ngày..22.. tháng 07 năm 2026 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Nguyễn Khắc Minh**

Tên đề án tốt nghiệp: **Nghiên cứu, triển khai phương pháp xác thực không mật khẩu cho ứng dụng Web**

Chuyên ngành: **Hệ thống thông tin** Mã số: **8480104**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ...05../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TS. Nguyễn Trọng Khánh**	Chủ tịch	
2	**TS. Đặng Ngọc Hùng**	Thư ký	
3	**TS. Trịnh Bảo Ngọc**	Phản biện 1	
4	**TS. Vũ Văn Thỏa**	Phản biện 2	
5	**TS. Nguyễn Văn Vinh**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên. (có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

1, So sánh và trình bày ưu nhược điểm của các cơ chế web authentication, cụ thể là TOTP và passkey?

2, Giải thích rõ hơn cơ chế hoạt động của giải pháp, trong work đề xuất trong thời gian?

3, Cách đối sánh giải pháp đề xuất dựa trên thực tế?

7. Trả lời của học viên:

1. Học viên trình bày chưa có chiều sâu giải thích vì nhiều điều chưa TOTP và phân tích.

2. Học viên mới tập trung vào nội framework ở web MVC, chưa tìm hiểu các nền tảng khác.

3. Học viên chưa thấy giá phần thấy lường trong đề xuất.

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS.TS. Nguyễn Trang Thiết
  2. Uỷ viên Ban kiểm phiếu: TS. Vũ Văn Thỏa
  3. Uỷ viên Ban kiểm phiếu: TS. Đặng Ngọc Hùng
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 7,5

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

Rà soát chính tả và các lỗi văn phong, trình bày như góp ý của Hội đồng

Chuẩn hóa các mục tài liệu tham khảo

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: Đề nghị công nhận

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....Không.....

Buổi làm việc kết thúc vào..... cùng ngày.

Chủ tịch **Thư ký**

Nguyễn Trang Thiết Đặng Ngọc Hùng

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài đề án tốt nghiệp: Nghiên cứu, triển khai phương pháp xác thực không mật khẩu cho ứng dụng Web

Chuyên ngành: Hệ thống thông tin
Mã chuyên ngành: 8.48.01.04
Họ và tên học viên: Nguyễn Khắc Minh
Họ và tên người nhận xét: Trịnh Bảo Ngọc
Học hàm, học vị: Tiến sĩ
Chuyên ngành: Kỹ thuật phần mềm
Cơ quan công tác: Trường đại học Hà Nội
Số điện thoại: 0987 355 799 E-mail: ngoctb@hanu.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Cơ sở khoa học: Đề tài được xây dựng trên nền tảng lý thuyết vững chắc về xác thực người dùng trong ứng dụng web, phân quyền truy cập và các phương pháp bảo mật hiện đại. Tác giả đã phân tích cơ chế hoạt động của xác thực bằng mật khẩu, chỉ ra các rủi ro như tấn công vét cạn, phishing, keylogger, đồng thời so sánh với các phương pháp xác thực không mật khẩu như OTP, TOTP, liên kết đăng nhập, passkey và WebAuthn. Việc tham chiếu mô hình Zero Trust và chuẩn OWASP giúp đề tài có cơ sở khoa học rõ ràng, đảm bảo tính hệ thống và phù hợp với xu hướng nghiên cứu bảo mật hiện nay.

Cơ sở thực tiễn: Đề tài xuất phát từ thực trạng người dùng phải ghi nhớ nhiều mật khẩu, dễ bị tấn công và gây bất tiện trong quản lý tài khoản. Các giải pháp không mật khẩu như OTP, liên kết đăng nhập, đăng nhập qua ứng dụng thứ ba hay passkey đã được triển khai thử nghiệm trên ứng dụng web mô phỏng, cho thấy khả năng áp dụng trong thực tế. Tác giả đã xây dựng giao diện, cơ sở dữ liệu, back-end và tiến hành kiểm thử nhiều kịch bản, từ đó đánh giá tính an toàn, thuận tiện, ổn định và khả năng tích hợp. Điều này chứng minh đề tài có giá trị thực tiễn cao.

Tính cấp thiết: Trong bối cảnh chuyển đổi số và gia tăng các cuộc tấn công mạng, việc nghiên cứu và triển khai giải pháp xác thực không mật khẩu là cấp thiết. Các hệ thống web hiện nay lưu trữ dữ liệu quan trọng, nếu chỉ dựa vào mật khẩu truyền thống sẽ tiềm ẩn nguy cơ rò rỉ và chiếm đoạt tài khoản. Đề tài hướng đến việc giảm phụ thuộc vào mật khẩu, nâng cao bảo mật và cải thiện trải nghiệm người dùng. Đây là nhu

cầu thực tế của tổ chức, doanh nghiệp và cá nhân, đặc biệt trong môi trường học thuật và quản lý dữ liệu nhạy cảm như đồ án, khóa luận tại Học viện.

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:

Kết cấu đề án:

Đề án gồm ba chương: cơ sở lý luận về xác thực và bảo mật, phân tích yêu cầu và đề xuất giải pháp không mật khẩu, thiết kế và xây dựng mô hình thử nghiệm, kiểm thử và đánh giá. Cuối cùng là kết luận, tài liệu tham khảo, đảm bảo cấu trúc khoa học và logic. Đề án được trình bày trong tổng cộng 89 trang.

Tóm tắt các kết quả đạt được:

Đề tài đã đạt được nhiều kết quả quan trọng. Về lý luận, tác giả đã hệ thống hóa cơ sở khoa học của các phương pháp xác thực, phân tích ưu nhược điểm của mật khẩu truyền thống và các giải pháp thay thế. Về thực tiễn, tác giả đã xây dựng một ứng dụng web thử nghiệm tích hợp nhiều phương pháp xác thực không mật khẩu như OTP, liên kết đăng nhập qua email, xác thực bằng ứng dụng Google, TOTP, quét mã QR và passkey. Các kịch bản kiểm thử được thực hiện cho thấy hệ thống có khả năng ngăn chặn truy cập trái phép, khóa tài khoản khi nhập sai nhiều lần, cảnh báo đăng nhập bất thường và xử lý tình huống OTP hoặc liên kết hết hạn.

Đề tài cũng đánh giá toàn diện theo các tiêu chí: an toàn, thuận tiện, ổn định và khả năng tích hợp. Kết quả cho thấy giải pháp không mật khẩu mang lại trải nghiệm tốt hơn, giảm rủi ro bảo mật, đồng thời vẫn còn một số hạn chế cần cải thiện như chi phí triển khai và yêu cầu hạ tầng. Đây là đóng góp tốt cho việc áp dụng trong thực tế.

III/ Những vấn đề cần giải thích thêm:

Về hình thức trình bày:

Đề án trình bày rõ ràng, có mục lục chi tiết, danh mục bảng và hình minh họa phong phú, giúp người đọc dễ theo dõi và nắm bắt nội dung.

Một số lỗi định dạng xuất hiện trong tài liệu. Cụ thể, ở trang 3, mục lục có lỗi căn chỉnh khi mục "1.4.2 Các rủi ro thực tế..." bị lệch dòng, không hiển thị số trang. Ở trang 7, bảng "Danh mục các bảng" có dòng bị ngắt không đúng vị trí, gây khó đọc, bảng "Danh mục các hình" có lỗi xuống dòng ở hình 3.17, khiến nội dung bị chia tách thành hai dòng không đồng bộ.

Ngoài ra, một số bảng không căn chỉnh đều, ví dụ bảng 1.1 ở trang 6 có phần mô tả dài gây mất cân đối giữa hai cột. Các lỗi này tuy không ảnh hưởng lớn đến nội dung nhưng làm giảm tính chuyên nghiệp của hình thức trình bày.

Về nội dung:

Một số vấn đề về kiến thức và logic cần lưu ý. Ở trang 5, phần mô tả quá trình xác thực chưa phân biệt rõ giữa xác thực một yếu tố và đa yếu tố, khiến người đọc dễ nhầm lẫn.
Trang 7, phần "Vai trò của xác thực" liệt kê nhiều ý nhưng chưa phân tích sâu mối liên hệ giữa xác thực và mức độ quan trọng tới chất lượng dự án, sự quan trọng với doanh nghiệp sử dụng, chỉ dừng ở mức khái quát.
Đề tài làm về Không mật khẩu tuy nhiên tổng quan về Có dùng mật khẩu lại quá dài (từ trang 7 tới trang 14, gần 1/10 của luận án) là không hợp lý và phần chính lại không được mô tả đủ kỹ.
Trang 9, Hình 1.1 minh họa quá trình đăng ký và xác thực bằng mật khẩu nhưng chưa giải thích chi tiết các bước, gây thiếu tính trực quan.
Trang 25, bảng 1.2 so sánh mật khẩu và không mật khẩu nhưng chưa đề cập đến chi phí triển khai, một yếu tố quan trọng trong thực tế.
Trang 41, phần thiết kế kiến trúc mô hình thử nghiệm mới dừng ở mức tổng quát, chưa phân tích chi tiết các thành phần bảo mật như cơ chế mã hóa dữ liệu truyền tải.
Câu hỏi:

1. Cơ chế WebAuthn hoạt động thế nào trong việc tạo và xác minh passkey?
2. So sánh ưu nhược điểm giữa xác thực OTP và xác thực bằng passkey?

IV/ Kết luận:
Đồng ý.

Ngày 21 tháng 07 năm 2026
NGƯỜI NHẬN XÉT

TS Trịnh Bảo Ngọc

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Tên đề tài đề án: Nghiên cứu, triển khai phương pháp xác thực không mật khẩu cho ứng dụng web

Chuyên ngành: Hệ thống thông tin **Mã số:** 8.48.01.04

Tên học viên: Nguyễn Khắc Minh

Họ tên người nhận xét: Vũ Văn Thỏa

Học hàm, học vị: Tiến sĩ **Chuyên ngành**: Toán học tính toán

Cơ quan công tác: Học viện Công nghệ Bưu chính Viễn thông

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài

Trong bối cảnh Việt Nam đang đẩy mạnh chuyển đổi số, phát triển Chính phủ số, ngân hàng số và thương mại điện tử, yêu cầu bảo vệ danh tính số, tài khoản người dùng và các dịch vụ trực tuyến ngày càng trở nên cấp thiết. Việc nghiên cứu và triển khai các giải pháp xác thực không mật khẩu không chỉ có ý nghĩa khoa học mà còn có giá trị thực tiễn cao, góp phần nâng cao mức độ an toàn thông tin, cải thiện trải nghiệm người dùng và tạo cơ sở để các cơ quan, tổ chức, doanh nghiệp triển khai các phương thức xác thực hiện đại, phù hợp với xu hướng phát triển công nghệ hiện nay.

Đề tài "Nghiên cứu, triển khai phương pháp xác thực không mật khẩu cho ứng dụng web" của học viên Nguyễn Khắc Minh hướng tới mục tiêu nghiên cứu, triển khai và đánh giá các phương pháp xác thực không mật khẩu nhằm nâng cao mức độ bảo mật và khả năng ứng dụng cho các hệ thống web cung cấp các dịch vụ trực tuyến. Đề tài được xây dựng trên cơ sở các công nghệ và tiêu chuẩn xác thực hiện đại như OAuth2, OpenID Connect, WebAuthn, FIDO2 và mô hình bảo mật Zero Trust, góp phần khắc phục những hạn chế của phương pháp xác thực bằng mật khẩu truyền thống trước các nguy cơ tấn công như Phishing, Brute Force và Credential Stuffing. Vì vậy, đề tài có tính cấp thiết, có cơ sở khoa học và mang ý nghĩa thực tiễn cao, đáp ứng yêu cầu bảo đảm an toàn thông tin trong bối cảnh chuyển đổi số hiện nay.

II/ Về nội dung chất lượng của đề án, những kết quả đạt được

Đề án được trình bày trong 79 trang bao gồm phần mở đầu, 3 chương nội dung, phần kết luận và danh mục các tài liệu tham khảo.

Chương 1 của đề án gồm 22 trang, trình bày cơ sở lý luận về xác thực người dùng trên ứng dụng web, phân tích những hạn chế của phương pháp xác thực bằng mật khẩu truyền thống và giới thiệu các phương pháp xác thực không mật khẩu như OTP, TOTP, Magic Link, OAuth2, Passkey/WebAuthn và mô hình Zero Trust. Nội dung được trình bày logic, làm rõ ưu điểm và hạn chế của từng phương pháp, đồng thời tạo cơ sở cho việc lựa chọn giải pháp triển khai. Tuy nhiên, chương này cần bổ sung thêm các tiêu chuẩn quốc tế như NIST SP 800-63B và OWASP

Authentication Cheat Sheet; đồng thời phân tích sâu hơn các nguy cơ an toàn thông tin như Replay Attack, Session Hijacking, Token Theft, ... để tăng tính học thuật và cập nhật.

Nội dung chính của chương 2 được trình bày trong 15 trang tập trung phân tích yêu cầu của hệ thống và đề xuất giải pháp triển khai trên nền tảng Spring Boot và Spring Security. Đề án đã xây dựng kiến trúc hệ thống tương đối hợp lý, lựa chọn các công nghệ hiện đại và phân tích khả năng tích hợp nhiều phương thức xác thực không mật khẩu: phương pháp xác thực bằng mã OTP, bằng liên kết đăng nhập, bằng mã xác thực TOTP, bằng ứng dụng bên thứ ba hoặc bằng Passkey. Tuy nhiên, cần bổ sung mô hình phân tích mối đe dọa (Threat Modeling) theo STRIDE hoặc Attack Tree, xác định rõ các tài nguyên cần bảo vệ, ranh giới tin cậy (Trust Boundary), cơ chế quản lý khóa, cũng như phân tích khả năng dự phòng và tính sẵn sàng của hệ thống khi các dịch vụ xác thực hoặc dịch vụ thư điện tử gặp sự cố.

Chương 3 của đề án gồm 35 trang, trình bày quá trình xây dựng, triển khai và đánh giá hệ thống thực nghiệm. Đề án đã xây dựng thành công ứng dụng tích hợp nhiều phương thức xác thực không mật khẩu như Passkey, OTP, TOTP, Magic Link và OAuth2; đồng thời mô tả quy trình triển khai, giao diện và khả năng tích hợp với Spring Security. Kết quả cho thấy hệ thống hoạt động đúng mục tiêu đề ra và có khả năng áp dụng trong thực tế. Tuy nhiên, chương này cần bổ sung các kết quả đánh giá định lượng về hiệu năng (thời gian xác thực, độ trễ, thông lượng), kiểm thử khả năng chịu tải bằng các công cụ như Apache JMeter hoặc Gatling, kiểm thử bảo mật bằng OWASP ZAP hoặc Burp Suite, đồng thời xây dựng bảng so sánh định lượng giữa các phương pháp xác thực về mức độ bảo mật, trải nghiệm người dùng, hiệu năng và chi phí triển khai.

Đề án được chia thành các chương mục rõ ràng và cơ bản bám sát các nội dung theo đề cương đã được phê duyệt. Tuy nhiên, cần lưu ý trình bày các chương mục theo đúng quy định. Phần Mở đầu cần viết gọn lại theo quy định, phần Danh mục tài liệu tham khảo lược bớt các tài liệu không trực tiếp sử dụng trong đề án.

Trong đề án còn có khá nhiều lỗi in ấn phải sửa chữa tại các trang vi, 1, 2, 22, 45, … Một số hình vẽ còn mờ (trong mục 3.2) cần hiệu chỉnh hoặc in màu.

III/ Những vấn đề cần giải thích thêm

(1) Giải thích rõ hơn về áp dụng giải pháp đề xuất trong đề án cho ứng dụng web được xây dựng trên các công nghệ khác nhau?

(2) Giải thích thêm về các đánh giá định lượng các kết quả thực nghiệm như thời gian đăng nhập, tỷ lệ đăng nhập thành công, khả năng phục hồi tài khoản khi có sự cố, …?

IV/ Kết luận

Tôi đồng ý cho học viên Nguyễn Khắc Minh được bảo vệ đề án với đề tài nêu trên trước Hội đồng chấm đề án tốt nghiệp thạc sĩ của Học viện. Tuy nhiên, để đảm bảo chất lượng đề án tốt nghiệp thạc sĩ chuyên ngành Hệ thống thông tin, học viên cần chỉnh sửa đề án và giải trình rõ trước Hội đồng các vấn đề được góp ý trên đây.

Ngày 20 tháng 07 năm 2026

Người nhận xét

Vũ Văn Thỏa

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP

Họ và tên học viên: Nguyễn Khắc Minh
Chuyên ngành: Hệ thống thông tin
Khóa: 2024 đợt 2

Tên đề tài: Nghiên cứu, triển khai phương thức xác thực không mật khẩu cho ứng dụng Web.

Người hướng dẫn khoa học: TS. Đinh Trường Duy

Ngày bảo vệ: 22/07/2026

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Rà soát, chỉnh sửa các lỗi văn phong, trình bày theo góp ý của Hội đồng.	Học viên đã sửa các lỗi văn phong trình bày. Đặc biệt đã căn chỉnh mục 1.4.2 để không bị lệch dòng. Mục bảng 1.2 và mục hình 3.17 được làm ngắn tiêu đề để không bị ngắt và khó đọc. Các trang có hình ảnh được in màu.
2	Chuẩn hóa danh mục tài liệu tham khảo.	Tác giả đã xem, định dạng lại để chuẩn hóa danh mục tài liệu tham khảo.

Hà Nội, ngày tháng năm 2026

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS. TS. Nguyễn Trọng Khánh	TS. Đặng Ngọc Hùng	TS. Đinh Trường Duy	Nguyễn Khắc Minh

18% Tính tương đồng nói chung

Tổng cộng của tất cả các kết quả trùng khớp, bao gồm cả các nguồn trùng lặp, cho mỗi c...

Đã lọc khỏi Báo cáo

- Mục lục tham khảo
- Văn bản được trích dẫn
- Văn bản được trích dẫn
- Kết quả trùng khớp nhỏ (ít hơn 8 từ)

Nguồn hàng đầu

16% Nguồn Internet
3% Ấn bản
4% Bài tập được nộp (bài của học sinh)

Học viên thực hiện	Giáo viên hướng dẫn
Nguyễn Khắc Minh	TS. Đinh Trường Duy