

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
HỆ THỐNG HỖ TRỢ HỌC FLASHCARD

Người hướng dẫn: **ThS. ĐỖ THỊ TUYẾT HOA**
Sinh viên thực hiện: **LÊ TẤT TUẤN**
Số thẻ sinh viên: 102210285
Lớp: 21TCLC_DT4

Đà Nẵng, 01/2026

NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

I. Thông tin chung:

- Họ và tên sinh viên: Lê Tất Tuấn
- Lớp: 21TCLC_DT4 Số thẻ SV: 102210285
- Tên đề tài: Hệ thống hỗ trợ học flashcard
- Người hướng dẫn: ThS.Đỗ Thị Tuyết Hoa Học hàm/ học vị: Thạc Sĩ

II. Nhận xét, đánh giá đồ án tốt nghiệp:

- Về tính cấp thiết, tính mới, khả năng ứng dụng của đề tài: (điểm tối đa là 2đ)

.....
.....

- Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án: (điểm tối đa là 4đ)

.....
.....

- Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp: (điểm tối đa là 2đ)

.....
.....

- Đề tài có giá trị khoa học/ có bài báo/ giải quyết vấn đề đặt ra của doanh nghiệp hoặc nhà trường: (điểm tối đa là 1đ)

.....
.....

- Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:

.....
.....

III. Tinh thần, thái độ làm việc của sinh viên: (điểm tối đa 1đ)

.....

IV. Đánh giá:

- Điểm đánh giá:/10 (lấy đến 1 số lẻ thập phân)
- Đề nghị: Được bảo vệ đồ án Bổ sung để bảo vệ Không được bảo vệ

Đà Nẵng, ngày tháng năm 2026

Người hướng dẫn

TÓM TẮT

Tên đề tài: Hệ thống hỗ trợ học flashcard

Sinh viên thực hiện: Lê Tất Tuấn

Lớp: 21TCLC_DT4

Số thẻ SV: 102210285

Trong thời đại công nghệ số, nhu cầu học tập và ghi nhớ kiến thức một cách hiệu quả ngày càng được chú trọng. Các phương pháp học truyền thống thường thiếu tính tương tác và không tối ưu cho việc ghi nhớ dài hạn. Phương pháp flashcard (thẻ ghi nhớ) đã được chứng minh là một trong những phương pháp học hiệu quả nhất, đặc biệt khi kết hợp với thuật toán Spaced Repetition – lặp lại ngắt quãng theo chu kỳ ghi nhớ của não bộ.

Tuy nhiên, hiện nay đa số các ứng dụng flashcard trên thị trường còn thiếu tính cá nhân hóa, giao diện phức tạp.

Do đó, đề tài “Hệ thống hỗ trợ học flashcard” được chọn nhằm xây dựng một ứng dụng học tập thân thiện, thông minh và hiệu quả, có khả năng gợi ý flashcard tự động bằng AI, giúp người dùng học nhanh – nhớ lâu – và quản lý kiến thức dễ dàng hơn.

Xuất phát từ nhu cầu đó, đề tài “Hệ thống hỗ trợ học flashcard” được nghiên cứu và phát triển nhằm giải quyết bài toán tối ưu hóa hiệu suất học tập cá nhân.

Mục tiêu cốt lõi của đề tài là xây dựng một nền tảng website hoàn chỉnh, tích hợp sâu **thuật toán Lặp lại ngắt quãng (Spaced Repetition System - SRS)** – cụ thể là biến thể của thuật toán SM-2/FSRS. Hệ thống sẽ tự động tính toán và điều chỉnh lịch trình ôn tập dựa trên lịch sử ghi nhớ của người dùng, đảm bảo kiến thức được ôn lại ngay trước thời điểm "quên", giúp chuyển đổi thông tin từ trí nhớ ngắn hạn sang dài hạn một cách bền vững.

Điểm đột phá của hệ thống so với các ứng dụng truyền thống là việc ứng dụng **Trí tuệ nhân tạo (Generative AI)** vào quy trình tạo nội dung. Thay vì phải nhập liệu thủ công tốn thời gian, hệ thống cho phép người dùng nhập văn bản thô, file tài liệu; sau đó AI sẽ tự động phân tích, trích xuất ý chính và tạo ra các bộ thẻ ghi nhớ (flashcard) cùng các câu hỏi trắc nghiệm (quiz) tương ứng chỉ trong vài giây.

Bên cạnh đó, hệ thống cũng chú trọng đến trải nghiệm người dùng (UX/UI) thông qua giao diện trực quan, các tính năng thống kê tiến độ học tập (heatmap, biểu đồ ghi nhớ) và cơ chế Gamification (trò chơi hóa) để duy trì động lực học tập.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Lê Tất Tuấn

Số thẻ sinh viên: 102210285

Lớp: 21TCLC_DT4

Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

1. Tên đề tài đồ án: Hệ thống hỗ trợ học flashcard

2. Đề tài thuộc diện: Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Không có.

4. Nội dung các phần thuyết minh và tính toán:

Nội dung của thuyết minh gồm:

Mở đầu. Giới thiệu tổng quan đề tài, mục đích, ý nghĩa và yêu cầu đặt ra của đề tài, phương pháp thực hiện và bố cục của đồ án.

Chương 1. Cơ sở lý thuyết: Trình bày những cơ sở lý thuyết được áp dụng trong đề tài.

Chương 2. Phân tích và thiết kế: trình bày các hồ sơ phân tích và hồ sơ thiết kế trong xây dựng hệ thống và luồng hoạt động của hệ thống.

Chương 3. Triển khai và đánh giá kết quả: Trình bày cách triển khai, cài đặt vận hành hệ thống và đánh giá kết quả đạt được.

Chương 4: Đánh giá kết quả đạt được, hạn chế và hướng phát triển.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

Không có.

6. Họ tên người hướng dẫn: ThS. Đỗ Thị Tuyết Hoa

7. Ngày giao nhiệm vụ đồ án:/...../2025

8. Ngày hoàn thành đồ án:/...../2026

Đà Nẵng, ngày tháng năm 2026

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Sự bùng nổ của công nghệ thông tin trong kỷ nguyên số đã và đang tạo ra những thay đổi mang tính bước ngoặt trong lĩnh vực giáo dục. Việc ứng dụng công nghệ để tối ưu hóa khả năng ghi nhớ và cá nhân hóa lộ trình học tập không còn là một xu hướng, mà đã trở thành nhu cầu thiết yếu. Xuất phát từ mong muốn giải quyết những hạn chế của phương pháp học tập truyền thống, đồng thời ứng dụng các kỹ thuật tiên tiến như Lặp lại ngắt quãng (Spaced Repetition) và Trí tuệ nhân tạo (AI), em đã lựa chọn và thực hiện đề tài: “**Xây dựng hệ thống hỗ trợ học flashcard**” làm đề án tốt nghiệp của mình.

Để hoàn thành đề án này, bên cạnh sự nỗ lực của bản thân, em đã nhận được sự quan tâm, động viên và hướng dẫn tận tình từ rất nhiều thầy cô, gia đình và bạn bè.

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến cô **ThS. Đỗ Thị Tuyết Hoa**. Cô đã tận tình hướng dẫn, định hướng và đưa ra những lời khuyên quý báu giúp em vượt qua những khó khăn trong suốt quá trình nghiên cứu và phát triển ứng dụng.

Em xin trân trọng cảm ơn quý Thầy, Cô trong khoa **Công Nghệ Thông Tin**, trường **Đại Học Bách Khoa – Đại Học Đà Nẵng** đã trang bị cho em những kiến thức nền tảng vững chắc trong suốt những năm tháng ngồi trên ghế nhà trường. Những kiến thức ấy chính là hành trang vô giá để em có thể hoàn thành đề án này cũng như vững bước trên con đường sự nghiệp sau này.

Cuối cùng, em xin gửi lời tri ân đến gia đình và bạn bè, những người đã luôn ở bên cạnh động viên, chia sẻ và tạo mọi điều kiện thuận lợi nhất để em hoàn thành tốt nhiệm vụ học tập.

Mặc dù đã rất nỗ lực cố gắng, song do giới hạn về mặt thời gian cũng như kiến thức và kinh nghiệm thực tế còn hạn chế, đề án chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp ý kiến, chỉ bảo của quý Thầy, Cô và các bạn để đề tài được hoàn thiện hơn và có thể phát triển thành một sản phẩm thực tế hữu ích trong tương lai.

Em xin chân thành cảm ơn!

CAM ĐOAN

Tôi xin cam đoan:

1. Báo cáo đồ án tốt nghiệp Tên đề tài: Hệ thống hỗ trợ học flashcard là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của giảng viên ThS. Đỗ Thị Tuyết Hoa.
2. Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.
3. Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Lê Tất Tuấn

MỤC LỤC

MỤC LỤC.....	iii
DANH SÁCH HÌNH ẢNH.....	vi
DANH SÁCH BẢNG.....	vii
MỞ ĐẦU.....	1
1. Mục đích thực hiện đề tài.....	1
2. Mục tiêu và nhiệm vụ đề tài.....	2
3. Phạm vi nghiên cứu.....	3
4. Đối tượng và phương pháp nghiên cứu.....	3
5. Ý nghĩa thực tiễn.....	4
6. Bố cục đồ án.....	4
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	5
1.1 Tổng quan về phương pháp học tập Flashcard.....	5
1.1.1 Khái niệm flashcard và E-learning.....	5
1.1.2 Thuật toán lặp lại ngắt quãng (Spaced Repetition System).....	6
1.2 Phân tích các nền tảng tạo Flashcard tương tự.....	8
1.2.1 Phân tích nền tảng Quizlet.....	8
1.2.2 Phân tích phần mềm Anki.....	9
1.2.2 Đánh giá ưu nhược điểm và bài học kinh nghiệm.....	10
1.3 Các công nghệ phát triển hệ thống.....	11
1.3.1 Framework NestJS (Node.js).....	12
1.3.2 Kiến trúc hệ thống Backend.....	13
1.3.3 Hệ quản trị cơ sở dữ liệu PostgreSQL.....	14
1.3.4 Công nghệ tích hợp AI (Generative AI).....	14
1.3.5 Quản lý hàng đợi tác vụ với BullMQ và Redis.....	15
1.3.6 Công nghệ Container hóa với Docker.....	16
1.3.7 Web Server và Reverse Proxy với Nginx.....	16
1.3.8 Framework phát triển giao diện Vue.js.....	17
1.3.9 Cơ chế xác thực người dùng Google OAuth 2.0.....	19
1.3.10 Tự động hóa quy trình phát triển với Github Actions (CI/CD).....	20
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	22
2.1 Phân tích yêu cầu chức năng.....	22
2.1.1 Nhóm chức năng Quản lý Tài khoản & Hệ thống.....	22
2.1.2 Nhóm chức năng Quản lý Nội dung (Content Management).....	22
2.1.3 Nhóm chức năng Học tập & Ôn luyện (Study & SRS Engine).....	23
2.2 Phân tích yêu cầu phi chức năng.....	24
2.2.1 Hiệu năng (Performance).....	24
2.2.2 Tính Bảo mật (Security).....	24
2.2.3 Tính Khả dụng & Tương thích (Usability & Compatibility).....	24
2.3 Biểu đồ use case.....	25
2.3.1 Biểu đồ use case tổng quan.....	25
2.3.2 Biểu đồ use case Quản lý thông tin cá nhân.....	26

2.3.3 Biểu đồ use case Quản lý bộ thẻ.....	27
2.3.4 Biểu đồ use case Quản lý thẻ.....	28
2.3.5 Biểu đồ use case Học Flashcard.....	29
2.3.6 Biểu đồ use case Quản lý thư mục.....	30
2.3.7 Biểu đồ use case Quản lý nhóm.....	31
2.4 Đặc tả biểu đồ Use – Case.....	32
2.4.1 Use case Đăng nhập bằng Google (Google Oauth).....	32
2.4.2 Use case Quản lý thông tin cá nhân.....	33
2.4.3 Use case Quản lý bộ thẻ.....	35
2.4.4 Use case Tạo thẻ tự động từ prompt.....	36
2.4.5 Use case Tạo thẻ tự động từ tài liệu (File Upload).....	38
2.4.6 Use case Quản lý thẻ.....	40
2.4.7 Use case Học flashcard.....	41
2.4.8 Use case Quản lý thư mục.....	42
2.4.9 Use case Dashboard thống kê.....	43
2.4.10 Use case Chế độ ôn tập Trắc nghiệm & Game (Gamification).....	45
2.4.11 Use case Quản lý nhóm.....	47
2.5 Biểu đồ hoạt động.....	48
2.5.1 Biểu đồ hoạt động của chức năng đăng nhập bằng Google OAuth.....	48
2.5.2 Biểu đồ hoạt động của chức năng quản lý thông tin cá nhân.....	49
2.5.3 Biểu đồ hoạt động của chức năng tạo bộ thẻ từ prompt.....	50
2.5.4 Biểu đồ hoạt động của chức năng tạo bộ thẻ từ file tài liệu PDF.....	51
2.5.5 Biểu đồ hoạt động của chức năng tự động bổ sung phát âm (Text-to-Speech Enrichment).....	52
2.5.6 Biểu đồ hoạt động của chức năng học flashcard.....	53
2.6 Biểu đồ tuần tự.....	54
2.6.1 Biểu đồ tuần tự của chức năng đăng nhập bằng Google Oauth.....	54
2.6.2 Biểu đồ tuần tự chức năng quản lý thông tin cá nhân.....	55
2.6.3 Biểu đồ tuần tự chức năng tạo bộ thẻ từ prompt.....	56
2.6.4 Biểu đồ tuần tự chức năng tạo bộ thẻ từ file pdf.....	56
2.6.5 Biểu đồ tuần tự của chức năng tự động bổ sung phát âm (Text-to-Speech Enrichment).....	57
2.6.6 Biểu đồ hoạt động của chức năng học flashcard.....	58
2.7 Cơ sở dữ liệu.....	59
2.7.1 Bảng Người dùng (user).....	60
2.7.2 Bảng Cài đặt người dùng (user_setting).....	61
2.7.3 Bảng Vai trò (role) & Phân quyền (permission).....	62
2.7.4 Bảng Phiên đăng nhập (session) & Thiết bị (device).....	63
2.7.5 Bảng Token thiết bị (device_token).....	65
2.7.6 Bảng Bộ thẻ (card_set).....	66
2.7.7 Bảng Thẻ ghi nhớ (card).....	67
2.7.8 Bảng Lựa chọn trả lời (card_option).....	68
2.7.9 Bảng Tiến độ học tập (study_progress).....	69
2.7.10 Bảng Phiên học (learning_sessions).....	70
2.7.11 Bảng Lịch sử ôn tập (review_logs).....	71
2.7.12 Bảng Thông báo (notification).....	72

2.7.13 Bảng Lịch sử gửi thông báo (notification_attempt).....	73
2.7.14 Bảng Quản lý File (file).....	74
CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG	75
3.1 Triển khai hệ thống.....	75
3.1.1 Kiến trúc triển khai (Deployment Architecture).....	75
3.1.2 Môi trường và công nghệ.....	76
3.1.3 Quy trình CI/CD (Tích hợp và Triển khai liên tục).....	76
3.2 Kết quả giao diện chương trình.....	77
3.2.1 Màn hình đăng nhập & dash board.....	77
3.2.2 Màn hình Quản lý Bộ thẻ.....	78
3.2.3 Màn hình Tạo thẻ bằng AI.....	78
3.2.4 Màn tạo flashcard từ file pdf.....	80
3.2.5 Màn hình Xem chi tiết bộ thẻ.....	80
3.2.6 Màn hình Ôn tập (Flashcard Study).....	80
3.2.7 Màn hình Quiz mode.....	81
3.2.8 Màn hình cấu hình cho bài test.....	82
3.2.9 Màn hình làm bài test.....	83
3.2.10 Màn hình kết quả bài test.....	83
3.3 Tài liệu API cho lập trình viên.....	84
3.4 Kiểm thử hệ thống (Testing).....	85
3.4.1 Kịch bản kiểm thử: Đăng nhập Google.....	85
3.4.2 Kịch bản kiểm thử: Tạo thẻ bằng AI.....	85
3.4.3 Kịch bản kiểm thử: Thuật toán SRS.....	86
3.5 Đánh giá chất lượng hệ thống.....	87
3.5.1 Đánh giá về mặt chức năng (Functional Assessment).....	87
3.5.2 Đánh giá về mặt hiệu năng (Performance Assessment).....	87
3.5.3 Đánh giá về trải nghiệm người dùng (UX/UI Assessment).....	88
3.6 Kết chương.....	89
CHƯƠNG 4: KẾT LUẬN	90
4.1 Những kết quả đạt được.....	90
4.1.1 Về mặt lý thuyết.....	90
4.1.2 Về mặt ứng dụng.....	91
4.2 Hạn chế.....	91
4.3 Hướng phát triển.....	92
TÀI LIỆU THAM KHẢO	93

DANH SÁCH HÌNH ẢNH

Công thức cập nhật hệ số dễ nhớ.....	7
Công thức tính khoảng cách ngày.....	7
Sơ đồ tổng quan kiến trúc hệ thống.....	11
Biểu đồ use case tổng quan của hệ thống.....	25
Biểu đồ use case Quản lý thông tin cá nhân.....	26
Biểu đồ use case quản lý bộ thẻ.....	27
Biểu đồ use case quản lý thẻ.....	28
Biểu đồ use case học flashcard.....	29
Biểu đồ use case quản lý thư mục.....	30
Biểu đồ use case quản lý nhóm.....	31
Biểu đồ hoạt động của chức năng đăng nhập bằng google.....	48
Biểu đồ hoạt động chức năng quản lý thông tin cá nhân.....	49
Biểu đồ hoạt động chức năng tạo bộ thẻ từ prompt.....	50
Biểu đồ hoạt động chức năng tạo bộ thẻ từ file pdf.....	51
Biểu đồ hoạt động chức năng tự động bổ sung phát âm.....	52
Biểu đồ hoạt động chức năng học flashcard.....	53
Biểu đồ tuần tự chức năng đăng nhập bằng google.....	54
Biểu đồ tuần tự chức năng quản lý thông tin cá nhân.....	55
Biểu đồ tuần tự chức năng tạo thẻ từ prompt.....	56
Biểu đồ tuần tự chức năng tạo bộ thẻ từ file pdf.....	56
Biểu đồ tuần tự chức năng tự động bổ sung phát âm.....	57
Biểu đồ tuần tự chức năng học flashcard.....	58
Lược đồ cơ sở dữ liệu.....	59
Kiến trúc triển khai hệ thống.....	75
Giao diện dashboard.....	77
Giao diện quản lý bộ thẻ.....	78
Giao diện tạo thẻ bằng AI.....	79
Màn hình tạo flashcard từ file pdf.....	80
Giao diện thống kê tiến độ học tập theo bộ thẻ.....	80
Giao diện học flashcard.....	81
Giao diện học theo chế độ quiz.....	81
Giao diện cài đặt bài test.....	82
Giao diện làm bài test.....	83
Giao diện kết quả bài test.....	83
Api Docs - Swagger.....	84

DANH SÁCH BẢNG

Bảng so sánh Quizlet & Anki.....	10
Đặc tả use-case đăng nhập bằng google.....	32
Đặc tả use-case quản lý thông tin cá nhân.....	33
Đặc tả use-case quản lý bộ thẻ.....	35
Đặc tả use-case tạo thẻ từ đoạn văn bản (prompt).....	36
Đặc tả use-case tạo thẻ từ file tài liệu.....	38
Đặc tả use-case quản lý danh thẻ trong 1 bộ thẻ.....	40
Đặc tả use-case học flashcard.....	41
Đặc tả use-case quản lý thư mục.....	42
Đặc tả use -case xem dashboard thống kê.....	43
Đặc tả use-case ôn tập chế độ quiz.....	45
Đặc tả use-case quản lý nhóm.....	47
Bảng Users.....	60
Bảng User_Setting.....	61
Bảng Roles.....	62
Bảng Permission.....	62
Bảng Device.....	63
Bảng Session.....	64
Bảng Device_Token.....	65
Bảng Card_Set.....	66
Bảng Card.....	67
Bảng Card Option.....	68
Bảng Study Progress.....	69
Bảng Learning Session.....	70
Bảng Review Logs.....	71
Bảng Notification.....	72
Bảng Notification Attempt.....	73
Bảng Files.....	74
Bảng kiểm thử chức năng đăng nhập bằng google.....	85
Bảng kiểm thử chức năng tạo thẻ bằng AI.....	85
Bảng kiểm thử chức năng học flashcard và thuật toán SRS.....	86

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong kỷ nguyên số và bùng nổ thông tin hiện nay, việc tiếp nhận và ghi nhớ kiến thức mới một cách hiệu quả là thách thức lớn đối với người học. Các phương pháp học tập truyền thống như đọc lại nhiều lần (re-reading) hay học nhồi nhét (cramming) thường tốn nhiều thời gian nhưng hiệu quả ghi nhớ dài hạn không cao. Nghiên cứu về "Đường cong quên lãng" (Forgetting Curve) của Hermann Ebbinghaus đã chỉ ra rằng não bộ con người có xu hướng quên đi flashcard phần lớn thông tin vừa học chỉ sau vài ngày nếu không có biện pháp ôn tập phù hợp.

Để giải quyết vấn đề này, phương pháp Lặp lại ngắt quãng (Spaced Repetition System - SRS) sử dụng thẻ ghi nhớ (Flashcard) đã được chứng minh là một trong những kỹ thuật học tập tối ưu nhất. Hiện nay, trên thị trường đã xuất hiện nhiều ứng dụng hỗ trợ phương pháp này như Anki, Quizlet, SuperMemo.

Tuy nhiên, qua khảo sát thực tế, các ứng dụng hiện có vẫn tồn tại một số hạn chế:

- **Anki:** Sở hữu thuật toán mạnh mẽ nhưng giao diện người dùng (UI/UX) cũ kỹ, phức tạp, khó tiếp cận với người dùng phổ thông. Việc tạo thẻ thủ công tốn rất nhiều thời gian.
- **Quizlet:** Giao diện thân thiện nhưng các tính năng học sâu (SRS) thường bị giới hạn ở bản trả phí, và thiếu khả năng tùy biến thuật toán cá nhân hóa.
- **Vấn đề chung:** Người học mất quá nhiều thời gian cho việc **tạo** tài liệu học tập thay vì **học** chúng.

Xuất phát từ thực tế trên, việc xây dựng một hệ thống mới kết hợp được sức mạnh thuật toán SRS chuẩn xác, giao diện hiện đại và đặc biệt là ứng dụng Trí tuệ nhân tạo (Generative AI) để tự động hóa quy trình tạo thẻ là vô cùng cần thiết.

Vì những lý do đó, em quyết định lựa chọn đề tài: “**Xây dựng hệ thống hỗ trợ học flashcard**” để nghiên cứu và thực hiện đồ án tốt nghiệp.

2. Mục tiêu và nhiệm vụ đề tài

Xây dựng một ứng dụng Website hoàn chỉnh cho phép người dùng tạo, quản lý và ôn tập các bộ thẻ ghi nhớ (Flashcard) dựa trên thuật toán Lặp lại ngắt quãng, tích hợp AI để hỗ trợ tạo nội dung học tập nhanh chóng.

- Về mặt kỹ thuật:
 - Triển khai thành công thuật toán Lặp lại ngắt quãng (dựa trên SM-2 hoặc các biến thể cải tiến) để tính toán lịch ôn tập cá nhân hóa.
 - Tích hợp API của các mô hình ngôn ngữ lớn (LLM như OpenAI, Gemini) để xây dựng tính năng "Tạo thẻ tự động từ văn bản/tài liệu".
 - Thiết kế Cơ sở dữ liệu tối ưu cho việc lưu trữ tiến độ học tập và lịch sử ôn tập (Review logs).
- Về mặt ứng dụng:
 - Xây dựng giao diện người dùng trực quan, dễ sử dụng, hỗ trợ thống kê tiến độ (Heatmap, biểu đồ ghi nhớ).
 - Cung cấp các chế độ học đa dạng: Lật thẻ cơ bản, Trắc nghiệm (Quiz).
 - Hỗ trợ quản lý bộ thẻ (Card set management) và chia sẻ bộ thẻ giữa các người dùng.

3. Phạm vi nghiên cứu

- **Phạm vi chức năng**

Hệ thống tập trung vào các chức năng cốt lõi sau:

- **Quản lý tài khoản:** Đăng ký, đăng nhập, lưu trữ dữ liệu cá nhân.
- **Quản lý Bộ thẻ (Card Set) & Thẻ (Card):** Thêm, sửa, xóa, tìm kiếm, import/export.
- **Hệ thống AI:** Nhập đoạn văn bản/từ vựng -> AI trả về danh sách thẻ (Mặt trước/Mặt sau) và câu hỏi trắc nghiệm.
- **Hệ thống ôn tập (Study System):** Hiện thị thẻ theo lịch trình SRS, xử lý đánh giá (Easy, Good, Hard, Again).
- **Thông kê:** Biểu đồ độ nhớ, số lượng thẻ đã học, chuỗi ngày học liên tục (Streak).

- **Phạm vi công nghệ**

- **Backend:** NestJS - Triển khai logic nghiệp vụ và API
- **Frontend:** Vuejs - Xây dựng giao diện người dùng
- **Database:** PostgreSQL - Lưu trữ dữ liệu flashcard và lịch sử học.
- **AI Integration:** Sử dụng API của [OpenAI GPT / Google Gemini]
- **Deployment:** Docker, VPS
- **CI/CD:** Github Actions / Jenkins
- **Các kỹ thuật bổ trợ:**
 - Caching, Logging, Error Handling
 - Authentication & Authorization

4. Đối tượng và phương pháp nghiên cứu

- **Đối tượng nghiên cứu**

- Các kỹ thuật Prompt Engineering để tối ưu hóa kết quả trả về từ AI.
- Hành vi và trải nghiệm người dùng trong các ứng dụng giáo dục (EdTech).
- Các thuật toán Lặp lại ngắt quãng (Spaced Repetition Algorithms): SM-2, FSRS, Leitner System.

- **Phương pháp nghiên cứu**

- **Phương pháp nghiên cứu lý thuyết:** Tìm hiểu tài liệu về cơ chế ghi nhớ của não bộ, cấu trúc dữ liệu và giải thuật SRS, kiến trúc phần mềm monolithic.
- **Phương pháp phân tích và thiết kế hệ thống:** Sử dụng ngôn ngữ mô hình hóa UML để thiết kế cơ sở dữ liệu, sơ đồ hoạt động, sơ đồ tuần tự.

- **Phương pháp thực nghiệm:** Xây dựng ứng dụng, chạy thử nghiệm với tập dữ liệu mẫu, đánh giá hiệu quả thuật toán và lấy ý kiến người dùng để tinh chỉnh.

5. Ý nghĩa thực tiễn

Đề tài không chỉ dừng lại ở mức độ nghiên cứu lý thuyết mà hướng tới tạo ra một sản phẩm có giá trị sử dụng thực tế:

- Giúp người học (đặc biệt là học sinh, sinh viên, người học ngoại ngữ) tiết kiệm đến 70% thời gian tạo tài liệu nhờ AI.
- Tăng hiệu quả ghi nhớ kiến thức, giảm thiểu tình trạng học trước quên sau.
- Cung cấp một công cụ mã nguồn mở hoặc nền tảng miễn phí thay thế cho các phần mềm trả phí đắt đỏ.

6. Bố cục đề án

Báo cáo đề án được chia thành 5 chương chính:

- **Mở đầu:** Trình bày lý do, mục tiêu, phạm vi và phương pháp nghiên cứu.
- **Chương 1: Cơ sở lý thuyết:** Giới thiệu về thuật toán SRS (SM-2), công nghệ AI và các công cụ phát triển.
- **Chương 2: Phân tích và thiết kế hệ thống:** Mô tả các yêu cầu chức năng, phi chức năng, thiết kế cơ sở dữ liệu và kiến trúc hệ thống.
- **Chương 3: Xây dựng và triển khai hệ thống:** Trình bày quá trình xây dựng ứng dụng, giao diện các chức năng chính và kết quả kiểm thử.
- **Chương 4: Kết luận:** Tổng kết kết quả đạt được, hạn chế và đề xuất hướng nâng cấp trong tương lai.
- **Tài liệu tham khảo**

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Tổng quan về phương pháp học tập Flashcard

1.1.1 Khái niệm flashcard và E-learning

a) Khái niệm Flashcard

Flashcard (thẻ ghi nhớ) là một công cụ hỗ trợ học tập dựa trên nguyên tắc cô đọng thông tin. Một thẻ flashcard tiêu chuẩn bao gồm hai mặt:

- **Mặt trước (Front/Question):** Chứa từ khóa, câu hỏi, hình ảnh hoặc thuật ngữ cần ghi nhớ.
- **Mặt sau (Back/Answer):** Chứa định nghĩa, câu trả lời, cách phát âm hoặc ví dụ minh họa.

Cơ chế hoạt động cốt lõi của Flashcard dựa trên hiện tượng tâm lý học gọi là **Gợi nhớ chủ động (Active Recall)**. Thay vì đọc thụ động (passive review), một đoạn văn bản, người học buộc não bộ phải "truy xuất" thông tin từ ký ức khi nhìn thấy câu hỏi ở mặt trước. Quá trình nỗ lực truy xuất này giúp củng cố các kết nối thần kinh, làm cho trí nhớ trở nên bền vững hơn so với việc chỉ đọc lại tài liệu.

b) Flashcard trong môi trường E-learning

Trong bối cảnh E-learning (Giáo dục điện tử), Flashcard đã chuyển dịch từ dạng thẻ giấy vật lý sang dạng kỹ thuật số (Digital Flashcards). Sự chuyển đổi này mang lại nhiều ưu điểm vượt trội:

- **Đa phương tiện:** Tích hợp được âm thanh (phát âm), hình ảnh động, video, giúp tăng khả năng liên tưởng.
- **Tính di động:** Người học có thể mang theo hàng nghìn thẻ trong một thiết bị di động duy nhất.
- **Tự động hóa:** Quan trọng nhất, E-learning cho phép tích hợp các thuật toán thông minh để tự động sắp xếp lịch học, điều mà thẻ giấy thủ công rất khó thực hiện chính xác.

1.1.2 Thuật toán lặp lại ngắt quãng (Spaced Repetition System)

a) Cơ sở khoa học: Đường cong quên lãng (Forgetting Curve)

Thuật toán Lặp lại ngắt quãng (SRS) được xây dựng dựa trên nghiên cứu của nhà tâm lý học người Đức Hermann Ebbinghaus vào năm 1885. Ông phát hiện ra rằng não bộ con người có xu hướng quên thông tin theo hàm mũ theo thời gian nếu không có sự ôn tập.

- Sau 20 phút, chúng ta quên khoảng 42% lượng thông tin.
- Sau 24 giờ, lượng thông tin bị quên lên tới gần 70%.

Tuy nhiên, Ebbinghaus cũng chỉ ra rằng nếu thông tin được ôn tập lại đúng vào thời điểm "sắp quên", đường cong quên lãng sẽ được "làm phẳng" ra. Nghĩa là tốc độ quên sẽ chậm lại, và khoảng thời gian lưu giữ thông tin sẽ kéo dài hơn sau mỗi lần ôn tập thành công.

b) Nguyên lý hoạt động của SRS

SRS là hệ thống tính toán khoảng thời gian tối ưu giữa các lần ôn tập. Thay vì ôn tập hàng ngày (gây lãng phí thời gian) hoặc ôn tập quá thưa (gây quên kiến thức), SRS xác định điểm rơi lý tưởng để nhắc nhở người học.

Nguyên tắc chung:

- Nếu người học trả lời **đúng/dễ dàng**: Khoảng thời gian đến lần ôn tiếp theo sẽ được giãn rộng ra (ví dụ: 1 ngày → 3 ngày → 7 ngày...).
- Nếu người học trả lời **sai/khó khăn**: Khoảng thời gian sẽ bị rút ngắn lại hoặc reset về đầu để học lại ngay lập tức.

c) Các mô hình thuật toán phổ biến

- **Hệ thống hộp Leitner (Leitner System):** Đây là phương pháp thủ công, sử dụng các hộp vật lý đánh số từ 1 đến 5. Thẻ trả lời đúng sẽ được chuyển sang hộp cao hơn (ôn ít hơn), thẻ sai sẽ bị đưa về hộp 1 (ôn hàng ngày). Đây là tiền đề cho các thuật toán máy tính sau này.
- **Thuật toán SuperMemo-2 (SM-2):** Đây là thuật toán phổ biến nhất được sử dụng trong các phần mềm mã nguồn mở như Anki và là thuật toán nền tảng cho “Hệ thống hỗ trợ học flashcard” trong đề án này. SM-2 quản lý mỗi thẻ dựa trên hai chỉ số chính:

1. **Interval (I):** Số ngày chờ đợi cho đến lần ôn tiếp theo
2. **Ease factor (EF):** Hệ số dễ nhớ (mặc định là 2.5) biểu thị mức độ dễ của thẻ đối với người học. Thẻ càng dễ, EF càng cao, khoảng thời gian giãn cách càng tăng nhanh. EF không bao giờ giảm dưới 1.3 để tránh việc thẻ bị lặp lại quá dày đặc vô ích.
3. Công thức cập nhật chỉ số EF:

$$EF_{mới} = EF_{cũ} + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02))$$

Hình 1. Công thức cập nhật hệ số dễ nhớ

4. Công thức cập nhật In:

$$I_n = \begin{cases} 1 & \text{nếu } n = 1 \\ 6 & \text{nếu } n = 2 \\ \lceil I_{n-1} \times EF \rceil & \text{nếu } n > 2 \end{cases}$$

Hình 2. Công thức tính khoảng cách ngày

Việc ứng dụng thuật toán này vào hệ thống giúp cá nhân hóa hoàn toàn lộ trình học tập, đảm bảo người dùng dành thời gian tập trung vào những kiến thức yếu nhất (thẻ có EF thấp), từ đó tối ưu hóa hiệu suất học tập.

1.2 Phân tích các nền tảng tạo Flashcard tương tự

Để xây dựng một hệ thống tối ưu, việc nghiên cứu các giải pháp hiện có trên thị trường là bước đi quan trọng nhằm kế thừa những ưu điểm và khắc phục những hạn chế còn tồn tại. Trong phạm vi đề án này, tôi tập trung phân tích hai nền tảng tiêu biểu nhất đại diện cho hai trường phái khác nhau: **Quizlet** (đại diện cho sự thân thiện, đại chúng) và **Anki** (đại diện cho tính chuyên sâu, tối ưu thuật toán).

1.2.1 Phân tích nền tảng Quizlet

a) Giới thiệu chung

Quizlet là nền tảng học tập trực tuyến phổ biến nhất thế giới hiện nay, được phát triển từ năm 2005. Quizlet tập trung vào trải nghiệm người dùng đơn giản, hỗ trợ đa nền tảng (Web, iOS, Android) và xây dựng một cộng đồng chia sẻ dữ liệu lớn.

b) Đặc điểm nổi bật

- Giao diện (UI/UX): Hiện đại, trực quan và dễ tiếp cận. Người dùng mới có thể tạo bộ thẻ và bắt đầu học ngay lập tức mà không cần đọc hướng dẫn.
- Chế độ học đa dạng: Ngoài lật thẻ truyền thống, Quizlet "Game hóa" (Gamification) việc học thông qua các chế độ như: *Match (Ghép thẻ)*, *Gravity (Trò chơi thiên thạch)* và *Learn (Học thông minh)*.
- Tính cộng đồng: Cho phép người dùng tìm kiếm và sử dụng lại (clone) hàng triệu bộ thẻ do người khác chia sẻ.

c) Hạn chế

- Thuật toán SRS hạn chế: Chế độ học Lặp lại ngắt quãng (trước đây là Quizlet Long-term Learning) hiện đã bị gỡ bỏ hoặc chuyển sang tính năng trả phí (Quizlet Plus) nhưng vẫn chưa thực sự mạnh mẽ và tùy biến sâu như Anki. Nó thiên về việc học thuộc lòng ngắn hạn (Short-term cramming) hơn là ghi nhớ dài hạn.
- Phụ thuộc vào Internet: Phiên bản miễn phí hạn chế khả năng học ngoại tuyến (Offline).
- Khả năng tùy biến thẻ thấp: Chỉ hỗ trợ văn bản và hình ảnh cơ bản, khó khăn khi cần nhập các công thức toán học phức tạp (LaTeX) hay tùy chỉnh CSS cho thẻ.

1.2.2 Phân tích phần mềm Anki

a) Giới thiệu chung

Anki là phần mềm mã nguồn mở (Open-source) ra đời năm 2006, được xem là "tiêu chuẩn vàng" trong cộng đồng những người học ngoại ngữ và sinh viên y khoa. Anki sử dụng biến thể của thuật toán SM-2 để tối ưu hóa việc ghi nhớ.

b) Đặc điểm nổi bật

- Thuật toán mạnh mẽ: Cho phép người dùng can thiệp sâu vào các thông số của thuật toán (Interval modifier, Ease factor, Leech threshold...) để phù hợp với từng cá nhân.
- Hệ sinh thái Add-on (Tiện ích mở rộng): Cộng đồng lập trình viên đóng góp hàng ngàn plugin giúp mở rộng tính năng (ví dụ: tự động tra từ điển, chuyển văn bản thành giọng nói, Heatmap theo dõi tiến độ).
- Hỗ trợ đa phương tiện mạnh mẽ: Hỗ trợ tốt LaTeX, HTML/CSS, Audio, Video nhúng trực tiếp vào thẻ

c) Hạn chế

- Giao diện (UI) lạc hậu: Giao diện của Anki trên máy tính (Desktop) khá cũ kỹ, khô khan và thiếu tính thẩm mỹ, tạo cảm giác "ngợp" cho người mới bắt đầu.
- Đường cong học tập (Learning Curve) cao: Người dùng mất nhiều thời gian để hiểu cách thiết lập (Settings) và cách tạo thẻ hiệu quả.
- Khó khăn trong việc tạo thẻ: Việc tạo thẻ hoàn toàn thủ công. Nếu muốn tạo 100 từ vựng, người dùng phải nhập liệu 100 lần hoặc phải biết thao tác import file CSV phức tạp.

1.2.2 Đánh giá ưu nhược điểm và bài học kinh nghiệm

Dưới đây là bảng so sánh tóm tắt giữa hai nền tảng và hướng đi của đề tài:

Bảng 1. Bảng so sánh Quizlet & Anki

Tiêu chí	Quizlet	Anki	Flashcard Maker (Đề tài)
Giao diện (UI/UX)	Rất tốt, hiện đại, dễ dùng	Khô khan, khó dùng	Hướng tới sự tối giản, hiện đại như Quizlet
Thuật toán SRS	Cơ bản, ít tùy biến	Rất mạnh, tùy biến sâu	Kế thừa thuật toán mạnh mẽ của Anki (SM-2)
Tạo nội dung	Thủ công hoặc tìm từ cộng đồng	Thủ công, tốn thời gian	Tự động hóa bằng Generative AI (Điểm đột phá)
Chi phí	Freemium (Miễn phí hạn chế)	Miễn phí (trừ bản iOS)	Miễn phí / Mã nguồn mở

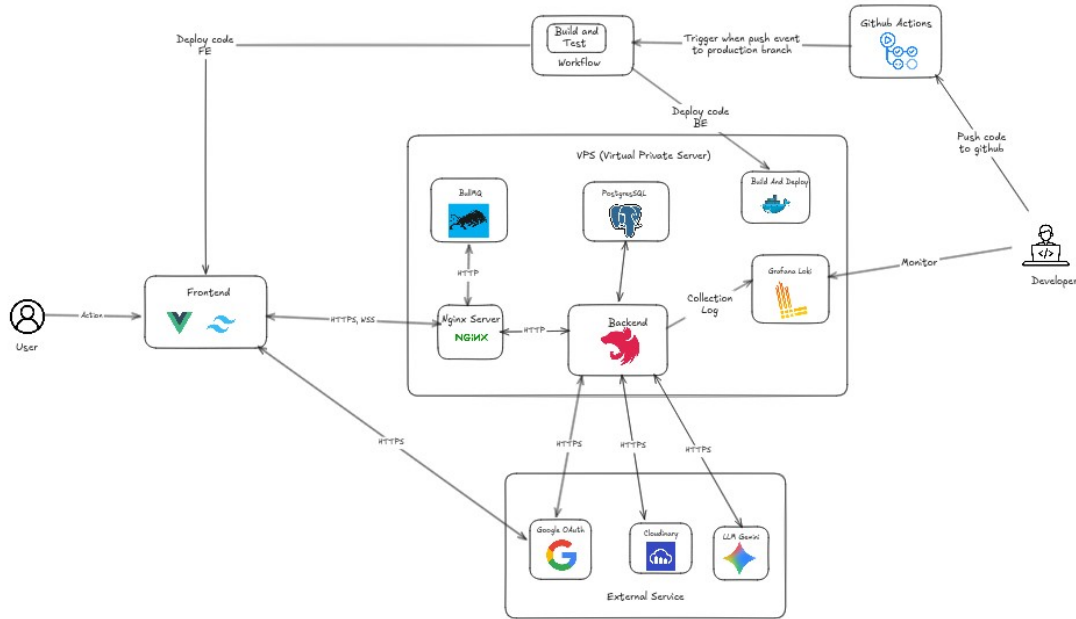
Bài học kinh nghiệm và Hướng phát triển cho đề tài:

Từ việc phân tích hai nền tảng trên, nhóm thực hiện rút ra các định hướng then chốt cho “**Hệ thống hỗ trợ học flashcard**” :

- Về trải nghiệm người dùng (UX):** Hệ thống phải hạ thấp rào cản gia nhập. Người dùng không cần hiểu sâu về thuật toán vẫn có thể bấm nút "Học ngay" và để hệ thống tự xử lý tính toán ngầm (giống Quizlet).
- Về cốt lõi công nghệ (Core):** Không thỏa hiệp về chất lượng thuật toán. Phải triển khai chính xác logic của Spaced Repetition (giống Anki) để đảm bảo hiệu quả học tập thực sự.
- Về tính năng đột phá (Killer Feature):** Giải quyết "nỗi đau" lớn nhất của người dùng cả hai nền tảng trên là **thời gian tạo thẻ**. Bằng cách tích hợp AI (LLM), hệ thống cho phép người dùng nhập một đoạn văn bản (ví dụ: một bài báo, một đoạn tài liệu chuyên ngành) và tự động sinh ra bộ thẻ ghi nhớ chất lượng cao chỉ trong vài giây. Đây chính là lợi thế cạnh tranh chính của đề án so với các phần mềm truyền thống.

1.3 Các công nghệ phát triển hệ thống

Để hiện thực hóa các mục tiêu đã đề ra, đặc biệt là việc xử lý các thuật toán tính toán thời gian (SRS) và tích hợp AI, hệ thống yêu cầu một nền tảng Backend mạnh mẽ, có cấu trúc chặt chẽ và khả năng mở rộng tốt. Dưới đây là các công nghệ cốt lõi được lựa chọn.



Hình 1. Sơ đồ tổng quan kiến trúc hệ thống

1.3.1 Framework NestJS (Node.js)

a) Giới thiệu

NestJS là một framework mã nguồn mở dành cho việc xây dựng các ứng dụng phía máy chủ (server-side) hiệu quả và dễ bảo trì, được xây dựng dựa trên nền tảng Node.js. NestJS sử dụng ngôn ngữ TypeScript làm mặc định, kết hợp các nguyên lý của Lập trình hướng đối tượng (OOP), Lập trình hàm (FP) và Lập trình phản ứng (FRP).

b) Kiến trúc cốt lõi của NestJS

NestJS được thiết kế lấy cảm hứng từ Angular, với kiến trúc module hóa cao. Các thành phần chính bao gồm:

- **Modules:** Giúp tổ chức ứng dụng thành các khối chức năng riêng biệt (ví dụ: AuthModule, CardSetModule, LearningModule). Điều này giúp code gọn gàng và dễ quản lý khi hệ thống mở rộng.
- **Controllers:** Chịu trách nhiệm tiếp nhận các yêu cầu (Requests) từ client và trả về phản hồi (Responses).
- **Providers (Services):** Nơi chứa toàn bộ logic nghiệp vụ (Business Logic). Trong đồ án này, các thuật toán SRS (SM-2) và logic gọi AI sẽ được đóng gói trong các Service này.
- **Dependency Injection (DI):** Cơ chế "Tiêm phụ thuộc" giúp quản lý sự phụ thuộc giữa các thành phần một cách lỏng lẻo, dễ dàng cho việc kiểm thử (Unit Test) và bảo trì.

c) Lý do lựa chọn cho đồ án

- **Cấu trúc chặt chẽ:** Khác với Express.js (khá tự do và dễ dẫn đến "Spaghetti code"), NestJS buộc người lập trình tuân theo một kiến trúc chuẩn. Điều này rất quan trọng để quản lý logic phức tạp của thuật toán Lập lại ngắt quãng.
- **Hỗ trợ TypeScript:** Việc định kiểu dữ liệu tĩnh (Static typing) giúp giảm thiểu lỗi trong quá trình tính toán các chỉ số Interval, Ease Factor và các cấu trúc dữ liệu JSON trả về từ AI.
- **Hệ sinh thái phong phú:** Hỗ trợ tốt các thư viện ORM (TypeORM, Prisma) để làm việc với Database và tích hợp sẵn Swagger để tạo tài liệu API tự động.

1.3.2 Kiến trúc hệ thống Backend

Hệ thống được thiết kế theo mô hình **Kiến trúc phân lớp (Layered Architecture)**, đảm bảo sự tách biệt rõ ràng giữa các trách nhiệm.

Các tầng xử lý dữ liệu bao gồm:

1. **Presentation Layer (Controller):**
 - Tiếp nhận HTTP Requests từ người dùng (ví dụ: POST /reviews).
 - Xác thực dữ liệu đầu vào (Validation) thông qua DTO (Data Transfer Object).
 - Gọi xuống tầng Service tương ứng.
2. **Business Logic Layer (Service):**
 - Đây là tầng quan trọng nhất của hệ thống.
 - **SRS Service:** Chứa hàm cài đặt thuật toán SM-2. Khi nhận được đánh giá từ người dùng, Service này sẽ lấy dữ liệu lịch sử, tính toán ngày ôn tập tiếp theo và cập nhật trạng thái thẻ.
 - **AI Service:** Chịu trách nhiệm giao tiếp với LLM (Large Language Model), xây dựng Prompt, gửi đi và xử lý kết quả JSON trả về để lưu vào Database.
3. **Data Access Layer (Repository/ORM):**
 - Tương tác trực tiếp với Cơ sở dữ liệu.
 - Sử dụng TypeORM (hoặc Prisma) để thực hiện các truy vấn CRUD mà không cần viết câu lệnh SQL thủ công, giúp code an toàn và dễ đọc hơn.

1.3.3 Hệ quản trị cơ sở dữ liệu PostgreSQL

a) Giới thiệu

PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở mạnh mẽ, nổi tiếng về độ tin cậy và tuân thủ các chuẩn SQL.

b) Lý do lựa chọn

- Tính toàn vẹn dữ liệu (ACID): Đảm bảo các giao dịch (Transactions) liên quan đến tiến độ học tập của người dùng được thực hiện chính xác tuyệt đối. Không xảy ra trường hợp lưu log review nhưng không cập nhật ngày ôn tập.
- Hỗ trợ kiểu dữ liệu JSONB: Đây là yếu tố then chốt. Hệ thống cần lưu trữ cấu trúc phức tạp của các loại thẻ (Trắc nghiệm, Điền từ) và hàng đợi phiên học (session_queue) một cách linh hoạt mà không cần tạo quá nhiều bảng phụ. PostgreSQL xử lý truy vấn trên JSONB nhanh hơn nhiều so với MySQL.

1.3.4 Công nghệ tích hợp AI (Generative AI)

Để thực hiện chức năng "Tạo thẻ tự động", hệ thống sử dụng API của các mô hình ngôn ngữ lớn (LLM).

- **Mô hình sử dụng:** Groq API hoặc Google Gemini
- **Cơ chế hoạt động:**
 1. Backend nhận văn bản đầu vào từ người dùng.
 2. Hệ thống xây dựng một **System Prompt** chuyên biệt, yêu cầu AI đóng vai trò là một "Giáo viên soạn bài", trích xuất các cặp câu hỏi - câu trả lời (Q&A).
 3. Yêu cầu định dạng đầu ra bắt buộc là **JSON** để hệ thống có thể tự động bóc tách và lưu vào Database mà không cần người dùng chỉnh sửa thủ công.

1.3.5 Quản lý hàng đợi tác vụ với BullMQ và Redis

a) Giới thiệu

BullMQ là một thư viện Node.js mạnh mẽ, dựa trên Redis, dùng để triển khai hàng đợi tin nhắn (Message Queue). Nó cho phép xử lý các tác vụ bất đồng bộ (Asynchronous tasks) và phân tán, giúp giảm tải cho luồng xử lý chính của ứng dụng.

b) Lý do lựa chọn cho hệ thống Flashcard

Trong ứng dụng Flashcard Maker, chức năng "Tạo thẻ tự động bằng AI" là một tác vụ nặng và tốn thời gian. Việc gọi API đến các mô hình ngôn ngữ lớn (LLM) và chờ phản hồi có thể mất từ 10 đến 30 giây.

- **Vấn đề:** Nếu xử lý đồng bộ (Synchronous) theo cách thông thường, người dùng sẽ phải treo trình duyệt chờ đợi, hoặc tệ hơn là gặp lỗi Timeout (Hết thời gian chờ) từ máy chủ.
- **Giải pháp với BullMQ:**
 1. Khi người dùng nhấn "Tạo thẻ", hệ thống chỉ ghi nhận yêu cầu và trả về thông báo "Đang xử lý" ngay lập tức (Non-blocking).
 2. Yêu cầu được đẩy vào hàng đợi của BullMQ.
 3. Một **Worker** chạy nền sẽ lấy yêu cầu ra, gọi AI API, xử lý dữ liệu và lưu vào Database.
 4. Khi hoàn tất, hệ thống gửi thông báo (Notification) hoặc cập nhật trạng thái qua WebSocket (Real-time) cho người dùng

Ngoài ra, BullMQ còn hỗ trợ cơ chế **Retry (Thử lại)** tự động. Nếu kết nối đến AI bị lỗi, hệ thống sẽ tự động thử lại sau một khoảng thời gian mà không làm mất yêu cầu của người dùng.

1.3.6 Công nghệ Container hóa với Docker

a) Giới thiệu

Docker là nền tảng mã nguồn mở cho phép đóng gói ứng dụng và toàn bộ các thư viện phụ thuộc (dependencies) vào một đơn vị chuẩn hóa gọi là Container. Điều này đảm bảo ứng dụng chạy nhất quán trên mọi môi trường, từ máy cá nhân của lập trình viên (Local) đến máy chủ thực tế (Production).

b) Ứng dụng trong đề tài

Hệ thống Flashcard Maker bao gồm nhiều thành phần dịch vụ khác nhau cần khởi chạy đồng thời:

- Backend API (NestJS).
- Database (PostgreSQL).
- Cache & Message Broker (Redis).
- Web Server (Nginx).

Việc cài đặt thủ công từng dịch vụ trên máy chủ rất phức tạp và dễ xung đột phiên bản. Docker và **Docker Compose** giải quyết vấn đề này bằng cách định nghĩa toàn bộ hạ tầng trong một file cấu hình (docker-compose.yml). Chỉ với một câu lệnh, toàn bộ hệ thống có thể được khởi tạo và kết nối với nhau thông qua mạng nội bộ ảo (Docker Network), giúp việc triển khai (Deployment) trở nên nhanh chóng và dễ dàng mở rộng (Scale).

1.3.7 Web Server và Reverse Proxy với Nginx

a) Giới thiệu

Nginx là một Web Server hiệu năng cao, mã nguồn mở, nổi tiếng với khả năng chịu tải lớn và tiêu thụ ít tài nguyên. Trong các kiến trúc ứng dụng hiện đại, Nginx thường đóng vai trò là Reverse Proxy (Máy chủ ủy quyền ngược).

b) Vai trò trong kiến trúc hệ thống

Trong đồ án này, Nginx không xử lý logic nghiệp vụ mà đứng trước hệ thống Backend để thực hiện các nhiệm vụ quan trọng:

- **Reverse Proxy:** Tiếp nhận mọi yêu cầu HTTP/HTTPS từ Internet (cổng 80/443) và điều hướng (forward) chúng đến đúng container Backend (đang chạy cổng nội bộ, ví dụ 3000). Điều này giúp che giấu cấu trúc mạng nội bộ, tăng tính bảo mật.

- **Serving Static Files:** Phục vụ trực tiếp các file tĩnh của phần Frontend (HTML, CSS, JS, hình ảnh) với tốc độ cực nhanh, giảm tải cho Backend NestJS để tập trung xử lý API.
- **Load Balancing (Cân bằng tải - Tùy chọn mở rộng):** Nếu lượng người dùng tăng cao, Nginx có thể phân phối lưu lượng truy cập đều giữa nhiều bản sao (Replicas) của Backend service

1.3.8 Framework phát triển giao diện Vue.js

a) Giới thiệu

Vue.js là một framework JavaScript mã nguồn mở dùng để xây dựng giao diện người dùng (User Interfaces - UI) và các ứng dụng đơn trang (Single-Page Applications - SPA). Được Evan You ra mắt vào năm 2014, Vue.js được mệnh danh là "The Progressive Framework" (Framework tiệm tiến), nghĩa là nó có khả năng thích ứng linh hoạt: có thể dùng để nhúng vào một phần nhỏ của trang web hoặc mở rộng để xây dựng một hệ thống Frontend phức tạp.

b) Các đặc điểm kỹ thuật nổi bật

- **Virtual DOM (DOM ảo):** Tương tự như React, Vue sử dụng Virtual DOM để tối ưu hóa hiệu suất. Thay vì thao tác trực tiếp và nặng nề lên DOM thật của trình duyệt mỗi khi dữ liệu thay đổi, Vue tính toán sự thay đổi trên bộ nhớ ảo và chỉ cập nhật những phần tử thực sự cần thiết, giúp giao diện phản hồi cực nhanh.
- **Reactive Data Binding (Liên kết dữ liệu phản ứng):** Vue sử dụng cơ chế liên kết dữ liệu hai chiều (Two-way binding) thông qua hệ thống phản ứng (Reactivity system). Khi trạng thái (state) của dữ liệu thay đổi, giao diện sẽ tự động cập nhật và ngược lại, giảm thiểu đáng kể lượng code xử lý sự kiện thủ công.
- **Component-based Architecture (Kiến trúc hướng thành phần):** Vue khuyến khích chia nhỏ giao diện thành các thành phần (Component) độc lập, có thể tái sử dụng (ví dụ: Component Flashcard, Component ProgressBar, Component Sidebar).

c) Lý do lựa chọn cho hệ thống hỗ trợ học flashcard

Đối với một ứng dụng đặc thù như Flash Study, trải nghiệm người dùng (UX) là yếu tố sống còn. Vue.js được lựa chọn vì những lý do sau:

1. **Trải nghiệm Single Page Application (SPA):** Việc học tập đòi hỏi sự tập trung cao độ. Mô hình SPA của Vue giúp ứng dụng chuyển đổi giữa các màn hình (ví dụ: từ "Danh sách bộ thẻ" sang "Màn hình ôn tập") mượt mà tức thì mà không cần tải lại toàn bộ trang (reload page). Điều này tạo cảm giác liền mạch giống như đang sử dụng một ứng dụng Native trên điện thoại (App-like experience).
2. **Tương tác thời gian thực mượt mà:** Trong quá trình ôn tập (Review Session), các thao tác như lật thẻ, đánh giá (Easy/Hard), chuyển thẻ tiếp theo diễn ra liên tục. Cơ chế Reactivity của Vue đảm bảo các tương tác này diễn ra với độ trễ thấp nhất, không gây giật lag, giúp duy trì dòng chảy học tập (Flow) của người dùng.
3. **Quản lý trạng thái phức tạp với Pinia/Vuex:** Hệ thống cần quản lý nhiều trạng thái phức tạp ở phía Client như: *Hàng đợi các thẻ đang học, Tiến độ phiên học, Thông tin người dùng đăng nhập*. Hệ sinh thái của Vue cung cấp thư viện quản lý trạng thái (State Management) như Pinia, giúp quản lý dữ liệu tập trung và đồng bộ hóa dễ dàng giữa các Component khác nhau.
4. **Dễ dàng tích hợp với Backend NestJS:** Vue.js hoạt động độc lập với Backend, giao tiếp hoàn toàn thông qua API (RESTful hoặc GraphQL). Điều này phù hợp hoàn hảo với kiến trúc tách biệt (Decoupled Architecture) mà đề án đang hướng tới, cho phép phát triển và nâng cấp Frontend/Backend độc lập với nhau.

1.3.9 Cơ chế xác thực người dùng Google OAuth 2.0

a) Giới thiệu về OAuth 2.0

OAuth 2.0 (Open Authorization) là một tiêu chuẩn mở về ủy quyền truy cập. Nó cho phép người dùng cấp quyền cho ứng dụng bên thứ ba (Flashcard Maker) truy cập vào thông tin cá nhân của họ trên một dịch vụ khác (Google) mà không cần phải tiết lộ mật khẩu.

b) Quy trình xác thực “Đăng nhập bằng google”

Quy trình tích hợp trong hệ thống diễn ra như sau:

1. **Yêu cầu:** Người dùng chọn "Login with Google" trên giao diện Vue.js.
2. **Chuyển hướng:** Ứng dụng chuyển hướng người dùng đến trang đăng nhập của Google.
3. **Xác thực & Cấp quyền:** Người dùng đăng nhập tài khoản Google và đồng ý chia sẻ thông tin cơ bản (Email, Tên, Ảnh đại diện).
4. **Mã ủy quyền:** Google trả về một mã xác thực (Auth Code) cho Backend NestJS.
5. **Trao đổi Token:** Backend sử dụng mã này để đổi lấy Access Token và thông tin người dùng từ máy chủ Google.
6. **Tạo phiên làm việc:** Hệ thống tạo tài khoản (nếu chưa có) và cấp phát JWT (JSON Web Token) riêng của hệ thống để duy trì phiên đăng nhập.

c) Ưu điểm và lý do lựa chọn

- **Nâng cao trải nghiệm người dùng (UX):** Loại bỏ rào cản tâm lý khi phải điền form đăng ký dài dòng và ghi nhớ thêm một mật khẩu mới. Người dùng có thể bắt đầu học ngay chỉ sau 2 cú click chuột.
- **Bảo mật:** Giảm thiểu rủi ro lộ lọt mật khẩu, vì hệ thống không trực tiếp lưu trữ mật khẩu của người dùng mà dựa vào cơ chế bảo mật cấp cao của Google.
- **Độ tin cậy:** Việc tích hợp đăng nhập bằng Google giúp ứng dụng mới tạo được sự tin tưởng bước đầu đối với người dùng.

1.3.10 Tự động hóa quy trình phát triển với Github Actions (CI/CD)

a) Giới thiệu về CI/CD và Github Actions

Trong quy trình phát triển phần mềm hiện đại, CI/CD (Continuous Integration - Tích hợp liên tục / Continuous Deployment - Triển khai liên tục) là một tiêu chuẩn không thể thiếu.

- **CI (Tích hợp liên tục):** Là quá trình tự động hóa việc hợp nhất code từ các nhà phát triển vào một kho lưu trữ chung. Mỗi lần hợp nhất đều được kiểm tra tự động (automated test) để phát hiện lỗi sớm.
- **CD (Triển khai liên tục):** Là quá trình tự động hóa việc phát hành phần mềm đã qua kiểm thử lên môi trường sản phẩm (Production) một cách nhanh chóng và an toàn.

Github Actions là một nền tảng CI/CD được tích hợp trực tiếp vào kho lưu trữ mã nguồn Github. Nó cho phép các nhà phát triển định nghĩa các quy trình làm việc (Workflows) tự động ngay trong repository mà không cần cài đặt server CI bên thứ ba (như Jenkins).

b) Cơ chế hoạt động

Github Actions hoạt động dựa trên các tệp cấu hình định dạng **YAML**.

Một quy trình (Workflow) cơ bản bao gồm:

1. **Events (Sự kiện):** Hành động kích hoạt quy trình (ví dụ: khi có lệnh git push lên nhánh main, hoặc khi có một Pull Request được tạo).
2. **Jobs (Công việc):** Tập hợp các bước thực hiện. Các Jobs có thể chạy song song hoặc tuần tự.
3. **Runners:** Là các máy chủ ảo (Linux, Windows, macOS) do Github cung cấp để thực thi các lệnh trong Job (như cài đặt Node.js, chạy test, build Docker).

c) **Ứng dụng trong hệ thống**

Việc áp dụng Github Actions vào đề án giúp giải quyết bài toán vận hành build version mới thủ công.

1. **Tự động hóa kiểm thử (Automated Testing):** Ngay khi sinh viên đẩy code (push code) lên Github, hệ thống sẽ tự động kích hoạt workflow để chạy các bài kiểm thử đơn vị (Unit Test) cho Backend NestJS. Nếu test thất bại, quy trình sẽ dừng lại và gửi cảnh báo, đảm bảo code lỗi không bao giờ được đưa vào hệ thống chính.
2. **Tự động hóa đóng gói (Automated Build & Dockerize):** Sau khi test thành công, Github Actions sẽ tự động thực hiện lệnh docker build để đóng gói ứng dụng thành các Docker Image và đẩy (push) lên Docker Hub (hoặc một Container Registry riêng).
3. **Tự động hóa triển khai (Automated Deployment):** Thông qua giao thức SSH, Github Actions sẽ kết nối an toàn đến máy chủ VPS, thực hiện lệnh docker pull để lấy phiên bản mới nhất về và khởi động lại các container. Quá trình này giúp việc cập nhật tính năng mới diễn ra chỉ trong vài phút mà không cần can thiệp thủ công, giảm thiểu sai sót do con người.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1 Phân tích yêu cầu chức năng

2.1.1 Nhóm chức năng Quản lý Tài khoản & Hệ thống

Đây là nhóm chức năng nền tảng đảm bảo tính bảo mật và cá nhân hóa cho người dùng.

- **Đăng ký/Đăng nhập:**
 - Cho phép người dùng tạo tài khoản mới bằng Email/Mật khẩu.
 - Tích hợp Social Login: Đăng nhập nhanh thông qua tài khoản Google (OAuth 2.0) để giảm thao tác nhập liệu.
- **Quản lý hồ sơ (Profile):** Cập nhật thông tin cá nhân (Họ tên, Ảnh đại diện - Avatar).
- **Cài đặt học tập (User Settings):**
 - Thiết lập mục tiêu số thẻ cần học mỗi ngày (Daily Target).
 - Cấu hình thời gian nhận thông báo nhắc nhở học tập.
 - Tùy chỉnh múi giờ (Timezone) để tính toán ngày qua đêm chính xác.

2.1.2 Nhóm chức năng Quản lý Nội dung (Content Management)

Cho phép người dùng xây dựng kho kiến thức cá nhân.

- **Quản lý Bộ thẻ (Card Set Management):**
 - Tạo mới, chỉnh sửa, xóa bộ thẻ.
 - Cấu hình quyền riêng tư cho bộ thẻ: Công khai (Public) hoặc Riêng tư (Private).
 - Sao chép (Clone) bộ thẻ công khai của người khác về kho cá nhân.
- **Quản lý Thẻ (Card CRUD):**
 - Thêm thẻ mới với hai mặt: Thuật ngữ (Front) và Định nghĩa (Back).
 - Hỗ trợ soạn thảo văn bản phong phú (Rich Text), chèn hình ảnh minh họa và âm thanh phát âm.
 - Hỗ trợ nhiều loại thẻ: Thẻ lật cơ bản, Thẻ trắc nghiệm (Multiple Choice).

- **Tạo thẻ tự động bằng AI (AI-Generated Content):**

- Người dùng nhập một đoạn văn bản, danh sách từ vựng hoặc tải lên một tệp tài liệu.
- Hệ thống sử dụng AI để phân tích và tự động trích xuất danh sách thẻ (Câu hỏi - Câu trả lời).

2.1.3 Nhóm chức năng Học tập & Ôn luyện (Study & SRS Engine)

Đây là trái tim của hệ thống, nơi áp dụng thuật toán SM-2/FSRS.

- **Phiên học (Learning Session):**

- Hệ thống tự động lọc và hiển thị các thẻ cần ôn tập trong ngày (Due cards) và các thẻ mới (New cards) theo giới hạn cài đặt.
- Hỗ trợ tính năng "Học tiếp" (Resume): Lưu trạng thái nếu người dùng thoát ứng dụng giữa chừng.

- **Đánh giá ghi nhớ (Rating):**

- Cung cấp 4 mức độ đánh giá sau khi lật thẻ: *Quên (Again)*, *Khó (Hard)*, *Được (Good)*, *Dễ (Easy)*.
- Dựa trên đánh giá, hệ thống tính toán lại thời gian lặp lại tiếp theo (Interval).

- **Thống kê tiến độ (Statistics):**

- Hiển thị biểu đồ nhiệt (Heatmap) thể hiện tần suất học tập trong năm.
- Thống kê số lượng thẻ thuộc các trạng thái: Mới, Đang học, Đã thuộc.
- Duy trì chuỗi ngày học liên tục (Streak) để khuyến khích thói quen.

2.2 Phân tích yêu cầu phi chức năng

2.2.1 Hiệu năng (Performance)

- **Thời gian phản hồi:** Các thao tác cơ bản (lật thẻ, chuyển trang) phải có độ trễ dưới 200ms để đảm bảo trải nghiệm học tập mượt mà ("Flow state").
- **Xử lý tác vụ nặng:** Tính năng tạo thẻ bằng AI (có thể mất 10-30 giây) phải được xử lý bất đồng bộ (Asynchronous) thông qua hàng đợi (Queue), không làm treo giao diện người dùng.
- **Khả năng chịu tải:** Hệ thống có thể hỗ trợ hàng nghìn người dùng truy cập đồng thời nhờ cơ chế Caching (Redis).

2.2.2 Tính Bảo mật (Security)

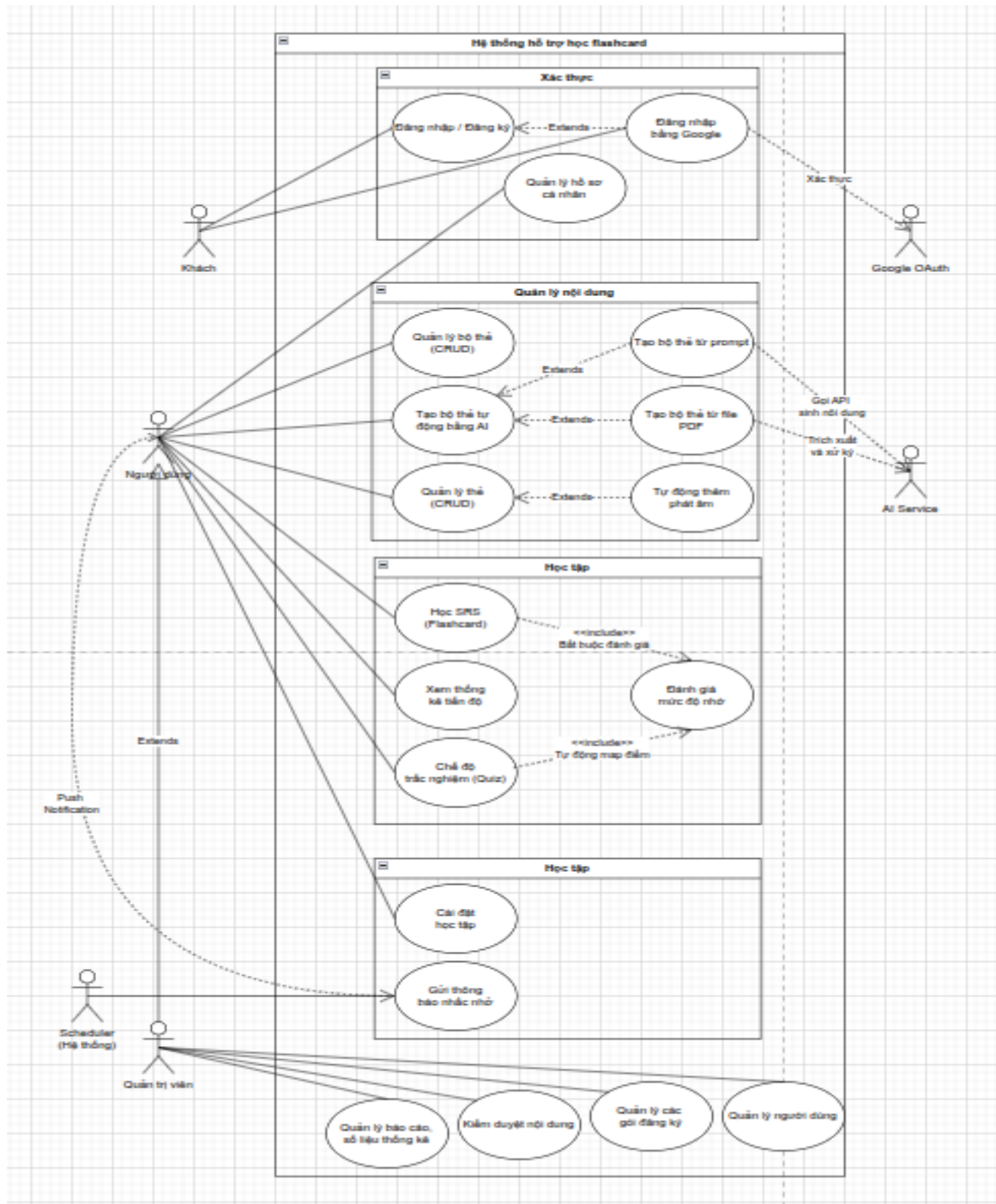
- **Mã hóa dữ liệu:** Mật khẩu người dùng phải được mã hóa một chiều (Hashing) bằng thuật toán mạnh (như Bcrypt hoặc Argon2) trước khi lưu vào cơ sở dữ liệu.
- **Xác thực & Phân quyền:**
 - Sử dụng JWT (JSON Web Token) để xác thực các phiên làm việc API.
 - Áp dụng mô hình RBAC (Role-Based Access Control) để đảm bảo người dùng chỉ có thể chỉnh sửa dữ liệu do chính mình tạo ra (trừ Admin).
- **Bảo vệ dữ liệu:** Cơ sở dữ liệu phải được sao lưu (Backup) định kỳ để tránh mất mát tiến độ học tập của người dùng.

2.2.3 Tính Khả dụng & Tương thích (Usability & Compatibility)

- **Giao diện:** Thiết kế theo phong cách tối giản, hiện đại, tập trung vào nội dung học tập (Content-first), giảm thiểu các yếu tố gây xao nhãng.
- **Đa nền tảng:** Ứng dụng Web phải có thiết kế đáp ứng (Responsive Design), hiển thị tốt trên cả máy tính (Desktop), máy tính bảng và điện thoại di động, hỗ trợ người dùng học mọi lúc mọi nơi.
- **Quốc tế hóa (i18n):** Hệ thống được thiết kế để dễ dàng hỗ trợ đa ngôn ngữ (Tiếng Việt, Tiếng Anh) trong tương lai.

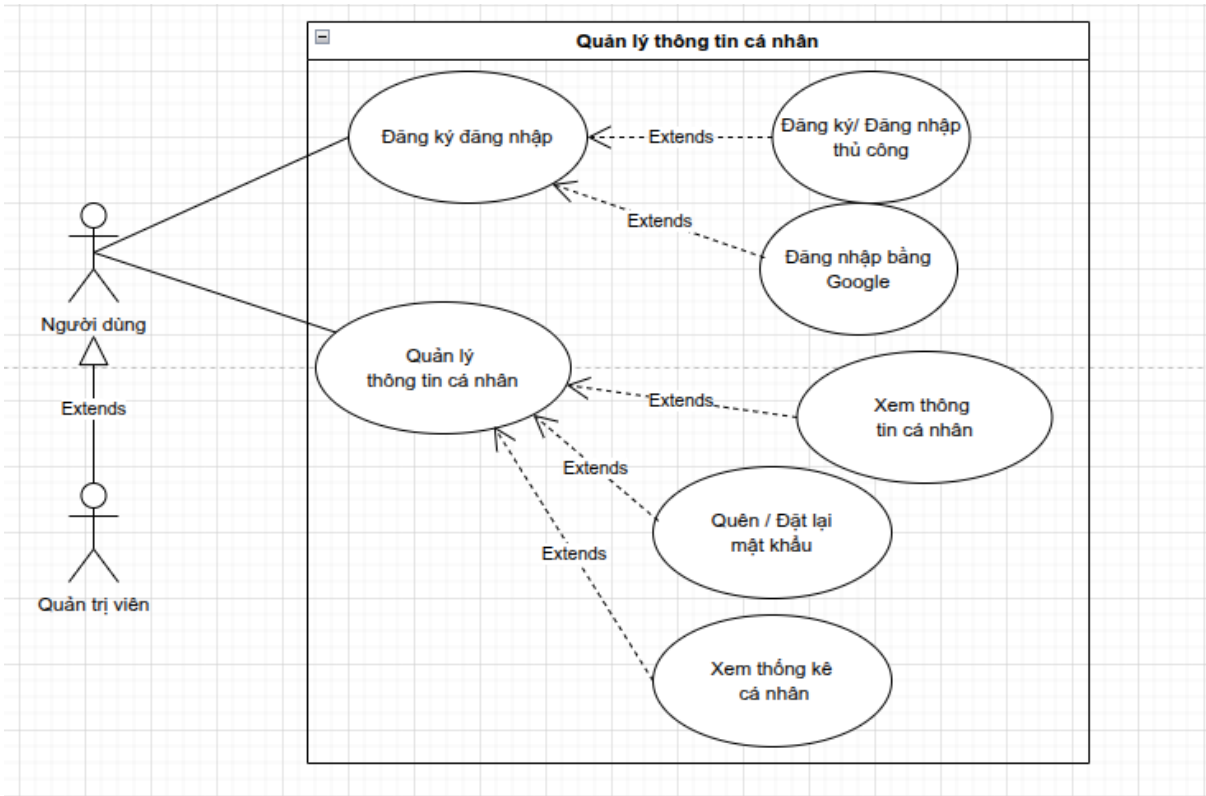
2.3 Biểu đồ use case

2.3.1 Biểu đồ use case tổng quan



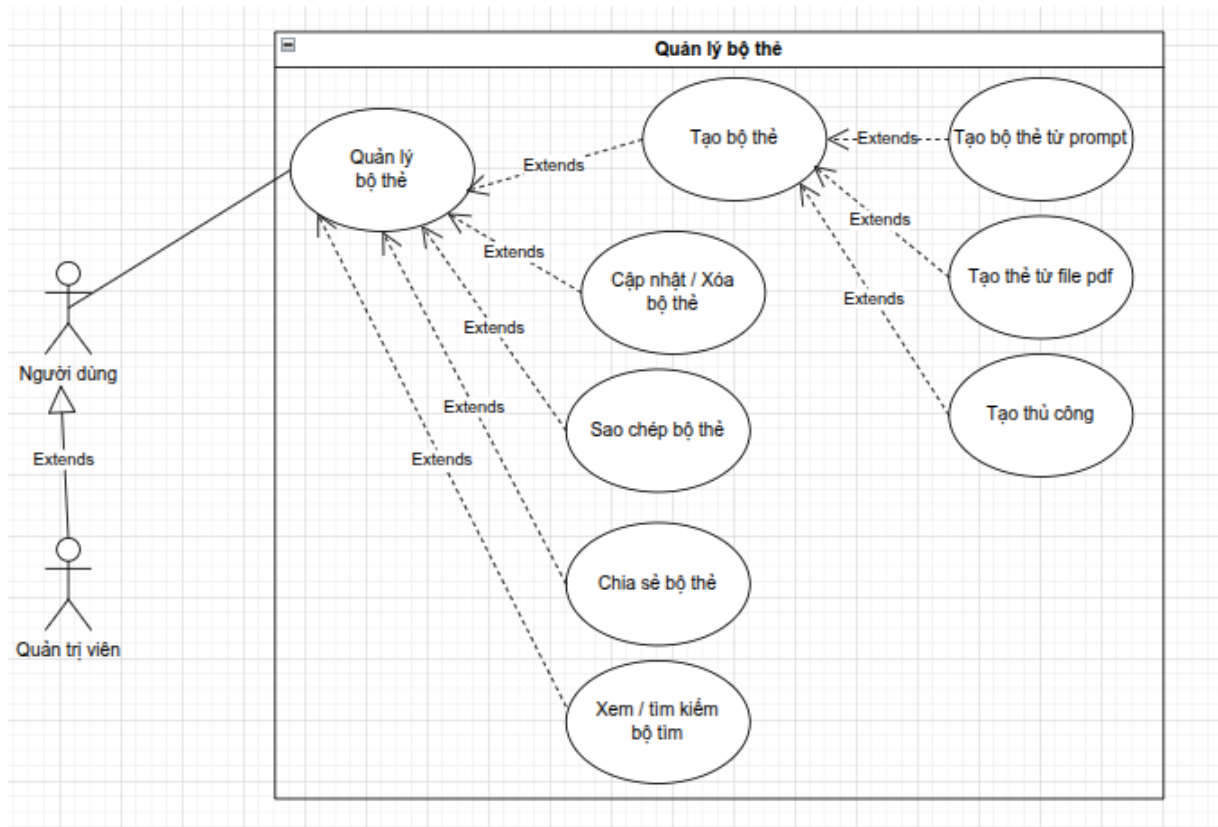
Hình 1. Biểu đồ use case tổng quan của hệ thống

2.3.2 Biểu đồ use case Quản lý thông tin cá nhân



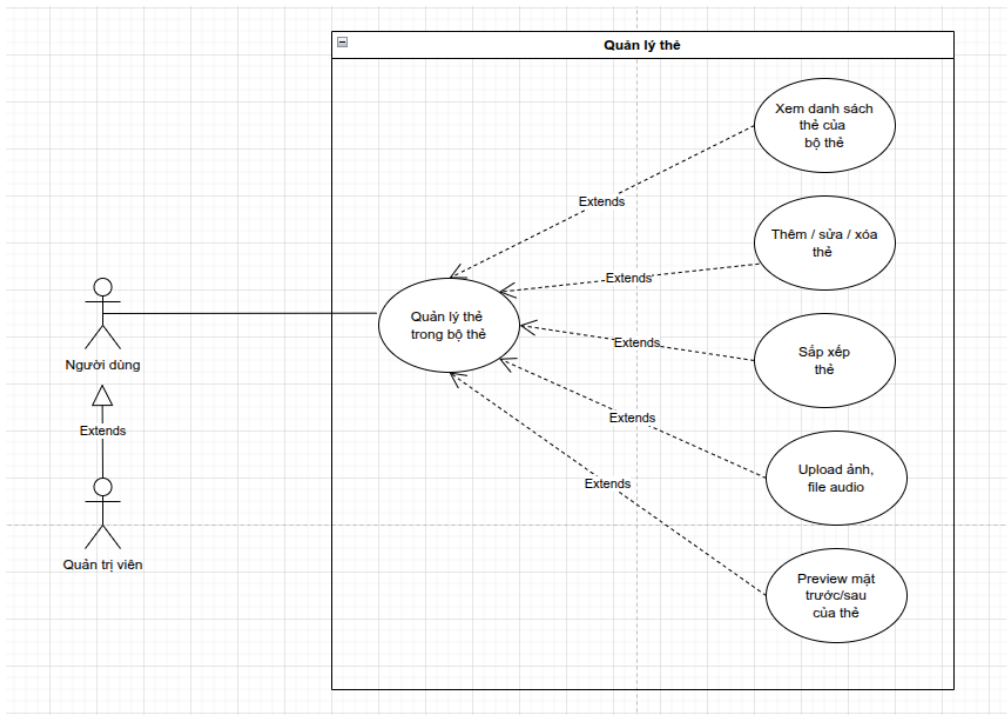
Hình 1. Biểu đồ use case Quản lý thông tin cá nhân

2.3.3 Biểu đồ use case Quản lý bộ thẻ



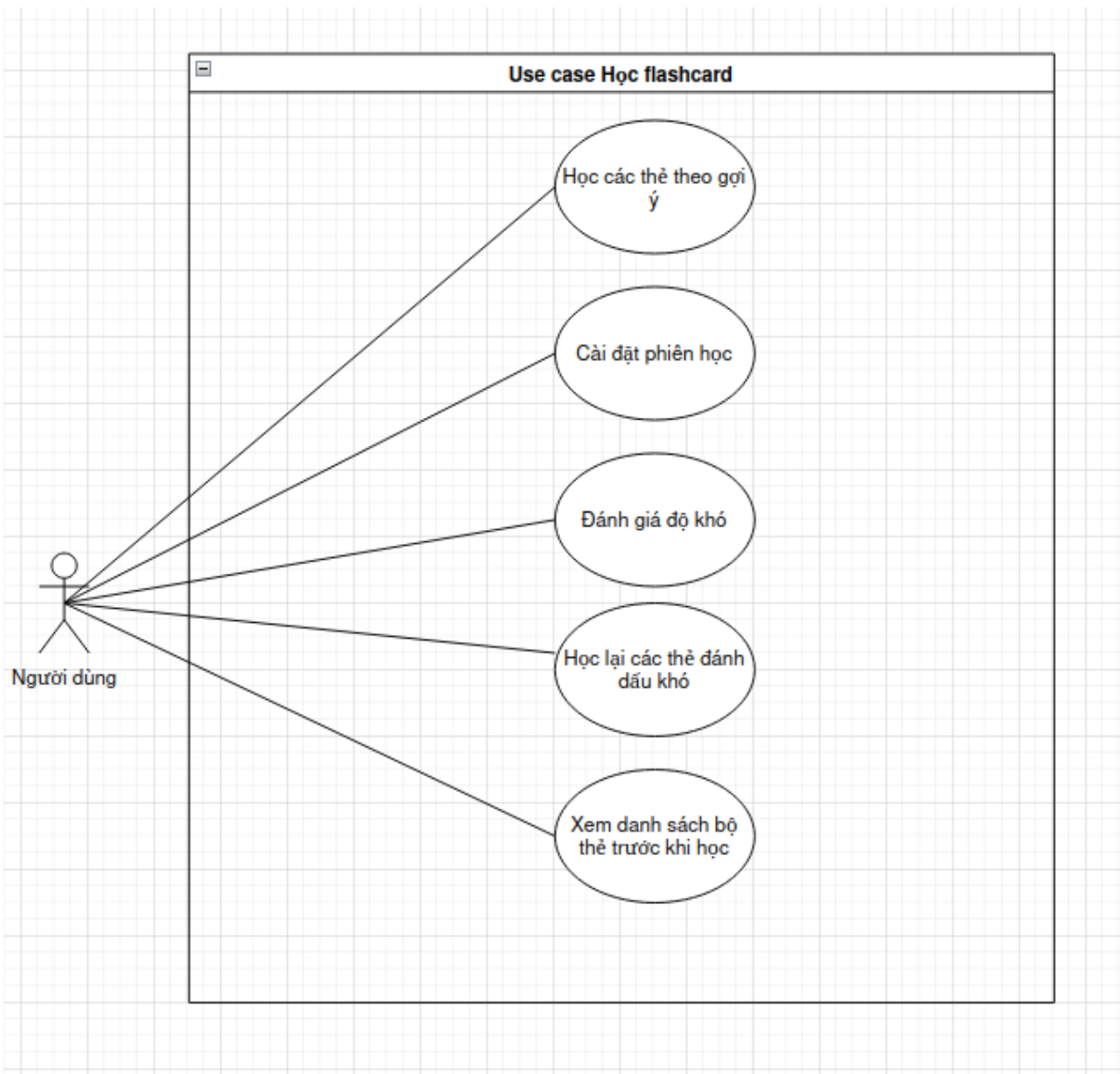
Hình 1. Biểu đồ use case quản lý bộ thẻ

2.3.4 Biểu đồ use case Quản lý thẻ



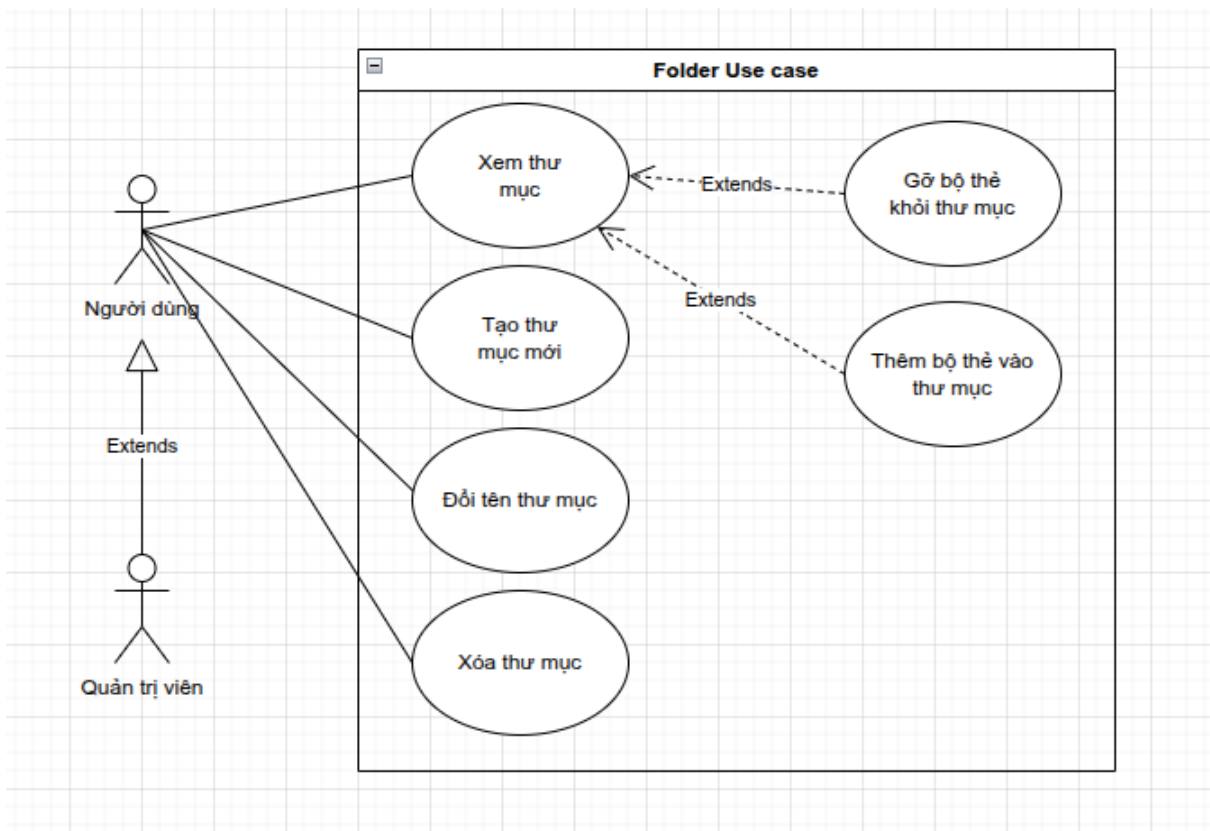
Hình 1. Biểu đồ use case quản lý thẻ

2.3.5 Biểu đồ use case Học Flashcard



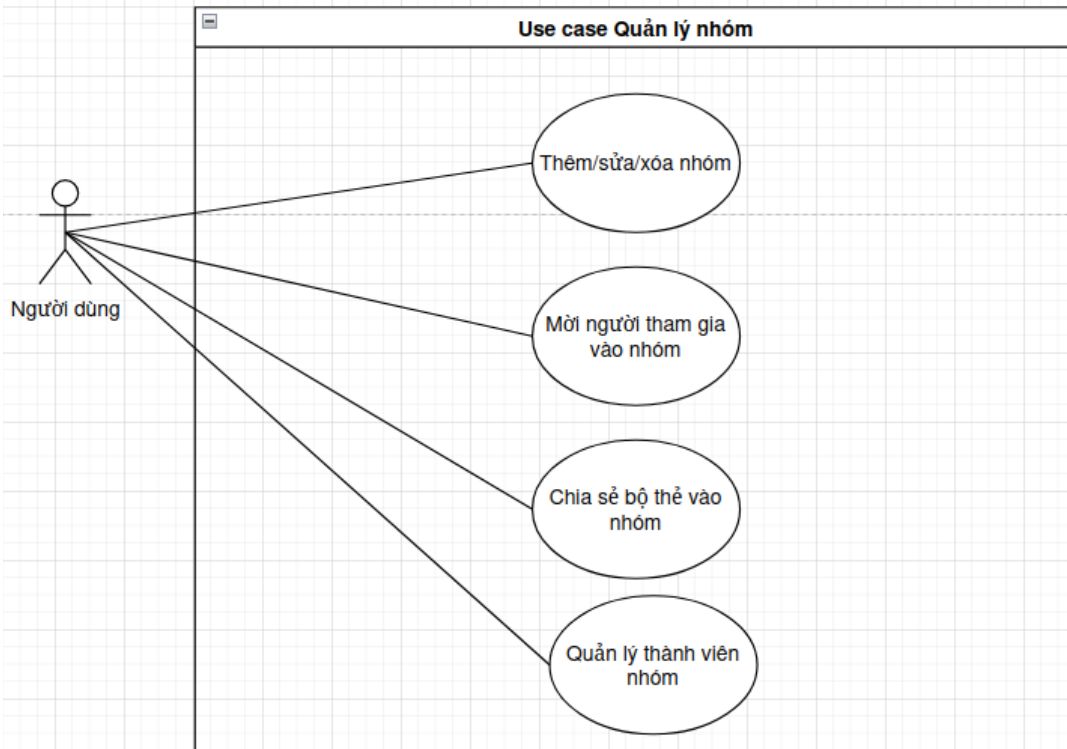
Hình 1. Biểu đồ use case học flashcard

2.3.6 Biểu đồ use case Quản lý thư mục



Hình 1. Biểu đồ use case quản lý thư mục

2.3.7 Biểu đồ use case Quản lý nhóm



Hình 1. Biểu đồ use case quản lý nhóm

2.4 Đặc tả biểu đồ Use – Case

2.4.1 Use case Đăng nhập bằng Google (Google OAuth)

Bảng 2. Đặc tả use-case đăng nhập bằng google

Tên Use Case	Đăng nhập bằng Google (Login with Google)
Mã số	UC-001
Tác nhân	Khách (Guest), Google Authentication Service
Mô tả	Cho phép người dùng đăng nhập hoặc đăng ký tài khoản mới thông qua tài khoản Google hiện có mà không cần tạo mật khẩu riêng.
Tiền điều kiện	<ul style="list-style-type: none"> - Khách đang ở trang Đăng nhập. - Hệ thống đã được đăng ký và cấu hình trên Google Cloud Console (Client ID, Client Secret).
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Khách: Nhấn nút "Tiếp tục với Google" trên giao diện. 2. Hệ thống (Client): Chuyển hướng trình duyệt đến trang xác thực của Google. 3. Google: Hiện thị form đăng nhập và yêu cầu cấp quyền truy cập (Scope: email, profile). 4. Khách: Đăng nhập tài khoản Google và nhấn "Đồng ý". 5. Google: Xác thực thành công và chuyển hướng về lại trang web (Callback URL) kèm theo mã xác thực (auth_code). 6. Hệ thống (Client): Gửi mã auth_code này xuống Backend API. 7. Hệ thống (Backend): <ul style="list-style-type: none"> - Dùng auth_code đổi lấy access_token từ Google. - Dùng access_token để lấy thông tin người dùng (Email, Avatar, Name). - Kiểm tra Email trong Database: <ul style="list-style-type: none"> + Nếu chưa tồn tại: Tạo tài khoản mới (user) và ghi nhận vai trò mặc định. + Nếu đã tồn tại: Cập nhật thông tin (Avatar/Tên) nếu cần. - Tạo phiên làm việc mới (session), sinh cặp JWT (Access Token & Refresh Token). 8. Hệ thống (Client): Nhận JWT, lưu vào LocalStorage/Cookie và chuyển hướng vào Trang chủ.

Luồng ngoại lệ	<p>Người dùng từ chối (User Denied): Tại bước 4, nếu khách nhấn "Hủy", Google trả về lỗi. Hệ thống hiển thị thông báo: "Đăng nhập bị hủy bỏ".</p> <p>Email bị khóa: Nếu tài khoản tồn tại nhưng is_active = false (bị ban), hệ thống từ chối đăng nhập và báo lỗi.</p> <p>Lỗi kết nối Google: Nếu API Google không phản hồi, hệ thống báo "Lỗi kết nối, vui lòng thử lại sau".</p>
Điều kiện sau	<ul style="list-style-type: none"> - Một bản ghi mới được tạo trong bảng session. - Thông tin thiết bị được cập nhật trong bảng device

2.4.2 Use case Quản lý thông tin cá nhân

Bảng 3. Đặc tả use-case quản lý thông tin cá nhân

Tên Use Case	Quản lý thông tin cá nhân (Manage User Profile)
Mã số	UC-002
Tác nhân	Người dùng đã đăng nhập (User)
Mô tả	Cho phép người dùng xem và cập nhật thông tin định danh (Tên, Avatar) và các cấu hình học tập cá nhân (Mục tiêu hàng ngày, Múi giờ).
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng: Truy cập vào trang "Hồ sơ cá nhân" (Profile). 2. Hệ thống: Tải thông tin hiện tại từ bảng user và user_setting để hiển thị lên Form. 3. Người dùng: Chỉnh sửa các thông tin mong muốn (ví dụ: đổi tên hiển thị, tăng mục tiêu học từ 10 lên 20 thẻ/ngày). 4. Người dùng: (Tùy chọn) Nhấn nút tải lên ảnh đại diện mới. 5. Người dùng: Nhấn nút "Lưu thay đổi". 6. Hệ thống (Client): Kiểm tra dữ liệu hợp lệ (Validate Form). Gửi yêu cầu cập nhật lên Server. 7. Hệ thống (Backend): <ul style="list-style-type: none"> - Nếu có file ảnh: Thực hiện upload lên Cloud Storage, lấy URL mới. - Sử dụng Transaction để cập nhật đồng thời bảng user và

	<p>user_setting.</p> <p>- Trả về thông tin profile mới nhất.</p> <p>8. Hệ thống (Client): Hiện thị thông báo "Cập nhật thành công" và cập nhật giao diện.</p>
Luồng ngoại lệ	<p>Lỗi định dạng ảnh: Người dùng upload file không phải ảnh hoặc > 5MB. Hệ thống báo lỗi và chặn request.</p> <p>Lỗi dữ liệu: Người dùng để trống tên. Hệ thống hiển thị lỗi Validation.</p> <p>Lỗi đồng bộ: Nếu cập nhật bảng user thành công nhưng user_setting thất bại, hệ thống Rollback (hoàn tác) cả hai để đảm bảo toàn vẹn dữ liệu.</p>
Điều kiện sau	<p>Dữ liệu trong Database được cập nhật. Ảnh cũ trên Cloud (nếu có) được đánh dấu xóa (hoặc xóa ngay tùy chính sách).</p>

2.4.3 Use case Quản lý bộ thẻ

Bảng 4. Đặc tả use-case quản lý bộ thẻ

Mã ca sử dụng	UC-003		
Tên ca sử dụng	Quản lý bộ thẻ Flashcard (Flashcard Set Management)		
Tác nhân	Người dùng, Quản trị viên		
Mô tả	Cho phép người dùng tạo, xem, chỉnh sửa, xóa và quản lý các bộ flashcard. Quản trị viên có thể xem và quản lý bộ thẻ của mọi người dùng (tùy chính sách).		
Điều kiện trước (Pre-Condition)	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống. - Có ít nhất 0 hoặc nhiều bộ thẻ đã được lưu trước đó. 		
Điều kiện sau (Post-Condition)	<ul style="list-style-type: none"> - Nếu chỉ xem → ghi log truy cập. - Nếu tạo mới → lưu bộ thẻ vào DB và hiển thị trong danh sách. - Nếu cập nhật → lưu thay đổi & ghi lịch sử thay đổi. - Nếu xóa → đánh dấu soft-delete hoặc xóa vĩnh viễn tùy cấu hình. 		
Kịch bản		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng mở mục "Bộ thẻ của tôi" (My Flashcards).	Hệ thống truy xuất danh sách bộ thẻ từ database → hiển thị.
	2	Người dùng chọn: Tạo mới / Xem / Chỉnh sửa / Xóa bộ thẻ.	Hệ thống phản hồi theo từng chức năng tương ứng.
	3	Nếu chọn Tạo mới → nhập thông tin bộ thẻ (Tên, mô tả, thẻ,...).	Nếu chọn Tạo mới → nhập thông tin bộ thẻ (Tên, mô tả, thẻ,...). Validate dữ liệu nhập. Nếu ok → Lưu bộ thẻ mới → Hiển thị thông báo thành công.
	4	Nếu chọn Xem → chọn bộ thẻ.	Hệ thống hiển thị danh sách thẻ trong bộ đó + lượt học + ngày tạo.
	5	Nếu chọn Chỉnh sửa → Người dùng cập nhật thông tin/thẻ.	Validate dữ liệu → lưu cập nhật vào DB → thông báo thành công.
	6	Nếu chọn Xóa → Người dùng xác nhận.	Hệ thống thực hiện soft delete → cập nhật DB → thông báo thành công.

2.4.4 Use case Tạo thẻ tự động từ prompt

Bảng 5. Đặc tả use-case tạo thẻ từ đoạn văn bản (prompt)

Mã ca sử dụng	UC-004		
Tên ca sử dụng	Tạo thẻ tự động từ Prompt (Generate Flashcards from Prompt)		
Tác nhân	Người dùng (User), Dịch vụ AI (AI Service)		
Mô tả	Cho phép người dùng nhập một chủ đề, từ khóa hoặc một đoạn văn bản ngắn. Hệ thống sử dụng AI để phân tích và tự động sinh ra danh sách các thẻ ghi nhớ (Flashcards) phù hợp.		
Tiền điều kiện	<ol style="list-style-type: none"> 1. Người dùng đã đăng nhập vào hệ thống. 2. Người dùng đang ở trong giao diện chi tiết của một bộ thẻ. 3. Kết nối Internet ổn định. 		
Điều kiện sau	<p>- Thành công: Các thẻ mới được lưu vào cơ sở dữ liệu và hiển thị lên giao diện.</p> <p>- Thất bại: Giữ nguyên trạng thái cũ, hiển thị thông báo lỗi.</p>		
Kịch bản chính		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng nhấn nút "Tạo thẻ bằng AI" trên thanh công cụ của bộ thẻ.	Hệ thống hiển thị hộp thoại (Modal) nhập liệu, bao gồm các trường: Nội dung (Prompt), Số lượng thẻ, Ngôn ngữ.
	2	Người dùng nhập yêu cầu (Ví dụ: " <i>Từ vựng về Marketing</i> ") và nhấn nút "Tạo ngay".	<ol style="list-style-type: none"> 1. Hệ thống kiểm tra dữ liệu đầu vào (Validation). 2. Nếu hợp lệ: Hiển thị trạng thái "<i>AI đang xử lý...</i>" (Loading). 3. Hệ thống Backend gửi yêu cầu đến AI Service và chờ kết quả.

	3	<i>(Người dùng chờ đợi)</i>	<ol style="list-style-type: none"> Hệ thống nhận kết quả từ AI, phân tích định dạng JSON. Hiển thị danh sách các thẻ nháp (Draft Cards) ở chế độ Xem trước (Preview).
	4	<p>Người dùng xem lại kết quả (Preview):</p> <ul style="list-style-type: none"> Bỏ chọn các thẻ không mong muốn. Chỉnh sửa nội dung thẻ (nếu cần). 	<p>Hệ thống cập nhật danh sách thẻ nháp theo thao tác chỉnh sửa của người dùng ngay trên giao diện.</p>
	5	<p>Người dùng nhấn nút "Lưu vào bộ thẻ".</p>	<ol style="list-style-type: none"> Hệ thống gửi danh sách thẻ đã chọn về Server. Lưu dữ liệu vào bảng card trong Database. Hiển thị thông báo "<i>Thêm thẻ thành công</i>". Đóng hộp thoại và tải lại danh sách thẻ mới.
<p>Kịch bản ngoại lệ</p>	<ul style="list-style-type: none"> - E1. Dữ liệu nhập không hợp lệ: Tại bước 2, nếu người dùng để trống nội dung Prompt, hệ thống hiển thị thông báo lỗi ngay dưới ô nhập liệu: "<i>Vui lòng nhập chủ đề hoặc nội dung cần tạo</i>". - E2. Lỗi từ phía AI (Timeout/Error): Tại bước 3, nếu quá thời gian chờ (ví dụ 30s) hoặc AI Service trả về lỗi, hệ thống tắt trạng thái Loading và hiển thị thông báo: "<i>Hệ thống AI đang bận hoặc gặp sự cố. Vui lòng thử lại sau.</i>" - E3. Người dùng hủy bỏ: Tại bước 4 (Xem trước), nếu người dùng nhấn nút "Hủy" hoặc tắt hộp thoại, hệ thống sẽ hủy bỏ toàn bộ quá trình và không lưu bất kỳ thẻ nào vào cơ sở dữ liệu. 		

2.4.5 Use case Tạo thẻ tự động từ tài liệu (File Upload)

Bảng 6. Đặc tả use-case tạo thẻ từ file tài liệu

Tên Use Case	Tạo thẻ từ File PDF (Generate from File)		
Mã số	UC-005		
Tác nhân	Người học (User), AI Service, Cloud Storage		
Mô tả	Người dùng tải lên một tệp tin pdf, hệ thống xử lý nền để tạo ra bộ thẻ.		
Tiền điều kiện	Người dùng đã đăng nhập. File có định dạng hợp lệ (PDF, DOCX) và dung lượng < 10MB.		
Điều kiện sau	<p>- Thành công: Nội dung file được xử lý, các thẻ mới được tạo và lưu vào cơ sở dữ liệu.</p> <p>- Thất bại: Hiển thị thông báo lỗi (Lỗi định dạng, lỗi đọc file, hoặc lỗi AI).</p>		
Luồng sự kiện chính		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng nhấn nút " Tải lên tài liệu " (Upload File) trong giao diện tạo bộ thẻ.	Hệ thống hiển thị hộp thoại chọn tệp tin hoặc vùng kéo thả file (Drag & Drop).
	2	Người dùng chọn một tệp tin từ máy tính và nhấn nút " Submit ".	<p>1. Hệ thống (Client) kiểm tra định dạng và kích thước file.</p> <p>2. Nếu hợp lệ: Hiển thị trạng thái "<i>Đang tải lên và phân tích...</i>".</p> <p>3. Hệ thống Backend nhận file, trích xuất văn bản (Text Extraction) và gửi nội dung đến AI Service.</p>

	3	<i>(Người dùng chờ đợi quá trình xử lý)</i>	<ol style="list-style-type: none"> Hệ thống nhận kết quả JSON từ AI. Hiển thị danh sách các thẻ nháp (Draft Cards) ở chế độ Xem trước (Preview).
	4	<p>Người dùng xem lại kết quả (Preview):</p> <ul style="list-style-type: none"> Đối chiếu nội dung thẻ với tài liệu gốc. Chỉnh sửa hoặc xóa các thẻ không chính xác. 	<p>Hệ thống cập nhật danh sách thẻ nháp theo thao tác của người dùng.</p>
	5	<p>Người dùng nhấn nút "Lưu vào bộ thẻ".</p>	<ol style="list-style-type: none"> Hệ thống gửi danh sách thẻ cuối cùng về Server. Lưu dữ liệu vào bảng card trong Database. Hiển thị thông báo <i>"Đã tạo n thẻ từ tài liệu thành công"</i>. Đóng hộp thoại và tải lại danh sách thẻ.
<p>Luồng ngoại lệ</p>	<p>File bị lỗi/File ảnh: Nếu thư viện không trích xuất được text (do PDF là file scan ảnh), hệ thống ghi log lỗi và thông báo: "Không thể đọc nội dung file, vui lòng thử file khác".</p> <p>AI quá tải: Nếu văn bản quá dài vượt Token limit, Worker sẽ thực hiện cắt nhỏ văn bản (Chunking) trước khi gửi.</p>		

2.4.6 Use case Quản lý thẻ

Bảng 7. Đặc tả use-case quản lý danh thẻ trong 1 bộ thẻ

Mã ca sử dụng	UC-006		
Tên ca sử dụng	Quản lý thẻ (Manage Flashcard Items)		
Tác nhân	Người dùng, Quản trị viên		
Mô tả	Cho phép người dùng tạo, xem, chỉnh sửa, xóa thẻ flashcard trong một bộ thẻ. Hỗ trợ thêm nhanh, nhập hàng loạt, gán tag, thay đổi thứ tự thẻ và đánh dấu độ khó. Quản trị viên có thể quản lý thẻ của mọi bộ nếu có quyền.		
Điều kiện trước	<ul style="list-style-type: none"> - Người dùng đã đăng nhập. - Người dùng đã chọn một bộ thẻ cần thao tác. 		
Điều kiện sau	<ul style="list-style-type: none"> - Nếu xem → ghi nhật ký truy cập. - Nếu thêm mới → dữ liệu được lưu vào DB. - Nếu cập nhật → thay đổi được lưu và ghi vào lịch sử chỉnh sửa. - Nếu xóa → xóa mềm hoặc xóa cứng tùy quyền. 		
Kịch bản		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng truy cập vào danh sách thẻ của bộ thẻ .	Hệ thống truy vấn DB và hiển thị danh sách thẻ + trạng thái học + độ khó.
	2	Người dùng chọn thao tác: Thêm thẻ mới / Chỉnh sửa / Xóa / Xem chi tiết .	Hệ thống chuyển sang màn hình tương ứng.
	3	Thêm thẻ mới → nhập "Term" + "Definition" (có thể kèm hình ảnh/audio).	Validate dữ liệu → Nếu hợp lệ lưu DB → thông báo thành công.
	4	Chỉnh sửa thẻ → thay đổi nội dung/độ khó/thứ tự.	Validate → Lưu DB → hiển thị cập nhật mới ngay tại giao diện.
	5	Xóa thẻ → hệ thống yêu cầu xác nhận.	Nếu xác nhận → Xóa mềm thẻ → cập nhật danh sách.
	6	Xem chi tiết → click vào thẻ.	Hệ thống hiển thị popup/chi tiết nội dung đầy đủ.

2.4.7 Use case Học flashcard

Bảng 8. Đặc tả use-case học flashcard

Mã ca sử dụng	UC-007		
Tên ca sử dụng	Chế độ học Flashcard (Flashcard Learning Mode)		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng học flashcard theo nhiều chế độ (flip card, autoplay, shuffle, đánh dấu độ khó, tracking tiến độ). Sau mỗi lượt học, hệ thống ghi nhận kết quả để phục vụ gợi ý ôn tập.		
Điều kiện trước	<ul style="list-style-type: none"> - Người dùng đã đăng nhập. - Có ít nhất một bộ thẻ hoặc danh sách thẻ đã chọn để học. 		
Điều kiện sau	<ul style="list-style-type: none"> - Tiến độ học được cập nhật vào DB. - Lưu lại các đánh giá độ khó và lịch sử học. - Nếu hoàn tất bộ thẻ → hệ thống có thể đề xuất lịch ôn tập tiếp theo (spaced repetition). 		
Kịch bản		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng chọn Học ngay tại một bộ flashcard.	Hệ thống load danh sách thẻ và hiển thị màn hình học.
	2	Người dùng lật thẻ (Term → Definition hoặc ngược lại).	Hệ thống hiển thị nội dung mặt sau.
	3	Người dùng đánh dấu độ khó: Easy / Medium / Hard / Again.	Hệ thống ghi nhận trạng thái và cập nhật tiến độ.
	4	Người dùng chuyển sang thẻ tiếp theo (Next).	Hệ thống hiển thị thẻ mới theo thứ tự hoặc shuffle.
	5	Khi học xong toàn bộ, hệ thống hiển thị tổng kết phiên học.	Ghi log phiên học + cập nhật progress + hiển thị kết quả.

2.4.8 Use case Quản lý thư mục

Bảng 9. Đặc tả use-case quản lý thư mục

Mã ca sử dụng	UC-008		
Tên ca sử dụng	Quản lý thư mục (Folder Management)		
Tác nhân	Người dùng		
Mô tả	Cho phép người dùng tạo, đổi tên, di chuyển, xem danh sách thư mục, sắp xếp và xóa thư mục chứa bộ thẻ hoặc tài liệu. Hỗ trợ phân cấp nhiều cấp thư mục, phục vụ việc tổ chức flashcard/eBook gọn gàng.		
Điều kiện trước	- Người dùng đã đăng nhập. - Bảng thư mục đã tồn tại trong hệ thống.		
Điều kiện sau	- Nếu tạo thư mục mới → lưu vào DB. - Nếu đổi tên/di chuyển/xóa → cập nhật DB + log history. - Nếu xóa thư mục chứa dữ liệu → xử lý theo rule (xóa mềm hoặc chuyển vào thùng rác).		
Kịch bản		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng mở màn quản lý thư mục.	Hệ thống truy xuất danh sách thư mục và hiển thị dạng list/tree.
	2	Người dùng chọn Tạo thư mục mới .	Hệ thống hiển thị form nhập tên thư mục.
	3	User nhập tên thư mục → xác nhận.	Validate trùng tên cùng cấp → Lưu DB → Thư mục mới xuất hiện.
	4	User chọn thư mục → xem nội dung flashcard/bộ thẻ.	Hệ thống load và hiển thị danh sách item bên trong.
	5	User có thể đổi tên / kéo thả để di chuyển thư mục.	Hệ thống cập nhật DB và UI.
	6	User xóa thư mục nếu không cần thiết.	Hệ thống xác nhận → Xóa mềm → Cập nhật UI.

2.4.9 Use case Dashboard thống kê

Bảng 10. Đặc tả use -case xem dashboard thống kê

Mục	Nội dung		
Mã ca sử dụng	UC-009		
Tên ca sử dụng	Xem Dashboard thống kê (View Statistics Dashboard)		
Tác nhân	Người dùng (User)		
Mô tả	Cho phép người dùng xem tổng quan về tiến độ học tập của bản thân, bao gồm: số lượng thẻ cần ôn, biểu đồ phân bố trạng thái thẻ, lịch sử hoạt động (Heatmap) và chuỗi ngày học liên tục (Streak).		
Tiền điều kiện	<ol style="list-style-type: none"> 1. Người dùng đã đăng nhập vào hệ thống. 2. Người dùng đã thực hiện ít nhất một vài phiên học (để có dữ liệu hiển thị). 		
Điều kiện sau	<ul style="list-style-type: none"> - Hệ thống hiển thị các biểu đồ và chỉ số chính xác dựa trên dữ liệu học tập hiện tại. - Không có dữ liệu nào bị thay đổi (Use case chỉ đọc - Read Only). 		
Kịch bản chính		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng truy cập vào trang "Tổng quan" (Dashboard) hoặc nhấn vào tab "Thống kê".	Hệ thống gửi yêu cầu lấy dữ liệu tổng hợp về tiến độ học tập của người dùng từ Server.
	2	(Người dùng chờ tải dữ liệu)	<ol style="list-style-type: none"> 1. Hệ thống tính toán các chỉ số: Streak hiện tại, Tổng số thẻ đã thuộc, Số thẻ cần ôn hôm nay. 2. Hệ thống truy vấn lịch sử học tập để vẽ biểu đồ nhiệt (Heatmap) và biểu đồ tròn trạng thái (Pie Chart).

			3. Hiện thị giao diện Dashboard với các Widget thông tin trực quan.
	3	<p>Người dùng xem thông tin tổng quan:</p> <ul style="list-style-type: none"> - Số lượng thẻ đến hạn (Due Cards). - Biểu đồ phân bố (New, Learning, Review, Mature). <p>Hệ thống hiển thị chi tiết con số khi người dùng di chuột (Hover) vào các phần của biểu đồ.</p>	Hệ thống hiển thị chi tiết con số khi người dùng di chuột (Hover) vào các phần của biểu đồ.
	4	<p>Người dùng thay đổi bộ lọc thời gian (Ví dụ: Chuyển từ "7 ngày qua" sang "30 ngày qua" hoặc "Toàn thời gian").</p>	<ol style="list-style-type: none"> 1. Hệ thống nhận tham số thời gian mới. 2. Tính toán lại dữ liệu biểu đồ cột (Bar Chart) theo khoảng thời gian đã chọn. 3. Cập nhật lại giao diện biểu đồ mà không tải lại toàn bộ trang.
	5	<p>Người dùng nhấn vào một ngày cụ thể trên Biểu đồ nhiệt (Heatmap).</p>	Hệ thống hiển thị Tooltip hoặc danh sách chi tiết các hoạt động đã thực hiện trong ngày đó (Ví dụ: "Đã ôn 50 thẻ, Đã tạo 1 bộ thẻ").
Kịch bản ngoại lệ (Alternative Flow)	<p>E1. Người dùng mới (Chưa có dữ liệu): Tại bước 2, nếu hệ thống phát hiện người dùng chưa học thẻ nào, thay vì hiện biểu đồ trống, hệ thống hiển thị màn hình Onboarding (Chào mừng): "Bạn chưa có dữ liệu học tập. Hãy bắt đầu bài học đầu tiên ngay!" kèm nút kêu gọi hành động (Call to Action).</p>		

	E2. Lỗi tải dữ liệu (API Error): Nếu Server gặp sự cố khi tính toán thống kê (do dữ liệu quá lớn hoặc lỗi DB), hệ thống hiển thị thông báo: "Không thể tải dữ liệu thống kê lúc này. Vui lòng tải lại trang." và hiển thị nút " Thử lại ".
--	--

2.4.10 Use case Chế độ ôn tập Trắc nghiệm & Game (Gamification)

Bảng 11. Đặc tả use-case ôn tập chế độ quiz

Mục	Nội dung		
Mã ca sử dụng	UC-010		
Tên ca sử dụng	Ôn tập Trắc nghiệm & Game (Quiz & Gamification Mode)		
Tác nhân	Người dùng (User)		
Mô tả	<p>Cung cấp các phương pháp ôn tập thay thế cho Flashcard truyền thống để tăng hứng thú:</p> <ol style="list-style-type: none"> Trắc nghiệm (Quiz): Chọn 1 đáp án đúng trong 4 lựa chọn. Ghép thẻ (Matching): Tìm và nối các cặp thuật ngữ - định nghĩa tương ứng trong thời gian nhanh nhất. 		
Tiền điều kiện	<ol style="list-style-type: none"> Người dùng đã đăng nhập. Bộ thẻ phải có tối thiểu 4 thẻ (để tạo đủ đáp án nhiều cho trắc nghiệm). Bộ thẻ phải có tối thiểu 6 thẻ (để chơi chế độ ghép thẻ hiệu quả). 		
Điều kiện sau	<ul style="list-style-type: none"> Kết quả học tập được ghi nhận (Điểm số, thời gian hoàn thành). (Tùy chọn) Cập nhật tiến độ SRS: Thẻ trả lời đúng được đánh dấu là "Good", sai là "Again". 		
Kịch bản chính		Hành động của tác nhân	Phản hồi của hệ thống
	1	Người dùng chọn bộ thẻ và nhấn nút "Làm trắc nghiệm" (Start Quiz).	<ol style="list-style-type: none"> Hệ thống lấy danh sách thẻ cần ôn. Với mỗi thẻ (câu hỏi), hệ thống lấy ngẫu nhiên 3

			<p>định nghĩa của các thẻ khác trong cùng bộ để làm đáp án nhiễu (Distractors).</p> <p>3. Hiện thị câu hỏi đầu tiên kèm 4 lựa chọn.</p>
	2	Người dùng đọc câu hỏi và chọn 1 trong 4 đáp án.	<p>1. Hệ thống kiểm tra đáp án ngay lập tức.</p> <p>2. Nếu đúng: Phát âm thanh "Đúng", hiển thị màu xanh, cộng điểm.</p> <p>3. Nếu sai: Phát âm thanh báo lỗi, hiển thị màu đỏ ở câu sai và viền xanh ở câu đúng để người dùng biết.</p>
	3	Người dùng nhấn "Tiếp tục" (hoặc hệ thống tự chuyển sau 2s).	Hệ thống hiển thị câu hỏi tiếp theo trong danh sách.
	4	Người dùng hoàn thành câu hỏi cuối cùng.	<p>1. Hệ thống tính toán tổng điểm và tỷ lệ chính xác (Accuracy).</p> <p>2. Hiện thị Màn hình tổng kết: Số câu đúng/sai, Thời gian làm bài.</p> <p>3. Cung cấp tùy chọn: "Làm lại các câu sai" hoặc "Về trang chủ".</p>

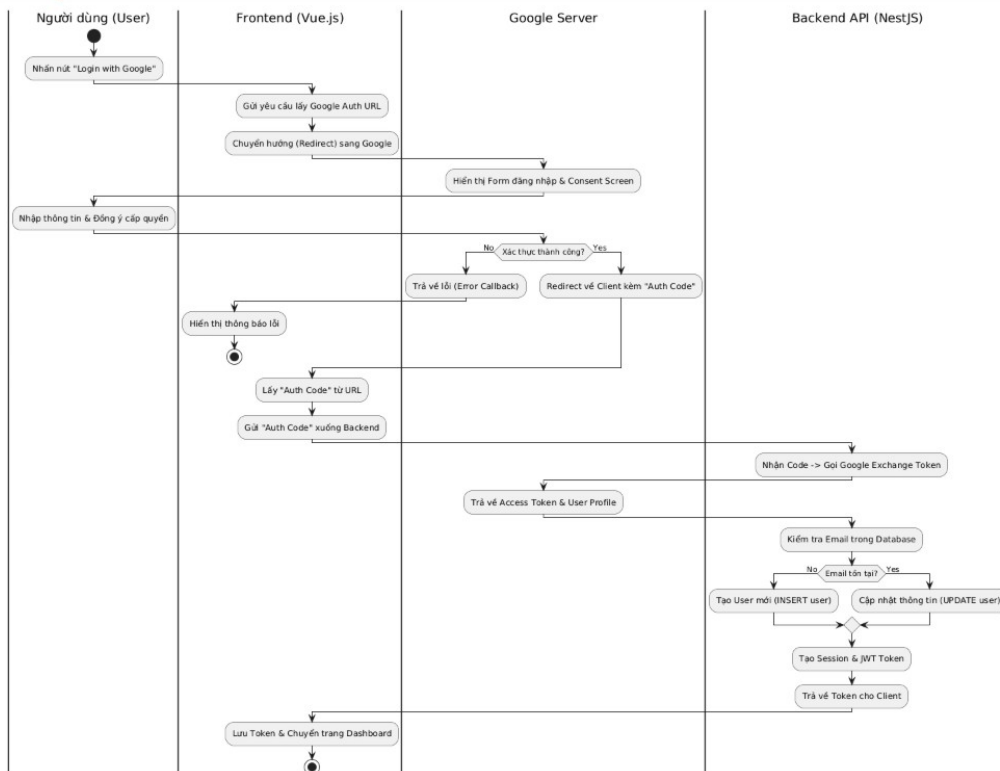
2.4.11 Use case Quản lý nhóm

Bảng 12. Đặc tả use-case quản lý nhóm

Mã ca sử dụng	UC-011		
Tên ca sử dụng	Quản lý nhóm (Create/Manage Groups)		
Tác nhân	Người dùng, Quản trị viên nhóm (Group Owner/Admin)		
Mô tả	Cho phép tạo nhóm học tập, mời thành viên, quản lý quyền, chia sẻ flashcard/tài liệu trong nhóm.		
Điều kiện trước (Pre-Condition)	<ul style="list-style-type: none"> - Người dùng đã đăng nhập. - Tài khoản không bị hạn chế tạo nhóm (nếu hệ thống có quota). 		
Điều kiện sau (Post-Condition)	<ul style="list-style-type: none"> - Nhóm được tạo/ cập nhật/ xóa theo thao tác. - Lịch sử hoạt động nhóm được lưu lại. - Thành viên được thêm/bỏ/đổi quyền thành công. 		
Kịch bản		Hành động của tác nhân	Phản hồi của hệ thống
	1	User vào trang Groups → chọn <i>Create Group</i> .	Hệ thống mở form tạo nhóm.
	2	User nhập tên, mô tả nhóm → <i>Confirm</i> .	Validate thông tin → Tạo nhóm → Gán User làm Owner.
	3	Owner mời thành viên (qua email/link invite).	Hệ thống gửi invite/ tạo link join.
	4	Thành viên chấp nhận tham gia.	Hệ thống thêm user vào danh sách thành viên.
	5	Owner/Admin quản lý nhóm: đổi tên, ảnh nhóm, mô tả.	Hệ thống cập nhật DB + hiển thị UI mới.
	6	Owner/Admin quản lý quyền thành viên.	Hệ thống cập nhật role và log thao tác.

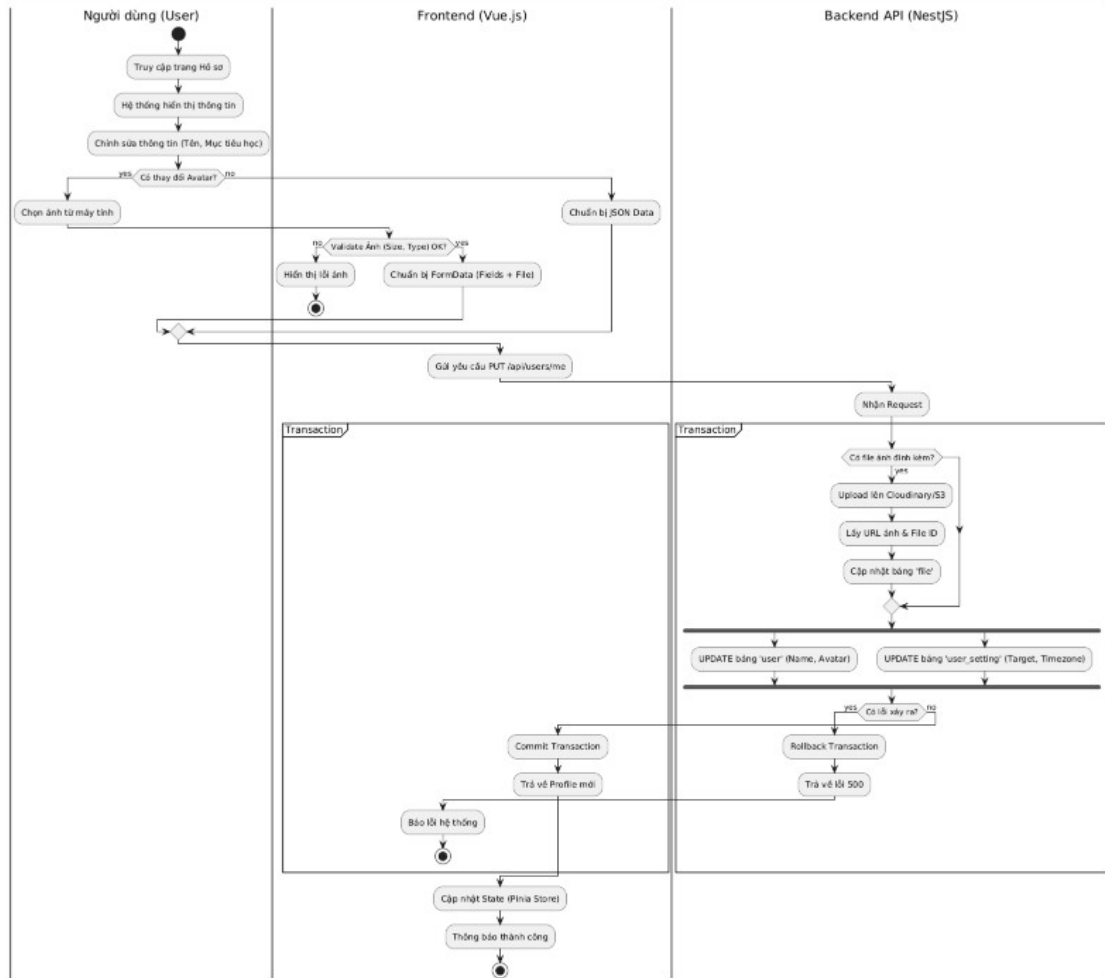
2.5 Biểu đồ hoạt động

2.5.1 Biểu đồ hoạt động của chức năng đăng nhập bằng Google OAuth



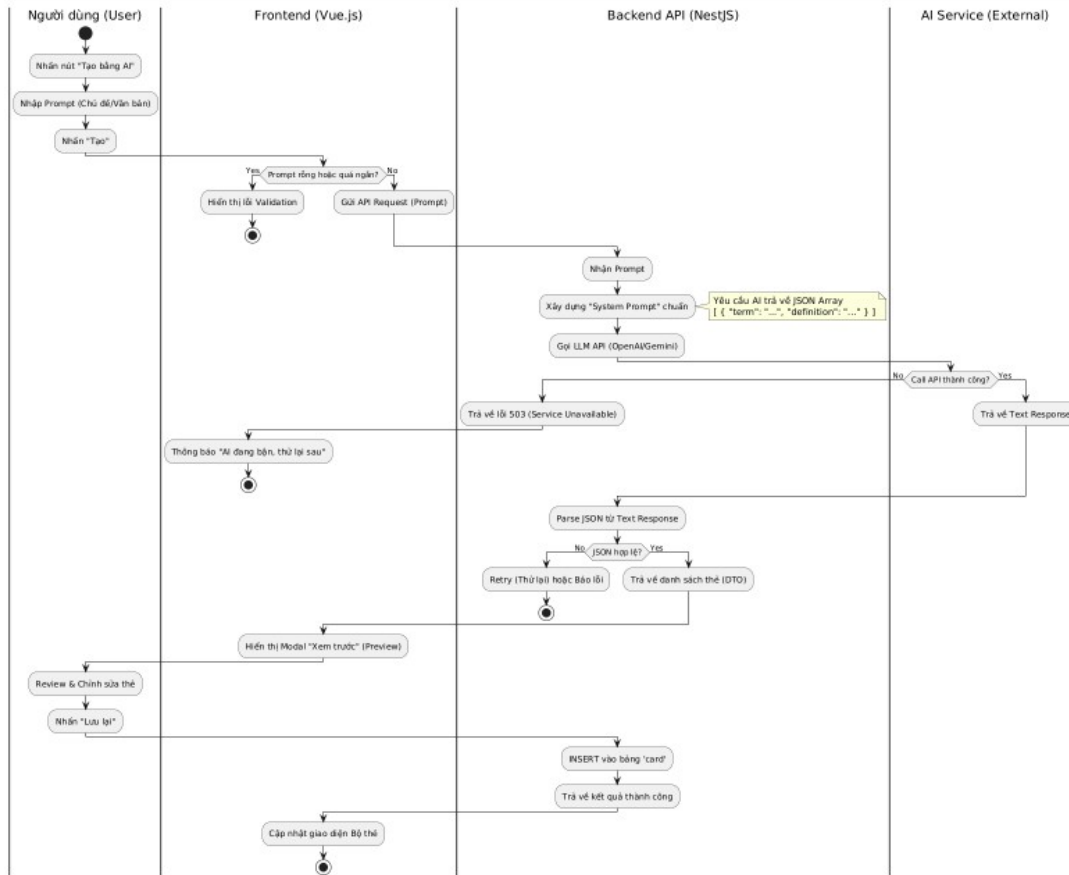
Hình 1. Biểu đồ hoạt động của chức năng đăng nhập bằng google

2.5.2 Biểu đồ hoạt động của chức năng quản lý thông tin cá nhân



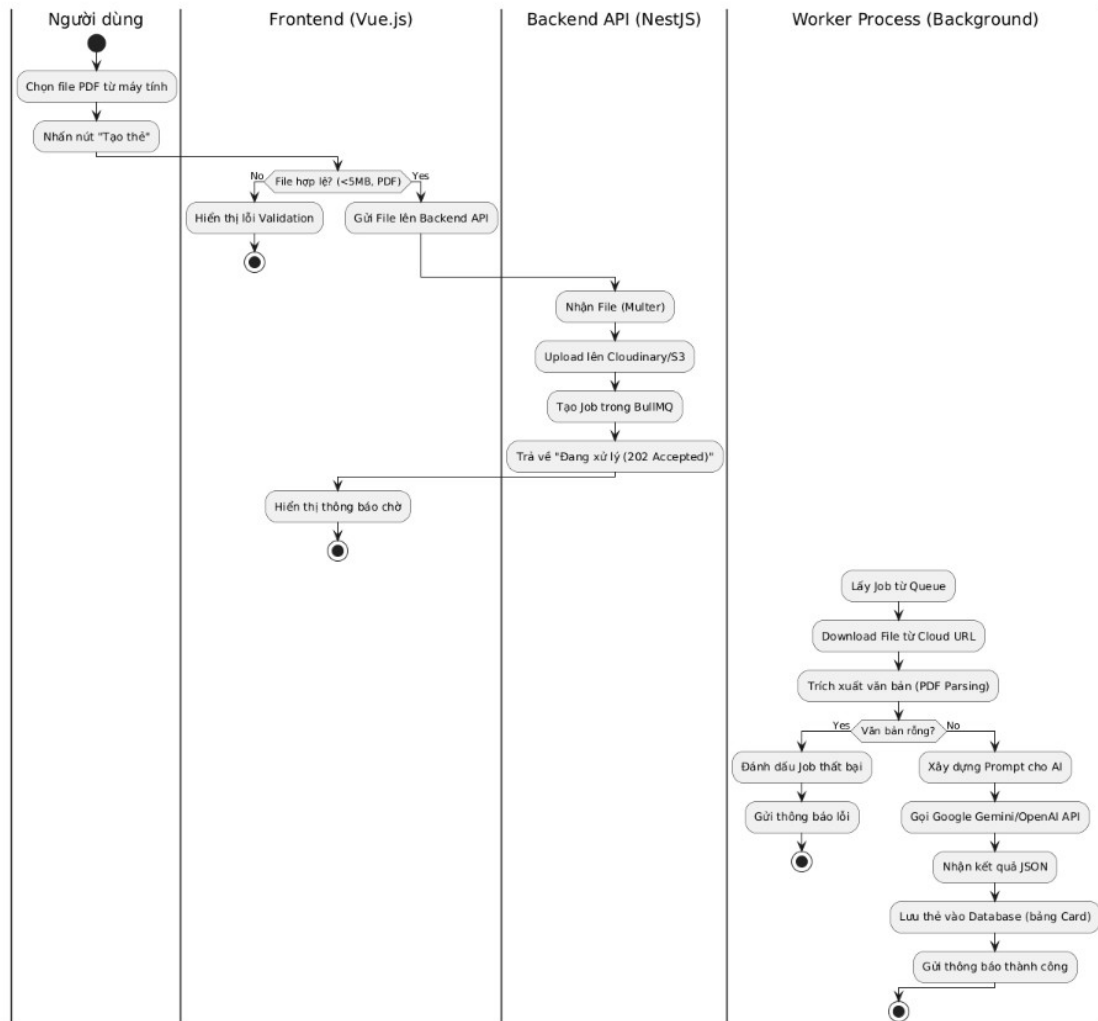
Hình 1. Biểu đồ hoạt động chức năng quản lý thông tin cá nhân

2.5.3 Biểu đồ hoạt động của chức năng tạo bộ thẻ từ prompt



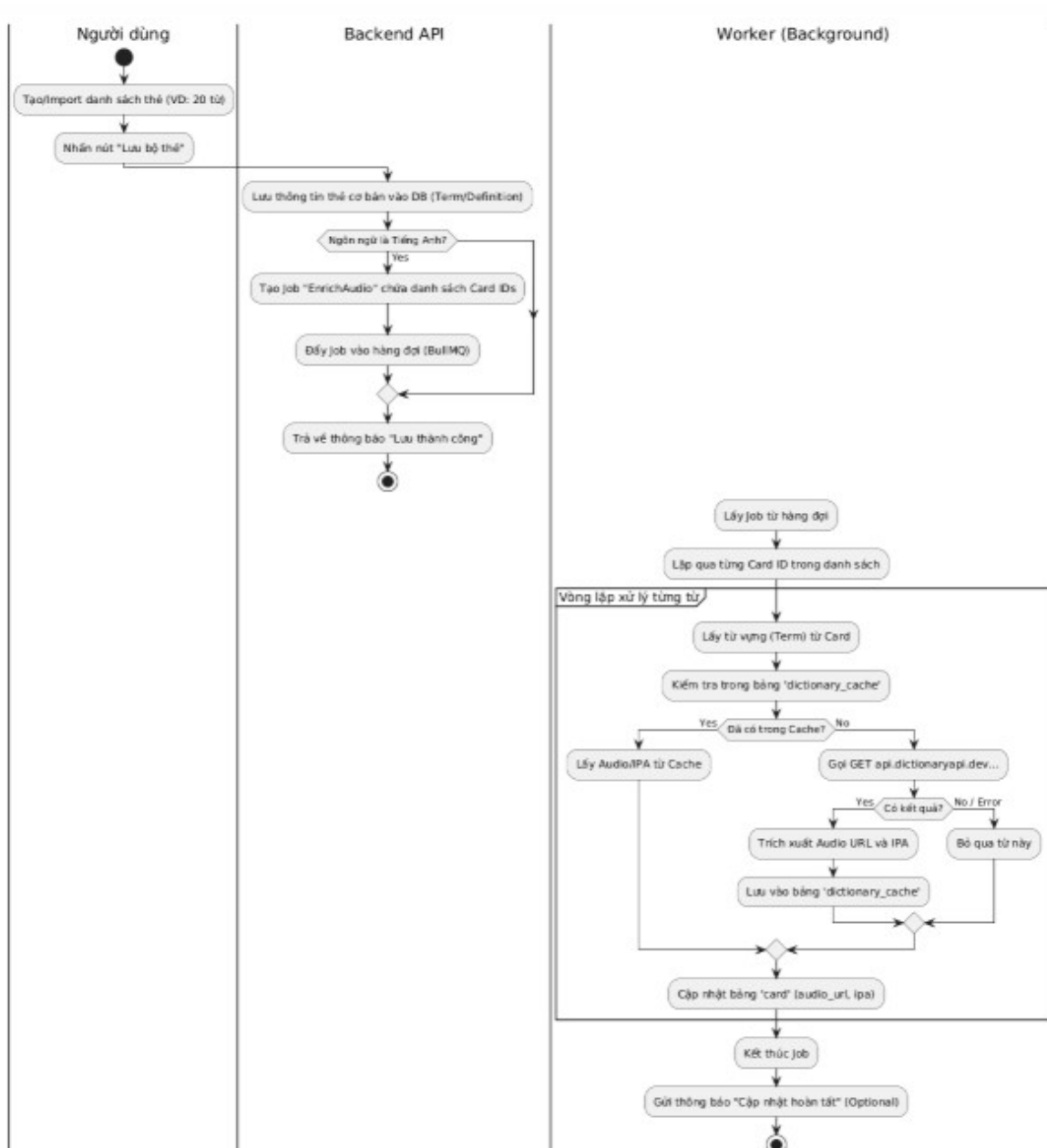
Hình 1. Biểu đồ hoạt động chức năng tạo bộ thẻ từ prompt

2.5.4 Biểu đồ hoạt động của chức năng tạo bộ thẻ từ file tài liệu PDF



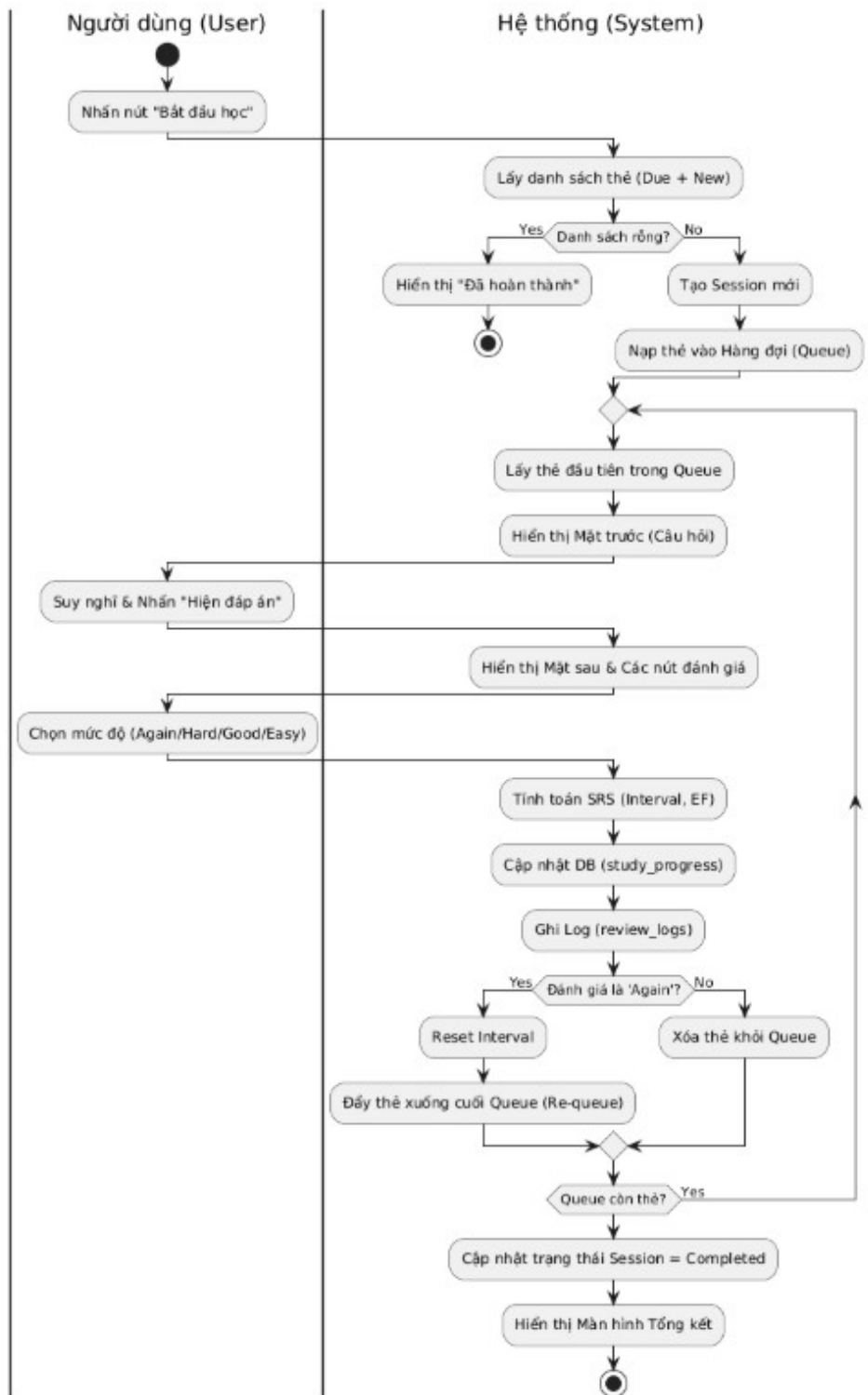
Hình 1. Biểu đồ hoạt động chức năng tạo bộ thẻ từ file pdf

2.5.5 Biểu đồ hoạt động của chức năng tự động bổ sung phát âm (Text-to-Speech Enrichment)



Hình 1. Biểu đồ hoạt động chức năng tự động bổ sung phát âm

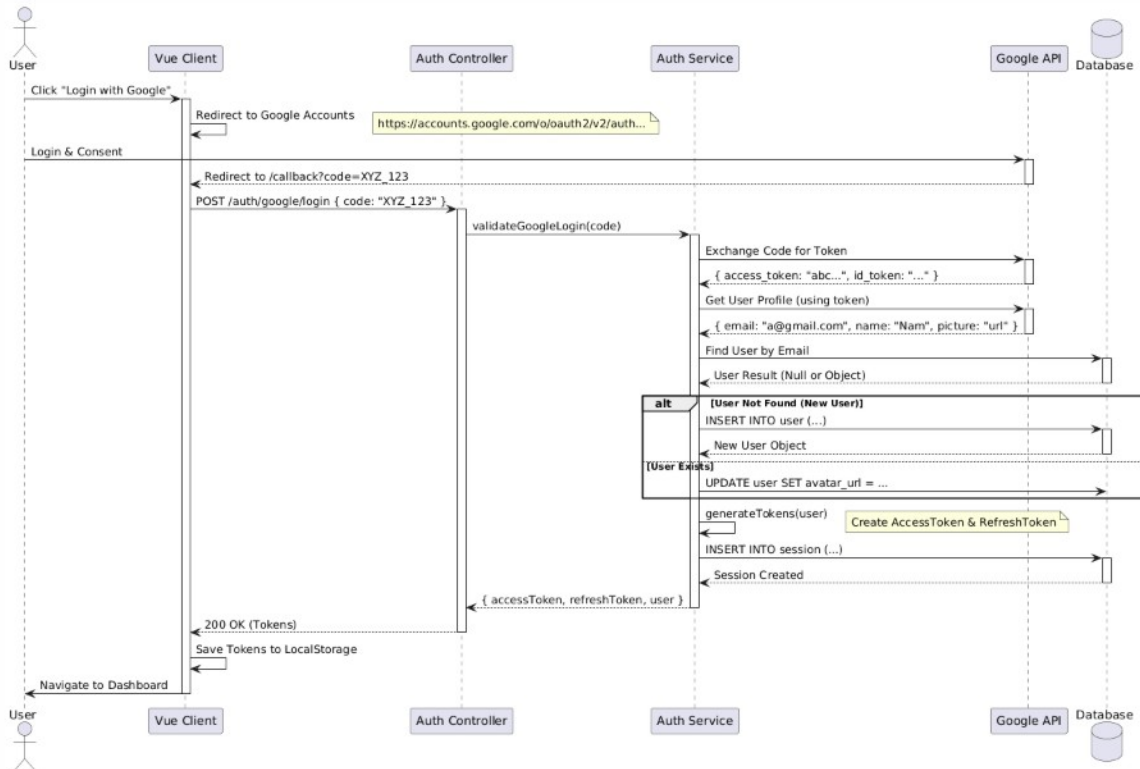
2.5.6 Biểu đồ hoạt động của chức năng học flashcard



Hình 1. Biểu đồ hoạt động chức năng học flashcard

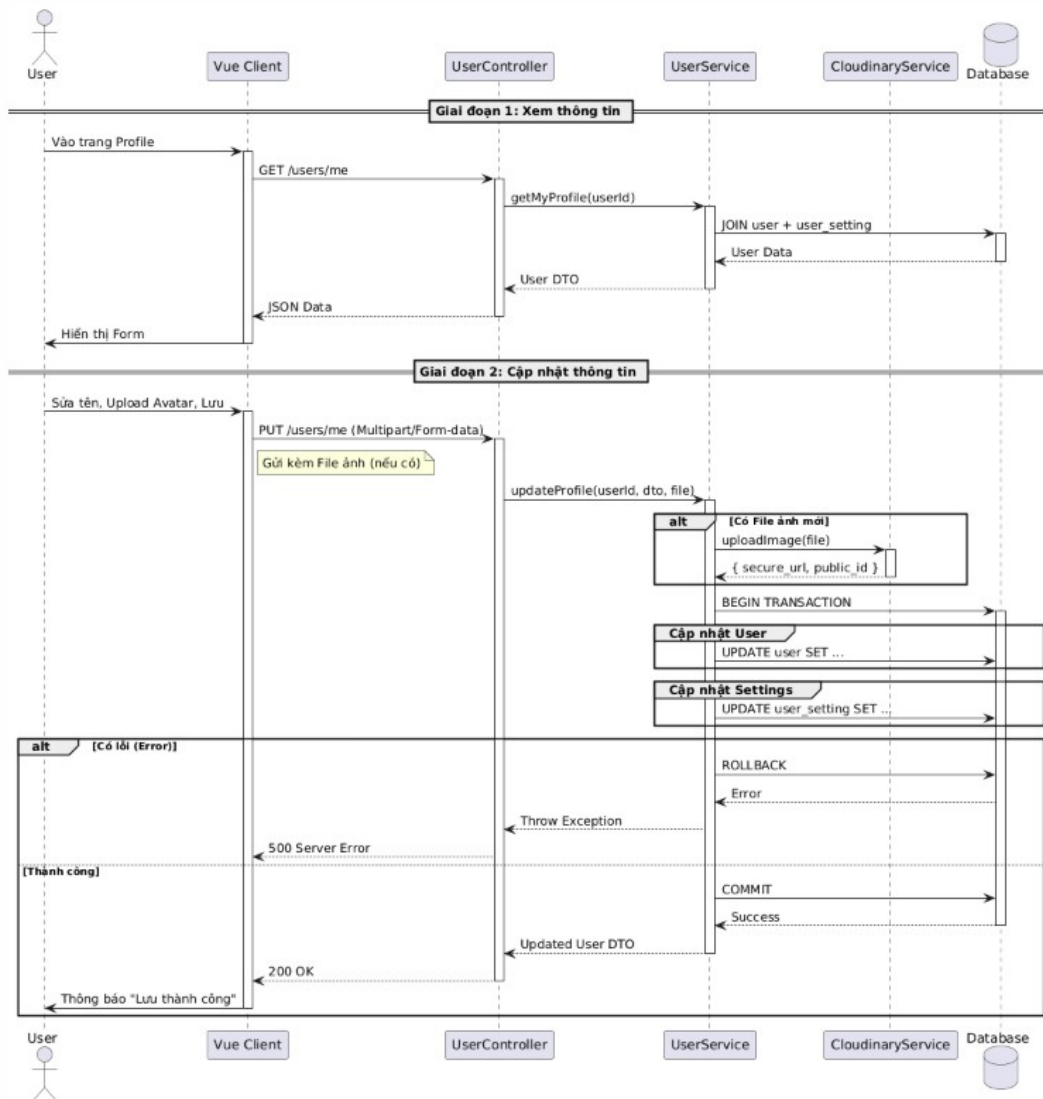
2.6 Biểu đồ tuần tự

2.6.1 Biểu đồ tuần tự của chức năng đăng nhập bằng Google OAuth



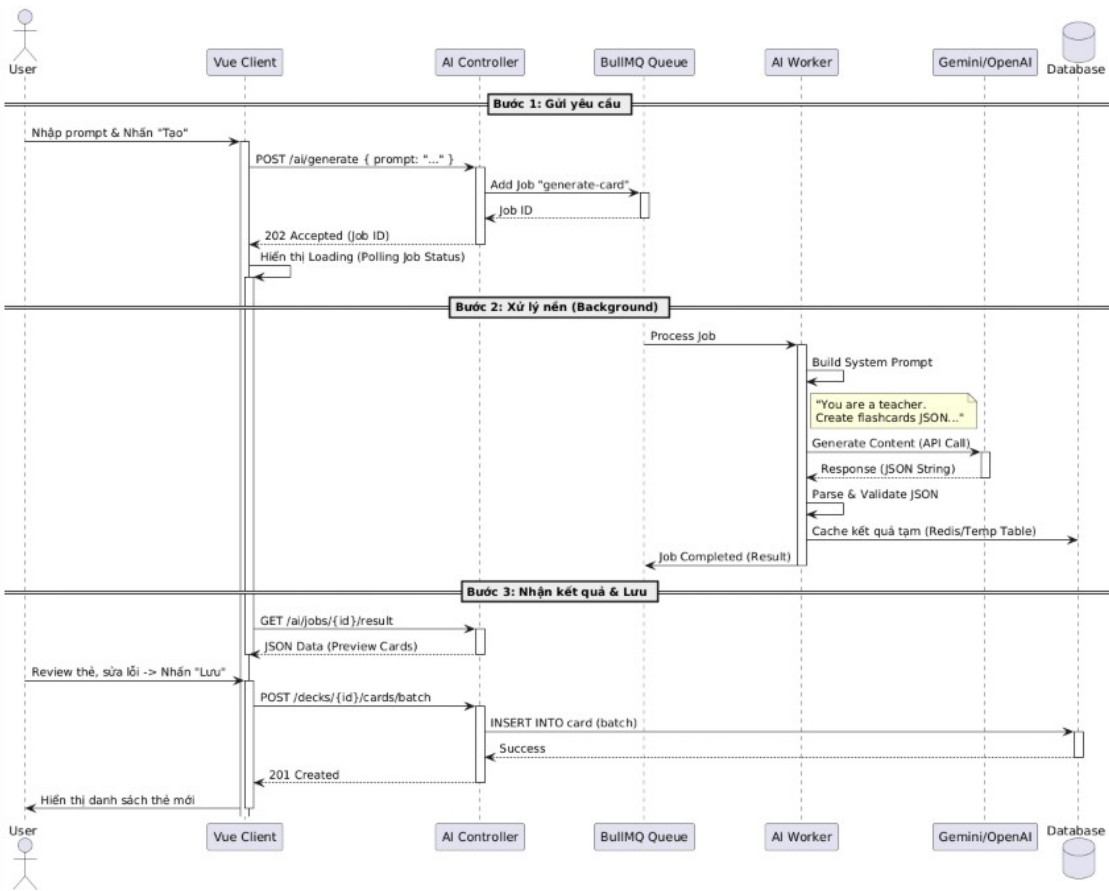
Hình 1. Biểu đồ tuần tự chức năng đăng nhập bằng google

2.6.2 Biểu đồ tuần tự chức năng quản lý thông tin cá nhân



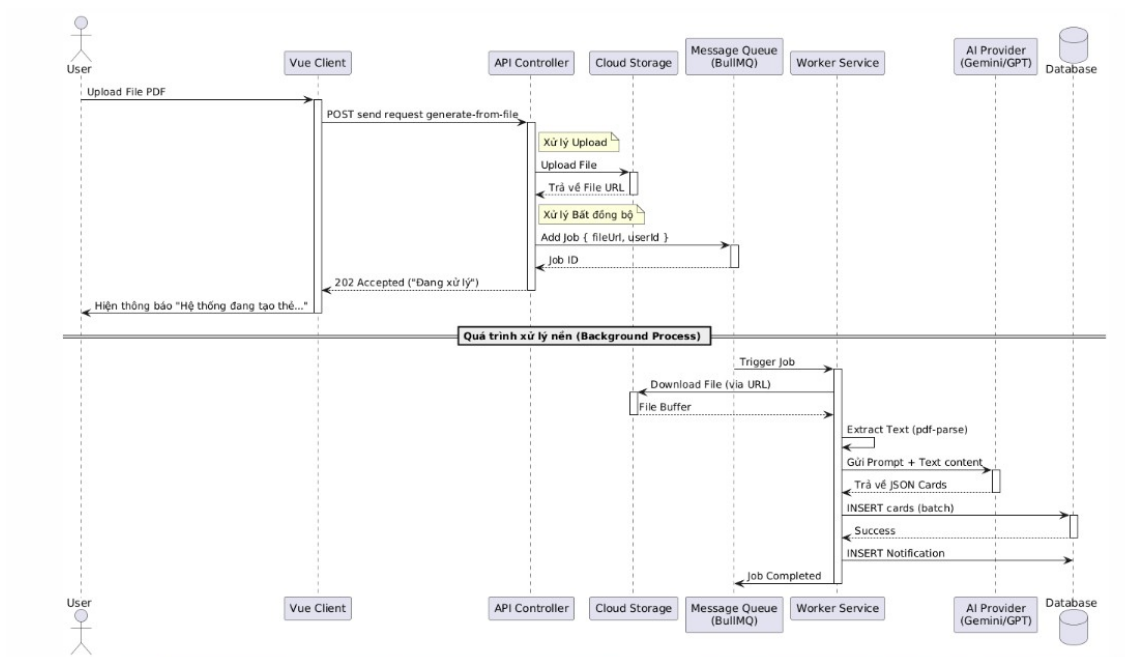
Hình 1. Biểu đồ tuần tự chức năng quản lý thông tin cá nhân

2.6.3 Biểu đồ tuần tự chức năng tạo bộ thẻ từ prompt



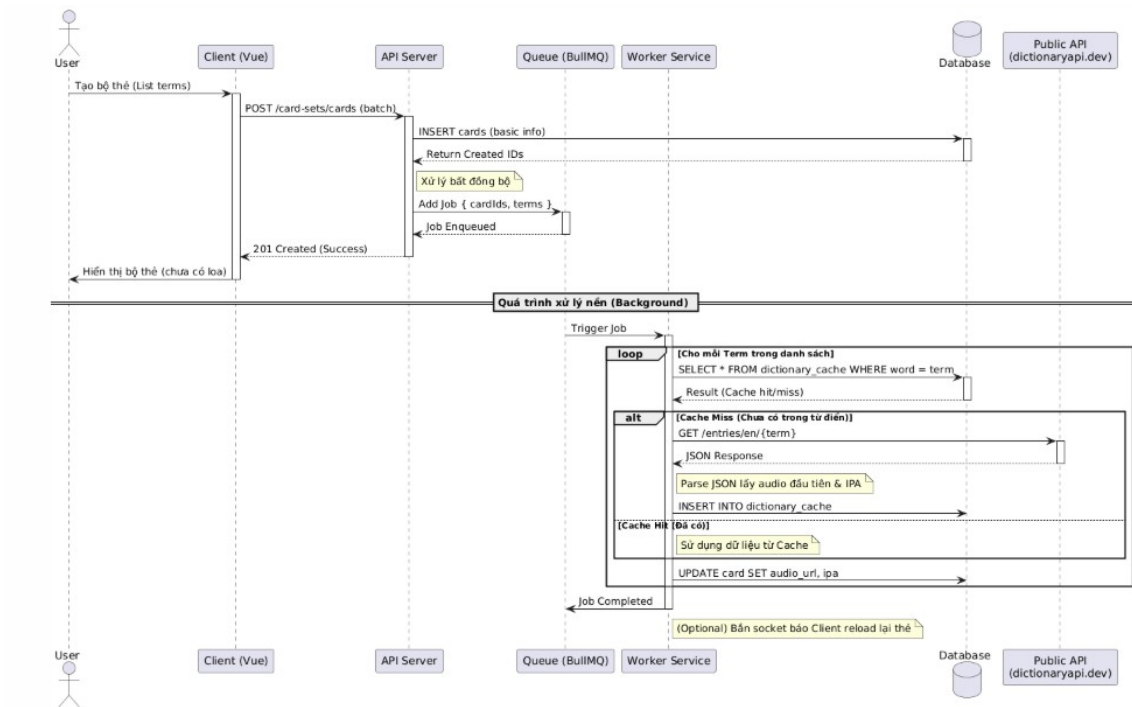
Hình 1. Biểu đồ tuần tự chức năng tạo thẻ từ prompt

2.6.4 Biểu đồ tuần tự chức năng tạo bộ thẻ từ file pdf



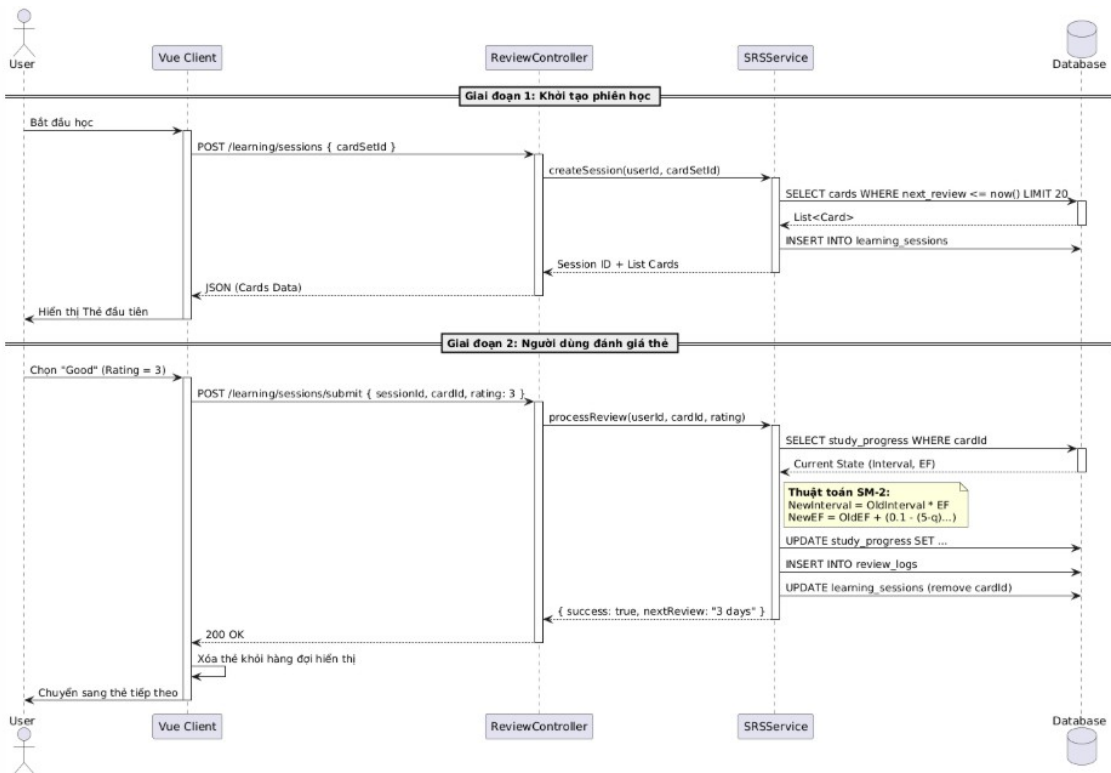
Hình 1. Biểu đồ tuần tự chức năng tạo bộ thẻ từ file pdf

2.6.5 Biểu đồ tuần tự của chức năng tự động bổ sung phát âm (Text-to-Speech Enrichment)



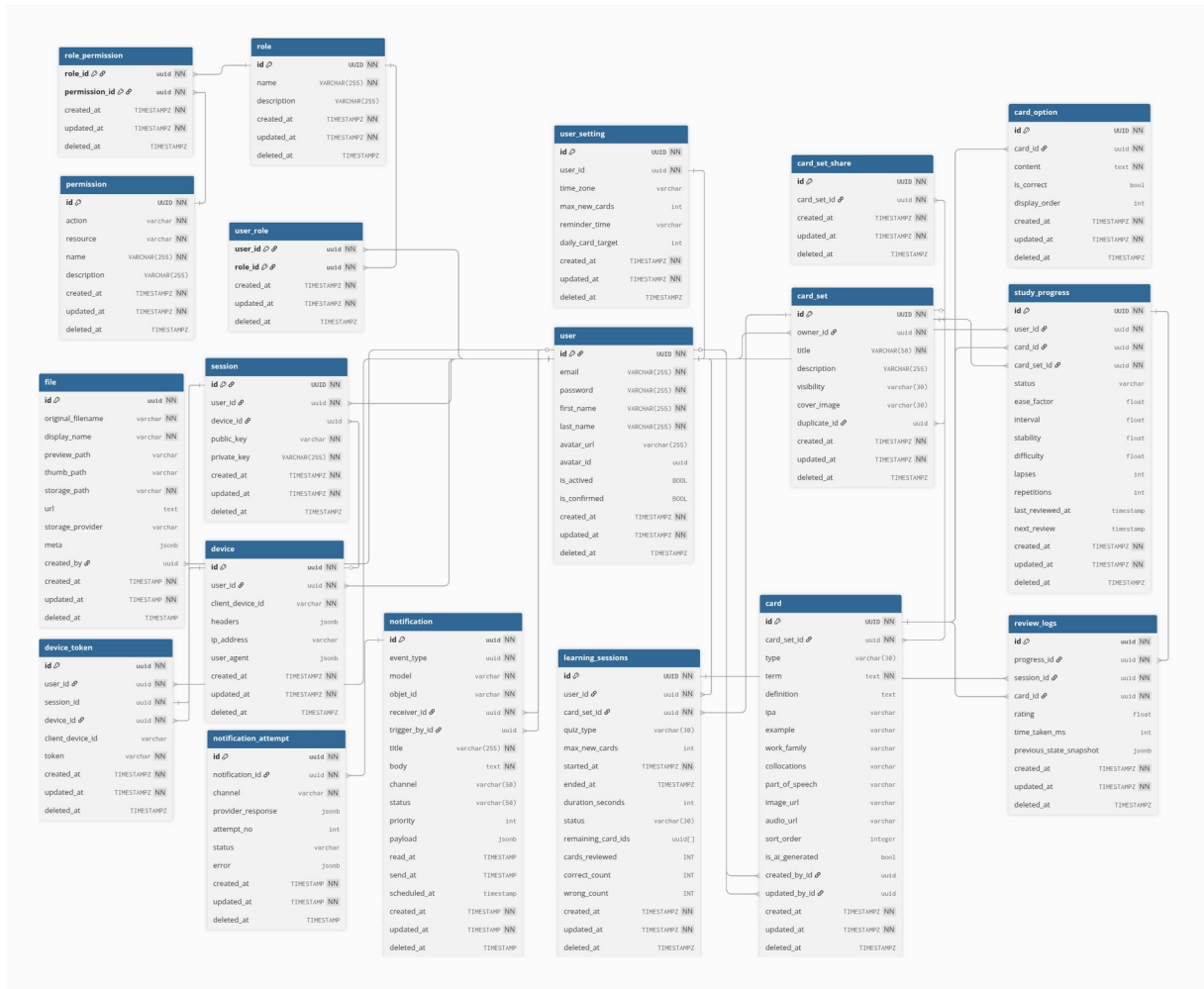
Hình 1. Biểu đồ tuần tự chức năng tự động bổ sung phát âm

2.6.6 Biểu đồ hoạt động của chức năng học flashcard



Hình 1. Biểu đồ tuần tự chức năng học flashcard

2.7 Cơ sở dữ liệu



Hình 1. Lược đồ cơ sở dữ liệu

2.7.1 Bảng Người dùng (user)

Lưu trữ thông tin định danh và trạng thái tài khoản của người dùng trong hệ thống.

Bảng 13. Bảng Users

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa Chính, định danh duy nhất của người dùng	Pk, Not Null
email	Varchar(255)	Email	Unique, Not Null
password	Varchar(255)	Mật khẩu đã mã hóa	Not Null
name	Varchar(255)	Tên người dùng	Not Null
avatar_url	Varchar(255)	Ảnh đại diện	
is_activated	Boolean	Trạng thái kích hoạt tài khoản	Default: false
is_confirmed	Boolean	Xác nhận email	Default: false
created_at	Timestampz	Thời điểm tạo tài khoản	Not null
updated_at	Timestampz	Thời điểm cập nhật gần nhất	Not null
deleted_at	Timestampz	Thời điểm xóa (soft delete)	Nullable

2.7.2 Bảng Cài đặt người dùng (*user_setting*)

Lưu trữ các cấu hình cá nhân hóa cho việc học tập.

Bảng 14. Bảng *User_Setting*

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	uuid	Khóa chính	Pk, not null
user_id	uuid	Id người dùng sở hữu cài đặt	Fk liên kết bảng user
time_zone	varchar	Múi giờ của người dùng (vd: 'asia/ho_chi_minh')	Đề tính ngày mới
max_new_cards	int	Số lượng thẻ mới tối đa mỗi ngày	Default: 20
daily_card_target	int	Mục tiêu tổng số thẻ cần ôn mỗi ngày	Default: 10
reminder_time	varchar	Thời gian nhắc nhở học tập hàng ngày (hh:mm)	
created_at	timestampz	Thời điểm tạo	Not null

2.7.3 Bảng Vai trò (role) & Phân quyền (permission)

Hệ thống sử dụng mô hình RBAC (Role-Based Access Control).

a) Bảng Vai trò (role)

Định nghĩa các nhóm quyền hạn trong hệ thống (Ví dụ: Admin, Member, Premium User).

Bảng 15. Bảng Roles

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	uuid	Khóa chính	PK, Not Null
name	varchar(255)	Tên vai trò (VD: 'admin', 'user')	Unique, Not Null
description	varchar(255)	Mô tả chi tiết về vai trò này	Default: NULL
created_at	timestampz	Thời điểm tạo	Not Null
updated_at	timestampz	Thời điểm cập nhật	Not Null
deleted_at	timestampz	Thời điểm xóa mềm	Nullable

b) Bảng Permission

Định nghĩa các hành động cụ thể được phép thực hiện trên tài nguyên hệ thống.

Bảng 16. Bảng Permission

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK, Not Null
name	Varchar(255)	Tên hiển thị của quyền hạn	Not Null
action	Varchar	Hành động cho phép (VD: 'create', 'read', 'update', 'delete')	Not Null
resource	Varchar	Tài nguyên hệ thống bị tác động (VD: 'card_set', 'user_profile')	Not Null

description	Varchar(255)	Mô tả chi tiết quyền hạn	Default: NULL
-------------	--------------	--------------------------	---------------

2.7.4 Bảng Phiên đăng nhập (session) & Thiết bị (device)

Quản lý bảo mật và phiên làm việc đa thiết bị.

a) Bảng Thiết bị (device)

Lưu trữ thông tin nhận dạng của các thiết bị mà người dùng đã từng đăng nhập. Giúp hệ thống phát hiện đăng nhập lạ và quản lý danh sách thiết bị tin cậy.

Bảng 17. Bảng Device

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK, Not Null
user_id	Uuid	ID người dùng	FK liên kết user
client_device_id	Varchar	Mã định danh duy nhất của thiết bị	Not Null
ip_address	Varchar	Địa chỉ IP	Nullable
user_agent	Jsonb	Thông tin chi tiết về trình duyệt/OS	Nullable
headers	Jsonb	Các thông tin header HTTP	Nullable
created_at	Timestampz	Thời điểm thiết bị được ghi nhận lần đầu	Not Null
updated_at	Timestampz	Thời điểm cập nhật thông tin thiết bị	Not Null

b) Bảng Phiên làm việc (session)

Quản lý trạng thái đăng nhập hiện hành. Mỗi khi người dùng đăng nhập thành công, một session mới được tạo ra.

Bảng 18. Bảng Session

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính - là session_id trong JWT	PK, Not Null
user_id	Uuid	ID người dùng	FK liên kết user
device_id	Uuid	ID thiết bị	FK liên kết device
public_key	Varchar	Khóa công khai	Not Null
private_key	Varchar(255)	Khóa bí mật	Not Null
created_at	Timestampz	Thời điểm bắt đầu phiên	Not Null
updated_at	Timestampz	Thời điểm làm mới phiên gần nhất	Not Null
deleted_at	Timestampz	Thời điểm đăng xuất (thu hồi phiên)	Nullable

2.7.5 Bảng Token thiết bị (device_token)

Bảng này đóng vai trò quan trọng trong hệ thống thông báo đẩy (Push Notification). Nó lưu trữ các mã định danh (Token) được cấp bởi các dịch vụ như Firebase Cloud Messaging (FCM) hoặc Apple Push Notification service (APNs).

Mục đích chính là để hệ thống Backend có thể gửi thông báo đến đúng thiết bị của người dùng ngay cả khi ứng dụng đang chạy ngầm hoặc đã tắt. Đặc biệt, bảng này liên kết chặt chẽ với session để đảm bảo khi người dùng đăng xuất (Logout), token tương ứng sẽ bị vô hiệu hóa, tránh việc gửi thông báo nhầm đến thiết bị cũ.

Bảng 19. Bảng Device_Token

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK, Not Null
user_id	Uuid	Định danh người dùng	FK liên kết user
session_id	Uuid	Liên kết với phiên đăng nhập hiện tại.	FK liên kết session
device_id	Uuid	Liên kết với thông tin thiết bị vật lý	FK liên kết device
client_device_id	Varchar	Mã định danh thiết bị từ phía Client	Nullable
token	Varchar	FCM token	Not Null
created_at	Timestampz	Thời điểm token được lưu vào hệ thống	Not Null
updated_at	Timestampz	Thời điểm cập nhật	Not Null
deleted_at	Timestampz	Thời điểm xóa mềm	Nullable

2.7.6 Bảng Bộ thẻ (card_set)

Quản lý các bộ sưu tập thẻ (Deck).

Bảng 20. Bảng Card_Set

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
owner_id	Uuid	Người tạo bộ thẻ	FK liên kết user
title	Varchar(50)	Tiêu đề bộ thẻ	Not Null
description	Varchar(255)	Mô tả ngắn gọn	
visibility	Varchar(30)	Quyền riêng tư ('public', 'private')	Default: 'private'
duplicate_id	Uuid	ID bộ thẻ gốc (nếu là bản sao chép)	Self-referencing FK
cover_image	Varchar(30)	URL ảnh bìa bộ thẻ	

2.7.7 Bảng Thẻ ghi nhớ (card)

Lưu trữ nội dung kiến thức của từng thẻ.

Bảng 21. Bảng Card

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
card_set_id	Uuid	Thuộc về bộ thẻ nào	FK liên kết card_set
type	Varchar(30)	Loại thẻ ('basic', 'quiz', 'fill_blank')	Default: 'basic'
term	Text	Thuật ngữ / Câu hỏi (Mặt trước)	Not Null
definition	Text	Định nghĩa / Đáp án (Mặt sau)	
ipa	Varchar	Phiên âm quốc tế (cho học ngoại ngữ)	
image_url	Varchar	URL hình ảnh minh họa	
audio_url	Varchar	URL âm thanh phát âm	
is_ai_generated	Boolean	Đánh dấu thẻ do AI tạo tự động	Default: false
sort_order	Integer	Thứ tự sắp xếp trong bộ thẻ	

2.7.8 Bảng Lựa chọn trả lời (card_option)

Lưu trữ các đáp án cho loại thẻ trắc nghiệm (Quiz).

Bảng 22. Bảng Card Option

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
card_id	Uuid	Thuộc về thẻ nào	FK liên kết card
content	Text	Nội dung lựa chọn	Not Null
is_correct	Boolean	Đánh dấu là đáp án đúng	Default: false
display_order	Integer	Thứ tự hiển thị lựa chọn	

2.7.9 Bảng Tiến độ học tập (*study_progress*)

Bảng quan trọng nhất để lưu trữ các tham số thuật toán SRS (SM-2/FSRS).

Bảng 23. Bảng Study Progress

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
user_id	Uuid	ID người học	FK
card_id	Uuid	ID thẻ đang học	FK
status	Varchar	Trạng thái ('new', 'learning', 'review', 'relearning')	Default: 'new'
ease_factor	Float	Hệ số dễ nhớ (quyết định độ giãn cách)	Default: 2.5
interval	Float	Khoảng thời gian (ngày) đến lần ôn tiếp theo	Default: 0
next_review	Timestamp	Thời điểm chính xác cần ôn tập lại	Index cho cột này
lapses	Int	Số lần quên thẻ (đánh giá 'Again' khi đang review)	Dùng để lọc Leech
repetitions	Int	Tổng số lần đã ôn tập thẻ này	
stability	Float	Độ bền trí nhớ (Dành cho thuật toán FSRS)	Optional
difficulty	Float	Độ khó của thẻ (Dành cho thuật toán FSRS)	Optional

2.7.10 Bảng Phiên học (*learning_sessions*)

Lưu trữ thông tin về một lần học cụ thể (Session).

Bảng 24. Bảng Learning Session

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
user_id	Uuid	Người thực hiện phiên học	FK
card_set_id	Uuid	Bộ thẻ đang học	FK
started_at	Timestampz	Thời gian bắt đầu	Not Null
ended_at	Timestampz	Thời gian kết thúc (Null nếu đang học)	
status	Varchar(30)	Trạng thái ('in_progress', 'completed')	
remaining_card_ids	Uuid[]	Mảng chứa ID các thẻ còn lại trong hàng đợi	Kiểu Array PostgreSQL
duration_seconds	Int	Tổng thời gian học thực tế (giây)	
cards_reviewed	Int	Số lượng thẻ đã ôn trong phiên	

2.7.11 Bảng Lịch sử ôn tập (review_logs)

Ghi lại chi tiết từng lần lật thẻ để phân tích và thống kê.

Bảng 25. Bảng Review Logs

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
progress_id	Uuid	Liên kết với tiến độ học tập	FK
session_id	Uuid	Liên kết với phiên học	FK
rating	Float	Điểm đánh giá (1: Again, 2: Hard, 3: Good, 4: Easy)	
time_taken_ms	Int	Thời gian suy nghĩ trước khi lật thẻ (ms)	
previous_state	Jsonb	Snapshot trạng thái (Interval, EF) trước khi ôn	Để debug thuật toán
created_at	Timestampz	Thời điểm thực hiện ôn tập	Not Null

2.7.12 Bảng Thông báo (notification)

Quản lý hệ thống thông báo gửi đến người dùng.

Bảng 26. Bảng Notification

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
receiver_id	Uuid	Người nhận thông báo	FK
title	Varchar(255)	Tiêu đề thông báo	Not Null
body	Text	Nội dung thông báo	Not Null
channel	Varchar(50)	Kênh gửi ('in_app', 'email', 'push')	Default: 'in_app'
status	Varchar(50)	Trạng thái ('pending', 'sent', 'read', 'failed')	
payload	Jsonb	Dữ liệu đính kèm (để điều hướng khi click)	
read_at	Timestamp	Thời điểm người dùng đã đọc	

2.7.13 Bảng Lịch sử gửi thông báo (*notification_attempt*)

Bảng này lưu trữ chi tiết lịch sử của từng lần hệ thống thực hiện gửi một thông báo. Trong thực tế, việc gửi thông báo (qua Email hoặc Push Notification) thường phụ thuộc vào các dịch vụ bên thứ ba (như Firebase, SendGrid, Amazon SES) và có thể thất bại do lỗi mạng hoặc quá tải.

Bảng *notification_attempt* giúp hệ thống:

1. **Truy vết (Traceability):** Biết chính xác thông báo đã được gửi đi lúc nào, qua kênh nào.
2. **Cơ chế Retry (Thử lại):** Nếu lần gửi 1 thất bại, hệ thống sẽ tạo một bản ghi mới cho lần thử 2, 3... thay vì ghi đè dữ liệu cũ.
3. **Debug lỗi:** Lưu lại phản hồi chi tiết từ nhà cung cấp để lập trình viên sửa lỗi.

Bảng 27. Bảng *Notification Attempt*

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK, Not Null
notification_id	Uuid	Liên kết với thông báo gốc cần gửi	FK liên kết notification
channel	Varchar	Kênh thực hiện gửi trong lần thử này (VD: 'fcm', 'email_smtp')	Not Null
attempt_no	Int	Số thứ tự lần thử (1, 2, 3...)	Default: 1
status	Varchar	Trạng thái của lần thử này ('success', 'failed', 'sending')	
provider_response	Jsonb	Dữ liệu phản hồi nguyên bản từ dịch vụ bên thứ 3 (Provider)	Lưu ID gửi của Firebase/AWS để tra cứu
error	Jsonb	Chi tiết lỗi nếu gửi thất bại (Mã lỗi,	Dùng để debug

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
		thông điệp lỗi)	
created_at	Timestamp	Thời điểm bắt đầu thực hiện gửi	Not Null
updated_at	Timestamp	Thời điểm nhận được kết quả gửi	Not Null
deleted_at	Timestamp	Thời điểm xóa (ít dùng)	Nullable

2.7.14 Bảng Quản lý File (file)

Lưu trữ metadata của các tệp tin đã tải lên hệ thống (Cloudinary/S3).

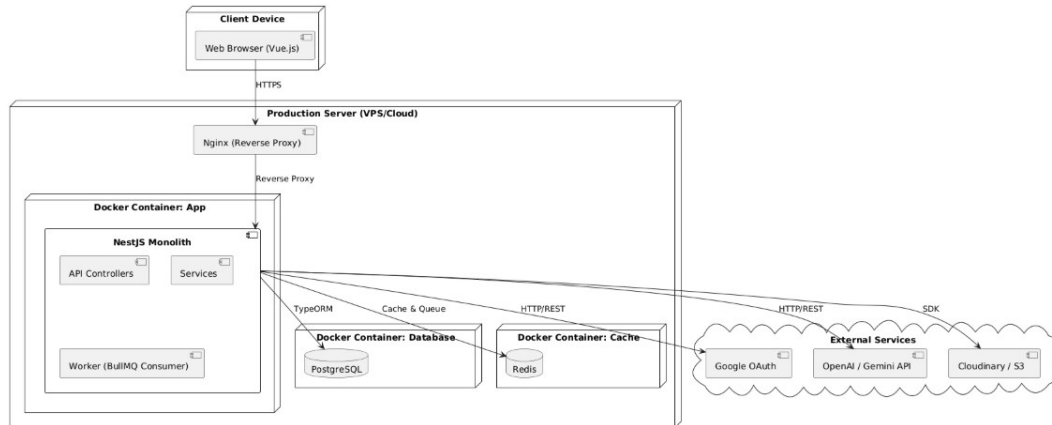
Bảng 28. Bảng Files

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Uuid	Khóa chính	PK
display_name	Varchar	Tên hiển thị của file	Not Null
url	Text	Đường dẫn truy cập file (public url)	
storage_path	Varchar	Đường dẫn lưu trữ trên cloud/disk	
storage_provider	Varchar	Nhà cung cấp ('cloudinary', 's3', 'local')	
created_by	Uuid	Người tải lên	FK
meta	Jsonb	Metadata (size, width, height, format)	

CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG

3.1 Triển khai hệ thống

3.1.1 Kiến trúc triển khai (Deployment Architecture)



Hình 1. Kiến trúc triển khai hệ thống

Hệ thống được xây dựng theo mô hình **Client-Server** truyền thống với kiến trúc Backend dạng **Monolithic** (Nguyên khối). Toàn bộ logic nghiệp vụ, kết nối cơ sở dữ liệu và xử lý tác vụ nền đều được đóng gói trong một ứng dụng duy nhất (Single Application) chạy trên Server.

Mô hình triển khai vật lý bao gồm các thành phần sau:

1. **Client (Frontend):** Ứng dụng Web (Single Page Application - SPA) được xây dựng bằng Vue.js, chạy trên trình duyệt của người dùng. Client giao tiếp với Server thông qua RESTful API.
2. **Web Server / Reverse Proxy (Nginx):** Đứng trước ứng dụng Backend, chịu trách nhiệm tiếp nhận request, cấu hình SSL/HTTPS, nén dữ liệu (Gzip) và phục vụ các file tĩnh (Static files).
3. **Application Server (Backend):**
 - Chạy ứng dụng NestJS Monolith.
 - Bên trong ứng dụng được chia thành các Module (Modular Monolith) để đảm bảo tính tổ chức code: AuthModule, CardModule, AIModule, StudyModule.
 - Sử dụng **BullMQ** (thư viện Queue) tích hợp ngay trong ứng dụng để xử lý các tác vụ nặng (AI Generation, Email) dưới dạng Background Process.

4. Data Persistence Layer:

- **PostgreSQL:** Lưu trữ dữ liệu quan hệ (User, Card, Study Progress).
- **Redis:** Đóng vai trò làm bộ nhớ đệm (Cache) và Broker cho hàng đợi (Message Queue).

5. **External Services:** Tương tác với Google (OAuth), Cloudinary (Lưu trữ ảnh), OpenAI/Gemini (AI API).

3.1.2 Môi trường và công nghệ

Để đảm bảo sự nhất quán giữa môi trường phát triển (Development) và môi trường vận hành (Production), nhóm sử dụng công nghệ Containerization.

- **Docker & Docker Compose:** Đóng gói toàn bộ ứng dụng Backend, Database và Redis vào các container. File docker-compose.yml được sử dụng để khởi chạy toàn bộ hệ thống chỉ với một câu lệnh.
- **Hệ điều hành Server:** Ubuntu Linux 22.04 LTS.
- **Node.js Runtime:** Phiên bản 20.x (LTS) để đảm bảo hiệu năng và ổn định.
- **Quản lý biến môi trường:** Sử dụng file .env để lưu trữ các thông tin nhạy cảm (DB Password, API Keys) và không đưa lên mã nguồn (Git).

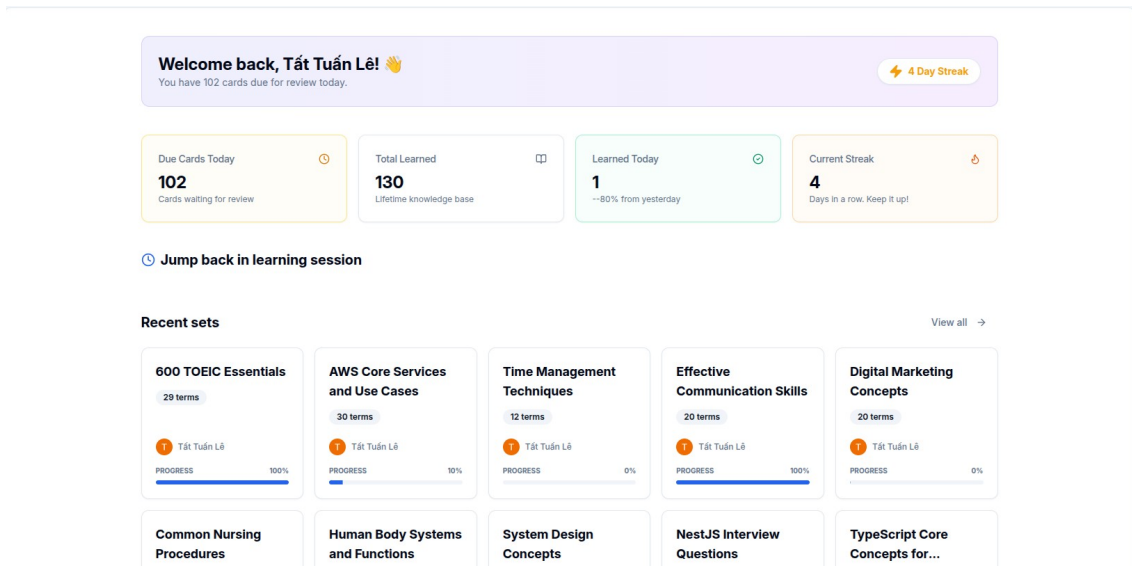
3.1.3 Quy trình CI/CD (Tích hợp và Triển khai liên tục)

- **Source Control:** GitHub.
- **CI/CD Tool:** GitHub Actions.
- **Quy trình:**
 1. **Push:** Lập trình viên đẩy code lên nhánh main.
 2. **Build:** GitHub Actions tự động build Docker Image mới nhất của Backend.
 3. **Deploy:** GitHub Actions SSH vào Server, thực hiện lệnh docker compose pull và docker compose up -d để khởi động lại ứng dụng với code mới nhất mà không cần can thiệp thủ công.

3.2 Kết quả giao diện chương trình

3.2.1 Màn hình đăng nhập & dash board

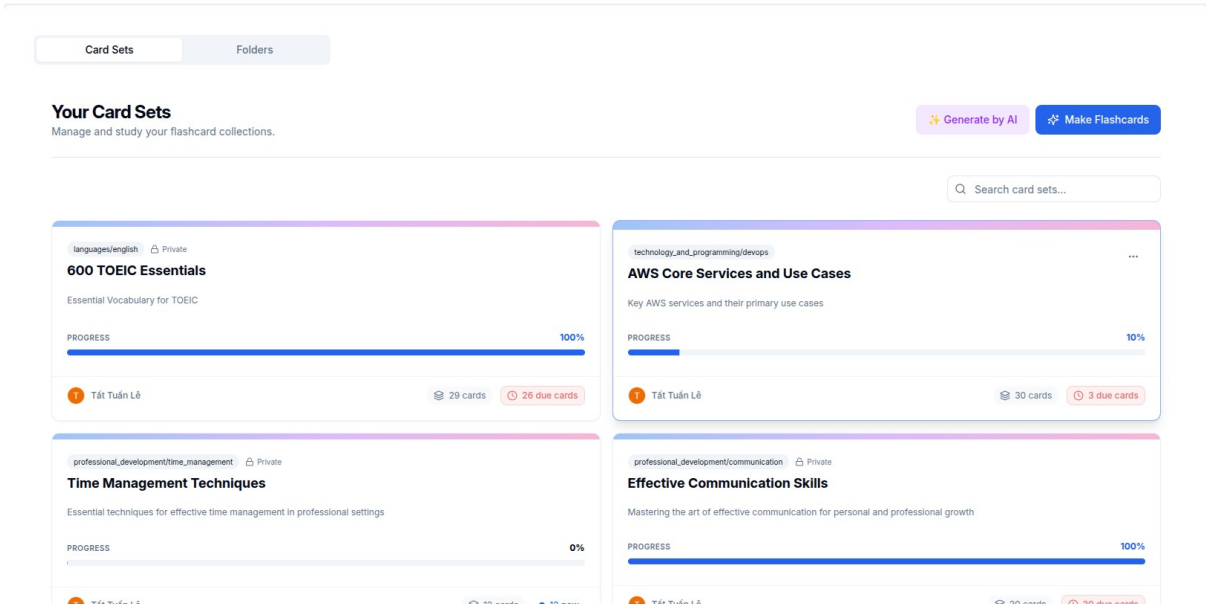
- **Đăng nhập:** Hỗ trợ đăng nhập qua Google OAuth 2.0. Giao diện tối giản, tập trung vào trải nghiệm người dùng.
- **Dashboard:** Hiện thị tổng quan tiến độ học tập, bao gồm biểu đồ Heatmap (trưng tự Github) thể hiện tần suất học và danh sách các bộ thẻ cần ôn tập trong ngày.



Hình 1. Giao diện dashboard

3.2.2 Màn hình Quản lý Bộ thẻ

Cho phép người dùng tạo bộ thẻ mới, chỉnh sửa thông tin mô tả, cài đặt quyền riêng tư (Public/Private). Danh sách các thẻ được hiển thị dạng bảng hoặc lưới, hỗ trợ tìm kiếm và lọc nhanh.



Hình 1. Giao diện quản lý bộ thẻ

3.2.3 Màn hình Tạo thẻ bằng AI

Giao diện Modal cho phép người dùng nhập Prompt hoặc tải lên file PDF. Hệ thống hiển thị trạng thái "Đang xử lý" (Loading) trong khi Backend thực hiện Background Job. Sau khi hoàn tất, danh sách thẻ nhập được hiển thị để người dùng xem trước (Preview) trước khi lưu.

Generate Card Set ✕

Generate a card set from AI powered prompts. Use your credits to generate images, audio, and text.

Your credits: 👉 100 👉 0

Category

Select Category ⌵

Sub Category

Select sub Category ⌵

Prompt

Describe what you want to generate

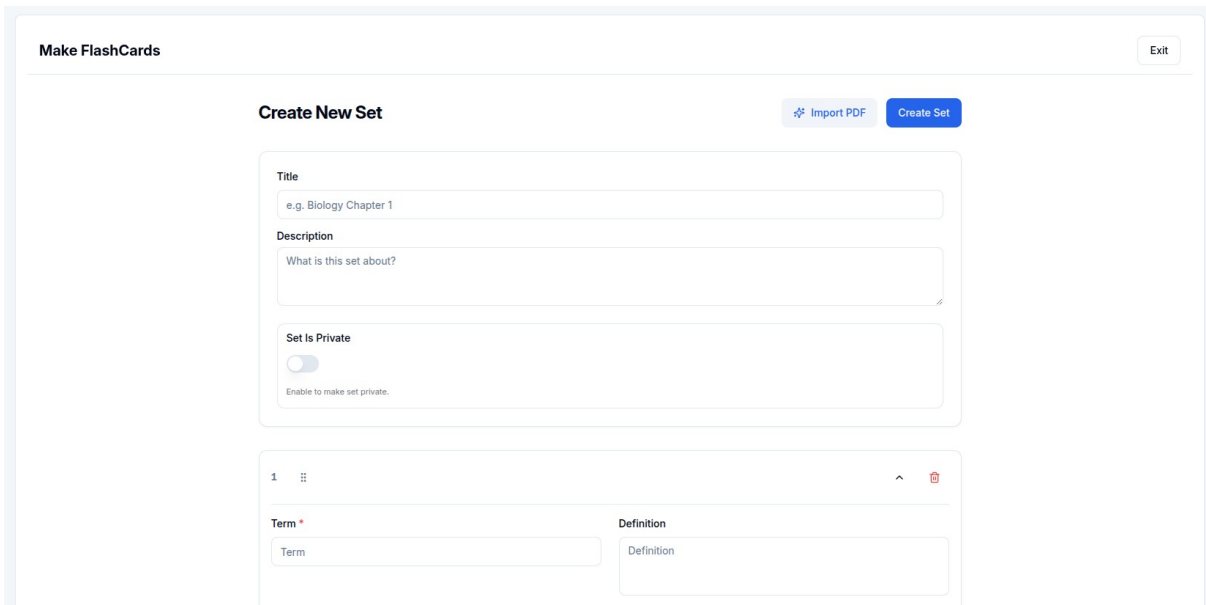
Card Count

20

Cancel 20 🎧 Generate

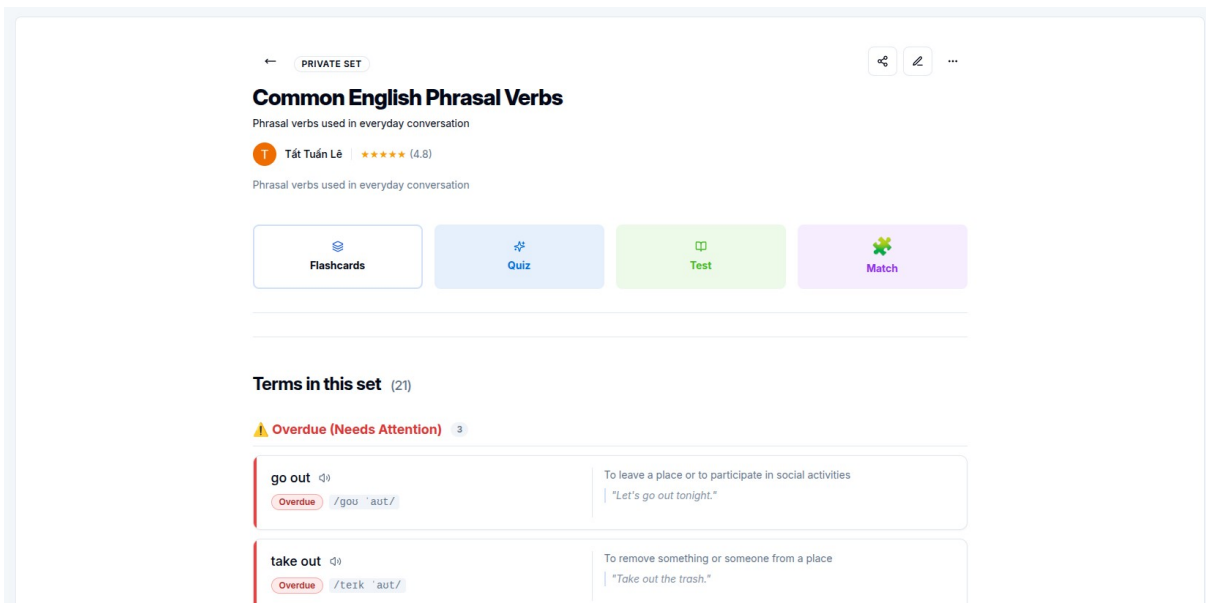
Hình 1. Giao diện tạo thẻ bằng AI

3.2.4 Màn tạo flashcard từ file pdf



Hình 1. Màn hình tạo flashcard từ file pdf

3.2.5 Màn hình Xem chi tiết bộ thẻ



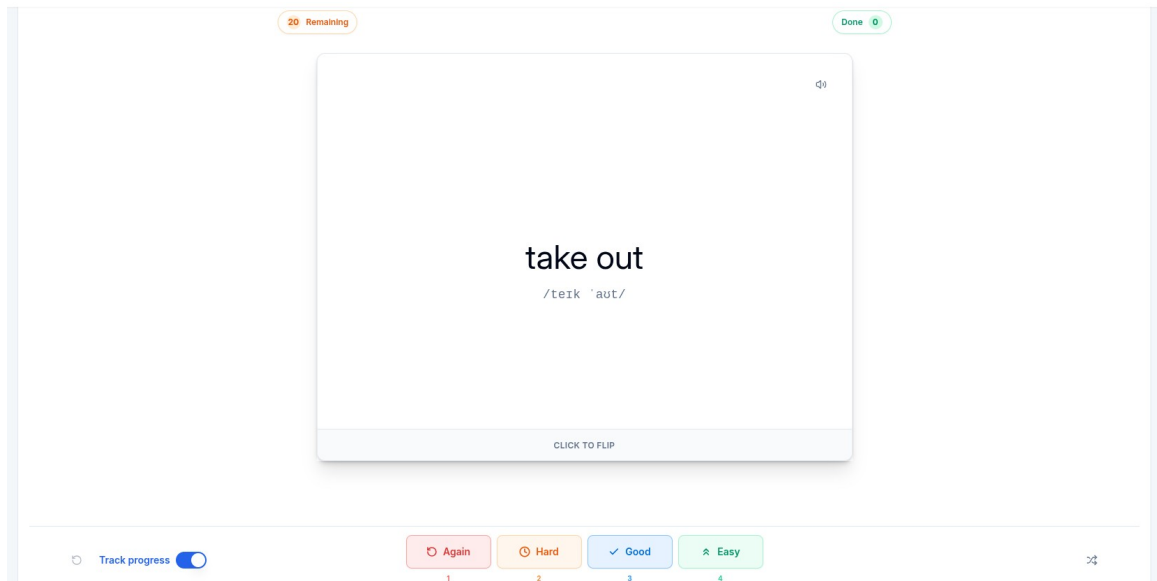
Hình 1. Giao diện thống kê tiến độ học tập theo bộ thẻ

3.2.6 Màn hình Ôn tập (Flashcard Study)

Giao diện học tập tập trung (Distraction-free).

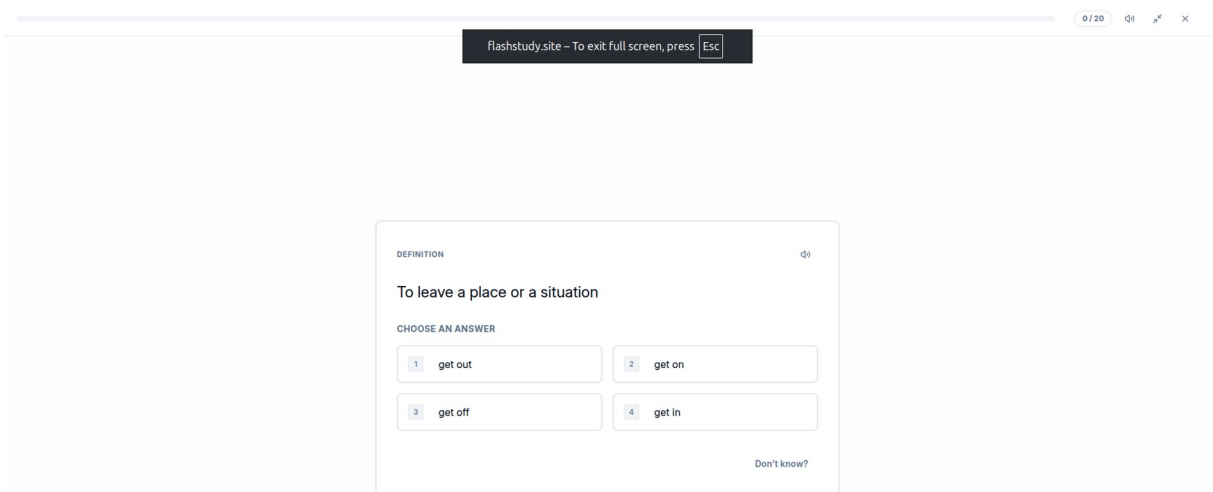
- **Mặt trước:** Hiện thị câu hỏi/thuật ngữ.
- **Mặt sau:** Hiện thị đáp án sau khi người dùng tương tác.

- **Thanh đánh giá:** 4 nút (Again, Hard, Good, Easy) tương ứng với thuật toán SRS.



Hình 1. Giao diện học flashcard

3.2.7 Màn hình Quiz mode



Hình 1. Giao diện học theo chế độ quiz

3.2.8 Màn hình cấu hình cho bài test

Create Test Session ×

Customize your test mode, time limit, and question types.

Question Count / 21 Max available cards

Timer (Minutes) Leave empty for no time limit

Shuffle Questions
Randomize question order

Question Types Default: Auto Balance

Multiple Choice

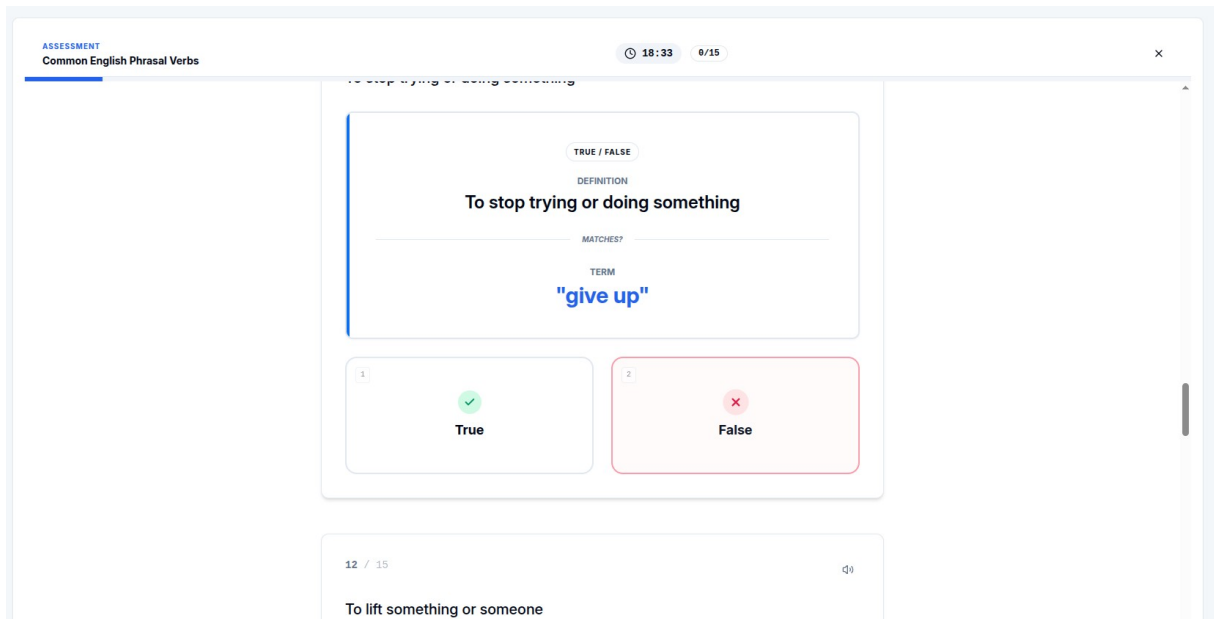
RATIO (0-1) LIMIT

True / False

RATIO (0-1) LIMIT

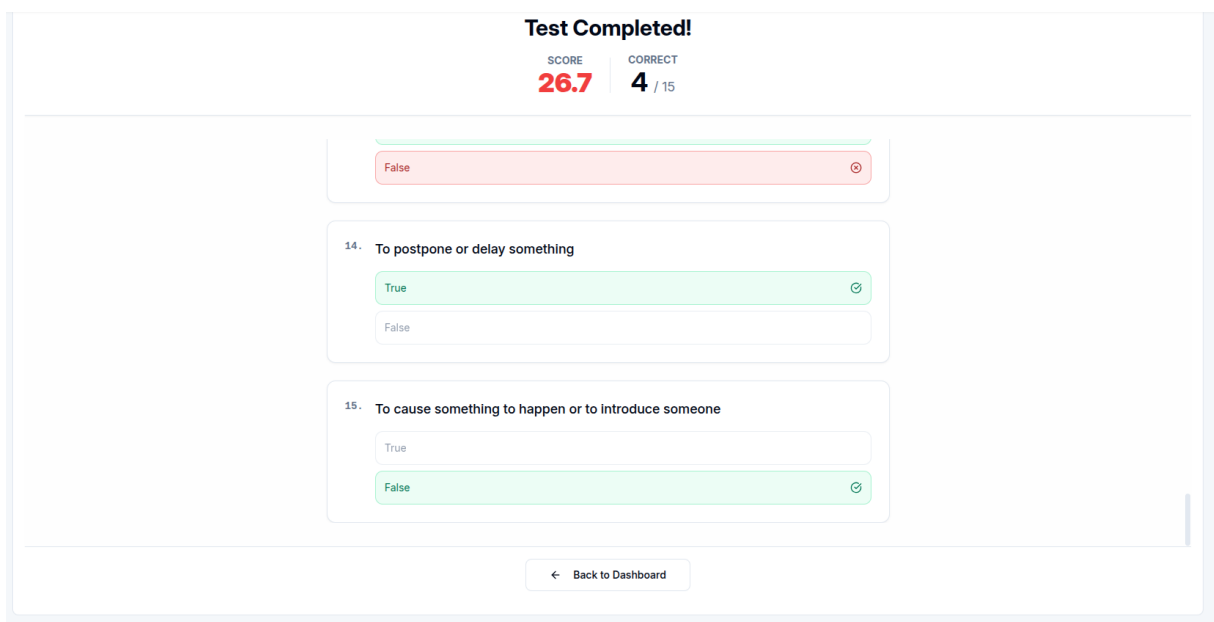
Hình 1. Giao diện cài đặt bài test

3.2.9 Màn hình làm bài test



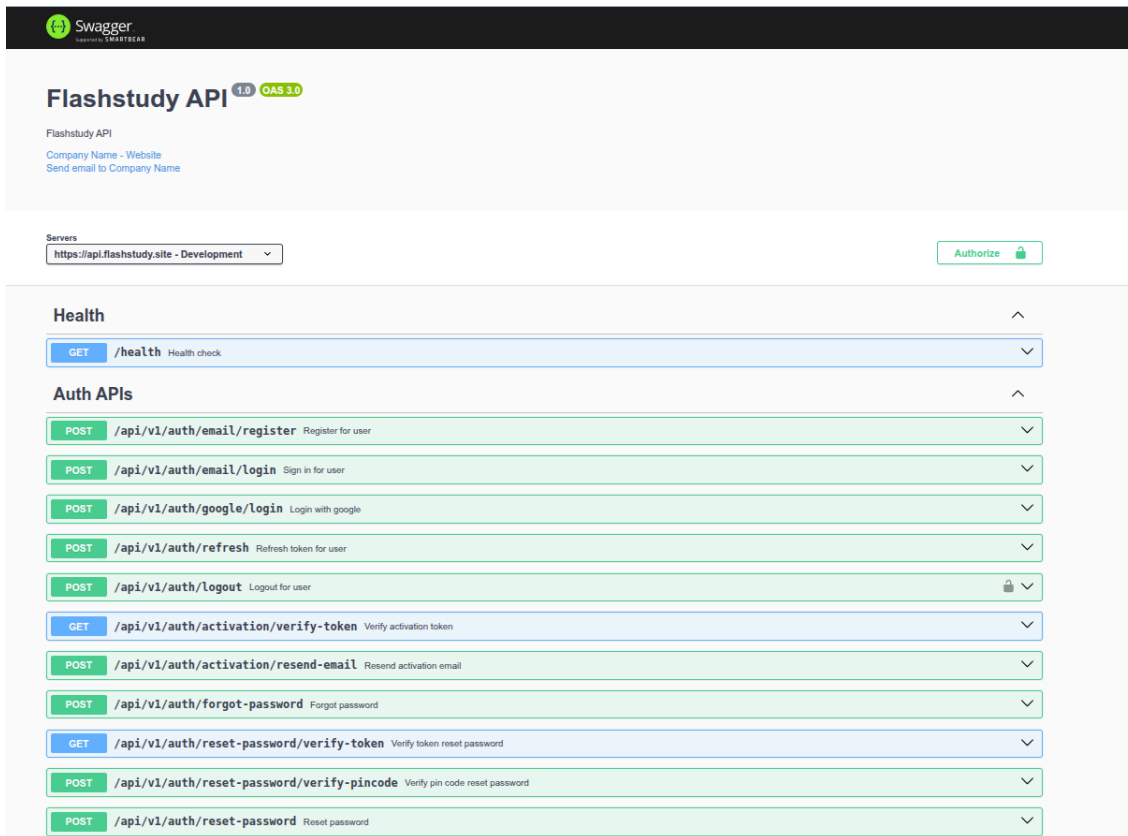
Hình 1. Giao diện làm bài test

3.2.10 Màn hình kết quả bài test



Hình 1. Giao diện kết quả bài test

3.3 Tài liệu API cho lập trình viên



Hình 1. Api Docs - Swagger

3.4 Kiểm thử hệ thống (Testing)

Quá trình kiểm thử được thực hiện theo phương pháp kiểm thử hộp đen (Black-box testing), tập trung vào việc xác minh các chức năng có hoạt động đúng theo đặc tả yêu cầu hay không.

3.4.1 Kịch bản kiểm thử: Đăng nhập Google

Bảng 29. Bảng kiểm thử chức năng đăng nhập bằng google

ID	Tên kịch bản	Các bước thực hiện	Kết quả mong đợi	Kết quả thực tế	Trạng thái
T C- 01	Đăng nhập thành công	<ol style="list-style-type: none"> Tại màn hình Login, nhấn "Tiếp tục với Google". Chọn tài khoản Google hợp lệ. Xác nhận quyền truy cập. 	Hệ thống chuyển hướng về Dashboard, lưu Token và hiển thị đúng tên/avatar người dùng.	Đúng như mong đợi	Đạt

3.4.2 Kịch bản kiểm thử: Tạo thẻ bằng AI

Bảng 30. Bảng kiểm thử chức năng tạo thẻ bằng AI

ID	Tên kịch bản	Các bước thực hiện	Kết quả mong đợi	Kết quả thực tế	Trạng thái
T C- 05	Tạo thẻ từ Prompt văn bản	<ol style="list-style-type: none"> Mở Modal "Tạo bằng AI". Nhập: "10 từ vựng chủ đề IT". Nhấn "Tạo". 	Hệ thống hiển thị Loading. Sau 5-10s, hiển thị danh sách 10 thẻ đúng chủ đề IT.	Đúng như mong đợi	Đạt
T C- 06	Xử lý khi AI quá tải	<ol style="list-style-type: none"> Ngắt kết nối mạng Backend với OpenAI. Thực hiện lại bước trên. 	Hệ thống hiển thị thông báo lỗi: "Hệ thống AI đang bận, vui lòng thử lại sau".	Đúng như mong đợi	Đạt

3.4.3 Kịch bản kiểm thử: Thuật toán SRS

Bảng 31. Bảng kiểm thử chức năng học flashcard và thuật toán SRS

ID	Tên kịch bản	Các bước thực hiện	Kết quả mong đợi	Kết quả thực tế	Trạng thái
T C- 10	Đánh giá thẻ "Good"	<ol style="list-style-type: none"> Mở phiên học, lật thẻ. Nhấn nút "Good". 	Thẻ biến mất khỏi hàng đợi hiện tại. Ngày ôn tập tiếp theo (next_review) được cập nhật > 1 ngày.	Đúng như mong đợi	Đạt
T C- 11	Đánh giá thẻ "Again"	<ol style="list-style-type: none"> Mở phiên học, lật thẻ. Nhấn nút "Again". 	Thẻ được đẩy xuống cuối hàng đợi để học lại ngay. Interval reset về < 1 phút.	Đúng như mong đợi	Đạt

3.5 Đánh giá chất lượng hệ thống

3.5.1 Đánh giá về mặt chức năng (Functional Assessment)

Hệ thống đã đáp ứng đầy đủ các yêu cầu cốt lõi được đặt ra trong giai đoạn phân tích:

- **Tính chính xác của thuật toán SRS:** Qua thử nghiệm thực tế với dữ liệu mô phỏng trong 30 ngày, thuật toán SM-2 hoạt động chính xác. Các thẻ được đánh giá "Again" xuất hiện lại ngay trong phiên học, trong khi các thẻ "Good/Easy" được giãn cách thời gian hợp lý, giúp tối ưu hóa khối lượng kiến thức cần ôn tập.
- **Hiệu quả của AI sinh nội dung:**
 - Tính năng tạo thẻ từ văn bản (Prompt) hoạt động ổn định với độ chính xác về mặt ngữ nghĩa đạt khoảng 85-90%.
 - Hệ thống trích xuất tốt các cặp "Thuật ngữ - Định nghĩa" từ các đoạn văn bản tiếng Anh và tiếng Việt.
 - **Lưu ý:** Với các tài liệu chuyên ngành hẹp hoặc chứa nhiều công thức toán học phức tạp, AI đôi khi vẫn gặp khó khăn trong việc định dạng thẻ, cần người dùng chỉnh sửa lại thủ công.
- **Quy trình quản lý người dùng:** Các chức năng Đăng ký, Đăng nhập (Google OAuth), và Quản lý phiên làm việc hoạt động trơn tru, bảo mật.

3.5.2 Đánh giá về mặt hiệu năng (Performance Assessment)

Mặc dù sử dụng kiến trúc Monolithic, hệ thống vẫn đảm bảo hiệu năng cao nhờ việc tối ưu hóa code và sử dụng các công nghệ hỗ trợ:

- **Tốc độ phản hồi API:**
 - Các API truy vấn dữ liệu thông thường (Lấy danh sách thẻ, Xem thống kê) có thời gian phản hồi trung bình < **100ms**.
 - Các API ghi dữ liệu (Lưu log ôn tập) được xử lý nhanh gọn nhờ tối ưu hóa câu lệnh SQL và đánh Index hợp lý trong PostgreSQL.
- **Xử lý tác vụ nặng (Background Jobs):**
 - Việc tích hợp **BullMQ (Redis)** đã giải quyết triệt để vấn đề "nghẽn cổ chai" khi gọi AI.
 - Người dùng không bị treo trình duyệt khi tạo thẻ. Thời gian trung bình để AI tạo xong một bộ 10 thẻ là khoảng **10-15 giây**, trong thời gian đó người dùng vẫn có thể thực hiện các thao tác khác bình thường.
- **Khả năng chịu tải:** Với cấu hình Server thử nghiệm (1 vCPU, 2GB RAM), hệ thống có thể chịu được khoảng 200-300 người dùng truy cập đồng thời

(Concurrent users) mà không gặp lỗi quá tải, nhờ cơ chế Reverse Proxy của Nginx và Caching của Redis.

3.5.3 Đánh giá về trải nghiệm người dùng (UX/UI Assessment)

Giao diện được xây dựng bằng Vue.js mang lại trải nghiệm mượt mà của một ứng dụng đơn trang (SPA):

- **Tính tương tác cao:** Các thao tác chuyển trang, lật thẻ diễn ra tức thì, không có độ trễ tải trang (Reload), tạo cảm giác liền mạch ("Flow") giúp người học tập trung hơn.
- **Giao diện đáp ứng (Responsive):** Hệ thống hiển thị tốt trên cả Laptop và Điện thoại di động. Các nút bấm trong chế độ ôn tập được thiết kế to, rõ ràng, thuận tiện cho thao tác chạm trên màn hình cảm ứng.
- **Phản hồi trực quan:** Hệ thống luôn có các thông báo (Toast notification) hoặc trạng thái Loading khi xử lý dữ liệu, giúp người dùng luôn nắm bắt được tình trạng của ứng dụng.

3.6 Kết chương

Chương 4 đã trình bày chi tiết quá trình hiện thực hóa hệ thống "Flashcard Maker". Từ việc thiết lập môi trường triển khai sử dụng Docker, tổ chức mã nguồn khoa học theo kiến trúc Modular Monolith, cho đến việc xây dựng hoàn thiện các chức năng quan trọng như Tích hợp AI và Thuật toán SRS.

Kết quả kiểm thử và đánh giá cho thấy hệ thống hoạt động ổn định, đáp ứng tốt các yêu cầu chức năng và phi chức năng đã đề ra. Các giải pháp kỹ thuật như sử dụng hàng đợi (Queue) cho AI và tối ưu hóa cơ sở dữ liệu đã chứng minh được hiệu quả thực tế.

Những kết quả đạt được trong chương này là cơ sở vững chắc để khẳng định tính khả thi và tiềm năng ứng dụng của đề án, đồng thời mở ra những hướng phát triển nâng cao sẽ được trình bày trong chương cuối cùng.

CHƯƠNG 4: KẾT LUẬN

4.1 Những kết quả đạt được

4.1.1 Về mặt lý thuyết

Trong quá trình thực hiện đồ án, nhóm thực hiện đã tìm hiểu và nắm vững các nền tảng kiến thức quan trọng:

- **Nghiên cứu về thuật toán Lặp lại ngắt quãng (Spaced Repetition System - SRS):** Hiểu rõ cơ chế hoạt động của trí nhớ con người dựa trên "Đường cong quên lãng" (Forgetting Curve). Đã phân tích và cài đặt thành công thuật toán SM-2 (SuperMemo-2), làm cơ sở cốt lõi cho tính năng ôn tập của hệ thống.
- **Ứng dụng Trí tuệ nhân tạo (Generative AI):** Nắm vững kỹ thuật Prompt Engineering để tương tác với các mô hình ngôn ngữ lớn (LLM) như OpenAI/Gemini. Hiểu cách thiết kế System Prompt để yêu cầu AI trả về dữ liệu có cấu trúc (JSON) từ văn bản thô hoặc tài liệu PDF.
- **Kiến trúc phần mềm và Công nghệ:**
 - Vận dụng thành thạo kiến trúc **Monolith** và **Clean Architecture** trong việc tổ chức mã nguồn Backend với NestJS.
 - Hiểu sâu về cơ chế xử lý bất đồng bộ (**Asynchronous Processing**) sử dụng Message Queue (BullMQ) và Redis để giải quyết các bài toán tốn thời gian như xử lý file và gọi API AI.
 - Thiết kế cơ sở dữ liệu quan hệ (PostgreSQL) tối ưu cho việc truy vấn dữ liệu lớn (Lịch sử ôn tập, Logs).
- **Bảo mật hệ thống:** Nghiên cứu và áp dụng mô hình bảo mật hiện đại với JWT (JSON Web Token), cơ chế Refresh Token Rotation và phân quyền RBAC (Role-Based Access Control).

4.1.2 Về mặt ứng dụng

Đồ án đã xây dựng thành công ứng dụng hoàn chỉnh với các chức năng vận hành ổn định trên môi trường Web:

- **Hệ thống tạo thẻ thông minh:**
 - Cho phép tạo thẻ thủ công với đầy đủ đa phương tiện (Hình ảnh, Âm thanh).
 - **Tính năng nổi bật:** Tự động sinh bộ thẻ từ văn bản (Prompt) và từ tài liệu tải lên (PDF/Docx) với tốc độ xử lý nhanh và độ chính xác cao nhờ AI.
- **Môi trường học tập hiệu quả:**
 - Hệ thống tự động tính toán và nhắc nhở lịch học cho từng người dùng dựa trên lịch sử đánh giá của họ.
 - Cung cấp đa dạng chế độ học: Lật thẻ (Flashcard), Trắc nghiệm (Quiz) và Ghép thẻ (Matching Game) giúp giảm sự nhàm chán.
 - Tự động bổ sung phát âm (Text-to-Speech) và phiên âm (IPA) cho các thẻ từ vựng ngoại ngữ.
- **Hệ thống theo dõi và tiện ích:**
 - Dashboard thống kê trực quan với biểu đồ nhiệt (Heatmap) và biểu đồ tiến độ, giúp người học theo dõi chuỗi ngày học (Streak).
 - Hệ thống thông báo đẩy (Push Notification) đa nền tảng giúp duy trì thói quen học tập.
 - Hỗ trợ đăng nhập nhanh qua Google (OAuth 2.0).

4.2 Hạn chế

Mặc dù đã nỗ lực hoàn thiện, đồ án vẫn còn một số hạn chế nhất định cần khắc phục:

- **Độ trễ khi xử lý AI:** Việc sinh thẻ từ file PDF có dung lượng lớn đôi khi vẫn mất nhiều thời gian (trên 30 giây) do phụ thuộc vào tốc độ phản hồi của bên thứ ba (OpenAI/Google API).
- **Hạn chế trong xử lý tài liệu:** Hiện tại hệ thống sử dụng thư viện trích xuất văn bản (Text Extraction) nên chưa xử lý tốt các file PDF dạng scan (chỉ chứa hình ảnh) hoặc các công thức toán học phức tạp.
- **Thuật toán SRS:** Mới chỉ dừng lại ở thuật toán SM-2 cổ điển. Các thuật toán hiện đại hơn như FSRS (Free Spaced Repetition Scheduler) với khả năng thích ứng cao hơn chưa được tích hợp.

- **Thiếu phiên bản Mobile App:** Hiện tại ứng dụng mới chỉ là Web App (Responsive), chưa có ứng dụng gốc (Native App) trên iOS/Android để tối ưu hóa trải nghiệm học tập offline.

4.3 Hướng phát triển

Dựa trên những hạn chế đã nêu, nhóm đề xuất các hướng phát triển tiếp theo để hoàn thiện sản phẩm:

1. **Nâng cấp Thuật toán AI (OCR & Vision):** Tích hợp công nghệ OCR (như Google Cloud Vision hoặc Tesseract) để cho phép người dùng chụp ảnh tài liệu giấy hoặc upload file scan, AI sẽ đọc và tạo thẻ từ hình ảnh.
2. **Triển khai Thuật toán FRS:** Nghiên cứu và thay thế SM-2 bằng FRS để tối ưu hóa lịch ôn tập cá nhân hóa sâu hơn, giúp giảm số lượng thẻ cần ôn mà vẫn đảm bảo độ nhớ.
3. **Xây dựng Mobile App:** Phát triển phiên bản ứng dụng di động sử dụng React Native hoặc Flutter, hỗ trợ tính năng **Offline Mode** (Học khi không có mạng và đồng bộ sau).
4. **Phát triển tính năng Cộng đồng (Marketplace):** Cho phép người dùng chia sẻ bộ thẻ của mình lên kho chung, có tính năng đánh giá (Rating), bình luận và có thể kiếm tiền từ các bộ thẻ chất lượng cao (Monetization).

TÀI LIỆU THAM KHẢO

- [1] H. Ebbinghaus, *Memory: A Contribution to Experimental Psychology*. New York: Teachers College, Columbia University, 1913. (Nguồn gốc về Đường cong quên lãng).
- [2] P. A. Wozniak, "Optimization of learning," Master's Thesis, University of Technology in Poznan 1990. [Online]. Available: <https://www.supermemo.com/en/archives1990-2015/english/ol/entry>. (Cha đẻ của thuật toán SM-2).
- [3] B. Settles and B. Meeder, "A Trainable Spaced Repetition Model for Language Learning," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 1284–1293. (Tài liệu về các mô hình SRS hiện đại như Duolingo).
- [4] OpenAI, "GPT-4 Technical Report," 2023. [Online]. Available: <https://arxiv.org/abs/2303.08774>. (Cơ sở lý thuyết về LLM).
- [5] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *NeurIPS*, 2022. (Cơ sở lý thuyết về Prompt Engineering).
- [6] K. Myśliwiec et al., "NestJS - A progressive Node.js framework," [Online]. Available: <https://docs.nestjs.com/>. [Accessed: Jan. 20, 2026].
- [7] E. You et al., "Vue.js - The Progressive JavaScript Framework," [Online]. Available: <https://vuejs.org/guide/introduction.html>. [Accessed: Jan. 20, 2026].
- [8] PostgreSQL Global Development Group, "PostgreSQL 16.1 Documentation," [Online]. Available: <https://www.postgresql.org/docs/>.
- [9] BullMQ Contributors, "BullMQ: Premium message queue for NodeJS based on Redis," [Online]. Available: <https://docs.bullmq.io/>. (Tài liệu về xử lý hàng đợi).
- [10] Docker Inc., "Docker Documentation," [Online]. Available: <https://docs.docker.com/>.
- [11] Google Developers, "Google Identity: OAuth 2.0 for Client-side Web Applications," [Online]. Available: <https://developers.google.com/identity/protocols/oauth2>.

PHỤ LỤC 1

PHỤ LỤC 2