

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

**Xây dựng hệ thống học từ vựng trên nền tảng
Android và web**

ĐINH THỊ HỒNG PHÚC

phuc.dth215118@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: TS. Đỗ Quốc Huy

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2025

LỜI CAM KẾT

Họ và tên sinh viên: Đinh Thị Hồng Phúc

Điện thoại liên lạc: 0946.720.573

Email: phuc.dth215118@sis.hust.edu.vn

Lớp: Việt Nhật 03 - K66

Hệ đào tạo: Cử nhân

Tôi – *Đinh Thị Hồng Phúc* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Đỗ Quốc Huy*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Đinh Thị Hồng Phúc

LỜI CẢM ƠN

Lời đầu tiên, tôi xin gửi lời cảm ơn chân thành đến TS. Đỗ Quốc Huy, người đã hướng dẫn tận tình, góp ý chuyên môn và định hướng rõ ràng trong suốt quá trình thực hiện đồ án. Sự hỗ trợ của thầy đã giúp tôi từng bước vượt qua những khó khăn trong quá trình thực hiện.

Tôi cũng xin gửi lời cảm ơn tới quý thầy cô Trường Công nghệ Thông tin và Truyền thông - Đại học Bách Khoa Hà Nội vì đã truyền đạt những kiến thức nền tảng và chuyên môn trong suốt quá trình học tập.

Cuối cùng, tôi muốn gửi lời cảm ơn đến gia đình và bạn bè đã luôn bên cạnh động viên, hỗ trợ cả về tinh thần lẫn vật chất trong suốt thời gian thực hiện đồ án. Sự đồng hành của mọi người là nguồn động lực to lớn giúp tôi vượt qua những khó khăn để hoàn thành đồ án.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong bối cảnh nhu cầu học từ vựng ngoại ngữ ngày càng tăng, đặc biệt với sự phát triển mạnh mẽ của công nghệ giáo dục số, các hệ thống học từ vựng trực tuyến đã trở nên phổ biến như một công cụ hỗ trợ học tập hiệu quả. Tuy nhiên, nhiều hệ thống hiện tại vẫn tồn tại những hạn chế nhất định như thiếu khả năng cá nhân hóa tiến trình học, chưa tối ưu hóa việc ôn tập dựa trên mô hình ghi nhớ của từng cá nhân. Một số hướng tiếp cận hiện có như sử dụng thuật toán SM-2, học qua flashcard hoặc tích hợp trò chơi hóa đã giúp cải thiện hiệu quả ghi nhớ, nhưng vẫn chưa đáp ứng đầy đủ các yêu cầu về linh hoạt, khả năng mở rộng và tương tác theo thời gian thực.

Trước những hạn chế đó, đồ án lựa chọn hướng tiếp cận xây dựng một hệ thống học từ vựng tích hợp thuật toán SM-2 đã được điều chỉnh nhằm hỗ trợ ôn tập ngắn hạn và dài hạn theo trạng thái học thực tế của từng người dùng. Hướng tiếp cận này được lựa chọn vì vừa kế thừa các ưu điểm từ SM-2 gốc, vừa cho phép cá nhân hóa tiến trình học, tăng cường khả năng phản hồi thời gian thực và hỗ trợ học các bộ sưu tập công khai mà không cần sao chép.

Giải pháp được đề xuất trong đồ án bao gồm việc thiết kế một hệ thống ba nền tảng: ứng dụng web (ReactJS), ứng dụng di động (Flutter) và máy chủ xử lý trung tâm (Laravel). Hệ thống cho phép người dùng học trực tiếp từ các bộ thẻ công khai, ghi nhận tiến độ học riêng biệt, điều chỉnh lịch ôn dựa trên thời gian và chất lượng phản hồi, đồng thời đánh giá mức độ ghi nhớ thông qua chỉ số độ dễ và số lượt trả lời sai.

Đóng góp chính của đồ án là đề xuất và triển khai thành công một cơ chế điều chỉnh SM-2 phù hợp với học ngắn hạn theo phút, kết hợp với mô hình lưu trữ trạng thái flashcard linh hoạt. Hệ thống đã bước đầu chứng minh hiệu quả trong việc theo dõi tiến độ học và đề xuất lịch ôn phù hợp, góp phần cải thiện đáng kể hiệu suất ghi nhớ từ vựng của người học.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	6
2.2.1 Biểu đồ use case tổng quát	6
2.2.2 Biểu đồ use case phân rã "Học các bộ thẻ"	7
2.2.3 Biểu đồ use case phân rã "Quản trị hệ thống"	8
2.2.4 Biểu đồ use case phân rã "Quản lý thông báo"	10
2.2.5 Biểu đồ use case phân rã "Quản lý tài khoản"	11
2.2.6 Quy trình nghiệp vụ "Học từ vựng hàng ngày"	12
2.3 Đặc tả chức năng	14
2.3.1 Đặc tả use case "Học bằng flashcard"	14
2.3.2 Đặc tả use case "Nhập từ"	15
2.3.3 Đặc tả use case "Quản lý giới hạn bộ thẻ của người dùng"	16
2.3.4 Đặc tả use case "Theo dõi số lượng người dùng, bộ thẻ"	17
2.4 Yêu cầu phi chức năng	17
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	19
3.1 Ngôn ngữ lập trình và nền tảng phát triển.....	19
3.2 Cơ sở dữ liệu	19

3.3 Thuật toán SM-2 và điều chỉnh trong hệ thống	20
3.3.1 Đường cong lặn quên của Ebbinghaus	20
3.3.2 Nguyên lý SM-2 gốc	21
3.3.3 Điều chỉnh SM-2 trong hệ thống.....	21
3.3.4 So sánh với các thuật toán có chức năng tương tự.....	22
3.4 Công cụ hỗ trợ phát triển và triển khai	22
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	24
4.1 Thiết kế kiến trúc.....	24
4.1.1 Lựa chọn kiến trúc phần mềm	24
4.1.2 Thiết kế tổng quan.....	25
4.1.3 Thiết kế chi tiết gói	27
4.2 Thiết kế chi tiết.....	28
4.2.1 Thiết kế giao diện	28
4.2.2 Thiết kế lớp	35
4.2.3 Thiết kế cơ sở dữ liệu	41
4.3 Xây dựng ứng dụng.....	45
4.3.1 Thư viện và công cụ sử dụng.....	45
4.3.2 Kết quả đạt được	45
4.3.3 Minh họa các chức năng chính	48
4.4 Kiểm thử.....	51
4.5 Triển khai	52
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	53
5.1 Cá nhân hóa tiến độ học trên tài nguyên công khai.....	53
5.1.1 Giới thiệu vấn đề.....	53
5.1.2 Giải pháp	53
5.1.3 Kết quả đạt được	54

5.2 Điều chỉnh thuật toán SM-2 để hỗ trợ ôn tập sớm	54
5.2.1 Giới thiệu vấn đề.....	54
5.2.2 Giải pháp	55
5.2.3 Kết quả	58
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	59
6.1 Kết luận	59
6.2 Hướng phát triển.....	60
TÀI LIỆU THAM KHẢO.....	61
PHỤ LỤC.....	61

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát của hệ thống	6
Hình 2.2	Biểu đồ use case phân rã: Học các bộ thẻ	7
Hình 2.3	Biểu đồ use case phân rã: Quản trị hệ thống	8
Hình 2.4	Biểu đồ use case phân rã: Quản lý thông báo	10
Hình 2.5	Biểu đồ use case phân rã: Quản lý tài khoản	11
Hình 2.6	Biểu đồ hoạt động: Quy trình học từ vựng hàng ngày	12
Hình 3.1	Đường cong lãng quên (mô phỏng theo Ebbinghaus)	20
Hình 4.1	Mô hình 3 tầng client-server	24
Hình 4.2	Biểu đồ phụ thuộc gói của hệ thống	26
Hình 4.3	Biểu đồ gói chi tiết chức năng ôn tập flashcard	27
Hình 4.4	Giao diện web: Trang danh sách bộ thẻ công khai	30
Hình 4.5	Giao diện web: Trang học bằng flashcard	31
Hình 4.6	Giao diện web: Trang thống kê tiến độ	32
Hình 4.7	Giao diện mobile: Trang danh sách bộ thẻ công khai	33
Hình 4.8	Giao diện mobile: Trang học bằng flashcard	34
Hình 4.9	Giao diện mobile: Trang thống kê tiến độ	35
Hình 4.10	Lớp users	36
Hình 4.11	Lớp collections	37
Hình 4.12	Lớp flashcards	38
Hình 4.13	Lớp flashcard_statuses	39
Hình 4.14	Biểu đồ trình tự chức năng Đánh giá mức độ ghi nhớ	40
Hình 4.15	Biểu đồ thực thể liên kết (E-R diagram)	42
Hình 4.16	Tiến độ và chức năng dành cho khách truy cập và người dùng .	46
Hình 4.17	Tiến độ và chức năng dành cho người dùng đã đăng nhập . . .	47
Hình 4.18	Tiến độ và chức năng dành cho quản trị viên hệ thống	47
Hình 4.19	Chức năng học từ vựng bằng flashcard (web)	49
Hình 4.20	Chức năng thống kê quá trình học tập (web)	50

DANH MỤC BẢNG BIỂU

Bảng 2.1	So sánh giữa hai hệ thống Anki và Quizlet	5
Bảng 2.2	Đặc tả use case "Học bằng flashcard"	14
Bảng 2.3	Đặc tả use case "Nhập từ"	15
Bảng 2.4	Đặc tả use case "Quản lý giới hạn bộ thẻ của người dùng" . .	16
Bảng 2.5	Đặc tả use case "Theo dõi số lượng người dùng, bộ thẻ"	17
Bảng 3.1	So sánh SM-2 với các thuật toán thay thế	22
Bảng 4.1	Ý nghĩa từng thực thể	41
Bảng 4.2	Thiết kế chi tiết bảng collections	43
Bảng 4.3	Thiết kế chi tiết bảng flashcards	43
Bảng 4.4	Thiết kế chi tiết bảng flashcard_statuses	44
Bảng 4.5	Thiết kế chi tiết bảng flashcard_review_logs	44
Bảng 4.6	Danh sách thư viện và công cụ sử dụng	45
Bảng 4.7	Thông kê hệ thống	48
Bảng 4.8	Các trường hợp kiểm thử chức năng chính	51

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CRUD	Thao tác cơ bản trong quản lý dữ liệu (CREATE, READ, UPDATE, DELETE)
EF	Chỉ số độ dễ easy factor của thuật toán SM-2
FSRS	Thuật toán ghi nhớ của Anki (Free Spaced Repetition Scheduler)
ORM	Kỹ thuật chuyển đổi dữ liệu giữa cơ sở dữ liệu quan hệ và ngôn ngữ lập trình hướng đối tượng
SR	Thuật toán lặp lại ngắt quãng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Chương đầu tiên trình bày bối cảnh và lý do chọn đề tài, chỉ ra những hạn chế còn tồn tại trong phương pháp học từ vựng phổ biến hiện nay, từ đó xác định rõ mục tiêu và định hướng giải pháp của đề tài. Nội dung chương gồm bốn phần chính: (i) đặt vấn đề và chỉ rõ tính cấp thiết của đề tài; (ii) trình bày mục tiêu và phạm vi đề tài; (iii) đưa ra định hướng giải pháp kỹ thuật; và (iv) mô tả bố cục đồ án.

1.1 Đặt vấn đề

Trong bối cảnh toàn cầu hóa, năng lực ngôn ngữ đã trở thành một yêu cầu thiết yếu trong giáo dục, phát triển nghề nghiệp. Nghiên cứu của [1] Schmitt (2008) chỉ ra rằng, đối với người học ngôn ngữ thứ hai, đặc biệt là những cá nhân không sống trong môi trường giao tiếp bản ngữ, việc ghi nhớ từ vựng là một thách thức lớn. Bên cạnh đó, [2] Nation (2013) phân tích sâu các quy trình học từ vựng và nhận thấy rằng, mặc dù phương pháp flashcard truyền thống được áp dụng rộng rãi, nó vẫn mang theo những hạn chế về mặt tối ưu hóa việc ôn tập và cá nhân hóa lộ trình học.

Cụ thể, **phương pháp flashcard truyền thống** còn tồn tại ba hạn chế chính, bao gồm (i) thiếu cơ chế ôn tập hiệu quả để chống lại "**đường cong lãng quên**" được [3] Ebbinghaus đề xuất từ năm 1885 - hiện tượng giảm sút trí nhớ nếu không được củng cố định kỳ, (ii) không hỗ trợ cá nhân hóa lộ trình học tập theo nhu cầu của từng người học, và (iii) còn hạn chế về khả năng tương tác đa phương tiện

Những hạn chế này đặt ra một bài toán trong lĩnh vực giáo dục ngôn ngữ: làm thế nào để xây dựng một phương pháp học từ vựng có thể hỗ trợ người học ghi nhớ hiệu quả, bền vững và linh hoạt hơn. Việc giải quyết bài toán này không chỉ giúp cải thiện hiệu quả học ngoại ngữ, mà còn có thể mở rộng ứng dụng trong các lĩnh vực khác như y học, kỹ thuật và đào tạo chuyên ngành, nơi việc **ghi nhớ hệ thống kiến thức** là yếu tố then chốt.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay, trên thị trường có nhiều ứng dụng hỗ trợ học từ vựng ngoại ngữ, tiêu biểu như **Anki**, **Quizlet**. Các ứng dụng này phần lớn đều dựa trên mô hình flashcard, cung cấp các chức năng cơ bản như tạo thẻ, ôn tập, và theo dõi tiến độ học tập. Bên cạnh đó, một số sản phẩm tích hợp **thuật toán lặp lại ngắt quãng (Spaced Repetition)** để cải thiện hiệu quả ghi nhớ. Tuy nhiên, qua phân tích và đánh giá tổng quan, phần lớn các hệ thống này vẫn tồn tại một số hạn chế như sau: (i) mức độ cá nhân hóa còn hạn chế, chưa theo sát năng lực và mục tiêu học tập

của từng người dùng; (ii) hệ thống có nhiều chức năng nhưng giao diện chưa thân thiện, gây khó khăn trong việc sử dụng hiệu quả; (iii) thuật toán ôn tập còn đơn giản, chưa thật sự tối ưu, một số hệ thống vẫn sử dụng cơ chế lặp lại cứng nhắc.

Từ những tồn tại nêu trên, đề tài này hướng tới mục tiêu phát triển một hệ thống học từ vựng hỗ trợ quá trình ghi nhớ hiệu quả thông qua cơ chế ôn tập thông minh, phù hợp với từng cá nhân, với các chức năng chính bao gồm: (i) quản lý bộ sưu tập từ vựng, (ii) học từ vựng theo cơ chế flashcard tích hợp **thuật toán SM-2**, (iii) theo dõi tiến độ và trạng thái học của từng thẻ từ (mới, đang học, đã nhớ, v.v.) theo thời gian thực, (iv) học tập linh hoạt trên cả nền tảng **web** và **ứng dụng di động**, đảm bảo khả năng đồng bộ dữ liệu và học tập liên tục, và (v) giao diện thân thiện, dễ sử dụng.

Hệ thống hướng tới việc nâng cao hiệu quả **ghi nhớ dài hạn**, đồng thời tối giản thao tác, tạo điều kiện cho người học duy trì quá trình học một cách bền vững và thuận tiện. Nội dung chi tiết về kiến trúc hệ thống, thuật toán và giao diện sẽ được trình bày cụ thể trong các chương tiếp theo.

1.3 Định hướng giải pháp

Để giải quyết bài toán đã xác định trong phần 1.2, đề án định hướng phát triển một hệ thống học từ vựng, áp dụng phương **pháp lặp lại ngắt quãng (Spaced Repetition)** nhằm tối ưu hóa khả năng ghi nhớ dài hạn. **Thuật toán SM-2** được lựa chọn làm thuật toán cốt lõi cho việc tổ chức lịch học và ôn tập từ vựng.

Về công nghệ, hệ thống được thiết kế với kiến trúc đa nền tảng bao gồm: (i) giao diện web phát triển bằng **ReactJS**, (ii) ứng dụng di động Android phát triển bằng **Flutter**, (iii) backend xây dựng bằng **Laravel**, (iv) cơ sở dữ liệu sử dụng **MySQL**, (v) triển khai backend và database trên máy chủ **AWS EC2**, nhằm đảm bảo khả năng mở rộng và hoạt động ổn định.

Giải pháp được đề xuất là thiết kế và triển khai một hệ thống học từ vựng sử dụng mô hình flashcard, cho phép người học ôn tập từ vựng theo lịch trình được cá nhân hóa dựa trên **thuật toán SM-2**. Dữ liệu học và trạng thái thẻ được đồng bộ giữa các nền tảng nhằm mang đến trải nghiệm học tập nhất quán và linh hoạt.

Đóng góp chính của đề án là xây dựng một hệ thống học từ vựng đa nền tảng, có khả năng cá nhân hóa quá trình học theo thời gian thực, đồng thời cung cấp giao diện trực quan, dễ sử dụng. Kết quả đạt được là một ứng dụng hoàn chỉnh hỗ trợ người học cải thiện khả năng **ghi nhớ từ vựng lâu dài**, đồng thời nâng cao **hiệu quả** và **trải nghiệm học tập**. Nội dung chi tiết sẽ được trình bày trong các chương tiếp theo.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày quá trình khảo sát và phân tích yêu cầu cho hệ thống. Trong chương này, các nhu cầu thực tiễn của người học được tổng hợp thông qua khảo sát và phân tích các hệ thống hiện có. Từ đó, các yêu cầu chức năng và phi chức năng của hệ thống được xác định và phân loại rõ ràng, làm cơ sở để thiết kế và phát triển hệ thống phù hợp với mục tiêu đề ra.

Chương 3 trình bày về các công nghệ được sử dụng trong quá trình xây dựng hệ thống. Cụ thể, chương này trình bày lý do lựa chọn các công nghệ sử dụng. Ngoài ra, chương cũng đề cập đến nguyên lý cơ bản của **thuật toán SM-2** được tích hợp trong hệ thống. Việc hiểu rõ các công nghệ này đóng vai trò nền tảng cho việc thiết kế và triển khai hệ thống hiệu quả.

Chương 4 trình bày về kết quả thực nghiệm, chi tiết về thiết kế kiến trúc hệ thống. Chương này cũng giới thiệu về việc xây dựng, kiểm thử và kết quả triển khai ứng dụng.

Chương 5 trình bày các giải pháp và đóng góp nổi bật của đồ án.

Chương 6 trình bày kết quả đồ án và định hướng phát triển trong tương lai sản phẩm.

Chương 1 đã trình bày tổng quan về bối cảnh nghiên cứu, đặt bài toán cần giải quyết và xác định mục tiêu, phạm vi cũng như định hướng giải pháp của đồ án. Hệ thống được đề xuất tích hợp **thuật toán SM-2** trên nền tảng web và di động nhằm cá nhân hóa quá trình học từ vựng và nâng cao khả năng ghi nhớ. Các nội dung chi tiết về khảo sát hệ thống hiện tại và xác định yêu cầu sẽ được trình bày trong chương 2.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Chương này trình bày quá trình khảo sát, phân tích yêu cầu và xác định các chức năng chính của hệ thống học từ vựng. Trên cơ sở nghiên cứu các hệ thống tương tự và đánh giá nhu cầu thực tế, chương sẽ mô tả tổng quan chức năng thông qua biểu đồ use case, phân rã các use case quan trọng, và mô tả quy trình nghiệp vụ. Bên cạnh đó, chương cũng đặc tả chi tiết một số use case tiêu biểu và đề cập đến các yêu cầu phi chức năng, làm cơ sở cho giai đoạn thiết kế và hiện thực hệ thống ở các chương sau.

2.1 Khảo sát hiện trạng

Để xây dựng một hệ thống hỗ trợ học và ghi nhớ từ vựng hiệu quả, việc khảo sát và phân tích các hệ thống hiện có đóng vai trò quan trọng nhằm rút ra những ưu điểm cần kế thừa cũng như các hạn chế cần khắc phục. Nội dung khảo sát trong phần này tập trung vào các ứng dụng học từ vựng đang phổ biến hiện nay, cụ thể là Anki và Quizlet. Đây là hai công cụ tiêu biểu cho hai hướng tiếp cận khác nhau: một bên chú trọng vào thuật toán ôn tập theo lộ trình (Anki), một bên chú trọng vào trải nghiệm người dùng và tính tương tác (Quizlet).

Anki là một phần mềm học tập dựa trên mô hình flashcard và ứng dụng *thuật toán SR* nhằm tối ưu hoá quá trình ghi nhớ. Hệ thống cho phép người dùng tạo và tùy biến các bộ thẻ với nhiều định dạng nội dung như văn bản, hình ảnh, âm thanh. Anki hỗ trợ đa nền tảng và cung cấp công cụ thống kê quá trình học tập, từ đó hỗ trợ người học điều chỉnh chiến lược ôn tập phù hợp. Giao diện của Anki tương đối đơn giản và tối giản, giúp người dùng dễ tiếp cận hơn. Tuy nhiên, đối với người dùng mới, việc làm quen với các tính năng và cách thức sử dụng có thể gặp nhiều khó khăn, đặc biệt khi thiếu hướng dẫn cụ thể. Ngoài ra, việc tạo flashcard thủ công tiêu tốn nhiều thời gian và công sức, khiến việc tiếp cận ban đầu trở nên khó khăn với người dùng phổ thông.

Quizlet là nền tảng học trực tuyến cho phép người dùng tạo, chia sẻ và sử dụng các bộ thẻ ghi nhớ. Ứng dụng tích hợp các tính năng như học bằng flashcard, bài kiểm tra, trò chơi tương tác và học nhóm. Ưu điểm nổi bật của Quizlet là giao diện trực quan, dễ sử dụng, hỗ trợ gợi ý nội dung khi tạo thẻ, và tích hợp âm thanh (tự động) giúp tăng khả năng ghi nhớ. Tuy nhiên, Quizlet không áp dụng *thuật toán SR* trong quy trình ôn tập, do đó hiệu quả ghi nhớ phụ thuộc hoàn toàn vào sự chủ động của người học. Các bộ thẻ thường chỉ bao gồm từ vựng và nghĩa tương ứng.

Tiêu chí	Anki	Quizlet
Phương pháp học	Áp dụng thuật toán SR để tối ưu hóa thời điểm ôn tập	Dựa vào sự chủ động của người học, không có cơ chế tự động đề xuất lịch ôn tập
Tạo flashcard	Tùy chỉnh cao, hỗ trợ chèn văn bản, hình ảnh, âm thanh, đường link	Tạo nhanh, có gợi ý nghĩa, tích hợp sẵn âm thanh
Chia sẻ nội dung	Cho phép chia sẻ bộ thẻ và tải về từ cộng đồng người dùng	Dễ dàng tìm kiếm, chia sẻ và sử dụng bộ thẻ từ cộng đồng, không thể tải bộ thẻ
Trò chơi / Hoạt động tương tác	Không hỗ trợ trò chơi học tập tích hợp sẵn	Cung cấp trò chơi học từ vựng, bài kiểm tra, học nhóm trực tuyến
Thống kê học tập	Thống kê chi tiết quá trình học, hiệu suất ghi nhớ	Có thống kê theo ngày học
Giao diện	Tối giản, có thể tùy chỉnh nhưng ít thu hút với người dùng mới	Giao diện hiện đại, thân thiện và dễ sử dụng
Khả năng đồng bộ	Đồng bộ giữa các nền tảng (máy tính, điện thoại, web) thông qua tài khoản Anki-Web	Hỗ trợ học đa nền tảng, đồng bộ tự động giữa các thiết bị
Hạn chế	Giao diện chưa thân thiện; việc tạo flashcard thủ công mất thời gian	Thiếu cơ chế học theo lịch trình khoa học

Bảng 2.1: So sánh giữa hai hệ thống Anki và Quizlet

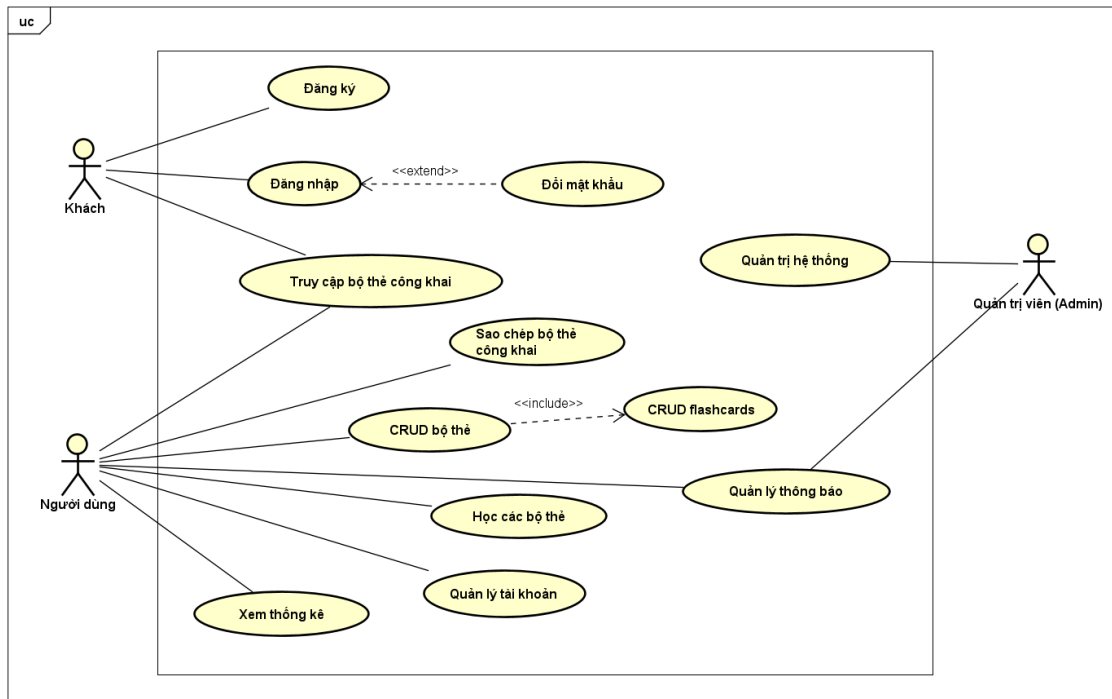
Từ việc phân tích hai hệ thống nêu trên, có thể rút ra một số điểm quan trọng về hướng thiết kế hệ thống. Anki có nền tảng thuật toán học tập vững chắc nhưng kém thân thiện với người mới bắt đầu. Quizlet có giao diện tốt và dễ tiếp cận nhưng lại thiếu chiến lược ôn tập hiệu quả. Do đó, việc phát triển một hệ thống kết hợp được ưu điểm của cả hai – cụ thể là ứng dụng thuật toán SR như Anki và giao diện thân thiện, dễ sử dụng như Quizlet – sẽ góp phần tạo nên một công cụ học từ vựng hiệu quả, phù hợp với người dùng phổ thông.

Trên cơ sở khảo sát, có thể xác định một số tính năng quan trọng cần được phát triển trong hệ thống bao gồm: hỗ trợ học tập dựa trên thuật toán SR (SM-2), cho phép tạo và tùy biến flashcard, hỗ trợ học đa nền tảng, cung cấp thống kê quá trình học và xây dựng giao diện đơn giản, dễ sử dụng.

2.2 Tổng quan chức năng

Phần này trình bày tổng quan các chức năng chính của hệ thống. Các chức năng được mô tả tổng quan, thông tin đặc tả chi tiết sẽ được trình bày trong phần 2.3.

2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quát của hệ thống

Hệ thống gồm ba tác nhân: **Khách**, **Người dùng** và **Quản trị viên (Admin)**

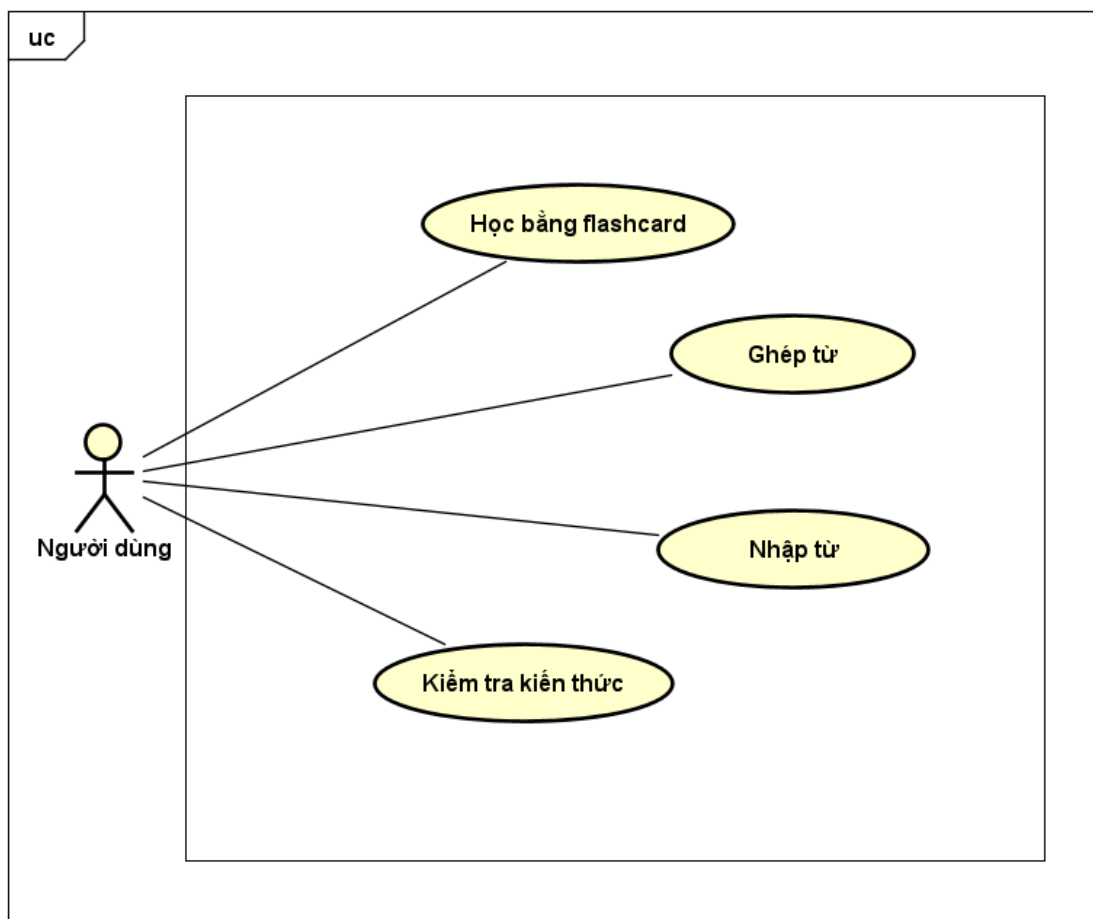
Khách là những người chưa đăng nhập vào hệ thống. Trong vai trò này, họ có thể thực hiện một số chức năng cơ bản như đăng ký tài khoản mới, đăng nhập, yêu cầu thiết lập lại mật khẩu, thiết lập lại mật khẩu thông qua đường dẫn gửi qua email, và xem các bộ thẻ công khai. Tuy nhiên, khách không có khả năng tương tác với nội dung của các bộ thẻ.

Người dùng là các khách đã đăng nhập thành công vào hệ thống. Sau khi đăng nhập, người dùng có quyền sử dụng đầy đủ các chức năng của hệ thống như học từ vựng theo thuật toán hỗ trợ ghi nhớ, theo dõi thống kê quá trình học, sao chép các bộ thẻ công khai, thực hiện các thao tác CRUD với bộ thẻ cá nhân, cũng như xem và cập nhật thông tin cá nhân, thay đổi mật khẩu.

Quản trị viên (Admin) là tác nhân có vai trò giám sát và quản lý nội dung hệ thống. Quản trị viên có thể duyệt hoặc xóa các bộ từ vựng bị người dùng đánh giá là vi phạm, quản lý người dùng và theo dõi các thống kê của hệ thống nhằm đảm bảo tính ổn định và nội dung phù hợp.

Các use case chính bao gồm: (i) Đăng ký/Đăng nhập: Để truy cập các tính năng nâng cao; (ii) Học từ vựng: Dựa trên thuật toán hỗ trợ ghi nhớ giúp hệ thống tự động xác định thời điểm ôn tập phù hợp; (iii) Quản lý bộ thẻ: Với các thao tác như tạo, chỉnh sửa, xóa các bộ thẻ cá nhân; (iv) Xem thống kê học tập: Theo dõi tiến độ và hiệu quả học tập; (v) Quản trị hệ thống (chỉ dành cho **Quản trị viên - Admin**): Bao gồm việc duyệt hoặc xóa bộ thẻ vi phạm và quản lý người dùng.

2.2.2 Biểu đồ use case phân rã "Học các bộ thẻ"



Hình 2.2: Biểu đồ use case phân rã: Học các bộ thẻ

Hình 2.2 trình bày biểu đồ use case phân rã cho chức năng “Học các bộ thẻ”. Trong chức năng này, người dùng thực hiện các hoạt động chính nhằm ghi nhớ và kiểm tra từ vựng một cách hiệu quả. Hệ thống cung cấp bốn phương pháp học chính, bao gồm: (i) **học bằng flashcard**, (ii) **ghép từ**, (iii) **nhập từ**, (iv) **kiểm tra kiến thức**.

Với chức năng **học bằng flashcard**, người dùng sẽ xem lần lượt các thẻ ghi nhớ, lật thẻ để kiểm tra nghĩa của từ và sau đó đánh giá mức độ ghi nhớ bằng cách lựa chọn một trong các mức: “Again”, “Hard”, “Good” hoặc “Easy”. Các lựa chọn này

được xử lý thông qua thuật toán SM-2 nhằm xác định thời điểm ôn tập tối ưu, qua đó cải thiện hiệu quả ghi nhớ.

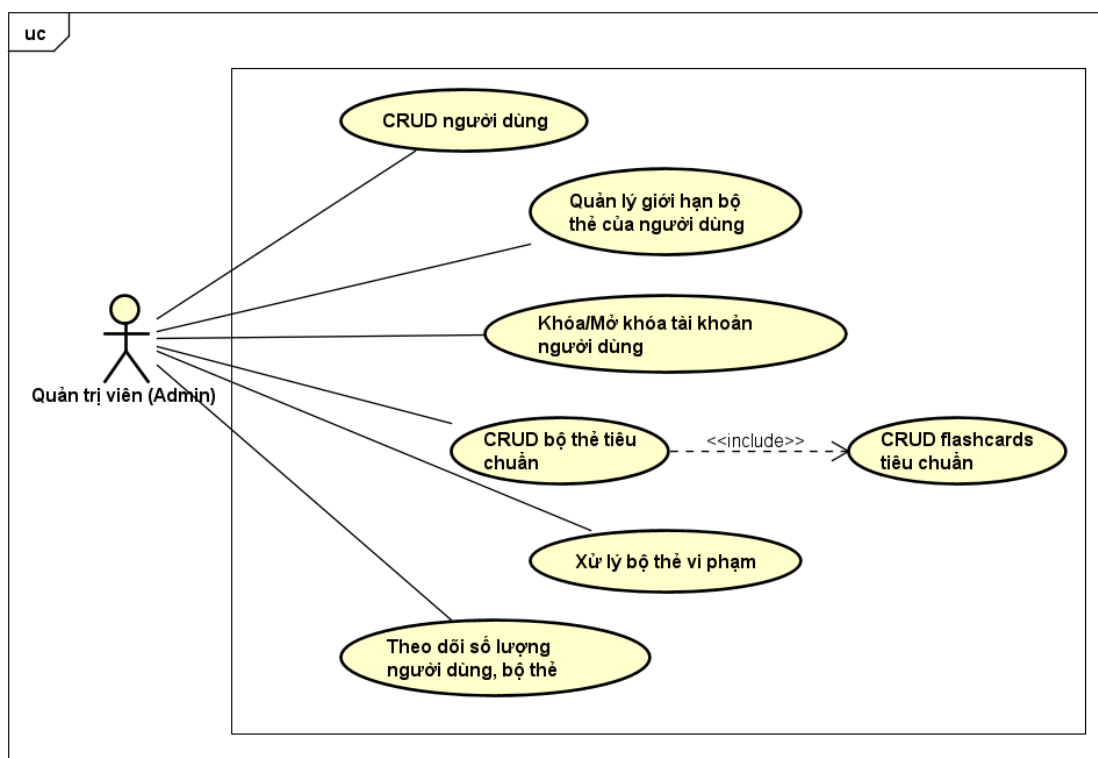
Đối với chức năng **ghép từ**, hệ thống hiển thị các cặp từ và nghĩa tương ứng. Người dùng sẽ lựa chọn cách ghép đúng giữa từ và nghĩa. Phương pháp này giúp củng cố khả năng ghi nhớ một cách nhanh chóng và trực quan.

Chức năng **nhập từ** yêu cầu người dùng chủ động gõ từ hoặc nghĩa từ bàn phím, nhằm kiểm tra khả năng ghi nhớ và sử dụng chính xác từ vựng đã học. Sau khi nhập đúng, hệ thống sẽ đưa ra bốn mức độ đánh giá giống như trong phương pháp **học bằng flashcard** (Again, Hard, Good, Easy) và sử dụng thuật toán SM-2 để tính toán thời điểm ôn tập tiếp theo. Điều này giúp kết hợp cả khả năng nhớ chính xác và khả năng tự đánh giá trong quá trình học.

Cuối cùng, chức năng **kiểm tra kiến thức**, giúp người dùng đánh giá mức độ thành thạo từ vựng thông qua bài kiểm tra tổng hợp với nhiều dạng câu hỏi khác nhau.

Các phương pháp học này hỗ trợ người dùng tiếp cận và củng cố từ vựng theo nhiều cách, qua đó nâng cao khả năng ghi nhớ và sử dụng ngôn ngữ một cách hiệu quả.

2.2.3 Biểu đồ use case phân rã "Quản trị hệ thống"



Hình 2.3: Biểu đồ use case phân rã: Quản trị hệ thống

Hình 2.3 trình bày biểu đồ use case phân rã cho chức năng "Quản lý hệ thống". Trong chức năng này, quản trị viên thực hiện các hoạt động nhằm duy trì và kiểm soát nội dung của hệ thống.

Trong chức năng **CRUD người dùng**, quản trị viên thực hiện các thao tác tạo mới, xem chi tiết, cập nhật và xóa tài khoản người dùng, nhằm đảm bảo dữ liệu người dùng luôn chính xác và được kiểm soát chặt chẽ.

Chức năng **Quản lý giới hạn bộ thể của người dùng** kiểm soát số lượng bộ thể mà mỗi người dùng có thể tạo, dựa trên hệ thống đánh giá cấp độ. Cơ chế đánh giá này hoạt động dựa trên điểm trung bình phản hồi từ cộng đồng người dùng theo thang điểm 1–5 sao. Khi người dùng đạt mức điểm trung bình nhất định, hệ thống sẽ tự động nâng cấp cấp độ tương ứng và mở rộng giới hạn bộ thể, giúp khuyến khích đóng góp nội dung chất lượng.

Bên cạnh đó, quản trị viên có quyền **Khóa/Mở khóa tài khoản người dùng** các tài khoản vi phạm, đảm bảo hệ thống hoạt động an toàn.

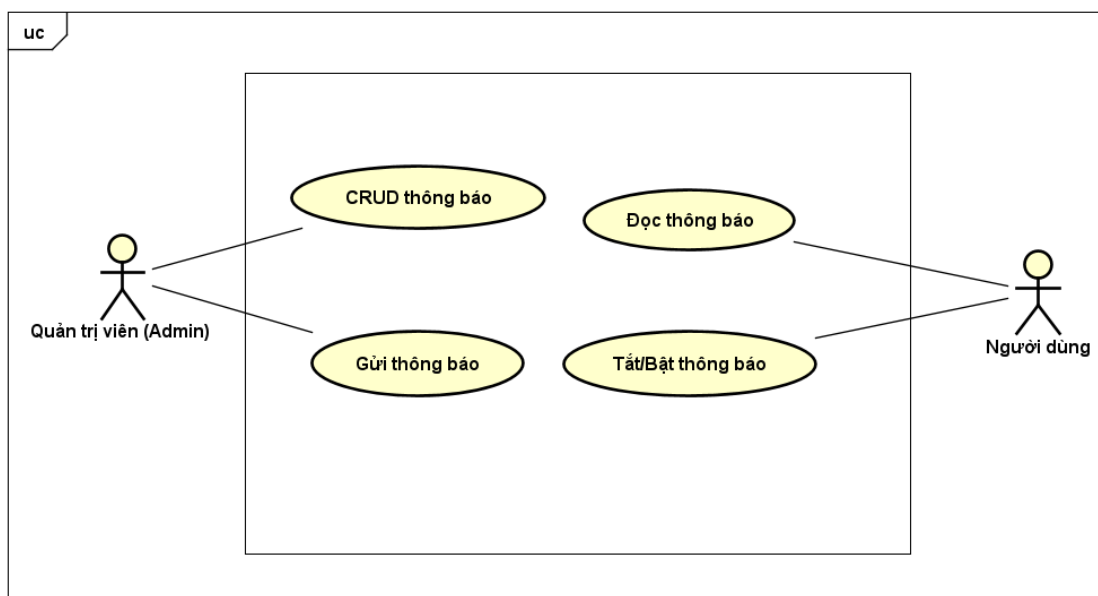
Trong chức năng **CRUD bộ thể tiêu chuẩn**, quản trị viên quản lý nội dung bộ thể chuẩn được sử dụng trong hệ thống, đảm bảo chất lượng dữ liệu học tập. Tương tự, chức năng **CRUD flashcards tiêu chuẩn** hỗ trợ kiểm soát nội dung flashcards chuẩn, giúp người dùng có tài liệu tham khảo phù hợp.

Chức năng **Xử lý bộ thể vi phạm** đóng vai trò quan trọng trong việc đảm bảo chất lượng và tính minh bạch của nội dung. Các bộ thể bị người dùng báo cáo sẽ được quản trị viên kiểm tra và đánh giá. Tùy theo mức độ vi phạm, các biện pháp xử lý sẽ được áp dụng. Nếu nội dung vi phạm nghiêm trọng, chẳng hạn như vi phạm bản quyền hoặc sai lệch thông tin nghiêm trọng, quản trị viên sẽ tiến hành xóa bộ thể và gửi thông báo đến tác giả về lý do vi phạm. Đối với những sai sót nhỏ, quản trị viên sẽ yêu cầu tác giả chỉnh sửa để đảm bảo nội dung phù hợp trước khi tiếp tục được hiển thị công khai toàn hệ thống.

Cuối cùng, chức năng **Theo dõi số lượng người dùng, bộ thể** giúp quản trị viên giám sát tổng quan số lượng tài khoản và bộ thể đang được sử dụng, cung cấp dữ liệu phân tích cho việc tối ưu hóa hệ thống.

Các chức năng này đảm bảo hệ thống vận hành ổn định, hỗ trợ quản trị viên trong việc duy trì chất lượng nội dung và kiểm soát người dùng một cách hiệu quả.

2.2.4 Biểu đồ use case phân rã "Quản lý thông báo"



Hình 2.4: Biểu đồ use case phân rã: Quản lý thông báo

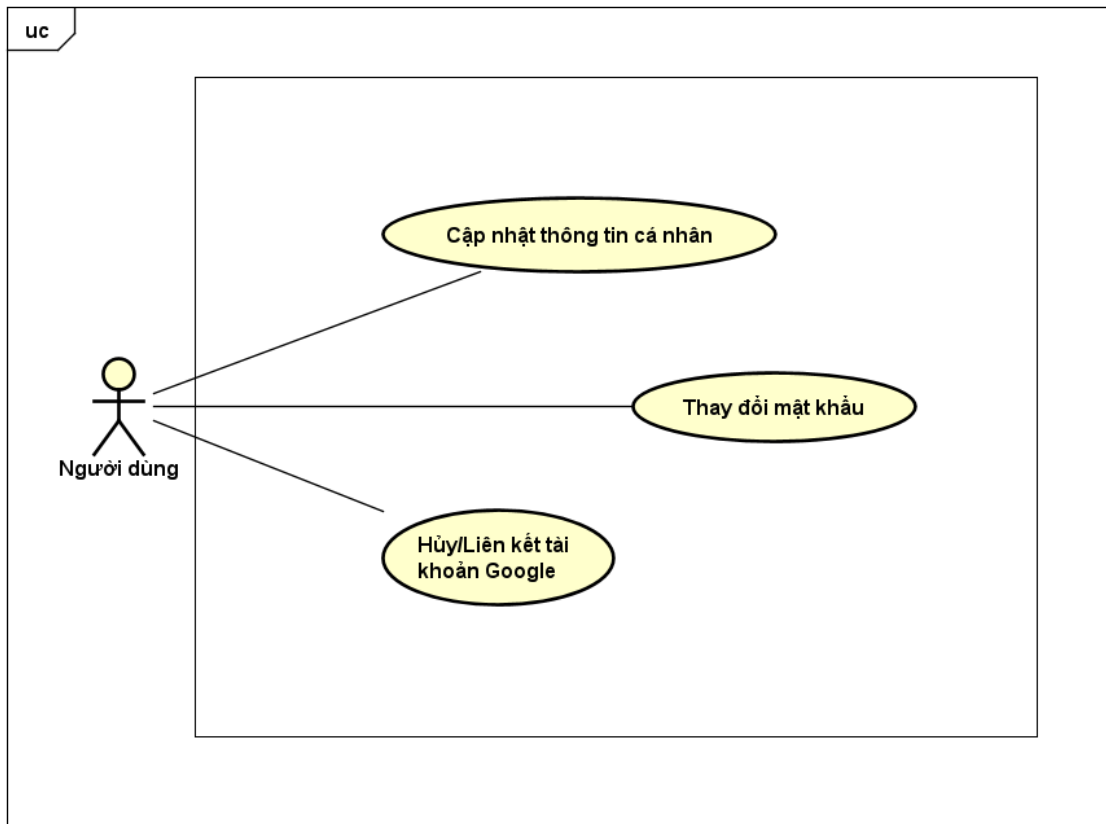
Hình 2.4 trình bày biểu đồ use case phân rã cho chức năng "Quản lý thông báo". Trong chức năng này, hệ thống hỗ trợ hai tác nhân là **quản trị viên** và **người dùng** thực hiện các thao tác liên quan đến thông báo nhằm đảm bảo quá trình truyền tải thông tin được diễn ra kịp thời và có kiểm soát.

Đối với **quản trị viên**, hệ thống cung cấp chức năng tạo, xem, chỉnh sửa và xóa thông báo (**CRUD thông báo**), giúp cập nhật nội dung thông báo một cách linh hoạt và phù hợp với từng thời điểm. Đồng thời, chức năng **Gửi thông báo** cho phép quản trị viên truyền tải các thông tin quan trọng đến người dùng một cách chính xác, góp phần duy trì sự kết nối giữa hệ thống và cộng đồng người dùng.

Về phía **người dùng**, hệ thống cung cấp chức năng **Đọc thông báo**, cho phép họ truy cập danh sách các thông báo đã được gửi đến, qua đó nhanh chóng nắm bắt các thay đổi hoặc thông tin quan trọng từ hệ thống. Bên cạnh đó, chức năng **Tắt/Bật thông báo** giúp người dùng kiểm soát việc tiếp nhận thông tin, tránh bị làm phiền bởi các thông báo không cần thiết theo nhu cầu cá nhân.

Các chức năng này đảm bảo rằng thông báo trong hệ thống được quản lý chặt chẽ, hỗ trợ cả quản trị viên và người dùng trong việc tiếp cận và xử lý thông tin một cách hiệu quả.

2.2.5 Biểu đồ use case phân rã "Quản lý tài khoản"



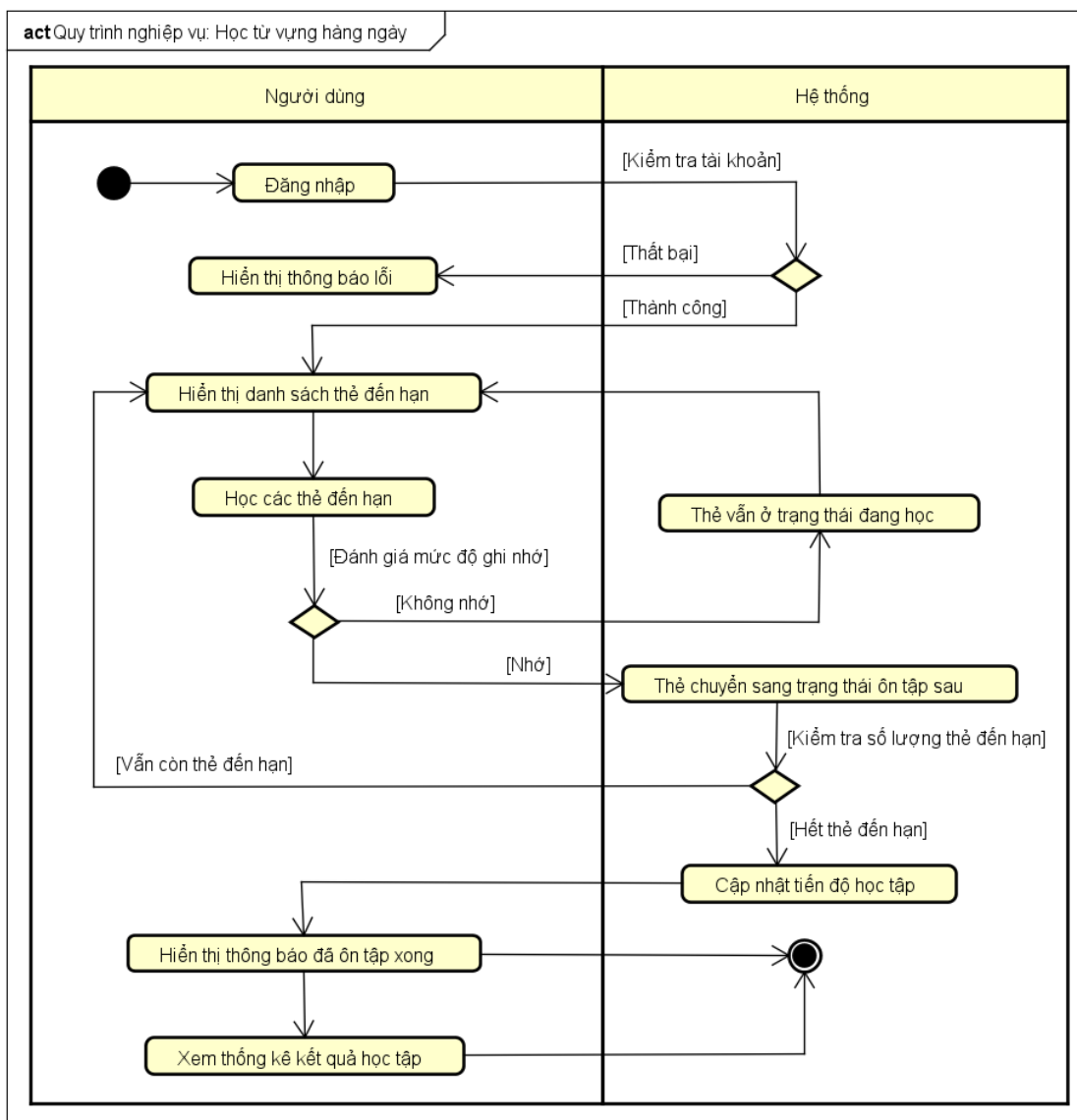
Hình 2.5: Biểu đồ use case phân rã: Quản lý tài khoản

Hình 2.5 trình bày biểu đồ use case phân rã cho chức năng "Quản lý tài khoản". Trong chức năng này, người dùng thực hiện các hoạt động nhằm cập nhật và quản lý thông tin cá nhân một cách hiệu quả.

Với chức năng **Cập nhật thông tin cá nhân**, người dùng có thể thay đổi họ tên, tên người dùng (username) và địa chỉ email - trong trường hợp tài khoản chưa liên kết với Google. Ngoài ra, chức năng **Thay đổi mật khẩu** cho phép người dùng cập nhật mật khẩu mới để bảo vệ tài khoản khỏi các nguy cơ bảo mật. Cuối cùng, chức năng **Hủy/Liên kết tài khoản Google** tạo điều kiện cho người dùng lựa chọn giữa việc sử dụng tài khoản độc lập hoặc đồng bộ hóa thông tin đăng nhập thông qua Google để nâng cao trải nghiệm sử dụng.

Các chức năng này giúp người dùng kiểm soát tài khoản cá nhân một cách linh hoạt, bảo mật và thuận tiện trong quá trình sử dụng hệ thống.

2.2.6 Quy trình nghiệp vụ "Học từ vưng hàng ngày"



Hình 2.6: Biểu đồ hoạt động: Quy trình học từ vựng hàng ngày

Hình 2.6 mô tả luồng hoạt động chính của người dùng trong một phiên học hàng ngày, kết hợp các use case **Đăng nhập**, **Học các bộ thể** và **Xem thống kê** nhằm hỗ trợ người dùng duy trì thói quen học tập đều đặn và hiệu quả.

Đầu tiên, người dùng truy cập hệ thống và đăng nhập tài khoản thông qua chức năng **Đăng nhập**. Sau khi đăng nhập thành công, hệ thống tự động truy xuất và hiển thị danh sách các flashcard cần được ôn tập trong ngày, dựa trên lịch trình ôn tập được tính toán bằng thuật toán ghi nhớ.

Người dùng sau đó bắt đầu học từng thể và đưa ra đánh giá mức độ ghi nhớ, thông qua chức năng **Học các bộ thể**, lựa chọn mức độ ghi nhớ như *Again*, *Hard*, *Good* hoặc *Easy*. Dữ liệu đánh giá này được hệ thống xử lý theo thuật toán SM-2

nhằm điều chỉnh khoảng thời gian ôn tập cho từng từ vựng, bảo đảm việc lặp lại được thực hiện vào thời điểm tối ưu nhất cho trí nhớ dài hạn.

Khi phiên học kết thúc, hệ thống cập nhật dữ liệu học tập, điều chỉnh lịch ôn tập kế tiếp cho từng flashcard dựa trên kết quả vừa thu được. Đồng thời, người dùng có thể truy cập chức năng **Xem thống kê** để theo dõi tiến trình học tập trong ngày, đánh giá sự tiến bộ và đưa ra những điều chỉnh phù hợp cho các phiên học tiếp theo.

Quy trình này tạo một chu trình học tập khoa học, giúp người dùng ghi nhớ từ vựng hiệu quả thông qua việc kết hợp giữa luyện tập lặp lại, tự đánh giá và tối ưu hóa lịch trình học dựa trên thuật toán.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case "Học bằng flashcard"

Tên use case	Học bằng flashcard		
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đã đăng nhập thành công		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn bộ thẻ muốn học
	2	Hệ thống	Hiển thị chi tiết về bộ thẻ được chọn
	3	Người dùng	Chọn chế độ học bằng flashcard
	4	Hệ thống	Lấy và hiển thị danh sách flashcard đến hạn
	5	Người dùng	Học từng flashcard và đánh giá mức độ ghi nhớ (Again, Hard, Good, Easy)
	6	Hệ thống	Tính toán và cập nhật thời gian ôn tập tiếp theo dựa trên đánh giá của người dùng (theo thuật toán SM-2)
	7	Người dùng	Học xong tất cả flashcard trong danh sách
	8	Hệ thống	Thông báo hoàn thành phiên học
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Nếu không có flashcard đến hạn: Thông báo “Không có thẻ cần ôn tập vào lúc này”
Hậu điều kiện	Tiến trình học được cập nhật vào hệ thống Lịch ôn tập tiếp theo được thiết lập cho từng flashcard		

Bảng 2.2: Đặc tả use case "Học bằng flashcard"

2.3.2 Đặc tả use case "Nhập từ"

Tên use case	Nhập từ		
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đã đăng nhập thành công		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn bộ thẻ muốn học
	2	Hệ thống	Hiển thị chi tiết bộ thẻ
	3	Người dùng	Chọn chế độ học bằng flashcard
	4	Hệ thống	Lấy và hiển thị danh sách flashcard đến hạn
	5	Người dùng	Gõ nghĩa của từ (hoặc từ) tương ứng với mặt còn lại của flashcard
	6	Hệ thống	Kiểm tra tính chính xác của từ được nhập
	7	Người dùng	Đánh giá mức độ ghi nhớ (Again, Hard, Good, Easy)
	8	Hệ thống	Tính toán và cập nhật thời gian ôn tập tiếp theo dựa trên đánh giá và kết quả nhập từ (theo thuật toán SM-2)
	9	Hệ thống	Hiển thị từ tiếp theo cần nhập hoặc thông báo hoàn thành nếu đã hết flashcard
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Nếu không có flashcard đến hạn: Thông báo “Không có thẻ cần ôn tập vào lúc này”
Hậu điều kiện	Tiến trình học và kết quả nhập từ được lưu lại Lịch ôn tập tiếp theo được thiết lập cho từng flashcard		

Bảng 2.3: Đặc tả use case "Nhập từ"

2.3.3 Đặc tả use case "Quản lý giới hạn bộ thẻ của người dùng"

Tên use case	Quản lý giới hạn bộ thẻ của người dùng		
Tác nhân	Hệ thống		
Tiền điều kiện	Người dùng thực hiện hành động tạo bộ thẻ mới		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Hệ thống	Kiểm tra tổng số bộ thẻ hiện tại của người dùng
	2	Hệ thống	Tính điểm đánh giá trung bình các bộ thẻ của người dùng từ phản hồi cộng đồng (1–5 sao)
	3	Hệ thống	Xác định cấp độ người dùng dựa trên bảng phân cấp
	4	Hệ thống	So sánh số bộ thẻ hiện tại với giới hạn cho phép của cấp độ
	5	Hệ thống	Cho phép tạo bộ thẻ mới nếu chưa vượt quá giới hạn
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Hệ thống	Nếu điểm đánh giá thay đổi: Tự động cập nhật lại cấp độ người dùng và giới hạn tương ứng
	4a	Hệ thống	Nếu cấp độ mới có giới hạn thấp hơn số bộ thẻ hiện có: Cho phép giữ nguyên nhưng không thể tạo thêm
Hậu điều kiện	5a	Hệ thống	Nếu đã đạt giới hạn: Từ chối yêu cầu tạo mới và hiển thị thông báo phù hợp
	Người dùng được cấp phép hoặc từ chối tạo bộ thẻ mới theo đúng giới hạn Cấp độ và giới hạn bộ thẻ được tự động cập nhật tương ứng với điểm đánh giá trung bình		

Bảng 2.4: Đặc tả use case "Quản lý giới hạn bộ thẻ của người dùng"

2.3.4 Đặc tả use case "Theo dõi số lượng người dùng, bộ thẻ"

Tên use case	Theo dõi số lượng người dùng, bộ thẻ		
Tác nhân	Quản trị viên		
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống quản trị		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Quản trị viên	Truy cập trang thống kê trong giao diện quản trị
	2	Hệ thống	Truy xuất dữ liệu về số lượng tài khoản người dùng, số lượng bộ thẻ hiện có
	3	Hệ thống	Hiển thị bảng thống kê tổng quan về người dùng và bộ thẻ
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Nếu dữ liệu không thể truy xuất: Hiển thị thông báo lỗi “Không thể lấy dữ liệu thống kê”
Hậu điều kiện	Quản trị viên nắm được số lượng người dùng và bộ thẻ hiện có, phục vụ cho việc giám sát và ra quyết định quản trị hệ thống		

Bảng 2.5: Đặc tả use case "Theo dõi số lượng người dùng, bộ thẻ"

2.4 Yêu cầu phi chức năng

Bên cạnh các yêu cầu chức năng, hệ thống cũng cần đáp ứng các yêu cầu phi chức năng nhằm đảm bảo hiệu quả và tính ổn định trong quá trình vận hành. Cụ thể như sau:

Độ tin cậy: Dữ liệu học tập của người dùng cần được lưu trữ và đồng bộ chính xác. Hệ thống phải đảm bảo rằng các thay đổi về trạng thái flashcard và tiến trình học được cập nhật đúng.

Tính dễ sử dụng: Giao diện người dùng được thiết kế đơn giản, trực quan, thân thiện với cả người mới bắt đầu và người dùng có kinh nghiệm. Các chức năng học, nhập từ và xem thống kê cần được bố trí rõ ràng, dễ truy cập.

Tính dễ bảo trì và mở rộng: Kiến trúc hệ thống được thiết kế theo hướng module hóa, giúp dễ dàng nâng cấp, chỉnh sửa hoặc tích hợp thêm các chức năng mới trong tương lai. Mã nguồn được tổ chức rõ ràng, tuân thủ các quy ước lập trình hiện đại để hỗ trợ công tác bảo trì.

Chương 2 đã khảo sát hiện trạng các hệ thống học từ vựng phổ biến như Anki và Quizlet, từ đó rút ra các điểm mạnh và hạn chế cần khắc phục. Trên cơ sở đó, các yêu cầu chức năng chính cho hệ thống được xác định, bao gồm học từ vựng theo thuật toán SR (SM-2), học đa phương pháp, hỗ trợ tạo và chia sẻ bộ từ vựng giữa người dùng, cũng như theo dõi tiến độ học tập. Các chức năng tổng quan được mô tả thông qua biểu đồ use case, với các tác nhân gồm Khách, Người dùng và Quản trị viên, cùng các use case chính như, học các bộ thẻ, quản lý bộ thẻ và quản trị hệ thống. Chương 3 sẽ trình bày các công nghệ được lựa chọn để xây dựng hệ thống.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Chương này trình bày các công nghệ và nền tảng được sử dụng để xây dựng hệ thống học từ vựng, bao gồm: công nghệ phát triển frontend và backend, cơ sở dữ liệu, mô hình thuật toán ghi nhớ theo thời gian (thuật toán SR), và các công cụ hỗ trợ phát triển. Mỗi công nghệ được lựa chọn dựa trên yêu cầu cụ thể của hệ thống đã phân tích ở Chương 2.

3.1 Ngôn ngữ lập trình và nền tảng phát triển

Hệ thống học từ vựng được phát triển dưới dạng ứng dụng web kết hợp với ứng dụng di động, nhằm mang đến trải nghiệm học tập liền mạch, tiện lợi và dễ sử dụng. Trên nền tảng web, hệ thống sử dụng **ReactJS** [4] – một thư viện JavaScript phổ biến và mạnh mẽ được duy trì bởi Facebook – cho phép xây dựng giao diện người dùng động, phản hồi nhanh và dễ bảo trì. ReactJS được lựa chọn thay vì các framework khác như Angular hay VueJS do khả năng tùy biến linh hoạt, cộng đồng lớn và khả năng tích hợp tốt với các thư viện mở rộng như Redux hay React Router.

Ở phía máy chủ, hệ thống sử dụng **Laravel** [5] – một framework PHP hiện đại – để xây dựng backend. Laravel cung cấp kiến trúc MVC rõ ràng, tích hợp sẵn các tính năng bảo mật, xác thực, phân quyền, tương tác cơ sở dữ liệu qua Eloquent ORM. So với các framework như CodeIgniter hay Symfony, Laravel được đánh giá cao về tính dễ đọc, dễ bảo trì, và phù hợp với quy mô của một đồ án cá nhân.

Ứng dụng di động của hệ thống được xây dựng bằng **Flutter** [6] – một bộ công cụ UI do Google phát triển, hỗ trợ viết một lần và chạy trên cả Android lẫn iOS. So với React Native hay Kotlin Multiplatform, Flutter nổi bật với hiệu suất gần như ứng dụng native, giao diện đẹp và thời gian phát triển ngắn.

3.2 Cơ sở dữ liệu

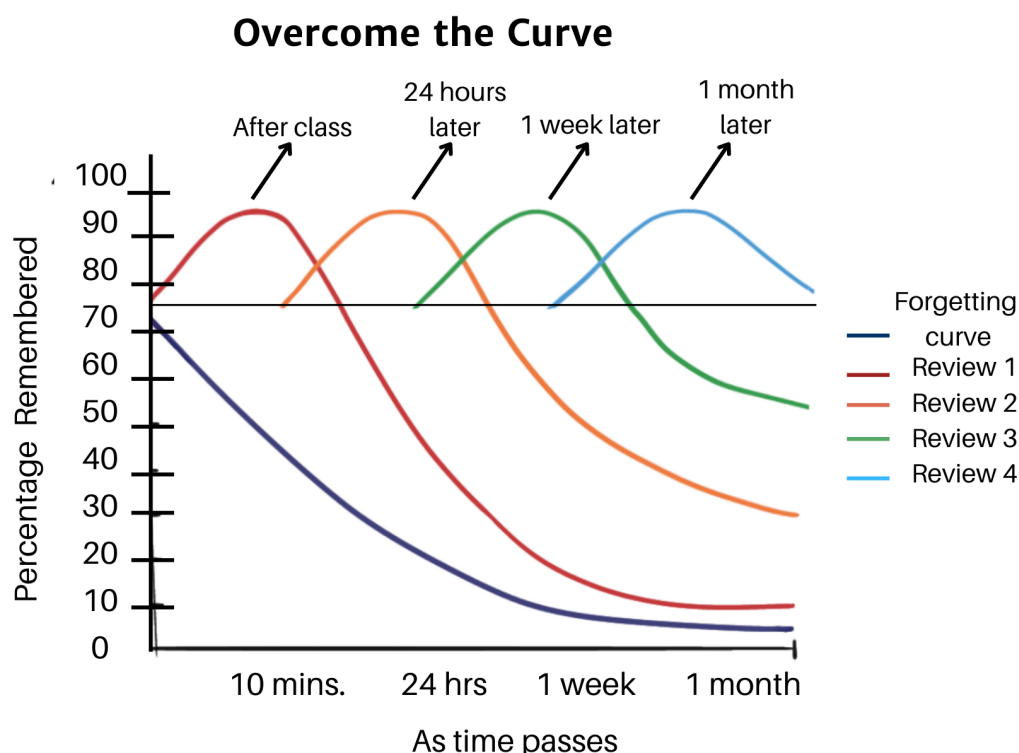
Cơ sở dữ liệu sử dụng trong hệ thống là **MySQL** [7], một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến. MySQL phù hợp cho ứng dụng vừa và nhỏ, hỗ trợ tốt trong môi trường web PHP, đồng thời có khả năng mở rộng khi cần. Cấu trúc dữ liệu của hệ thống được thiết kế để lưu trữ thông tin về người dùng, bộ thẻ, thẻ flashcard và trạng thái học tập cá nhân. Đặc biệt, hệ thống sử dụng bảng `flashcard_statuses` để lưu trạng thái hiện tại của từng thẻ cho từng người dùng, và bảng `user_flashcards` để lưu lịch sử học chi tiết, đảm bảo mô hình học tập theo thuật toán SM-2 được triển khai chính xác và dễ theo dõi.

3.3 Thuật toán SM-2 và điều chỉnh trong hệ thống

Trong hệ thống học từ vựng, thuật toán SM-2 được sử dụng để xác định thời điểm ôn tập tối ưu cho từng flashcard. SM-2 (SuperMemo-2) là thuật toán lặp cách quãng được phát triển bởi Piotr Wozniak và lần đầu tiên được công bố trong luận văn thạc sĩ của ông vào năm 1990 [8]. Đây là phiên bản đầu tiên được máy tính hóa của hệ thống SuperMemo, trở thành nền tảng cho nhiều ứng dụng học tập hiện đại. Mục tiêu chính của thuật toán là tận dụng đường cong lãng quên của con người để tối đa hóa hiệu quả ghi nhớ trong dài hạn.

3.3.1 Đường cong lãng quên của Ebbinghaus

Thuật toán SM-2 được xây dựng dựa trên quan sát tâm lý học của Hermann Ebbinghaus vào cuối thế kỷ XIX, được gọi là **đường cong lãng quên** (*forgetting curve*). Theo nghiên cứu này, con người có xu hướng quên dần thông tin theo thời gian nếu không được ôn lại. Tỷ lệ ghi nhớ giảm mạnh chỉ sau vài giờ và tiếp tục giảm theo hàm mũ.



Hình 3.1: Đường cong lãng quên (mô phỏng theo Ebbinghaus)

Nguồn hình ảnh: <https://cep.barnard.edu/studying-memory-comprehension>

Từ quan sát này, các thuật toán như SM-2 được phát triển nhằm mục tiêu "đặt lịch ôn lại" tại những thời điểm trước khi trí nhớ suy giảm đáng kể, giúp người học duy trì thông tin lâu dài với nỗ lực tối thiểu.

3.3.2 Nguyên lý SM-2 gốc

Thuật toán SM-2 hoạt động dựa trên điểm số đánh giá chất lượng ghi nhớ (gọi là **quality**, ký hiệu là q , từ 0 đến 5). Thang điểm này được định nghĩa từ mức 5 (hoàn hảo - trả lời đúng dễ dàng) đến mức 0 (hoàn toàn sai - không nhớ gì cả), với các mức trung gian bao gồm mức 4 (đúng nhưng cần suy nghĩ), mức 3 (đúng khó khăn), mức 2 (sai nhưng nhớ được đáp án khi được nhắc), và mức 1 (sai nhưng đáp án có vẻ quen thuộc).

Với mỗi lượt ôn tập, thuật toán tính lại ba thông số chính: khoảng thời gian chờ giữa các lượt tiếp theo (interval), số lần ôn liên tiếp thành công (repetitions) và chỉ số độ dễ (ease factor, viết tắt là EF). Chỉ số EF được giới hạn trong khoảng từ 1.1 đến 2.5, với giá trị khởi tạo là 2.5.

Công thức tính toán EF sau mỗi lần ôn tập được biểu diễn qua phương trình:

$$EF_{n+1} = EF_n + (0.1 - (5 - q) \cdot (0.08 + (5 - q) \cdot 0.02)) \quad (3.1)$$

Trong đó q là chất lượng câu trả lời và EF được giới hạn trong khoảng [1.1, 2.5]. Khi $q < 3$, thẻ được reset về trạng thái ban đầu và bắt đầu lại chu kỳ ôn tập.

Khoảng cách giữa các lần ôn (interval) được cập nhật theo nguyên tắc đặc biệt: lần đầu tiên là 1 ngày, lần thứ hai là 6 ngày, và từ lần thứ ba trở đi được tính theo công thức:

$$\text{interval}_{n+1} = \text{interval}_n \times EF \quad \text{với } n \geq 2 \quad (3.2)$$

3.3.3 Điều chỉnh SM-2 trong hệ thống

Thuật toán SM-2 vốn được thiết kế để tối ưu hóa quá trình ghi nhớ thông tin trong dài hạn. Tuy nhiên, để đáp ứng yêu cầu ôn tập nhanh và linh hoạt trong quá trình học từ vựng, hệ thống đã tiến hành điều chỉnh thuật toán SM-2 nguyên bản. Cụ thể, thay vì áp dụng khoảng cách ôn theo đơn vị ngày như trong phiên bản gốc, hệ thống chuyển sang sử dụng đơn vị phút, cho phép người học có thể nhanh chóng ôn lại chỉ sau 10 hoặc 15 phút kể từ lần học đầu tiên. Việc thay đổi đơn vị thời gian này giúp tăng cường khả năng ghi nhớ ngắn hạn và làm nền tảng cho việc củng cố trí nhớ dài hạn sau này.

Chi tiết các điều chỉnh đối với thuật toán SM-2 – bao gồm chuyển đổi đơn vị thời gian, cơ chế đánh giá chất lượng phản hồi, cách cập nhật chỉ số độ dễ, cách

tính thời gian ôn kế tiếp, cũng như việc xử lý các thẻ khó – đã được trình bày cụ thể tại mục 5.2. Những cải tiến này không chỉ giúp cá nhân hóa quá trình học mà còn hỗ trợ hệ thống thích ứng với khả năng ghi nhớ của từng người học.

3.3.4 So sánh với các thuật toán có chức năng tương tự

Ngoài SM-2, có thể xem xét các phương pháp thay thế khác nhau. Hệ thống Leitner phân loại thẻ theo hộp với mỗi hộp có tần suất ôn khác nhau, tuy nhiên phương pháp này không theo dõi độ dễ cụ thể và thiếu khả năng cá nhân hóa. Các phiên bản nâng cao như SM-5 và SM-18 sử dụng mô hình toán học phức tạp hơn nhưng thiếu tài liệu chi tiết công khai và không dễ điều chỉnh. Thuật toán FSRS của Anki [9] sử dụng mô hình hồi quy logistic với dữ liệu lớn, tuy nhiên độ phức tạp cao và yêu cầu tập dữ liệu lớn khiến nó không phù hợp với các ứng dụng nhỏ.

Thuật toán	Đặc điểm chính	Lý do không chọn
Leitner System	Phân loại thẻ theo hộp; mỗi hộp có tần suất ôn khác nhau	Không theo dõi độ dễ cụ thể, thiếu khả năng cá nhân hóa.
SM-5, SM-18	Phiên bản nâng cao của SM-2, mô hình toán học phức tạp hơn	Thiếu tài liệu chi tiết công khai, không dễ điều chỉnh.
Anki's FSRS	Sử dụng mô hình hồi quy logistic, dữ liệu lớn	Phức tạp, cần tập dữ liệu lớn, không phù hợp ứng dụng nhỏ.

Bảng 3.1: So sánh SM-2 với các thuật toán thay thế

Thuật toán SM-2, với sự điều chỉnh hợp lý, vẫn đảm bảo sự cân bằng giữa hiệu quả và tính đơn giản, phù hợp với hệ thống học từ vựng cỡ nhỏ đến trung bình.

3.4 Công cụ hỗ trợ phát triển và triển khai

Để đảm bảo quá trình phát triển hệ thống diễn ra hiệu quả, ổn định và có thể mở rộng trong tương lai, tác giả đã sử dụng một số công cụ hỗ trợ. Hệ thống sử dụng **Git** để quản lý phiên bản mã nguồn và **GitHub** làm nền tảng lưu trữ, theo dõi lịch sử phát triển. **Docker** được sử dụng để tạo môi trường phát triển nhất quán và hỗ trợ quá trình triển khai lên máy chủ một cách thuận tiện. **Postman** được sử dụng để kiểm thử và mô phỏng các API REST trong quá trình phát triển và kiểm tra backend.

Chương này đã trình bày các công nghệ và nền tảng được sử dụng trong quá trình phát triển hệ thống học từ vựng. Từng công nghệ được lựa chọn dựa trên sự phù hợp với yêu cầu chức năng và phi chức năng của hệ thống đã được phân tích trước đó. Ngoài ra, thuật toán SM-2 cũng đã được tìm hiểu, phân tích và điều chỉnh

để đáp ứng mục tiêu học tập hiệu quả, cá nhân hóa và phù hợp với hệ thống. Những nền tảng và thuật toán được trình bày trong chương này sẽ tiếp tục được triển khai chi tiết trong các Chương 4.

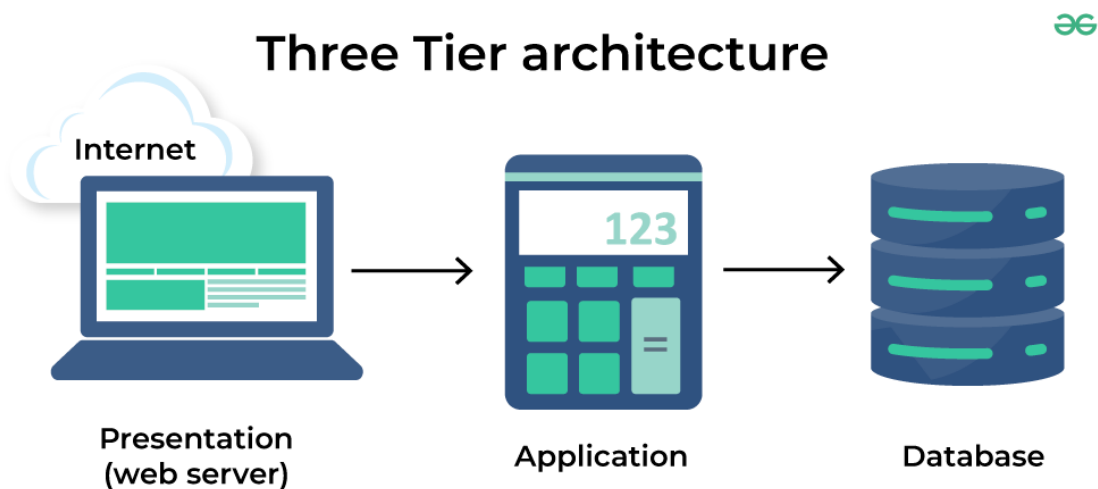
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Sau khi đã phân tích yêu cầu và xác định được các chức năng chính của hệ thống, chương này trình bày chi tiết quá trình thiết kế, triển khai và đánh giá hệ thống học từ vựng. Mục tiêu của chương là mô tả toàn diện cách xây dựng một hệ thống có thể hỗ trợ người học ghi nhớ từ vựng hiệu quả thông qua kỹ thuật lặp lại ngắt quãng (SR).

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Hệ thống được xây dựng theo kiến trúc phần mềm **đa tầng client-server** (multi-tier client-server architecture). Cụ thể, kiến trúc gồm ba tầng chính: **tầng trình diễn** (presentation tier), **tầng xử lý nghiệp vụ** (application tier) và **tầng dữ liệu** (data tier). Việc tổ chức theo kiến trúc này giúp phân tách rõ ràng giữa giao diện người dùng, logic nghiệp vụ và quản lý dữ liệu. Nhờ đó, hệ thống có khả năng mở rộng, dễ bảo trì, dễ kiểm thử và có thể tích hợp với các hệ thống khác trong tương lai.



Hình 4.1: Mô hình 3 tầng client-server

Nguồn hình ảnh: <https://www.geeksforgeeks.org/three-tier-client-server-architecture-in-distributed-system/>

Trong kiến trúc này, các thành phần frontend và backend được tách biệt hoàn toàn. Các client (ứng dụng web và ứng dụng di động) tương tác với backend thông qua giao thức HTTP và các API RESTful. Backend không đảm nhiệm việc sinh giao diện mà chỉ đóng vai trò xử lý logic và phản hồi dữ liệu (thường ở định dạng JSON) cho các client.

Tầng trình diễn của hệ thống bao gồm hai nền tảng chính: ứng dụng web sử dụng **ReactJS** và ứng dụng di động sử dụng **Flutter**. Ứng dụng web cung cấp giao diện người dùng hiện đại, thân thiện và dễ sử dụng trên trình duyệt, trong khi ứng dụng di động được triển khai trên hệ điều hành Android, đảm bảo sự thống nhất về giao diện và hiệu năng ổn định trên các thiết bị khác nhau. Các client này có vai trò hiển thị dữ liệu học tập như bộ thẻ, tiến độ ôn tập và thống kê cá nhân; đồng thời nhận các thao tác của người học như đánh giá mức độ ghi nhớ của từng thẻ. Dữ liệu sau đó được gửi đến server thông qua API, và phản hồi từ server được sử dụng để cập nhật giao diện tương ứng.

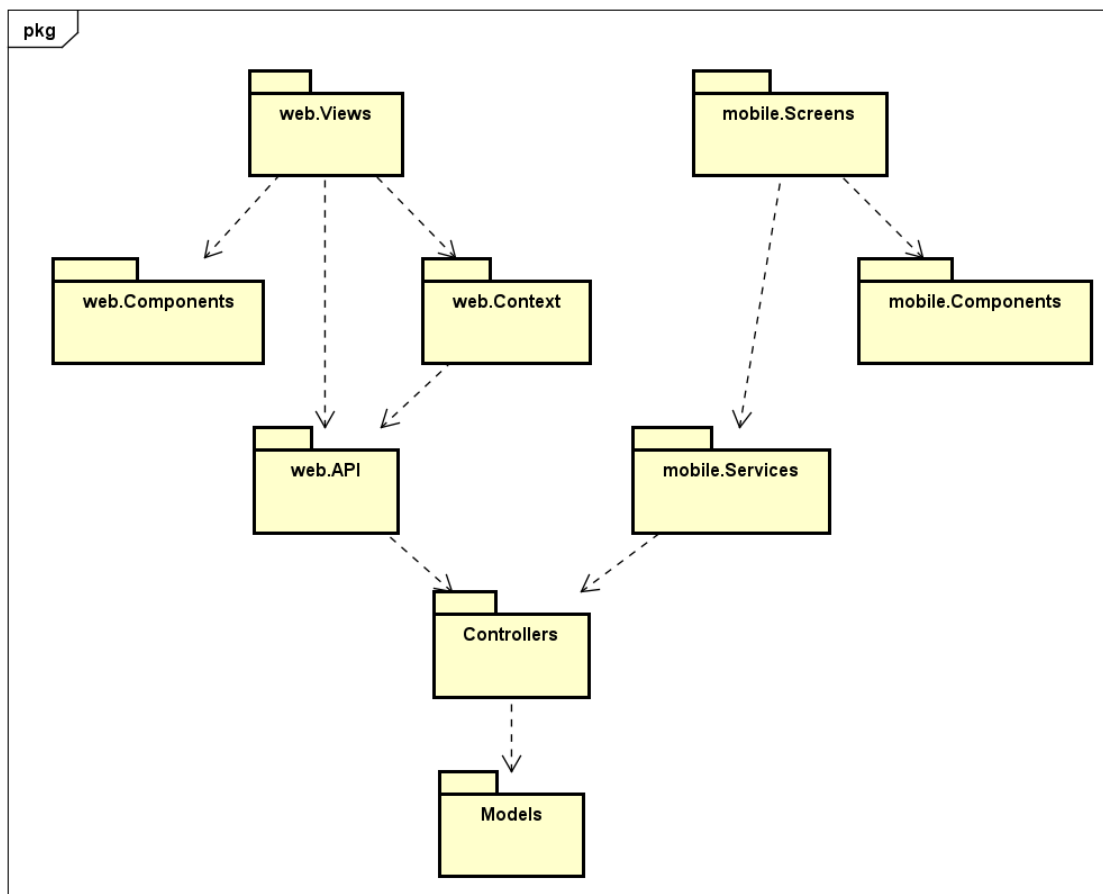
Tầng xử lý nghiệp vụ được xây dựng bằng **Laravel**. Tầng này đảm nhiệm toàn bộ xử lý logic nghiệp vụ và cung cấp API RESTful cho các client. Khác với mô hình MVC đầy đủ, hệ thống chỉ áp dụng mô hình **Model–Controller** trong **tầng xử lý nghiệp vụ** vì phần View được tách riêng hoàn toàn và thực thi ở **tầng trình diễn**. Cụ thể, phần **Model** biểu diễn các thực thể như người dùng, bộ thẻ, thẻ ghi nhớ, trạng thái ghi nhớ và lịch sử học tập, v.v. **Controller** đóng vai trò xử lý các yêu cầu từ client, như đăng nhập, đăng ký, quản lý thẻ từ, áp dụng thuật toán SM-2 để cá nhân hóa lịch ôn tập và trả về kết quả cho client. Hệ thống được tích hợp cơ chế xác thực sử dụng Laravel Sanctum nhằm đảm bảo an toàn cho các API riêng tư và bảo vệ dữ liệu người dùng.

Tầng dữ liệu của hệ thống sử dụng hệ quản trị cơ sở dữ liệu **MySQL**. Các bảng dữ liệu được thiết kế theo hướng chuẩn hóa, đảm bảo tính toàn vẹn và hiệu quả truy vấn. Các bảng chính bao gồm người dùng, bộ thẻ, các thẻ ghi nhớ, trạng thái học của từng người dùng với mỗi thẻ và lịch sử học tập chi tiết.

Việc áp dụng kiến trúc đa tầng client–server kết hợp với mô hình RESTful API đã cung cấp cho hệ thống một nền tảng vững chắc, linh hoạt và phù hợp với yêu cầu học tập đa nền tảng, hỗ trợ cá nhân hóa và mở rộng trong giai đoạn tiếp theo.

4.1.2 Thiết kế tổng quan

Hình 4.2 thể hiện biểu đồ phụ thuộc giữa các gói trong hệ thống, minh họa mối quan hệ giữa các thành phần chính trong kiến trúc tổng thể của toàn bộ hệ thống.



Hình 4.2: Biểu đồ phụ thuộc gói của hệ thống

Trong phần ứng dụng web, gói **web.Views** có vai trò quản lý các trang giao diện chính và định tuyến của ứng dụng. Gói này phụ thuộc vào **web.Components**, nơi định nghĩa các thành phần giao diện nhỏ có khả năng tái sử dụng cao như nút bấm, biểu mẫu hoặc các modal. Đồng thời, **web.Views** cũng sử dụng **web.Context** để truy xuất và quản lý trạng thái toàn cục như thông tin đăng nhập của người dùng, ngôn ngữ giao diện. Bên cạnh đó, các tương tác với hệ thống backend như lấy dữ liệu, xác thực người dùng hay cập nhật thông tin đều được thực hiện thông qua **web.API**. Gói **web.API** chứa các hàm gọi đến các endpoint được cung cấp bởi backend và phụ thuộc trực tiếp vào gói **Controllers** ở phía máy chủ.

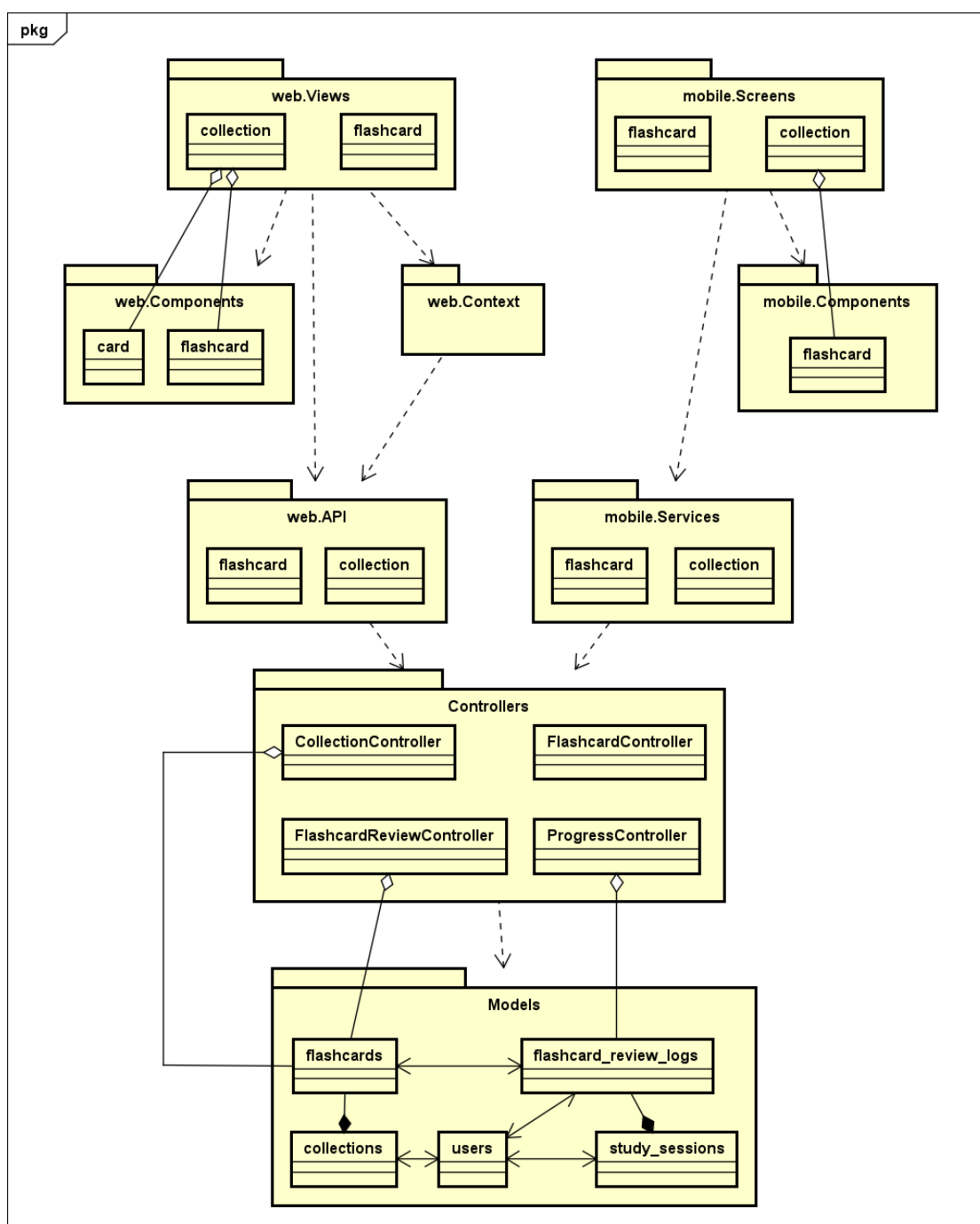
Về phía ứng dụng di động, **mobile.Screens** quản lý các màn hình hiển thị tương ứng với từng chức năng như màn hình chính, đăng nhập, v.v. Các màn hình này sử dụng **mobile.Components** để tổ chức giao diện và gọi đến **mobile.Services** nhằm xử lý các logic như gọi API, lưu trữ dữ liệu cục bộ hoặc phản hồi các thao tác từ người dùng. Tương tự như **web.API**, **mobile.Services** cũng phụ thuộc vào gói **Controllers** để giao tiếp với backend thông qua các yêu cầu HTTP.

Phần backend của hệ thống đóng vai trò xử lý các logic nghiệp vụ và truy xuất

dữ liệu. Gói **Controllers** nhận các yêu cầu từ cả ứng dụng web và ứng dụng di động, sau đó xử lý và chuyển tiếp đến gói **Models**. Gói **Models** có nhiệm vụ định nghĩa các mô hình dữ liệu, ánh xạ đến cơ sở dữ liệu thông qua các thao tác ORM, và cung cấp dữ liệu cần thiết để phản hồi cho client.

4.1.3 Thiết kế chi tiết gói

Biểu đồ trong Hình 4.3 là biểu đồ phụ thuộc gói, mô tả chi tiết các thành phần của hệ thống và cách chúng phối hợp để thực hiện chức năng ôn tập flashcard dựa trên thuật toán lặp lại ngắt quãng.



Hình 4.3: Biểu đồ gói chi tiết chức năng ôn tập flashcard

Khi người dùng truy cập vào tính năng ôn tập flashcard trên giao diện web hoặc ứng dụng di động, các thành phần thuộc **web.Views.collection** hoặc **mobile.Screens.collection** sẽ được hiển thị, đồng thời tương tác với các component như **web.Components.flashcard** hoặc **mobile.Components.flashcard** để lấy dữ liệu flashcard cần thiết. Các giao diện này gửi yêu cầu đến lớp **web.API.collection**, **web.API.flashcard** hoặc **mobile.Services.collection**, **mobile.Services.flashcard**, nơi đóng vai trò trung gian truyền dữ liệu giữa giao diện và lớp điều khiển phía máy chủ.

Sau khi tiếp nhận yêu cầu ôn tập từ phía người dùng, lớp **FlashcardReviewController** sẽ được gọi để xử lý logic. Tại đây, thuật toán SRS được triển khai nhằm tính toán các thông số như thời điểm ôn tập tiếp theo (**next_review_at**), khoảng thời gian giữa các lần ôn tập (**interval**), và chỉ số dễ nhớ (**ease_factor**). Các thông số này được cập nhật dựa trên phản hồi của người dùng sau mỗi lượt ôn tập (ví dụ: nhớ dễ, nhớ khó hoặc quên). Kết quả xử lý sẽ được ghi lại vào bảng **flashcard_review_logs**, đảm bảo hệ thống có thể theo dõi lịch sử học tập chi tiết của từng người dùng. Đồng thời, bảng **study_sessions** cũng được cập nhật nhằm phản ánh tiến độ học tập của người dùng theo từng phiên làm việc.

Tất cả dữ liệu liên quan đến flashcard, collection và người dùng được lưu trữ trong các bảng **flashcards**, **collections** và **users** thuộc gói **Models**. Các bảng này có mối quan hệ phụ thuộc chặt chẽ với nhau để đảm bảo rằng mỗi flashcard đều thuộc về một bộ sưu tập cụ thể và được gán cho một người dùng nhất định. Cuối cùng, thông tin đã xử lý sẽ được phản hồi lại cho giao diện thông qua API hoặc service tương ứng, từ đó cập nhật lại giao diện học tập và cho phép người dùng tiếp tục quá trình ôn luyện.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

a, Giao diện ứng dụng web

Hệ thống trên nền tảng web được thiết kế và kiểm thử với độ phân giải màn hình 1560×1080 pixel, trên màn hình có kích thước vật lý 15.6 inch. Với độ phân giải này, toàn bộ các thành phần giao diện được thiết kế để hiển thị rõ ràng, dễ đọc và có khả năng phản hồi tốt trên cả các trình duyệt hiện đại như Chrome, Microsoft Edge.

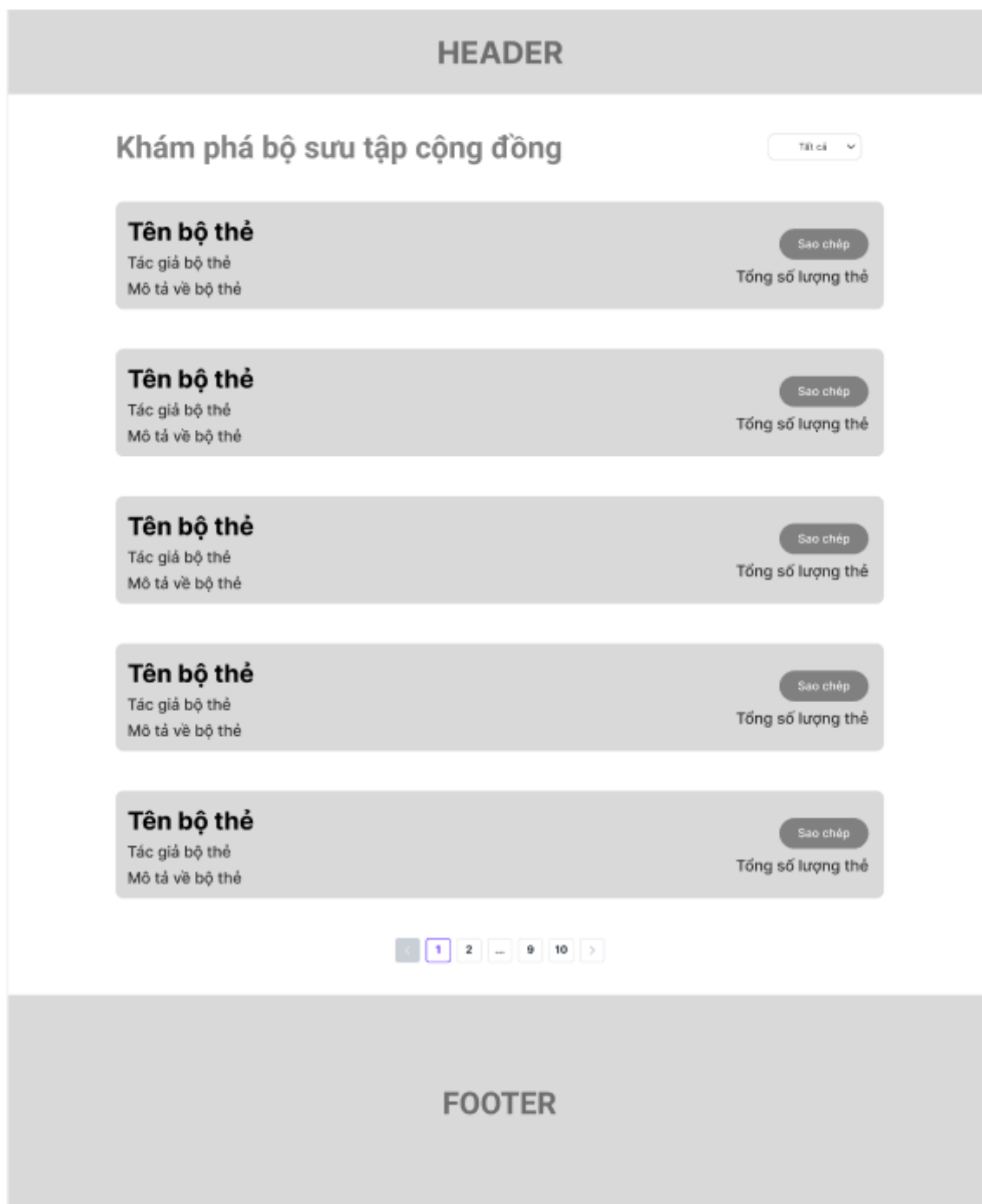
Giao diện web sử dụng hệ màu sáng chủ đạo. Tông màu nền chính là **#f1f2ff**, đem lại cảm giác dịu nhẹ, không gây mỏi mắt khi học tập lâu dài. Các thành phần nhấn như nút bấm sử dụng màu xanh dương **#166dba** và khi rê chuột sẽ chuyển sang **#8bbdee**, tạo hiệu ứng phản hồi trực quan mà vẫn giữ được sự nhất quán. Các

thông báo lỗi được làm nổi bật bằng màu đỏ #953333 để tăng khả năng nhận diện và giảm thiểu sai sót của người dùng.

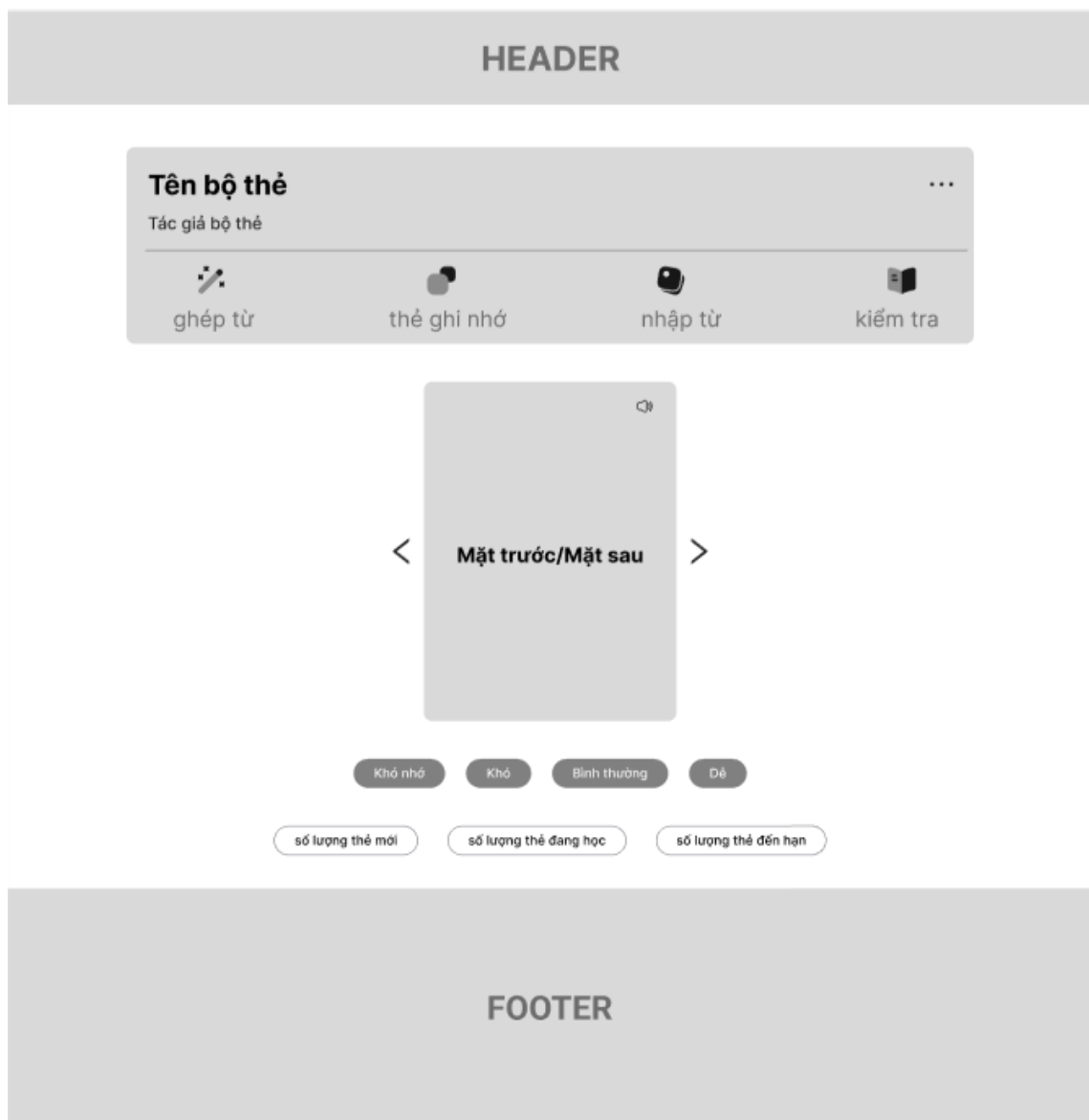
Toàn bộ văn bản được hiển thị với font chữ hệ thống (-apple-system, Segoe UI, Roboto, Helvetica Neue, Arial...) nhằm đảm bảo tương thích trên nhiều nền tảng và thiết bị khác nhau. Đối với phần tiêu đề thương hiệu, một font chữ riêng biệt là Just Me Again Down Here từ Google Fonts được sử dụng nhằm tạo ấn tượng trực quan và tăng khả năng nhận diện thương hiệu. Kích thước font được phân chia rõ ràng, từ 35px cho tiêu đề lớn, 28px cho tên bộ sưu tập, đến 18px cho nội dung văn bản và 14px cho phần mô tả ngắn, đảm bảo độ tương phản và phân cấp thị giác hợp lý.

Việc bố trí các thành phần giao diện tuân theo nguyên tắc thiết kế hiện đại: phân chia rõ ràng giữa phần điều hướng, nội dung chính, và thông báo phản hồi. Các nút bấm đều được bo tròn nhẹ với bán kính 5px hoặc 15px, tạo cảm giác thân thiện, dễ sử dụng. Các thông báo phản hồi (ví dụ khi đăng nhập thành công, tạo bộ thẻ thành công, v.v.) được hiển thị ở khu vực cố định phía trên giữa màn hình, giúp người dùng không bị gián đoạn dòng chảy sử dụng chính.

Hình ảnh minh họa một số màn hình giao diện quan trọng như trang danh sách bộ thẻ công khai, trang học bằng flashcard và trang thống kê tiến độ được trình bày tại Hình 4.4, Hình 4.5 và Hình 4.6.



Hình 4.4: Giao diện web: Trang danh sách bộ thẻ công khai



Hình 4.5: Giao diện web: Trang học bằng flashcard



Hình 4.6: Giao diện web: Trang thống kê tiến độ

b, Giao diện ứng dụng di động

Ứng dụng phiên bản di động được phát triển và kiểm thử trên thiết bị Samsung Galaxy A50, với kích thước màn hình 6.4 inch và độ phân giải 1080×2340 pixel.

Giao diện di động tuân thủ hệ màu giống với bản web nhằm giữ sự thống nhất nhận diện giữa các nền tảng. Mặc dù Android sử dụng font mặc định của hệ điều hành (Roboto hoặc Noto Sans), các thành phần giao diện vẫn được quy chuẩn về kích thước và độ tương phản, đảm bảo khả năng đọc tốt trong điều kiện ánh sáng khác nhau.

Thiết kế giao diện trên di động tuân theo nguyên tắc “đơn giản, tập trung, dễ thao tác”. Các nút bấm chính có kích thước lớn, dễ chạm với khoảng cách tối thiểu 48dp theo khuyến nghị của Material Design. Hệ thống sử dụng các thành phần điều khiển quen thuộc như Bottom Navigation để điều hướng giữa các chức năng chính:

Học tập, Bộ sưu tập, và Thống kê. Các thông báo trạng thái và phản hồi người dùng được hiển thị qua Snackbar hoặc Toast ở phần dưới, giữa màn hình, tránh gây nhiễu với nội dung chính.

Các màn hình học flashcard được tối ưu để thao tác bằng một tay. Người dùng có thể vuốt sang trái/phải để chuyển thẻ, hoặc chọn các mức độ ghi nhớ như "Khó nhớ", "Khó", "Bình thường", "Dễ" bằng nút bấm phía dưới. Giao diện phản hồi trực tiếp giúp người dùng nắm được kết quả ôn tập mà không cần rời khỏi màn hình chính.

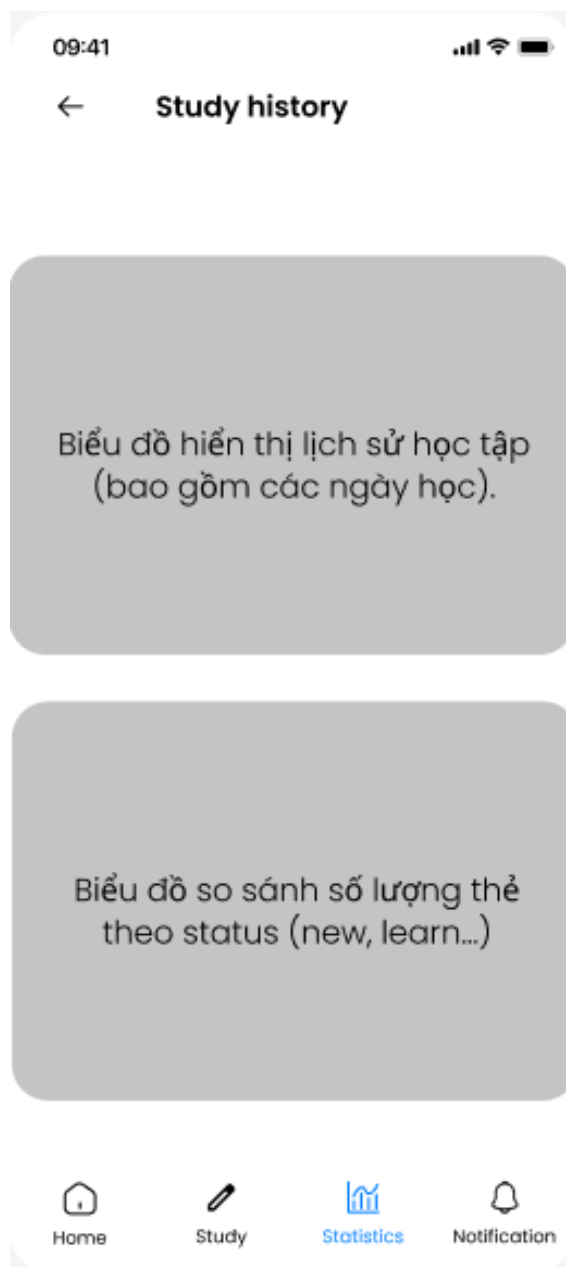
Một số hình ảnh minh họa cho các màn hình chính của ứng dụng di động được trình bày ở Hình 4.7, Hình 4.8, và Hình 4.9.



Hình 4.7: Giao diện mobile: Trang danh sách bộ thẻ công khai



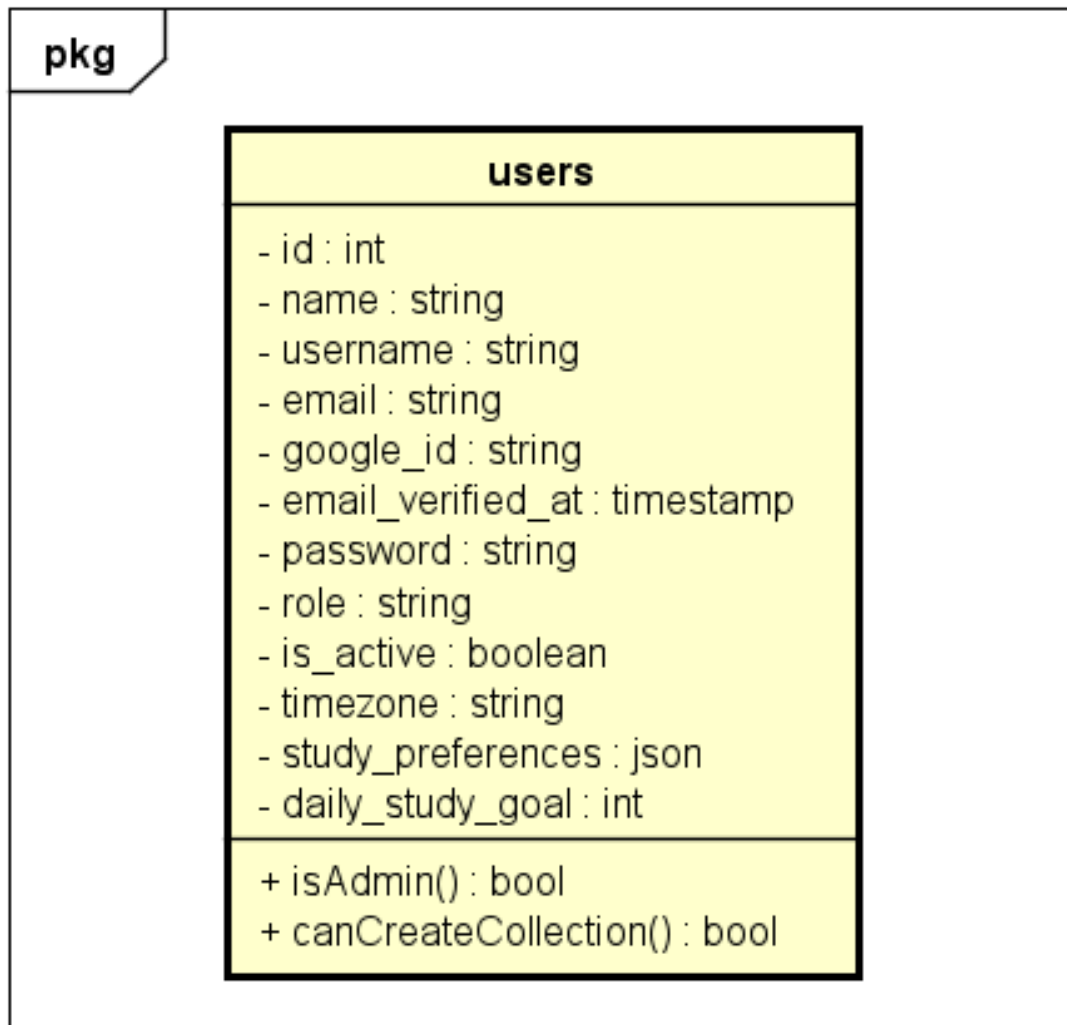
Hình 4.8: Giao diện mobile: Trang học bằng flashcard



Hình 4.9: Giao diện mobile: Trang thống kê tiến độ

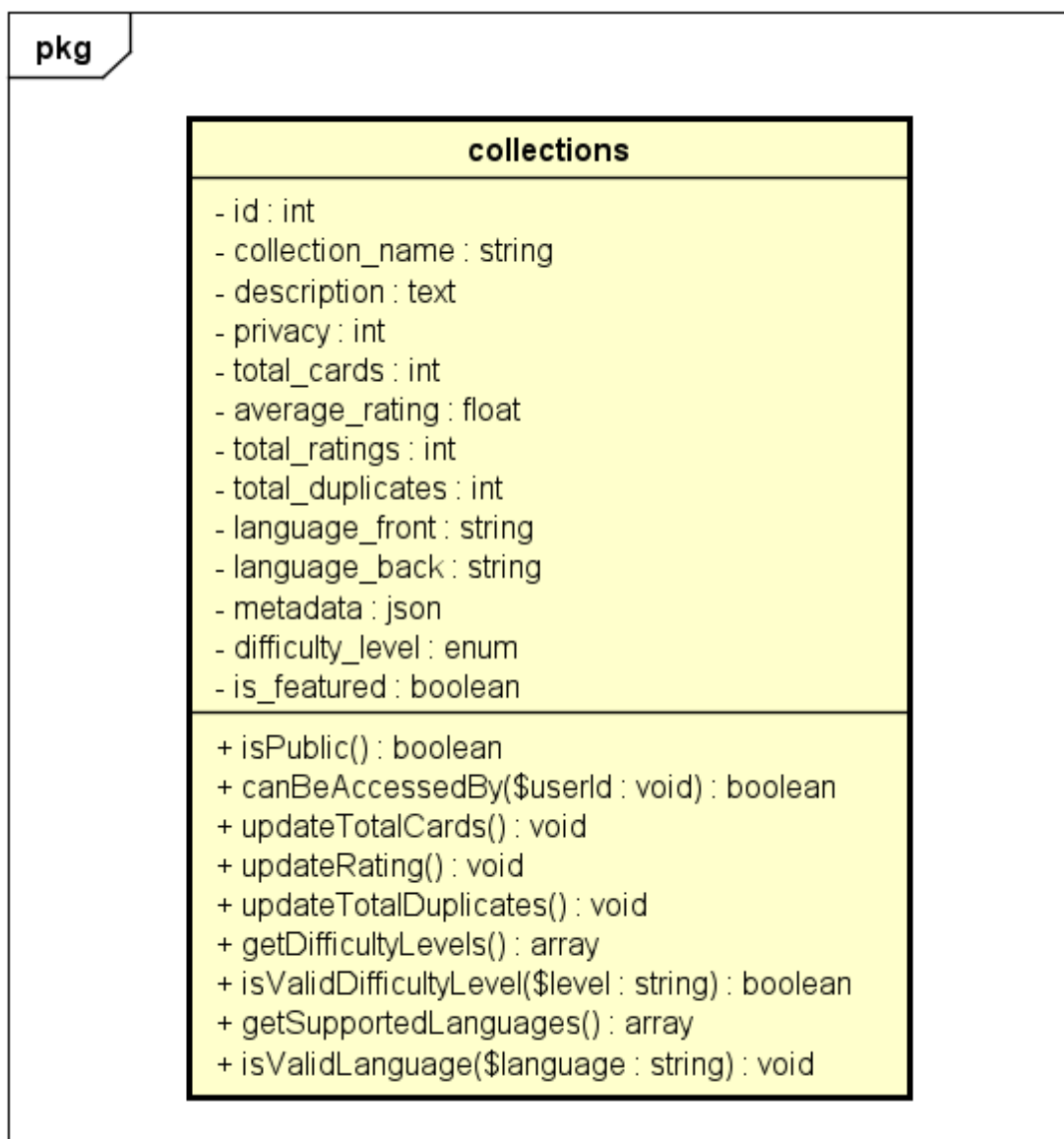
4.2.2 Thiết kế lớp

Trong phần này, bốn lớp được lựa chọn để trình bày thiết kế chi tiết là: users, collections, flashcards, và flashcard_statuses. Đây là những lớp có vai trò quan trọng, nhằm phục vụ cho quá trình học từ vựng theo mô hình lặp lại ngắt quãng.



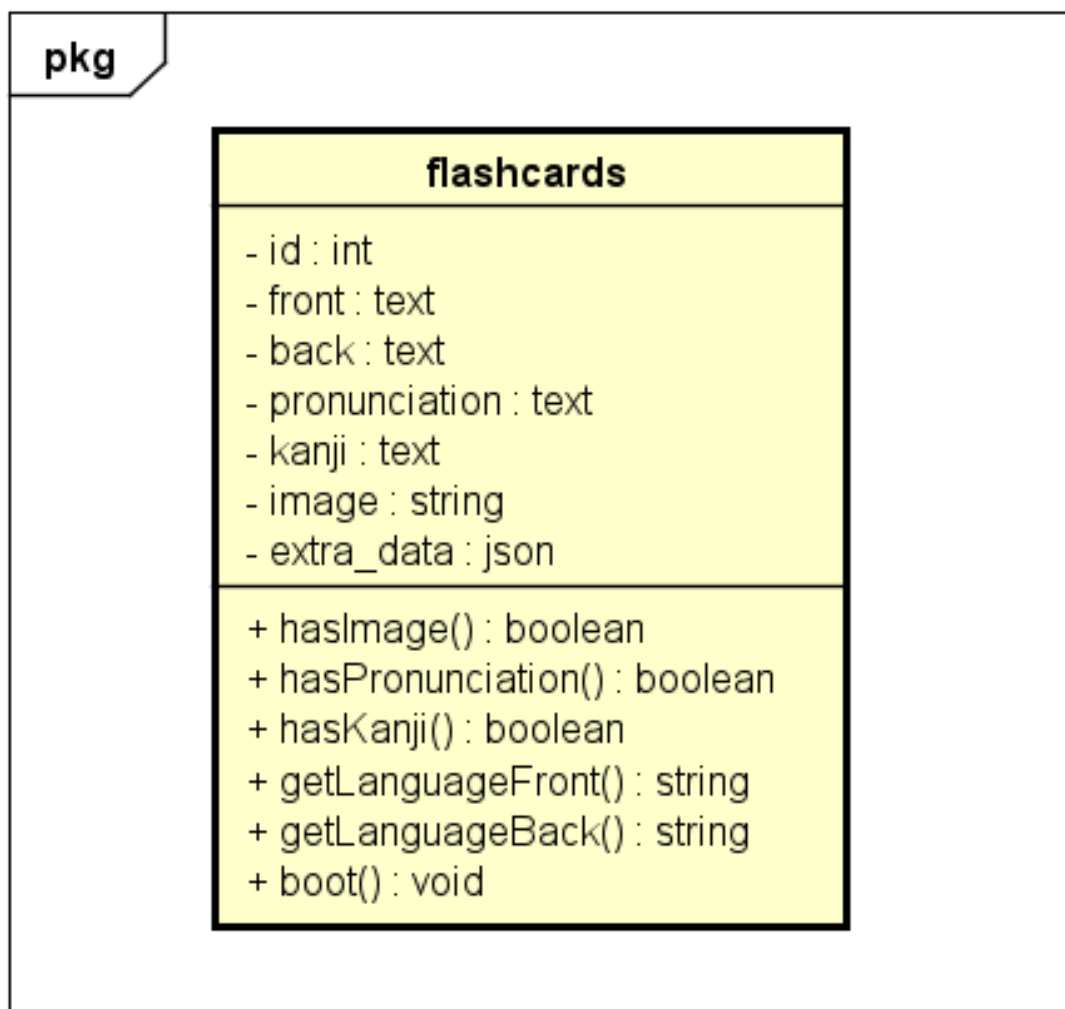
Hình 4.10: Lớp users

Lớp **users** đại diện cho người dùng trong hệ thống, chứa các thông tin như tên (name), tên đăng nhập (username), email, ID Google (nếu đăng nhập bằng Google), mật khẩu và vai trò (role). Ngoài ra, lớp còn hỗ trợ các thuộc tính mở rộng như múi giờ (timezone), sở thích học tập (study_preferences), và mục tiêu học hàng ngày (daily_study_goal). Các thuộc tính này giúp cá nhân hóa trải nghiệm học tập của từng người dùng. Phương thức `isAdmin()` trả về true nếu người dùng có vai trò là quản trị viên. Phương thức `canCreateCollection()` kiểm tra xem người dùng có được phép tạo thêm bộ sưu tập hay không, dựa trên giới hạn tối đa do cấp độ người dùng quy định. Nếu cấp độ cho phép tạo không giới hạn (giá trị -1), phương thức sẽ luôn trả về true. Ngược lại, hệ thống sẽ kiểm tra số lượng bộ sưu tập hiện tại của người dùng để đưa ra quyết định.

**Hình 4.11:** Lớp **collections**

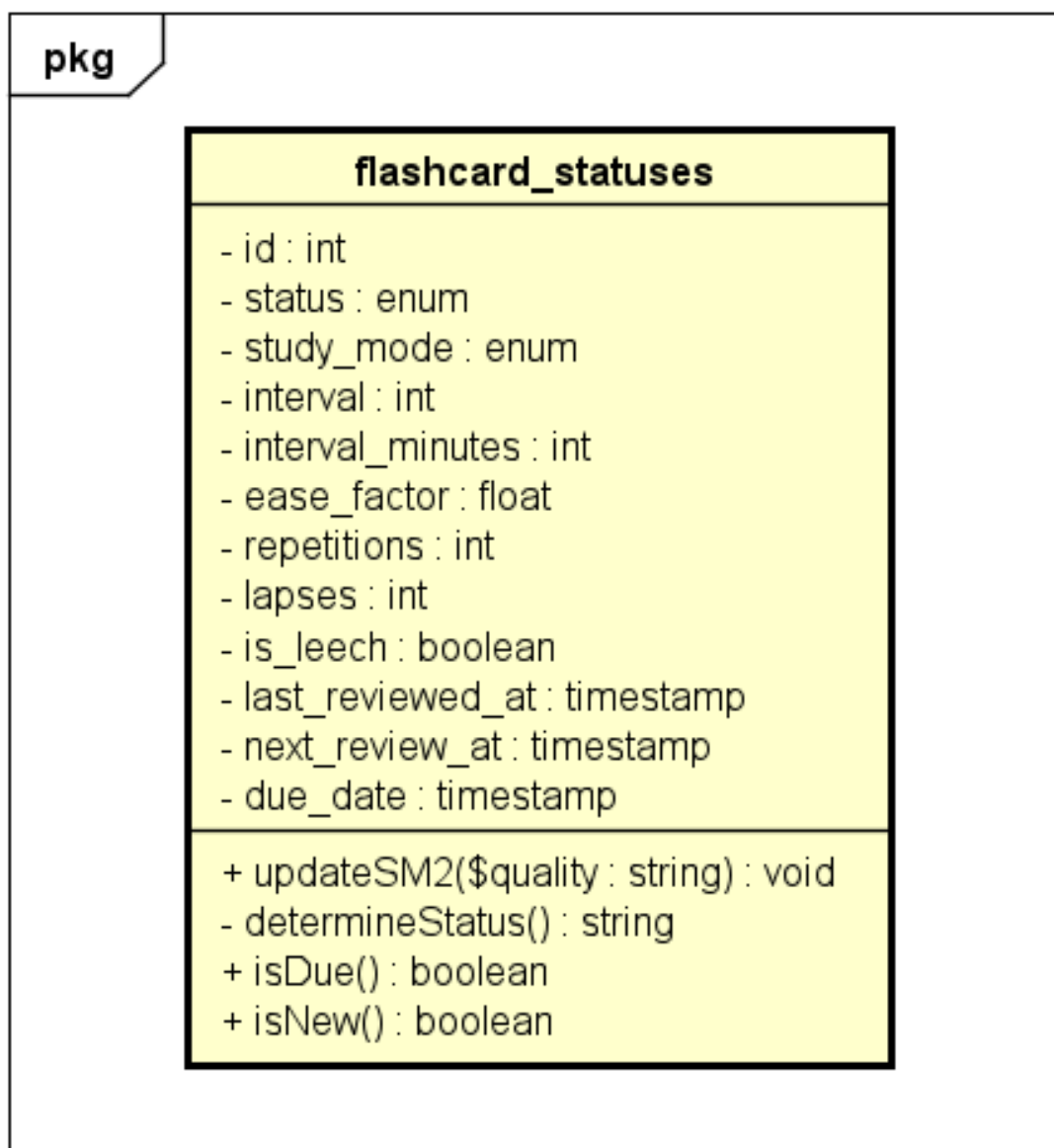
Lớp **collections** mô tả một bộ sưu tập flashcard, với các thuộc tính như: tên (collection_name), mô tả (description), chế độ hiển thị (privacy), tổng số flashcard (total_cards), số lần sao chép (total_duplicates), số lượt đánh giá (total_ratings) và điểm trung bình (average_rating). Bên cạnh đó, lớp còn lưu thông tin ngôn ngữ cho cả hai mặt của thẻ (language_front, language_back), cấp độ độ khó (difficulty_level), trạng thái nổi bật (is_featured), và thông tin bổ sung dưới dạng JSON (metadata). Phương thức `canBeAccessedBy($userId)` xác định xem người dùng có thể truy cập bộ sưu tập không, bằng cách kiểm tra quyền sở hữu hoặc chế độ hiển thị. Để quản lý thông tin thống kê, lớp có các phương thức như `updateTotalCards()`, `updateRating()` và `updateTotalDuplicates()`, lần lượt cập nhật tổng số thẻ, điểm đánh giá và số lượt sao chép bộ sưu tập. Ngoài ra, lớp hỗ trợ kiểm tra hợp lệ các mức độ khó và ngôn ngữ thông qua các phương thức

`getDifficultyLevels()`, `isValidDifficultyLevel($level)`, `getSupportedLanguages()` và `isValidLanguage($language)`.



Hình 4.12: Lớp flashcards

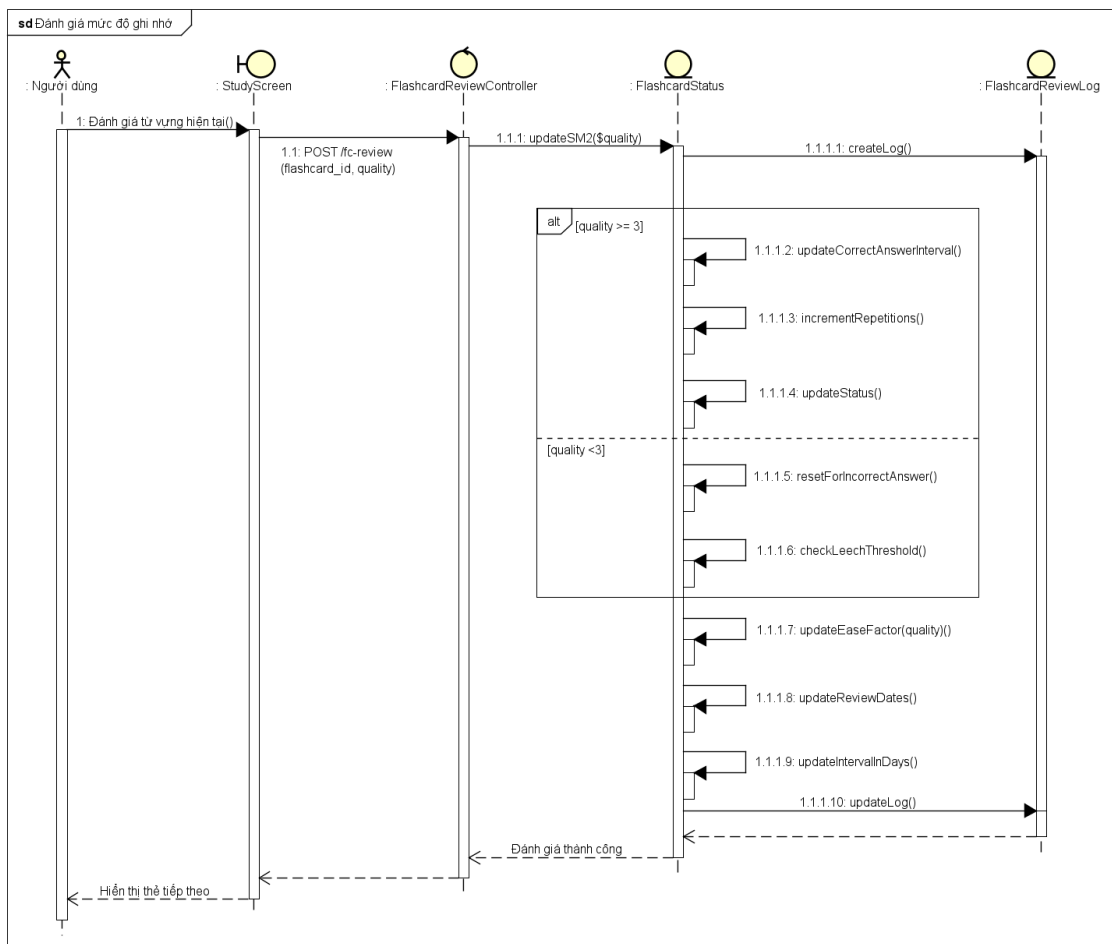
Lớp **flashcards** đại diện cho một thẻ học từ vựng, bao gồm nội dung mặt trước (front) và mặt sau (back), cách phát âm (pronunciation), Hán Việt (kanji), hình ảnh minh họa (image), và dữ liệu bổ sung dạng JSON (extra_data). Lớp này bao gồm các phương thức như `hasImage()`, `hasPronunciation()` và `hasKanji()` nhằm kiểm tra xem thẻ có chứa các thành phần tương ứng hay không. Hai phương thức `getLanguageFront()` và `getLanguageBack()` trả về ngôn ngữ của mặt trước và mặt sau của thẻ, kế thừa từ cài đặt của bộ sưu tập chứa thẻ đó. Lớp cũng định nghĩa các sự kiện trong phương thức `boot()`, trong đó khi một thẻ được tạo hoặc xóa, hệ thống sẽ tự động cập nhật lại tổng số thẻ của bộ sưu tập thông qua phương thức `updateTotalCards()` từ lớp **collections**.

**Hình 4.13:** Lớp **flashcard_statuses**

Lớp **flashcard_statuses** đóng vai trò quan trọng trong việc lưu trữ tiến độ học của người dùng đối với từng thẻ. Lớp này chứa các thuộc tính như trạng thái học hiện tại (status) với các giá trị như new, learning, re-learning, young, và mastered, cùng với chế độ học (study_mode) như học từ mặt trước sang mặt sau, ngược lại hoặc cả hai chiều. Các thuộc tính hỗ trợ cho thuật toán SM-2 bao gồm số lần lặp lại (repetitions), chỉ số dễ nhớ (ease_factor), khoảng thời gian giữa các lần ôn (interval tính theo ngày và interval_minutes tính theo phút), số lần quên (lapses), và đánh dấu thẻ khó (is_leech). Ngoài ra, lớp còn lưu lại thời điểm ôn gần nhất (last_reviewed_at), thời điểm ôn tiếp theo (next_review_at) và thời hạn ôn (due_date). Phương thức `updateSM2($quality)` là trung tâm của thuật toán, thực hiện tính toán lại thông số sau mỗi lần người dùng đánh giá chất lượng câu

trả lời. Dựa trên giá trị quality, phương thức xác định xem người dùng trả lời đúng hay sai và cập nhật lại khoảng cách ôn tập, chỉ số dễ nhớ, trạng thái, và cả việc đánh dấu thẻ là "leech" nếu lặp lại sai quá nhiều lần. Phương thức cũng ghi lại nhật ký học (**FlashcardReviewLog**) trước và sau khi cập nhật. Ngoài ra, phương thức `determineStatus()` được sử dụng để xác định trạng thái hiện tại của thẻ dựa trên số lần lặp lại và khoảng thời gian ôn. Hai phương thức hỗ trợ là `isDue()` để kiểm tra thẻ có đến hạn ôn chưa, và `isNew()` để kiểm tra thẻ có đang ở trạng thái mới không.

Việc tách riêng lớp **flashcards** và **flashcard_statuses** giúp đảm bảo phân tách rõ ràng giữa dữ liệu nội dung và dữ liệu tiến độ học tập. Lớp **flashcard_statuses** cho phép theo dõi tiến độ học riêng biệt của từng người dùng trên cùng một flash-card, tạo điều kiện thuận lợi cho việc học cộng đồng và chia sẻ bộ sưu tập công khai. Các chỉ mục được thêm vào để tối ưu hóa việc truy vấn các thẻ đến hạn học hoặc theo trạng thái học của người dùng.



Hình 4.14: Biểu đồ trình tự chức năng **Đánh giá mức độ ghi nhớ**

Hình 4.14 mô tả quá trình người dùng đánh giá mức độ ghi nhớ một thẻ từ sau khi học. Khi người dùng thực hiện đánh giá, hệ thống gửi thông tin lên server để

xử lý. Bộ điều khiển gọi hàm cập nhật thuật toán SM-2, khi đó hệ thống ghi lại log, tính toán khoảng thời gian ôn tập tiếp theo dựa trên chất lượng câu trả lời, cập nhật các thông số như số lần ôn, độ khó, trạng thái thẻ và thời gian ôn tiếp theo. Sau khi hoàn tất, hệ thống phản hồi kết quả để người dùng chuyển sang thẻ tiếp theo.

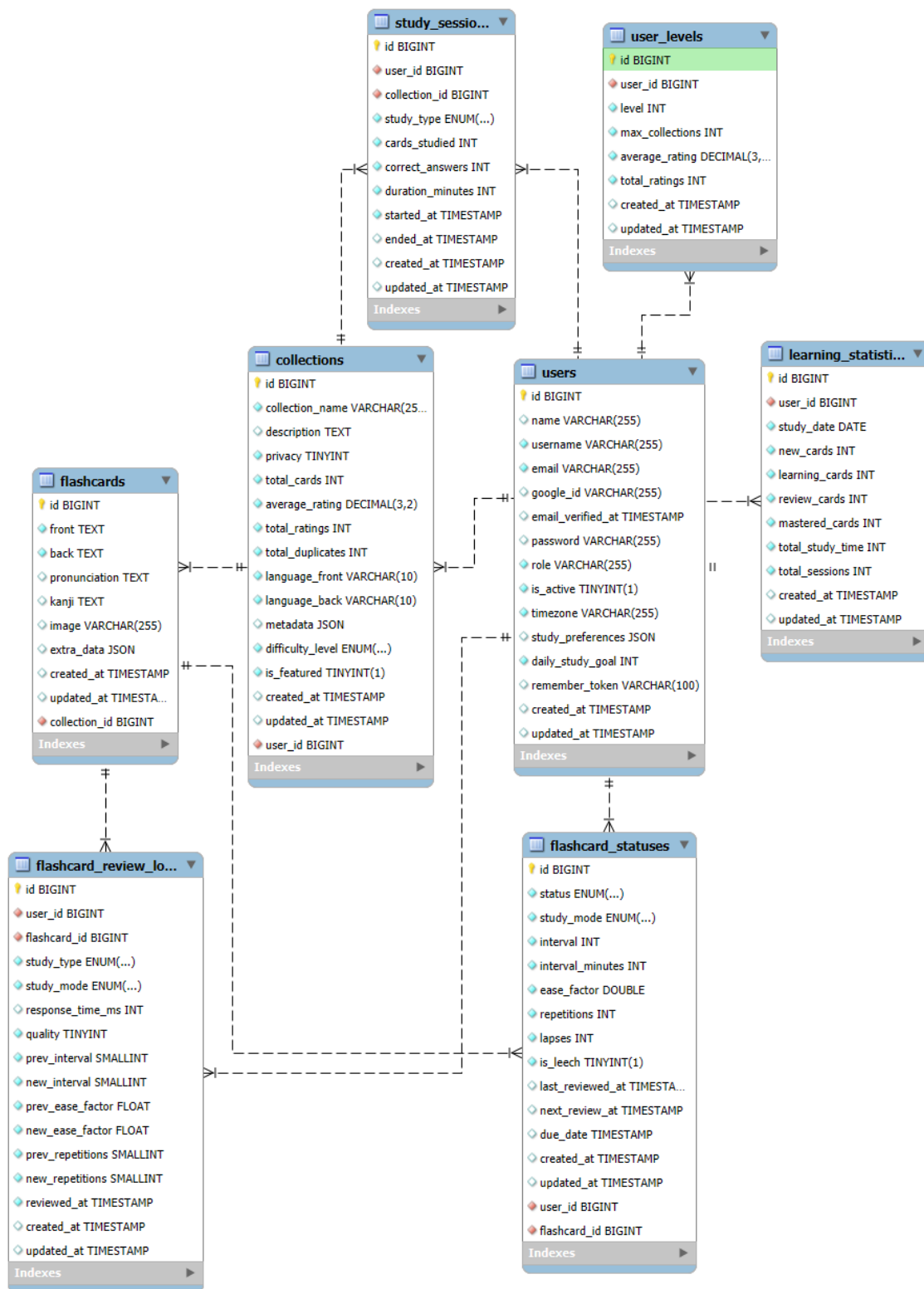
4.2.3 Thiết kế cơ sở dữ liệu

Trong hệ thống, cơ sở dữ liệu được thiết kế dựa trên hệ quản trị MySQL và tuân theo mô hình quan hệ, đảm bảo tính chuẩn hóa và toàn vẹn dữ liệu. Hình 4.15 là tập hợp các bảng cốt lõi, được lựa chọn nhằm phục vụ cho chức năng chính của hệ thống là hỗ trợ người dùng học từ vựng hiệu quả thông qua thuật toán lặp lại ngắt quãng.

Mỗi bảng có vai trò riêng biệt nhưng có mối liên hệ chặt chẽ thông qua các khóa chính và khóa ngoại để đảm bảo việc truy xuất và cập nhật dữ liệu được thực hiện chính xác. Tuy chưa thể hiện toàn bộ cơ sở dữ liệu của hệ thống, nhưng các bảng được đưa ra đã phản ánh đầy đủ các thành phần thiết yếu để triển khai chức năng học từ vựng theo mô hình cá nhân hóa, dựa trên việc tối ưu hóa khoảng cách lặp lại nhằm tăng cường khả năng ghi nhớ lâu dài cho người học.

Tên thực thể	Mục đích chính
users	Lưu thông tin người dùng
user_levels	Quản lý cấp độ người dùng dựa vào đánh giá trung bình
collections	Đại diện cho các bộ thẻ từ vựng
flashcard	Lưu thông tin từng thẻ từ vựng
flashcard_statuses	Lưu trạng thái học của từng thẻ từ cho từng người dùng
flashcard_review_logs	Ghi lại lịch sử đánh giá từng thẻ từ của người dùng
study_sessions	Thể hiện một phiên học của người dùng
learning_statistics	Thống kê tổng hợp hàng ngày quá trình học của người dùng

Bảng 4.1: Ý nghĩa từng thực thể



Hình 4.15: Biểu đồ thực thể liên kết (E-R diagram)

Dựa trên các thực thể chính được mô tả ở bảng 4.1, một số bảng dữ liệu tiêu biểu sẽ được phân tích chi tiết nhằm làm rõ vai trò và cấu trúc của các thành phần có liên quan trực tiếp đến quá trình xây dựng và triển khai thuật toán SM-2. Cụ thể, thiết kế chi tiết của bảng **collections** được trình bày tại bảng 4.2; bảng **flashcards** tại

bảng 4.3; bảng **flashcard_statuses** tại bảng 4.4; và bảng **flashcard_review_logs** tại bảng 4.5.

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	BIGINT	Khóa chính, định danh bộ thẻ
collection_name	STRING	Tên bộ thẻ
description	TEXT	Mô tả về bộ thẻ
privacy	TINYINT	Chế độ hiển thị của bộ thẻ (0 là công khai, 1 là riêng tư)
total_cards	INT	Tổng số thẻ
average_rating	DECIMAL	Điểm đánh giá trung bình của bộ thẻ
language_front	STRING	Ngôn ngữ mặt trước của thẻ
language_back	STRING	Ngôn ngữ mặt sau của thẻ
difficulty_level	ENUM	Mức độ khó của bộ thẻ (beginner, intermediate, advanced)
user_id	BIGINT	Người tạo bộ thẻ

Bảng 4.2: Thiết kế chi tiết bảng **collections**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	BIGINT	Khóa chính, định danh từng thẻ từ vựng
front	TEXT	Mặt trước của thẻ
back	TEXT	Mặt sau của thẻ
pronunciation	TEXT	Cách phát âm (nếu có)
kanji	TEXT	Hán viết (nếu có)
image	STRING	Đường dẫn hình ảnh minh họa (nếu có)
collection_id	BIGINT	Bộ thẻ của từ vựng

Bảng 4.3: Thiết kế chi tiết bảng **flashcards**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	BIGINT	Khóa chính, định danh...
status	ENUM	Trạng thái hiện tại của thẻ (new, learning, re-learning, young, mastered)
interval_minutes	INT	Khoảng thời gian lặp hiện tại
ease_factor	FLOAT	Chỉ số dễ ghi nhớ của từ vựng với từng người dùng
repetitions	INT	Số lần lặp lại đúng liên tiếp
lapses	INT	Số lần quên của thẻ từ vựng
is_leech	BOOL	Cờ đánh dấu thẻ hay bị quên
next_review_at	TIMESTAMP	Thời gian ôn tập kế tiếp
user_id	BIGINT	Người dùng đang học thẻ từ
flashcard_id	BIGINT	Thẻ tương ứng

Bảng 4.4: Thiết kế chi tiết bảng **flashcard_statuses**

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	BIGINT	Khóa chính, định danh...
user_id	BIGINT	Người dùng đang học thẻ từ
flashcard_id	BIGINT	Thẻ tương ứng
study_type	ENUM	Các phương pháp học (flashcard, nhập từ, ghép từ, kiểm tra)
quality	TINYINT	Mức độ ghi nhớ
prev_interval	SMALLINT	Khoảng thời gian lặp trước khi học
new_interval	SMALLINT	Khoảng thời gian lặp sau khi học
prev_ease_factor	FLOAT	Chỉ số dễ ghi nhớ của thẻ từ trước khi học
new_ease_factor	FLOAT	Chỉ số dễ ghi nhớ của thẻ từ sau khi học
reviewed_at	TIMESTAMP	Thời điểm đánh giá (khi học)

Bảng 4.5: Thiết kế chi tiết bảng **flashcard_review_logs**

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.6 liệt kê chi tiết các công cụ, ngôn ngữ lập trình, thư viện và môi trường phát triển đã được sử dụng của hệ thống.

Mục đích	Công cụ/Thư viện (kèm liên kết)
Ngôn ngữ lập trình Web	JavaScript (ReactJS 19.0.0)
Công cụ xây dựng dự án Web	Vite 6.2.0
Ngôn ngữ lập trình Backend	PHP 8.2 (Laravel 12.0.3)
Ngôn ngữ lập trình mobile	Dart 3.7.2 (Flutter 3.19.6)
Xác thực người dùng bằng Google	Google OAuth 2.0 & Google API
Thư viện giao diện người dùng Web	Ant Design (antd 5.24.4)
Hỗ trợ đa ngôn ngữ Web	i18next 24.2.3, react-i18next 15.4.1
Giao tiếp API Web	axios 1.8.4
Giao tiếp API mobile	http 1.4.0, dio 5.8.0+1
Web IDE	Visual Studio Code
Backend IDE	PhpStorm
Mobile IDE	Android Studio
Truy vấn cơ sở dữ liệu	MySQL Workbench
Kiểm thử API	Postman
Đóng gói Backend	Docker
Triển khai backend trên server	Amazon Web Services (AWS EC2)
Quản lý mã nguồn	Github
Vẽ biểu đồ	Astah UML
Thiết kế giao diện	Figma

Bảng 4.6: Danh sách thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Hệ thống học từ vựng đã được hoàn thiện với đầy đủ các chức năng cốt lõi. Hệ thống bao gồm ba thành phần chính là ứng dụng web, ứng dụng di động và hệ thống backend. Ứng dụng web hỗ trợ quản lý nội dung học tập và cung cấp giao diện học tập trực quan. Ứng dụng di động cho phép người dùng học mọi lúc mọi nơi. Hệ thống backend chịu trách nhiệm xử lý logic nghiệp vụ, lưu trữ và đồng bộ dữ liệu, đồng thời đảm bảo bảo mật thông qua cơ chế xác thực bằng Google.

Tiến độ triển khai các chức năng của hệ thống được phân chia theo ba nhóm đối tượng người dùng cụ thể. Đối với khách truy cập và người dùng, các chức năng được xây dựng cùng tiến độ thực hiện được thể hiện trong hình 4.16.

Role	Chức năng chính	Chức năng con	Platform	Tiến độ
Khách Người dùng	Quản lý tài khoản	Đăng ký	Web	100,00%
			App	100,00%
		Đăng nhập	Web	100,00%
			App	100,00%
		Quên mật khẩu (reset mật khẩu)	Web	100,00%
			App	100,00%
	Tìm kiếm và xem collection công khai	Tìm kiếm collection công khai	Web	100,00%
			App	100,00%
		Xem danh sách collection công khai	Web	100,00%
			App	100,00%
		Xem danh sách flashcard của collection công khai	Web	100,00%
			App	100,00%
	Đa ngôn ngữ		Web	100,00%
			App	100,00%

Hình 4.16: Tiến độ và chức năng dành cho khách truy cập và người dùng

Đối với người dùng đã đăng nhập, các chức năng dành cho nhóm đối tượng này được mô tả chi tiết trong Hình 4.17.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Role	Chức năng chính	Chức năng con	Platform	Tiến độ
Người dùng	Quản lý tài khoản	Đổi mật khẩu	Web	100,00%
			App	100,00%
		Cập nhật thông tin cá nhân	Web	100,00%
			App	100,00%
	Quản lý collection	Tạo collection mới	Web	100,00%
			App	100,00%
		Cập nhật thông tin collection	Web	100,00%
			App	100,00%
		Xóa collection	Web	100,00%
			App	100,00%
		Xem danh sách collection (user sở hữu)	Web	100,00%
			App	100,00%
		Sao chép collection công khai vào folder cá nhân	Web	100,00%
			App	100,00%
	Quản lý flashcard	Xem flashcards	Web	100,00%
			App	100,00%
		Thêm từ vựng - flashcard mới vào collection	Web	100,00%
			App	100,00%
	Học từ vựng	Xóa từ vựng - flashcard khỏi collection	Web	100,00%
			App	100,00%
		Học bằng flashcards - Lật thẻ	Web	100,00%
			App	100,00%
		Học - ghép từ	Web	100,00%
			App	100,00%
		Học - tự nhập nghĩa của từ	Web	100,00%
			App	100,00%
		Làm kiểm tra - test	Web	100,00%
			App	100,00%
	Thống kê, theo dõi quá trình học	Xem số lượng từ đã học, đang học, cần ôn tập	Web	100,00%
			App	100,00%
		Hiển thị biểu đồ tiến độ học tập	Web	100,00%
			App	100,00%
	Quản lý thông báo	Xem lịch sử học tập	Web	100,00%
			App	100,00%
		Nhận thông báo từ hệ thống: collection mới, ôn tập...	Web	100,00%
			App	100,00%
		Nhận thông báo từ user khác: đánh giá, duplicate collection, report collection	Web	100,00%
			App	100,00%

Hình 4.17: Tiến độ và chức năng dành cho người dùng đã đăng nhập

Đối với quản trị viên hệ thống, hình 4.18 thể hiện các chức năng cùng tiến độ hoàn thành dành cho nhóm người dùng này.

Role	Chức năng chính	Chức năng con	Platform	Tiến độ
Quản trị viên	Quản lý người dùng	Xem danh sách người dùng	Web	100,00%
		Tạo người dùng mới		100,00%
		Khóa/ mở khóa tài khoản người dùng		100,00%
		Xóa tài khoản người dùng		100,00%
	Quản lý thông báo	Gửi thông báo cho người dùng		100,00%
		Xem danh sách thông báo		100,00%
	Quản lý collection từ hệ thống	CRUD collection		100,00%
		CRUD từ vựng của collection - flashcard		100,00%

Hình 4.18: Tiến độ và chức năng dành cho quản trị viên hệ thống

Cuối cùng, bảng 4.7 dưới đây trình bày các thông số thống kê tổng quan về mã nguồn và sản phẩm đóng gói của hệ thống.

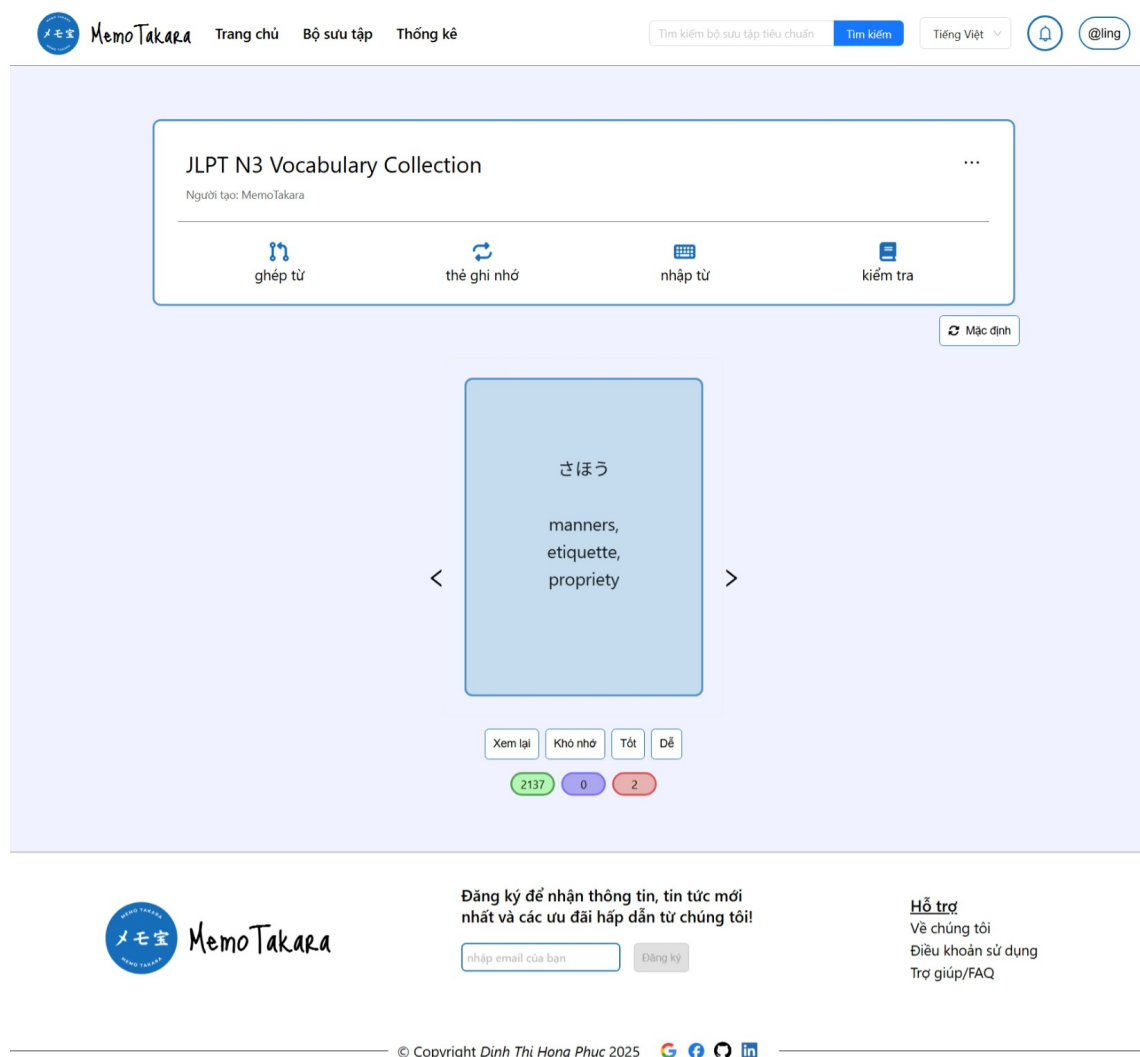
Thành phần	Thông số
Số lượng lớp (classes) trong toàn hệ thống	112 lớp
Số lượng gói (packages/modules)	34 gói
Dung lượng mã nguồn web	9.8 MB
Dung lượng mã nguồn mobile	11.2 MB
Dung lượng mã nguồn backend	15.5 MB
Dung lượng file APK mobile	29.3 MB
Sản phẩm web đóng gói (build)	12.7 MB

Bảng 4.7: Thống kê hệ thống

4.3.3 Minh họa các chức năng chính

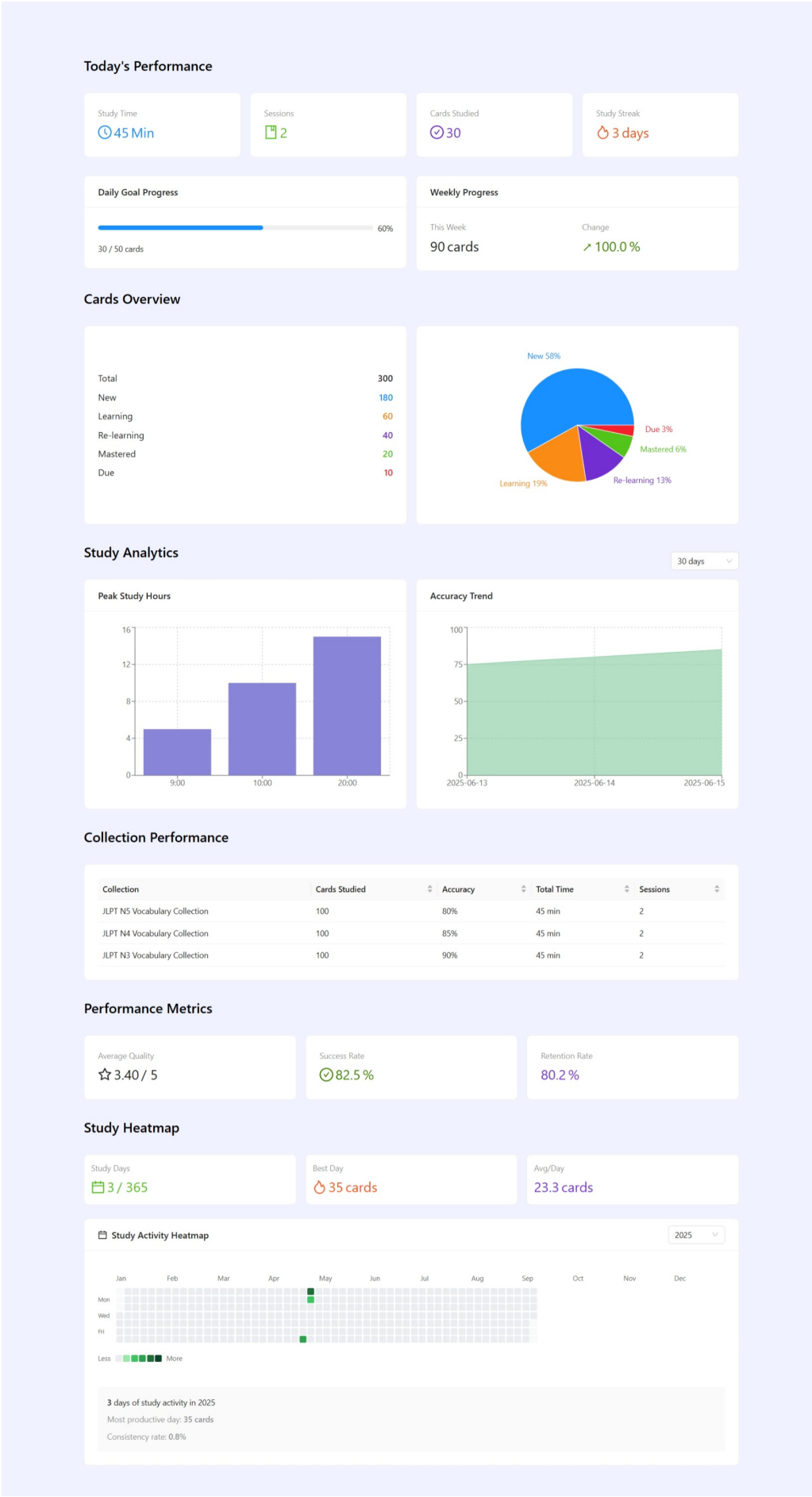
Trong phần này, hai chức năng quan trọng và nổi bật nhất của hệ thống sẽ được lựa chọn để minh họa, bao gồm: chức năng học từ vựng bằng flashcard và chức năng thống kê quá trình học tập. Các chức năng này không chỉ thể hiện rõ giá trị cốt lõi của hệ thống mà còn cho thấy sự khác biệt trong trải nghiệm người dùng trên nền tảng web và ứng dụng di động.

Hình 4.19 minh họa giao diện chức năng học từ vựng bằng flashcard trên nền tảng web. Giao diện này cho phép người dùng lật thẻ để xem nghĩa của từ, sau đó lựa chọn mức độ ghi nhớ theo bốn mức độ: “Xem lại”, “Khó nhớ”, “Tốt”, và “Đế”. Các lựa chọn này sẽ ảnh hưởng trực tiếp đến lịch trình ôn tập tiếp theo theo thuật toán SM-2.



Hình 4.19: Chức năng học từ vựng bằng flashcard (web)

Hình 4.20 trình bày giao diện thống kê học tập trên nền tảng web, cho phép người dùng theo dõi hiệu suất hàng ngày, tiến độ hàng tuần và trạng thái các thẻ từ vựng (mới, đang học, ôn lại, đã thành thạo, đến hạn). Giao diện bao gồm biểu đồ, bảng số liệu và bản đồ nhiệt, hỗ trợ trực quan trong việc đánh giá tiến trình học và điều chỉnh kế hoạch học tập một cách hiệu quả.



Hình 4.20: Chức năng thống kê quá trình học tập (web)

4.4 Kiểm thử

Phương pháp kiểm thử được áp dụng trong hệ thống là kiểm thử hộp đen (black-box testing). Các trường hợp kiểm thử dưới đây tập trung vào ba chức năng: học từ vựng bằng flashcard, thống kê quá trình học và xác thực người dùng bằng tài khoản Google. Việc kiểm thử được tiến hành trên cả hai nền tảng: web và ứng dụng di động nhằm đảm bảo tính ổn định và nhất quán của chức năng trên các môi trường triển khai khác nhau.

Tên chức năng	Đầu vào	Đầu ra mong đợi	Kết quả
Học flashcard (web)	Người dùng đăng nhập, chọn bộ sưu tập có flashcard, bắt đầu học và chọn mức đánh giá "Tốt"	Hệ thống ghi nhận mức đánh giá, cập nhật chỉ số ôn tập và hẹn lịch ôn tiếp theo đúng thuật toán SM-2	Đạt
Học flashcard (app)	Người dùng mở app, chọn bộ sưu tập và học flashcard, chọn mức đánh giá "Quên"	Hệ thống ghi nhận mức đánh giá, đưa thẻ vào trạng thái cần ôn lại sớm, cập nhật thống kê lượt sai	Đạt
Thống kê quá trình học (web)	Người dùng đã học 10 thẻ trong một bộ sưu tập	Giao diện thống kê hiển thị số lượng từ đã học, tỉ lệ trả lời đúng và biểu đồ tiến độ	Đạt
Thống kê quá trình học (app)	Người dùng hoàn thành một lượt học flashcard trong app	Trang thống kê hiển thị đúng thời gian, số lượng từ đã học hôm nay	Đạt
Đăng nhập bằng Google	Người dùng nhấn nút đăng nhập, chọn đúng tài khoản Google	Hệ thống xác thực thành công và chuyển hướng đến trang chính, tạo tài khoản nếu chưa tồn tại	Đạt
Đăng nhập thất bại	Người dùng nhập tài khoản không hợp lệ hoặc từ chối cấp quyền truy cập	Hệ thống hiển thị thông báo lỗi, không chuyển hướng	Đạt

Bảng 4.8: Các trường hợp kiểm thử chức năng chính

Kết quả kiểm thử cho thấy các chức năng trên đều hoạt động đúng như yêu cầu, cả về xử lý nghiệp vụ và giao diện người dùng. Trong toàn bộ quá trình kiểm thử, không có trường hợp nào thất bại ở giai đoạn triển khai chính thức. Một số lỗi nhỏ được phát hiện trong giai đoạn phát triển sớm đã được khắc phục và không còn

xuất hiện trong các lần kiểm thử cuối.

4.5 Triển khai

Ứng dụng web được xây dựng bằng ReactJS và đóng gói bằng Vite. Sau khi biên dịch, sản phẩm web được triển khai trên nền tảng Vercel, một dịch vụ đám mây chuyên dụng cho các ứng dụng frontend hiện đại. Địa chỉ truy cập chính thức của ứng dụng web là <https://memo-takara-fe-web.vercel.app/>. Việc sử dụng Vercel giúp tối ưu hóa tốc độ tải trang, hỗ trợ CI/CD, và tự động hóa quá trình triển khai mỗi khi có cập nhật mã nguồn.

Ứng dụng di động được phát triển bằng Flutter và biên dịch dưới dạng gói APK dành cho hệ điều hành Android. Việc phân phối ứng dụng được thực hiện thủ công thông qua cài đặt trực tiếp trên thiết bị trong quá trình phát triển và kiểm thử.

Thành phần backend được triển khai trên nền tảng Amazon Web Services (AWS), sử dụng dịch vụ EC2 với máy chủ cấu hình t2.medium (2 vCPU, 4 GB RAM) và hệ điều hành Ubuntu 22.04. Hệ thống backend được phát triển bằng Laravel, đóng gói thông qua Docker. Docker giúp đơn giản hóa quy trình triển khai, đảm bảo tính nhất quán của môi trường và hỗ trợ mở rộng linh hoạt trong tương lai.

Chương này đã trình bày toàn diện quá trình thiết kế và triển khai hệ thống học từ vựng, từ kiến trúc tổng thể đến thiết kế lớp, cơ sở dữ liệu và giao diện người dùng. Hệ thống sau khi hoàn thiện đã được kiểm thử và triển khai thành công, đáp ứng các tiêu chí về chức năng, hiệu suất và trải nghiệm người dùng. Chương tiếp theo sẽ tập trung phân tích các giải pháp kỹ thuật đã áp dụng và những đóng góp nổi bật của hệ thống.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này trình bày các đóng góp nổi bật trong quá trình phát triển hệ thống học từ vựng. Mỗi giải pháp đều xuất phát từ một vấn đề thực tiễn trong quá trình thiết kế và triển khai, sau đó được cụ thể hóa thành các cơ chế kỹ thuật hoặc thuật toán phù hợp.

5.1 Cá nhân hóa tiến độ học trên tài nguyên công khai

5.1.1 Giới thiệu vấn đề

Trong các hệ thống học trực tuyến hiện nay, việc chia sẻ tài nguyên học tập dưới dạng công khai là điều phổ biến. Người dùng có thể truy cập các bộ từ vựng, các tài liệu học tập do cộng đồng đóng góp, từ đó mở rộng phạm vi học tập và tiết kiệm thời gian tìm kiếm. Tuy nhiên, một hạn chế lớn thường gặp là các hệ thống này chỉ cho phép người học sử dụng tài nguyên công khai dưới dạng tham khảo. Nếu muốn tương tác với dữ liệu (chẳng hạn như lưu lại tiến độ học), người dùng thường phải sao chép toàn bộ nội dung về tài khoản cá nhân.

Cách tiếp cận này tuy đơn giản về mặt kỹ thuật nhưng lại mang đến nhiều hạn chế trong quản lý dữ liệu và trải nghiệm người dùng. Một mặt, nó tạo ra sự trùng lặp không cần thiết trong cơ sở dữ liệu, khi cùng một nội dung bị sao chép nhiều lần với các bản sao độc lập. Điều này không chỉ tiêu tốn tài nguyên lưu trữ mà còn gây khó khăn trong việc cập nhật hay bảo trì dữ liệu chung. Mặt khác, người dùng phải thực hiện thao tác thủ công để sao chép và chỉnh sửa, làm giảm tính liền mạch và linh hoạt trong trải nghiệm học tập.

Từ đó, bài toán đặt ra là: Làm thế nào để cho phép người học có thể tương tác trực tiếp với các bộ từ vựng công khai mà vẫn đảm bảo rằng mỗi người dùng có một tiến độ học riêng biệt, được cá nhân hóa theo lịch sử học và mức độ ghi nhớ cá nhân? Giải pháp cần đảm bảo đồng thời hai mục tiêu: tái sử dụng tối đa tài nguyên chung và theo dõi chi tiết quá trình học của từng cá nhân một cách độc lập.

5.1.2 Giải pháp

Để giải quyết vấn đề nêu trên, hệ thống được thiết kế theo hướng tách biệt giữa phần dữ liệu dùng chung và phần dữ liệu cá nhân hóa. Cụ thể, nội dung flashcard được lưu trong các bảng dữ liệu công khai như `collections` và `flashcards`. Đây là nơi chứa thông tin gốc về bộ từ vựng, bao gồm các trường như mặt trước (front), mặt sau (back), cách đọc (pronunciation), và các thuộc tính ngôn ngữ khác. Tất cả người dùng đều có thể truy cập và học từ các dữ liệu này mà không cần sao chép.

Để theo dõi quá trình học tập cá nhân, một bảng riêng có tên là `flashcard_statuses` được thiết kế. Mỗi bản ghi trong bảng này đại diện cho mối quan hệ giữa một người dùng cụ thể và một thẻ flashcard cụ thể, và chứa các thông tin đặc trưng cho quá trình học như: trạng thái học (ví dụ: mới, đang học, đã nhớ), chỉ số độ dễ (`ease_factor`), khoảng cách giữa hai lần ôn tập (`interval`), thời điểm ôn tiếp theo (`next_review_at`), và số lần lặp lại (`repetition`).

Chức năng của các trường này dựa trên thuật toán lặp lại cách quãng (`spaced repetition`), cụ thể là phiên bản điều chỉnh của SM-2. Chi tiết về kiến trúc này sẽ được trình bày trong phần 5.2. Việc áp dụng các tham số này cho phép hệ thống xây dựng một lộ trình ôn tập cá nhân hóa hoàn toàn cho từng người học, ngay cả khi họ học chung một bộ thẻ công khai.

Về mặt kỹ thuật, giải pháp này không chỉ đảm bảo tính độc lập giữa phần nội dung và phần trạng thái học, mà còn dễ dàng mở rộng theo hướng quy mô lớn. Mỗi bảng dữ liệu đều có các khóa ngoại hợp lý (`foreign keys`) và chỉ mục thích hợp (`indexes`) để tối ưu hóa truy vấn, đồng thời tránh trùng lặp và đảm bảo tính toàn vẹn dữ liệu. Hệ thống cũng xử lý tốt các tình huống như người dùng chưa từng học một flashcard (khi đó bản ghi trong `flashcard_statuses` sẽ được tạo mới), hoặc người dùng thay đổi bộ sưu tập trong khi vẫn giữ được lịch sử học cá nhân.

Giải pháp này giúp tạo nên một nền tảng học tập mở, nơi người dùng có thể học cùng nhau trên một bộ tài nguyên chung, nhưng mỗi người vẫn có một hành trình học tập riêng, được ghi nhận đầy đủ và chính xác. Nó thúc đẩy sự hợp tác trong chia sẻ tài nguyên và khuyến khích học tập cá nhân hóa theo năng lực và tiến độ của từng người học.

5.1.3 Kết quả đạt được

Việc áp dụng kiến trúc và thuật toán như đã trình bày mang lại nhiều lợi ích thực tiễn cho cả người dùng và nhà phát triển hệ thống. Người dùng có thể học ngay trên các bộ thẻ công khai mà không cần thực hiện thao tác sao chép, nhờ đó giảm tiêu tốn tài nguyên lưu trữ và tiết kiệm thời gian người dùng. Hệ thống vẫn đảm bảo lưu vết đầy đủ tiến độ học của từng người dùng với từng thẻ flashcard cụ thể.

5.2 Điều chỉnh thuật toán SM-2 để hỗ trợ ôn tập sớm

5.2.1 Giới thiệu vấn đề

Thuật toán SM-2 được thiết kế cho việc ghi nhớ dài hạn với các khoảng thời gian ôn tập tính theo ngày. Tuy nhiên, trong hệ thống này, thuật toán được điều chỉnh lại để người học có thể nhanh chóng ôn lại các từ vựng mới trong khoảng thời gian ngắn sau lần học đầu tiên. Điều này đòi hỏi một mô hình học linh hoạt hơn, cho

phép áp dụng SM-2 theo đơn vị thời gian nhỏ hơn, ví dụ như tính theo phút, trong các lần ôn đầu tiên.

5.2.2 Giải pháp

a, Chuyển đổi đơn vị thời gian từ ngày sang phút

Một trong những thay đổi đáng chú ý nhất là việc **chuyển đổi đơn vị thời gian** từ ngày sang phút. Với thuật toán gốc, các hệ thống sử dụng đơn vị ngày để ước lượng khoảng nghỉ giữa hai lần ôn tập, nhưng điều này làm mất tính linh hoạt trong các giai đoạn đầu, khi người học cần được nhắc lại nhanh chóng để củng cố trí nhớ ngắn hạn.

Trong hệ thống hiện tại, toàn bộ các thông số về khoảng thời gian ôn (*interval*) được tính bằng đơn vị phút và lưu trong trường *interval_minutes*. Ví dụ, sau lượt trả lời đúng đầu tiên, thẻ sẽ được đề xuất ôn lại sau khoảng 1 phút, và nếu tiếp tục trả lời đúng, thời gian này sẽ tăng lên theo cấp số nhân, giúp mô phỏng chính xác hơn quy luật quên tự nhiên. Việc này giúp hệ thống có thể đề xuất thời điểm ôn tập rất gần với thời điểm học ban đầu, chẳng hạn 10 phút sau khi ghi nhớ. Điều này đặc biệt hữu ích trong giai đoạn đầu, khi việc ôn lại ngay có thể giúp củng cố trí nhớ ngắn hạn và tăng khả năng ghi nhớ lâu dài.

b, Theo dõi trạng thái học của từng thẻ từ

Ngoài ra, hệ thống cũng bổ sung cơ chế **theo dõi trạng thái học** của từng thẻ từ để phản ánh chính xác tiến độ học tập của người dùng. Mỗi thẻ ghi nhớ trong hệ thống sẽ được gắn với một trong năm trạng thái: **new** (mới), **learn** (đang học), **relearn** (ôn lại), **young** (nhớ ngắn hạn) và **master** (ghi nhớ vững).

Trạng thái **new** được gán cho các thẻ từ mà người học chưa từng ôn tập. Đây là trạng thái khởi điểm của mọi thẻ, và thường được ưu tiên đưa vào phiên học đầu tiên để làm quen. Khi người học bắt đầu luyện tập với một thẻ mới, hệ thống sẽ chuyển trạng thái của nó sang **learn**, biểu thị rằng thẻ đang trong quá trình học ban đầu, chưa đủ điều kiện để được xem là ghi nhớ tạm thời.

Nếu sau khi đã ghi nhớ tạm thời mà người học trả lời sai, thẻ sẽ được chuyển sang **relearn**, tức là quay trở lại quá trình học ban đầu để được ôn lại từ đầu. Ngược lại, nếu người học liên tục trả lời đúng trong một khoảng thời gian ngắn, hệ thống sẽ chuyển thẻ sang **young** – trạng thái phản ánh rằng thẻ đã được người học ghi nhớ ở mức độ ngắn hạn, và hệ thống bắt đầu giãn cách thời gian giữa các lần ôn. Cuối cùng, nếu người học tiếp tục duy trì kết quả đúng sau nhiều lần ôn cách xa nhau, chẳng hạn như sau ba tuần, thẻ sẽ được nâng lên trạng thái **master**, thể hiện rằng nội dung đã được ghi nhớ vững trong trí nhớ dài hạn.

Sự chuyển đổi giữa các trạng thái phụ thuộc hai yếu tố: số lần người học ôn đúng liên tiếp (*repetitions*) và khoảng thời gian giữa các lần ôn đó (*interval_minutes*). Nhờ đó, hệ thống có thể phản ánh chính xác năng lực ghi nhớ thực tế của từng người học với từng thẻ cụ thể.

c, Theo dõi số lần trả lời sai và xác định thẻ khó

Song song với việc theo dõi trạng thái học, hệ thống cũng theo dõi **số lần trả lời sai** của người học đối với từng thẻ từ thông qua trường *lapses*. Việc này nhằm xác định chính xác những thẻ gây khó khăn trong quá trình ghi nhớ. Nếu một thẻ bị trả lời sai quá nhiều lần – chẳng hạn từ tám lần trở lên – hệ thống sẽ tự động đánh dấu thẻ đó là **thẻ khó**. Đây là một tín hiệu để người học biết rằng họ nên chú ý hơn đến thẻ này, đồng thời hệ thống cũng sẽ ưu tiên đưa nó vào các phiên ôn tập tiếp theo. Với những thẻ khó, người học có thể được khuyến nghị sử dụng các phương pháp hỗ trợ khác như xem hình ảnh minh họa, nghe phát âm chuẩn, hoặc học thông qua ngữ cảnh sử dụng thực tế, giúp cải thiện khả năng ghi nhớ một cách hiệu quả hơn.

d, Đánh giá mức độ ghi nhớ và điều chỉnh tiến trình học

Sau mỗi lượt ôn tập, người học sẽ **đánh giá mức độ ghi nhớ** của mình đối với thẻ vừa học bằng một thang điểm từ 0 đến 5 (*quality*). Dựa vào kết quả đánh giá này, hệ thống sẽ điều chỉnh tiến trình học của thẻ theo hai hướng khác nhau. Nếu người học **trả lời đúng** (tức là chọn mức điểm từ 3 trở lên), hệ thống sẽ tăng dần khoảng thời gian chờ trước lần ôn tiếp theo theo công thức:

$$I_n = \begin{cases} 1, & n = 10 \\ 6, & n = 15 \\ I_{n-1} \times EF, & n \geq 3 \end{cases} \quad (5.1)$$

Trong đó *EF* là chỉ số độ dễ hiện tại, và I_n là khoảng thời gian đến lần ôn tiếp theo (đơn vị: phút). Đồng thời, số lần lặp lại đúng liên tiếp (*repetitions*) được tăng thêm 1 đơn vị.

Ví dụ: Lần đầu tiên trả lời đúng sẽ dẫn đến một khoảng nghỉ ngắn, chẳng hạn 10 phút. Nếu tiếp tục trả lời đúng trong lượt kế tiếp, khoảng nghỉ sẽ được kéo dài hơn, chẳng hạn 15 phút. Từ lần đúng thứ ba trở đi, hệ thống sẽ tiếp tục tăng khoảng nghỉ theo mức độ phù hợp với khả năng ghi nhớ của người học (theo các thông số gì nêu ra). Đồng thời, hệ thống cũng ghi nhận số lượt ôn thành công liên tiếp, giúp xác định rõ ràng quá trình tiến bộ của người học.

Ngược lại, nếu người học **trả lời sai** (tức là chọn mức điểm dưới 3), hệ thống sẽ

coi như người học chưa thực sự ghi nhớ được thẻ đó. Khi đó, toàn bộ tiến trình ôn tập của thẻ sẽ được đặt lại từ đầu: số lượt ôn liên tiếp trở về 0 và khoảng thời gian chờ trước lần ôn tiếp theo được đặt lại về mức tối thiểu, ví dụ như 1 phút. Hệ thống cũng tăng thêm một lượt vào số liệu thống kê các lần sai để tiếp tục theo dõi mức độ khó của thẻ này. Nếu số lần trả lời sai tiếp tục tăng lên, thẻ đó sẽ được xếp vào danh sách các thẻ khó và được ưu tiên trong các phiên ôn tiếp theo.

e, Chỉ số độ dễ và tác động đến lịch ôn tập

Một yếu tố quan trọng trong việc xác định khoảng nghỉ giữa các lần ôn là **chỉ số độ dễ** (ease factor). Chỉ số này được cập nhật tự động sau mỗi lần ôn tập, tùy vào chất lượng phản hồi của người học. Nếu người học trả lời đúng một cách dễ dàng, chỉ số này sẽ tăng, làm kéo dài khoảng nghỉ ở lần ôn kế tiếp. Ngược lại, nếu người học trả lời sai hoặc gặp khó khăn khi nhớ lại, chỉ số sẽ giảm, khiến các lần ôn kế tiếp diễn ra sớm hơn. Công thức cập nhật chỉ số độ dễ như sau:

$$EF_{\text{new}} = \max(EF_{\text{min}}, EF_{\text{old}} + (0.1 - (5 - q) \cdot (0.08 + (5 - q) \cdot 0.02))) \quad (5.2)$$

Trong đó q là chất lượng phản hồi (quality) và EF là chỉ số độ dễ hiện tại. Hệ thống ràng buộc chỉ số này không được nhỏ hơn một ngưỡng tối thiểu, thường là $EF_{\text{min}} = 1.3$ (từ thuật toán SM-2 gốc), để tránh việc các **thẻ khó** bị lặp lại quá dày đặc gây quá tải cho người học. Việc điều chỉnh linh hoạt chỉ số độ dễ như vậy cho phép hệ thống phù hợp với từng thẻ từ và từng người học một cách cá nhân hóa. Các thẻ dễ sẽ được ôn với tần suất thưa hơn, còn các thẻ khó sẽ được đưa vào ôn lại thường xuyên hơn.

f, Lưu lịch sử học chi tiết

Cuối cùng, để đảm bảo khả năng phân tích và đánh giá hiệu quả học tập, hệ thống thiết kế bảng `flashcard_review_logs`, được sử dụng để lưu lại lịch sử học tập chi tiết theo từng phiên học. Mỗi bản ghi tại đây đại diện cho một lần người dùng ôn tập một thẻ flashcard và chứa đầy đủ các thông tin liên quan đến phương pháp học, phản ứng của người học và các thông số thuật toán SM-2.

Cụ thể, bảng này lưu trữ các trường như thời điểm học (`reviewed_at`), phương pháp học (`study_type`: flashcard, ghép từ, nhập từ, kiểm tra). Ngoài ra, thời gian phản hồi (`response_time_ms`) cũng được ghi lại để hỗ trợ phân tích mức độ tự tin và tốc độ phản xạ của người học.

Về phía thuật toán SM-2, bảng ghi nhận cả giá trị trước và sau của các thông số quan trọng như: khoảng cách giữa hai lần ôn (`prev_interval`, `new_interval`), chỉ số độ dễ (`prev_ease_factor`, `new_ease_factor`), và số lần lặp lại liên

tiếp thành công (`prev_repetitions`, `new_repetitions`). Chất lượng phản hồi (`quality`) được ghi nhận theo thang điểm 0–5, phản ánh mức độ ghi nhớ chủ quan của người học tại thời điểm ôn tập.

Việc lưu lại lịch sử học tập đầy đủ và có cấu trúc này không chỉ phục vụ mục tiêu đánh giá tiến độ và hiệu quả học tập của từng người dùng, mà còn là cơ sở để xây dựng các mô hình thống kê, phát hiện thói quen học tập, tối ưu thuật toán gợi ý.

5.2.3 Kết quả

Hệ thống giúp người học nhớ lại nội dung nhanh chóng trong giai đoạn đầu, hạn chế tình trạng quên nhanh sau khi học. Các mốc thời gian ôn tập được điều chỉnh phù hợp với mức độ ghi nhớ thực tế, tăng cường tính hiệu quả trong giai đoạn học ngắn hạn và vẫn đảm bảo được nguyên lý ghi nhớ lặp lại cách quãng trong dài hạn.

Chương này đã trình bày chi tiết các giải pháp kỹ thuật và đóng góp nổi bật trong quá trình thiết kế và triển khai hệ thống. Từ việc giải quyết bài toán học tập trực tiếp trên tài nguyên công khai mà không cần sao chép, cho đến việc cải tiến thuật toán SM-2 nhằm phù hợp với nhịp độ học linh hoạt và cá nhân hóa, các giải pháp đều được phát triển dựa trên nhu cầu thực tế của người học và định hướng tối ưu hoá hiệu quả ghi nhớ từ vựng. Chương tiếp theo sẽ tổng kết những kết quả đã đạt được và đề xuất một số hướng phát triển trong tương lai.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chương này trình bày kết quả tổng kết sau quá trình thực hiện đồ án tốt nghiệp. Đồng thời, chương cũng đề xuất các hướng phát triển trong tương lai nhằm hoàn thiện hệ thống và mở rộng phạm vi ứng dụng.

6.1 Kết luận

Trong quá trình thực hiện đồ án tốt nghiệp, hệ thống học từ vựng đã được thiết kế và phát triển với mục tiêu tạo ra một công cụ học tập tiện lợi, cá nhân hóa và hiệu quả cho người học. So với các hệ thống học từ vựng phổ biến hiện nay như Anki hay Quizlet, Hệ thống đã đạt được một số điểm nổi bật riêng. Cụ thể, hệ thống cho phép người dùng **học trực tiếp trên các bộ sưu tập công khai** mà không cần sao chép, giúp tiết kiệm thời gian, hạn chế trùng lặp dữ liệu và tạo điều kiện cho việc cộng tác học tập. Ngoài ra, việc **điều chỉnh thuật toán SM-2** theo hướng linh hoạt hơn, tính theo đơn vị phút trong các giai đoạn đầu ôn tập, giúp người học củng cố trí nhớ ngắn hạn và xây dựng nền tảng vững chắc cho việc ghi nhớ lâu dài. Các cải tiến này đã được trình bày chi tiết trong mục 5.2.

Trong suốt quá trình thực hiện, hệ thống đã được hoàn thiện với đầy đủ các chức năng học tập cơ bản, bao gồm tạo bộ sưu tập từ vựng, học và ôn theo mô hình lặp lại ngắt quãng, theo dõi tiến độ học tập, và lưu lịch sử hoạt động. Giao diện người dùng trên cả nền tảng web và di động cũng được xây dựng thân thiện và tối ưu cho trải nghiệm học.

Tuy nhiên, do hạn chế về thời gian và nguồn lực, một số tính năng nâng cao vẫn chưa được triển khai như đề xuất thông minh các bộ sưu tập phù hợp, đồng bộ hoá đa nền tảng theo thời gian thực, hay hỗ trợ học qua ngữ cảnh (context-based learning). Ngoài ra, việc đánh giá hiệu quả của hệ thống mới chỉ dừng lại ở kiểm thử chức năng và chưa tiến hành khảo sát trên quy mô lớn để đo lường tác động đến quá trình ghi nhớ thực tế của người học.

Những đóng góp nổi bật của đồ án bao gồm: thiết kế mô hình học trực tiếp trên tài nguyên công khai, cải tiến thuật toán SM-2 theo hướng linh hoạt và cá nhân hóa, xây dựng cơ sở dữ liệu phản ánh trạng thái học tập chi tiết của người dùng, và triển khai giao diện người dùng nhất quán giữa các nền tảng. Trong quá trình thực hiện, tôi cũng rút ra nhiều bài học kinh nghiệm về việc thiết kế hệ thống học tập phù hợp với nhu cầu người dùng, tầm quan trọng của việc kiểm thử và hiệu chỉnh thuật toán học tập, cũng như kỹ năng quản lý tiến độ trong các dự án phần mềm lớn.

6.2 Hướng phát triển

Trong thời gian tới, để hoàn thiện hệ thống, cần thực hiện một số công việc cụ thể như **mở rộng bộ dữ liệu từ vựng** thông qua đóng góp cộng đồng, **cải tiến giao diện người dùng** để hỗ trợ tốt hơn cho việc học trên thiết bị di động. Đồng thời, các tính năng quản lý nâng cao như **thống kê học tập** theo tuần/tháng và **tạo báo cáo học tập cá nhân** cũng nên được bổ sung để nâng cao tính thực tiễn.

Về định hướng lâu dài, hệ thống có thể được tích hợp thêm **trí tuệ nhân tạo** để đề xuất lộ trình học tập tối ưu dựa trên hành vi và kết quả học tập của từng cá nhân. Một hướng phát triển tiềm năng khác là **áp dụng học máy** để phân tích dữ liệu học tập nhằm phát hiện các xu hướng, từ đó tối ưu hoá thuật toán gợi ý và cải thiện hiệu quả học tập. Ngoài ra, việc hỗ trợ học theo ngữ cảnh, ví dụ như qua video, hình ảnh minh họa hoặc đoạn hội thoại thực tế, cũng là một bước tiến cần thiết để làm phong phú trải nghiệm học ngôn ngữ.

Cuối cùng, nếu được đầu tư và phát triển thêm, hệ thống hoàn toàn có thể trở thành một nền tảng học từ vựng mở rộng, phục vụ cho nhiều ngôn ngữ khác nhau và đáp ứng tốt hơn nhu cầu học tập đa dạng của người dùng trong kỷ nguyên số.

Tổng kết lại, đồ án đã xây dựng được một hệ thống học từ vựng phù hợp với nhu cầu thực tế, có khả năng thích ứng với người học thông qua các điều chỉnh thuật toán và mô hình lưu trữ linh hoạt. Dù còn một số hạn chế nhất định, kết quả đạt được đã đặt nền tảng vững chắc cho việc mở rộng và nâng cấp trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] N. Schmitt, “Instructed second language vocabulary learning,” *Language Teaching Research*, vol. 12, no. 3, pp. 329–363, 2008.
- [2] I. S. P. Nation, *Learning Vocabulary in Another Language*. United Kingdom: Cambridge University Press, 2013.
- [3] H. Ebbinghaus, *Memory: A Contribution to Experimental Psychology*. Dover Publications, 1885.
- [4] *Reactjs documentation*. [Online]. Available: <https://reactjs.org/docs/getting-started.html> (visited on 06/10/2025).
- [5] *Laravel documentation*. [Online]. Available: <https://laravel.com/docs> (visited on 06/10/2025).
- [6] *Flutter documentation*. [Online]. Available: <https://flutter.dev/docs> (visited on 06/10/2025).
- [7] *Mysql reference manual*. [Online]. Available: <https://dev.mysql.com/doc/> (visited on 06/10/2025).
- [8] P. A.Wozniak, *Optimization of learning*. [Online]. Available: <https://super-memory.com/english/ol.htm> (visited on 06/10/2025).
- [9] *Fsrs for anki*. [Online]. Available: <https://github.com/open-spaced-repetition/fsrs4anki/wiki> (visited on 06/10/2025).