**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC**

**CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG**

**ĐỀ TÀI: ỨNG DỤNG FLASHCARDS HỌC TỪ VỰNG**

**Giảng viên:** ThS. Nguyễn Thanh Truyền

**Sinh viên thực hiện:** Trần Minh Hợp

**MSSV:** 1050080180

**Lớp:** 10\_ĐH\_CNPM3

**Khóa:** 2021 - 2025

***TP. Hồ Chí Minh, tháng 5 năm 2025***

**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**CÔNG NGHỆ LẬP TRÌNH ĐA NỀN TẢNG**

**ĐỀ TÀI: ỨNG DỤNG FLASHCARDS HỌC TỪ VỰNG**

**Giảng viên:** ThS. Nguyễn Thanh Truyền

**Sinh viên thực hiện:** Trần Minh Hợp

**MSSV:** 1050080180

**Lớp:** 10\_ĐH\_CNPM3

**Khóa:** 2021 - 2025

***TP. Hồ Chí Minh, tháng 5 năm 2025***

**LỜI CẢM ƠN**

**LỜI CAM ĐOAN**

**NHẬN XÉT**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

………….., ngày….tháng….năm……

Giảng viên bộ môn

*(ký tên)*

**TÓM TẮT ĐỒ ÁN**

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN DỰ ÁN 1](#_Toc22380)

[1.1. TỔNG QUAN DỰ ÁN VÀ CÁC CÔNG VIỆC LIÊN QUAN 1](#_Toc12915)

[1.1.1. Các đề tài, dự án đã thực hiện 1](#_Toc28330)

[1.1.2. So sánh, đánh giá các công trình trước đây và sự khác biệt của đề tài 2](#_Toc30670)

[1.2. TỔNG QUAN VỀ DỰ ÁN 4](#_Toc29181)

[1.2.1. What – Dự án là gì? 4](#_Toc5581)

[1.2.2. Where – Dự án được triển khai ở đâu? 4](#_Toc14771)

[1.2.3. When – Dự án được thực hiện khi nào? 5](#_Toc3254)

[1.2.4. Why – Mục tiêu, lý do thực hiện dự án 5](#_Toc20860)

[1.2.5. How – Cách hoạt động và triển khai 6](#_Toc7148)

[CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN 9](#_Toc15299)

[2.1. YÊU CẦU KỸ THUẬT 9](#_Toc17022)

[2.1.1. Yêu cầu chức năng 9](#_Toc4220)

[2.1.2. Yêu cầu phi chức năng 11](#_Toc293)

[2.1.3. Yêu cầu về giao diện và trải nghiệm người dùng 12](#_Toc23468)

[2.2. THIẾT KẾ TỔNG QUAN HỆ THỐNG 12](#_Toc20800)

[2.2.1. Kiến trúc hệ thống 12](#_Toc253)

[2.2.2. Các thành phần chính của hệ thống 13](#_Toc8327)

[2.2.3. Mối quan hệ và luồng dữ liệu giữa các thành phần 14](#_Toc21783)

[2.2.4. Sơ đồ tổng quan hệ thống (System Overview Diagram) 15](#_Toc24466)

[2.3. SƠ ĐỒ USECASE 15](#_Toc25235)

[2.3.1. Các tác nhân (Actors) và vai trò 15](#_Toc24937)

[2.3.2. Mô tả các chức năng chính 15](#_Toc16328)

[2.3.3. Sơ đồ Use Case thể hiện tương tác người dùng với hệ thống 17](#_Toc31925)

[2.3.4. Sơ đồ Use Case con 17](#_Toc17348)

[2.4. SƠ ĐỒ LỚP (CLASS DIAGRAM) 22](#_Toc9030)

[2.4.1. Các lớp chính trong hệ thống 22](#_Toc7203)

[2.4.2. Mối quan hệ giữa các lớp 22](#_Toc2831)

[2.4.3. Thuộc tính và phương thức của từng lớp 22](#_Toc4785)

[2.5. SƠ ĐỒ TUẦN TỰ (SEQUENCE DIAGRAM) 22](#_Toc7115)

[2.6. LỰA CHỌN THIẾT KẾ VÀ CÔNG NGHỆ 28](#_Toc2287)

[2.6.1. Các công nghệ sử dụng 28](#_Toc11995)

[2.6.2. Phương án thiết kế kiến trúc hệ thống 28](#_Toc8441)

[2.6.3. Lý do lựa chọn các công nghệ và phương án thiết kế 28](#_Toc23341)

[2.7. THÁCH THỨC VÀ GIẢI PHÁP THUẬT TOÁN 28](#_Toc1627)

[CHƯƠNG 3. CÀI ĐẶT THỬ NGHIỆM, KẾT QUẢ VÀ ĐÁNH GIÁ 29](#_Toc5408)

[3.1. QUY TRÌNH TRIỂN KHAI HỆ THỐNG 29](#_Toc7898)

[3.1.1. Môi trường và công cụ phát triển 29](#_Toc27328)

[3.1.2. Quy trình cài đặt và cấu hình ban đầu 29](#_Toc27198)

[3.1.3. Từng bước thực hiện các thành phần hệ thống 29](#_Toc9951)

[3.1.4. Khó khăn và thách thức trong quá trình thực hiện 29](#_Toc20889)

[3.2. CẤU TRÚC VÀ CHỨC NĂNG HỆ THỐNG 29](#_Toc14790)

[3.2.1. Cấu trúc tổng thể của hệ thống phần mềm 29](#_Toc27043)

[3.2.2. Các chức năng chính đã triển khai 29](#_Toc31889)

[3.2.3. Mô tả các module chính và vai trò của từng module 29](#_Toc2340)

[3.2.4. Giao diện người dùng và tương tác hệ thống 29](#_Toc19588)

[3.3. KẾT QUẢ THỬ NGHIỆM 29](#_Toc23429)

[3.4. ĐÁNH GIÁ VÀ PHÂN TÍCH KẾT QUẢ 29](#_Toc9681)

[3.4.1. Mức độ đáp ứng yêu cầu kỹ thuật đã đặt ra 29](#_Toc6391)

[3.4.2. Phân tích độ chính xác và hiệu quả 29](#_Toc24988)

[3.4.3. Ưu điểm của hệ thống 29](#_Toc6359)

[3.4.4. Hạn chế hiện tại và hướng cải thiện trong tương lai 29](#_Toc6485)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 30](#_Toc22588)

[KẾT LUẬN 30](#_Toc406)

[HẠN CHẾ 30](#_Toc7624)

[HƯỚNG PHÁT TRIỂN 30](#_Toc24866)

[TÀI LIỆU THAM KHẢO 31](#_Toc27958)

**DANH MỤC SƠ ĐỒ, HÌNH**

*[Hình](#_Toc24953)* [1](#_Toc24953) *[. Sơ đồ luồng tương tác giữa các đối tượng](#_Toc24953)* [15](#_Toc24953)

*[Hình](#_Toc18919)* [2](#_Toc18919) *[. Sơ đồ tổng quan hệ thống](#_Toc18919)* [17](#_Toc18919)

*[Hình](#_Toc2810)* [3](#_Toc2810) *[. Sơ đồ Usecase tổng quát](#_Toc2810)* [20](#_Toc2810)

*[Hình](#_Toc25091)* [4](#_Toc25091) *[. Sơ đồ Usecase Nhập tiêu chí](#_Toc25091)* [20](#_Toc25091)

*[Hình](#_Toc11295)* [5](#_Toc11295) *[. Sơ đồ Usecase Nhập phương án](#_Toc11295)* [21](#_Toc11295)

*[Hình](#_Toc11950)* [6](#_Toc11950) *[. Sơ đồ Usecase Tạo ma trận so sánh](#_Toc11950)* [21](#_Toc11950)

*[Hình](#_Toc29866)* [7](#_Toc29866) *[. Sơ đồ Usecase Xem kết quả xếp hạng](#_Toc29866)* [22](#_Toc29866)

*[Hình](#_Toc28524)* [8](#_Toc28524) *[. Sơ đồ Lớp](#_Toc28524)* [22](#_Toc28524)

*[Hình](#_Toc16098)* [9](#_Toc16098) *[. Sơ đồ tuần tự](#_Toc16098)* [24](#_Toc16098)

*[Hình](#_Toc25676)* [10](#_Toc25676) *[. Sơ đồ tuần tự Nhập tiêu chí](#_Toc25676)* [25](#_Toc25676)

*[Hình](#_Toc31729)* [11](#_Toc31729) *[. Sơ đồ tuần tự Nhập phương án](#_Toc31729)* [26](#_Toc31729)

*[Hình](#_Toc15725)* [12](#_Toc15725) *[. Sơ đồ tuần tự Tạo ma trận so sánh](#_Toc15725)* [26](#_Toc15725)

*[Hình](#_Toc16267)* [13](#_Toc16267) *[. Sơ đồ tuần tự Xem kết quả xếp hạng](#_Toc16267)* [27](#_Toc16267)

*[Hình](#_Toc9153)* [14](#_Toc9153) *[. Sơ đồ kiến trúc với REST API & các tầng công nghệ](#_Toc9153)* [28](#_Toc9153)

*[Hình](#_Toc11984)* [15](#_Toc11984) *[. Công thức chuẩn hóa ma trận](#_Toc11984)* [32](#_Toc11984)

*[Hình](#_Toc16145)* [16](#_Toc16145) *[. Công thức Tính vector trọng số](#_Toc16145)* [32](#_Toc16145)

*[Hình](#_Toc22132)* [17](#_Toc22132) *[. Công thức Tính chỉ số nhất quán CI](#_Toc22132)* [32](#_Toc22132)

*[Hình](#_Toc23874)* [18](#_Toc23874) *[. Công thức tính Tỷ lệ nhất quán CR](#_Toc23874)* [33](#_Toc23874)

*[Hình](#_Toc11293)* [20](#_Toc11293) *[. Giao diện thêm, xóa tiêu chí](#_Toc11293)* [40](#_Toc11293)

*[Hình](#_Toc7894)* [21](#_Toc7894) *[. Giao diện thêm, xóa phương án](#_Toc7894)* [40](#_Toc7894)

*[Hình](#_Toc16140)* [22](#_Toc16140) *[. Giao diện nhập phương án và so sánh theo tiêu chí](#_Toc16140)* [41](#_Toc16140)

*[Hình](#_Toc19638)* [23](#_Toc19638) *[. Giao diện đánh trọng số](#_Toc19638)* [42](#_Toc19638)

*[Hình](#_Toc23552)* [24](#_Toc23552) *[. Dựa trên tiêu chí, trọng số và điểm vị trí đề xuất các vị trí](#_Toc23552)* [43](#_Toc23552)

*[Hình](#_Toc15150)* [25](#_Toc15150) *[. Sơ đồ cột](#_Toc15150)* [43](#_Toc15150)

*[Hình](#_Toc10327)* [26](#_Toc10327) *[. Sơ đồ tròn](#_Toc10327)* [44](#_Toc10327)

*[Hình](#_Toc24121)* [27](#_Toc24121) *[. Kết quả so sánh](#_Toc24121)* [45](#_Toc24121)

*[Hình](#_Toc30839)* [28](#_Toc30839) *[. Export Word](#_Toc30839)* [45](#_Toc30839)

*[Hình](#_Toc29439)* [29](#_Toc29439) *[. Export Excel](#_Toc29439)* [46](#_Toc29439)

*[Hình](#_Toc25964)* [30](#_Toc25964) *[. Map](#_Toc25964)* [47](#_Toc25964)

*[Hình](#_Toc25244)* [31](#_Toc25244) *[. Bản đồ vệ tinh](#_Toc25244)* [48](#_Toc25244)

*[Hình](#_Toc26066)* [32](#_Toc26066) *[. AI AHP](#_Toc26066)* [48](#_Toc26066)

*[Hình](#_Toc16931)* [33](#_Toc16931) *[. Import Excel](#_Toc16931)* [49](#_Toc16931)

*[Hình](#_Toc28150)* [34](#_Toc28150) *[. Import Excel](#_Toc28150)* [49](#_Toc28150)

*[Hình](#_Toc5090)* [35](#_Toc5090) *[. Import Excel](#_Toc5090)* [49](#_Toc5090)

*[Hình](#_Toc23352)* [36](#_Toc23352) *[. Kết quả sau khi import excel](#_Toc23352)* [50](#_Toc23352)

**DANH MỤC BẢNG**

[Bảng 1 . So sánh cặp đôi 4](#_Toc24718)

[Bảng 2 . Ví dụ ma trận so sánh tiêu chí 5](#_Toc576)

[Bảng 3 . RI (Chỉ số ngẫu nhiên trung bình) 5](#_Toc29908)

[Bảng 4 . Bảng chỉ số ngẫu nhiên 7](#_Toc29303)

[Bảng 5 . Tác nhân và vai trò 18](#_Toc25481)

[Bảng 6 . Mô tả các chức năng 19](#_Toc22250)

[Bảng 7 . Khó khăn, hạn chế trong việc áp dụng thuật toán AHP 33](#_Toc29066)

**MỞ ĐẦU**

Đề tài “Hệ thống hỗ trợ địa điểm chọn cửa hàng áp dụng phương pháp AHP, sử dụng Python” được thực hiện nhằm xây dựng một công cụ hỗ trợ quyết định khoa học trong việc lựa chọn vị trí đặt cửa hàng kinh doanh dựa trên các tiêu chí đa dạng. Cơ sở khoa học của đề tài là phương pháp phân tích thứ bậc (Analytic Hierarchy Process - AHP), một kỹ thuật ra quyết định đa tiêu chí được phát triển bởi Thomas L. Saaty, giúp xác định trọng số và mức độ ưu tiên các yếu tố ảnh hưởng thông qua so sánh cặp, từ đó đưa ra quyết định tối ưu.

Vấn đề lựa chọn địa điểm kinh doanh có tính thực tiễn cao, bởi vị trí đặt cửa hàng ảnh hưởng trực tiếp đến hiệu quả kinh doanh, chi phí vận hành và khả năng cạnh tranh trên thị trường. Trong bối cảnh thị trường ngày càng phát triển và cạnh tranh khốc liệt, việc ra quyết định dựa trên cảm tính hoặc kinh nghiệm truyền thống thường dẫn đến sai sót và rủi ro cao. Do đó, ứng dụng phương pháp AHP kết hợp với công nghệ lập trình Python để xây dựng hệ thống hỗ trợ quyết định sẽ giúp doanh nghiệp và cá nhân đưa ra lựa chọn chính xác, nhanh chóng, giảm thiểu rủi ro và tối ưu hóa hiệu quả kinh doanh.

Mục tiêu của đề tài là thiết kế và triển khai một hệ thống phần mềm có khả năng nhập dữ liệu các tiêu chí, phân tích, so sánh và đánh giá các địa điểm tiềm năng, từ đó đưa ra bảng xếp hạng các vị trí ưu tiên phù hợp nhất cho việc đặt cửa hàng.

Phương pháp nghiên cứu được áp dụng bao gồm:

- Tìm hiểu lý thuyết về phương pháp phân tích thứ bậc (AHP) và các kỹ thuật ra quyết định đa tiêu chí.

- Phân tích yêu cầu và xây dựng mô hình toán học cho bài toán lựa chọn địa điểm dựa trên AHP.

- Sử dụng Python và các thư viện hỗ trợ (như NumPy, Pandas, thư viện AHP) để lập trình hệ thống xử lý dữ liệu, tính toán trọng số và kiểm tra tính nhất quán.

- Thiết kế giao diện người dùng đơn giản, trực quan giúp nhập dữ liệu và hiển thị kết quả.

Nội dung công việc cụ thể gồm:

- Thu thập và phân tích các tiêu chí ảnh hưởng đến việc lựa chọn địa điểm đặt cửa hàng.

- Xây dựng ma trận so sánh cặp các tiêu chí và đánh giá mức độ ưu tiên.

- Phát triển phần mềm hỗ trợ nhập liệu, xử lý và tính toán bằng AHP trên nền Python.

- Kiểm thử và đánh giá hiệu quả hệ thống với các bộ dữ liệu thực tế.

Đối tượng nghiên cứu của đề tài là các tiêu chí và phương pháp hỗ trợ quyết định trong lựa chọn vị trí kinh doanh. Phạm vi nghiên cứu tập trung vào các cửa hàng nhỏ và vừa trong lĩnh vực thương mại dịch vụ, không mở rộng sang các mô hình phức tạp hơn hoặc các lĩnh vực kinh doanh quy mô lớn.

Đề tài được thực hiện dưới dạng nghiên cứu độc lập, không phụ thuộc vào bất kỳ đơn vị hay công ty cụ thể nào, nhằm phát triển giải pháp ứng dụng có tính thực tiễn cao và khả năng triển khai rộng rãi cho các doanh nghiệp và cá nhân có nhu cầu lựa chọn địa điểm kinh doanh hiệu quả.

Quyển báo cáo sẽ bao gồm các nội dung chính sau:

Chương 1: Tổng quan

Chương 2: Phương pháp thực hiện

Chương 3: Cài đặt thử nghiệm, kết quả và đánh giá

Phần Kết luận và hướng phát triển

# CHƯƠNG 1. TỔNG QUAN DỰ ÁN

## WHAT – **ỨNG DỤNG FLASHCARDS HỌC TỪ VỰNG** LÀ GÌ?

Dự án là một ứng dụng web học từ vựng theo phương pháp flashcards, được phát triển nhằm cung cấp một công cụ hỗ trợ học tập tiện lợi và hiệu quả cho người học ở nhiều cấp độ khác nhau, từ học sinh, sinh viên đến người đi làm muốn nâng cao vốn từ ngữ. Ứng dụng cho phép người dùng tự tạo hoặc lựa chọn các bộ thẻ từ vựng phù hợp với nhu cầu học tập cá nhân, giúp tối ưu hóa quá trình ghi nhớ thông qua các bài luyện tập được thiết kế dựa trên nguyên tắc lặp lại ngắt quãng (Spaced Repetition).

Ứng dụng tích hợp nhiều chế độ học tập đa dạng như học theo thứ tự, luyện tập ngẫu nhiên, kiểm tra trắc nghiệm, giúp người học thay đổi hình thức học để tăng sự hứng thú và hiệu quả ghi nhớ. Đồng thời, hệ thống cung cấp các báo cáo thống kê chi tiết về tiến độ học tập, tỷ lệ nhớ từ, thời gian học tập,... giúp người học dễ dàng theo dõi và điều chỉnh phương pháp học phù hợp.

Bên cạnh đó, dự án cũng chú trọng đến khía cạnh quản lý nội dung và người dùng, cho phép người quản trị hệ thống có thể phân quyền rõ ràng, kiểm soát chất lượng bộ thẻ do người dùng tạo ra, quản lý tài khoản, đồng thời theo dõi hoạt động học tập chung của cộng đồng người dùng. Việc sử dụng công nghệ đám mây Firebase giúp dữ liệu được đồng bộ thời gian thực, đảm bảo trải nghiệm mượt mà trên nhiều thiết bị khác nhau.

Tóm lại, dự án không chỉ đơn thuần là một công cụ học từ vựng mà còn là một nền tảng quản lý học tập toàn diện, hỗ trợ tối đa cho cả người học và người quản lý trong việc phát triển và duy trì thói quen học tập lâu dài, hiệu quả. [[1]](#endnote-0)

## WHERE – DỰ ÁN ĐƯỢC TRIỂN KHAI Ở ĐÂU?

Dự án được phát triển trên nền tảng Flutter SDK, một bộ công cụ phát triển giao diện người dùng mã nguồn mở do Google phát hành, cho phép xây dựng ứng dụng di động đa nền tảng (cross-platform) chỉ với một codebase duy nhất, hỗ trợ triển khai trên cả Android và iOS [[2]](#endnote-1).

Phía backend của ứng dụng được xây dựng dựa trên Firebase, một nền tảng Backend-as-a-Service (BaaS) của Google, được tích hợp để xử lý các chức năng quan trọng bao gồm:

- Firebase Authentication: Xác thực người dùng qua email/password.

- Firebase Cloud Firestore: Cơ sở dữ liệu NoSQL thời gian thực dùng để lưu trữ và đồng bộ dữ liệu flashcard của người dùng.

- Firebase Core: Cấu hình và khởi tạo kết nối đến dịch vụ Firebase.

Các thư viện chính được sử dụng trong dự án bao gồm:

- flutter\_bloc: Quản lý trạng thái ứng dụng theo mô hình Bloc (Business Logic Component).

- firebase\_core: Thiết lập và khởi tạo Firebase trong ứng dụng Flutter.

- firebase\_auth: Thư viện xác thực người dùng Firebase.

- cloud\_firestore: Thư viện truy xuất dữ liệu với Cloud Firestore.

- google\_fonts: Tùy chỉnh font chữ trong giao diện người dùng.

- flutter\_spinkit: Thư viện hiệu ứng loading animation.

- equatable: Hỗ trợ so sánh các object trong quá trình quản lý trạng thái.

Việc lựa chọn Flutter kết hợp Firebase giúp rút ngắn thời gian phát triển, đồng thời đảm bảo hiệu năng tốt và tính mở rộng của hệ thống trong tương lai. [[3]](#endnote-2)

## WHEN – DỰ ÁN ĐƯỢC THỰC HIỆN KHI NÀO?

Dự án được khởi động và bắt đầu phát triển từ đầu năm 2025, với mục tiêu hoàn thiện trong cùng năm. Quá trình phát triển trải qua nhiều giai đoạn từ nghiên cứu, thiết kế, triển khai cho đến kiểm thử và hoàn thiện sản phẩm. Thời gian thực hiện dự án được lên kế hoạch kỹ lưỡng nhằm đảm bảo đáp ứng các yêu cầu kỹ thuật cũng như mục tiêu về trải nghiệm người dùng.

Dự án được phát triển song song với việc nghiên cứu các phương pháp học tập hiện đại, tập trung vào ứng dụng công nghệ để tối ưu hóa hiệu quả ghi nhớ từ vựng. Việc triển khai trong năm 2025 nhằm tận dụng các công nghệ mới nhất và đáp ứng xu hướng học tập trực tuyến ngày càng phát triển trong bối cảnh xã hội hiện đại.

Bên cạnh mục tiêu phục vụ nhu cầu học tập và nghiên cứu cá nhân, dự án còn hướng tới việc hỗ trợ rộng rãi cộng đồng người học từ vựng, đặc biệt là học sinh, sinh viên và những người có nhu cầu nâng cao trình độ ngoại ngữ trên nền tảng kỹ thuật số tiện lợi và dễ tiếp cận.

## WHY – MỤC TIÊU, LÝ DO THỰC HIỆN DỰ ÁN

Dự án được thực hiện nhằm đáp ứng nhu cầu học tập ngày càng cá nhân hóa và linh hoạt của người dùng trong thời đại số. Các mục tiêu chính bao gồm:

- Hỗ trợ người học ôn tập kiến thức hiệu quả với flashcards: Flashcards là một trong những phương pháp học tập đã được chứng minh có hiệu quả cao nhờ vào kỹ thuật gợi nhớ chủ động (active recall) và lặp lại ngắt quãng (spaced repetition) – hai kỹ thuật giúp tăng cường trí nhớ dài hạn và nâng cao hiệu quả ôn tập.

- Cung cấp giải pháp học tập cá nhân hóa, đơn giản nhưng hiệu quả: Ứng dụng cho phép người dùng tự tạo nội dung học tập theo nhu cầu cá nhân, dễ sử dụng, không đòi hỏi thao tác phức tạp, phù hợp với nhiều đối tượng, đặc biệt là học sinh – sinh viên.

- Giúp người học có thể tạo thẻ ghi nhớ bất kỳ lúc nào, ở bất kỳ đâu: Nhờ việc triển khai trên thiết bị di động với khả năng đồng bộ dữ liệu thời gian thực qua Firebase, người dùng có thể học mọi lúc mọi nơi, không bị giới hạn bởi không gian hay thời gian.

- Rèn luyện kỹ năng xây dựng ứng dụng di động với Flutter & Firebase: Quá trình phát triển dự án giúp người thực hiện nâng cao kỹ năng thiết kế giao diện người dùng, quản lý trạng thái, tích hợp backend Firebase, triển khai ứng dụng đa nền tảng – là những năng lực thiết yếu trong phát triển ứng dụng hiện đại.

Dự án không chỉ mang ý nghĩa về mặt học thuật mà còn có giá trị thực tiễn cao, góp phần phát triển tư duy công nghệ, kỹ năng lập trình và khả năng giải quyết vấn đề thực tế của người học.

## HOW – CÁCH HOẠT ĐỘNG VÀ TRIỂN KHAI

Ứng dụng học tập flashcards được triển khai theo mô hình client-server với kiến trúc hiện đại, tận dụng sức mạnh của Flutter cho giao diện người dùng và Firebase cho dịch vụ backend. Các bước triển khai chính bao gồm:

### Giao diện người dùng (Frontend với Flutter)

Ứng dụng được xây dựng bằng Flutter, một SDK đa nền tảng cho phép viết một lần và chạy trên cả Android và iOS. Giao diện được thiết kế tối ưu cho thiết bị di động, trực quan, dễ sử dụng, bao gồm các màn hình chính:

- Login Screen: Màn hình đăng nhập bằng email và mật khẩu, kết nối với Firebase Authentication.

- Register Screen: Màn hình tạo tài khoản mới.

- Home Screen: Màn hình chính hiển thị danh sách các bộ flashcards đã tạo.

- FlashcardSets Screen: Giao diện quản lý các bộ flashcards, thêm/sửa/xoá.

- FlashcardDetail Screen: Màn hình học tập flashcard, hỗ trợ chế độ xem trước/sau (front/back) cho từng thẻ.

### Quản lý người dùng (Firebase Authentication)

Firebase Authentication được sử dụng để:

- Xác thực người dùng bằng email và mật khẩu.

- Quản lý phiên đăng nhập (session).

- Đảm bảo mỗi người dùng có một không gian dữ liệu riêng tư trong hệ thống.

### Quản lý dữ liệu (Firebase Cloud Firestore)

Cloud Firestore, cơ sở dữ liệu NoSQL thời gian thực của Firebase, được sử dụng để:

- Lưu trữ danh sách các bộ flashcards và các thẻ bên trong từng bộ.

- Đồng bộ dữ liệu tức thời giữa thiết bị và server.

- Cho phép người dùng chỉnh sửa, thêm mới hoặc xoá flashcards mọi lúc, mọi nơi.

### Tối ưu trải nghiệm người dùng

- Ứng dụng sử dụng các thư viện như flutter\_bloc để quản lý trạng thái logic hiệu quả và google\_fonts để tuỳ chỉnh giao diện theo phong cách hiện đại.

- Hiệu ứng tải (loading animations) được tích hợp với flutter\_spinkit nhằm tăng tính tương tác.

- Thiết kế responsive đảm bảo trải nghiệm mượt mà trên nhiều kích thước màn hình.

# CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN

## PHÂN TÍCH YÊU CẦU NGHIỆP VỤ

### Đối tượng sử dụng

### Yêu cầu chức năng

#### Home screen

- Hiển thị danh sách các bộ flashcard của người dùng hiện có.

- Cho phép người dùng dễ dàng xem tổng quan các bộ flashcard để lựa chọn học hoặc chỉnh sửa.

#### Thêm/Sửa/Xóa bộ flashcard

- Cho phép người dùng tạo mới một bộ flashcard với tên bộ, mô tả và ngày tạo.

- Hỗ trợ chỉnh sửa tên bộ flashcard và cập nhật ngày chỉnh sửa.

- Cho phép xóa bỏ bộ flashcard không còn sử dụng hoặc không cần thiết.

#### Chi tiết bộ flashcard

- Khi người dùng chọn một bộ flashcard, ứng dụng sẽ hiển thị danh sách tất cả các thẻ (card) thuộc bộ đó.

- Hiển thị đầy đủ thông tin của từng thẻ bao gồm từ khóa, nghĩa, ví dụ minh họa, và ghi chú.

#### Thêm/Sửa/Xóa thẻ

- Người dùng có thể thêm thẻ mới vào bộ flashcard với các trường thông tin: từ khóa (keyword), nghĩa (meaning), ví dụ minh họa (example), ghi chú thêm (note).

- Hỗ trợ chỉnh sửa nội dung từng thẻ để cập nhật hoặc sửa lỗi.

- Cho phép xóa thẻ không cần thiết hoặc lỗi.

#### Chế độ học flashcard

- Xem thẻ: Hiển thị từ khóa và nghĩa, giúp người học tập trung vào ghi nhớ từ mới.

- Lật thẻ (Flip Card): Người dùng có thể lật thẻ để xem nghĩa hoặc các thông tin chi tiết khác trên mặt sau của thẻ.

- Kiểm tra trí nhớ (Quiz/Recall Mode): Ứng dụng sẽ tạo các câu hỏi ngắn hoặc yêu cầu nhớ nghĩa từ khóa, giúp tăng cường khả năng ghi nhớ và ôn luyện hiệu quả.

#### Tìm kiếm flashcard

- Cho phép người dùng tìm kiếm nhanh các bộ flashcard hoặc thẻ theo từ khóa nhập vào.

- Hỗ trợ tìm kiếm theo từng bộ hoặc trên toàn bộ flashcard của người dùng.

#### Quản lý người dùng

- Hệ thống đăng nhập, đăng ký tài khoản người dùng mới bằng email và mật khẩu.

- Cho phép người dùng đăng xuất và cập nhật thông tin cá nhân như tên, mật khẩu, email.

- Bảo mật dữ liệu người dùng và xác thực đăng nhập.

#### Đồng bộ Firebase

- Tất cả dữ liệu người dùng (bộ flashcard, thẻ, thông tin cá nhân) được lưu trữ trên Firestore của Firebase.

- Đảm bảo đồng bộ dữ liệu realtime, cập nhật ngay lập tức khi có thay đổi trên bất kỳ thiết bị nào.

#### Thống kê học tập

- Ghi nhận và hiển thị tiến độ học tập của người dùng.

- Thống kê số lượng thẻ đã học, thời gian học, tỉ lệ ghi nhớ,… giúp người dùng theo dõi và đánh giá kết quả học tập.

### Yêu cầu phi chức năng

- Hiệu năng: Ứng dụng phải hoạt động mượt mà, phản hồi nhanh, đặc biệt khi tải dữ liệu và chuyển đổi giữa các màn hình. Đồng bộ dữ liệu với Firebase phải hiệu quả và không gây gián đoạn.

- Bảo mật: Bảo vệ thông tin người dùng và dữ liệu flashcards khỏi truy cập trái phép. Áp dụng xác thực an toàn qua Firebase Authentication và phòng tránh các lỗ hổng bảo mật phổ biến như XSS, SQL Injection.

- Khả năng mở rộng: Thiết kế modul dễ dàng mở rộng tính năng mới, dữ liệu và số lượng người dùng lớn vẫn được hỗ trợ tốt.

- Khả năng bảo trì: Mã nguồn rõ ràng, có chú thích đầy đủ, dễ dàng cho việc bảo trì, nâng cấp và kiểm soát lỗi.

- Khả năng tương thích: Hỗ trợ đa trình duyệt hiện đại và thiết kế giao diện tương thích trên cả máy tính và thiết bị di động (responsive).

- Sao lưu và phục hồi: Dữ liệu được sao lưu tự động trên Firebase và có khả năng phục hồi khi xảy ra sự cố.

### Yêu cầu về giao diện và trải nghiệm người dùng

- Giao diện đơn giản, thân thiện: Thiết kế trực quan, dễ dùng, phù hợp với đa dạng đối tượng. Các thành phần UI rõ ràng, dễ nhận biết và dễ truy cập.

- Tương tác nhanh chóng, phản hồi tức thì: Mỗi thao tác có phản hồi rõ ràng như hiệu ứng nhấn nút, thông báo thành công hoặc lỗi, giảm thiểu các bước phức tạp để thao tác nhanh.

- Tính nhất quán trong thiết kế: Màu sắc, font chữ, bố cục đồng bộ trên tất cả màn hình. Sử dụng icon, nút bấm hợp lý để người dùng dễ hiểu chức năng.

- Responsive design: Giao diện tự động điều chỉnh phù hợp với các kích thước màn hình khác nhau như điện thoại, máy tính bảng, máy tính để bàn, đảm bảo trải nghiệm liền mạch.

- Hỗ trợ tìm kiếm và lọc: Thanh tìm kiếm dễ nhìn, dễ dùng; kết quả tìm kiếm hiển thị nhanh và có thể lọc theo nhiều tiêu chí khác nhau.

- Hướng dẫn sử dụng cơ bản: Cung cấp tooltip hoặc màn hình giới thiệu nhanh cho người dùng mới, giúp làm quen với các chức năng chính của ứng dụng.

## THIẾT KẾ TỔNG QUAN HỆ THỐNG

### Kiến trúc hệ thống

Hệ thống ứng dụng học từ vựng bằng flashcard được thiết kế theo mô hình client-server, hỗ trợ hoạt động trên cả nền tảng web và thiết bị di động (Android, iOS). Kiến trúc tổng thể bao gồm các thành phần chính sau:

#### Client (Giao diện người dùng)

- Phát triển bằng Flutter, hỗ trợ đa nền tảng: Web, Android, iOS.

- Triển khai trên web dưới dạng Progressive Web App (PWA) và đóng gói dưới dạng ứng dụng di động (APK, IPA) cho các hệ điều hành.

- Cung cấp giao diện cho người dùng thực hiện các chức năng: đăng ký, đăng nhập, học từ vựng, tạo và chỉnh sửa flashcards, theo dõi tiến độ học tập.

#### Backend (Firebase Services)

Firebase đảm nhiệm vai trò backend, cung cấp nhiều dịch vụ tích hợp sẵn:

- Firebase Authentication: Xác thực và quản lý tài khoản người dùng.

- Cloud Firestore: Cơ sở dữ liệu NoSQL thời gian thực, lưu trữ thông tin người dùng, flashcards, tiến độ học tập,...

- Firebase Cloud Functions (tùy chọn mở rộng): Xử lý các nghiệp vụ backend phức tạp (ví dụ: gửi thông báo, thống kê nâng cao).

- Firebase Hosting: Lưu trữ và phân phối phiên bản web (PWA) của ứng dụng.

- Firebase Cloud Messaging (tùy chọn): Gửi thông báo đẩy đến thiết bị người dùng.

#### Realtime Data Synchronization

Dữ liệu được đồng bộ hóa theo thời gian thực giữa client và backend thông qua Firestore, giúp người dùng cập nhật tức thì khi thao tác trên flashcards.

### Các thành phần chính của hệ thống

Hệ thống học từ vựng bằng flashcard bao gồm bốn thành phần chính, phối hợp chặt chẽ nhằm mang đến trải nghiệm học tập liền mạch, hiệu quả và thân thiện cho người dùng.

#### Giao diện người dùng (UI)

- Màn hình chính (Home)

- Màn hình chi tiết bộ flashcard

- Màn hình thêm/sửa thẻ

- Màn hình học (Learning Mode)

- Màn hình thống kê

- Màn hình đăng nhập/đăng ký

#### Firebase Authentication

- Xác thực và phân quyền người dùng.

- Lưu trữ thông tin cá nhân: tên, email, thời gian đăng ký.

#### Cloud Firestore

- Cấu trúc dữ liệu theo từng người dùng.

- Mỗi người dùng có nhiều bộ flashcard, mỗi bộ chứa nhiều thẻ.

#### Realtime Updates

Khi người dùng học hoặc chỉnh sửa dữ liệu, mọi thay đổi được cập nhật tức thì lên UI và lưu vào Firestore.

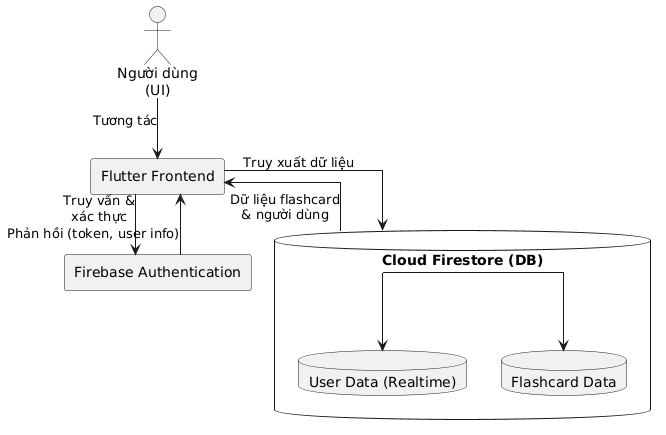
### Mối quan hệ và luồng dữ liệu giữa các thành phần

- Người dùng tương tác với UI → gọi hàm trên Firebase → nhận phản hồi → cập nhật UI.

- Khi học flashcard → hệ thống lưu tiến độ (số thẻ đã học) theo thời gian thực.

- Khi đăng nhập → Firestore truy xuất bộ flashcard cá nhân.

### Sơ đồ tổng quan hệ thống (System Overview Diagram)



*Hình 1. Sơ đồ tổng quan hệ thống*

## SƠ ĐỒ USECASE

### Các tác nhân (Actors) và vai trò

Bảng 1. Tác nhân và vai trò

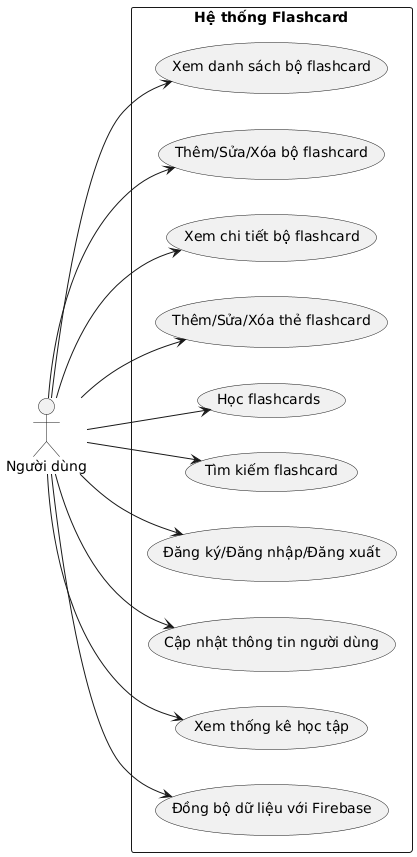
|  |  |
| --- | --- |
| **Tác nhân (Actor)** | **Vai trò / Mô tả** |
| User (Người dùng) | Người sử dụng ứng dụng để học từ vựng, tạo và quản lý flashcard, tra cứu và theo dõi tiến độ học tập. |
| Firebase Authentication | Hệ thống xác thực tài khoản người dùng (đăng nhập/đăng ký). |
| Firestore Database | Lưu trữ và đồng bộ dữ liệu flashcard, thông tin người dùng và kết quả học tập theo thời gian thực. |

### Mô tả các chức năng chính

Bảng 2. Mô tả các chức năng

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả chi tiết** |
| Trang chủ (Home screen) | Hiển thị danh sách các bộ flashcard của người dùng sau khi đăng nhập. |
| Thêm/Sửa/Xóa bộ flashcard | Cho phép người dùng tạo mới bộ flashcard, sửa tên bộ, cập nhật ngày tạo, hoặc xóa bộ không còn dùng. |
| Xem chi tiết bộ flashcard | Khi người dùng chọn một bộ, hệ thống hiển thị danh sách tất cả thẻ nằm trong bộ đó. |
| Thêm/Sửa/Xóa thẻ | Người dùng có thể thêm mới, chỉnh sửa hoặc xóa từng thẻ flashcard trong bộ. Mỗi thẻ bao gồm: từ khóa (keyword), nghĩa (meaning), ví dụ, và ghi chú thêm. |
| Học flashcards | Cung cấp các chế độ học gồm: xem thẻ (keyword → meaning), lật thẻ (flip), và kiểm tra ghi nhớ (quiz). |
| Tìm kiếm flashcard | Tìm nhanh bộ hoặc các thẻ trong bộ dựa trên từ khóa mà người dùng nhập. |
| Quản lý người dùng | Cho phép người dùng đăng ký, đăng nhập, đăng xuất và cập nhật thông tin cá nhân. |
| Đồng bộ dữ liệu Firebase | Tất cả bộ flashcard, thẻ và thông tin học tập được lưu trữ trên Firebase Firestore và đồng bộ thời gian thực. |
| Thống kê học tập | Ghi nhận số lượng flashcard đã học, ngày học gần nhất và tiến độ học của người dùng để hỗ trợ quá trình ôn tập. |

### Sơ đồ Use Case thể hiện tương tác người dùng với hệ thống



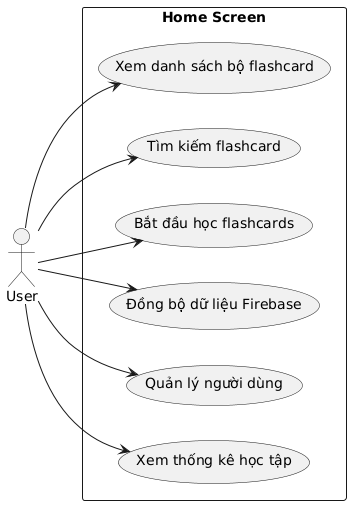
*Hình 2. Sơ đồ Usecase tương tác với hệ thống*

### Sơ đồ Use Case con

#### Sơ đồ Usecase Home screen

Màn hình chính của ứng dụng cho phép người dùng truy cập nhanh đến các chức năng chính như xem danh sách các bộ flashcard đã tạo hoặc có sẵn, thực hiện tìm kiếm theo từ khóa để dễ dàng tìm bộ flashcard mong muốn.

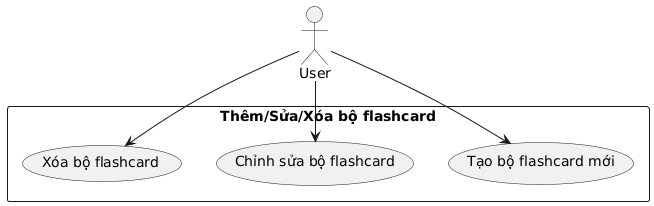
Người dùng có thể bắt đầu ngay một phiên học flashcard từ màn hình này. Ngoài ra, chức năng đồng bộ dữ liệu với Firebase giúp đảm bảo dữ liệu luôn được cập nhật trên nhiều thiết bị. Người dùng cũng có thể quản lý tài khoản cá nhân và xem các thống kê tổng quan về quá trình học tập ngay trên màn hình Home.



*Hình 3. Sơ đồ Usecase Home screen*

#### Sơ đồ Usecase Thêm/Sửa/Xóa bộ flashcard

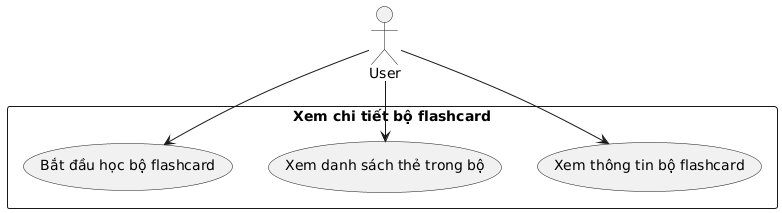
Người dùng có thể tạo mới một bộ flashcard với tên gọi và mô tả chi tiết, phục vụ việc tổ chức các chủ đề học tập. Các bộ flashcard cũng có thể được chỉnh sửa để cập nhật hoặc điều chỉnh nội dung khi cần thiết. Ngoài ra, người dùng có thể xóa những bộ flashcard không còn sử dụng, giúp quản lý dữ liệu được gọn gàng và dễ dàng hơn.



*Hình 4. Sơ đồ Usecase Thêm/Sửa/Xóa bộ flashcard*

#### Sơ đồ Usecase Xem chi tiết bộ flashcard

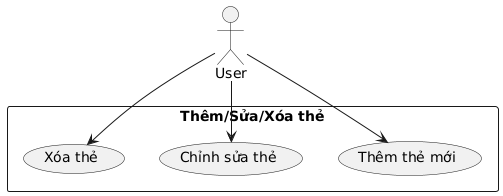
Khi chọn một bộ flashcard, người dùng có thể xem chi tiết thông tin như tên bộ, mô tả, số lượng thẻ trong bộ. Màn hình chi tiết này cũng liệt kê danh sách các thẻ flashcard thuộc bộ đó để người dùng kiểm tra hoặc chỉnh sửa. Từ đây, người dùng cũng có thể trực tiếp khởi động phiên học cho bộ flashcard đã chọn.



*Hình 5. Sơ đồ Usecase Xem chi tiết bộ flashcard*

#### Sơ đồ Usecase Thêm/Sửa/Xóa thẻ

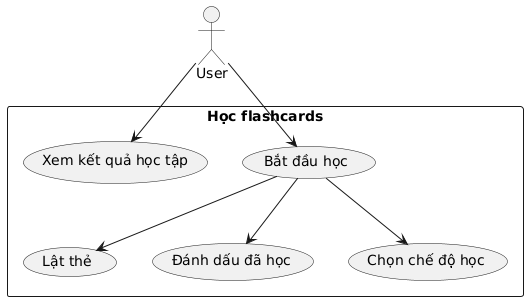
Trong một bộ flashcard, người dùng có thể thêm các thẻ mới chứa cặp câu hỏi và câu trả lời phục vụ cho việc học tập. Các thẻ này có thể được chỉnh sửa nếu phát hiện sai sót hoặc cần bổ sung thông tin. Khi một thẻ không còn cần thiết, người dùng có thể xóa thẻ đó để giữ cho bộ flashcard luôn chính xác và phù hợp với nhu cầu học tập.



*Hình 6. Sơ đồ Usecase Thêm/Sửa/Xóa thẻ*

#### Sơ đồ Usecase Học flashcards

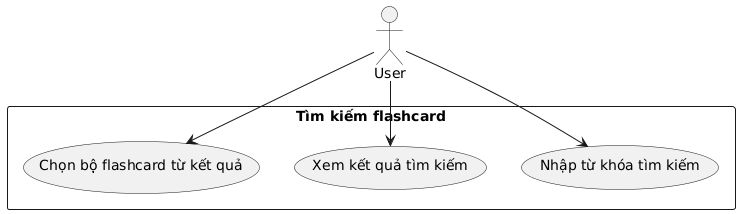
Chức năng học flashcards cho phép người dùng bắt đầu phiên học, trong đó có thể lật thẻ để xem câu trả lời, đánh dấu các thẻ đã nhớ để theo dõi tiến trình học. Người dùng còn có thể chọn nhiều chế độ học khác nhau, ví dụ như học tuần tự, học theo mức độ khó, hoặc học theo nhóm thẻ. Sau mỗi phiên học, người dùng có thể xem lại kết quả và thống kê chi tiết để đánh giá hiệu quả của quá trình học.



*Hình 7. Sơ đồ Usecase Học flashcard*

#### Sơ đồ Usecase Tìm kiếm flashcard

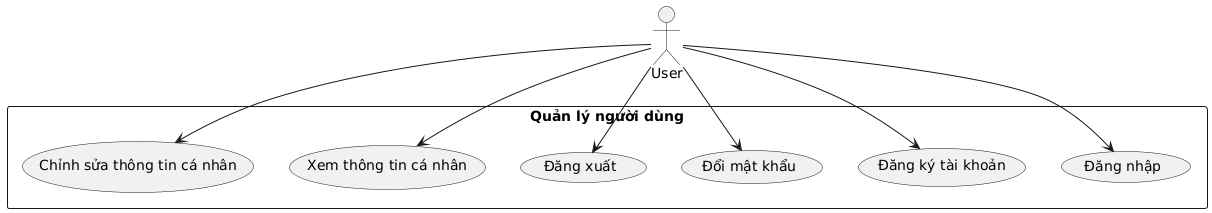
Người dùng nhập từ khóa tìm kiếm liên quan đến tên bộ flashcard hoặc nội dung thẻ để tìm nhanh các dữ liệu phù hợp. Hệ thống hiển thị danh sách kết quả tương ứng, giúp người dùng dễ dàng lựa chọn bộ hoặc thẻ flashcard cần thiết để xem hoặc bắt đầu học.



*Hình 8. Sơ đồ Usecase Tìm kiếm flashcard*

#### Sơ đồ Usecase Quản lý người dùng

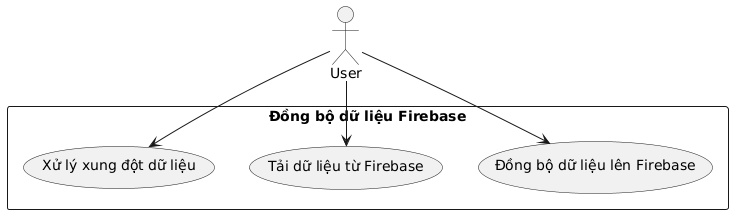
Ứng dụng cung cấp các chức năng quản lý tài khoản người dùng bao gồm đăng ký tài khoản mới, đăng nhập vào hệ thống để truy cập các dữ liệu cá nhân. Người dùng có thể thay đổi mật khẩu để bảo mật tài khoản, chỉnh sửa thông tin cá nhân như tên, email hoặc ảnh đại diện, và đăng xuất khỏi hệ thống khi kết thúc phiên làm việc.



*Hình 9. Sơ đồ Usecase Quản lý người dùng*

#### Sơ đồ Usecase Đồng bộ dữ liệu Firebase

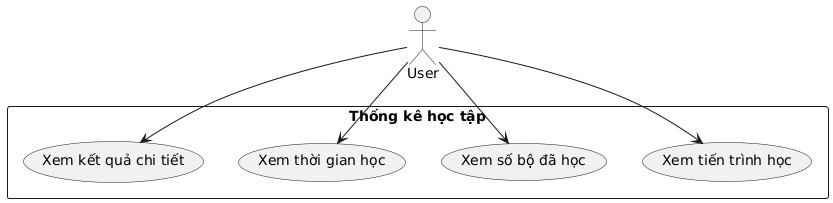
Chức năng đồng bộ cho phép người dùng lưu trữ các bộ flashcard, thẻ flashcard lên hệ thống Firebase, giúp bảo vệ dữ liệu và hỗ trợ sử dụng trên nhiều thiết bị. Khi tải dữ liệu về thiết bị, ứng dụng đảm bảo dữ liệu được cập nhật mới nhất. Trong trường hợp có xung đột dữ liệu do thay đổi trên nhiều thiết bị, hệ thống sẽ tự động hoặc yêu cầu người dùng xử lý để đảm bảo dữ liệu đồng nhất.



*Hình 10. Sơ đồ Usecase Đồng bộ dữ liệu Firebase*

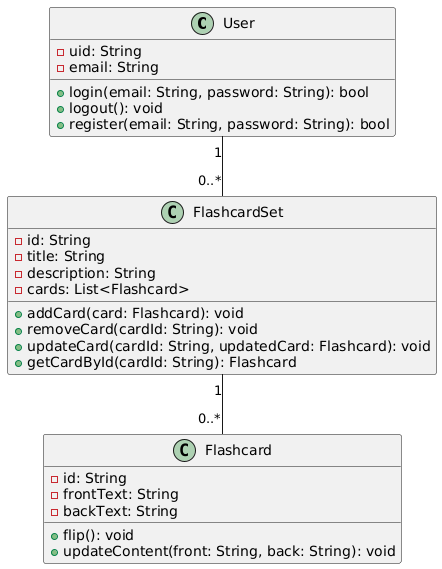
#### Sơ đồ Usecase Thống kê học tập

Người dùng có thể xem các báo cáo thống kê chi tiết về quá trình học tập như tiến trình học flashcard qua từng phiên, số lượng bộ flashcard đã học hoặc hoàn thành, tổng thời gian dành cho việc học. Ngoài ra, người dùng còn được cung cấp các kết quả phân tích chi tiết về mức độ nhớ, quên các thẻ flashcard để từ đó điều chỉnh phương pháp học tập hiệu quả hơn.



*Hình 11. Sơ đồ Usecase Thống kê học tập*

## SƠ ĐỒ LỚP (CLASS DIAGRAM)



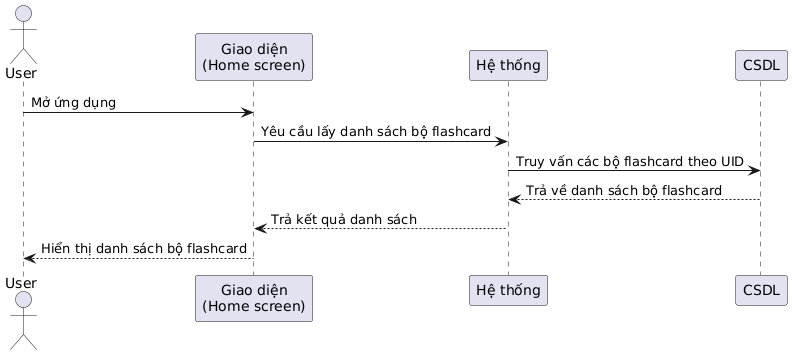
*Hình 12. Sơ đồ Class*

### Các lớp chính trong hệ thống

### Mối quan hệ giữa các lớp

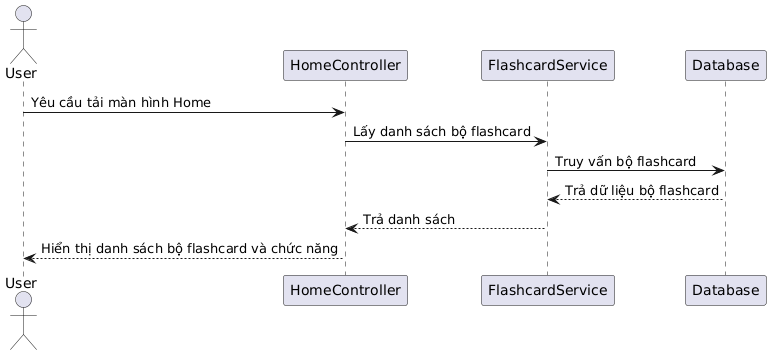
### Thuộc tính và phương thức của từng lớp

## SƠ ĐỒ TUẦN TỰ (SEQUENCE DIAGRAM)



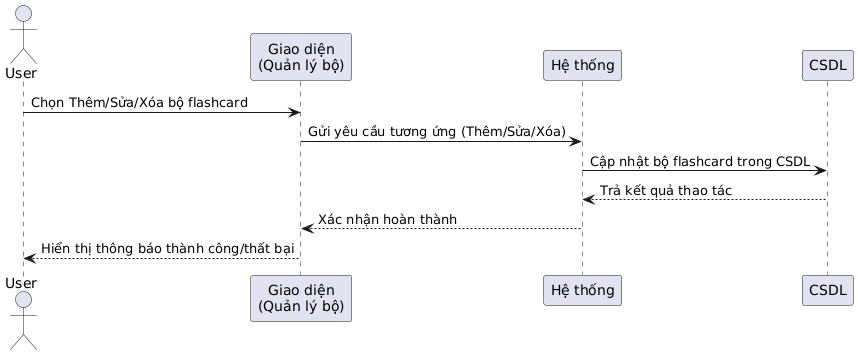
*Hình 13. Sơ đồ tuần tự tổng quát*

#### Sơ đồ tuần tự Home screen



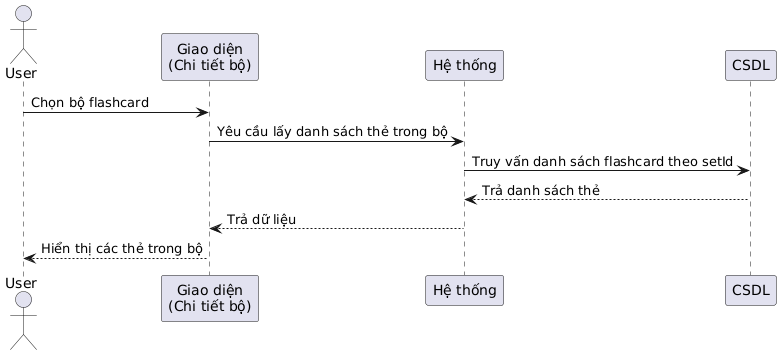
*Hình 14. Sơ đồ tuần tự Home screen*

#### Sơ đồ tuần tự Thêm/Sửa/Xóa bộ flashcard



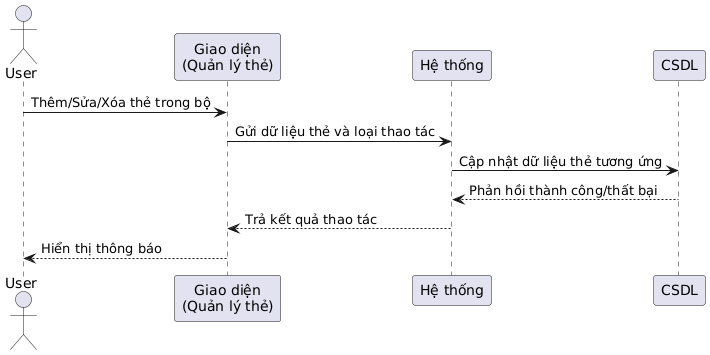
*Hình 15. Sơ đồ tuần tự Thêm/Sửa/Xóa bộ flashcard*

#### Sơ đồ tuần tự Xem chi tiết bộ flashcard



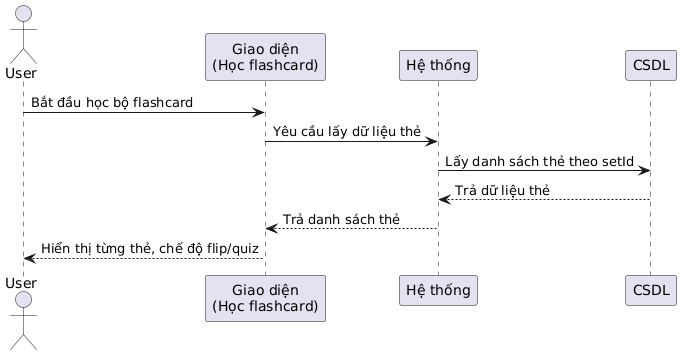
*Hình 16. Sơ đồ tuần tự Xem chi tiết nộ flashcard*

#### Sơ đồ tuần tự Thêm/Sửa/Xóa thẻ



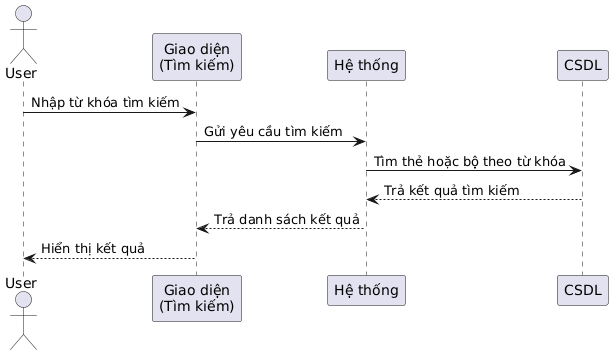
*Hình 17. Sơ đồ tuần tự Thêm/Sửa/Xóa thẻ*

#### Sơ đồ tuần tự Học flashcards



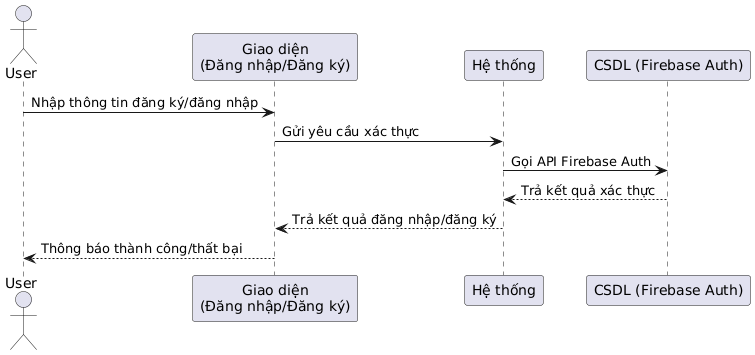
*Hình 18. Sơ đồ tuần tự Học flashcard*

#### Sơ đồ tuần tự Tìm kiếm flashcard



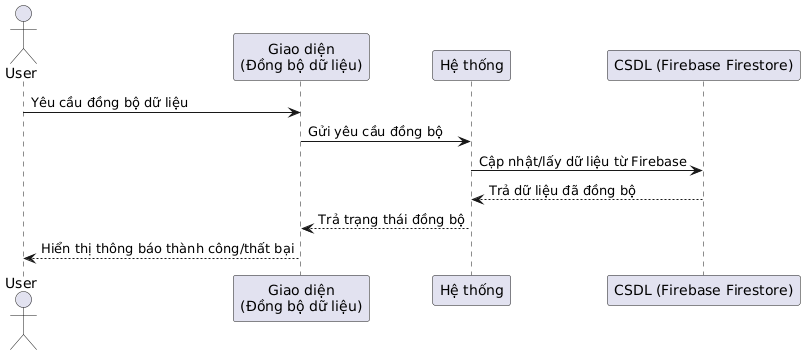
*Hình 19. Sơ đồ tuần tự Tìm kiếm flashcard*

#### Sơ đồ tuần tự Quản lý người dùng



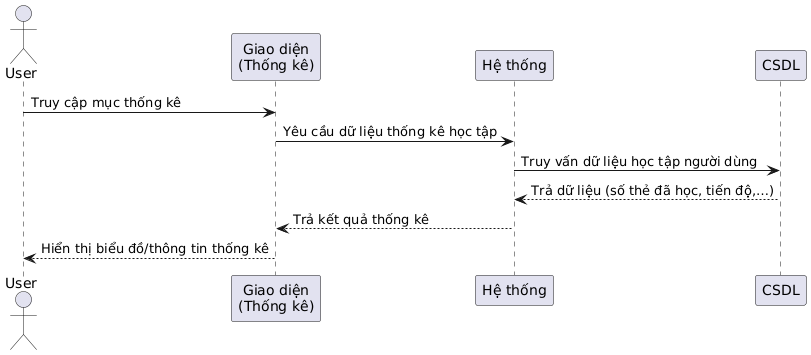
*Hình 20. Sơ đồ tuần tự Quản lý người dùng*

#### Sơ đồ tuần tự Đồng bộ dữ liệu Firebase



*Hình 21. Sơ đồ tuần tự Đồng bộ dữ liệu Firebase*

#### Sơ đồ tuần tự Thống kê học tập



*Hình 22. Sơ đồ tuần tự Thống kê học tập*

## LỰA CHỌN THIẾT KẾ VÀ CÔNG NGHỆ

### Các công nghệ sử dụng

Dự án sử dụng các công nghệ hiện đại, phù hợp với yêu cầu phát triển ứng dụng di động đa nền tảng và hỗ trợ realtime:

- Flutter SDK (ngôn ngữ Dart): Bộ công cụ phát triển giao diện người dùng (UI) mã nguồn mở do Google phát hành, hỗ trợ phát triển ứng dụng Android và iOS từ cùng một codebase.

- Firebase Authentication: Cung cấp dịch vụ xác thực người dùng nhanh chóng, bảo mật, hỗ trợ đăng ký/đăng nhập bằng email và mật khẩu.

- Firebase Cloud Firestore: Cơ sở dữ liệu NoSQL realtime giúp lưu trữ và đồng bộ dữ liệu flashcard của người dùng qua nhiều thiết bị.

Ngoài ra, dự án sử dụng các thư viện hỗ trợ như:

- flutter\_bloc: Quản lý trạng thái theo mô hình Bloc.

- firebase\_core, cloud\_firestore, firebase\_auth: Giao tiếp với Firebase.

- equatable: Hỗ trợ so sánh các trạng thái/phản ứng trong Bloc.

- flutter\_spinkit: Hiệu ứng loading.

- google\_fonts: Tuỳ biến font giao diện.

### Phương án thiết kế kiến trúc hệ thống

Kiến trúc hệ thống được thiết kế dựa trên nguyên tắc phân tách rõ ràng giữa giao diện và xử lý logic, giúp dễ dàng mở rộng và bảo trì:

Phân lớp rõ ràng: Bao gồm các lớp UI (giao diện), Bloc (xử lý logic và trạng thái), Model (định nghĩa dữ liệu), Repository (tương tác với Firebase).

Realtime binding: Sử dụng FutureBuilder cho các thao tác bất đồng bộ (fetch dữ liệu ban đầu), và StreamBuilder cho việc theo dõi và cập nhật dữ liệu theo thời gian thực từ Firestore.

Bảo mật dữ liệu theo người dùng: Mỗi người dùng chỉ truy cập được dữ liệu flashcard của chính mình, dựa vào UID.

### Lý do lựa chọn các công nghệ và phương án thiết kế

Flutter SDK giúp tiết kiệm thời gian phát triển và duy trì ứng dụng trên cả Android và iOS, với hiệu năng cao và cộng đồng phát triển lớn.

Firebase Authentication đơn giản hoá quá trình xác thực người dùng mà vẫn đảm bảo bảo mật.

Cloud Firestore cung cấp khả năng lưu trữ dữ liệu có cấu trúc động, hỗ trợ realtime, đồng bộ hoá giữa các thiết bị và dễ dàng tích hợp với Flutter.

Thiết kế theo mô hình Bloc và sử dụng các công cụ như StreamBuilder giúp quản lý trạng thái rõ ràng, xử lý dữ liệu đồng bộ hiệu quả, và hỗ trợ mở rộng tính năng trong tương lai.

# CHƯƠNG 3. CÀI ĐẶT THỬ NGHIỆM, KẾT QUẢ VÀ ĐÁNH GIÁ

## QUY TRÌNH TRIỂN KHAI HỆ THỐNG

### Môi trường và công cụ phát triển

- Hệ điều hành: Windows 10 / macOS Ventura

- IDE: Visual Studio Code với Flutter plugin

- Ngôn ngữ: Dart (Flutter SDK)

- Cơ sở dữ liệu và dịch vụ backend: Firebase Authentication, Firebase Cloud Firestore

- Thiết bị thử nghiệm: Android Emulator và thiết bị thật (Android 10+)

### Quy trình cài đặt và cấu hình ban đầu

1. Cài đặt Flutter SDK và cấu hình môi trường phát triển.
2. Khởi tạo project Flutter: flutter create flashcards\_app
3. Thêm các gói cần thiết vào pubspec.yaml:

firebase\_core

firebase\_auth

cloud\_firestore

1. Thiết lập Firebase project:

- Đăng ký ứng dụng trên Firebase Console.

- Tải về và cấu hình file google-services.json / firebase\_options.dart.

1. Khởi tạo các màn hình: đăng nhập, đăng ký, trang chủ, bộ flashcard, chi tiết flashcard.

### Từng bước thực hiện các thành phần hệ thống

- Authentication: Viết logic xử lý đăng ký, đăng nhập qua FirebaseAuth.

- FlashcardSet CRUD: Kết nối tới Firestore để tạo/lấy/xoá bộ flashcard.

- Flashcard CRUD: Thêm, chỉnh sửa, xóa flashcard trong từng set.

- Giao diện người dùng: Xây dựng UI với các StatefulWidget và xử lý tương tác người dùng.

- Hiển thị nội dung học: Lật thẻ flashcard bằng cách thay đổi front/back.

### Khó khăn và thách thức trong quá trình thực hiện

- Firebase có giới hạn truy vấn phức tạp, yêu cầu tái cấu trúc dữ liệu.

- Việc đồng bộ UI – dữ liệu Firestore yêu cầu quản lý trạng thái hiệu quả.

- Giao diện cần đơn giản nhưng trực quan, dễ sử dụng cho người học.

## CẤU TRÚC VÀ CHỨC NĂNG HỆ THỐNG

### Cấu trúc tổng thể của hệ thống phần mềm

### Các chức năng chính đã triển khai

### Mô tả các module chính và vai trò của từng module

### Giao diện người dùng và tương tác hệ thống

## KẾT QUẢ THỬ NGHIỆM

## ĐÁNH GIÁ VÀ PHÂN TÍCH KẾT QUẢ

### Mức độ đáp ứng yêu cầu kỹ thuật đã đặt ra

### Phân tích độ chính xác và hiệu quả

### Ưu điểm của hệ thống

### Hạn chế hiện tại và hướng cải thiện trong tương lai

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## KẾT LUẬN

## HẠN CHẾ

## HƯỚNG PHÁT TRIỂN

# TÀI LIỆU THAM KHẢO

1. Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving students’ learning with effective learning techniques: Promising directions from cognitive and educational psychology. Psychological Science in the Public Interest, 14(1), 4–58. https://doi.org/10.1177/1529100612453266 [↑](#endnote-ref-0)
2. Google. (n.d.). Flutter – Build apps for any screen. Flutter.dev. Truy cập tại: https://flutter.dev [↑](#endnote-ref-1)
3. Google. (n.d.). Firebase Documentation. Firebase.google.com. Truy cập tại: https://firebase.google.com/docs [↑](#endnote-ref-2)