**1TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**-------------------------**

ĐỒ ÁN TỐT NGHIỆP KỸ SƯ

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 748021**

**KHÓA: 2020-2025**

***ĐỀ TÀI:***

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÔNG TÁC SINH VIÊN CHO KHOA CNTT TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**SINH VIÊN THỰC HIỆN: LÊ HUY TÙNG**

**LỚP: 20CN1**

**HÀ NỘI 02-2023**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**-------------------------**



ĐỒ ÁN TỐT NGHIỆP KỸ SƯ

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 748021**

**KHÓA: 2020-2025**

***ĐỀ TÀI:***

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CÔNG TÁC SINH VIÊN CHO KHOA CNTT TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**GIÁO VIÊN HƯỚNG DẪN: ThS. NGUYỄN QUỐC HUY SINH VIÊN THỰC HIỆN: LÊ HUY TÙNG**

**LỚP: 20CN1**

**HÀ NỘI 02-2023**

# LỜI CẢM ƠN

Đề tài “Xây dựng ứng dụng quản lý công tác sinh viên khoa CNTT trường đại học Kiến Trúc Hà Nội” là nội dung tôi chọn để nghiên cứu và làm đồ án tốt nghiệp sau 5 năm theo học chương trình đại học chuyên ngành công nghệ thông tin tại trường Trường Đại Học Kiến Trúc Hà Nội.

Để hoàn thành quá trình nghiên cứu và hoàn thiện đồ án này, tôi xin gửi lời cảm ơn chân thành và sâu sắc đến các quý thầy cô trong khoa Công nghệ thông tin, cũng như tất cả các thầy cô Trường Đại Học Kiến Trúc Hà Nội đã truyền đạt kiến thức, kinh nghiệm quý báu của mình cho tôi trong suốt quá trình học tập và nghiên cứu tại trường.

Đặc biệt tôi xin gửi lời cảm ơn đến thầy ThS. Nguyễn Quốc Huy thuộc khoa Công nghệ thông tin – Đại học Kiến Trúc Hà Nội. Thầy đã trực tiếp chỉ bảo và hướng dẫn tôi trong suốt quá trình nghiên cứu để tôi có thể hoàn thiện đồ án này.

Tôi cũng xin gửi lời cảm ơn đến các anh chị và bạn bè sinh viên trong trường đã giúp đỡ, trao đổi kiến thức trong suốt quá trình học tập và thực hiện đồ án này.

Mặc dù đã nỗ lực hoàn thành đồ án này nhưng không thể tránh khỏi những sai sót.

Rất mong nhận được sự góp ý của quý thầy cô và các bạn.

# LỜI CAM ĐOAN

Tôi xin cam đoan đồ án tốt nghiệp “Xây dựng ứng dụng quản lý công tác sinh viên khoa CNTT trường đại học Kiến Trúc Hà Nội” là công trình nghiên cứu của tôi. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

Hà Nội, tháng 02 năm 2023

**SINH VIÊN THỰC HIỆN ĐỒ ÁN**

LÊ HUY TÙNG

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc187436950)

[LỜI CAM ĐOAN 2](#_Toc187436951)

[MỤC LỤC 3](#_Toc187436952)

[DANH MỤC KÝ HIỆU VÀ TỪ VIẾT TẮT 6](#_Toc187436953)

[DANH MỤC BẢNG BIỂU 7](#_Toc187436954)

[DANH MỤC HÌNH ẢNH 8](#_Toc187436955)

[LỜI MỞ ĐẦU 10](#_Toc187436956)

[CHƯƠNG 1. MỞ ĐẦU 11](#_Toc187436957)

[1.1. Sự cần thiết của đề tài 11](#_Toc187436958)

[1.2. Mục đích của việc chọn đề tài 11](#_Toc187436959)

[1.3. Phạm vi ứng dụng của đề tài 11](#_Toc187436960)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 12](#_Toc187436961)

[2.1. Tổng quan về Flutter Framework 12](#_Toc187436962)

[2.1.1. Giới thiệu về Flutter 12](#_Toc187436963)

[2.1.2. Thành phần chính của Flutter 12](#_Toc187436964)

[2.1.3. Kiến trúc ứng dụng Flutter. 13](#_Toc187436965)

[2.2. Tìm hiểu về cách thức hoạt động của Restful API 15](#_Toc187436966)

[2.2.1. Khái niệm Rest API, Restful API 15](#_Toc187436967)

[2.2.2. Cách thức hoạt động của Rest API 18](#_Toc187436968)

[2.3. Khái niệm Firebase 18](#_Toc187436969)

[2.3.1. Tóm tắt lịch sử 18](#_Toc187436970)

[2.3.2. Firebase Cloud Messaging (FCM) 19](#_Toc187436971)

[2.3.3. Firebase Authentication 20](#_Toc187436972)

[2.3.4. Cloud Firestore 21](#_Toc187436973)

[2.3.5. Firebase Cloud Storage 21](#_Toc187436974)

[2.4. Tổng kết chương 21](#_Toc187436975)

[CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 23](#_Toc187436976)

[3.1. Mô tả bài toán 23](#_Toc187436977)

[3.2 . Đối tượng sử dụng hệ thống 23](#_Toc187436978)

[3.3. Yêu cầu hệ thống 23](#_Toc187436979)

[3.3.1. Yêu cầu chức năng 23](#_Toc187436980)

[3.3.2. Yêu cầu phi chức năng 23](#_Toc187436981)

[3.4. Sơ đồ phân rã chức năng 24](#_Toc187436982)

[3.5. Biểu đồ Use-case 24](#_Toc187436983)

[3.5.1. Use case đăng nhập 25](#_Toc187436984)

[3.5.2. Use case quản lý thông tin sinh viên 26](#_Toc187436985)

[3.5.3. Use case tính điểm rèn luyện 28](#_Toc187436986)

[3.5.4. Use case quản lý tin tức 28](#_Toc187436987)

[3.6.Thiết kế cơ sở dữ liệu 30](#_Toc187436988)

[3.6.1.Bảng user 30](#_Toc187436989)

[3.6.2.Bảng sinh viên 30](#_Toc187436990)

[3.6.3.Bảng lớp 30](#_Toc187436991)

[3.6.5.Bảng tin tức 31](#_Toc187436992)

[3.6.6.Bảng điểm rèn luyện 31](#_Toc187436993)

[**CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG** 32](#_Toc187436994)

[4.1.Tạo dự án Flutter 32](#_Toc187436995)

[4.2. Một số hình ảnh giao diện 34](#_Toc187436996)

[4.2.1. Giao diện login 34](#_Toc187436997)

[4.2.2. Giao diện màn home 35](#_Toc187436998)

[4.2.3. Giao diện màn thông tin lớp 36](#_Toc187436999)

[4.2.4. Giao diện màn phiếu điểm rèn luyện 37](#_Toc187437000)

[4.2.5. Giao diện màn bài viết 38](#_Toc187437001)

[4.2.6. Giao diện màn thêm bài viết 39](#_Toc187437002)

[4.2.7. Giao diện màn thông báo 40](#_Toc187437003)

[4.2.8. Giao diện màn tài khoản 41](#_Toc187437004)

[4.2.9. Giao diện màn sửa thông tin cá nhân 42](#_Toc187437005)

[4.2.10. Giao diện màn đổi mật khẩu 43](#_Toc187437006)

[DANH MỤC TÀI LIỆU THAM KHẢO 44](#_Toc187437007)

# DANH MỤC KÝ HIỆU VÀ TỪ VIẾT TẮT

|  |  |
| --- | --- |
| API | Command Line Interface |
| SDK | Software Development Kit |
| UI | User Interface |
| REST | Representational State Transfer |
| HTTP | Hypertext Transfer Protocol |
| URL | Uniform Resource Locator |
| JSON | JavaScript Object Notation |
| FCM | Firebase Cloud Messaging |

# DANH MỤC BẢNG BIỂU

[Bảng 3. 1 Các yêu cầu phi chức năng 24](#_Toc187437256)

[Bảng 3. 2 Bảng mô tả chức năng đăng nhập và đăng xuất 26](#_Toc187437257)

[Bảng 3. 3 Bảng mô tả chức năng quản lý thông tin sinh viên 28](#_Toc187437258)

[Bảng 3. 4 Bảng mô tả chức năng điểm rèn luyện 28](#_Toc187437259)

[Bảng 3. 5 Bảng mô tả chức năng quản lý tin tức 29](#_Toc187437260)

[Bảng 3. 8 Mô tả bảng dữ liệu người dùng 30](#_Toc187437261)

[Bảng 3. 9 Mô tả bảng dữ liệu sinh viên 30](#_Toc187437262)

[Bảng 3. 10 Mô tả bảng dữ liệu lớp 31](#_Toc187437263)

[Bảng 3. 12 Mô tả bảng dữ liệu tin tức 31](#_Toc187437264)

[Bảng 3. 13 Mô tả bảng dữ liệu điểm rèn luyện 31](#_Toc187437265)

# DANH MỤC HÌNH ẢNH

[Hình 2. 1 Kiến trúc tổng quát Flutter 13](#_Toc187437128)

[Hình 2. 2 Layer Flutter 14](#_Toc187437129)

[Hình 2. 3 Kiến trúc ứng dụng widget 15](#_Toc187437130)

[Hình 2. 4 Rest API Design 15](#_Toc187437131)

[Hình 2. 5 Rest API 17](#_Toc187437132)

[Hình 2. 6 Mô hình hoạt động của Rest API 18](#_Toc187437133)

[Hình 2. 7 Firebase Clould Messageing 20](#_Toc187437134)

[Hình 2. 8 Firebase Authentication 20](#_Toc187437136)

[Hình 2. 9 Clould Firestore 21](#_Toc187437137)

[Hình 3.1. Sơ đồ phân rã chức năng 24](#_Toc187437138)

[Hình 3.2. Biểu đồ Use case tổng quát 25](#_Toc187437139)

[Hình 3.3. Biểu đồ Usecase đăng nhập 25](#_Toc187437140)

[Hình 3.4. Biểu đồ Usecase quản lý sinh viên 27](#_Toc187437141)

[Hình 3.5. Biểu đồ Usecase tính điểm rèn luyện 28](#_Toc187437142)

[Hình 3.6. Biểu đồ Usecase quản lý tin tức 29](#_Toc187437143)

[Hình 4.1. Tạo project Flutter 32](#_Toc187437144)

[Hình 4.2. Cấu trúc thư mục project Flutter 33](#_Toc187437145)

[Hình 4.3. Giao diện màn login 34](#_Toc187437146)

[Hình 4.4. Giao diện màn home 35](#_Toc187437147)

[Hình 4.5. Giao diện màn thông tin lớp 36](#_Toc187437148)

[Hình 4.6. Giao diện màn phiếu điểm rèn luyện 37](#_Toc187437149)

[Hình 4.7. Giao diện màn bài viết 38](#_Toc187437150)

[Hình 4.8. Giao diện màn thêm bài viết 39](#_Toc187437151)

[Hình 4.9. Giao diện màn thông báo 40](#_Toc187437152)

[Hình 4.10. Giao diện màn tài khoản 41](#_Toc187437153)

[Hình 4.11. Giao diện màn sửa thông tin cá nhân 42](#_Toc187437154)

[Hình 4.12. Giao diện màn đổi mật khẩu 43](#_Toc187437155)

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, Công nghệ Thông tin là một trong những ngành có vị thế dẫn đầu và có vai trò rất lớn trong sự phát triển chung đó. Các ứng dụng của công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong mọi lĩnh vực của đời sống. Là một phần của Công nghệ Thông tin, Công nghệ mobile đang có được sự phát triển mạnh mẽ và phổ biến rất nhanh bởi những lợi ích mà nó mang lại cho cộng đồng là rất lớn.

Hiện nay ta có thể quản lý gần như mọi thứ thông qua chiếc điện thoại thông minh của mình. Để hỗ trợ mảng quản lý công tác sinh viên cho khoa CNTT trường đại học Kiến Trúc tôi quyết định sẽ tạo ra ứng dụng quản lý này.

Ứng dụng mảng quản lý công tác sinh viên cho khoa CNTT trường đại học Kiến Trúc sẽ đóng vai trò quan trọng trong việc quản lý sinh viên cũng như các hoạt động của khoa CNTT trường đại học Kiến Trúc. Giáo viên có thể quản lý các thông tin của sinh viên như điểm rèn luyện, thành tích học tập cũng như các sự kiện, tin tức của khoa và nhà trường, …

Mục đích của đề tài nhằm tập trung vào nghiên cứu phát triển ứng dụng di động đa nền tảng dựa trên Dart sử dụng Flutter Framework. Tìm hiểu về bài toán quản lý công tác sinh viên trên thiết bị di động, nghiên cứu sử dụng công nghệ Flutter Framework. Nội dung của đồ án được cấu trúc như sau:

**Chương 1. Mở đầu**

**Chương 2. Cơ sở lý thuyết.**

**Chương 3. Phân tích thiết kế hệ thống.**

**Chương 4. Xây dựng ứng dụng.**

**Chương 5. Kết luận và hướng phát triển**

# CHƯƠNG 1. MỞ ĐẦU

## Sự cần thiết của đề tài

Trong bối cảnh quản lý công tác sinh viên ngày càng phức tạp, việc dựa vào các phương pháp thủ công truyền thống đã bộc lộ nhiều hạn chế. Những quy trình thủ công không chỉ tốn kém thời gian và công sức mà còn tiềm ẩn nguy cơ sai sót, gây khó khăn trong việc truy xuất và cập nhật thông tin. Để giải quyết những vấn đề này, việc xây dựng một hệ thống quản lý tự động là một giải pháp tối ưu. Hệ thống này không chỉ nâng cao hiệu quả công việc, giảm thiểu sai sót mà còn tạo ra một môi trường quản lý minh bạch, chuyên nghiệp. Với mục tiêu xây dựng một nền tảng quản lý toàn diện, tôi đã lựa chọn Flutter Framework để phát triển ứng dụng, nhằm đảm bảo giao diện thân thiện, khả năng tương thích cao trên nhiều nền tảng và đáp ứng nhu cầu sử dụng đa dạng của sinh viên và cán bộ quản lý. Ứng dụng này hứa hẹn sẽ trở thành một công cụ hữu ích, hỗ trợ đắc lực cho công tác quản lý sinh viên tại khoa CNTT và có tiềm năng mở rộng ra các khoa khác trong trường.

## Mục đích của việc chọn đề tài

Việc chọn đề tài này nhằm mục đích chính là nhằm nghiên cứu về hệ điều hành ứng dụng đa nền tảng cả android và IOS sử dụng Flutter Framework, tìm hiểu về ứng dụng trên thiết bị di động. Tiếp đến là tìm hiểu các ngôn ngữ lập trình hỗ trợ mới giúp có thể xây dựng được một ứng dụng android, IOS và ở trong phạm vi đề tài này nhằm tìm hiểu về công cụ hỗ trợ lập trình đa nền tảng Flutter Framework và nhằm xây dựng ứng dụng quản lý công tác sinh viên.

## Phạm vi ứng dụng của đề tài

Với đề tài này mục đích ứng dụng của tôi nhằm đưa ứng dụng đến tất cả giảng viên trong khoa CNTT trường đại học Kiến Trúc. Qua ứng dụng được xây dựng giảng viên sẽ quản lý được tất cả sinh viên, sự kiện, tin tức, thành tích trong khoa CNTT.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## Tổng quan về Flutter Framework

### Giới thiệu về Flutter

Flutter là một bộ công cụ phát triển phần mềm giao diện người dùng với mã nguồn mở do Google tạo ra. Flutter được sử dụng để phát triển các ứng dụng đa nền tảng cho nhiều hệ điều hành như: IOS, Android, Linux,… từ một cơ sở mã nguồn duy nhất.

Tuy Flutter được Google công bố vào đầu năm 2015 nhưng đến tháng 5 năm 2017 nền tảng này mới chính thức ra mắt thị trường. Với mục đích xây dựng và phát triển các ứng dụng chất lượng, mang đến cho người dùng những trải nghiệm tuyệt vời nhất trên hệ điều hành iOS và Android

### Thành phần chính của Flutter

Có một điều các bạn nên biết rằng, Flutter không chỉ được tạo nên bởi một bộ phận duy nhất. Có hai thành phần chính tạo nên nền tảng Flutter mạnh mẽ và hữu dụng:

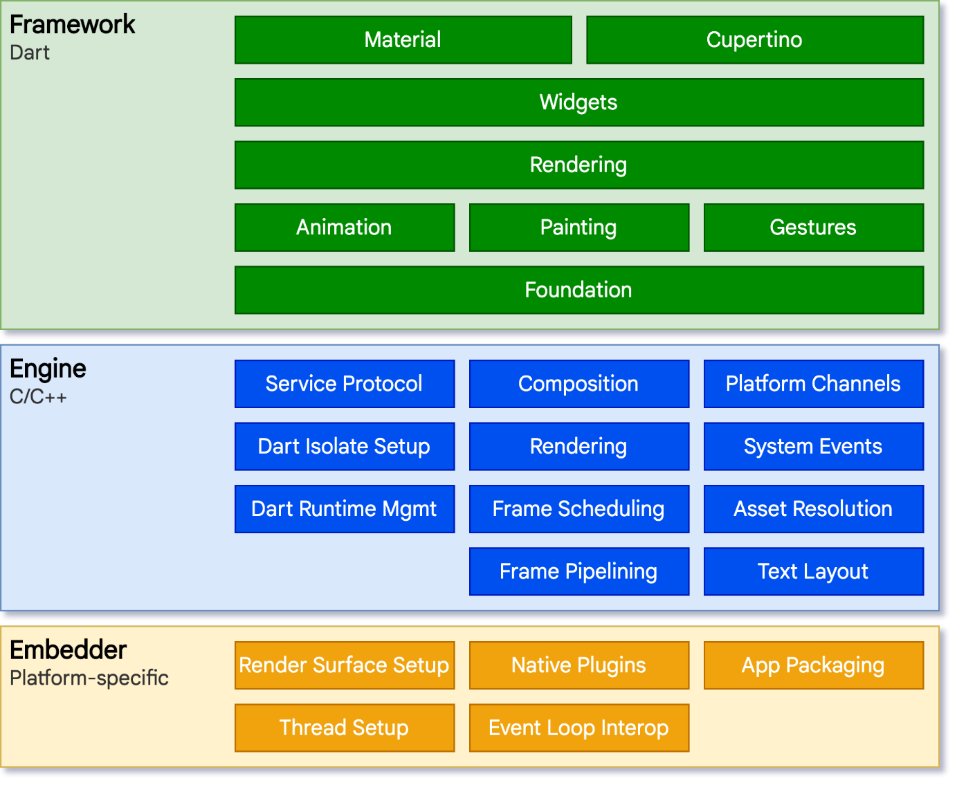
* Một SDK (Software Development Kit): SDK được biết đến là bộ sưu tập những công cụ giúp cho người dùng phát triển được ứng dụng nền của mình. Người dùng sử dụng công cụ này để hỗ trợ phần lớn các thao tác phát triển ứng dụng. Trong đó bao gồm các trình biên dịch mã thành các mã gốc dành riêng cho hai hệ điều hành IOS và Android.
* Một Framework (UI Library based on widgets): Framework cho phép người dùng tập hợp những thành phần của giao diện. Chính vì điều đó, việc tái sử dụng những mã code trên Framework trở nên dễ dàng hơn. Tiết kiệm thời gian và tính hữu dụng của Framework các bạn có thể tự tìm hiểu để biết thêm chi tiết. Tuy nhiên, điều này thật sự có ích với việc cá nhân hóa tùy theo nhu cầu của người dùng.
* Đây chính là hai thành phần chính trả lời cho câu hỏi thành phần quan trọng nhất của Flutter là gì. Hầu hết các nền tảng xây dựng web hay ứng dụng hiện nay đều có chứa framework. Tùy theo mục đích của nền tảng và người sử dụng mà framework này có thể thay đổi khác nhau. Tuy nhiên nhìn chung, những tính năng mà nó đem lại thật sự hiệu quả.

Flutter có hai thành phần chính đó là:

* Một Software Development Kit (SDK): Ở đây tích hợp các công cụ có khả năng phát triển các ứng dụng của bạn, trong đó có các công cụ giúp bạn biên dịch mã của mình thành mã máy gốc (mã cho Android và iOS).
* Một UI Library based on widgets (Framework): Tổng hợp các thành phần giao diện người dùng (UI) có thể tái sử dụng như: text input, button, slider, … hỗ trợ bạn có thể cá nhân hóa theo nhu cầu sử dụng.

### Kiến trúc ứng dụng Flutter.

Flutter được thiết kế như một hệ thống nhiều lớp, có thể mở rộng. Nó tồn tại dưới dạng một loạt các thư viện độc lập mà mỗi thư viện phụ thuộc vào lớp bên dưới. Không có lớp nào có đặc quyền truy cập vào các lớp bên dưới, mỗi phần framework được thiết kế để có thể tùy chọn và thay thế. Để hiểu hơn các bạn xem hình ảnh bên dưới đây:

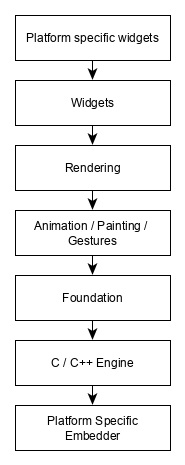


Hình 2. 1 Kiến trúc tổng quát Flutter

* + - 1. **Tổng quan về layer của Flutter**

Một khái niệm quan trọng của Flutter framework đó là các thành phần sẽ được nhóm lại theo độ phức tạp và được sắp xếp rõ ràng trong các tầng có độ phức tạp giảm dần.

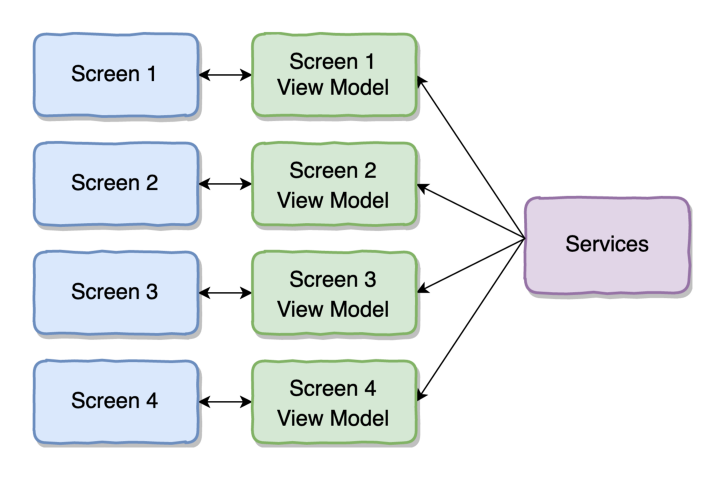
Một layer (tầng, lớp) được tạo thành bằng việc sử dụng các class tiếp theo ngay canh nó. Top của tất cả các layer là các widget đặc biệt cho Android và iOS. Layer tiếp theo là widget gốc của flutter. Tiếp nữa là Rendering layer, đây là level thấp nhất trong việc sinh các thành phần của flutter app. Layer tiếp theo là nền tảng gốc hệ điều hành.



Hình 2. Layer Flutter

* + - 1. **Kiến trúc của một ứng dụng Flutter**

Trong ứng dụng Flutter, bản thân chính ứng dụng đã là một widget, một ứng dụng cũng có thể là tập hợp của nhiều ứng dụng nhỏ khác (nhiều widget con khác) được lồng vào nhau, liên kết với nhau để tạo ra một ứng dụng chính.



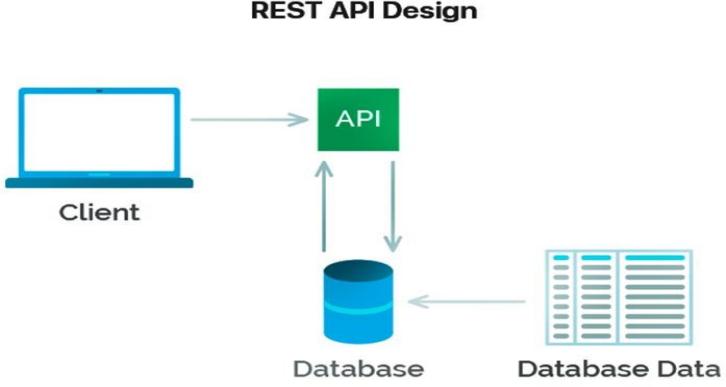
Hình 2. Kiến trúc ứng dụng widget

## Tìm hiểu về cách thức hoạt động của Restful API

### Khái niệm Rest API, Restful API

#### Giới thiệu về Rest API

Rest API là một ứng dụng được dùng để chuyển đổi cấu trúc của dữ liệu có những phương thức giúp kết nối với các thư viện và ứng dụng khác nhau. Rest Api thường không được xem là công nghệ, mà nó là giải pháp giúp tạo ra những ứng dụng web services chuyên dụng để thay thế cho nhiều kiểu khác như: SOAP, WSDL, …



Hình 2. Rest API Design

#### Giới thiệu về Restful API

Restful API là một trong những tiêu chuẩn được sử dụng để thiết kế API cho các ứng dụng web, để tiện hơn cho việc quản lý các resource. Rest API thường chú trọng vào các tài nguyên của hệ thống bao gồm: ảnh, văn bản, âm thanh, video hoặc các dữ liệu di động,… Nó thường bao gồm các trạng thái tài nguyên đã được định dạng sẵn và được truyền tải thông qua HTTP.

Một chức năng quan trọng nhất của REST là: quy định các cách sử dụng HTTP method chẳng hạn như: Post, Get, Delete, Put, … và cách có thể định dạng các URL cho ứng dụng web để có thể quản lý được các resource.

Restful không quy định logic code ứng dụng và không được giới hạn bởi những ngôn ngữ lập trình ứng dụng. Bất kỳ framework hoặc ngôn ngữ nào cũng có thể được dùng để thiết kế được một Restful API

#### Giới thiệu API

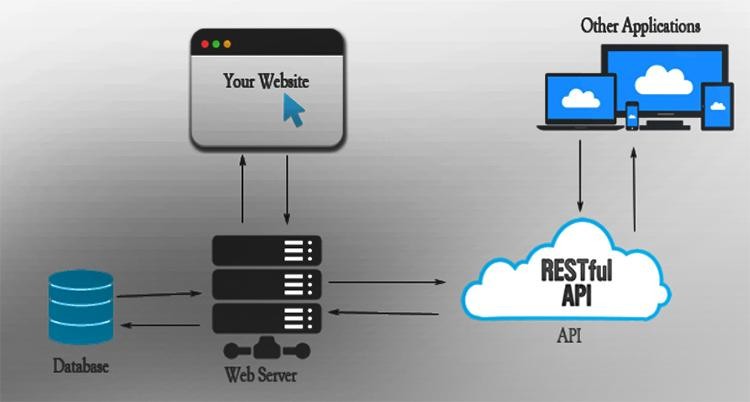
API là từ viết tắt của cụm từ Application Programming Interface, đây là tập hợp những quy tắc và cơ chế mà theo đó thì: Một ứng dụng hoặc một thành phần nào đó sẽ tương tác với một ứng dụng hoặc một số thành phần khác.

API có thể sẽ được trả về dữ liệu mà người dùng cần cho chính ứng dụng của bạn với những kiểu dữ liệu được dùng phổ biến như JSON hoặc XML.

#### Giới thiệu REST

Rest là từ viết tắt của Representational State Transfer: Nó là một trong những dạng chuyển đổi cấu trúc, với kiểu kiến trúc thường được viết API. Rest thường sử dụng dụng phương thức HTTP đơn giản để có thể tạo ra giao tiếp giữa các máy.

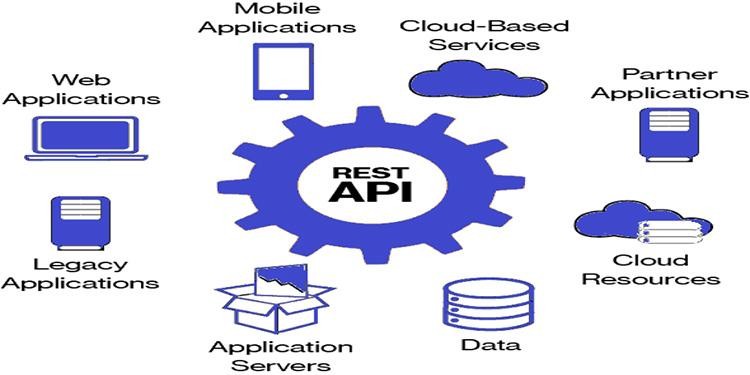
Bởi vì thế, thay vì phải sử dụng một URL cho việc xử lý một số thông tin của người dùng thì Rest sẽ yêu cầu HTTP như: GET, POST, DELETE,… đến với bất kỳ một URL để được xử lý dữ liệu.



Hình 2. Rest API

### Cách thức hoạt động của Rest API

Sau khi bạn đã biết được RESTful API là gì thì sau đây chính là nguyên lý hoạt động của Restful API. Tương tự với các giao thức truyền thông hoặc các cấu trúc dữ liệu khác thì: để bạn có thể tìm hiểu được bản chất của vấn đề thì đầu tiên việc hiểu rõ nguyên lý hoạt động của nó vô cần thiết.



Hình 2. 6 Mô hình hoạt động của Rest API

REST thường hoạt động chủ yếu dựa vào những giao thức:

HTTP, các cơ sở hoạt động cơ bản nêu trên sẽ được sử dụng những phương thức HTTP riêng biệt

* GET: Trả về với một Resource hoặc có một danh sách Resource.
* POST: Nó hỗ trợ tạo mới một Resource.
* PUT: Thường hỗ trợ cho việc cập nhật các thông tin cho Resource.
* DELETE: Xóa một Resource.

Đây là phương thức hoạt động được gọi là Crud tương ứng với một Create, Read, Update, Delete ý nghĩa theo thứ tự là: Tạo, đọc, sửa và xóa.

Cho đến hiện tại, thì đa số các lập trình viên viết Restful API đều lựa chọn JSON làm format chính thức, tuy nhiên vẫn còn nhiều người lựa chọn XML

## Khái niệm Firebase

### Tóm tắt lịch sử

Trở lại những năm 2011, trước khi Firebase là Firebase bây giờ thì nó là một startup, được gọi là Envolve. Envolve cung cấp cho các nhà phát triển một API cho phép tích hợp chức năng trò chuyện trực tuyến vào trang web của họ. Điều thú vị là mọi người đã sử dụng Envolve để truyền dữ liệu ứng dụng chứ không chỉ là tin nhắn trò chuyện đơn thuần.

Các nhà phát triển đang sử dụng Envolve để đồng bộ hóa dữ liệu ứng dụng chẳng hạn như trạng thái trò chơi trong thời gian thực trên người dùng của họ. Điều này đã dẫn dắt những người sáng lập Envolve, James Tamplin và Andrew Lee, tách riêng chat system và the real-time architecture. Vào tháng 4 năm 2012, Firebase được tạo ra dưới dạng công ty riêng biệt Backend-as-a-Service với chức năng thời gian thực. Sau khi được Google mua lại vào năm 2014, Firebase nhanh chóng phát triển thành ứng dụng đa năng của nền tảng di động và web hiện nay.

### Firebase Cloud Messaging (FCM)

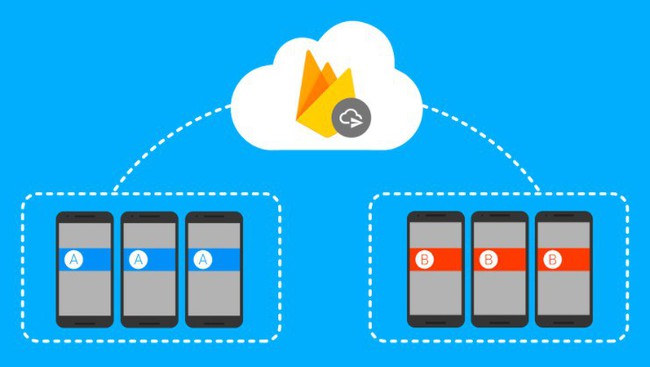
**Firebase Cloud Messaging (FCM)** cung cấp kết nối đáng tin cậy và tiết kiệm pin giữa máy chủ và thiết bị cho phép bạn gửi và nhận tin nhắn cũng như thông báo trên iOS, Android và web miễn phí.

Bạn có thể gửi tin nhắn thông báo (giới hạn 2KB) và tin nhắn dữ liệu (giới hạn 4KB).

Sử dụng FCM, bạn có thể dễ dàng nhắm mục tiêu các thư bằng cách sử dụng các phân đoạn được xác định trước hoặc tạo các phân đoạn của riêng bạn bằng cách sử dụng nhân khẩu học và hành vi. Bạn có thể gửi tin nhắn tới một nhóm thiết bị được đăng ký với các chủ đề cụ thể.

FCM có thể gửi tin nhắn ngay lập tức hoặc vào thời điểm tương lai theo múi giờ địa phương của người dùng. Bạn có thể gửi dữ liệu ứng dụng tùy chỉnh như đặt mức độ ưu tiên, âm thanh và ngày hết hạn cũng như theo dõi các sự kiện chuyển đổi tùy chỉnh.

Điểm nổi bật của FCM là hầu như không có bất kỳ mã hóa nào liên quan! FCM được tích hợp hoàn toàn với Firebase Analytics, mang đến cho bạn sự tương tác chi tiết (detailed engagement) và theo dõi chuyển đổi (conversion tracking).



Hình 2. 7 Firebase Clould Messageing

### Firebase Authentication

Firebase Authentication cung cấp một giải pháp hoàn chỉnh cho việc quản lý xác thực người dùng. Dịch vụ này hỗ trợ nhiều phương thức đăng nhập, bao gồm đăng nhập bằng email/mật khẩu, đăng nhập bằng tài khoản mạng xã hội phổ biến (Google, Facebook, Twitter, GitHub, v.v.), xác thực bằng số điện thoại và khả năng tùy chỉnh quy trình xác thực theo yêu cầu cụ thể của ứng dụng, đảm bảo tính bảo mật và linh hoạt cho ứng dụng.

A group of cell phones with different icons

Description automatically generated

Hình 2. 8 Firebase Authentication

### Cloud Firestore

Cloud Firestore là một cơ sở dữ liệu NoSQL linh hoạt và có khả năng mở rộng cao, được thiết kế tối ưu cho các ứng dụng di động, web và máy chủ. Dữ liệu trong Firestore được tổ chức thành các tài liệu (documents) và bộ sưu tập (collections), cho phép truy vấn dữ liệu một cách mạnh mẽ và hỗ trợ đồng bộ dữ liệu theo thời gian thực, giúp ứng dụng luôn cập nhật và phản hồi nhanh chóng.

A computer and mobile devices

Description automatically generated

Hình 2. 9 Clould Firestore

### Firebase Cloud Storage

Firebase Cloud Storage cung cấp khả năng lưu trữ và truy cập các tệp như hình ảnh, video và âm thanh trên đám mây. Dịch vụ này có khả năng mở rộng linh hoạt, tích hợp tốt với các dịch vụ khác của Firebase và đảm bảo an toàn cho dữ liệu được lưu trữ, cung cấp một giải pháp lưu trữ mạnh mẽ và đáng tin cậy cho ứng dụng.

## Tổng kết chương

Chương này đã cung cấp nền tảng kiến thức quan trọng cho việc xây dựng ứng dụng di động và web hiện đại. Đầu tiên, chúng ta đã khám phá Flutter Framework, một bộ công cụ UI mạnh mẽ của Google cho phép phát triển ứng dụng đa nền tảng từ một mã nguồn duy nhất. Flutter nổi bật với hiệu suất cao, khả năng tùy biến giao diện linh hoạt và khả năng phát triển nhanh chóng nhờ tính năng Hot Reload. Tiếp theo, chúng ta đã tìm hiểu về cách thức hoạt động của RESTful API, một kiến trúc thiết kế API phổ biến cho phép ứng dụng giao tiếp và trao đổi dữ liệu với máy chủ. Hiểu rõ về RESTful API là chìa khóa để xây dựng các ứng dụng động, có khả năng truy cập và xử lý dữ liệu từ backend. Cuối cùng, chúng ta đã giới thiệu Firebase, một nền tảng BaaS (Backend as a Service) cung cấp nhiều dịch vụ hỗ trợ phát triển ứng dụng, từ quản lý cơ sở dữ liệu (Firestore), xác thực người dùng (Authentication), lưu trữ tệp (Cloud Storage) đến nhắn tin (FCM).

Ba khái niệm này liên kết chặt chẽ với nhau trong quá trình phát triển ứng dụng. Flutter cung cấp giao diện người dùng (frontend), RESTful API đóng vai trò là cầu nối để Flutter giao tiếp với backend, và Firebase cung cấp các dịch vụ backend giúp đơn giản hóa việc quản lý dữ liệu, xác thực người dùng và nhiều chức năng khác. Sự kết hợp giữa Flutter, RESTful API và Firebase cho phép xây dựng các ứng dụng hoàn chỉnh, mạnh mẽ, có khả năng mở rộng và tiết kiệm thời gian phát triển.

# CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương 3 của báo cáo sẽ trình bày phân tích thiết kế hệ thống cho ứng dụngquản lý công tác sinh viên cho khoa CNTT trường đại học Kiến Trúc. Trước tiên là đặc tả yêu cầu của hệ thống, sau đó sẽ đi phân tích thiết kế chi tiết hệ thống thông qua các biểu đồ usecase, biểu đồ tuần tự, biểu đồ hoạt động, biểu đồ lớp và cuối cùng thiết kế cơ sở dữ liệu.

## 3.1. Mô tả bài toán

Hệ thống quản lý công tác sinh viên nhằm tự động hóa quy trình quản lý thông tin sinh viên, lớp học, điểm danh, điểm rèn luyện và cập nhật tin tức. Hệ thống có các chức năng chính bao gồm:

* Quản lý thông tin lớp.
* Quản lý thông tin sinh viên.
* Quản lý điểm rèn luyện.
* Cập nhật tin tức và thông báo.

## . Đối tượng sử dụng hệ thống

* Nhân viên phòng đào tạo.
* Ban lãnh đạo khoa.

## 3.3. Yêu cầu hệ thống

### 3.3.1. Yêu cầu chức năng

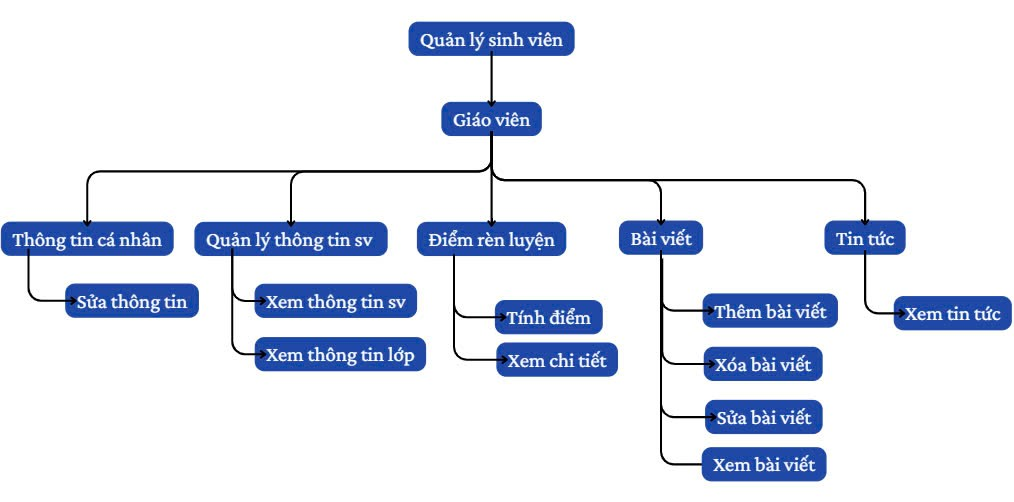
* Quản lý danh sách sinh viên, bao gồm thông tin cá nhân, thông tin lớp học.
* Thống kê điểm rèn luyện.
* Tạo và gửi tin tức, thông báo.

### 3.3.2. Yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **Mục** | **Tên yêu cầu** | **Mô tả yêu cầu** |
| 1 | Giao diện | Giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với mọi người dùng. |
| 2 | Tốc độ xử lý | Hệ thống phải xử lý nhanh chóng và chính xác. |
| 3 | Bảo mật | Tính bảo mật và độ an toàn cao. |
| 4 | Tương thích | Tương thích với đa phần các trình duyệt web hiện tại. |

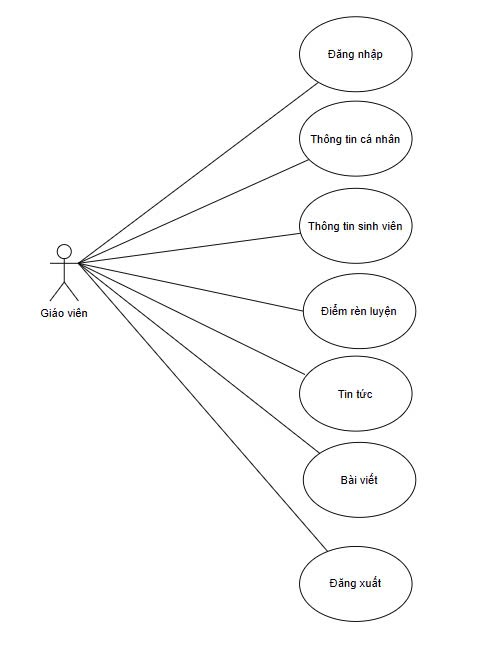
***Bảng 3. 1 Các yêu cầu phi chức năng***

## 3.4. Sơ đồ phân rã chức năng

****

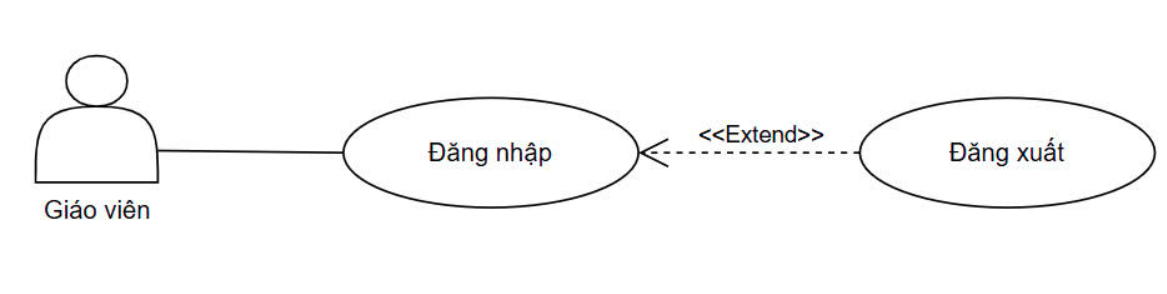
Hình 3.1. Sơ đồ phân rã chức năng

## 3.5. Biểu đồ Use-case

****

Hình 3.2. Biểu đồ Use case tổng quát

### 3.5.1. Use case đăng nhập

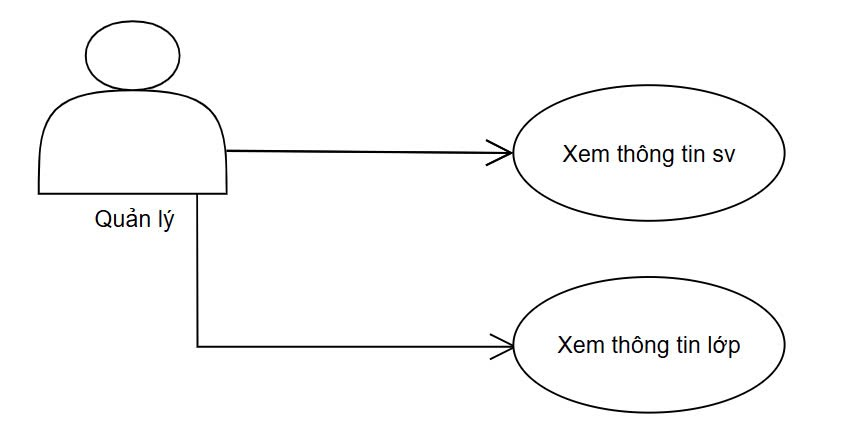
****

Hình 3.3. Biểu đồ Usecase đăng nhập

|  |  |
| --- | --- |
| **Tên Usecase** | **Đăng nhập / Đăng xuất** |
| Tác nhân | Giáo viên, Nhân viên phòng đào tạo, Ban lãnh đạo khoa |
| Tiền điều kiện (đăng nhập) | Người dùng đã có tài khoản trong hệ thống và đăng nhập thành công |
| Mục đích  (đăng nhập) | Xác thực người dùng  Người dùng đăng nhập vào hệ thống thông qua “Email” và “Mật khẩu” |
| Dòng sự kiện chính (đăng nhập) | **Dòng sự kiện chính:**   1. Người dùng muốn đăng nhập vào hệ thống 2. Hệ thống hiển thị giao diện đăng nhập 3. Người dùng nhập “Email” và “Mật khẩu” 4. Hệ thống kiểm tra và xác nhận thông tin đăng nhập 5. Hiển thị giao diện chính của hệ thống 6. Kết thúc Usecase. |
| Dòng sự kiện phụ (đăng nhập) | 1. Nếu người dùng nhập sai hoặc không nhập tên đăng nhập/mật khẩu thì không thực hiện đăng nhập 2. Nếu người dùng nhập sai mật khẩu sẽ thông báo cho người dùng biết |
| Hậu điều kiện (đăng nhập) | **Trường hợp thành công**: Hệ thống sẽ hiển thị giao diện quản lý chính. Người dùng có thể thực hiện các chức năng theo đúng quyền hạn của mình.  **Trường hợp thất bại**: Hệ thống sẽ đưa ra thông báo” Tài khoản nhập sai hoặc không tồn tại” và yêu cầu đăng nhập lại. |
| Tiền điều kiện (đăng xuất) | Người dùng đã đăng nhâp trước khi Usecase xảy ra |
| Hậu điều kiện (đăng xuất) | Trường hợp thành công: Hệ thống sẽ hiển thị giao diện màn hình trang chủ hệ thống ở trạng thái chưa đăng nhập  Trường hợp thất bại: Hệ thống sẽ đưa ra thông báo, yêu cầu thức hiện lại |
| Mục đích (đăng xuất) | Đăng xuất ra khỏi hệ thống |
| Dòng sự kiện chính (đăng xuất) | 1. Người dùng muốn đăng xuất ra khỏi hệ thống 2. Người dùng nhấn vào biểu tượng đăng xuất 3. Hiển thị giao diện trang chủ chưa đăng nhập 4. Kết thúc Usecase |

***Bảng 3. 2 Bảng mô tả chức năng đăng nhập và đăng xuất***

### 3.5.2. Use case quản lý thông tin sinh viên

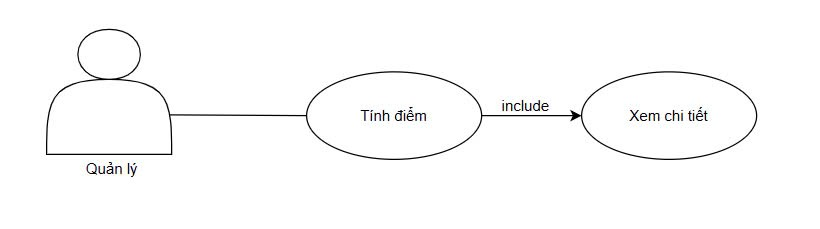
****

Hình 3.4. Biểu đồ Usecase quản lý sinh viên

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý thông tin sinh viên** |
| Tác nhân | Thành viên của hệ thống |
| Tiền điều kiện | Thành viên phải đăng nhập vào hệ thống |
| Mục đích | UC cho phép thành viên thêm, xoá, thay đổi thông tin sinh viên |
| Dòng sự kiện chính | **Dòng sự kiện chính:**   1. Thành viên chọn kiểu tác động lên sinh viên: Thêm sinh viên, loại bỏ sinh viên, thay đổi thông tin sinh viên. 2. Thêm sinh viên: 3. Hệ thống hiển thị form nhập thông tin sinh viên mới 4. Người quản lý nhập thông tin sinh viên mới 5. Nhấn nút thêm sinh viên 6. Nếu nhập thành công thì thực hiện bước cuối. Nếu sai thực hiện luồng sự kiện rẽ nhánh A1. 7. Lưu thông tin sinh viên. 8. Thay đổi thông tin sinh viên 9. Hệ thống hiển thị form sửa thông tin của sinh viên 10. Thành viên nhập các thông tin cần thay đổi. 11. Nhấn nút lưu thông tin 12. Nếu việc thay đổi thành công thì thực hiện bước cuối. Nếu sai thực hiện luồng sự kiện rẽ nhánh A1. 13. Lưu thông tin 14. Xóa sinh viên     1. Người quản lý chọn sinh viên cần xoá     2. Nhấn nút xoá để thực hiện loại sinh viên.     3. Hệ thống hiển thị thông báo xác nhận loại bỏ. Nếu người quản lý đồng ý thì thực hiện bước 14. Nếu không đồng ý thì thực hiện bước 15     4. Thông báo sinh viên đã được xoá     5. Hệ thống hiển thị lại danh sách sinh viên. 15. UC kết thúc 16. Luồng sự kiện rẽ nhánh 17. Luồng sự kiện rẽ nhánh A1: 18. Hệ thống thông báo việc nhập dữ liệu không hợp lệ 19. Người quản lý nhập lại thông tin 20. Nhấn nút lưu thông tin |
| Hậu điều kiện | Các thông tin về sinh viên được cập nhật vào cơ sở dữ liệu |

***Bảng 3. 3 Bảng mô tả chức năng quản lý thông tin sinh viên***

### 3.5.3. Use case tính điểm rèn luyện

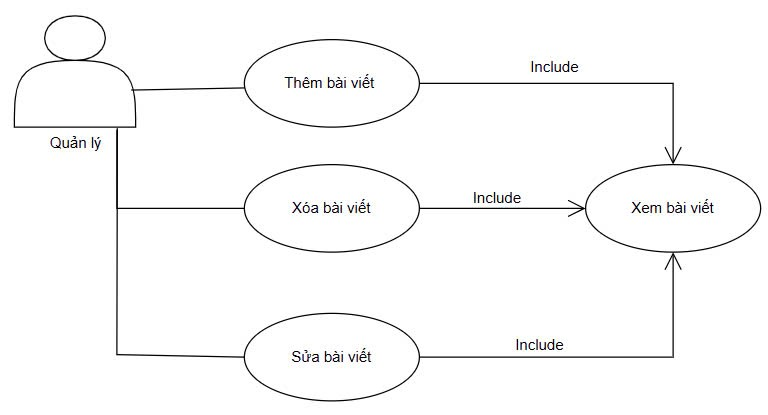
****

Hình 3.5. Biểu đồ Usecase tính điểm rèn luyện

|  |  |
| --- | --- |
| **Tên Usecase** | **Tính điểm rèn luyện** |
| Tác nhân | Thành viên hệ thống |
| Tiền điều kiện | Thành viên phải đăng nhập vào hệ thống |
| Mục đích | Tính điểm rèn luyện cho sinh viên |
| Dòng sự kiện chính | **Dòng sự kiện chính:**   1. Thành viên chọn vào học sinh cần tính điểm 2. Thành viên chọn kỳ học, năm học cần tính điểm 3. Hiển thị các tiêu chí tính điểm thành viên tích chọn 4. Lưu thông tin 5. Uc kết thúc |
| Hậu điều kiện | Điểm rèn luyện của sinh viên được cập nhật vào cơ sở dữ liệu |

***Bảng 3. 4 Bảng mô tả chức năng điểm rèn luyện***

### 3.5.4. Use case quản lý tin tức

****

Hình 3.6. Biểu đồ Usecase quản lý tin tức

|  |  |
| --- | --- |
| **Tên Usecase** | **Quản lý tin tức** |
| Tác nhân | Thành viên hệ thống |
| Tiền điều kiện | Thành viên phải đăng nhập vào hệ thống |
| Mục đích | UC cho phép thành viên thêm, xoá, thay đổi bài viết tin tức |
| Dòng sự kiện chính | **Dòng sự kiện chính:**   1. Thành viên chọn vào icon thêm để thêm tin tức mới 2. Hệ thống hiển thị form nhập bài viết tin tức mới 3. Thành viên nhập nội dung tin tức mới 4. Bấm nút thêm bài viết 5. Bấm vào icon xóa để xóa bài viết 6. Hiển thị thông báo chắc chắn muốn xóa 7. Bấm “Xóa” hiển thị xóa thành công. Nếu không bấm “Hủy” 8. Bấm vào icon sửa để sửa bài viết 9. Hệ thống hiển thị form nhập bài viết tin tức cũ 10. Thành viên nhập thông tin cần sửa 11. Bấm cập nhật, hiển thị cập nhật thành công 12. Uc kết thúc |
| Hậu điều kiện | Điểm rèn luyện của sinh viên được cập nhật vào cơ sở dữ liệu |

***Bảng 3. 5 Bảng mô tả chức năng quản lý tin tức***

## 3.6.Thiết kế cơ sở dữ liệu

### ****3.6.1.Bảng user****

Thông tin người dùng.

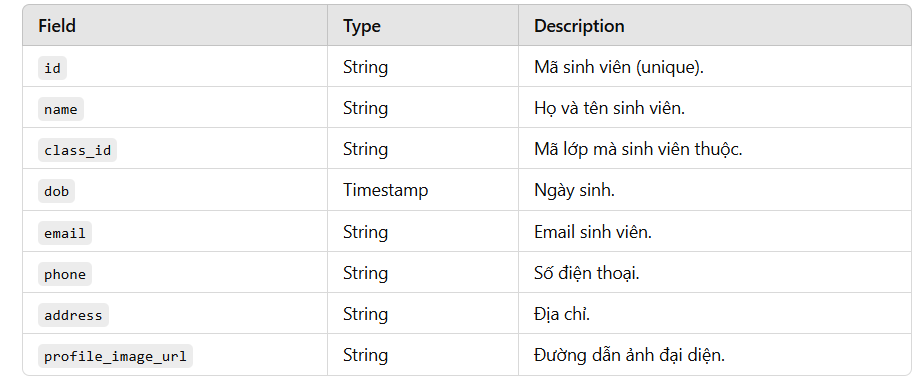
A screenshot of a computer

Description automatically generated

***Bảng 3. 8 Mô tả bảng dữ liệu người dùng***

### **3.6.2.Bảng sinh viên**

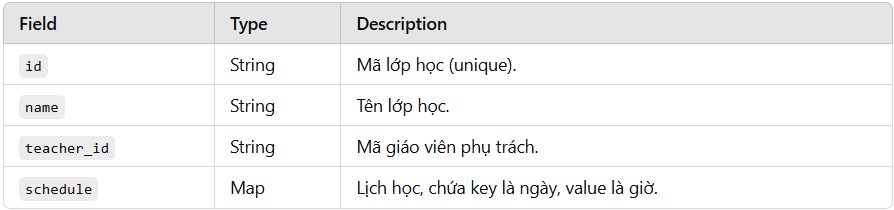
Bảng sinh viên: Thông tin sinh viên



***Bảng 3. 9 Mô tả bảng dữ liệu sinh viên***

### ****3.6.3.Bảng lớp****

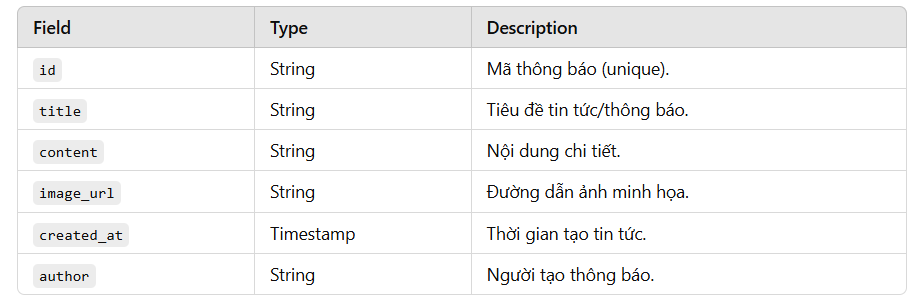
Thông tin lớp, danh sách sinh viên.



***Bảng 3. 10 Mô tả bảng dữ liệu lớp***

### ****3.6.5.Bảng tin tức****

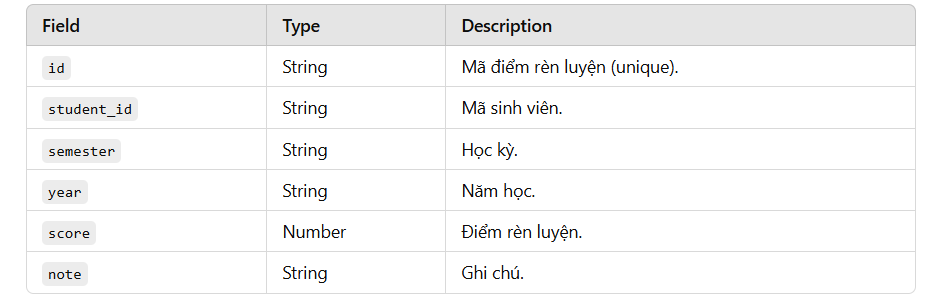
Thông tin về tin tức.



***Bảng 3. 12 Mô tả bảng dữ liệu tin tức***

### ****3.6.6.Bảng điểm rèn luyện****

Thông tin điểm rèn luyện.



***Bảng 3. 13 Mô tả bảng dữ liệu điểm rèn luyện***

**CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG**

## 4.1.Tạo dự án Flutter

Đầu tiên, tạo dự án Flutter bằng Android Studio

A screenshot of a computer program

Description automatically generated

Hình 4.1. Tạo project Flutter

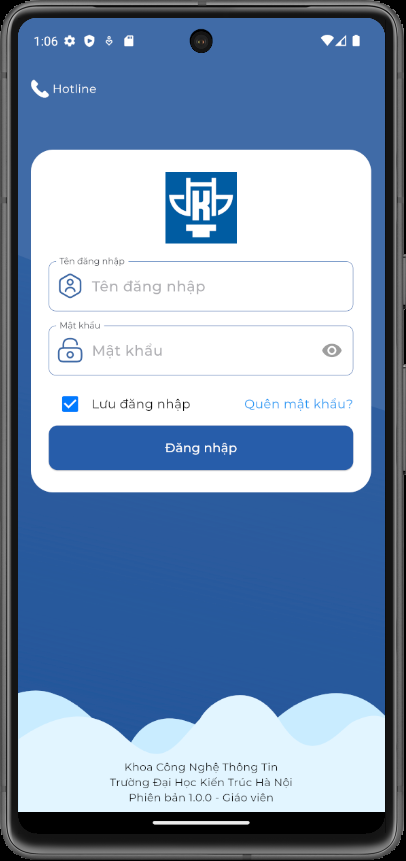
A screenshot of a computer

Description automatically generated

Hình 4.2. Cấu trúc thư mục project Flutter

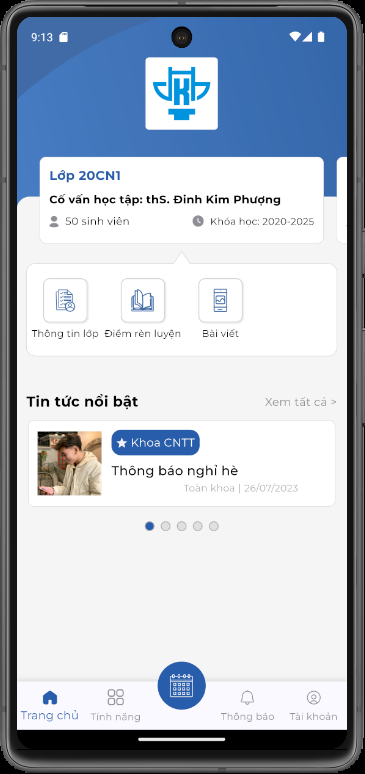
## 4.2. Một số hình ảnh giao diện

### 4.2.1. Giao diện login



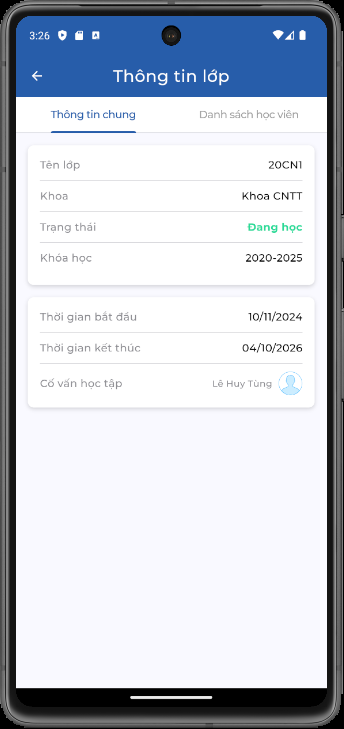
Hình 4.3. Giao diện màn login

### 4.2.2. Giao diện màn home



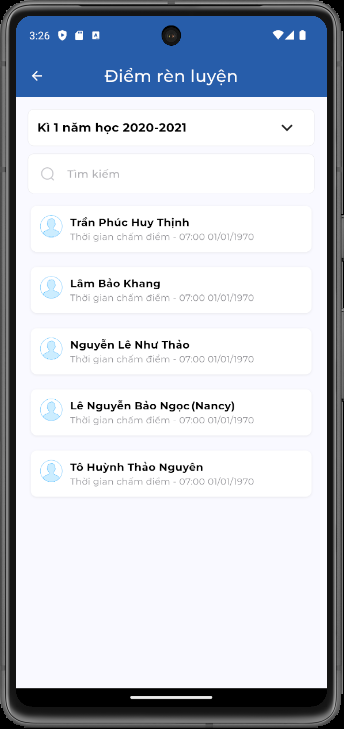
Hình 4.4. Giao diện màn home

### 4.2.3. Giao diện màn thông tin lớp



Hình 4.5. Giao diện màn thông tin lớp

### 4.2.4. Giao diện màn phiếu điểm rèn luyện



Hình 4.6. Giao diện màn phiếu điểm rèn luyện

### 4.2.5. Giao diện màn bài viết

******

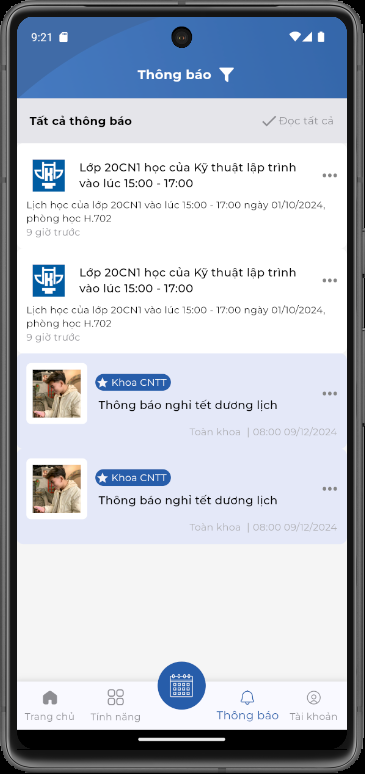
Hình 4.7. Giao diện màn bài viết

### 4.2.6. Giao diện màn thêm bài viết



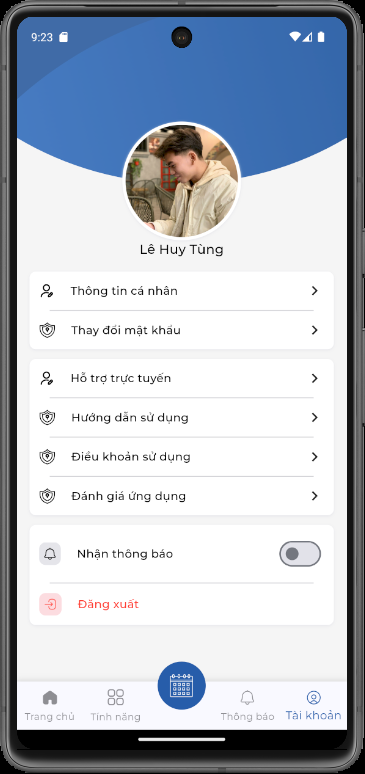
Hình 4.8. Giao diện màn thêm bài viết

### 4.2.7. Giao diện màn thông báo



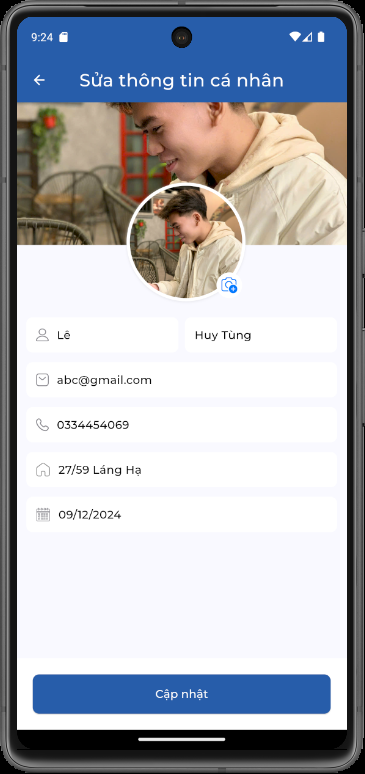
Hình 4.9. Giao diện màn thông báo

### 4.2.8. Giao diện màn tài khoản



Hình 4.10. Giao diện màn tài khoản

### 4.2.9. Giao diện màn sửa thông tin cá nhân



Hình 4.11. Giao diện màn sửa thông tin cá nhân

### 4.2.10. Giao diện màn đổi mật khẩu



Hình 4.12. Giao diện màn đổi mật khẩu

# DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Flutter, "https://docs.flutter.dev/," [Online]. Available: https://docs.flutter.dev/. |
| [2] | T. N. T. Thủy, "Lập Trình Android Là Gì? 6 Điều Cần Biết Về Ngôn Ngữ Lập Trình Android," 23 08 2022. [Online]. Available: https://glints.com/vn/blog/lap-trinh-android-la-gi/#.Y72n9f5ByHs. |
| [3] | Admin, "https://codefresher.vn/top-6-ngon-ngu-lap-trinh-mobile-dang-hoc-nhat-nam-2022/," codefresher.vn, 23 02 2022. [Online]. Available: https://codefresher.vn/top-6-ngon-ngu-lap-trinh-mobile-dang-hoc-nhat-nam-2022/. [Accessed 23 02 2022]. |
| [4] | V. Trần, "Flutter là gì? Nó có ưu điểm vượt trội ra sao để làm một ứng dụng mobile?," topdev.vn, [Online]. Available: https://topdev.vn/blog/flutter-la-gi/. |
| [5] | TuanNM, "Giới thiệu ngôn ngữ Dart," viblo.asia, 1 05 2021. [Online]. Available: https://viblo.asia/p/gioi-thieu-ngon-ngu-dart-ORNZqdv3K0n. |
| [6] | N. T. Tuấn, "[Flutter] Tìm hiểu về state management với Provider," viblo.asia, 19 03 2020. [Online]. Available: https://viblo.asia/p/flutter-tim-hieu-ve-state-management-voi-provider-LzD5d1wOKjY. |
| [7] | C. N. Thành, "Tìm hiểu các mô hình Kiến trúc trong phát triển ứng dụng trong android.," 29 11 2015. [Online]. Available: https://viblo.asia/p/tim-hieu-cac-mo-hinh-kien-truc-trong-phat-trien-ung-dung-trong-android-5WQvzW9LRk3E. |
| [8] | S. Siripathi, "Các ngôn ngữ lập trình ứng dụng di động," 25 04 2022. [Online]. Available: https://code.tutsplus.com/vi/articles/mobile-development-languages--cms-29138. |