TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÁO CÁO BÀI TẬP LỚN**

XÂY DỰNG ỨNG DỤNG MẠNG XÃ HỘI ĐA NỀN TẢNG

Môn học: Chuyên đề

Giảng viên hướng dẫn: TS. Nguyễn Nhất Hải

Nhóm 07:

|  |  |
| --- | --- |
| Nguyễn Bắc Việt | 20173464 |
| Nguyễn Kiên Trung | 20173421 |
| Nguyễn Khương Duy | 20173072 |
| Nguyễn Quang Tạo | 20173356 |
| Nguyễn Minh Tuấn | 20173436 |

*Hà Nội, tháng* *5 năm 2021*

**Mục lục**

[Lời nói đầu 3](#_Toc73054770)

[1. Giới thiệu 4](#_Toc73054771)

[2. Công nghệ sử dụng 4](#_Toc73054772)

[2.1 Công nghệ 4](#_Toc73054773)

[2.2 Flutter 4](#_Toc73054774)

[2.3 Firebase 5](#_Toc73054775)

[3. Mô hình phát triển ứng dụng 5](#_Toc73054776)

[4. Quy trình phát triển 6](#_Toc73054777)

[4.1 Mô hình thác nước 6](#_Toc73054778)

[4.2 Công cụ quản lý tiến độ 6](#_Toc73054779)

[4.3 Công cụ quản lý mã nguồn 6](#_Toc73054780)

[5. Phân tích thiết kế hệ thống 6](#_Toc73054781)

[5.1 Biểu đồ usecase tổng quan 6](#_Toc73054782)

[5.2 Phân tích thiết kế chi tiết 7](#_Toc73054783)

[6. Cơ sở dữ liệu 15](#_Toc73054784)

[a) Thiết kế colection User 15](#_Toc73054785)

[b) Thiết kế colection Post 16](#_Toc73054786)

[c) Thiết kế colection Conservation 16](#_Toc73054787)

[d) Thiết kế colection chat 16](#_Toc73054788)

[e) Thiết kế colection friend 17](#_Toc73054789)

[7. Tài liệu tham khảo 17](#_Toc73054790)

[8. Phân chia công việc 17](#_Toc73054791)

# Lời nói đầu

Cuộc cách mạng công nghiệp 4.0 đang phát triển mạnh mẽ, chúng ta cần áp dụng công nghệ thông tin vào trong công việc ngày càng nhiều hơn để tiết kiệm thời gian và chi phí. Việc sử dụng app trên thiết bị di động trở nên tiện lợi và rất hiệu quả, phát triển mạnh mẽ trong những thập niên gần đây. Với đất nước đang phát triển như Việt Nam, sử dụng những công nghệ giúp con người có thể giao tiếp và chia sẻ thông tin với nhau, thuận tiện cho làm việc, trao đổi.

Nhóm em đã quyết định chọn đề tài xây dựng Ứng dụng mạng xã hội đa nền tảng. Qua dự án ngoài việc bổ sung vào vốn kỹ năng phát triển ứng dụng di động, nhóm em còn biết cách hợp tác và làm việc nhóm quy trình, kỷ luật và hiệu quả. Qua đó có thể áp dụng kiến thức bộ môn vào thực tế, đúng với tiêu chí môn học thực hành.

Để có thể hoàn thành tốt bài tập lớn này, nhóm em xin gửi lời cảm ơn chân thành đến thầy giáo hướng dẫn – **Tiến sĩ Nguyễn Nhất Hải**, Giảng viên Trường Đại học Bách Khoa Hà Nội, đã giúp đỡ, hướng dẫn tận tình để nhóm em hoàn thành đề tài này với những mục tiêu đã đặt ra.

*Chúng em xin chân thành cảm ơn!*

# Giới thiệu

Để sử dụng ứng dụng, người dùng trước hết cần có tài khoản. Sau đó, người dùng có thể sử dụng những chức năng sau:

* Đăng ký, đăng nhập, đăng xuất
* Đăng bài viết (kèm icon), ảnh, video
* Xem bài viết
* Xem các bài viết video
* Chỉnh sửa bài viết
* Like, bình luận bài viết
* Tìm kiếm người dùng và kết bạn.
* Chỉnh sửa thông tin cá nhân
* Thông báo khi có người kết bạn, like, bình luận
* Nhắn tin

Yêu cầu phi chức năng đối với ứng dụng: Xử lý dữ liệu tốt, thời gian thực; thông tin hiển thị phải hữu ích với người dùng; giao diện phải mang tính dễ dùng.

# Công nghệ sử dụng

## 2.1 Công nghệ

Để tạo lên một ứng dụng di động có giao diện đẹp, dễ sử dụng, thân thiện với người dùng, đồng thời có thể chạy trên nền tảng Android và iOS, nhóm em chọn framework **Flutter**.

Về Backend, ứng dụng có những chức năng yêu cầu xử lý thời gian thực. Vì vậy, nhóm em sử dụng **Firebase** xử lý dữ liệu một cách hiệu quả nhất.

## 2.2 Flutter

Flutter là UI Framework mã nguồn mở dành cho di động, miễn phí được tạo bởi Google và phát hành vào tháng 5 năm 2017.

Nói đơn giản, Flutter nó cho phép tạo một ứng dụng di động gốc (native app) với một codebase. Điều này có nghĩa là lập trình viên có thể sử dụng một ngôn ngữ lập trình và một codebase để tạo hai ứng dụng khác nhau (cho iOS và Android).

Flutter gồm 2 thành phần quan trọng:

* Một SDK (Software Development Kit): Một bộ sưu tập các công cụ sẽ giúp phát triển các ứng dụng của mình. Điều này bao gồm các công cụ để biên dịch mã nguồn thành mã máy gốc (mã cho iOS và Android).
* Một Framework (UI Library based on widgets): Một tập hợp các thành phần UI có thể tái sử dụng (button, text inputs, slider, ...) giúp người lập trình có thể cá nhân hóa tùy theo nhu cầu của riêng mình.

Để lập trình với Flutter, nhóm em sử dụng một ngôn ngữ lập trình gọi là Dart. Dart tập trung vào phát triển front end và có thể sử dụng nó để tạo ứng dụng di động và cả ứng dụng web. Ngoài ra, Dart cũng là ngôn ngữ lập trình hướng đối tượng.

## 2.3 Firebase

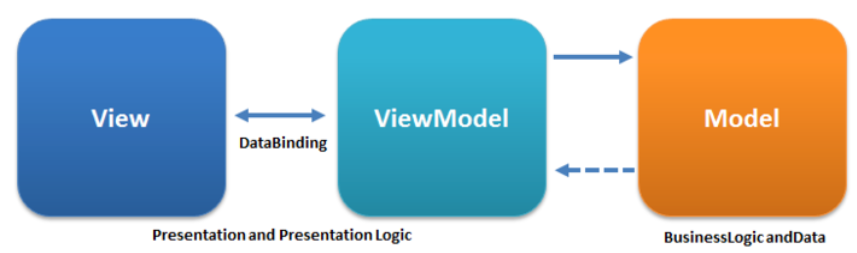
Firebase là một nền tảng để phát triển ứng dụng di động và trang web, bao gồm các API đơn giản và mạnh mẽ. Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google.

Firebase giúp các lập trình viên rút ngắn thời gian triển khai và mở rộng quy mô của ứng dụng mà họ đang phát triển.

# Mô hình phát triển ứng dụng

Để xây dựng một ứng dụng di động, có thể mở rộng trong tương lai, nhóm em quyết định sử dụng mô hình MVVM (Model – View – ViewModel). Cụ thể mô hình như sau:

* **View**: là phần giao diện của ứng dụng để hiển thị dữ liệu và nhận tương tác của người dùng. Một điểm khác biệt so với các ứng dụng truyền thống là View trong mô hình này tích cực hơn. Nó có khả năng thực hiện các hành vi và phản hồi lại người dùng thông qua tính năng binding, command.
* **Model**: là các đối tượng giúp truy xuất và thao tác trên dữ liệu thực sự.
* **ViewModel**: Lớp trung gian giữa View và Model. ViewModel có thể được xem là thành phần thay thế cho Controller trong mô hình MVC. Nó chứa các mã lệnh cần thiết để thực hiện data binding, command.



*Mô hình MVVM*

# Quy trình phát triển

## Mô hình thác nước

Nhóm sử dụng mô hình phát triển phần mềm thác nước, phù hợp với ứng dụng nhỏ và vừa, ít thành viên phát triển.

## Công cụ quản lý tiến độ

Nhóm sử dụng một file excel chung, phân chia công việc và check deadline

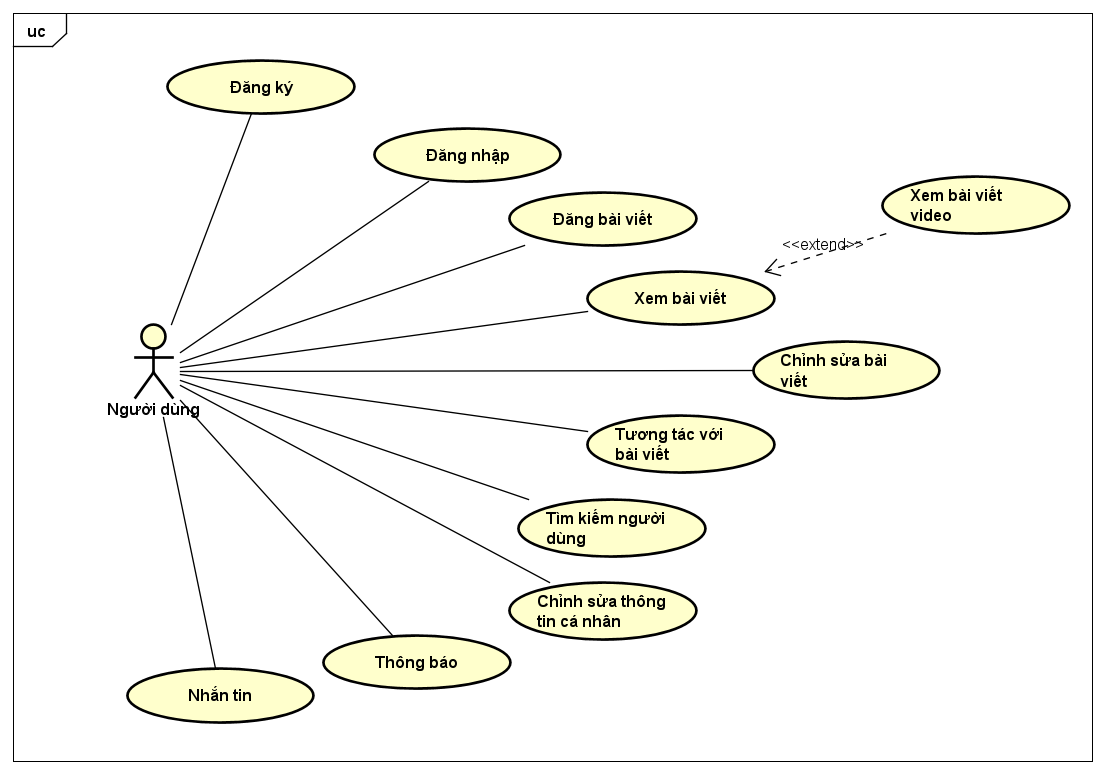
## Công cụ quản lý mã nguồn

Github là một công cụ quản lý mã nguồn phổ biến và nhóm em đã lựa chọn để sử dụng

# Phân tích thiết kế hệ thống

## Biểu đồ usecase tổng quan

Hệ thống chỉ có một tác nhân chính đó là Người dùng. Người dùng có thể sử dụng các chức năng của hệ thống nhưng cần phải đăng ký, đăng nhập.



*Biểu đồ usecase Tổng quan*

## Phân tích thiết kế chi tiết

**Usecase đăng ký:**

* Tóm tắt: usecase này dùng đăng ký tài khoản.
* Dòng sự kiện:

Dòng sự kiện chính: Use case này bắt đầu khi người dùng nhấn vào nút đăng ký

1. Người dùng điền các thông tin theo thứ tự màn hình
2. Kiểm tra điều kiện nhập không trống, email đúng định dạng, password đủ độ dài hợp lệ
3. Kiểm tra Email đã được đăng ký trên hệ thống hay chưa
4. Tạo mới tài khoản trên hệ thống
5. Thông báo tạo tài khoản thành công

* Điều kiện bắt buộc: Email và Password không được để trống, Email đúng định dạng

**Usecase đăng nhập:**

* Tóm tắt: usecase này dùng đăng ký tài khoản.
* Dòng sự kiện:

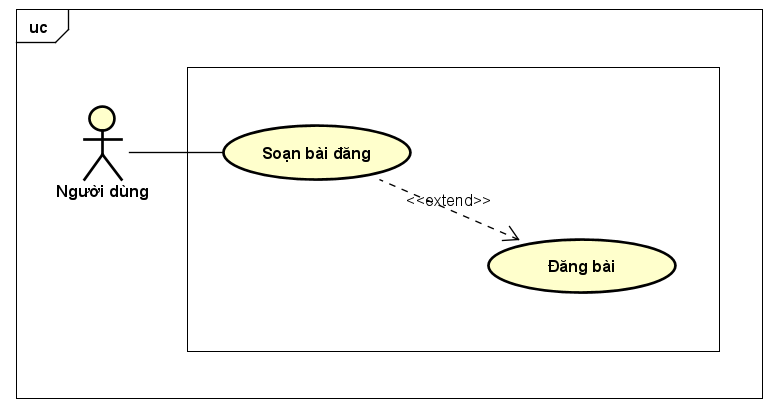
Dòng sự kiện chính: Use case này bắt đầu khi người dùng nhấn vào nút đăng nhập

1. Người dùng điền Email và Password
2. Kiểm tra điều kiện nhập không trống, email đúng định dạng, password đủ độ dài hợp lệ
3. Kiểm tra đúng Email hoặc Password hay chưa
4. Thông báo đăng nhập thành công

* Điều kiện bắt buộc: Email và Password không được để trống, Email đúng định dạng

**Usecase đăng bài viết:**

* Tóm tắt: use case dùng để người dụng có thể soạn và đăng bài viết



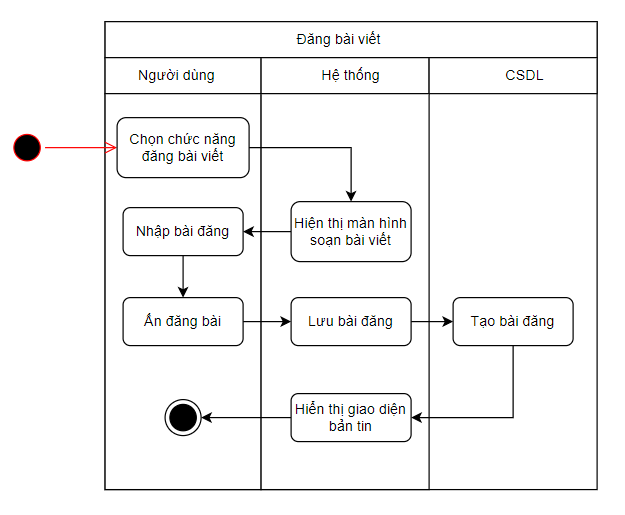
*Usecase phân rã chức năng Đăng bài viết*

* Dòng sự kiện:

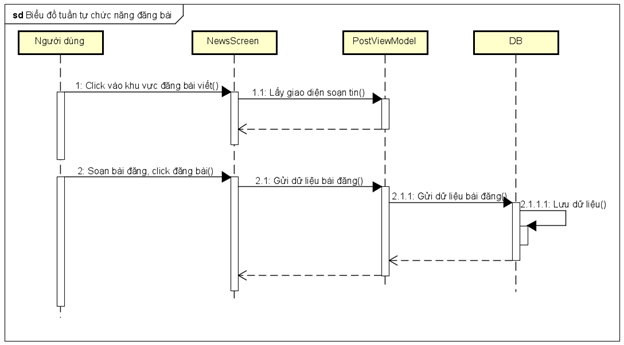
Dòng sự kiện chính: Use case này bắt đầu khi người dùng nhấn khu vực đăng bài

|  |  |
| --- | --- |
| Soạn bài viết | 1. Người dùng nhấn vào khu vực đăng bài 2. Hệ thống hiển thị giao diện đăng bài 3. Người dùng nhập nội dung bài đăng |
| Đăng bài viết | 1. Người dùng nhấn vào icon đăng bài 2. Chuyển đến màn hình bản tin, hiện thị bài vừa đăng |

* Điều kiện bắt buộc: Đã đăng nhập



*Biểu đồ hoạt động chức năng Đăng bài viết*



*Biểu đồ tuần tự chức năng Đăng bài viết*

**Usecase Nhắn tin:**

* Tóm tắt: usecase này dùng để mô tả chức năng Nhắn tin.

Diagram

Description automatically generated

*Usecase phân rã chức năng Nhắn tin*

* Dòng sự kiện:

Dòng sự kiện chính: Use case này bắt đầu khi người dùng nhấn vào một người dùng ở màn hình chat

|  |  |
| --- | --- |
| Hiển thị tin nhắn đã gửi | 1. Hệ thống hiển thị tất cả tin nhắn cho người dùng |
| Gửi tin nhắn | 1. Người dùng nhập nội dung tin nhắn 2. Kiểm tra tin nhắn trống 3. Hệ thống thêm tin nhắn mới |
| Nhận tin nhắn | 1. Hiển thị tin nhắn mới cho cả bên người gửi và người nhận |

* Điều kiện bắt buộc: Nội dung tin nhắn không được để trống

Diagram

Description automatically generated

*Biểu đồ tuần tự chức năng Nhắn tin-tin nhắn văn bản*

*Diagram

Description automatically generated*

*Biểu đồ tuần tự chức năng Nhắn tin-tin nhắn hình ảnh*

**Usecase tìm kiếm người dùng:**

* Tóm tắt: usecase này dùng để tìm kiếm người dùng
* Dòng sự kiện:

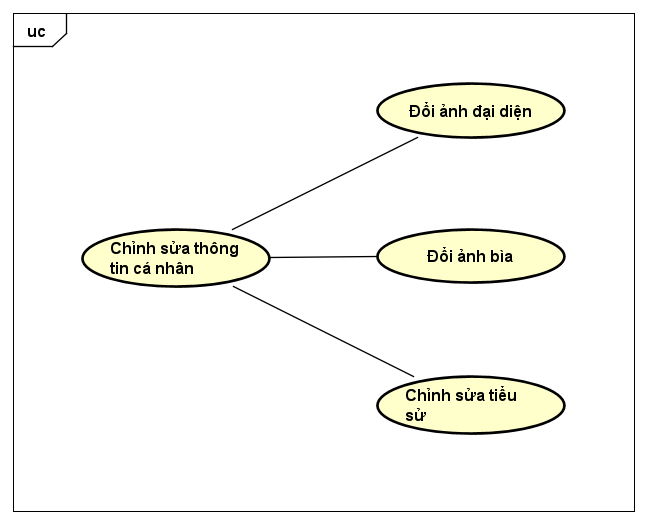
Dòng sự kiện chính: Use case này bắt đầu khi người dùng nhấn vào thanh tìm kiếm

1. Người dùng nhập tên người dùng cần tìm kiếm
2. Trả về danh sách người dùng

* Điều kiện bắt buộc: không

**Usecase chỉnh sửa thông tin cá nhân:**

* Tóm tắt: use case liên quan đến các chức năng chỉnh sửa thông tin cá nhân của người dùng: thay đổi ảnh đại diện, thay đổi ảnh bìa, chỉnh sửa tiểu sử



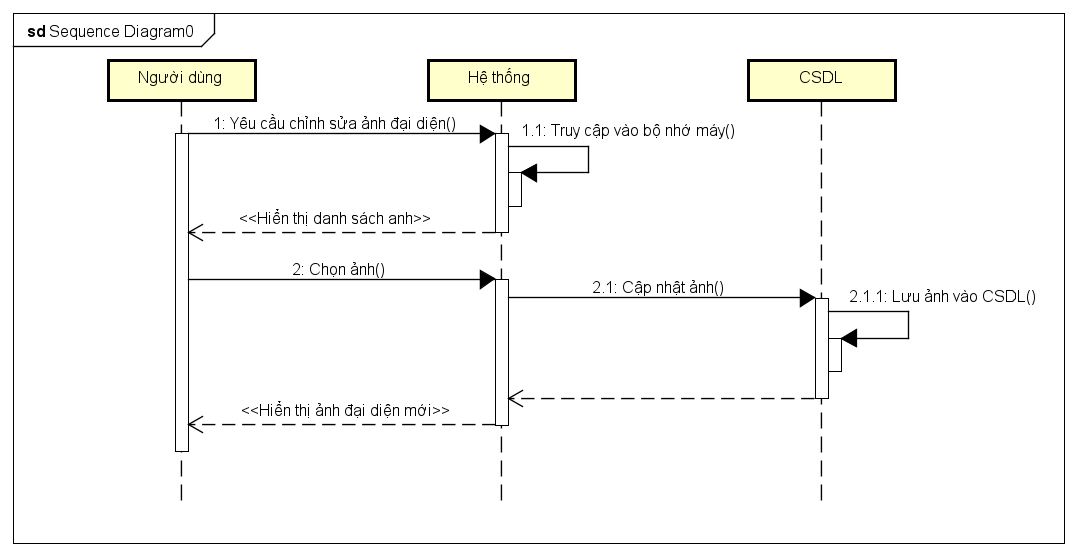
*Usecase phân rã chức năng Chỉnh sửa thông tin cá nhân*

* Dòng sự kiện:

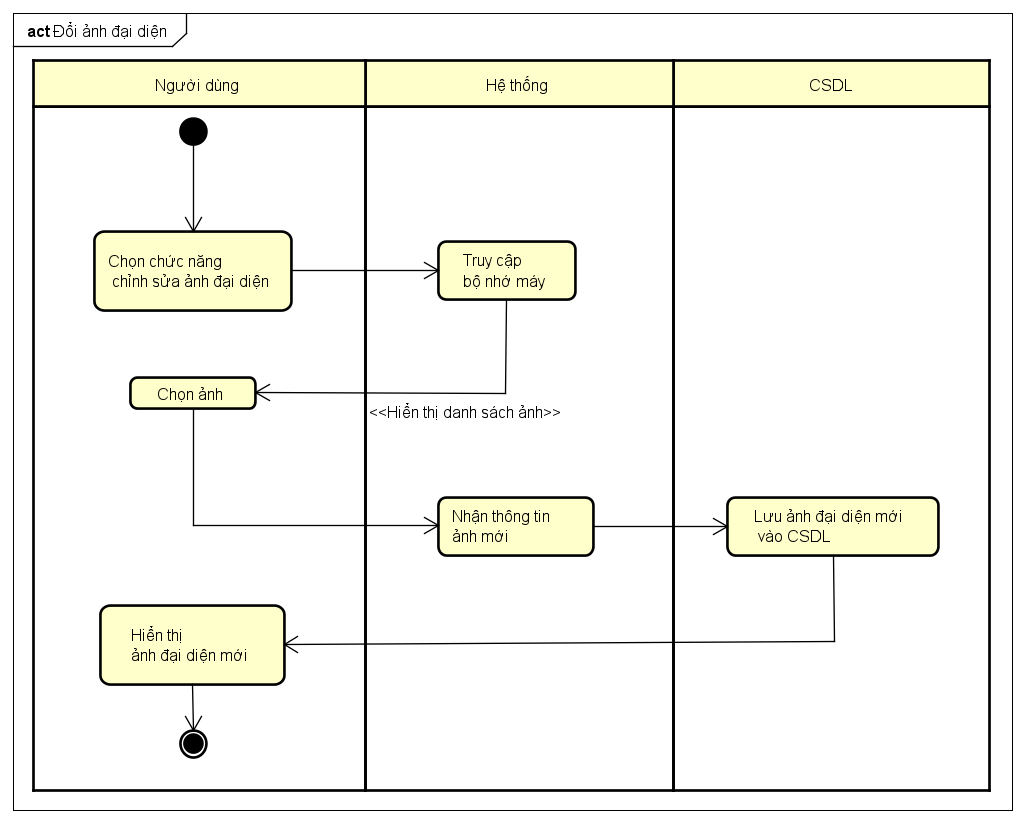
Dòng sự kiện chính: Use case này bắt đầu khi người dùng nhấn vào chỉnh sửa thông tin cá nhân

|  |  |
| --- | --- |
| Thay đổi ảnh đại diện | 1. Người dùng nhấn vào chỉnh sửa ảnh đại diện 2. Người dùng chọn ảnh từ bộ nhớ máy 3. Người dùng chọn ảnh 4. Hệ thống đổi ảnh đại diện cho người dùng 5. Hiển thị ảnh đại diện đã được cập nhật |
| Thay đổi ảnh bìa | 1. Người dùng nhấn vào chỉnh sửa ảnh bìa 2. Người dùng chọn ảnh từ bộ nhớ máy 3. Người dùng chọn ảnh 4. Hệ thống đổi ảnh bìa cho người dùng 5. Hiển thị ảnh bìa đã được cập nhật |
| Chỉnh sửa tiểu sử | 1. Người dùng vào mục thay đổi thông tin cá nhân 2. Người dùng điền hoặc thay đổi tiểu sử 3. Nhấn chỉnh sửa 4. Hệ thống cập nhật tiểu sử người dùng |

* Điều kiện bắt buộc: Không



*Biểu đồ tuần tự chức năng Đổi ảnh đại diện*



*Biểu đồ hoạt động chức năng Đổi ảnh đại diện*

**Usecase thông báo:**

* Tóm tắt: usecase này dùng để thông báo khi có người khác tương tác với người dùng hiện tại (kết bạn, like hoặc bình luận bài viết)
* Dòng sự kiện:

Dòng sự kiện chính: Use case này bắt đầu khi người dùng khác tương tác với người dùng hiện tại

|  |  |
| --- | --- |
| Thông báo | 1. Người dùng khác tương tác với người dùng hiện tại (kết bạn, like hoặc bình luận bài viết) 2. Hệ thống nhận biết thay đổi, tạo thông báo đến người dùng 3. Người nhận nhận được thông báo |

* Điều kiện bắt buộc: không

# Cơ sở dữ liệu

Dữ liệu sẽ được lưu ở firebase cloud firestore dưới dạng nosql(json), hình ảnh và video sẽ được lưu ở firebase cloud storage.

## Thiết kế colection User

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Giải thích** |
| User\_Id | int | 11 | ID người dùng |
| Email | String | 250 | Email người dùng |
| Password | String | 250 | Mật khẩu |
| First\_Name | String | 250 | Họ |
| Last\_Name | String | 250 | Tên |
| Phone | String | 12 | Số điện thoại |
| City | String | 250 | Địa chỉ |
| Description | String | 250 | Mô tả, tiểu sử |
| Genre | String | 8 | Giới tính |
| Avatar | String | 250 | Đường dẫn ảnh đại diện |
| Corver\_Image | String | 250 | Đường dẫn ảnh bìa |
| Birthday | TimeStamp |  | Ngày sinh |

## Thiết kế colection Post

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Giải thích** |
| Post\_ID | int | 11 | ID bài đăng |
| Described | String | 1024 | Nội dung bài đăng |
| Owner | Reference |  | Tham chiếu user là người đăng |
| Modified | String | 250 | Thời gian cập nhật |
| Created | String | 250 | Thời gian đăng |
| Likes | array |  | Danh sách tham chiếu đến người like bài đăng |
| Comment | array | 250 | Danh sách comment |
| Images | array | 250 | Danh sách đường dẫn ảnh của bài đăng |
| Video | String | 250 | Đường dẫn video |

## Thiết kế colection Conservation

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Giải thích** |
| Conservation\_ID | int | 11 | ID hội thoại |
| users | array |  | Danh sách id của user trong hội thoại |
| Current\_Message | Reference |  | Tham chiếu message hiện tại |

## Thiết kế colection Chat

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Giải thích** |
| Chat\_id | int | 11 | ID chat |
| Message | String | 1024 | Nội dung tin nhắn(text hoặc link ảnh, video) |
| From | Reference |  | Tham chiếu user là người gửi |
| To | Reference |  | Tham chiếu user là người nhận |
| Send\_Time | String | 250 | Thời gian gửi |
| Type | int |  | Loại message (text, đa phương tiện) |

## Thiết kế colection friend

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Kích thước** | **Giải thích** |
| Friend\_id | int | 11 | ID friend |
| Created | String | 250 | Thời gian gửi lời mời |
| Modified | String | 250 | Thời gian cập nhật |
| First | Reference |  | Tham chiếu user thứ nhất |
| Second | Reference |  | Tham chiếu user thứ nhất |
| Stattus | int |  | Trạng thái(None, pending, accept) |

# Tài liệu tham khảo

https://niithanoi.edu.vn/flutter-la-gi.html

https://wiki.matbao.net/firebase-la-gi-giai-phap-lap-trinh-khong-can-backend-tu-google

<https://viblo.asia/p/co-ban-ve-mvvm-model-view-viewmodel-pattern-Do754wdWlM6>

# Phân chia công việc

|  |  |
| --- | --- |
| Nguyễn Bắc Việt | Backend, Logic app, Tạo Base |
| Nguyễn Kiên Trung | Frontend, UI Trang cá nhân |
| Nguyễn Khương Duy | Frontend, UI Trang chủ, Bài viết và tương tác |
| Nguyễn Quang Tạo | Frontend, Chat Messenger, Tìm kiếm |
| Nguyễn Minh Tuấn | Frontend, UI Thông báo, Tìm kiếm |