

**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
**BỘ MÔN TIN HỌC ỨNG DỤNG**



**Lớp DI17Y1A1 – Ngành Công nghệ thông tin**  
**Luận văn tốt nghiệp – TN418**

**ĐỀ TÀI**  
**XÂY DỰNG ỨNG DỤNG MẠNG XÃ HỘI**  
**CHO SINH VIÊN ĐẠI HỌC CẦN THƠ BẰNG FLUTTER**

**GIÁO VIÊN HƯỚNG DẪN**  
ThS. Hoàng Minh Trí

**SINH VIÊN THỰC HIỆN**  
Huỳnh Hữu Lộc  
MSSV: B1706719

Cần Thơ, 2022

**XÁC NHẬN CHỈNH SỬA LUẬN VĂN**  
**THEO YÊU CẦU CỦA HỘI ĐỒNG**

Tên luận văn (tiếng Việt và tiếng Anh): Xây dựng ứng dụng mạng xã hội cho sinh viên Đại học Cần Thơ bằng Flutter – Building a social network application for Can Tho University students with Flutter.

Họ tên sinh viên: Huỳnh Hữu Lộc

MASV: B1706719

Mã lớp: DI17Y1A1

Đã báo cáo tại hội đồng ngành: Tin Học Ứng Dụng

Ngày báo cáo: 20/05/2022

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

*Cần Thơ, ngày ... tháng ... năm 2022*

**Giáo viên hướng dẫn**

*(Ký và ghi họ tên)*

# LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến gia đình đã động viên em trong suốt quá trình học tập.

Em xin cảm ơn Bộ môn Tin học ứng dụng, khoa Công nghệ thông tin và Truyền thông, Trường Đại học Cần Thơ đã tạo điều kiện tốt nhất để em thực hiện đề tài này.

Dưới sự hướng dẫn của ThS. Hoàng Minh Trí, em đã hoàn thành đề tài “**Xây dựng ứng dụng mạng xã hội cho sinh viên Đại học Cần Thơ bằng Flutter**”. Em xin bày tỏ lời cảm ơn sâu sắc đến thầy vì đã tận tình giúp đỡ, định hướng cho em trong suốt thời gian qua.

Trong quá trình thực hiện, em đã tích lũy được thêm nhiều kiến thức, giúp bản thân em dần hoàn thiện hơn. Đồng thời, kết hợp với kiến thức nền tảng đã có trong thời gian học tập những năm qua, em đã hoàn thành đề tài này.

Em rất mong nhận được sự góp ý, nhận xét từ thầy để đề tài được hoàn thiện hơn.

Cuối lời, em xin chúc thầy dồi dào sức khỏe, hạnh phúc và đạt được nhiều thành công trong công việc và cuộc sống.

Trân trọng!

*Cần Thơ, ngày ... tháng ... năm 2022*

**Sinh viên thực hiện**

**Huỳnh Hữu Lộc**

[illegible]

## Cán bộ hướng dẫn

## MỤC LỤC

<b>MỤC LỤC .....</b>	<b>i</b>
<b>DANH MỤC VIẾT TẮT.....</b>	<b>iii</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>iv</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>v</b>
<b>TÓM TẮT .....</b>	<b>viii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>CHƯƠNG 1. GIỚI THIỆU.....</b>	<b>1</b>
1.1 Đặt vấn đề .....	1
1.2 Các nghiên cứu liên quan.....	2
1.2.1 Mạng xã hội tiêu biểu .....	2
1.2.2 Facebook.....	2
1.2.3 Instagram .....	3
1.2.4 Zalo .....	4
1.2.5 Ứng dụng mạng xã hội CTUPG được thực hiện trong đề tài .....	6
1.3 Mục tiêu của đề tài .....	7
1.4 Đối tượng và phạm vi nghiên cứu .....	8
1.5 Phương pháp nghiên cứu .....	9
1.5.1 Về lý thuyết.....	9
1.5.2 Về kỹ thuật.....	9
1.5.3 Về nguồn tài liệu.....	9
1.6 Bố cục luận văn.....	10
1.7 Kế hoạch làm việc.....	11
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>14</b>
2.1 Giới thiệu .....	14
2.1.1 Tổng quan về mạng xã hội.....	14
2.1.1.1 Khái niệm .....	14
2.1.1.2 Đặc điểm của mạng xã hội .....	14
2.1.1.3 Vai trò của mạng xã hội.....	14
2.1.1.4 Vai trò của mạng xã hội trong học tập.....	15
2.2 Ngôn ngữ và công nghệ sử dụng .....	16
2.2.1 Ngôn ngữ Dart .....	16

2.2.2 Widgets .....	16
2.2.3 SDK Flutter .....	18
2.2.4 Layers.....	18
2.2.5 Firebase (Server) .....	19
2.2.6 Mô hình thiết kế Model View Controller (MVC) .....	20
2.2.6.1 Model View Controller (MVC) là gì? .....	20
2.2.6.2 Đặc điểm của Model View Controller (MVC).....	21
2.2.6.3 Lợi điểm của Model View Controller (MVC).....	21
2.2.7 Các Widgets quan trọng.....	23
2.2.7.1 Gestures .....	23
2.2.7.2 State .....	23
<b>CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN.....</b>	<b>26</b>
3.1 Giải pháp phân tích và thiết kế các mô hình.....	26
3.1.1 Mô hình Use Case.....	26
3.1.1.1 Danh sách Actor .....	26
3.1.1.2 User.....	26
3.1.1.3 Admin .....	31
3.1.2 Sơ đồ ER .....	34
3.1.3 Sơ đồ BFD .....	35
3.1.4 Sơ đồ tuần tự.....	38
3.2 Thiết kế cơ sở dữ liệu.....	55
3.3 Cấu trúc và thể hiện của mô hình MVC .....	64
3.3.1 Cấu trúc mô hình MVC trong ứng dụng.....	64
3.3.1.1 Cấu trúc thư mục: .....	64
3.3.2 Thể hiện của mô hình MVC trong ứng dụng.....	65
3.3.2.1 Khởi tạo Models .....	65
3.3.2.2 Khởi tạo Views .....	67
3.3.2.3 Khởi tạo Controller.....	68
<b>CHƯƠNG 4. KẾT QUẢ XÂY DỰNG ỨNG DỤNG.....</b>	<b>71</b>
4.1 Kết quả đạt được .....	71
4.2 Hướng phát triển .....	83
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>84</b>

## DANH MỤC VIẾT TẮT

Ký hiệu	Diễn giải
API	Application Program Interface
JSON	JavaScript Object Notation
MVC	Model-View-Controller
SDK	Software Development Kit
CTUPG	Can Tho University Playground
ER	Entity Relationship
BFD	Business Function Diagram
iOS	iPhone Operating System

## DANH MỤC BẢNG BIỂU

Bảng 2.1 Bảng danh sách Actor .....	26
Bảng 2.2 Bảng danh sách Use-Case của User .....	29
Bảng 2.3 Bảng danh sách User-Case của Admin .....	32
Bảng 2.4 Thực thể users collection.....	56
Bảng 2.5 Thực thể feeds collection .....	57
Bảng 2.6 Thực thể trivias collection.....	57
Bảng 2.7 Thực thể questionDetail sub-collection.....	58
Bảng 2.8 Thực thể posts collection.....	58
Bảng 2.9 Thực thể qaGames collection .....	59
Bảng 2.10 Thực thể tasks collection.....	60
Bảng 2.11 Thực thể forums collection.....	60
Bảng 2.12 Thực thể reports collection.....	61
Bảng 2.13 Thực thể chats collection.....	61
Bảng 2.14 Thực thể conversation sub-collection .....	62
Bảng 2.15 Thực thể comments/forumComments collection .....	63
Bảng 2.16 Thực thể followers/ followings collection .....	63



## DANH MỤC HÌNH ẢNH

Hình 1.1 Giao diện mạng xã hội Facebook .....	2
Hình 1.2 Giao diện mạng xã hội Instagram .....	4
Hình 1.3 Giao diện mạng xã hội Zalo.....	5
Hình 1.4 Tóm tắt các giai đoạn.....	11
Hình 1.5 Giai đoạn 1 của kế hoạch (1) .....	11
Hình 1.6 Giai đoạn 1 của kế hoạch (2) .....	12
Hình 1.7 Giai đoạn 2 của kế hoạch.....	12
Hình 1.8 Giai đoạn 3.....	13
Hình 2.1 Cấu trúc Widget.....	17
Hình 2.4 Tổng quan về layer của Flutter .....	19
Hình 2.2 Mô hình vòng đời của State Object .....	24
Hình 2.3 Mô hình hoạt động của StatelessWidget .....	25
Hình 2.10 Mô hình Use Case của User mức tổng quát .....	27
Hình 2.11 Mô hình Use-Case của User mức phân rã .....	30
Hình 2.12 Mô hình Use-Case của Admin mức tổng quát.....	31
Hình 2.13 Mô hình Use Case của Admin mức phân rã.....	33
Hình 2.14 Hình sơ đồ ER.....	34
Hình 2.15 Sơ đồ BFD của người dùng thường.....	35
Hình 2.16 Sơ đồ BFD của người dùng ẩn danh.....	36
Hình 2.17 Sơ đồ BFD của admin.....	37
Hình 2.18 Sơ đồ tuần tự đăng nhập .....	38
Hình 2.19 Sơ đồ tuần tự đăng ký .....	39
Hình 2.20 Sơ đồ tuần tự thêm bài viết.....	40
Hình 2.21 Sơ đồ tuần tự xóa bài viết .....	41
Hình 2.22 Sơ đồ tuần tự tạo phòng chat .....	42
Hình 2.23 Sơ đồ tuần tự xóa phòng chat .....	43

Hình 2.24 Sơ đồ tuần tự nhấn tin .....	44
Hình 2.25 Sơ đồ tuần tự xóa tin nhắn .....	45
Hình 2.26 Sơ đồ tuần tự chỉnh sửa profile .....	46
Hình 2.27 Sơ đồ tuần tự đổi mật khẩu .....	47
Hình 2.28 Sơ đồ tuần tự đăng xuất .....	48
Hình 2.29 Sơ đồ tuần tự tạo diễn đàn .....	49
Hình 2.30 Sơ đồ tuần tự xóa diễn đàn .....	50
Hình 2.31 Sơ đồ tuần tự ghép cặp .....	51
Hình 2.32 Tuần tự đăng bình luận .....	52
Hình 2.33 Sơ đồ tuần tự xóa bình luận .....	53
Hình 2.34 Sơ đồ tuần tự sửa profile ẩn danh .....	54
Hình 2.35 Cấu trúc mô hình MVC của ứng dụng .....	64
Hình 2.36 Package models của ứng dụng .....	65
Hình 2.37 Class User thể hiện thuộc tính người dùng .....	66
Hình 2.38 Package views của ứng dụng .....	67
Hình 2.39 Class Home là trang chủ .....	67
Hình 2.40 Cấu trúc package controller .....	70
Hình 2.41 Giao diện đăng nhập .....	71
Hình 2.42 Giao diện đăng ký .....	72
Hình 2.43 Giao diện quên mật khẩu .....	72
Hình 2.44 Email khôi phục mật khẩu được gửi .....	73
Hình 2.45 Giao diện home/ đăng bài viết .....	74
Hình 2.46 Giao diện thông tin user .....	75
Hình 2.47 Giao diện bắt chuyện – game .....	75
Hình 2.48 Giao diện công việc .....	76
Hình 2.49 Giao diện phòng chat .....	77
Hình 2.50 Giao diện thông báo .....	78
Hình 2.51 Giao diện thông tin cá nhân .....	79

## DANH MỤC HÌNH ẢNH

---

Hình 2.52 Giao diện diễn đàn .....	80
Hình 2.53 Giao diện admin.....	82

# TÓM TẮT

Hiện nay, tình hình phát triển của các mạng xã hội cũng rất khả quan. Số lượng người truy cập và đăng ký thành viên ở các mạng xã hội này ngày càng tăng. Điển hình như một số ứng dụng mạng xã hội như: Facebook, Instagram, Twitter, Zalo... Tuy nhiên, dù đã có rất nhiều các trang và ứng dụng mạng xã hội như thế nhưng nhu cầu của người dùng vẫn rất cao và tính phân loại các mạng xã hội càng cần thiết. Đơn cử, mạng xã hội cho sinh viên đang là một chủ đề ít nổi bật, trong khi nhu cầu của sinh viên trong việc trao đổi kiến thức, tài liệu giảng dạy là rất cao. Từ cơ sở thực tế trên, tôi đề xuất đề tài “**Xây dựng ứng dụng mạng xã hội cho sinh viên Đại học Cần Thơ**” cung cấp 1 môi trường học tập đi đôi với giải trí, cho phép người dùng sử dụng như một mạng xã hội riêng cho sinh viên. Từ đó sinh viên có thể dễ dàng kết nối với nhau và trao đổi thông tin học tập một cách đơn giản và dễ dàng.

Giao diện app được xây dựng trên nền framework Flutter được phát triển và cung cấp bởi Google, cùng với đặc điểm là một bộ SDK đa nền tảng, các ứng dụng Flutter có thể hoạt động trên cả iOS và Android giúp cho vấn đề về tương thích các nền tảng trở nên dễ dàng quản lý hơn. Kèm theo đó quá trình lưu trữ dữ liệu của hệ thống được xây dựng dựa trên Firebase, là một nền tảng để phát triển ứng dụng di động và trang web, bao gồm các API đơn giản và mạnh mẽ mà không cần Backend hay Server.

Kết quả đề tài sẽ là một ứng dụng mạng xã hội đa nền tảng cho sinh viên góp phần làm sáng tỏ vai trò của các trang mạng xã hội đối với hoạt động giáo dục, học tập. Kết quả nghiên cứu cũng sẽ đưa ra một mô hình học tập thông qua internet, góp phần làm đa dạng hơn các hình thức giáo dục, đào tạo hiện nay cũng như phát huy hiệu quả của internet trong hoạt động học tập.

**Từ khóa:** SDK, iOS, API.

# ABSTRACT

Nowaday, the development of social networks is also very positive. The number of people accessing and registering members in these social networks is increasing. Typically, some social networking applications such as: Facebook, Instagram, Twitter, Zalo... However, although there are so many social networking sites and applications, the demand of users is still very high and classification of social networks is more necessary. For example, social networking for students is a less prominent topic, while the needs of students in the exchange of knowledge and teaching materials are very high. Based on the above fact, I propose the topic "Building social networking applications for students of Can Tho University" providing a learning environment coupled with entertainment, allowing users to use A private social network for students. From there, students can easily connect with each other and exchange learning information simply and easily.

The app interface is built on the Flutter framework developed and provided by Google, along with the feature of a cross-platform SDK, Flutter applications can work on both iOS and Android to help with interoperability. prefer platforms to be easier to manage. Along with that, the data storage process of the system is built on top of Firebase, which is a platform for developing mobile applications and websites, including simple and powerful APIs without the need for a Backend or Server.

The result of the project will be a cross-platform social networking application for students, contributing to clarifying the role of social networking sites in education and learning activities. The research results will also provide a model of learning through the internet, contributing to the diversification of current forms of education and training as well as promoting the effectiveness of the internet in learning activities.

***Keywords:*** SDK, iOS, API.

## CHƯƠNG 1. GIỚI THIỆU

### 1.1 Đặt vấn đề

Cùng với quá trình toàn cầu hóa và sự phát triển của công nghệ thông tin, mạng internet trên thế giới và Việt Nam ngày càng phát triển mạnh mẽ. Sự tham gia của các cá nhân trên mạng ngày càng tích cực. Và nhu cầu chia sẻ thông tin, kết nối bạn bè là nhu cầu thiết yếu thúc đẩy sự ra đời và phát triển của các mạng xã hội.

Thông thường trong trường Đại học Cần Thơ, thì sinh viên thường trao đổi thông tin tài liệu thông qua một hay nhiều forum, nhưng chủ yếu nhiều nhất là qua các nhóm Facebook hoặc Zalo, mỗi forum có thể được thành lập từ những tập thể của trường có thể hoạt động riêng rẽ không đồng bộ và không mang tính xã hội cao. Nhưng thực tế cho thấy forum không đáp ứng đủ nhu cầu chia sẻ, giải trí, trao đổi và nhất là mức độ thân thiện giữa các thành viên với nhau như mạng xã hội được. Nhưng nếu có thể xây dựng một mạng xã hội của riêng trường, và quản lý nó, định hướng sinh viên trong trường sinh hoạt trong mạng đó và sử dụng nó vào mục đích học tập, trao đổi, kết bạn, giải trí. Một mạng xã hội theo đúng nghĩa cho sinh viên là cần thiết. Đây cũng là lý do chính mà đề tài “Xây dựng mạng xã hội cho sinh viên Đại học Cần Thơ” được thực hiện

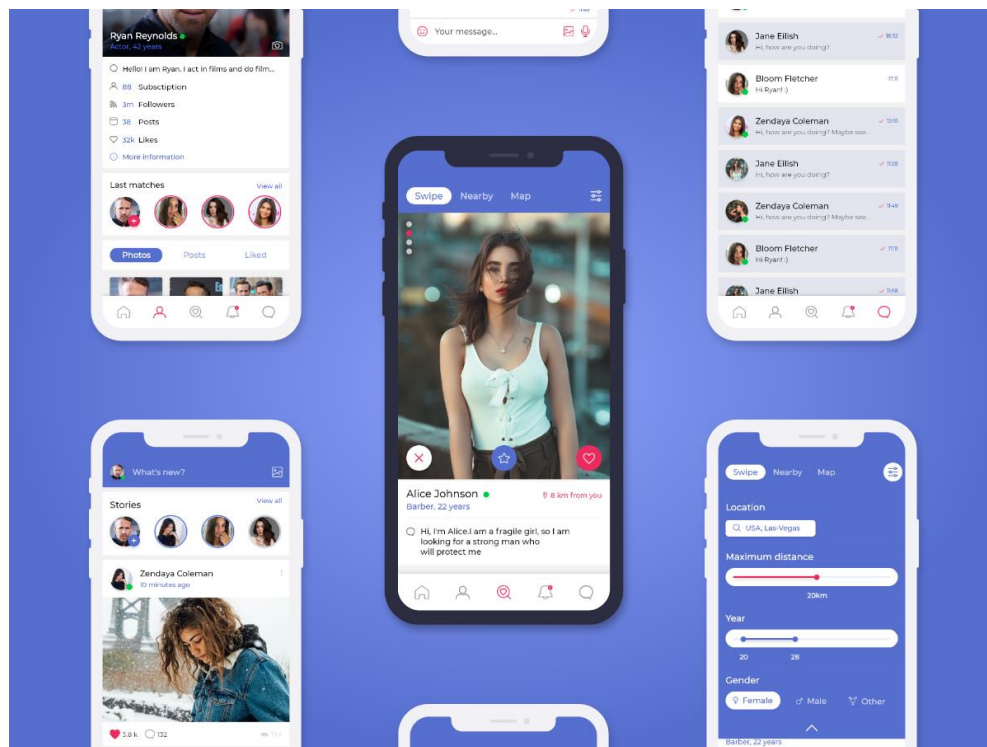
## 1.2 Các nghiên cứu liên quan

### 1.2.1 Mạng xã hội tiêu biểu

Mạng xã hội hay gọi là mạng xã hội ảo (Social Network) là dịch vụ nối kết các thành viên cùng sở thích trên internet lại với nhau với nhiều mục đích khác nhau không phân biệt không gian và thời gian.

Mạng xã hội có những tính năng như chat, e-mail, phim ảnh, voice chat, chia sẻ files, blog, và xã luận. Mạng đổi mới hoàn toàn cách cư dân mạng liên kết với nhau và trở thành một phần tất yếu của mỗi ngày cho hàng trăm triệu thành viên khắp thế giới. Các dịch vụ này có nhiều phương cách để các thành viên tìm kiếm bạn bè, đối tác: dựa theo Groups (ví dụ như tên trường hoặc tên thành phố), dựa trên thông tin cá nhân (như địa chỉ e-mail hoặc screen name), hoặc dựa trên sở thích cá nhân (như thể thao, phim ảnh, sách báo, hoặc ca nhạc), lĩnh vực quan tâm: kinh doanh, mua bán...

### 1.2.2 Facebook<sup>1</sup>



Hình 1.1 Giao diện mạng xã hội Facebook

<sup>1</sup> <https://trainghiemkhac.vn/tim-hieu-loi-ich-va-tac-hai-cua-mang-xa-hoi-facebook/>

Điều tích cực đầu tiên của Facebook là xu hướng thích khám phá cái mới - động lực để người sử dụng Internet, nhất là giới trẻ nồng nhiệt đón nhận Facebook. Kế tiếp, ưu điểm của Facebook so với các mạng xã hội trước đây là độ tương tác, tính trò chuyện và kết nối cao hơn. Nguyên nhân thứ ba là Facebook đáp ứng được nhu cầu đa dạng của con người một cách dễ dàng và nhanh chóng. Con người, sau những nhu cầu thiết yếu để tồn tại như: ăn, uống... thì các nhu cầu về tinh thần như kết nối, liên lạc với cộng đồng, thể hiện khả năng và cập nhật thông tin trở nên cực kỳ quan trọng trong thời đại ngày nay.

Tuy nhiên, thay vì gặp trực tiếp để nói chuyện thì giới trẻ chỉ dành thời gian để trao đổi trên mạng xã hội Facebook. Dần dần họ sẽ mất các kỹ năng sống, kỹ năng xử lý tình huống. Không ít người có biểu hiện “nghiện” Facebook 8 như việc sử dụng trở thành thói quen, có hệ thống và tâm lý bị lệ thuộc mạnh mẽ. Đây cũng là một bệnh lý về tâm thần của người nghiện Facebook. Nếu không sử dụng thì những người “nghiện” Facebook này sẽ có trạng thái nôn nao, khó chịu, buồn bã...

Từ đó thấy được Facebook là kênh giao tiếp hiện đại, tiện lợi, tiết kiệm cho con người. Thông qua Facebook, con người trên toàn thế giới có thể “nối vòng tay lớn”. Bên cạnh đó, Facebook chính là nguyên nhân gây ra một số tiêu cực bởi sự tác động mạnh mẽ của nó.

### 1.2.3 Instagram<sup>2</sup>

Một trong những lợi thế của Instagram chính là giao diện với rất nhiều hình ảnh nội dung mà bạn có thể mong đợi. Trái ngược với điều này, Instagram rất gọn gàng và có tổ chức thay vì loạn như trên Facebook. Bạn cũng có thể tổ chức hồ sơ một cách rất hiệu quả để nó hấp dẫn mọi người. Nó rất hữu ích để chia sẻ ảnh, như kiểu tóc mới, ăn mặc sang trọng, địa điểm du lịch... mà bạn muốn khoe với thế giới với một số lượng bộ lọc tuyệt vời nhằm mang đến trải nghiệm đẹp mắt hơn.

Ban đầu được thiết kế để kết nối mọi người nhưng giờ đây Instagram trông như một sản phẩm hỗ trợ tiếp thị hiệu quả. Các thương hiệu tạo ra Instagram để giới thiệu sản phẩm, trong khi các startup dễ dàng hơn trong việc thu hút khách hàng. Ngay cả những cá nhân như người mẫu hoặc nghệ sĩ cũng có thể giới thiệu tác phẩm riêng trước khi thu hút fan hâm mộ. Và trên hết Instagram rất coi trọng tính riêng tư khi bạn có thể lựa chọn ẩn ảnh trước những người theo dõi. Ngay cả những ảnh hồ sơ, Instagram cũng không cho phép tải xuống, mặc dù các trang mạng xã hội khác hỗ trợ điều này.

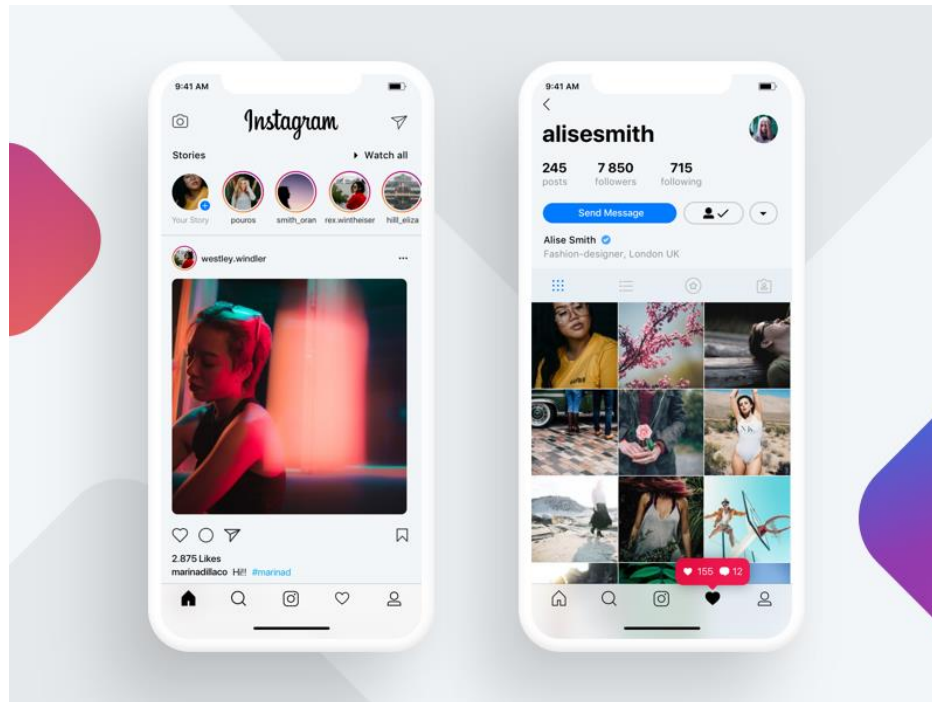
---

<sup>2</sup> <https://www.24h.com.vn/cong-nghe-thong-tin/nhung-uu-va-nhuoc-diem-nen-biet-trong-the-gioi-showbiz-instagram-c55a907982.html>



Bên cạnh đó Instagram được thành lập như một trang web chia sẻ ảnh nên nó vẫn mãi bám với chức năng này. Đáng buồn là khả năng chia sẻ ảnh đã giới hạn đối với đối tượng, điều mà Facebook và Twitter rất năng động mỗi khi người dùng cập nhật trạng thái.

Mặc dù có giao diện và các tính năng tuyệt vời nhưng Instagram có những nhược điểm về kỹ thuật. Nền tảng này chỉ được tối ưu cho ứng dụng, do đó trải nghiệm trên web là khá tệ, chẳng hạn hình ảnh không được tối ưu hóa cho mọi hình dạng như Facebook, trong khi cài đặt bảo mật không thể đặt cho các ảnh riêng lẻ.



**Hình 1.2** Giao diện mạng xã hội Instagram

### 1.2.4 Zalo<sup>3</sup>

Zalo là ứng dụng nhắn tin và gọi điện miễn phí hoạt động trên nền tảng di động và máy tính. Zalo được phát triển bởi công ty VNG và ra mắt phiên bản đầu tiên vào ngày 8/8/2012. Với những tính năng cho phép người dùng cập nhật thông tin, hình ảnh trên trang cá nhân Đăng ký cực kỳ đơn giản: Không cần khai báo bất kỳ thông tin nào cũng không cần check mail để xác nhận, người dùng chỉ cần đăng ký bằng chính số điện thoại của mình, thậm chí cho dù có bị mất tài khoản, mật khẩu, người dùng cũng chỉ cần duy nhất số điện thoại để lấy lại. Khả năng chia sẻ thông tin (file, hình ảnh với chất lượng HD) lên tới 20MB và có thể thực hiện với nhiều file cùng một lúc giúp Zalo là lựa chọn số một của dân công chức, sinh viên

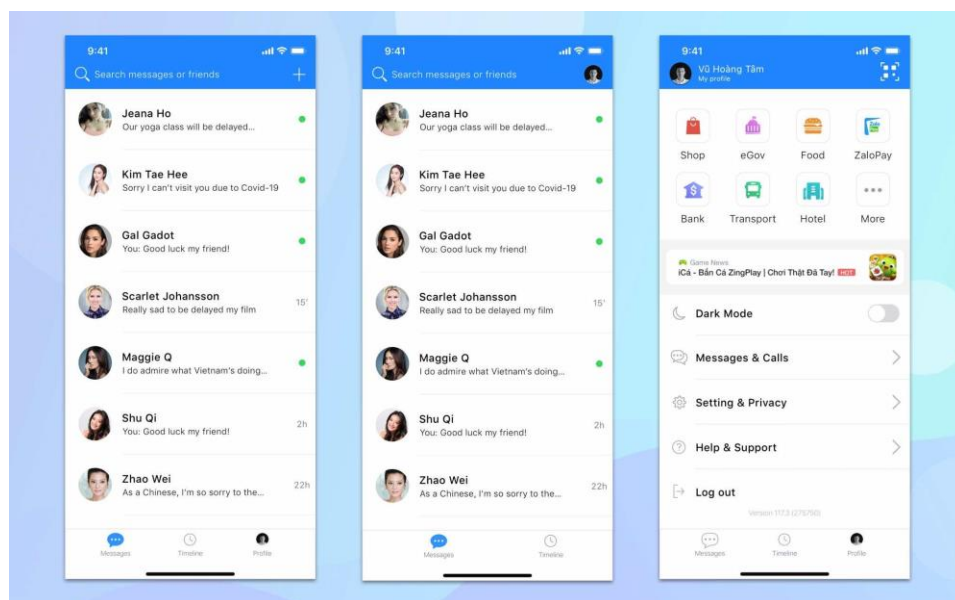
---

<sup>3</sup> <https://savitel.com.vn/tin-tuc/uu-va-nhuoc-diem-hop-truc-tuyen-qua-zalo.html>

hay giới văn phòng. Khả năng ghi âm lời thoại tới 5 phút cùng chất lượng cuộc thoại, khi nhắn tin vô cùng ổn định dù chỉ được kết nối 3G.

Ngoài ra, với việc có thể sử dụng được trên hầu hết các nền tảng smartphone (iOS, Android, Windows phone) hay hệ điều hành của máy tính (Windows, Mac OS X, Linux) cũng góp phần không nhỏ biến Zalo thành lựa chọn số một trong số những OTT cùng loại.

Nhưng do chỉ sử dụng số điện thoại để đăng ký nên số lượng tài khoản ảo của Zalo rất nhiều, trao đổi thông tin còn hạn chế do chỉ xoay quanh những người quen biết, và còn chưa thể kết nối trực tiếp với các trang mạng xã hội Facebook, Twitter...



**Hình 1.3** Giao diện mạng xã hội Zalo

### **1.2.5 Ứng dụng mạng xã hội CTUPG được thực hiện trong đề tài**

Với ưu điểm là mạng xã hội dành riêng và duy nhất cho sinh viên Đại học Cần Thơ, ứng dụng có thể đảm bảo tính bảo mật cho người dùng vì ứng dụng sử dụng mail của trường, và cũng là email duy nhất sinh viên được cấp. Ứng dụng sẽ hỗ trợ chức năng khôi phục mật khẩu bằng email và điều này đồng nghĩa với đảm bảo một trăm phần trăm rằng sẽ không xảy ra việc mất cắp tài khoản. Ứng dụng còn hỗ trợ chế độ ẩn danh, để sinh viên có thể sử dụng diễn đàn để đặt và được giải đáp những thắc mắc trong vấn đề học tập hay cá nhân. Các nội dung sẽ được quản lý bởi admin để đảm bảo sự tự do đặt câu hỏi đó đều không vượt quá giới hạn. Ngoài ra, chế độ khách cũng sẽ được tích hợp để người từ ngoài trường có thể tham quan môi trường mạng xã hội của sinh viên Đại học Cần Thơ. Giúp mọi người có cái nhìn tốt hơn về 1 mạng xã hội mang tính giáo dục lành mạnh.

Tuy nhiên ứng dụng vẫn chưa thực sự liên kết chặt chẽ được với nhà trường, đơn thuần là 1 mạng xã hội cho sinh viên trao đổi, học tập. Các chức năng chưa được đa dạng. Nhưng cũng sẽ đảm bảo đáp ứng được nhu cầu trao đổi học tập, kết bạn của sinh viên.

### 1.3 Mục tiêu của đề tài

Mục tiêu của đề tài là nhằm đề xuất một mô hình mạng xã hội dành riêng cho sinh viên để góp phần làm phong phú các hình thức học tập song song với giải trí, phát huy hiệu quả của Internet đối với hoạt động giáo dục, đào tạo. Từ đó xây dựng một ứng dụng mạng xã hội cho sinh viên chia sẻ khoảnh khắc, trao đổi thông tin với giao diện tiếng Việt thân thiện và dễ sử dụng. Từ việc tham khảo ý kiến từ các diễn đàn và khảo sát sinh viên trong và ngoài trường và hiện trạng thực tế của sinh viên, có thể thấy nhu cầu cho 1 mạng xã hội dành cho sinh viên là không hề nhỏ. Nắm được các ưu nhược điểm của các mạng xã hội hiện tại đơn cử như vấn đề bảo mật do số lượng người dùng quá lớn, giới hạn giữa các nền tảng lớn như Android, iOS, web. Tác giả đã chọn thực hiện đề tài **Xây dựng ứng dụng mạng xã hội dành riêng cho sinh viên Đại học Cần Thơ** sử dụng framework Flutter nhằm đáp ứng những nhu cầu thực tiễn như sau của sinh viên:

- Tạo một môi trường liên kết thân thiện giữa các sinh viên của trường: bằng việc kết bạn, giao lưu, tham gia hội nhóm môn học, sở thích, thuộc cùng một khoa, một khóa hay giữa các khoa các ngành, các khóa khác nhau. Sinh viên có thể thấy được sinh viên ngành khác hoạt động như thế nào để tham khảo kinh nghiệm và tiện cho việc liên lạc trong những nghiên cứu liên quan đến nhiều ngành, hoặc xin thông tin liên hệ của các thầy cô trong và ngoài ngành học để trao đổi được dễ dàng hơn. Hơn thế nữa môi trường này sẽ xóa đi sự gò bó về khoảng cách về khác biệt kiến thức, độ tuổi hay giới tính, tăng sự gần gũi hơn giữa những sinh viên với nhau.
- Một ứng dụng mạng xã hội đa nền tảng nhờ vào sử dụng framework Flutter, hỗ trợ ứng dụng hoạt động mượt mà trên các thiết bị Android, iOS và trên các trình duyệt web.
- Mỗi sinh viên sẽ có một không gian chia sẻ riêng biệt cho riêng mình: Họ có thể trao đổi, chat trực tuyến với những người bạn của mình, và cũng có thể upload chia sẻ những hình ảnh ngộ nghĩnh, những video lý thú của mình cho mọi người.
- Các sinh viên của trường có thể liên hệ, kết bạn, chia sẻ những kinh nghiệm trong công việc, đời sống mà họ đã đích thân trải nghiệm. Điều đó rất dễ dàng đối với một mạng xã hội, vượt ngoài giới hạn về địa lý và thời gian. Đó cũng là cơ hội cho những tân sinh viên của trường có thể dễ dàng tiếp nhận cách học và cách sinh hoạt trong môi trường mới mẻ này.
- Nơi mà những sinh viên có thể sớm tiếp cận với những nhà tuyển dụng trong tương lai, và học hỏi rất nhiều từ các yêu cầu chuyên môn hay những kỹ năng cần thiết cho công việc sau này. Tại đây họ có thể trao đổi thông tin một cách gần gũi, nhẹ nhàng, không mang nặng

tâm lý như những cuộc điện thoại hay gặp mặt lần đầu tiên nhờ vào khả năng ẩn danh của ứng dụng.

- Các bạn tình nguyện viên có thể kêu gọi sự ủng hộ của mọi người thông qua các cuộc vận động đầy ý nghĩa: chiến dịch mùa hè xanh, tình nguyện viên trong mùa thi đại học, hay đền ơn đáp nghĩa. Sẽ thật dễ dàng nếu lời kêu gọi đó được truyền đi một cách nhanh chóng trên các danh sách bạn bè của họ.

- Một một sân chơi lành mạnh, hấp dẫn đối với sinh viên: Thông qua một số cuộc hội thảo online tự quyền về các chủ đề môn học, hay việc làm, và các trò chơi trực tuyến thú vị.

- Nơi sinh viên có thể đóng góp ý kiến của mình đối với các hiện trạng xã hội hiện nay, những vấn đề cộng đồng thông qua các bài xã luận, hay những bức xúc của sinh viên đối với trường học một cách tự do đúng đắn.

### **1.4 Đối tượng và phạm vi nghiên cứu**

Đề tài tập trung nghiên cứu về đối tượng chính là sinh viên, các lý thuyết và kinh nghiệm thực tế về mạng xã hội, những nhu cầu cần thiết cho một mạng xã hội sinh viên. Kèm theo đó là nghiên cứu về mối quan hệ giữa mạng xã hội và hoạt động giáo dục, giữa sinh viên và trường từ đó đưa ra mô hình ứng dụng mạng xã hội phù hợp với sinh viên trường Đại học Cần Thơ cùng với giao diện thân thiện cho người dùng và quản trị.

### **1.5 Phương pháp nghiên cứu**

Phương pháp chủ yếu là tổng hợp, nghiên cứu dựa trên các kết quả nghiên cứu đã có, từ đó đề xuất mô hình, giải pháp phù hợp với thực trạng mạng xã hội sinh viên hiện tại ở Việt Nam. Sau đó là tìm hiểu và nắm rõ về mặt lý thuyết cũng như cách hoạt động của SDK Flutter, Dart, mô hình MVC, Firebase, Widget. Và sau khi nắm rõ lý thuyết, sẽ bắt tay vào xây dựng một ứng dụng mạng xã hội sinh viên mang tên CTUPG.

#### **1.5.1 Về lý thuyết**

- Tìm hiểu về framework Flutter và các thư viện hỗ trợ cho việc xây dựng giao diện cho ứng dụng
- Tìm hiểu về Firebase/Firestore/Firebase Auth/ Storage để hỗ trợ cho việc lưu trữ/ truy vấn dữ liệu ứng dụng
- Tìm hiểu ngôn ngữ OOP Dart.
- Tìm hiểu về mô hình phù hợp cho mạng xã hội sinh viên trong trường.

#### **1.5.2 Về kỹ thuật**

- Sử dụng ngôn ngữ lập trình Dart và framework Flutter để thiết kế giao diện
- Sử dụng Firebase/Firestore để quản lý cơ sở dữ liệu

#### **1.5.3 Về nguồn tài liệu**

- Các tài liệu thu thập phần lớn là từ Internet, còn lại là tham khảo từ các sách tài liệu, nghiên cứu liên quan trong chuyên ngành.

### 1.6 Bố cục luận văn

Bố cục của luận văn: Ngoài phần các ký hiệu và chữ viết tắt, danh mục các bảng, danh mục các hình vẽ, mở đầu và kết luận, danh mục tài liệu tham khảo, nội dung luận văn gồm bốn phần:

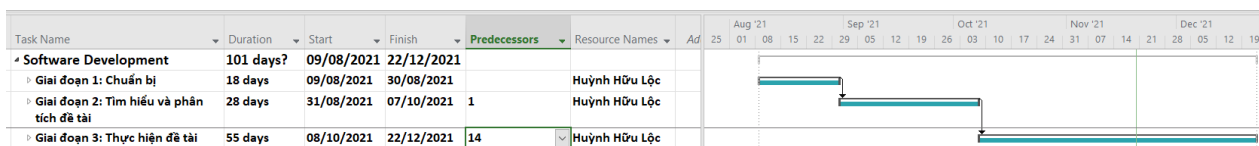
**Chương 1. Giới thiệu** - Chương này tác giả giới thiệu về khái niệm, tầm quan trọng, hiện trạng mạng xã hội trên thế giới và Việt Nam, nhu cầu trao đổi học tập và tương tác với bạn bè của sinh viên hiện nay.

**Chương 2. Cơ sở lý thuyết** - Trong chương này tác giả giới thiệu về các cơ sở lý thuyết cơ bản, về Framework Flutter kèm theo 1 số thành phần cơ bản và chức năng của các thành phần đó, từ đó ta có cơ sở xây dựng một cách tổng quan và khái quát các chức năng chính của hệ thống ở Chương 3.

**Chương 3. Phương pháp thực hiện** - Xây dựng mạng xã hội cho sinh viên Đại học Cần Thơ - Chương này tác giả trình bày quy trình thực hiện việc lập trình ứng dụng. Đưa ra giải pháp công nghệ sử dụng cơ sở dữ liệu và ngôn ngữ lập trình. Cuối cùng là phân tích các chức năng của hệ thống mạng xã hội và đưa ra một số giao diện chính.

**Chương 4. Kết quả xây dựng ứng dụng** - Cài đặt và thử nghiệm - Chương này đưa ra yêu cầu phần cứng và phần mềm của hệ thống, dữ liệu thử nghiệm và đưa ra bộ tiêu chí đánh giá kèm hình ảnh.

## 1.7 Kế hoạch làm việc



Hình 1.4 Tóm tắt các giai đoạn

<b>Giai đoạn 1: Chuẩn bị</b>	<b>18 days</b>	<b>09/08/2021</b>	<b>30/08/2021</b>		<b>Huỳnh Hữu Lộc</b>
<b>Tạo giao diện cơ bản</b>	<b>4 days</b>	<b>09/08/2021</b>	<b>12/08/2021</b>		
Tạo sẵn giao diện, chọn màu phù hợp	1 day	09/08/2021	09/08/2021		
Tham khảo và sử dụng các thư viện design phù hợp	2 days	10/08/2021	11/08/2021	3	
Đảm bảo sự ổn định của giao diện trên các nền tảng khác nhau	1 day	12/08/2021	12/08/2021	4	
<b>Thao tác trên CSDL</b>	<b>4 days</b>	<b>13/08/2021</b>	<b>18/08/2021</b>	<b>2</b>	
Thiết kế CSDL	2 days	13/08/2021	16/08/2021		
Đưa mô hình CSDL lên server và bổ sung dữ liệu	1 day	17/08/2021	17/08/2021	7	
Kiểm tra độ chính xác dữ liệu qua các thao tác thêm sửa xóa	1 day	18/08/2021	18/08/2021	8	
<b>Phân quyền</b>	<b>4 days</b>	<b>19/08/2021</b>	<b>22/08/2021</b>	<b>6</b>	
Phân quyền và đảm bảo nhân	3 days	19/08/2021	21/08/2021		

Hình 1.5 Giai đoạn 1 của kế hoạch (1)



## ABSTRACT

---

➤ <b>Phân quyền</b>	<b>4 days</b>	<b>19/08/2021</b>	<b>22/08/2021</b>	<b>6</b>	
Phân quyền và đảm bảo phân quyền đúng cho Admin, User	3 days	19/08/2021	21/08/2021		
Mở rộng thêm nhóm người dùng (guest)	1 day	22/08/2021	22/08/2021	11	
<b>Kiểm tra độ ổn định của framework</b>	<b>6 days</b>	<b>23/08/2021</b>	<b>30/08/2021</b>	<b>10</b>	

**Hình 1.6 Giai đoạn 1 của kế hoạch (2)**

➤ <b>Giai đoạn 2: Tìm hiểu và phân tích đề tài</b>	<b>28 days</b>	<b>31/08/2021</b>	<b>07/10/2021</b>	<b>1</b>	<b>Huỳnh Hữu Lộc</b>
Tìm tất cả các thông tin liên quan đến đề tài và các vấn đề liên quan	7 days	31/08/2021	08/09/2021		
➤ <b>Tìm hiểu các đề tài/ ứng dụng mạng xã hội tương tự</b>	<b>21 days</b>	<b>09/09/2021</b>	<b>07/10/2021</b>	<b>15</b>	
Tìm hiểu điểm giống và khác nhau giữa các ứng dụng	7 days	09/09/2021	17/09/2021		
Tìm hiểu và phân tích ưu nhược điểm của ứng dụng	7 days	20/09/2021	28/09/2021	17	
Tham khảo và thu thập thông tin để cải tiến cơ sở dữ liệu	7 days	29/09/2021	07/10/2021	18	

**Hình 1.7 Giai đoạn 2 của kế hoạch**

## ABSTRACT

◀ <b>Giai đoạn 3: Thực hiện đề tài</b>	<b>55 days</b>	<b>08/10/2021</b>	<b>22/12/2021</b>	<b>14</b>	<b>Huỳnh Hữu Lộc</b>
◀ <b>Thiết kế</b>	<b>16 days</b>	<b>08/10/2021</b>	<b>29/10/2021</b>		
Xem xét các thông số kỹ thuật sơ bộ của phần mềm	2 days	08/10/2021	11/10/2021	19	
Thiết lập các thư viện cần thiết	3 days	12/10/2021	14/10/2021	22	
Tạo các model và function cốt lõi	7 days	15/10/2021	25/10/2021	23	
Kết nối tới cơ sở dữ liệu Firebase và thực hiện thao tác chỉnh sửa cơ bản	3 days	26/10/2021	28/10/2021	24	
Nhận phản hồi từ giáo viên để tiếp tục	8 hrs	29/10/2021	29/10/2021	25	
Thiết kế hoàn tất	0 days	29/10/2021	29/10/2021	26	
◀ <b>Xây dựng phần mềm</b>	<b>37 days</b>	<b>30/10/2021</b>	<b>20/12/2021</b>	<b>21</b>	
Viết ứng dụng	31 days	30/10/2021	10/12/2021		Huỳnh Hữu Lộc
Hoàn thiện ứng dụng	6 days	13/12/2021	20/12/2021	29	
◀ <b>Kiểm thử</b>	<b>2 days</b>	<b>21/12/2021</b>	<b>22/12/2021</b>	<b>28</b>	
Kiểm thử trên hệ điều hành Android, iOS	1 day	21/12/2021	21/12/2021		

**Hình 1.8 Giai đoạn 3**

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1 Giới thiệu

#### 2.1.1 Tổng quan về mạng xã hội

##### 2.1.1.1 Khái niệm

Mạng xã hội là một nền tảng trực tuyến, là nơi mà mọi người có thể xây dựng các mối quan hệ ảo với những người có chung tính cách, sở thích, nghề nghiệp... hoặc với cả những người có mối quan hệ ngoài đời thực.

Mạng xã hội hiện nay có nhiều dạng thức và tính năng khác nhau, và có thể truy cập dễ dàng từ nhiều phương tiện, thiết bị như máy tính bảng, laptop, điện thoại di động, ...

##### 2.1.1.2 Đặc điểm của mạng xã hội

Các đặc điểm của 1 mạng xã hội cụ thể như sau:

- Mạng xã hội là nền tảng trên Internet.
- Người dùng trên mạng xã hội phải có tài khoản và hồ sơ riêng.
- Mạng xã hội tạo ra các liên kết thông qua các tài khoản ảo do người dùng tạo ra.
- Mọi nội dung trên mạng xã hội đều do chính người dùng sáng tạo ra.

##### 2.1.1.3 Vai trò của mạng xã hội

Nhìn từ góc độ quản lý nhà nước, MXH là hệ thống thông tin cung cấp cho cộng đồng người sử dụng mạng các dịch vụ lưu trữ, cung cấp, sử dụng, tìm kiếm, chia sẻ và trao đổi thông tin với nhau, bao gồm dịch vụ tạo trang thông tin điện tử cá nhân, diễn đàn, trò chuyện trực tuyến, chia sẻ âm thanh, hình ảnh và các hình thức dịch vụ tương tự khác.

Nhìn từ góc độ văn hóa- xã hội, MXH là tập hợp các mối quan hệ giữa các cá nhân, nhóm cá nhân, tổ chức trên môi trường internet. Chính vì thế, MXH có thể coi là một loại hình cộng đồng song mang tính chất ảo, trong đó bao gồm nhiều cộng đồng trực tuyến khác nhau nhằm thỏa mãn các nhu cầu về vật chất và tinh thần của con người. Một số cộng đồng như Facebook, Youtube, Zalo... thu hút một số lượng lớn người tham gia, ngày càng đóng vai trò quan trọng trong đời sống xã hội.

Từ đó ta thấy được rằng MXH góp phần tích cực vào sự phát triển nhận thức, tư duy và kỹ năng sống của con người.

### **2.1.1.4 Vai trò của mạng xã hội trong học tập**

Đến với hình thức này, người học chia sẻ những kiến thức có được thông qua các trang mạng xã hội để nhiều cá nhân khác có thể tiếp thu. Theo thời gian, phương pháp này tạo nên một cộng đồng lớn, nơi mà chúng ta có thể trao đổi thông tin, kiến thức, cùng nhau tranh luận về một vấn đề nào đó. Nhờ vậy, người học có thể cải thiện kiến thức của mình, tăng khả năng tương tác với mọi người cũng như có thêm nhiều mối quan hệ mới.

Mạng xã hội có thể định hình và trình bày thông tin theo cách có ý nghĩa giúp kích thích người học hơn các công cụ truyền thống, cho dù đó là một bài viết được chia sẻ với chức năng bình luận, phát trực tiếp một sự kiện quan trọng hay khảo sát liên quan đến tài liệu khóa học.

Hơn nữa, chia sẻ kiến thức với người học khác, thúc đẩy sự tham gia sâu hơn và hiệu suất tốt hơn từ mọi người. Nếu người học tương tác lẫn nhau với các tài liệu khóa học trên những mạng xã hội, cùng nhau họ có thể nỗ lực nhiều hơn tham gia các chương trình đào tạo.

Lợi ích giáo dục mà mạng xã hội mang lại giúp người học tìm kiếm, tiếp nhận những kiến thức quan trọng và học các khái niệm nhất định với hiệu quả tuyệt vời giúp họ giải quyết được vấn đề của mình. Mạng xã hội luôn cập nhật và chứa đựng những thông tin mới nhất về các chủ đề khác nhau. Mạng xã hội như một Wikipedia cho người học ngày nay. Chúng ta đang được chứng kiến một kỷ nguyên học tập thông qua mạng xã hội.

## 2.2 Ngôn ngữ và công nghệ sử dụng

### 2.2.1 Ngôn ngữ Dart<sup>4</sup>

Dart là ngôn ngữ được phát triển bởi Google, và phiên bản hiện tại của Dart trong lúc tác giả thực hiện đề tài là **2.14.3**.

Dart là một ngôn ngữ thuần OOP (hướng đối tượng), giống giống Java, là cải tiến lên từ Javascript, cũng là hệ C nên cú pháp tương tự nhau (tương tự cả C#), một ngôn ngữ mà nó cải tiến hơn Javascript và hỗ trợ cả 2 cách biên dịch là Just In Time (JIT) và Ahead Of Time (AOT):

- Ahead Of Time (AOT): Với AOT thì trình biên dịch chuyển ngôn ngữ Dart thẳng sang Native Code giúp hiệu năng tốt nhất có thể (tức là khi chạy chương trình, nó sẽ biên dịch từ đầu đến cuối)
- Just In Time (JIT): Còn với JIT cho phép hot reloading hoạt động, giúp phát triển sản phẩm nhanh và tiện dụng hơn (được hiểu như việc debug trong ngôn ngữ khác là debug hàm nào chạy hàm đó thì ở đây nó sẽ viết đến đâu biên dịch ngay đến đấy).

### 2.2.2 Widgets<sup>5</sup>

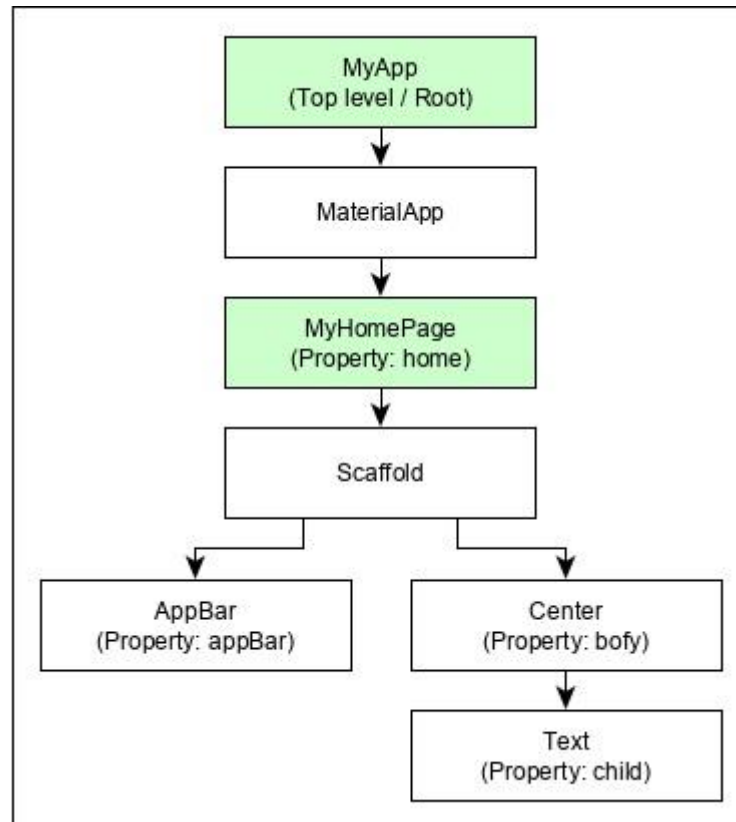
Khái niệm cốt lõi nhất trong Flutter framework đó là **trong Flutter, mọi thứ đều là widget**. Widget (tiện ích) là thành phần giao diện cơ bản nhất tạo nên toàn bộ giao diện người dùng của ứng dụng.

Trong *Flutter*, bản thân chính ứng dụng đã là một widget. Mỗi ứng dụng chính là một **top-level widget** và nó bao gồm một hoặc nhiều các widget con, mỗi widget này lại có thể bao gồm một hoặc nhiều widget con khác. Nhờ sự kết hợp linh hoạt này chúng ta có thể tạo ra bất kì ứng dụng phức tạp nào.

---

<sup>4</sup> <https://game.tintucthegioi.net/dart-la-gi/>

<sup>5</sup> <https://vncoder.vn/bai-hoc/kien-truc-ung-dung-flutter-215>



**Hình 2.1 Cấu trúc Widget**

Giải thích sơ đồ trên:

- **MyApp** là một widget được tạo ra bằng widget gốc của Flutter, MaterialApp.
- **MaterialApp** có các thuộc tính của màn hình home và mô tả giao diện người dùng, nó lại được tạo ra bởi một widget khác, MyHomePage.
- **MyHomePage** được tạo bởi một widget gốc của Flutter, Scaffold
- **Scaffold** có 2 thuộc tính – body và appBar
- body chứa giao diện chính còn **appBar** chứa phần đầu (header) của ứng dụng
- **Header UI** là một widget gốc của flutter, AppBar và Body UI sử dụng Center widget.
- **Center** widget có một thuộc tính, Child, nó chứa phần nội dung chính là một Text widget

Trong **Flutter** tất các widget được phân loại dựa trên chức năng thành 4 nhóm sau:

- Các widget giao diện đặc thù theo từng nền tảng (**Platform widgets**)

- Các widget hỗ trợ bố trí giao diện (**Layout widgets**)
- Các widget quản lý trạng thái (**State maintenance widgets**)
- Các widget cơ bản độc lập với nền tảng (**Platform independent / basic widgets**)

### 2.2.3 SDK Flutter<sup>6</sup>

Flutter được phát triển nhằm giải quyết bài toán thường gặp trong mobile là Fast Development và Native Performance. Nếu như React Native chỉ đảm bảo Fast Development và code native thuần chỉ đảm bảo Native Performance thì Flutter làm được cả 2 điều trên.

Là một SDK mới của google dành cho các thiết bị di động giúp developers và designers xây dựng nhanh chóng ứng dụng dành cho các thiết bị di động (Android, iOS). Flutter là dự án mã nguồn mở đang trong giai đoạn thử nghiệm. Flutter bao gồm Reactive framework và công nghệ hiển thị 2D (2D rendering engine) và các công cụ phát triển (development tool). Các thành phần này làm việc cùng nhau giúp ta thiết kế, xây dựng, test, debug ứng dụng.

Flutter được thiết kế từ đầu để đạt tới tốc độ khung hình 60fps. Trong khi đây không phải là một con số hiếm gặp trên iOS, nhưng với Android, bạn có thể cảm thấy sự khác biệt rõ rệt ngay lập tức. Cũng nhờ việc xuất xưởng cùng với cả một nền tảng cho ứng dụng của mình, các nhà phát triển sẽ tránh được nhiều vấn đề về sự phân mảnh của Android.

Nó dựa trên ngôn ngữ Dart và cung cấp thư viện đồ họa hấp dẫn. Các feature chính của platform này là:

- SDK cho platform chính xác và chi tiết
- Các widget được thiết kế chi tiết, chính xác
- Môi trường hoàn toàn tùy chỉnh
- Native Performance
- Feature Stateful Hot Reload để phát triển app nhanh hơn

### 2.2.4 Layers<sup>7</sup>

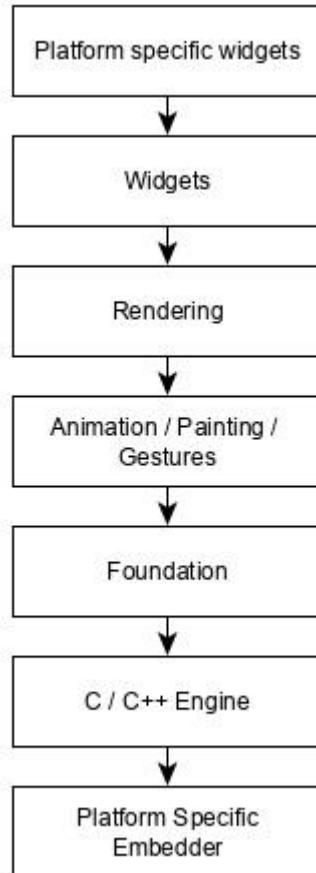
Một khái niệm quan trọng của Flutter framework đó là các thành phần sẽ được nhóm lại theo độ phức tạp và được sắp xếp rõ ràng trong các tầng có độ phức tạp giảm dần. Một **Layer** (tầng, lớp) được tạo thành bằng việc sử dụng các class tiếp theo ngay cạnh nó.

---

<sup>6</sup> <https://topdev.vn/blog/flutter-la-gi/>

<sup>7</sup> <https://vncoder.vn/bai-hoc/kien-truc-ung-dung-flutter-215>

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 2.2 Tổng quan về layer của Flutter**

### 2.2.5 Firebase (Server) <sup>8</sup>

**Firebase** là một dịch vụ cơ sở dữ liệu thời gian thực được cung cấp bởi Google và hoạt động trên nền tảng đám mây. Nó giúp các lập trình phát triển nhanh các ứng dụng di động bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Firebase cung cấp cho chúng ta 2 nhóm sản phẩm chính tập trung vào 2 đối tượng là:

---

<sup>8</sup> <https://topdev.vn/blog/firebase-la-gi/#:~:text=Firebase%20%C3%A0%20m%E1%BB%99t%20d%E1%BB%8Bch%20v%E1%BB%A5,v%E1%BB%9Bi%20c%C6%A1%20s%E1%BB%9F%20d%E1%BB%AF%20li%E1%BB%87u.>



- Develop & test your app: phát triển và kiểm thử các ứng dụng được thiết kế.
- Grow & engage your audience: Phân tích dữ liệu và tối ưu hóa trải nghiệm đối với người dùng.

### 2.2.6 Mô hình thiết kế Model View Controller (MVC)<sup>9</sup>

#### 2.2.6.1 Model View Controller (MVC) là gì?

MVC là tên một phương pháp chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau và ảnh hưởng lẫn nhau, đó là models, views, và controllers.

- **Model** (DAL – Data Access Layer): Models trong các ứng dụng dựa trên MVC là những thành phần có nhiệm vụ lưu trữ thông tin, trạng thái của các đối tượng, thông thường nó là một lớp được ánh xạ từ một bảng trong CSDL. Lấy ví dụ, chúng ta có lớp Product được sử dụng để mô tả dữ liệu từ bảng Products trong SQL, bao gồm ProductID, OrderDate...

- **Views** (Web – Website): Views, nó chính là các thành phần chịu trách nhiệm hiển thị các thông tin lên cho người dùng thông qua giao diện. Thông thường, các thông tin cần hiển thị được lấy từ thành phần Models. Ví dụ, đối tượng Product có một "Edit" view bao gồm các textboxes, các dropdowns và checkboxes để chỉnh sửa các thuộc tính của sản phẩm; có một "Display" view gồm 2 dòng, cột dòng là ProductID, dòng sau là OrderDate... để xem thông tin về sản phẩm.

- **Controllers** (Core): Controllers trong các ứng dụng kiểu MVC chịu trách nhiệm xử lý các tác động về mặt giao diện, các thao tác đối với models, và cuối cùng là chọn một view thích hợp để hiển thị ra màn hình. Trong kiến trúc MVC, view chỉ có tác dụng hiển thị giao diện mà thôi, còn điều khiển dòng nhập xuất của người dùng vẫn do Controllers đảm trách.

Khi người sử dụng ra một lệnh (gõ câu lệnh, bấm nút chuột, bấm phím, chọn menu ...) thì lệnh này sẽ được gửi tới phần Controller.

Phần Điều khiển sẽ khởi tạo phần Model (nếu cần thiết), gửi các yêu cầu tới phần Model để thực hiện. - Căn cứ trên các lệnh và thông tin nhận được từ lệnh, phần Model sẽ đảm nhận việc lấy thông tin và cập nhật thông tin trong các hệ thống khác, ví dụ trong các Enterprise Server như Application Server, Mail Server, Database Server, từ hệ thống.

---

<sup>9</sup> <http://thanhchien3d.vn/model-view-controller-la-gi/>

Sau khi hoàn thành việc thu thập, cập nhật thông tin, Model sẽ truyền những thông tin cần thiết về phần Controller.

Lúc này, phần Controller sẽ quyết định chọn thành phần nào trong phần View để hiện dữ liệu ra cho người dùng.

Phần View khi làm nhiệm vụ hiện thông tin cho người dùng cũng có thể truy cập các thông tin hiển thị từ Model, hoặc gửi thông tin hiển thị tới Model. Trường hợp này xảy ra khi Model chứa các thông tin có thể dùng để hiện trực tiếp, ví dụ một danh sách khách hàng, hoặc một danh sách các email trong mailbox. Khi phần View hiện thông tin, nó có thể báo cho phần Model biết nó đang hiện phần nào của thông tin, ví dụ như “Đang hiện thông tin khách hàng từ 20 đến 40”. Những thông tin loại này không cần thiết phải gửi qua trung gian Controller

### **2.2.6.2 Đặc điểm của Model View Controller (MVC)**

Cho phép sự chia tách rành mạch giữa các thành phần, mỗi thành phần đều có khả năng được kiểm tra riêng biệt. Tất cả các mối liên hệ chính yếu trong MVC đều có giao diện để hiển thị và đều có thể xây dựng thành các mô hình, và bao gồm interface-based `IHttpRequest/IHttpResponse`. Bạn có thể chạy unit test cho ứng dụng mà không phải chạy các Controllers trong ASP.NET process, điều này làm unit test chạy nhanh hơn. Bạn có thể sử dụng unit testing framework nào cũng được, kể cả NUnit, MBUnit, MS Test...

Được thiết kế để có thể dễ mở rộng và tiện dụng, tất cả mọi thứ trong MVC framework đều có thể nhanh chóng được thay đổi, tùy chọn hóa. Ví dụ bạn có thể viết một View và sử dụng cho nhiều ứng dụng khác nhau, đây là tính pluggable.

Bao gồm thành phần ánh xạ URL cho phép bạn xây dựng các ứng dụng thân thiện với bộ máy tìm kiếm và với người dùng. Lấy ví dụ tôi có thể ánh xạ URL `/products/edit/4` là một trang cho phép chỉnh sửa thông tin về các sản phẩm, hoặc là để xem các thông tin.

MVC framework hỗ trợ sử dụng các các tập tin. ASPX, ASCX và. Master như là thành phần View, điều đó có nghĩa là bạn vẫn có thể sử dụng các tính năng của ASP.NET như master pages, snippets, server controls, templates, data-binding, localization... Tuy nhiên nó không sử dụng mô hình post-back từ giao diện gửi đến server nữa, thay vào đó, bạn có thể chủ động đưa những post-back từ giao diện đó đến thẳng lớp Controller. Tóm lại, không còn viewstate hay là page lifecycle còn tồn tại trong mô hình MVC.

### **2.2.6.3 Lợi điểm của Model View Controller (MVC)**

Nó bắt buộc phải tạo ra một sự rành mạch trong liên kết giữa 3 thành phần models, views và controllers trong ứng dụng. Duy trì được mối quan hệ rành mạch rõ ràng này khiến việc

## ABSTRACT

---

kiểm tra ứng dụng trở nên dễ dàng hơn, đồng thời trách nhiệm của từng thành phần được định nghĩa rõ ràng và cả ba đều làm việc ăn ý với nhau.

Nó cho phép bạn dễ dàng bảo trì sự phân tách giữa các thành phần trong ứng dụng, đồng thời cũng rất tốt trong công việc testing

## 2.2.7 Các Widgets quan trọng<sup>10</sup>

### 2.2.7.1 Gestures

**Flutter Widget** hỗ trợ tương tác thông qua một widget đặc biệt gọi là **GestureDetector**. **GestureDetector** là một tiện ích không hiển thị trên giao diện nhưng có khả năng nắm bắt các thao tác của người dùng như nhấp, kéo, vuốt, chạm.... Phần lớn widget gốc của Flutter hỗ trợ tương tác giao diện thông qua **GestureDetector**.

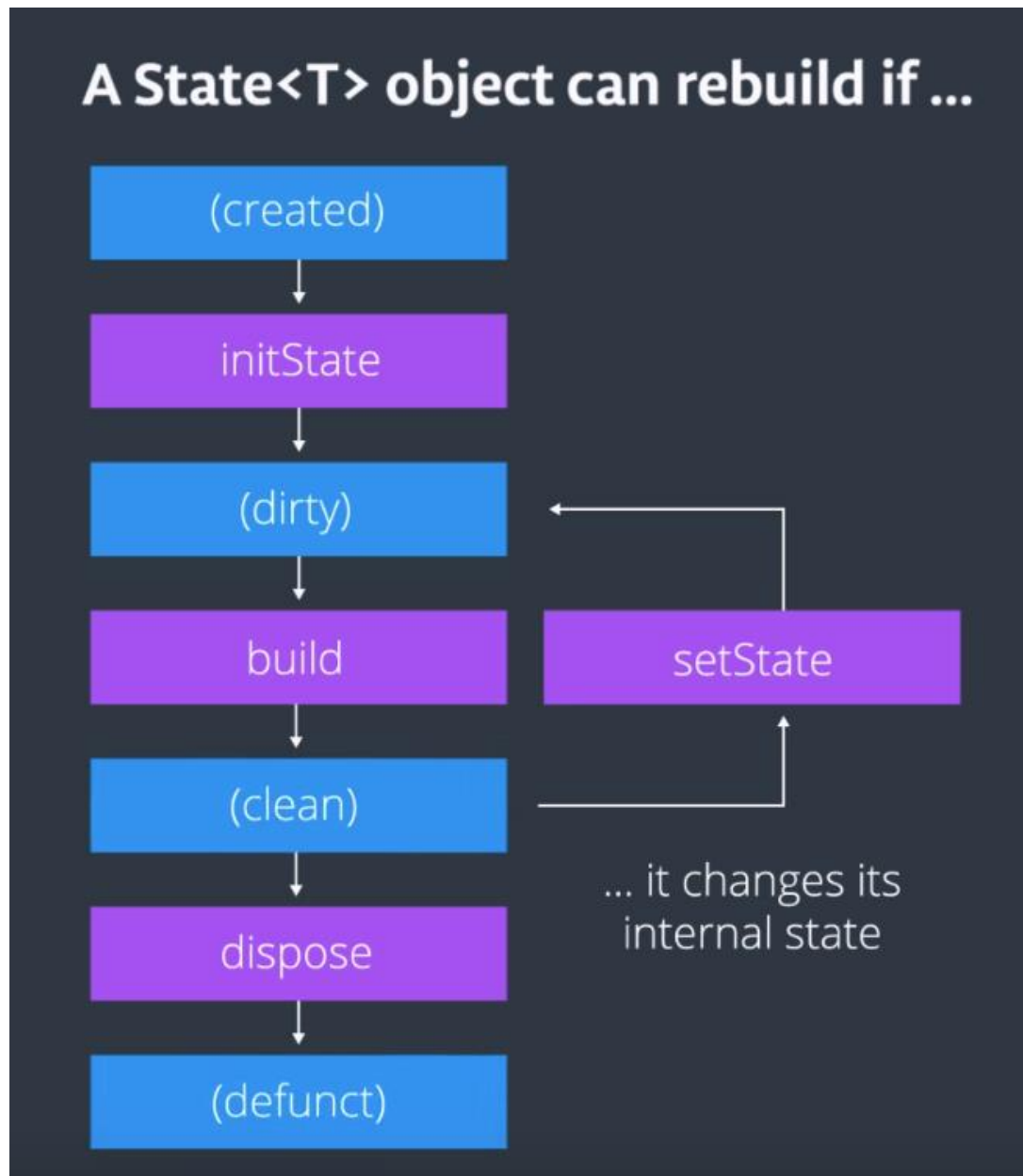
### 2.2.7.2 State

**Flutter Widget** quản lý các **State** (trạng thái) thông qua widget đặc biệt tên **StatefulWidget**:

- StatefulWidget là widget mà trạng thái của chúng có thể thay đổi. Để tạo state thì StatefulWidget phải gọi method createState();
- State là nơi chứa logic và trạng thái của StatefulWidget, nó chứa một số dữ liệu có thể thay đổi trong suốt thời gian tồn tại của widget, và phương thức build() để vẽ lại widget khi có sự thay đổi
- Khi có sự thay đổi trạng thái (state changes), state object sẽ gọi đến setState() nghĩa là trạng thái của widget đã thay đổi -> build widget

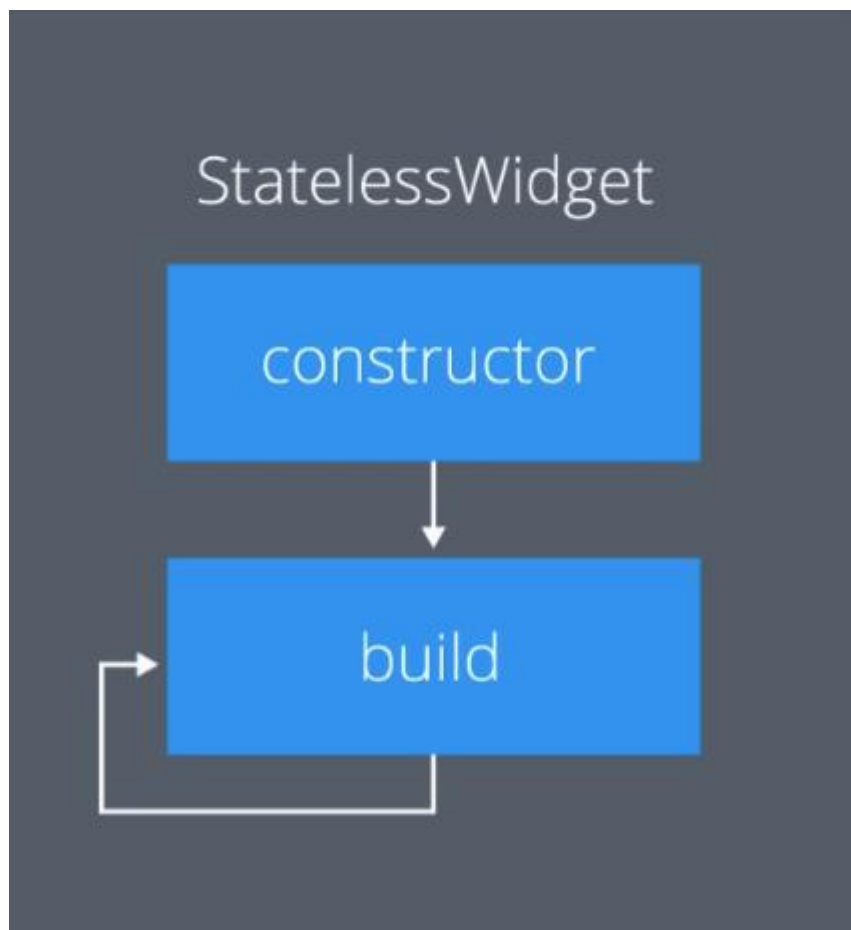
---

<sup>10</sup> <https://vncoder.vn/bai-hoc/kien-truc-ung-dung-flutter-215>



**Hình 2.3 Mô hình vòng đời của State Object**

**StatelessWidgets** là các widget không chứa State. Tất cả các giá trị của StatelessWidget đều là final -> không thể thay đổi trong thời gian chạy, do đó StatelessWidget chỉ hiển thị những gì được truyền vào trong constructor. Ngược lại, để thay đổi trạng thái của widget ta dùng **StatefulWidget**



**Hình 2.4** Mô hình hoạt động của StatelessWidget

## CHƯƠNG 3. PHƯƠNG PHÁP THỰC HIỆN

### 3.1 Giải pháp phân tích và thiết kế các mô hình

#### 3.1.1 Mô hình Use Case

##### 3.1.1.1 Danh sách Actor

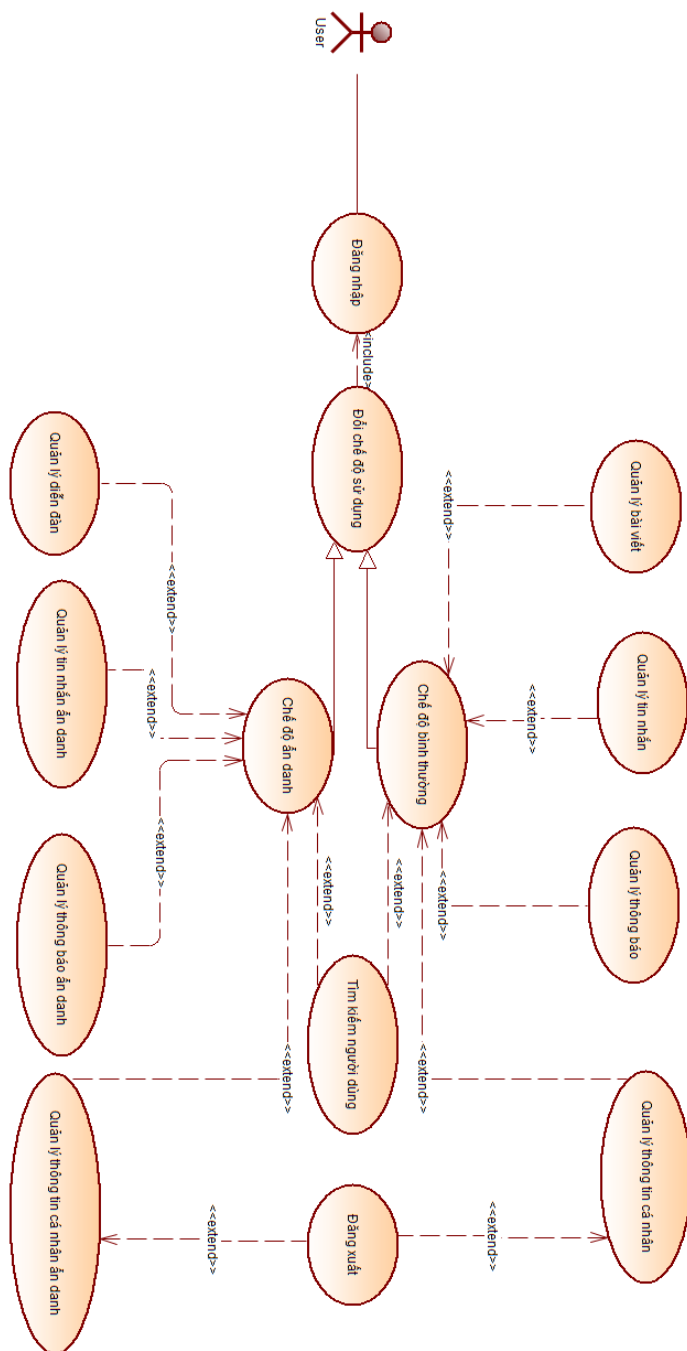
Stt	Actor	Ghi chú
1	Admin	Có quyền cao nhất trong hệ thống, quản lý người dùng và nội dung người dùng đăng tải
2	User	Người sử dụng, có quyền đăng bài, thay đổi thông tin cá nhân, đổi chế độ sử dụng

**Bảng 3.1** Bảng danh sách Actor

##### 3.1.1.2 User

User sẽ có những chức năng chính như sau:

- Đăng nhập ứng dụng
- Đổi chế độ sử dụng sang bình thường:
- Quản lý bài viết
- Quản lý tin nhắn
- Quản lý quản lý thông báo
- Quản lý thông tin cá nhân
- Tìm kiếm người dùng
- Đổi chế độ sử dụng sang ẩn danh:
- Quản lý diễn đàn
- Quản lý thông báo (ẩn danh)
- Quản lý thông tin cá nhân (ẩn danh)
- Tìm kiếm người dùng (ẩn danh)



Hình 3.1 Mô hình Use Case của User mức tổng quát



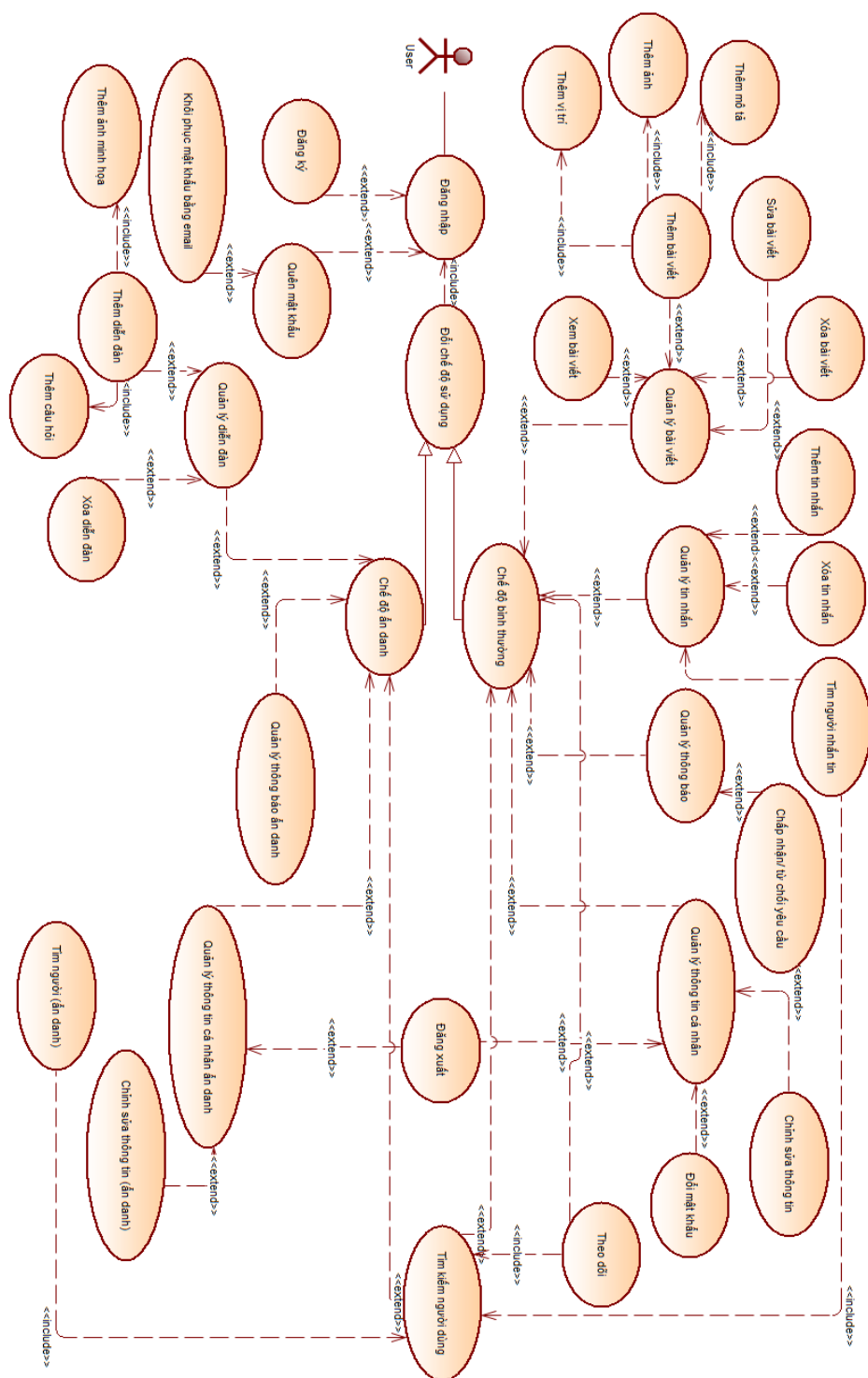
**3.1.1.2.1 Đặc tả**

## 3.1.1.2.1.1 Danh sách Use-case

Stt	Use-case	Mô tả
1	Đăng nhập	Đăng nhập ứng dụng
2	Đổi chế độ sử dụng	Chuyển đổi giữa 2 chế độ bình thường và ẩn danh
3	Chế độ bình thường	Chế độ mặc định của người dùng, ở chế độ này người sử dụng có thể sử dụng các mục như quản lý bài viết, tin nhắn, thông báo, thông tin cá nhân công khai của mình
4	Quản lý bài viết	Thêm, sửa, xóa bài viết của bản thân và xem các bài viết được đăng từ người đã theo dõi
5	Quản lý tin nhắn	Xem, thêm, sửa, xóa danh sách bạn bè mà ta muốn nhắn tin
6	Quản lý thông báo	Xem, xóa thông báo từ bài viết, người theo dõi
7	Quản lý thông tin cá nhân	Xem, sửa, xóa thông tin tài khoản cá nhân
8	Chế độ ẩn danh	Chế độ khi người dùng bấm vào nút ẩn danh, ở chế độ này người dùng sẽ được ẩn hết thông tin cá nhân, có thể sử dụng các mục như quản lý bài viết, tin nhắn, thông báo, thông tin cá nhân ẩn danh của mình
9	Quản lý diễn đàn	Thêm, sửa, xóa diễn đàn của bản thân và xem các bài viết được đăng từ người đã theo dõi
10	Quản lý tin nhắn (ẩn danh)	Xem, sửa, xóa danh sách bạn bè mà ta muốn nhắn tin ở mức ẩn

		danh, cả 2 đều không biết thông tin thật của nhau khi nhắn
11	Quản lý thông báo (ẩn danh)	Xem, xóa thông báo từ bài viết, người theo dõi
12	Quản lý thông tin cá nhân (ẩn danh)	Xem, sửa, xóa thông tin tài khoản (ẩn danh)
13	Tìm kiếm người dùng	Tìm kiếm thông tin của người dùng

**Bảng 3.2 Bảng danh sách Use-Case của User**

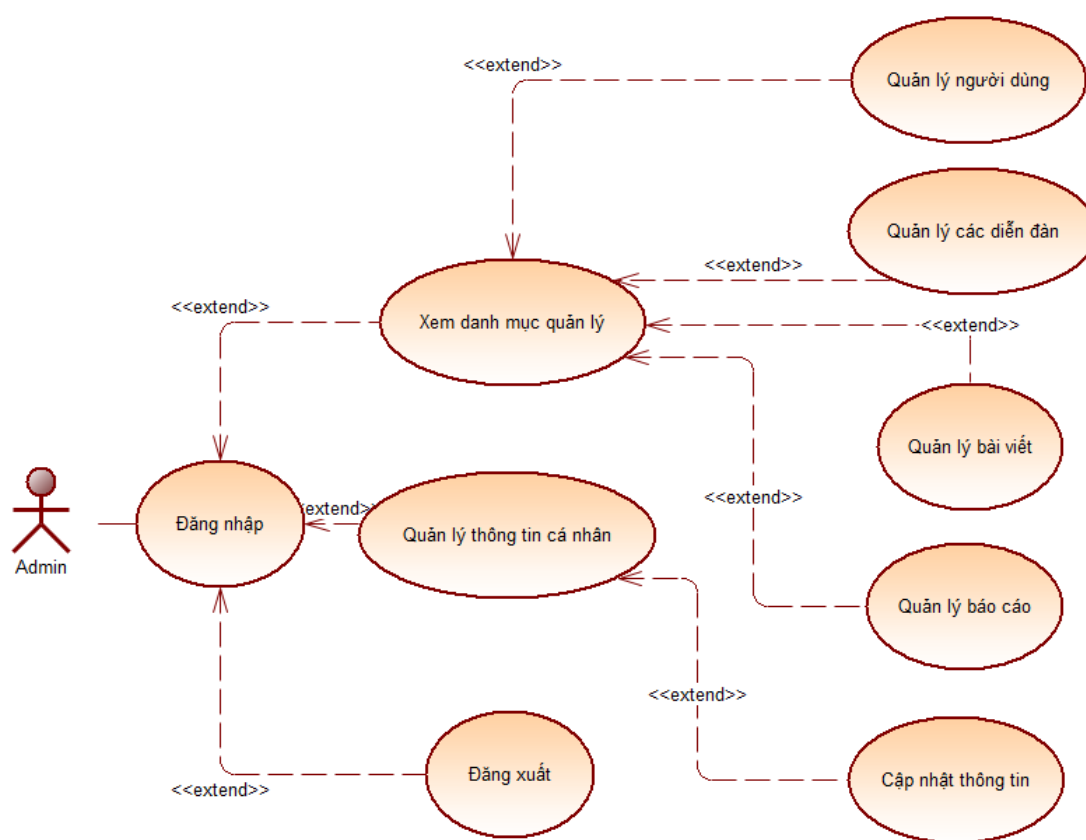


Hình 3.2 Mô hình Use-Case của User mức phân rã

### 3.1.1.3 Admin

Admin sẽ có những chức năng chính như sau:

- Đăng nhập ứng dụng
- Xem danh mục quản lý
- Quản lý thông tin cá nhân
- Đăng xuất

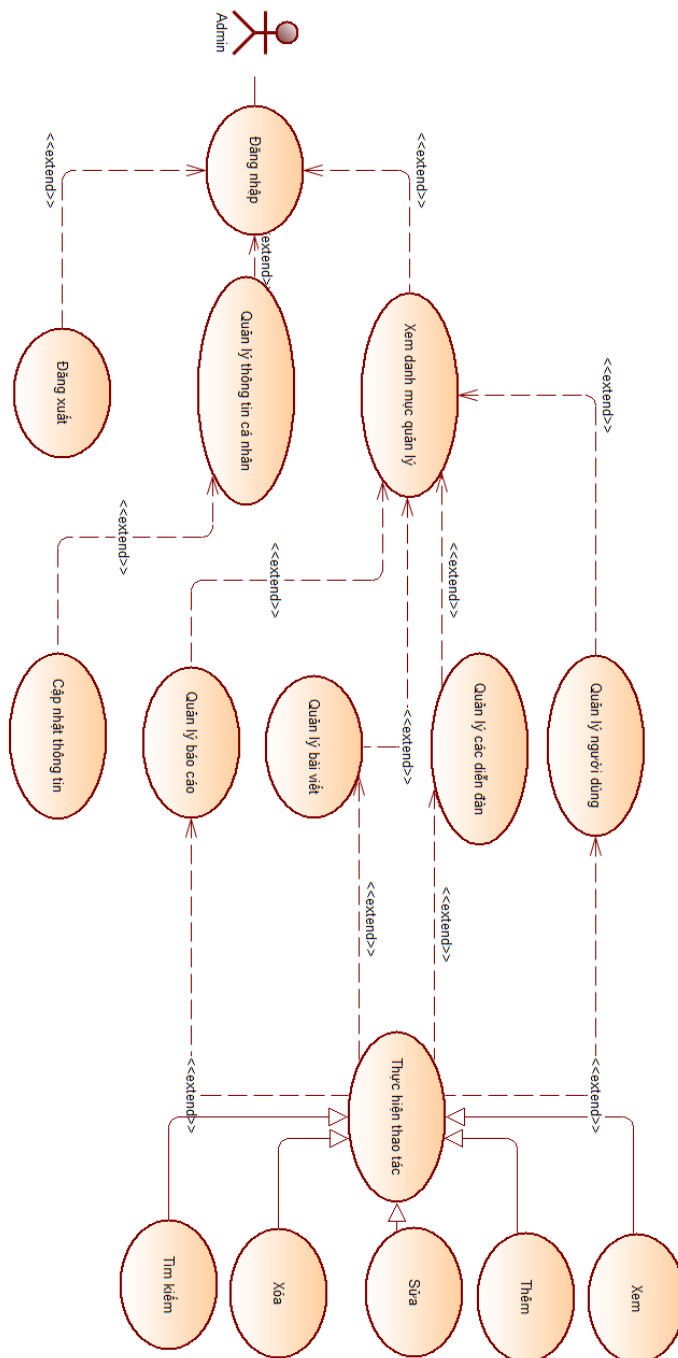


**Hình 3.3 Mô hình Use-Case của Admin mức tổng quát**

**3.1.1.3.1 Danh sách Use-case**

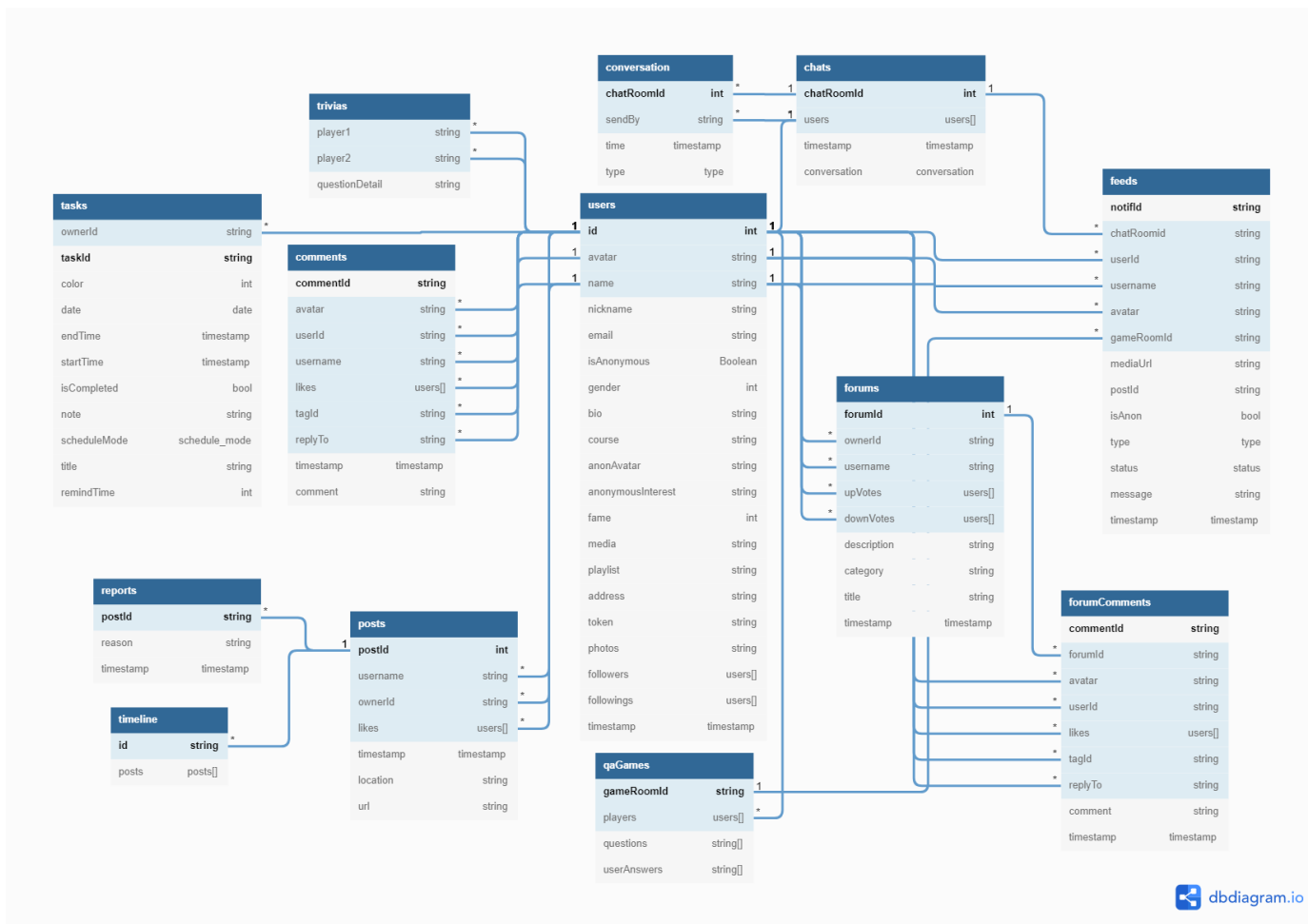
<b>Stt</b>	<b>Use-case</b>	<b>Mô tả</b>
1	Đăng nhập	Đăng nhập ứng dụng
2	Quản lý thông tin cá nhân	Quản lý thông tin tài khoản cá nhân và các cài đặt
3	Cập nhật thông tin	Cập nhật lại thông tin cá nhân của admin
3	Quản lý diễn đàn	Thêm, sửa, xóa diễn đàn của bản thân và xem các bài viết được đăng từ người đã theo dõi
4	Quản lý người dùng	Xem, sửa, xóa người dùng trong hệ thống
5	Quản lý bài viết	Xem, xóa bài viết
6	Quản lý báo cáo	Xem, sửa, xóa các báo cáo từ người dùng

**Bảng 3.3 Bảng danh sách User-Case của Admin**



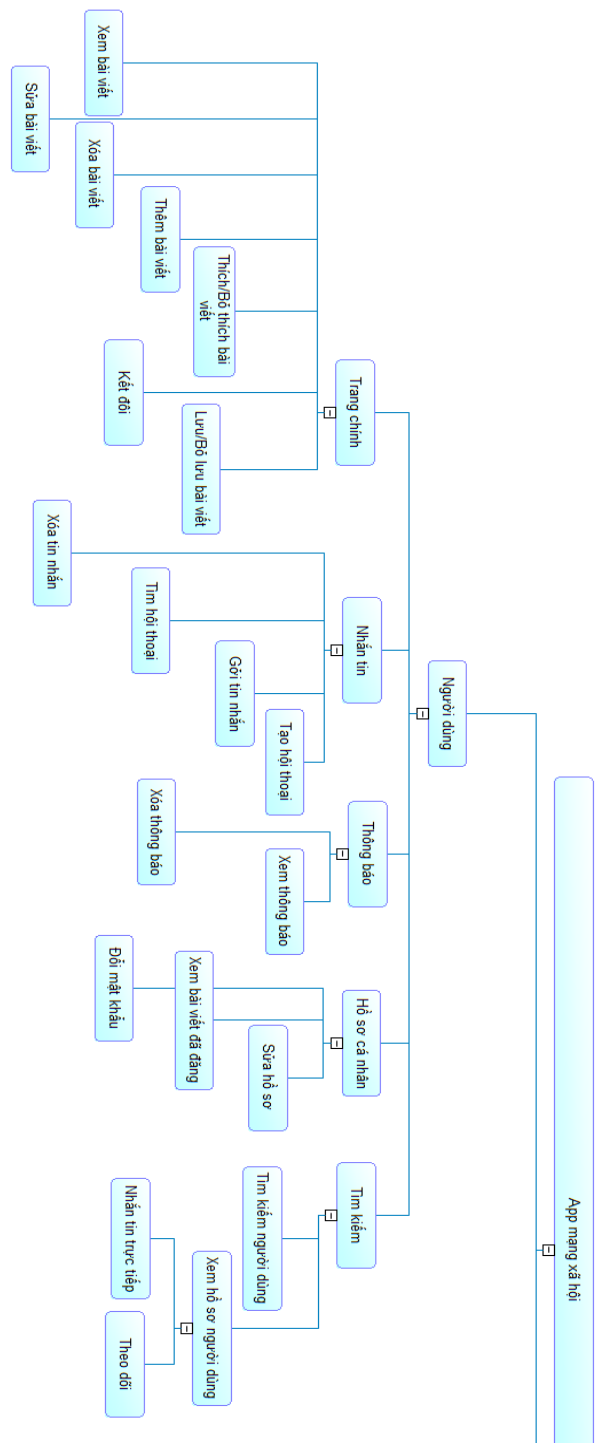
Hình 3.4 Mô hình Use Case của Admin mức phân rã

### 3.1.2 Sơ đồ ER



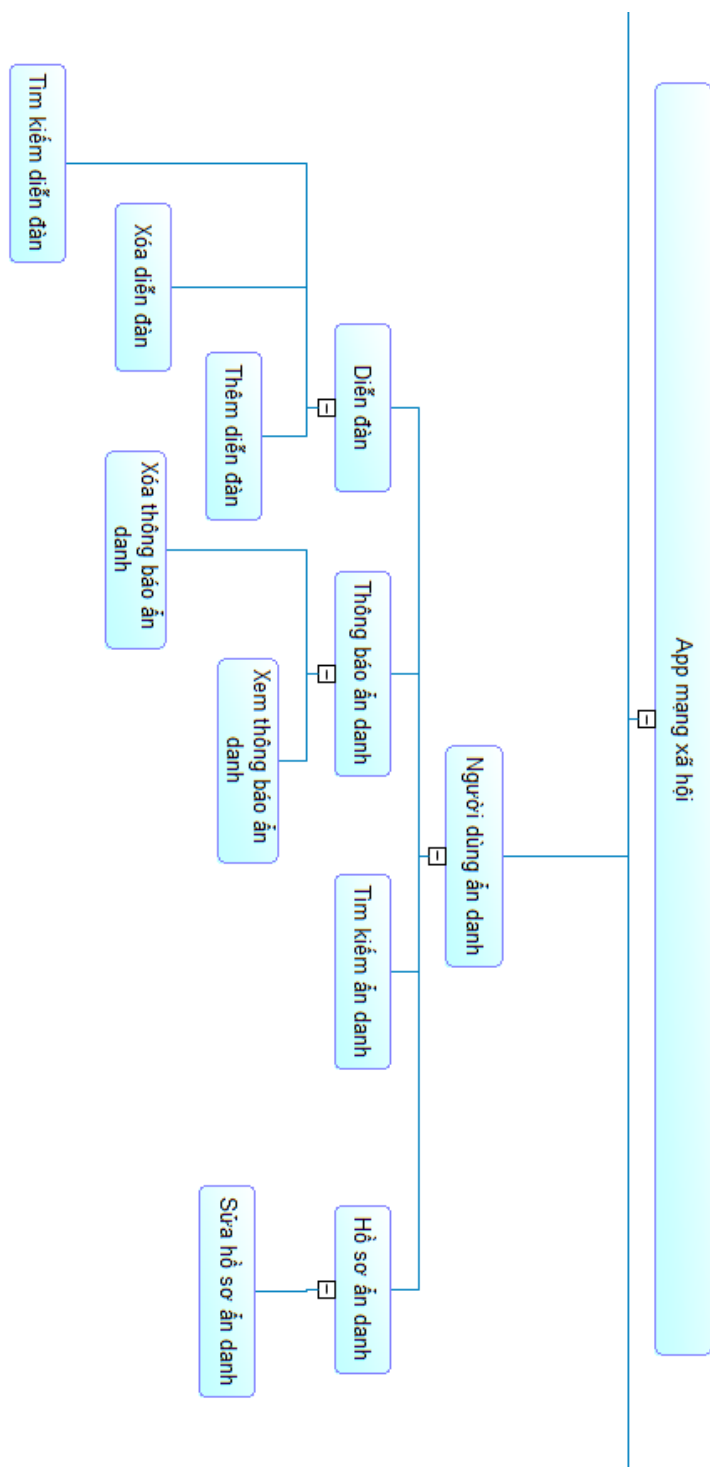
Hình 3.5 Hình sơ đồ ER

3.1.3 Sơ đồ BFD

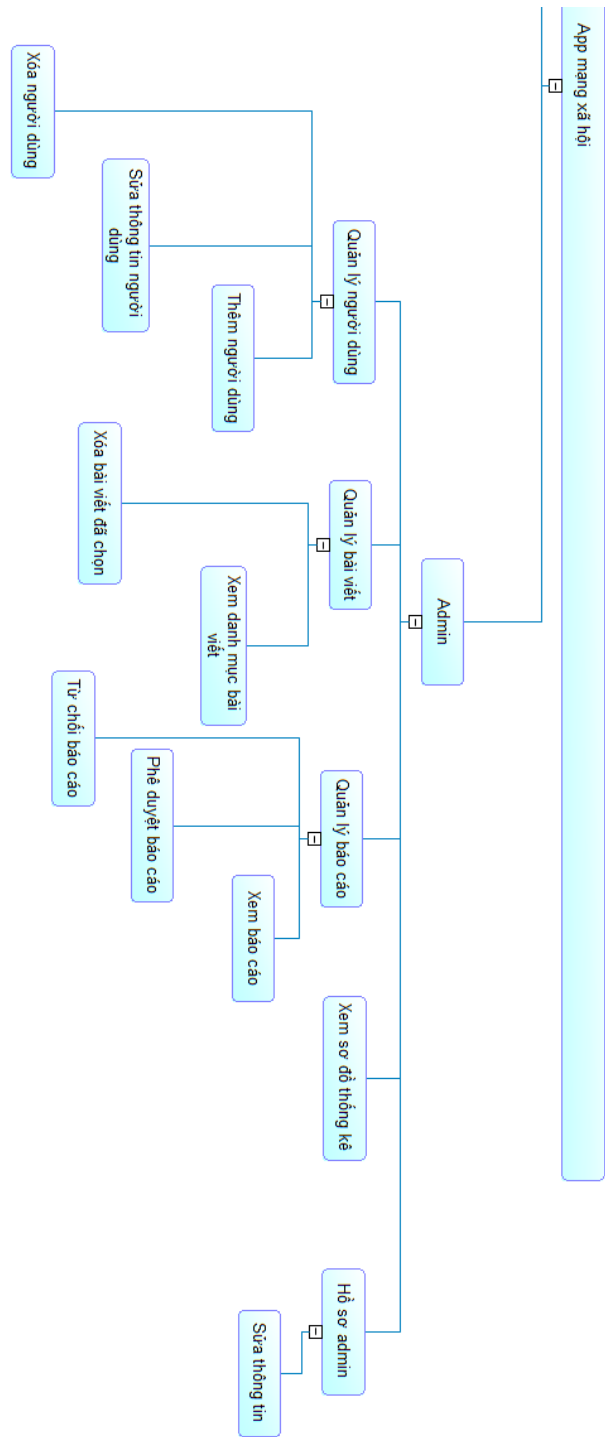


Hình 3.6 Sơ đồ BFD của người dùng thường



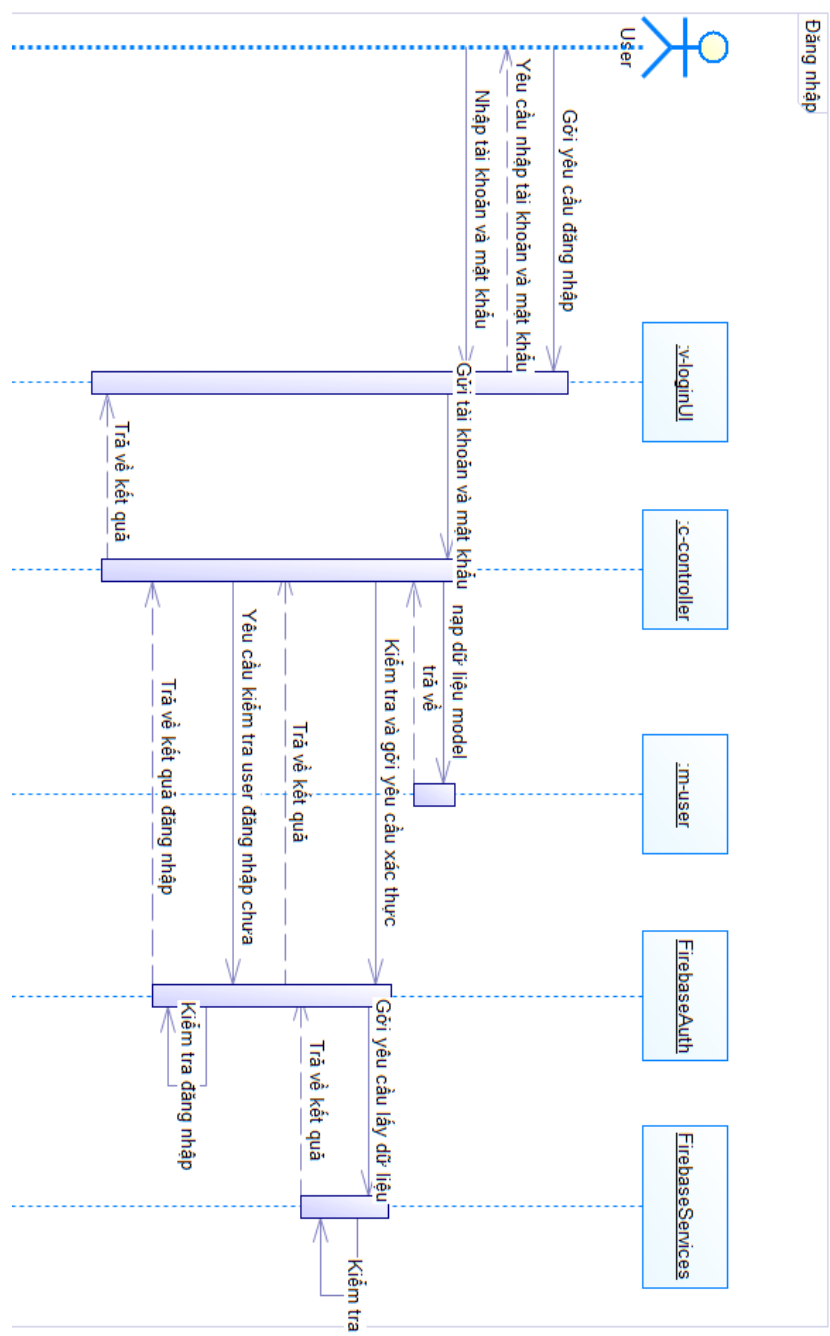


Hình 3.7 Sơ đồ BFD của người dùng ấn danh

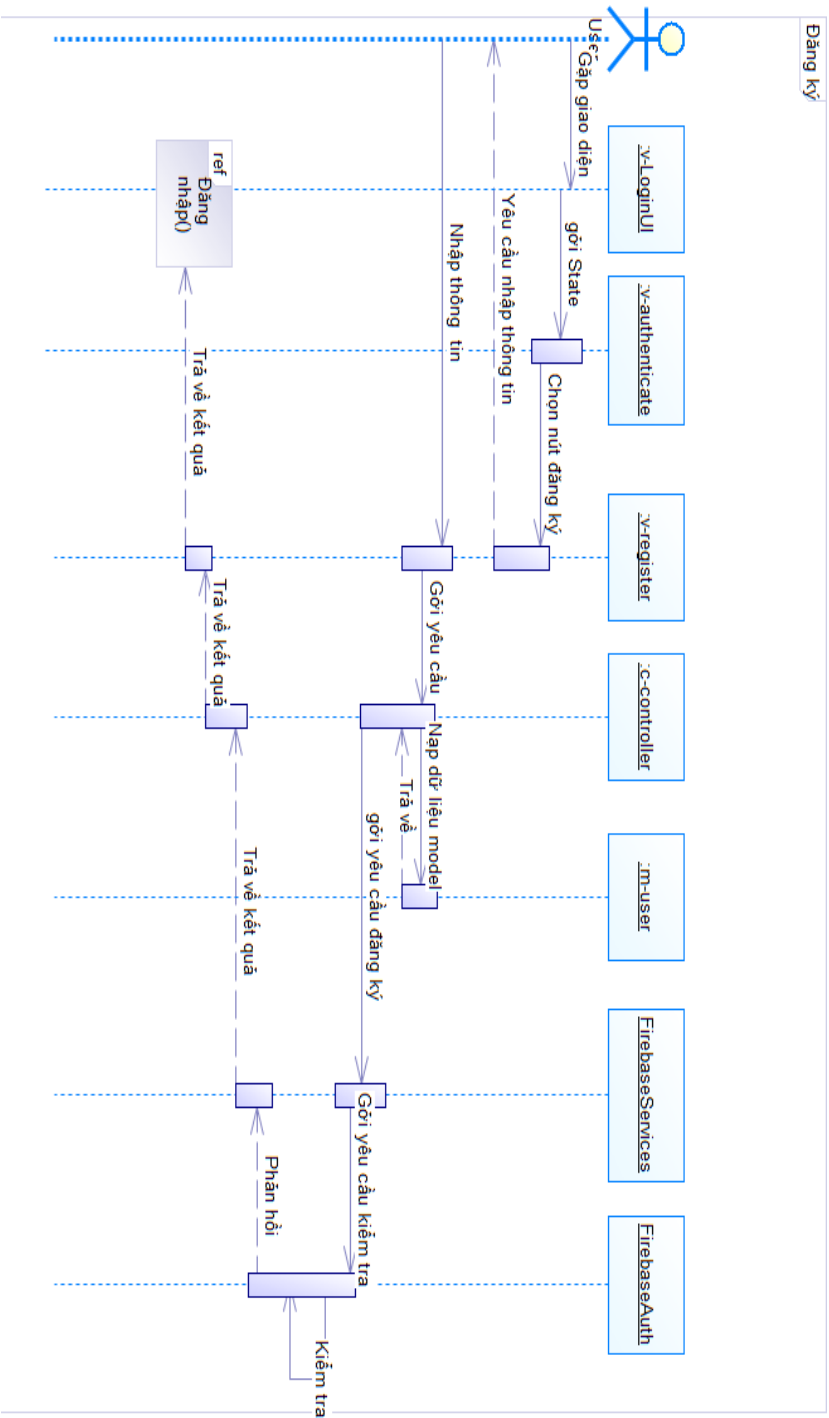


Hình 3.8 Sơ đồ BFD của admin

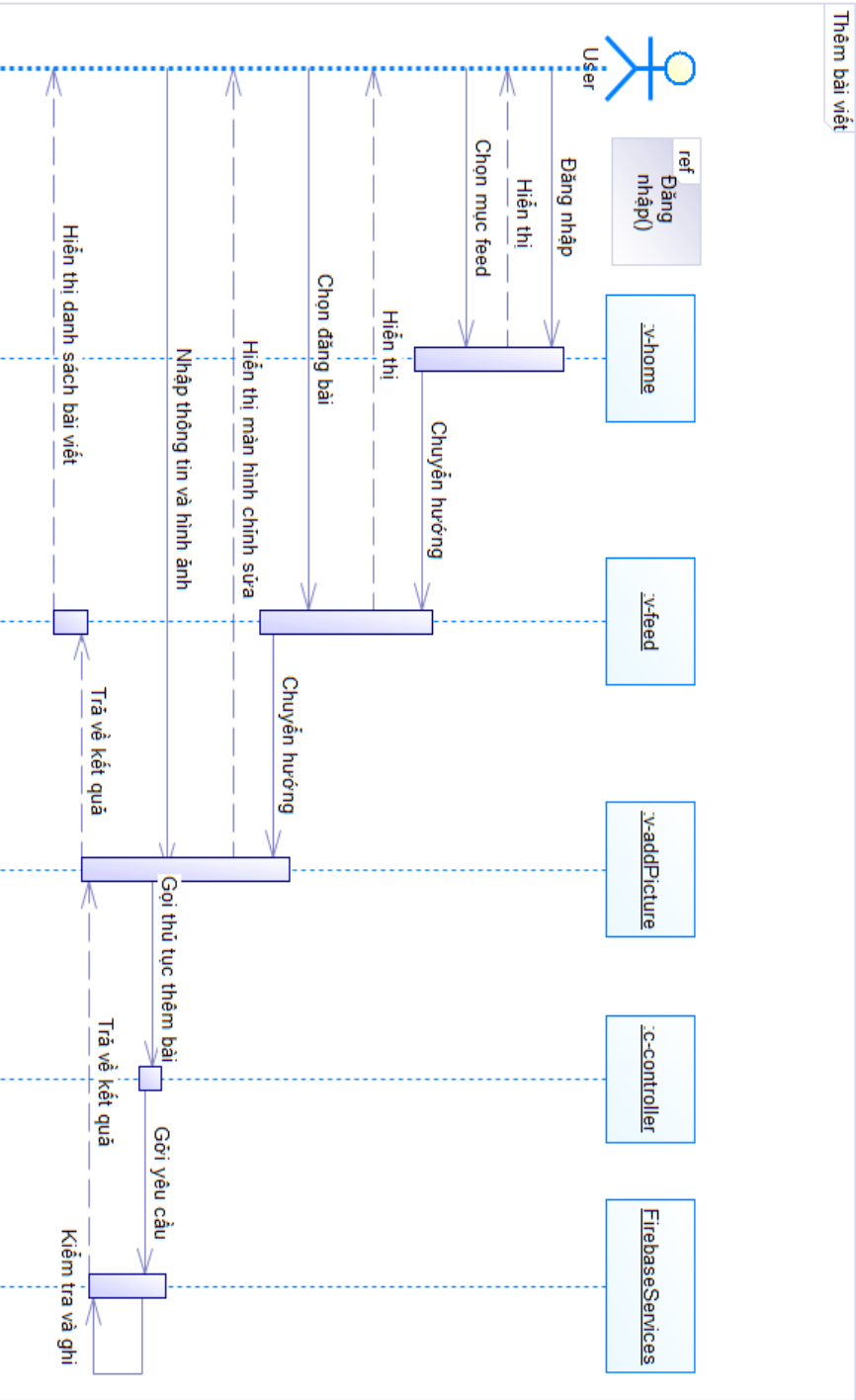
3.1.4 Sơ đồ tuần tự



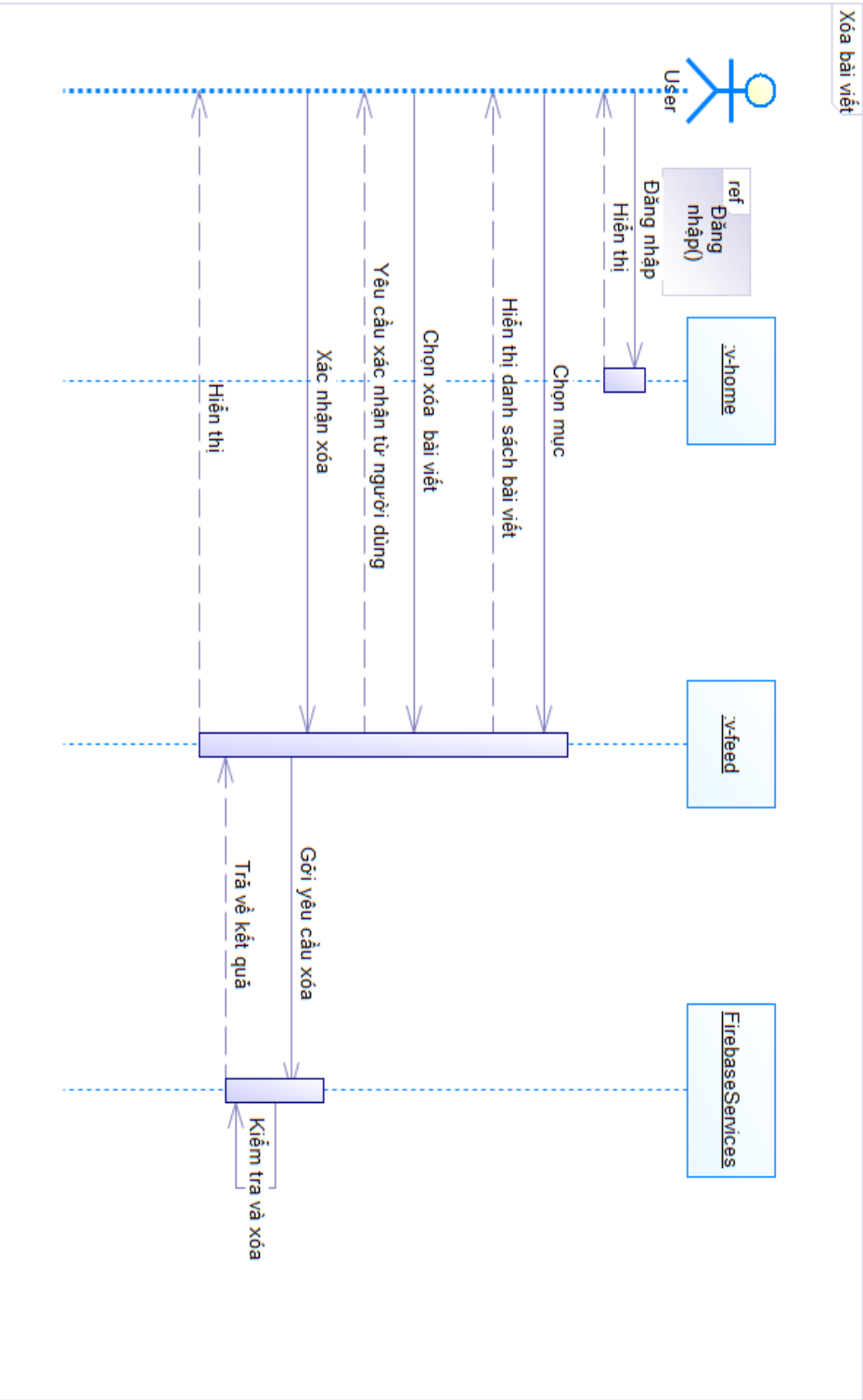
Hình 3.9 Sơ đồ tuần tự đăng nhập



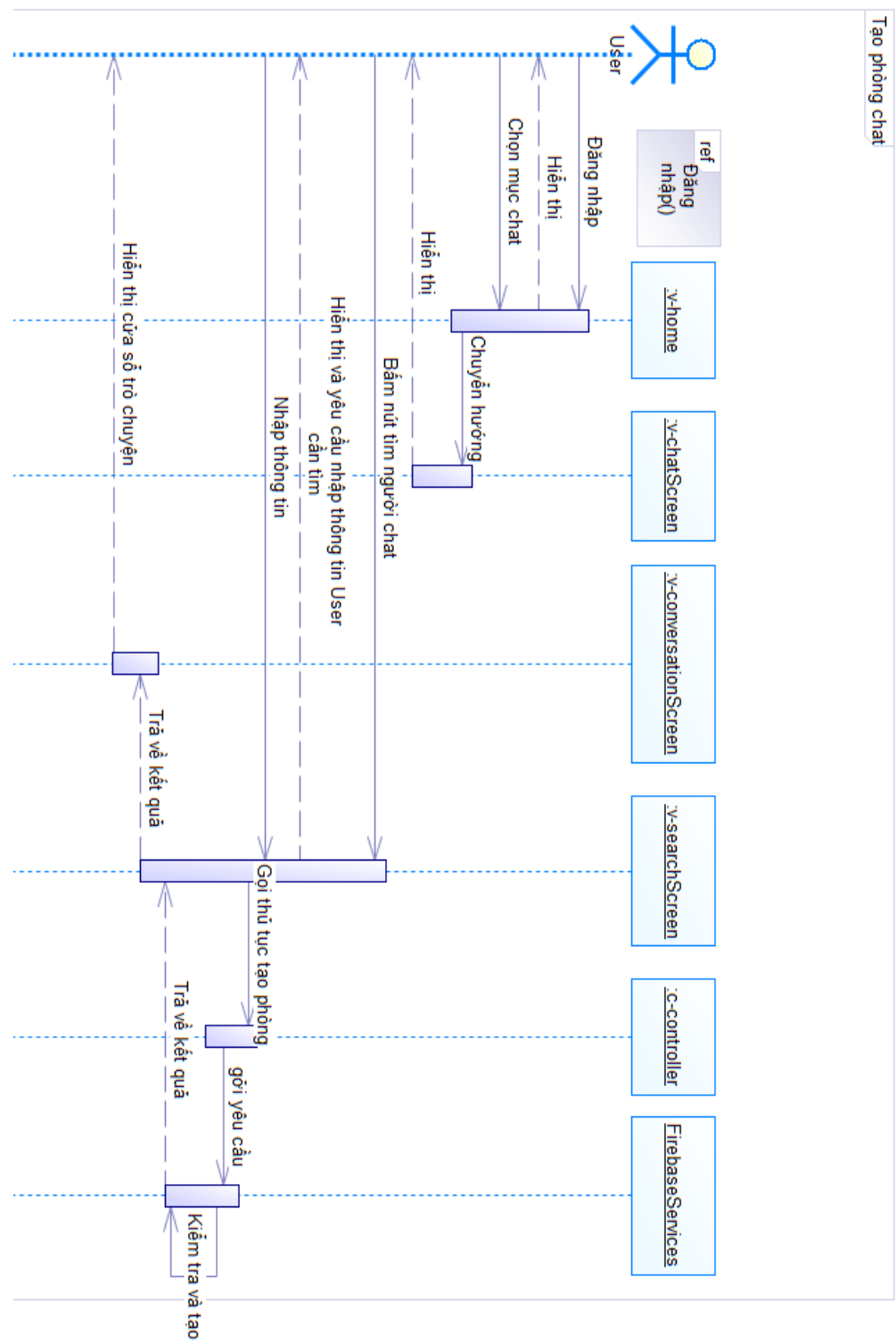
Hình 3.10 Sơ đồ tuần tự đăng ký



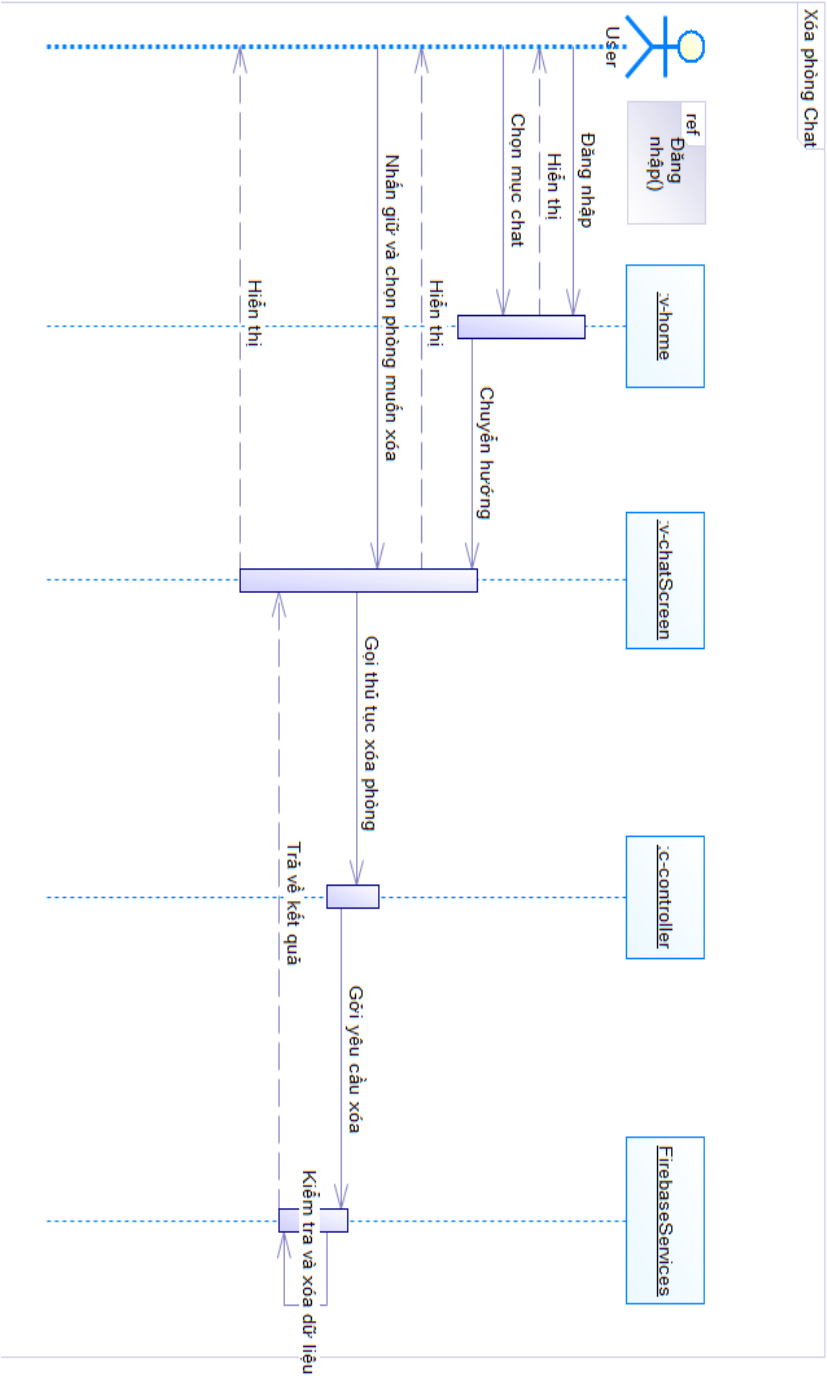
Hình 3.11 Sơ đồ tuần tự thêm bài viết



Hình 3.12 Sơ đồ tuần tự xóa bài viết

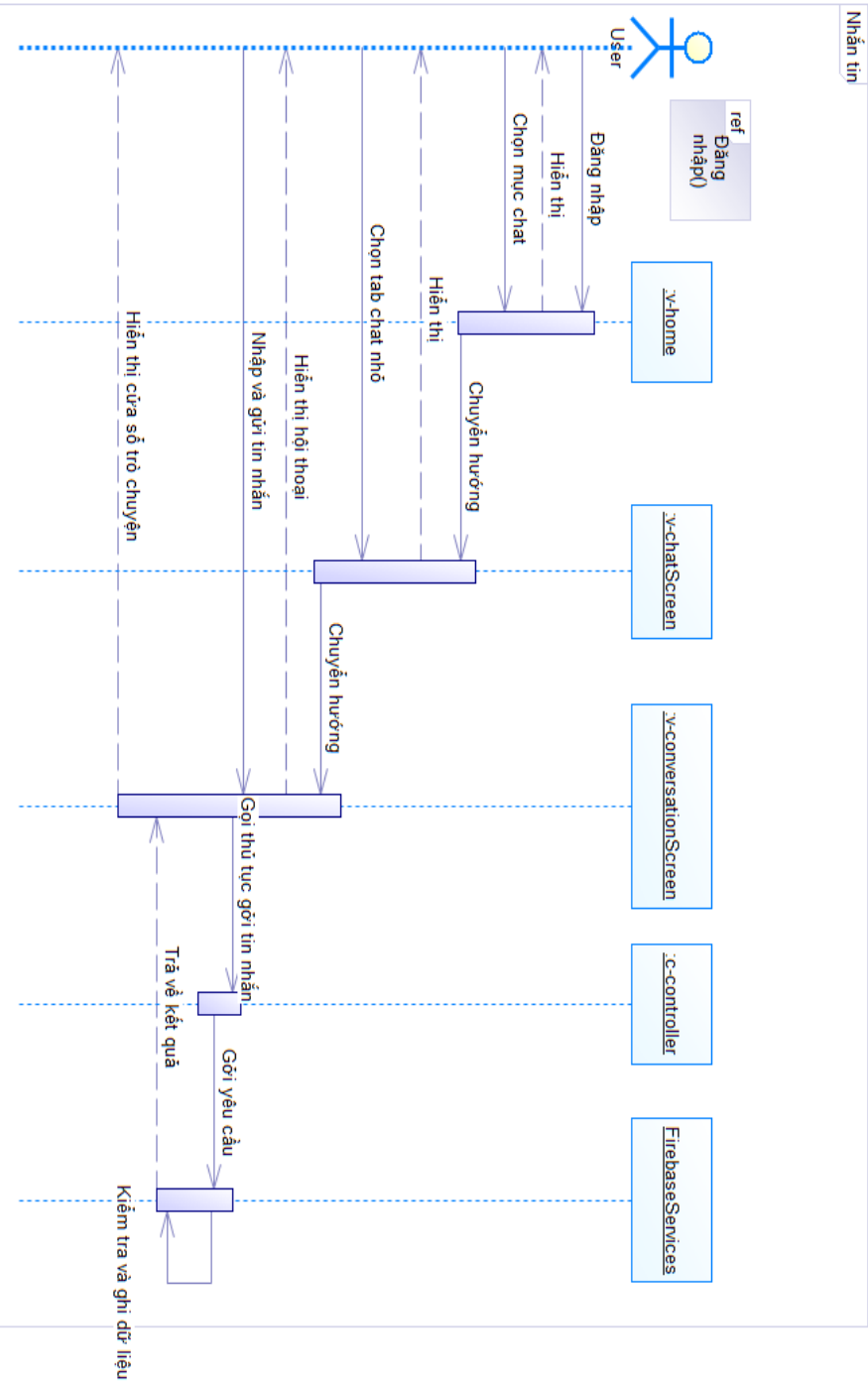


Hình 3.13 Sơ đồ tuần tự tạo phòng chat

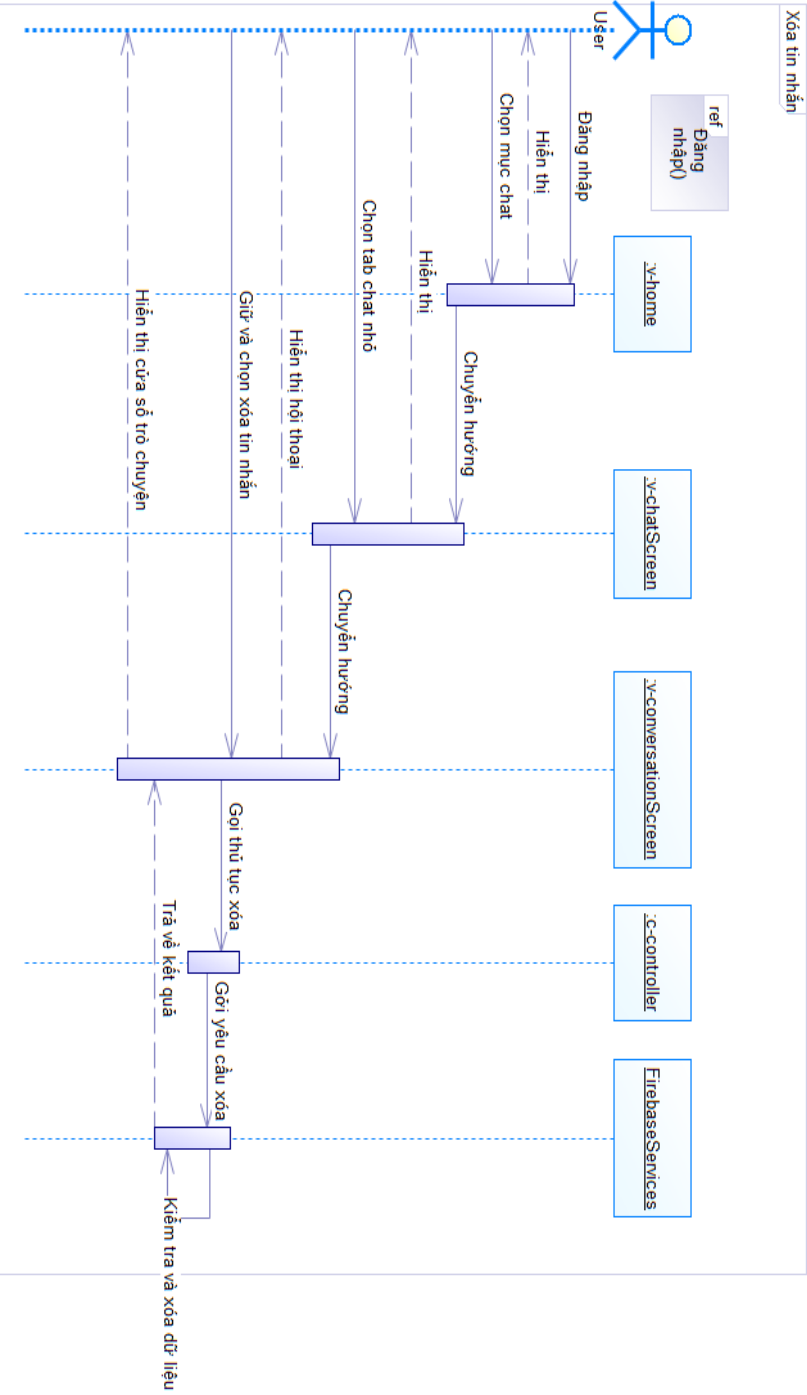


Hình 3.14 Sơ đồ tuần tự xóa phòng chat

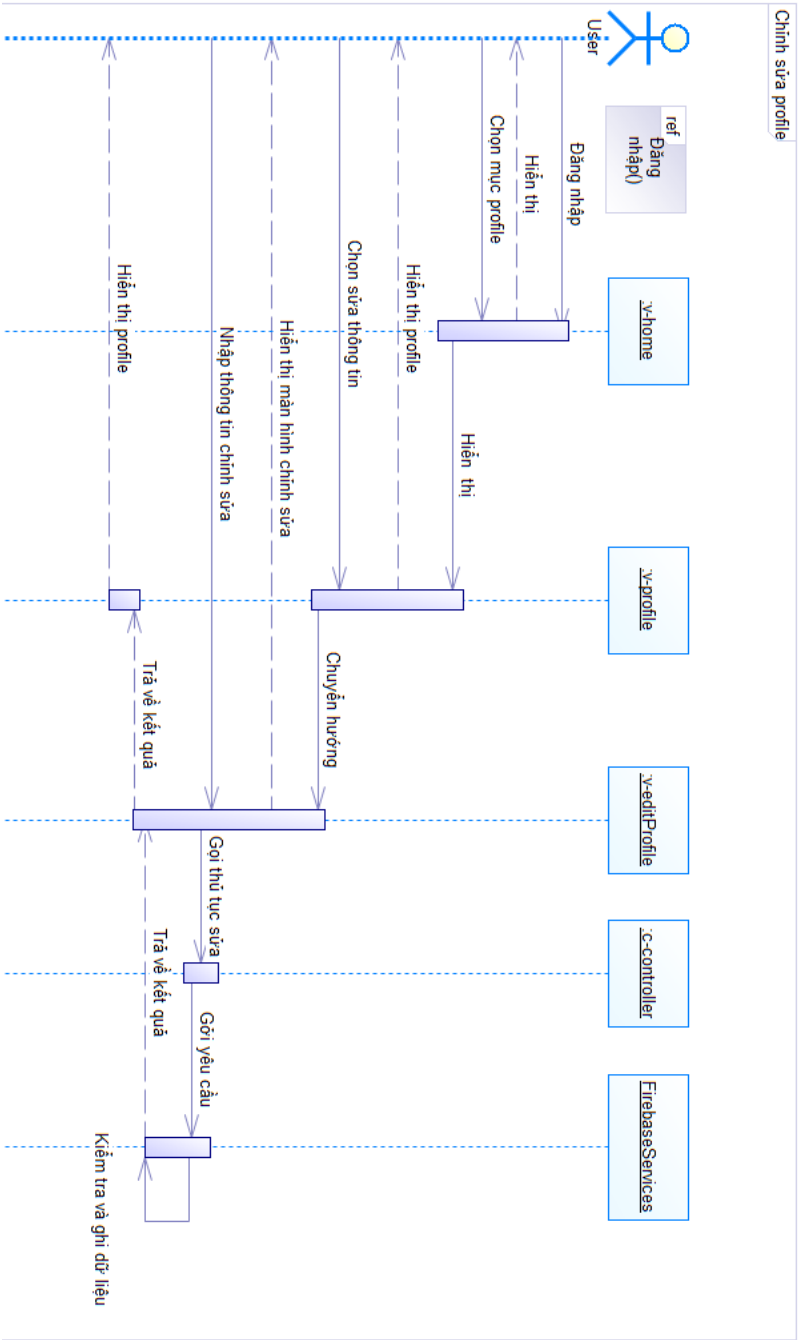




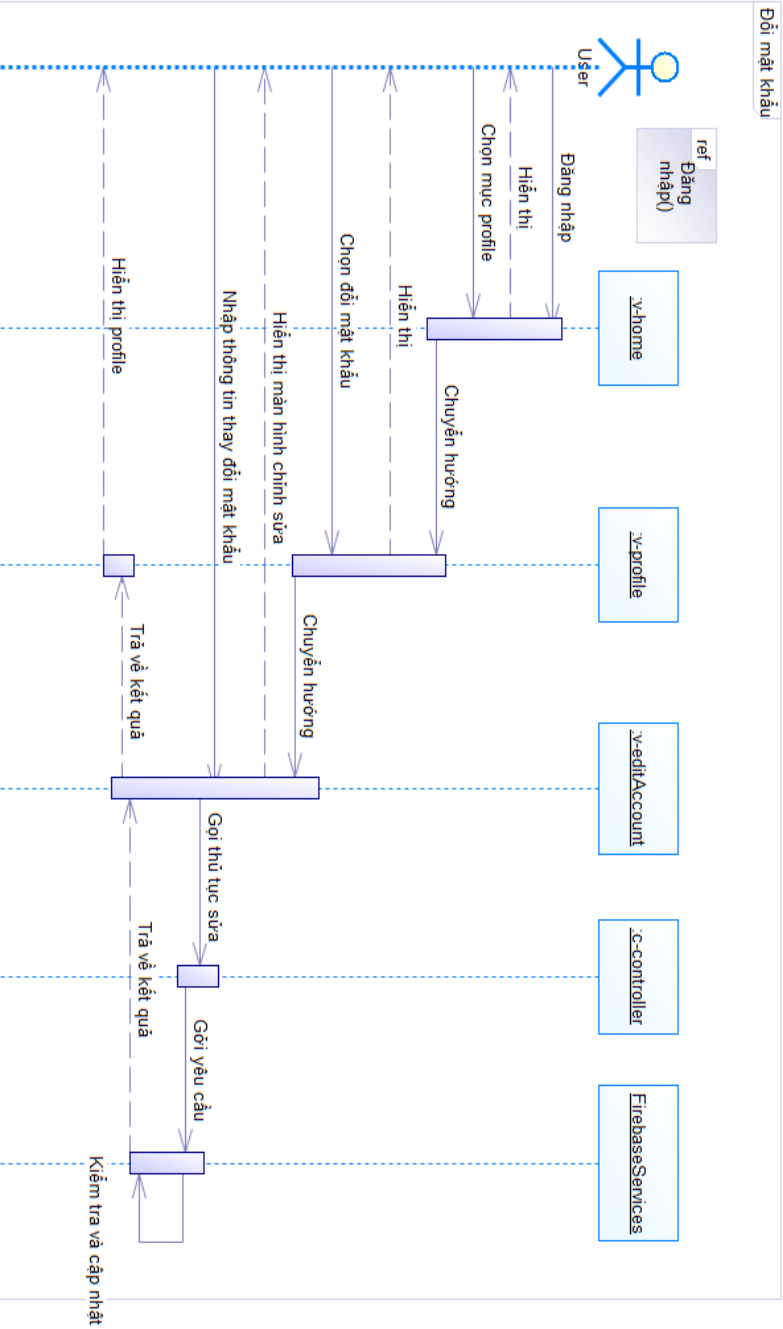
Hình 3.15 Sơ đồ tuần tự nhắn tin



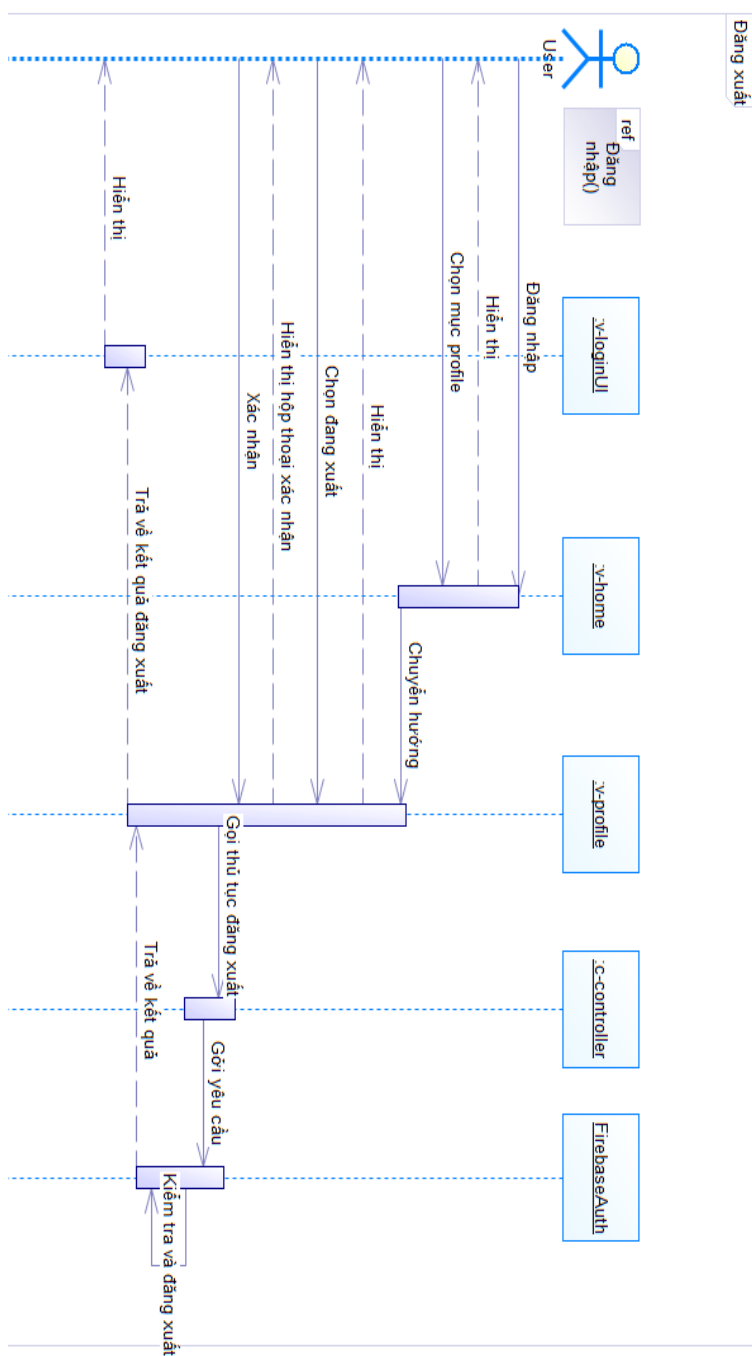
Hình 3.16 Sơ đồ tuần tự xóa tin nhắn



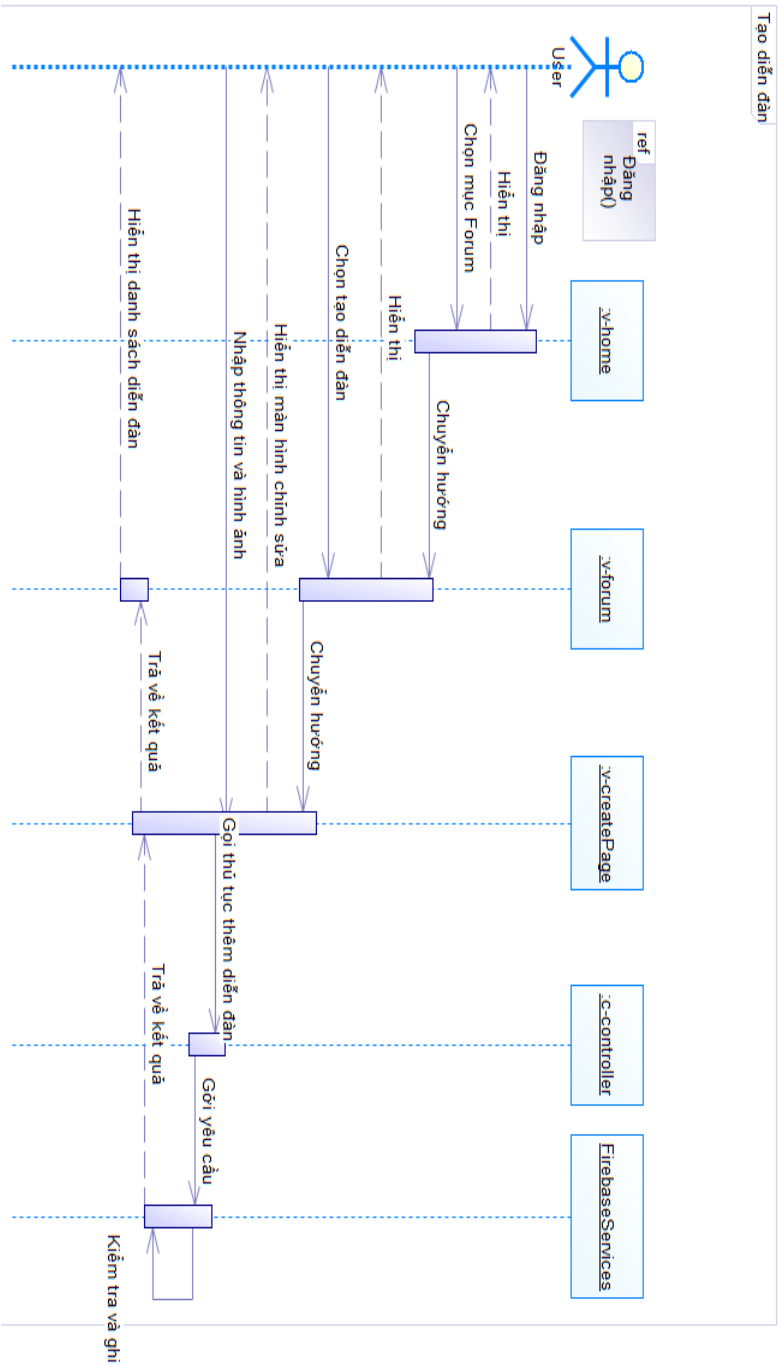
Hình 3.17 Sơ đồ tuần tự chỉnh sửa profile



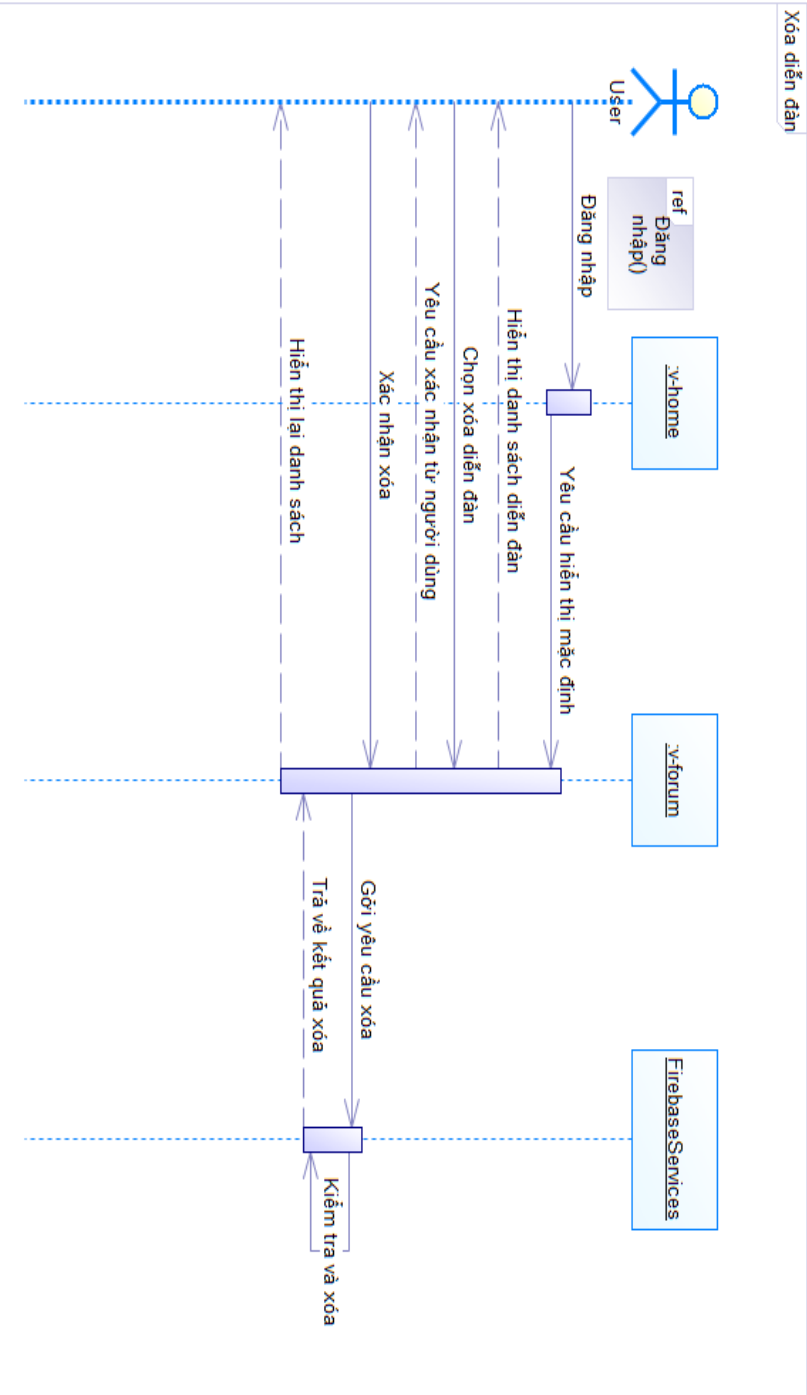
Hình 3.18 Sơ đồ tuần tự đổi mật khẩu



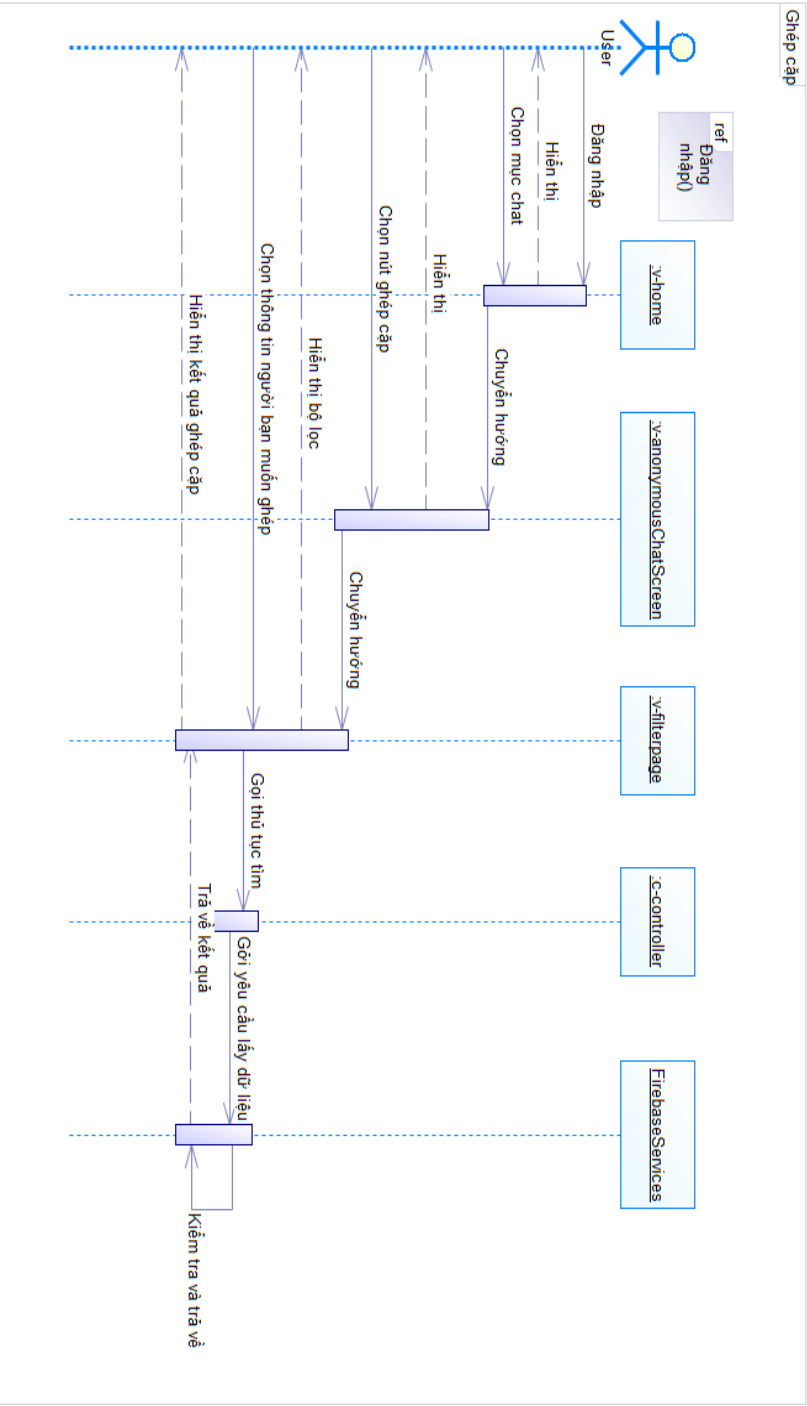
### Hình 3.19 Sơ đồ tuần tự đăng xuất



Hình 3.20 Sơ đồ tuần tự tạo diễn đàn

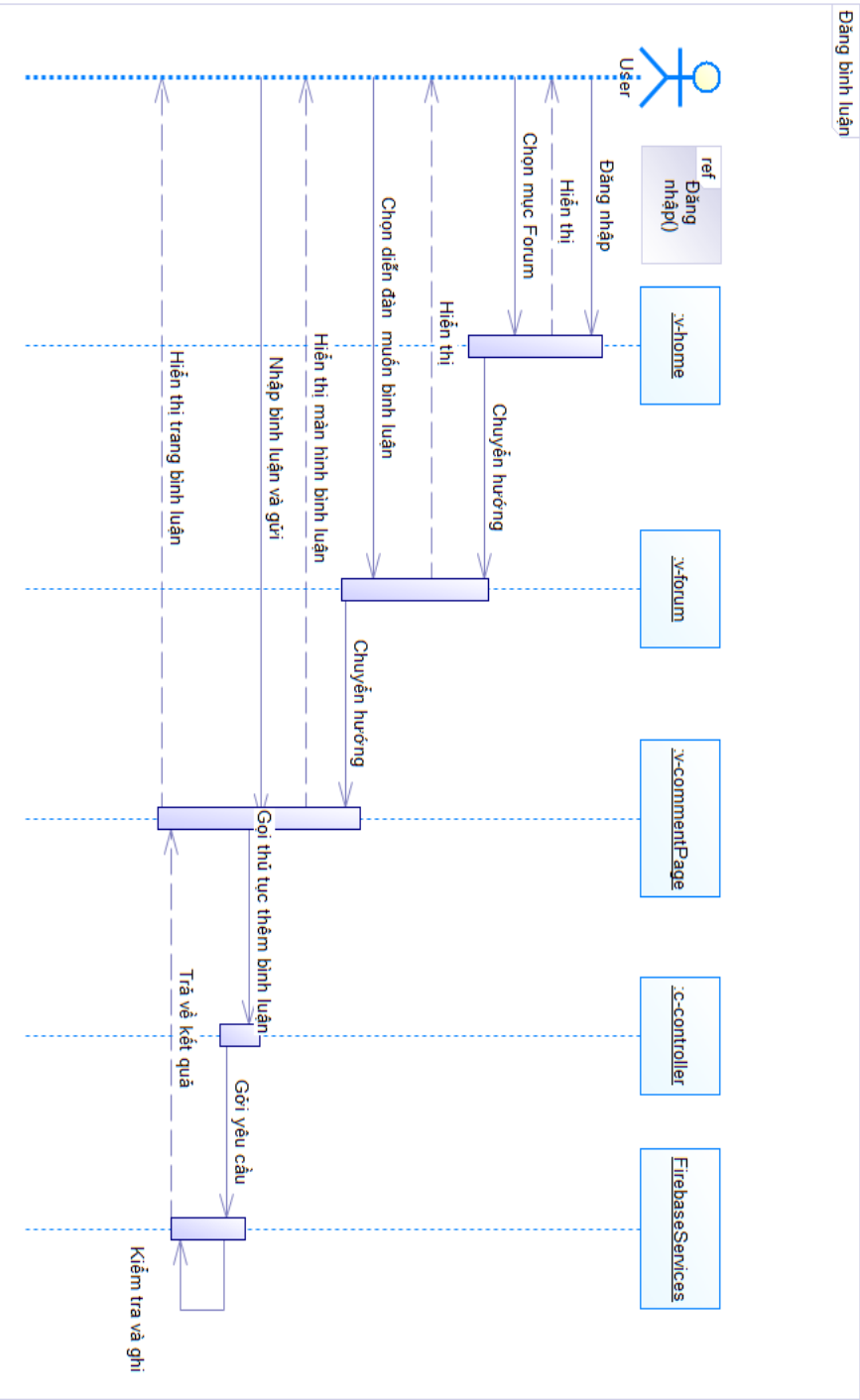


Hình 3.21 Sơ đồ tuần tự xóa diễn đàn

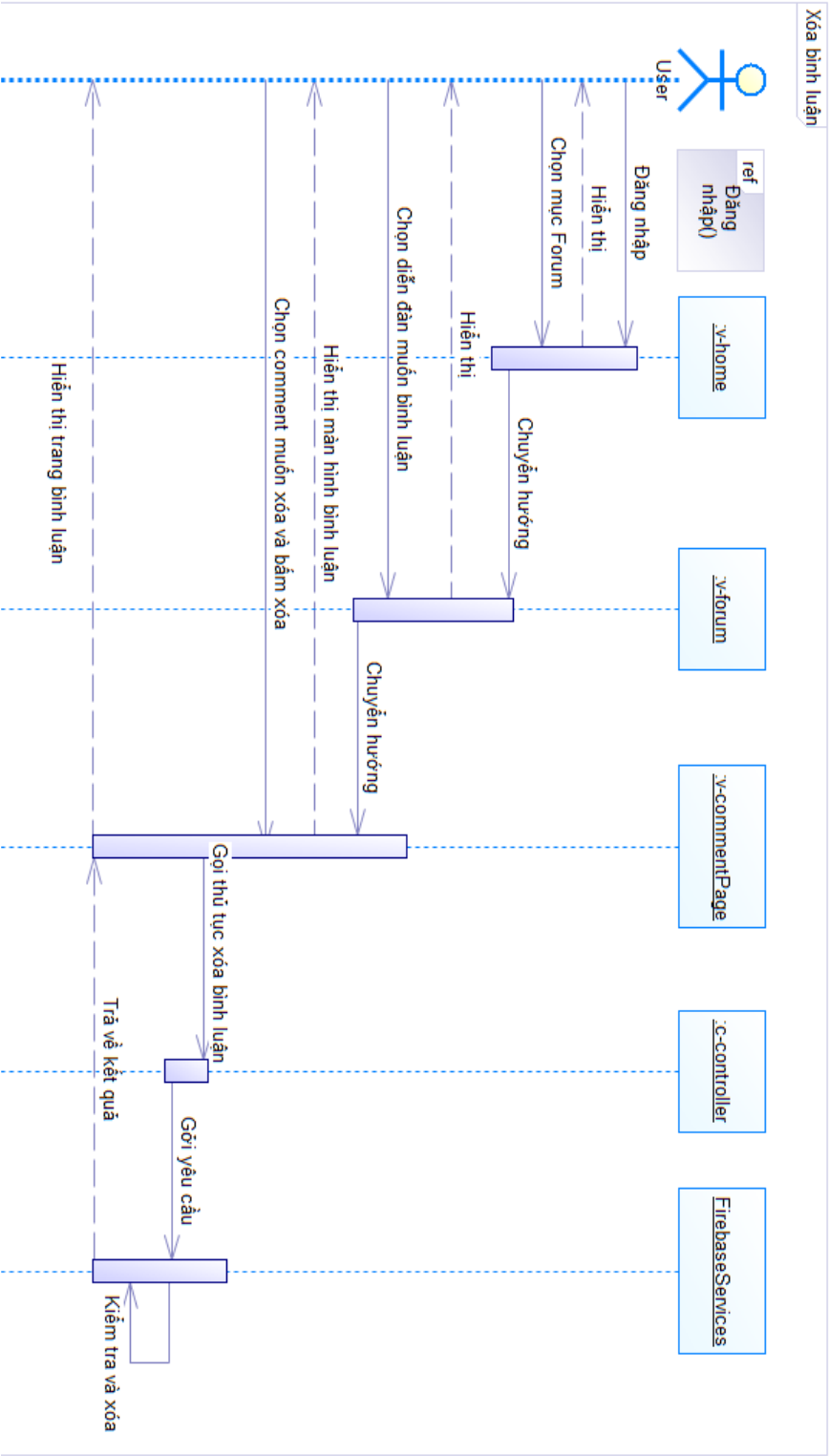


Hình 3.22 Sơ đồ tuần tự ghép cặp

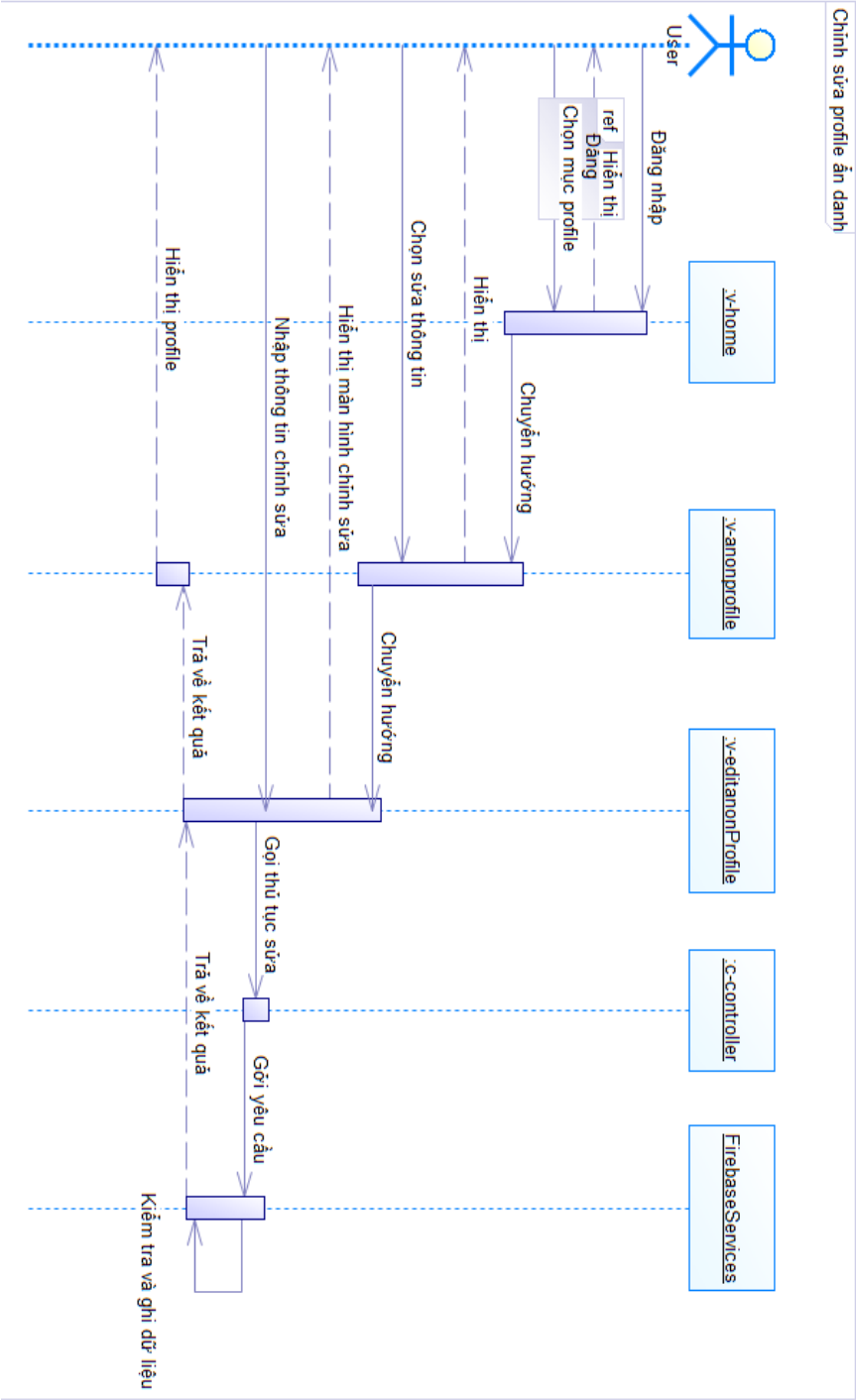




Hình 3.23 Tuần tự đăng bình luận



Hình 3.24 Sơ đồ tuần tự xóa bình luận



Hình 3.25 Sơ đồ tuần tự sửa profile ẩn danh

### 3.2 Thiết kế cơ sở dữ liệu

Khái quát mô hình tổng quát của mỗi Collection bao gồm: Field (thuộc tính) của collection và collection con của nó.

STT	Tên thuộc tính (Field)	Kiểu dữ liệu	Mô tả
1	_id	String	Mỗi user có 1 mã số riêng biệt Kiểu dữ liệu: chữ
2	name	String	Họ và tên user Kiểu dữ liệu: chữ
3	nickname	String	Biệt danh của user Kiểu dữ liệu: chữ
4	email	String	Email của user Kiểu dữ liệu: chữ
5	isAnon	Boolean	User có phải người dùng ẩn danh hay không Kiểu dữ liệu: chữ
6	gender	Boolean	Giới tính của user Kiểu dữ liệu: chữ
7	bio	String	Tự giới thiệu bản thân Kiểu dữ liệu: chữ
8	course	String	Lĩnh vực học của user Kiểu dữ liệu: chữ
9	avatar	String	Ảnh đại diện Kiểu dữ liệu: chữ
10	anonBio	String	Tự giới thiệu của người dùng ẩn danh

			Kiểu dữ liệu: chữ
11	anonAvatar	String	Ảnh đại diện của người dùng ẩn danh Kiểu dữ liệu: chữ
12	anonInterest	String	Mục yêu thích của người dùng ẩn danh Kiểu dữ liệu: chữ
13	fame	Integer	Điểm danh vọng của user Kiểu dữ liệu: số
14	media	String	Các ảnh user đã đăng Kiểu dữ liệu: chữ
15	playlist	String	Playlist của user Kiểu dữ liệu: chữ
16	address	String	Nơi ở của user Kiểu dữ liệu: chữ
17	tokens	SubCollection	Collection (bảng) con của mỗi user Kiểu dữ liệu: collection
18	photos	SubCollection	Collection con của mỗi user Kiểu dữ liệu: collection

**Bảng 3.4 Thực thể users collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	userId	String	Id của user Kiểu dữ liệu: chữ
2	avatar	String	Avatar của user Kiểu dữ liệu: chữ

3	isAnon	bool	Có phải là User ẩn danh Kiểu dữ liệu: Đúng/Sai
4	messageData	String	Thông điệp Kiểu dữ liệu: chữ
5	notifId	String	Id của thông báo Kiểu dữ liệu: chữ
6	status	String	Trạng thái của thông báo Kiểu dữ liệu: seen/unseen
7	timestamp	Timestamp	Thời gian gửi thông báo Kiểu dữ liệu: thời gian
8	type	String	Loại thông báo Kiểu dữ liệu: chữ

**Bảng 3.5 Thực thể feeds collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	triviaId	String	id của trivia Kiểu dữ liệu: chữ
2	player1	String	Người chơi 1 Kiểu dữ liệu: chữ
3	player2	String	Người chơi 2 Kiểu dữ liệu: chữ
4	questionsDetail	subcollection	Collection con của trivia Kiểu dữ liệu: bảng

**Bảng 3.6 Thực thể trivias collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	questions	String	Câu hỏi Kiểu dữ liệu: chữ
2	answer1	String	Câu trả lời của người chơi 1 Kiểu dữ liệu: chữ
3	answer2	String	Câu trả lời của người chơi 2 Kiểu dữ liệu: chữ

**Bảng 3.7 Thực thể questionDetail sub-collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	postId	String	Id của bài post Kiểu dữ liệu: chữ
2	ownerId	String	Id người đăng Kiểu dữ liệu: chữ
3	likes	Array<User>	Số lượt like Kiểu dữ liệu: Mảng
4	description	String	Nội dung bài viết Kiểu dữ liệu: chữ
5	timestamp	Timestamp	Thời gian đăng Kiểu dữ liệu: ngày tháng năm
6	url	String	Đường link dẫn tới bài post Kiểu dữ liệu: chữ
7	username	String	Tên của người đăng Kiểu dữ liệu: chữ
8	location	Geometry	Nơi đăng bài

**Bảng 3.8 Thực thể posts collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	players	List<User>	Danh sách người chơi Kiểu dữ liệu: Mảng
2	questions	Sub-collection	Câu hỏi Kiểu dữ liệu: collection
3	userAnswers	Sub-collection	Câu trả lời Kiểu dữ liệu: collection

**Bảng 3.9 Thực thể qaGames collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	taskID	String	Id của công việc Kiểu dữ liệu: chữ
2	title	String	Tựa công việc Kiểu dữ liệu: chữ
3	endTime	String	Thời gian kết thúc Kiểu dữ liệu: chữ
4	startTime	String	Thời gian bắt đầu Kiểu dữ liệu: chữ
5	color	int	Màu của công việc Kiểu dữ liệu: số
6	date	String	Ngày thực hiện công việc Kiểu dữ liệu: chữ
7	isCompleted	bool	Trạng thái hoàn thành công việc Kiểu dữ liệu: Đúng/sai
8	remindTime	int	Thời gian báo trước (phút)



			Kiểu dữ liệu: số
--	--	--	------------------

**Bảng 3.10 Thực thể tasks collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	username	String	Tên chủ bài diễn đàn Kiểu dữ liệu: chữ
2	description	String	Mô tả diễn đàn Kiểu dữ liệu: chữ
3	mediaUrl	String	Link ảnh của diễn đàn Kiểu dữ liệu: chữ
4	timestamp	Timestamp	Thời gian tạo diễn đàn Kiểu dữ liệu: Ngày tháng năm
5	title	String	Tựa đề của blog Kiểu dữ liệu: chữ
6	upVotes	Map	Lượt đánh giá tốt Kiểu dữ liệu: Json
7	downVotes	Map	Lượt đánh giá xấu Kiểu dữ liệu: Json
8	forumId	String	Id của diễn đàn Kiểu dữ liệu: chữ
9	ownerId	String	Id của chủ sở hữu Kiểu dữ liệu: chữ
10	category	String	Phân loại diễn đàn Kiểu dữ liệu: chữ

**Bảng 3.11 Thực thể forums collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	postId	String	Id bài post bị báo cáo Kiểu dữ liệu: chữ
2	reason	String	Nội dung báo cáo Kiểu dữ liệu: chữ
3	timestamp	Timestamp	Ngày giờ báo cáo Kiểu dữ liệu: thời gian

**Bảng 3.12 Thực thể reports collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	chatroomId	String	Id của phòng chat Kiểu dữ liệu: chữ
2	users	Array	Người tham gia phòng chat Kiểu dữ liệu: mảng
3	conversation	Sub-collection	Chi tiết chat Kiểu dữ liệu: bảng
4	timestamp	Timestamp	Ngày tạo phòng chat Kiểu dữ liệu: thời gian

**Bảng 3.13 Thực thể chats collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	senderId	String	Id người gửi Kiểu dữ liệu: chữ
2	message	String	Đoạn chat Kiểu dữ liệu: chữ
3	sendBy	String	Tên người gửi Kiểu dữ liệu: chữ
4	timestamp	Timestamp	Thời gian gửi đoạn chat Kiểu dữ liệu: Ngày tháng năm
5	type	String	Kiểu dữ liệu của đoạn chat Kiểu dữ liệu: chữ

**Bảng 3.14 Thực thể conversation sub-collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	avatar	String	Id người gửi Kiểu dữ liệu: chữ
2	commentId	String	Đoạn chat Kiểu dữ liệu: chữ
3	comment	String	Tên người gửi Kiểu dữ liệu: chữ
4	likes	Timestamp	Thời gian gửi đoạn chat Kiểu dữ liệu: Ngày tháng năm
5	replyTo	String	Kiểu dữ liệu của đoạn chat Kiểu dữ liệu: chữ
6	tagId	String	Id người được tag Kiểu dữ liệu: chữ

7	timestamp	Timestamp	Thời gian đăng bình luận Kiểu dữ liệu: thời gian
8	userId	String	Id của người bình luận Kiểu dữ liệu: chữ
9	username	String	Tên của người bình luận Kiểu dữ liệu: chữ

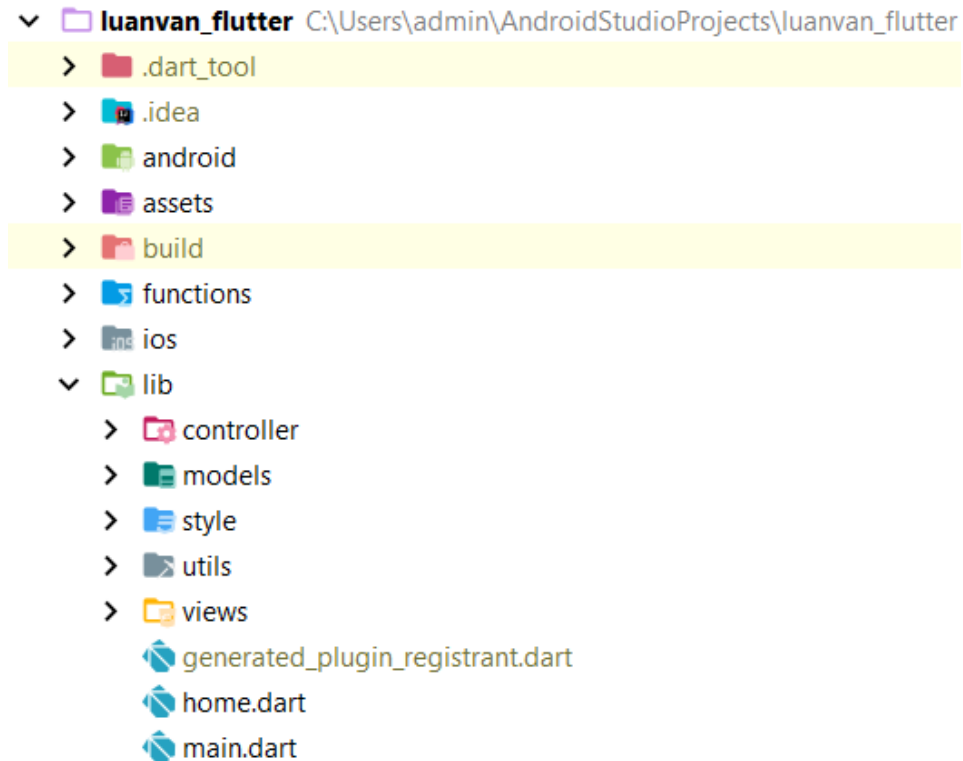
**Bảng 3.15 Thực thể comments/forumComments collection**

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	id	String	Id của người theo dõi Kiểu dữ liệu: chữ
2	avatar	String	Avatar của user Kiểu dữ liệu: chữ
3	username	String	Tên người theo dõi Kiểu dữ liệu: chữ

**Bảng 3.16 Thực thể followers/ followings collection**

### 3.3 Cấu trúc và thể hiện của mô hình MVC

#### 3.3.1 Cấu trúc mô hình MVC trong ứng dụng



Hình 3.26 Cấu trúc mô hình MVC của ứng dụng

##### 3.3.1.1 Cấu trúc thư mục:

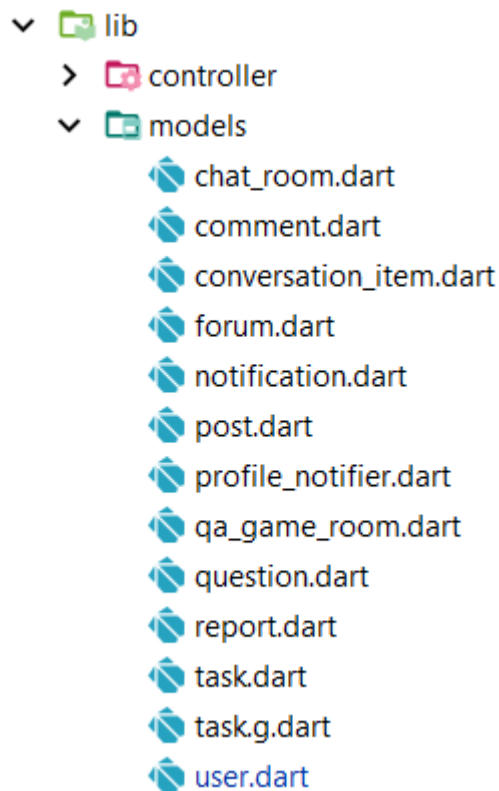
- Thư mục **lib**: là nơi xây dựng ứng dụng, chứa các thư mục con:
- Thư mục **models**: chứa các file dart tạo model kết nối đến cơ sở dữ liệu firebase.
- Thư mục **controller**: chứa các file dart xử lý yêu cầu từ người dùng .
- Thư mục **views**: chứa các file dart render UI người dùng.
- Thư mục **style**: chứa các file dart cung cấp các biến, hàm cho việc sử dụng các thành phần trang trí như màu, kiểu chữ, phông chữ.
- Thư mục **utils**: chứa các file dart chứa function tiện ích như thêm ảnh, tải ảnh, lấy vị trí thiết bị được dùng cho nhiều file.
- Thư mục **assets**: chứa các file media cần dùng trong ứng dụng
- Thư mục **functions**: chứa các scripts để thêm, sửa dữ liệu vào các bảng con khác khi thực hiện các thao tác thay đổi trên cơ sở dữ liệu.

- File **pubspec.yaml**: đây là file chứa các package, khai báo các ảnh offline cho ứng dụng, fonts, là nơi bạn liệt kê các dependencies của dự án .

### 3.3.2 Thể hiện của mô hình MVC trong ứng dụng

#### 3.3.2.1 Khởi tạo Models

Trong package models chứa các class thể hiện thuộc tính của các đối tượng, trong đó có class User chứa các đặc tính của người dùng, theo đó là class Hmmmie chỉ user mà người dùng tương tác như chat hoặc chơi game cùng, class question với mục đích lấy câu hỏi từ game đồ vui và cuối cùng là class profileNotifier giúp hỗ trợ render dữ liệu cho class User.



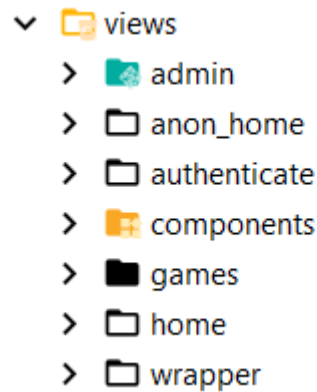
**Hình 3.27 Package models của ứng dụng**

```
//dữ liệu User chi tiết
class UserData extends Equatable {
    late final String id;
    late final String email;
    late final String username;
    late final String nickname;
    late final String gender;
    late final String major;
    late final String bio;
    late final String avatar;
    late final bool isAnon;
    late final String anonBio;
    late final String anonInterest;
    late final String anonAvatar;
    late final List<SubUserData> likes;
    late final String media;
    late final String course;
    late final String playlist;
    late final String address; //chỗ ở
    late final String role;
```

**Hình 3.28** Class User thể hiện thuộc tính người dùng

### 3.3.2.2 Khởi tạo Views

Trong package View chứa các thể hiện GUI và xử lý sự kiện trên GUI của ứng dụng.



**Hình 3.29 Package views của ứng dụng**

```
Size size = MediaQuery.of(context).size;

user = Provider.of<CurrentUserId?>(context);
_saveDeviceToken(user!.uid);
final tabs = [
  const Feed(),
  const TaskManagerPage(),
  const ChatScreen(),
  NotificationPage(
    uid: user!.uid,
  ),
  const MyProfile(),
  SearchScreen(userId: user!.uid),
];

//Các tab ở chế độ ẩn danh
final anonymousTabs = [
  const AnonForumFeed(),
  AnonNotificationPage(
    uid: user!.uid,
  ),
  const AnonProfile(),
  AnonSimpleSearch(),
];
```

**Hình 3.30 Class Home là trang chủ**



### 3.3.2.3 Khởi tạo Controller

Controller bao gồm 2 class **FirestoreAuth** để xác thực đăng nhập, kết nối csdl và class **FirestoreServices** để thực hiện yêu cầu người dùng gửi đến. Tạo file controller.dart và viết các hàm thao tác với dữ liệu.

Class FirestoreServices:

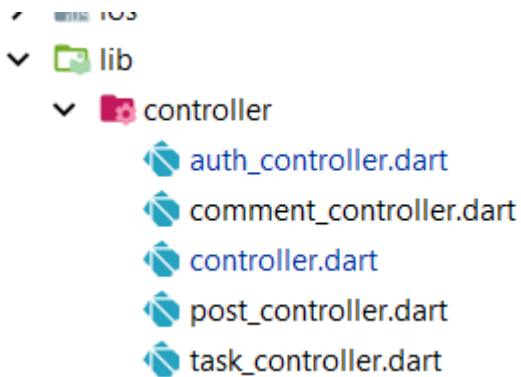
- addForumData : thêm dữ liệu Forum
- getForum: lấy dữ liệu Forum
- addForumMessages: thêm bình luận cho diễn đàn
- getForumMessages: lấy bình luận từ diễn đàn
- updateGame: cập nhật game
- uploadToken: cập nhật token cho người dùng
- updateTrivia: cập nhật câu đố vui
- createTriviaRoom: tạo phòng câu đố
- uploadCompAnswers: tải câu trả lời game tương hợp
- uploadFriendCompAnswers: tải câu trả lời game tương hợp từ bạn bè
- uploadCompQuestions: tải câu hỏi game tương hợp
- createCompRoom: tạo phòng game tương hợp
- getMyCompResult: nhận kết quả game tương hợp
- getFriendCompResult: nhận kết quả game tương hợp từ bạn bè
- getCompQuestion: nhận câu hỏi game tương hợp dạng snapshot
- getDocMyCompResult: nhận Document kết quả game tương hợp
- getDocFriendCompResult: nhận Document kết quả tương hợp của bạn bè
- getWho: nhận dữ liệu 1 User
- getReceiverToken: nhận Token đã được tạo
- acceptRequest: chấp nhận yêu cầu
- getRequestStatus: lấy trạng thái của lời yêu cầu
- uploadBondData: tải dữ liệu ghép đôi lên
- getBond: nhận dữ liệu ghép đôi
- uploadWhoData: cập nhật thông tin của 1 user

- uploadPhotos: tải ảnh lên
- getPhotos: nhận về ảnh
- uploadUserData: tải dữ liệu User
- updateUserData: cập nhật dữ liệu User
- updateAnonymous: cập nhật User ẩn danh
- updateAnonData: cập nhật dữ liệu User ẩn danh
- increaseFame: tăng điểm danh vọng
- decreaseFame: giảm điểm danh vọng
- getPost: trả về bài viết
- likePost: like bài viết
- unlikePost: không like bài viết nữa
- \_hmmieListFromSnapshot: snapshot danh sách những người dùng của Hmmie
- \_userDataFromSnapshot: snapshot danh sách dữ liệu của user
- createChatRoom: tạo chat room
- getChatRoom: trả về chat room
- getNoOfChatRoom: trả về index của chat room
- getNoOfAnonChatRoom: trả về index của chat room chế độ ẩn danh
- createAnonChatRoom: tạo chat room ẩn danh
- getAnonChatRoom: trả về room ẩn danh
- addConversationMessages: gửi tin nhắn
- getConversationMessages: trả về tin nhắn
- addAnonConversationMessages: gửi tin nhắn ẩn danh
- getAnonConversationMessages: trả về tin nhắn ẩn danh

### Class FirebaseAuth:

- \_getFirebaseUserx: trả về user người dùng khởi tạo
- getUser: trả về user từ server
- signInAnon: đăng nhập không danh tính
- signInWithEmailAndPassword: đăng nhập bằng email và mật khẩu
- registerWithEmailAndPassword: đăng ký bằng email và mật khẩu lên server

- updatePassword: cập nhật mật khẩu
- signOut: đăng xuất tài khoản đang đăng nhập
- validatePassword: kiểm tra mật khẩu đăng nhập



**Hình 3.31** Cấu trúc package controller

- **auth\_controller:** Quản lý phiên đăng nhập đăng xuất của người dùng
- **controller:** Chứa tất cả các function fetch dữ liệu
- **post\_controller/comment\_controller/task\_controller:** Sử dụng function từ class controller để lấy dữ liệu

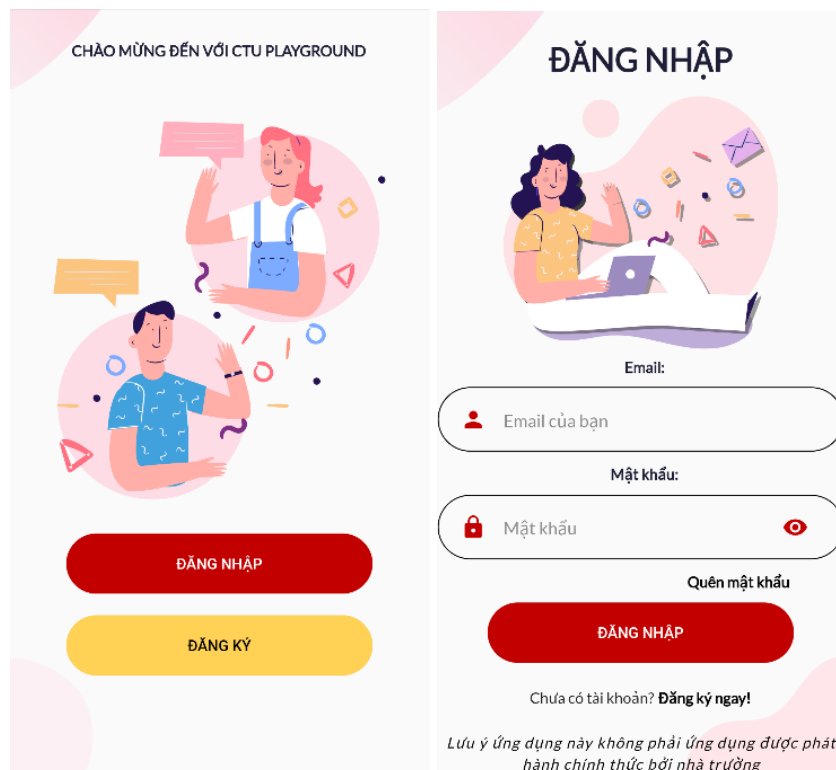
## CHƯƠNG 4. KẾT QUẢ XÂY DỰNG ỨNG DỤNG

### 4.1 Kết quả đạt được

Các chức năng cơ bản dành cho người dùng thực hiện đúng yêu cầu, giao diện người dùng dễ sử dụng, thao tác đơn giản nhưng vẫn đảm bảo các tính năng của người dùng.

Ở bài báo cáo này ứng dụng đã thực hiện được các thao tác như sau:

- Khi mở ứng dụng, người dùng sẽ được yêu cầu đăng nhập.
- Khi người dùng đăng nhập vào hệ thống, hệ thống xác nhận thông tin tài khoản người dùng. Nếu chính xác hệ thống thông báo đăng nhập thành công, ngược lại thông báo đăng nhập thất bại.
- Người dùng có thể khôi phục lại mật khẩu qua mục quên mật khẩu, email khôi phục sẽ được gửi tới người dùng trong hộp thư
- Sau khi đăng nhập thành công, người dùng sẽ được thực hiện các chức năng.



Hình 4.1 Giao diện đăng nhập

## ĐĂNG KÝ

Email:


@ example@student.ctu.edu.vn

Mật khẩu:

Mật khẩu

Nhập lại mật khẩu:

Nhập lại mật khẩu




Tên hiển thị:

Tên của bạn

Biệt danh:

Tên ẩn danh của bạn




Chào mừng đã đến với CTUPG

Cùng nhau tham gia vào cộng đồng nào!

Thành công

Tạo tài khoản thành công, chuyển hướng tới trang chính

Hình 4.2 Giao diện đăng ký

 QUÊN MẬT KHẨU

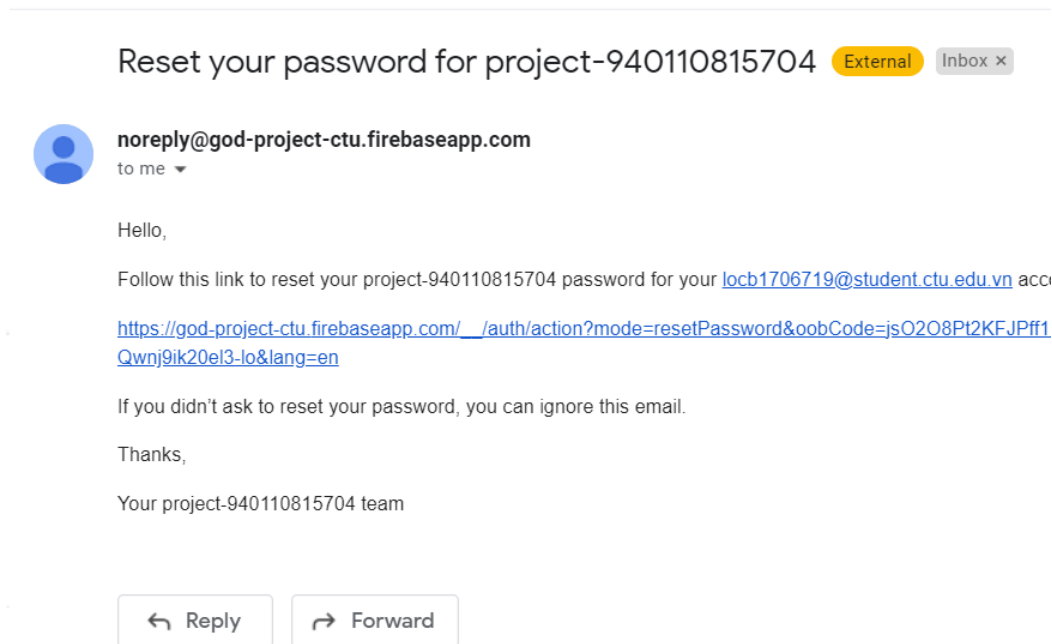
Email:

Email tài khoản đã đăng ký của bạn

XÁC NHẬN

Đường dẫn khôi phục mật khẩu sẽ được gửi đến email của bạn sau khi nhấn XÁC NHẬN

Hình 4.3 Giao diện quên mật khẩu

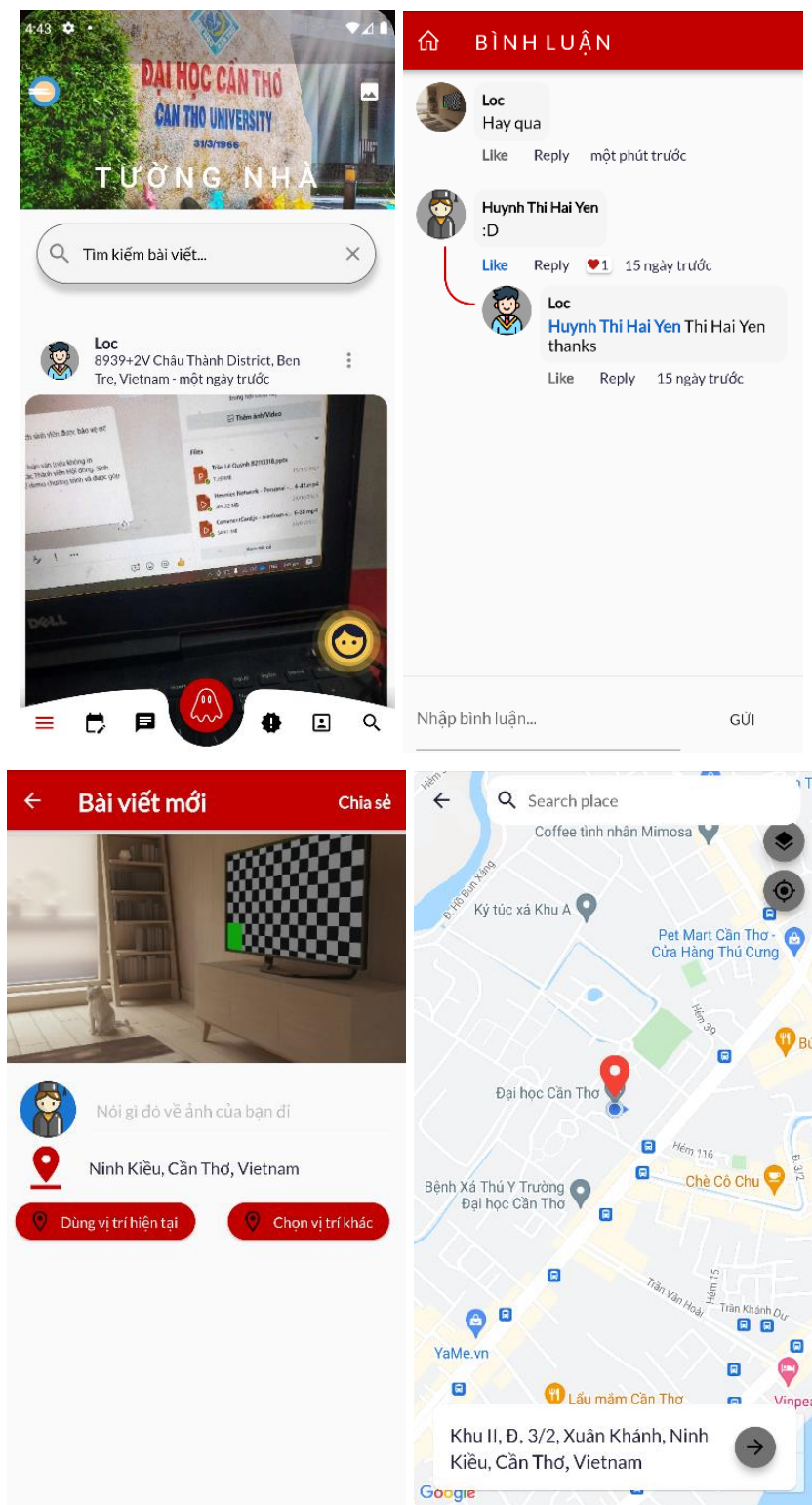


**Hình 4.4 Email khôi phục mật khẩu được gửi**

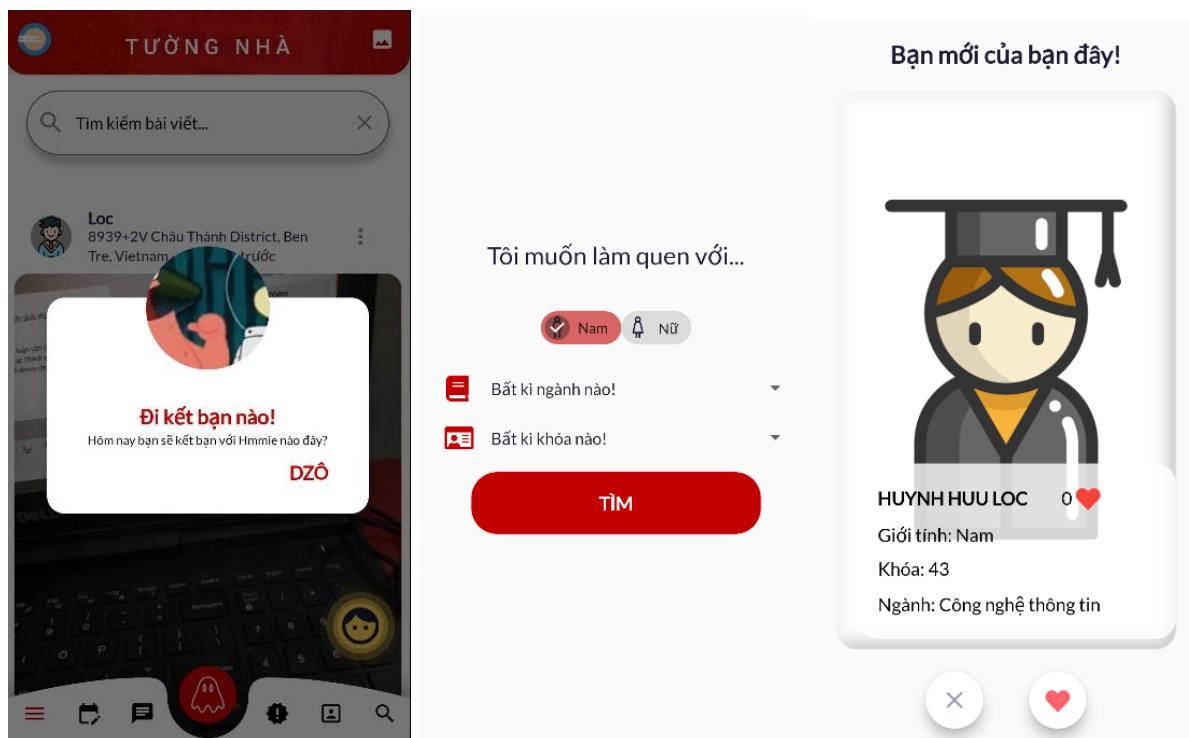
Sau khi đăng nhập, người dùng sẽ tới Home, bao gồm 4 trang: Dòng thời gian, Công việc, Tin nhắn, Thông báo, Hồ sơ cá nhân và Tìm kiếm.

Giao diện dòng thời gian:

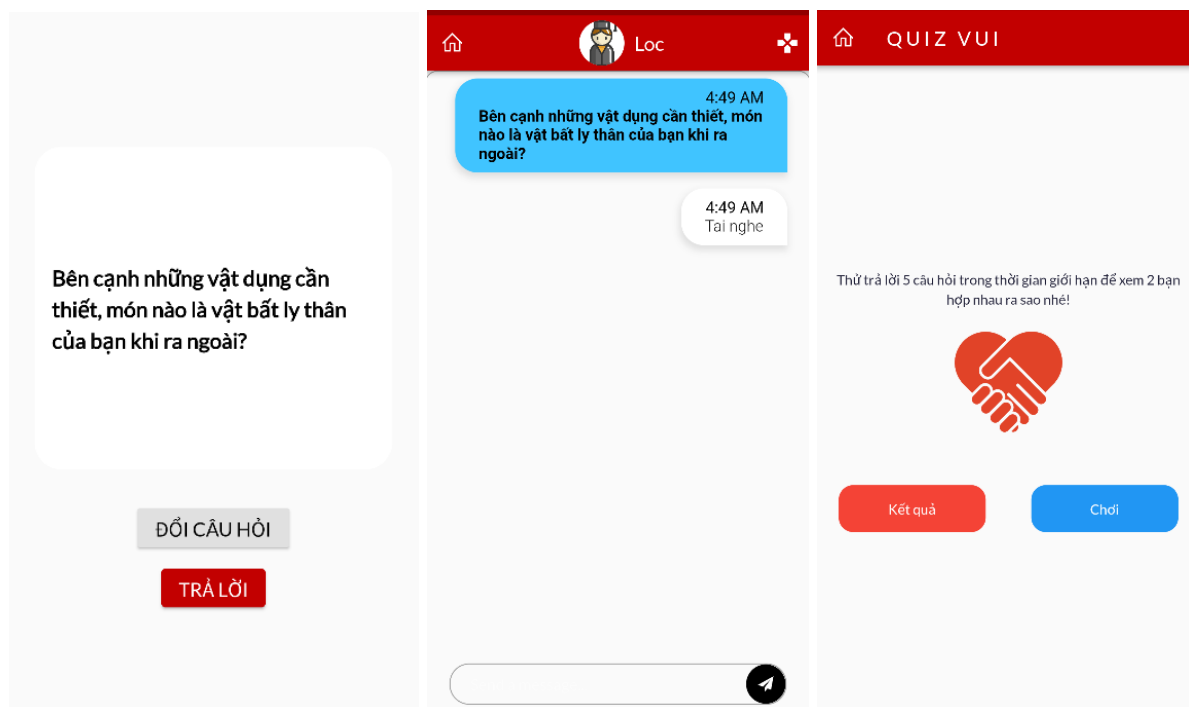
- Khi đến giao diện này, người dùng có thể xem bài viết mọi người đăng cũng như thích, bình luận, thêm vào đó là chức năng tìm kiếm bạn bè mới.
- Khi người dùng bấm vào nút đăng bài, giao diện chọn/ chụp ảnh – video sẽ hiện ra.
- Người dùng sau khi chọn được hình ảnh hoặc video ưng ý có thể xem preview, chọn vị trí đăng trước khi đăng bài.
- Khi người dùng bấm vào nút tìm bạn bè mới, 1 dialog sẽ hiện ra cho người dùng chọn giới tính, ngành học, khóa của người bạn mới.
- Sau khi xác nhận, 1 thẻ bài ghi thông tin của người bạn mới sẽ hiện ra, bấm vào tìm thì người dùng sẽ được đưa đến màn hình bắt chuyện, chọn 1 câu hỏi ngẫu nhiên và trả lời để bắt đầu cuộc trò chuyện.



Hình 4.5 Giao diện home/ đăng bài viết



Hình 4.6 Giao diện thông tin user

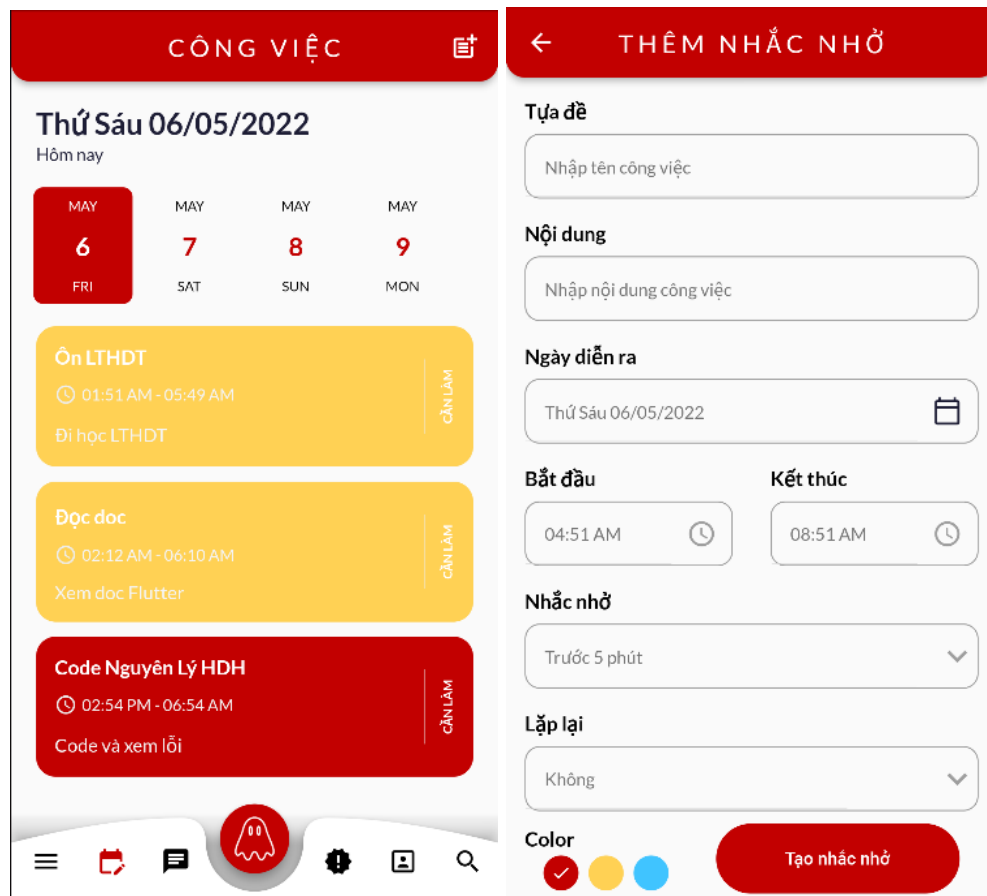


Hình 4.7 Giao diện bắt chuyện – game



Giao diện công việc:

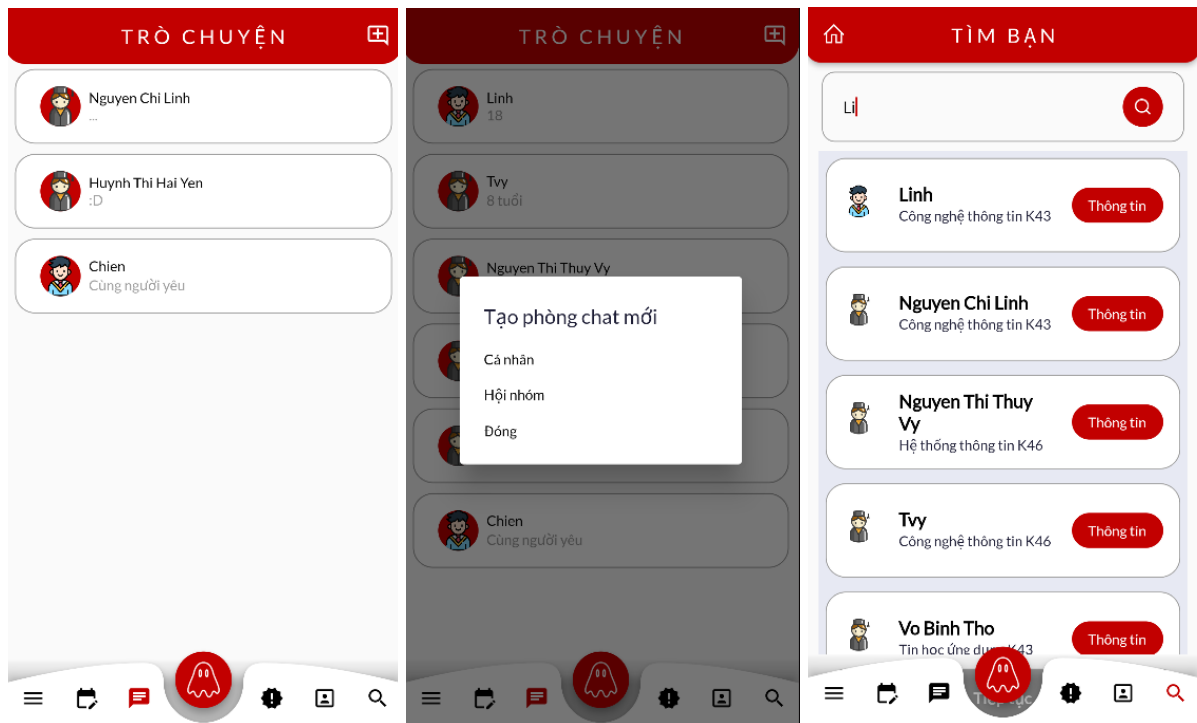
- Là nơi người dùng quản lý công việc của mình
- Khi người dùng bấm vào nút thêm, màn hình thêm công việc sẽ hiện ra, sau khi nhập xong các thông tin cần có cho công việc, công việc mới sẽ được thêm vào danh sách và sẽ đổ chuông vào thời gian người dùng đã thiết lập.



Hình 4.8 Giao diện công việc

Giao diện trò chuyện:

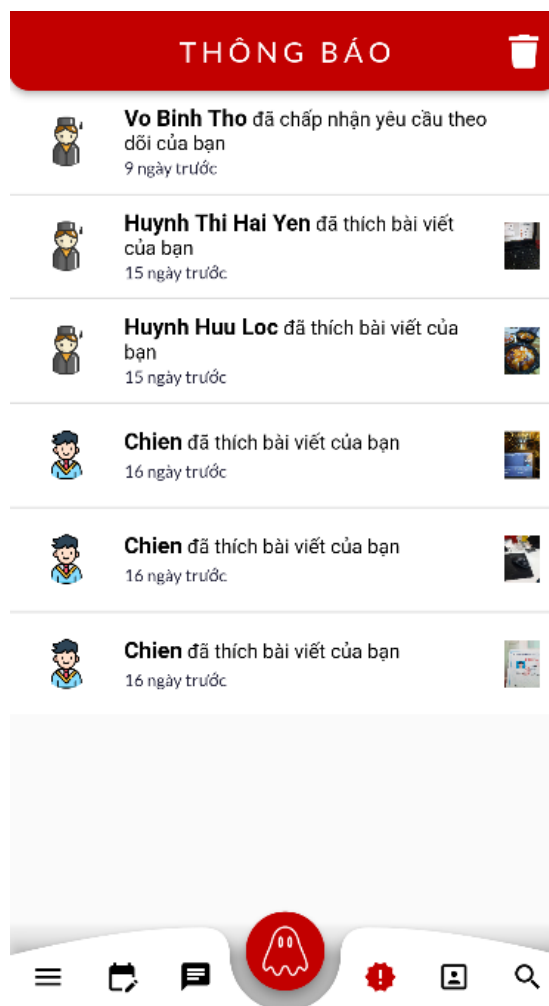
- Người dùng có thể trò chuyện với bạn bè ở đây.
- Khi người dùng bấm vào thêm cuộc trò chuyện, 1 dialog sẽ hiện ra cho người dùng chọn nhắn tin cá nhân hoặc theo nhóm.
- Sau khi chọn người dùng sẽ được đưa đến màn hình chọn bạn chat, sau khi chọn bạn nếu là theo nhóm thì sẽ mở popup đặt tên nhóm.



Hình 4.9 Giao diện phòng chat

Giao diện thông báo:

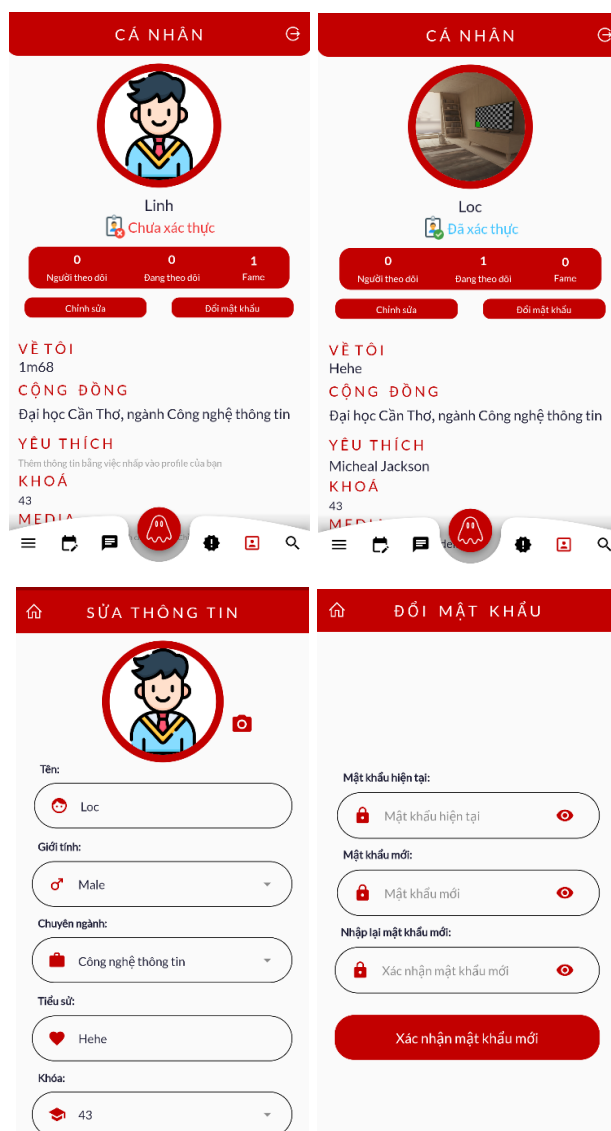
- Người nhận thông báo tương tác từ người dùng khác tại đây.
- Khi người dùng bấm vào 1 thông báo thì sẽ được chuyển hướng tới bài viết/ diễn đàn/ phòng chat



Hình 4.10 Giao diện thông báo

### Giao diện hồ sơ cá nhân:

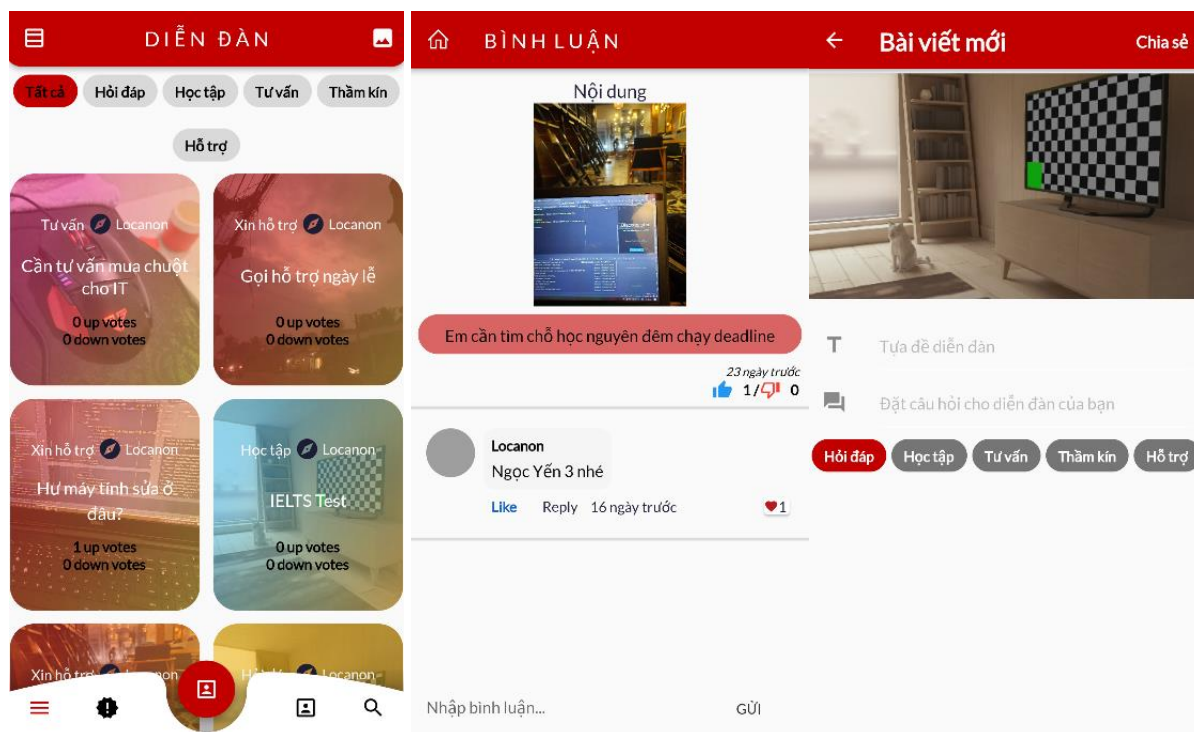
- Là nơi hiển thị hồ sơ người dùng, ở đây người dùng có thể thay đổi thông tin cá nhân, xác thực email, đổi mật khẩu hoặc đăng xuất.
- Khi người dùng bấm vào dòng chữ chưa xác thực, một email xác thực sẽ được gửi tới hòm thư của bạn. Dùng đường link được cung cấp trong mail để xác thực bạn đang sử dụng tài khoản chính thức của trường.



**Hình 4.11** Giao diện thông tin cá nhân

Giao diện diễn đàn:

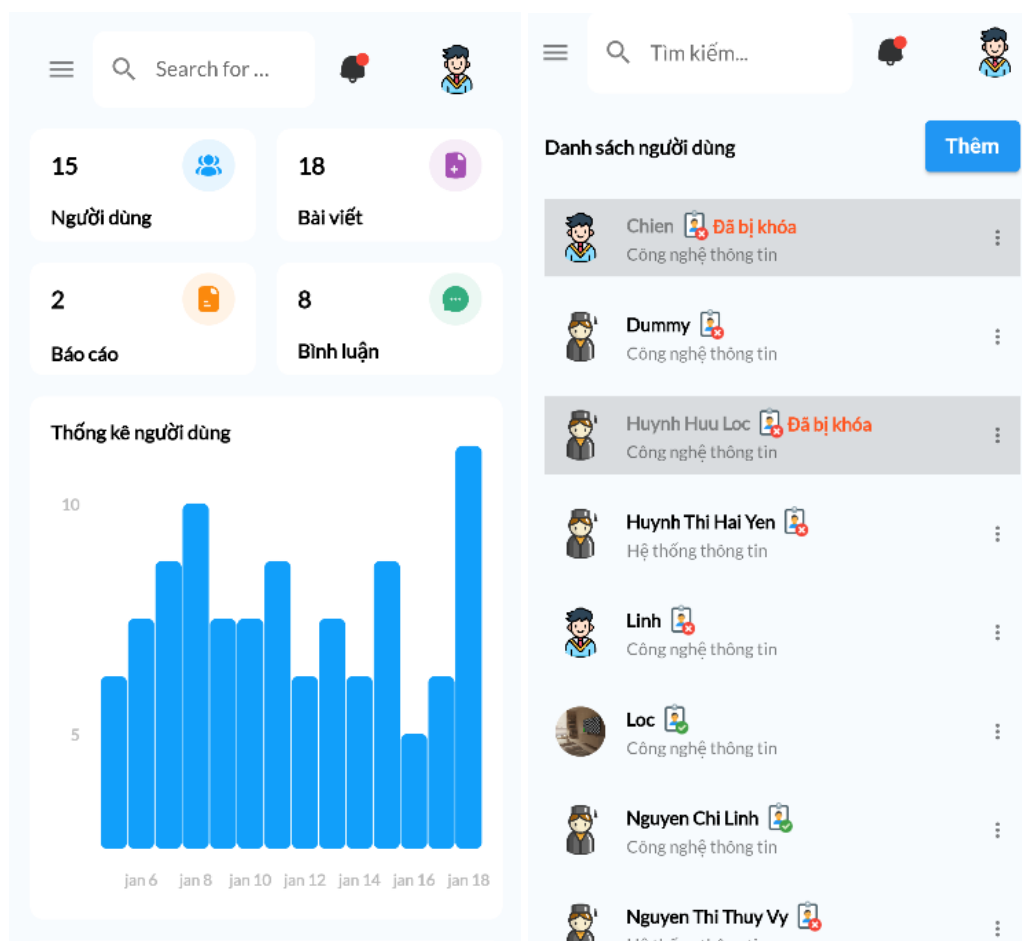
- Nơi mọi người dùng trao đổi thông tin dưới dạng ẩn danh, tương tự như xem bài viết.

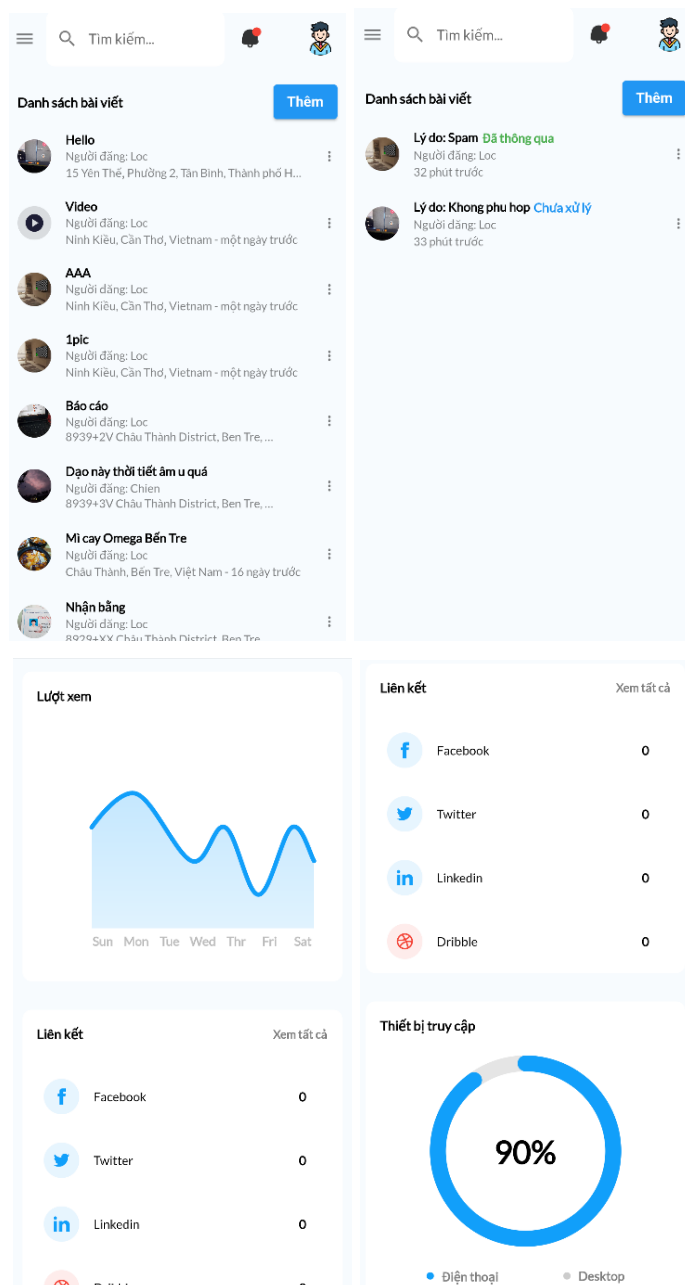


Hình 4.12 Giao diện diễn đàn

Giao diện diễn đàn:

- Nơi Admin xem các thống kê, báo cáo, quản lý các bài viết, diễn đàn, các báo cáo cho những nội dung không phù hợp
- Admin có thể khóa hoặc xóa tài khoản nếu tài khoản đó không đạt đủ yêu cầu hoặc không phù hợp (chưa xác thực quá lâu, quá nhiều bài viết bị báo cáo,...)
- Admin có thể thực hiện các thao tác thêm sửa xóa ở các mục khác





**Hình 4.13** Giao diện admin

## **4.2 Hướng phát triển**

“Ứng dụng mạng xã hội cho sinh viên Đại học Cần Thơ” sẽ được phát triển tích hợp trên cả PC và trên thiết bị di động. Với thời đại công nghệ 4.0 như hiện nay thì việc chia sẻ thông tin là điều cần thiết và nhất thiết phải thực hiện. Do đó cần tìm hiểu thêm về mong muốn của người dùng trong quá trình tương tác với ứng dụng để tiếp tục hoàn thiện các chức năng đáp ứng được nhu cầu của người dùng. Thực hiện thêm chức năng hỗ trợ cho các giảng viên, cựu sinh viên, các nhà tuyển dụng được tham gia. Hiện nay, trí tuệ nhân tạo hữu ích ở tất cả mọi lĩnh vực, từ việc nhỏ đến việc lớn, trí tuệ nhân tạo đều phát huy lợi ích của mình một cách triệt để. Do đó sẽ thật là hữu ích khi ứng tích hợp trí tuệ nhân tạo như chatbot, hệ thống gợi ý để hỗ trợ học tập không chỉ cho sinh viên còn giúp tiết kiệm thời gian và nhân lực cho các giảng viên, hiệu quả cao mà đảm bảo chất lượng phục vụ sinh viên. Và đó là những gì mà em đang định hướng phát triển cho đề tài của mình.



## TÀI LIỆU THAM KHẢO

### TÀI LIỆU SÁCH, BÁO, GIÁO TRÌNH

[1] Kevin D Moore, Michael Katz and Vincent Ngo (2021). *Flutter Apprentice (First Edition)*.

[2] Jonathan Sande & Matt Galloway (2021). *Dart Apprentice (First Edition)*.

### TÀI LIỆU TRỰC TUYẾN

#### Tiếng Việt

[3] Lê Thành Trung (2021). *Tổng hợp các tài liệu học Flutter - Tự học*. Địa chỉ <<https://fullstack.edu.vn/blog/tong-hop-cac-tai-lieu-hoc-flutter-tu-hoc.html>> [Truy cập ngày 05/07/2020]

[4] Nguyễn Thành Minh (2020). *Học Flutter từ cơ bản đến nâng cao*. Địa chỉ <<https://viblo.asia/p/hoc-flutter-tu-co-ban-den-nang-cao-phan-1-lam-quen-co-nang-flutter-4dbZNJOvZYM>> [Truy cập ngày 05/08/2020]

[5] Đỗ Đức Đình Đặc (2020). *Firebase là gì? Giải pháp lập trình không cần Backend từ Google*. Địa chỉ <<https://wiki.matbao.net/firebase-la-gi-giai-phap-lap-trinh-khong-can-backend-tu-google/>> [Truy cập ngày 05/08/2020]

[6] Vncoder (2019). *Kiến trúc ứng dụng Flutter - Học lập trình Flutter cơ bản*. Địa chỉ <<https://vncoder.vn/bai-hoc/kien-truc-ung-dung-flutter-215>> [Truy cập ngày 20/10/2020]

#### Tiếng Anh

[7] Dart (2018). *Dart Documentation*. Địa chỉ <<https://dart.dev/guides>> [Truy cập năm 2019].

[8] Flutter (2020). *Flutter Documentation*. Địa chỉ <<https://docs.flutter.dev/>> [Truy cập năm 2019].

[9] Flutter (2014). *Architecture Application*. Địa chỉ <[https://www.tutorialspoint.com/flutter/flutter\\_architecture\\_application.htm#:~:text=The%20core%20concept%20of%20the,application%20is%20itself%20a%20widget.](https://www.tutorialspoint.com/flutter/flutter_architecture_application.htm#:~:text=The%20core%20concept%20of%20the,application%20is%20itself%20a%20widget.)> [Truy cập tháng 06/2021].

[10] Flutter (2020). *Flutter Template Project using MVC Design Pattern*. Địa chỉ <<https://medium.com/follow-flutter/your-next-mvc-flutter-project-1fabe2069b01>> [Truy cập tháng 07/2021].