|  |  |
| --- | --- |
|  | **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**  **TP. HCM**  **KHOA CÔNG NGHỆ THÔNG TIN** |

**BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH**

**Đề tài:**

**APP MẠNG XÃ HỘI**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Giảng viên Hướng dẫn: **Trần Thị Vân Anh**

Sinh viên thực hiện: MSSV:

**1. Hồ Đại Minh 2180604350**

**2. Phan Thị Hằng 2180605032**

**3. Phạm Tấn Phát 2180606067**

Lớp: **21DTHC1**

*TP. Hồ Chí Minh, 2024*

# **LỜI CAM ĐOAN**

Nhóm chúng em xin được cam đoan đồ án chuyên ngành "App mạng xã hội" là một đề tài được nghiên cứu độc lập của chúng em dưới sự hướng dẫn của GV: Trần Thị Vân Anh.

Những số liệu và kết quả nghiên cứu là trung thực, hoàn toàn được thực hiện bởi chúng em. Ngoài ra, trong đồ án có sử dụng một số nguồn tài liệu đã được chúng em trích dẫn nguồn và chú thích rõ ràng. Chúng em xin hoàn toàn chịu mọi trách nhiệm về lời cam đoan này.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[LỜI CAM ĐOAN 1](#_Toc185566368)

[CHƯƠNG 1. TỔNG QUAN 4](#_Toc185566369)

[**1.1. Tổng quan về vấn đề nghiên cứu** 4](#_Toc185566370)

[1.1.1. Khảo sát thực trạng 4](#_Toc185566371)

[**1.2. Nhiệm vụ đồ án** 4](#_Toc185566372)

[1.2.1. Đối tượng và phạm vi của ứng dụng 4](#_Toc185566373)

[1.2.1.1. Đối tượng 4](#_Toc185566374)

[1.2.1.2. Phạm vi 4](#_Toc185566375)

[**1.3. Cấu trúc đồ án** 6](#_Toc185566376)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 7](#_Toc185566377)

[**2.1. Các khái niệm và cơ chế hoạt động** 7](#_Toc185566378)

[2.1.1 Công cụ và ngôn ngữ lập trình 7](#_Toc185566379)

[2.1.1.1. Công cụ lập trình 7](#_Toc185566380)

[2.1.1.2. Ngôn ngữ lập trình React Native 7](#_Toc185566381)

[2.1.1.3. Giới thiệu Node.js 8](#_Toc185566382)

[2.1.1.4. Tổng quan về cơ sở dữ liệu Firebase 9](#_Toc185566383)

[2.1.1.5. Giới thiệu về Android Studio 10](#_Toc185566384)

[2.1.1.6. Cloudinary 11](#_Toc185566385)

[**2.2. Thông tin và yêu cầu về hệ thống** 12](#_Toc185566386)

[**2.3. Các biểu đồ về chức năng của hệ thống** 14](#_Toc185566387)

[2.3.1. Use Case Diagram toàn hệ thống 14](#_Toc185566388)

[2.3.2. Activity Diagram toàn hệ thống 15](#_Toc185566389)

[2.3.2.1. Activity Diagram dòng thời gian 15](#_Toc185566390)

[2.3.2.2. Activity Diagram đăng bài viết 16](#_Toc185566391)

[2.3.2.3. Activity Diagram nhắn tin 16](#_Toc185566392)

[2.3.2.4. Activity Diagram quản lý trang cá nhân 17](#_Toc185566393)

[2.3.2.5. Activity Diagram quản lý quyền riêng tư 18](#_Toc185566394)

[2.3.3. Sequence Diagram toàn hệ thống 20](#_Toc185566395)

[2.3.3.1. Sequence Diagram dòng thời gian 20](#_Toc185566396)

[2.3.3.2. Sequence Diagram đăng bài viết 21](#_Toc185566397)

[2.3.3.3. Sequence Diagram nhắn tin 22](#_Toc185566398)

[2.3.3.4. Sequence Diagram quản lí trang cá nhân 23](#_Toc185566399)

[2.3.3.5. Sequence Diagram quản lý quyền riêng tư 24](#_Toc185566400)

[2.4. Cơ sở dữ liệu FireBase 26](#_Toc185566401)

[2.4.1. API của FireBase 26](#_Toc185566402)

[2.4.2. Collection 26](#_Toc185566403)

[CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM 29](#_Toc185566404)

[3.1. Các thành phần chức năng của hệ thống 29](#_Toc185566405)

[**3.2. Demo** 30](#_Toc185566406)

[3.2.1. Giao diện đăng ký và đăng nhập 30](#_Toc185566407)

[3.2.2. Giao diện trang chủ 31](#_Toc185566408)

[3.2.3. Giao diện bình luận 33](#_Toc185566409)

[3.2.4. Giao diện tìm kiếm người dùng 35](#_Toc185566410)

[3.2.5. Giao diện tạo bài viết 37](#_Toc185566411)

[3.2.6. Giao diện trang cá nhân 40](#_Toc185566412)

[3.2.6. Giao diện trang tin nhắn 41](#_Toc185566413)

[CHƯƠNG 4. KẾT LUẬN VÀ ĐÁNH GIÁ 43](#_Toc185566414)

[4.1. Kết luận 43](#_Toc185566415)

[4.2. Kết quả đạt được 43](#_Toc185566416)

[4.3. Những thứ chưa làm được 44](#_Toc185566417)

[TÀI LIỆU THAM KHẢO 45](#_Toc185566418)

**DANH SÁCH HÌNH VẼ**

Hình 2.6. Use Case tổng quát …………………………………………………………15

Hình 2.7. Activity Diagram dòng thời gian……………………………………………16

Hình 2.8. Activity Diagram đăng bài viết ……………………………………………16

Hình 2.9. Activity Diagram nhắn tin ………………………………………..………..17

Hình 2.10. Activity Diagram quản lý trang cá nhân………………………………..…18

Hình 2.11. Activity Diagram quản lý quyền riêng tư ……………...………………….19

Hình 2.12. Sequence Diagram dòng thời gian .………………………...……………20

Hình 2.13. Sequence Diagram đăng bài viết ..…………….…………………………..21

Hình 2.14. Sequence Diagram nhắn tin ……………………………………………….22

Hình 2.15. Sequence Diagram quản lí trang cá nhân …………………………...…….23

Hình 2.16. Sequence Diagram quản lý quyền riêng tư………………………………..24

**CHƯƠNG 1. TỔNG QUAN**

**1.1. Tổng quan về vấn đề nghiên cứu**

*1.1.1. Khảo sát thực trạng*

Hiện nay, mạng xã hội đã trở thành một phần quan trọng trong cuộc sống hàng ngày của nhiều người, giúp kết nối và chia sẻ thông tin một cách nhanh chóng. Tuy nhiên, nhiều người vẫn gặp khó khăn khi sử dụng các nền tảng mạng xã hội phổ biến vì giao diện phức tạp hoặc thiếu các tính năng đơn giản phục vụ nhu cầu giao tiếp cơ bản.

Một số vấn đề phổ biến bao gồm:

* Giao diện của nhiều nền tảng hiện nay chưa tối ưu, gây khó khăn cho người dùng mới.
* Chưa tập trung vào các chức năng giao tiếp và chia sẻ cơ bản, như đăng bài viết, bình luận hoặc gửi tin nhắn.
* Các ứng dụng hiện có thường nặng và yêu cầu nhiều tài nguyên thiết bị, không phù hợp với tất cả người dùng.

Những vấn đề này đặt ra nhu cầu phát triển một ứng dụng mạng xã hội mới, tập trung vào sự đơn giản, tiện dụng và đáp ứng các nhu cầu cơ bản của người dùng.

**1.2. Nhiệm vụ đồ án**

Mục tiêu của đồ án là phát triển một ứng dụng mạng xã hội với các tính năng cơ bản, giao diện thân thiện và dễ sử dụng. Ứng dụng sẽ giúp người dùng dễ dàng kết nối, chia sẻ trạng thái và tương tác với bạn bè mà không cần phải làm quen với các chức năng phức tạp.

*1.2.1. Đối tượng và phạm vi của ứng dụng*

1.2.1.1. Đối tượng

* Người dùng phổ thông có nhu cầu chia sẻ trạng thái, hình ảnh hoặc trò chuyện với bạn bè
* Người quản trị hệ thống chịu trách nhiệm duy trì và giám sát hoạt động của ứng dụng.

1.2.1.2. Phạm vi

1. Thông tin tổng quan:

* Đơn vị sử dụng: Cá nhân hoặc nhóm người có nhu cầu sử dụng mạng xã hộ.
* Tên dự án: App mạng xã hội.

1. Phát biểu vấn đề:

* Trong bối cảnh ngày càng gia tăng nhu cầu kết nối xã hội, các nền tảng mạng xã hội cần tập trung vào sự đơn giản và dễ sử dụng để phục vụ nhiều đối tượng người dùng khác nhau**.**

1. Mục tiêu:

* Cung cấp nền tảng mạng xã hội với các tính năng đầy đủ, từ đăng bài, nhắn tin đến quản lý thông báo và mối quan hệ bạn bè.
* Đảm bảo giao diện thân thiện, dễ sử dụng, phù hợp với mọi đối tượng.

1. Mô tả chức năng:

* Đăng ký và đăng nhập:
  + Đăng bài, chỉnh sửa, xóa bài viết.
  + Thích, bình luận và chia sẻ bài viết của người khác.
* Nhắn tin:
  + Hỗ trợ nhắn tin trực tiếp giữa các người dùng.
  + Tích hợp gửi hình ảnh và emoji.
* Thông báo:
  + Gửi thông báo khi có lượt thích, bình luận, hoặc tin nhắn mới.
  + Tùy chỉnh cài đặt nhận thông báo.
* Trang cá nhân:
  + Quản lý thông tin cá nhân, cập nhật ảnh đại diện và ảnh bìa.
  + Hiển thị danh sách bài viết và tương tác của người dùng.
  + Chỉnh sửa quyền riêng tư
* Tìm kiếm:
  + Hỗ trợ tìm kiếm bạn bè, bài viết.
* Chia sẻ vị trí trực tuyến của người dùng

1. Các lợi ích mang lại:

* Giúp người dùng dễ dàng kết nối, chia sẻ và tương tác với cộng đồng.
* Đáp ứng nhu cầu sử dụng mạng xã hội một cách toàn diện và hiệu quả.
* Tăng cường khả năng quản lý mối quan hệ, nhóm và thông báo cá nhân.

1. Các bước thực hành dự án:

* Lập kế hoach phát triển dự án.
* Phân tích hệ thống.
* Lập trình.
* Cài đặt.
* Kiểm thử phần mềm.
* Biện soạn và hướng dẫn sử dụng.

**1.3. Cấu trúc đồ án**

Đồ án gồm có 4 chương:

* Chương 1: Tổng quan

Phần này giới thiệu tổng quan, nhiệm vụ của đồ án, giúp chúng ta hiểu nội dung căn bản của đồ án

* Chương 2: Cơ sở lý thuyết
* Chương 3: Kết quả thực nghiệm

Phần này sẽ cho chúng ta thấy mô hình mục tiêu ứng dụng, hình ảnh, mô tả về ứng dụng.

* Chương 4: Kết luận và kiến nghị.

Phần này sẽ rút ra những lưu ý và lời khuyên khi sử dụng ứng dụng và đưa ra hướng phát triển hoàn thiện hơn.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Các khái niệm và cơ chế hoạt động**

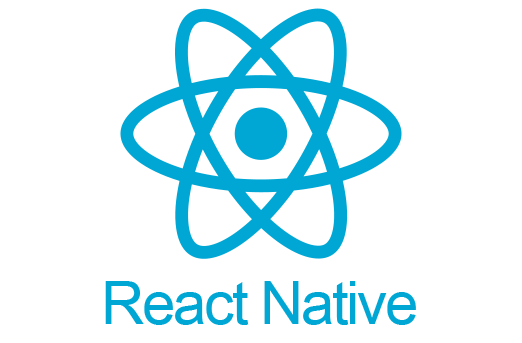
*2.1.1 Công cụ và ngôn ngữ lập trình*

2.1.1.1. Công cụ lập trình

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí, và đa nền tảng do Microsoft phát triển. Ra mắt vào năm 2015, VS Code nhanh chóng trở thành một trong những công cụ lựa chọn hàng đầu cho các nhà phát triển phần mềm nhờ khả năng tích hợp linh hoạt, giao diện dễ sử dụng, và hiệu suất cao.

VS Code hỗ trợ nhiều ngôn ngữ lập trình và cung cấp các tính năng hữu ích như IntelliSense, gỡi ý mã nguồn, debugging, tích hợp Git, và nhiều tiện ích khác thông qua các extension.

2.1.1.2. Ngôn ngữ lập trình React Native



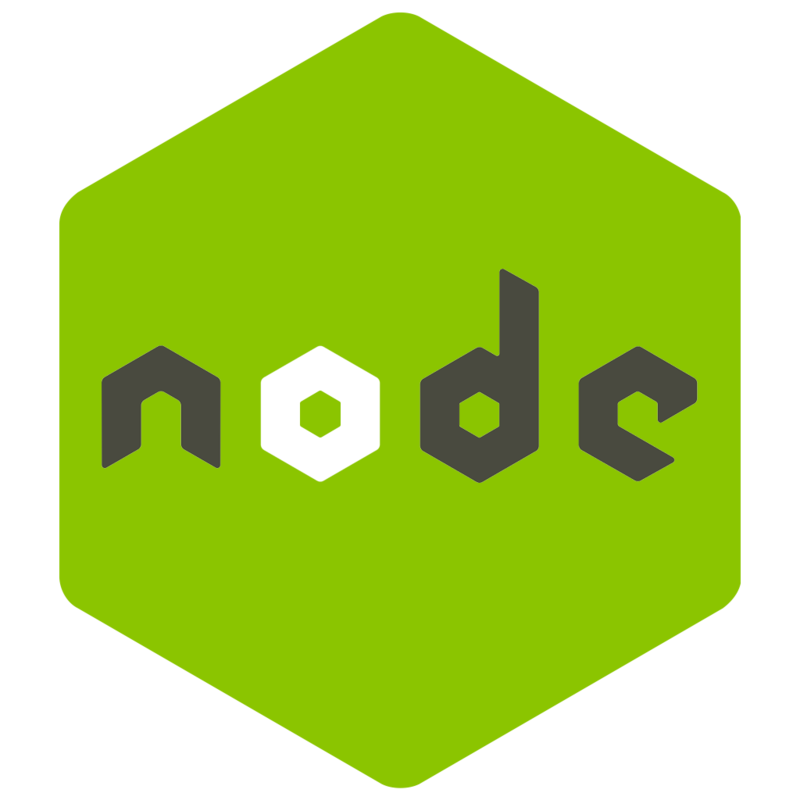
Hình 2.1. React Native

React Native là một framework mã nguồn mở được phát triển bởi Facebook, cho phép lập trình viên xây dựng ứng dụng di động cho cả iOS và Android bằng cách sử dụng JavaScript và React. Với mục tiêu mang lại trải nghiệm phát triển ứng dụng nhanh chóng và hiệu quả, React Native cho phép lập trình viên viết mã một lần và triển khai trên nhiều nền tảng, từ đó tiết kiệm thời gian và công sức đáng kể. Framework này sử dụng các thành phần giao diện người dùng (UI components) được biên dịch thành mã gốc (native code), giúp ứng dụng chạy mượt mà và hiệu suất cao hơn so với các ứng dụng web di động truyền thống.

Một trong những tính năng nổi bật của React Native là khả năng hot reloading, cho phép lập trình viên xem ngay lập tức các thay đổi trong mã mà không cần phải biên dịch lại ứng dụng. Điều này không chỉ tăng tốc độ phát triển mà còn cải thiện quy trình làm việc, giúp lập trình viên thử nghiệm và sửa lỗi một cách dễ dàng hơn. Hệ sinh thái của React Native rất phong phú, với hàng ngàn thư viện và module từ cộng đồng, giúp mở rộng tính năng của ứng dụng và hỗ trợ tích hợp các API khác nhau, như GPS, camera và cảm biến.

Ngoài ra, React Native còn hỗ trợ quản lý trạng thái hiệu quả thông qua Redux hoặc Context API, giúp duy trì và quản lý dữ liệu trong ứng dụng một cách mạch lạc. Với kiến trúc component-based, các lập trình viên có thể xây dựng giao diện bằng cách kết hợp các component lại với nhau, tạo ra mã nguồn dễ đọc và bảo trì. Nhờ vào những ưu điểm vượt trội này, React Native đã trở thành một trong những lựa chọn hàng đầu cho các lập trình viên trong việc phát triển ứng dụng di động hiện đại.

2.1.1.3. Giới thiệu Node.js



Hình 2.2. Node.js

Node.js là một nền tảng phát triển ứng dụng mã nguồn mở, cho phép chạy JavaScript trên máy chủ, được phát triển bởi Ryan Dahl vào năm 2009. Sử dụng engine V8 của Google Chrome, Node.js mang lại hiệu suất cao và khả năng xử lý đồng thời tốt nhờ vào mô hình lập trình bất đồng bộ. Điều này cho phép các ứng dụng xử lý nhiều kết nối đồng thời mà không bị chặn, làm cho Node.js trở thành lựa chọn lý tưởng cho các ứng dụng thời gian thực như chat và dịch vụ API. Hệ sinh thái npm (Node Package Manager) cung cấp hàng triệu thư viện và module, giúp lập trình viên dễ dàng mở rộng chức năng cho ứng dụng của mình. Với sự hỗ trợ mạnh mẽ từ cộng đồng, Node.js đã trở thành một trong những công nghệ phổ biến nhất trong phát triển ứng dụng web hiện đại.

2.1.1.4. Tổng quan về cơ sở dữ liệu Firebase



Hình 2.3. Firebase

Firebase là một nền tảng phát triển ứng dụng được Google phát triển, cung cấp một loạt các công cụ và dịch vụ để xây dựng, phát triển và quản lý ứng dụng di động và web. Được ra mắt vào năm 2011 và sau đó được Google mua lại vào năm 2014, Firebase giúp lập trình viên dễ dàng tích hợp các tính năng phức tạp mà không cần xây dựng từ đầu. Một số dịch vụ chính của Firebase bao gồm cơ sở dữ liệu thời gian thực (Realtime Database), Firestore, lưu trữ tệp (Cloud Storage), xác thực người dùng (Authentication), và dịch vụ phân tích (Analytics).

Firebase nổi bật với khả năng cung cấp cơ sở dữ liệu thời gian thực, cho phép người dùng đồng bộ hóa dữ liệu ngay lập tức giữa nhiều thiết bị khác nhau. Điều này rất hữu ích cho các ứng dụng yêu cầu cập nhật dữ liệu liên tục, chẳng hạn như trò chuyện hay các ứng dụng cộng đồng. Ngoài ra, Firebase cung cấp tính năng xác thực người dùng đơn giản và an toàn, hỗ trợ nhiều phương thức đăng nhập như email, Google, Facebook và nhiều dịch vụ khác.

Hệ sinh thái của Firebase còn bao gồm Cloud Functions cho phép chạy mã backend mà không cần quản lý máy chủ, cùng với Cloud Messaging để gửi thông báo đẩy đến người dùng. Với sự hỗ trợ mạnh mẽ từ Google và cộng đồng, Firebase đã trở thành một trong những nền tảng phổ biến nhất cho việc phát triển ứng dụng hiện đại, giúp lập trình viên tập trung vào việc xây dựng trải nghiệm người dùng tốt mà không cần lo lắng về hạ tầng.

2.1.1.5. Giới thiệu về Android Studio



Hình 2.4. Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức được Google phát triển cho việc xây dựng ứng dụng Android. Được phát hành lần đầu vào năm 2013, Android Studio dựa trên IntelliJ IDEA và cung cấp một loạt các công cụ mạnh mẽ giúp lập trình viên dễ dàng phát triển, thử nghiệm và triển khai ứng dụng Android. Với giao diện thân thiện và tính năng đa dạng, Android Studio hỗ trợ lập trình viên từ giai đoạn thiết kế giao diện đến việc viết mã và gỡ lỗi.

Một trong những tính năng nổi bật của Android Studio là trình thiết kế giao diện người dùng (UI Designer), cho phép lập trình viên xây dựng giao diện trực quan thông qua các thành phần kéo và thả. Ngoài ra, Android Studio cung cấp các công cụ gỡ lỗi mạnh mẽ, giúp phát hiện và khắc phục lỗi trong mã một cách dễ dàng. Hệ thống quản lý phiên bản tích hợp cũng hỗ trợ làm việc với Git, giúp theo dõi và quản lý thay đổi mã nguồn hiệu quả.

Android Studio còn có một bộ công cụ phong phú cho việc thử nghiệm ứng dụng, bao gồm công cụ giả lập Android (Android Emulator) cho phép chạy ứng dụng trên nhiều thiết bị ảo với các cấu hình khác nhau. Nó cũng hỗ trợ tích hợp với Firebase và các dịch vụ khác, giúp phát triển ứng dụng nhanh chóng và hiệu quả. Nhờ vào những tính năng mạnh mẽ và sự hỗ trợ liên tục từ Google, Android Studio đã trở thành lựa chọn hàng đầu cho lập trình viên Android trên toàn thế giới.

2.1.1.6. Cloudinary



Hình 2.5. Cloudinary

Cloudinary là một nền tảng quản lý và tối ưu hóa hình ảnh và video được sử dụng rộng rãi trong phát triển web và ứng dụng di động. Nó cung cấp các công cụ để tải lên, lưu trữ, biến đổi, tối ưu hóa và phân phối các tệp đa phương tiện, giúp cải thiện hiệu suất và trải nghiệm người dùng.

Cloudinary cho phép người dùng lưu trữ hình ảnh và video trên đám mây, giúp giảm tải cho máy chủ và cung cấp khả năng truy cập nhanh chóng. Nền tảng này hỗ trợ nhiều tính năng biến đổi hình ảnh như thay đổi kích thước, cắt, xoay, và áp dụng bộ lọc, giúp tạo ra các phiên bản khác nhau của tệp gốc. Ngoài ra, Cloudinary tự động tối ưu hóa hình ảnh và video để giảm kích thước tệp mà không làm giảm chất lượng, giúp cải thiện tốc độ tải trang. Nền tảng cũng cung cấp mạng phân phối nội dung (CDN) để phục vụ tệp đa phương tiện nhanh chóng từ các máy chủ gần người dùng nhất.

Cloudinary cung cấp API mạnh mẽ và dễ dàng tích hợp với nhiều ngôn ngữ lập trình và framework, từ JavaScript đến Ruby và PHP. Giao diện quản lý trực quan giúp người dùng dễ dàng tổ chức, tìm kiếm và quản lý các tệp đa phương tiện. Bên cạnh hình ảnh, Cloudinary cũng hỗ trợ video với khả năng tối ưu hóa và biến đổi tương tự như với hình ảnh. Ngoài ra, nền tảng cung cấp các báo cáo và phân tích chi tiết về cách thức sử dụng tệp đa phương tiện, giúp người dùng theo dõi hiệu suất.

Sử dụng Cloudinary mang lại nhiều ưu điểm như tăng tốc độ tải trang nhờ tối ưu hóa và phân phối nội dung qua CDN, giảm chi phí lưu trữ nhờ lưu trữ trên đám mây, và dễ dàng tích hợp vào các ứng dụng hiện có, giúp tiết kiệm thời gian và công sức phát triển.

Cloudinary là một giải pháp toàn diện cho việc quản lý và tối ưu hóa hình ảnh và video, giúp các nhà phát triển tiết kiệm thời gian và tài nguyên trong việc xử lý nội dung đa phương tiện. Với tính năng mạnh mẽ và khả năng tích hợp linh hoạt, Cloudinary trở thành lựa chọn hàng đầu cho các ứng dụng web và di động hiện đại.

**2.2. Thông tin và yêu cầu về hệ thống**

**\* Yêu cầu của hệ thống:**

Hệ thống ứng dụng mạng xã hội được xây dựng nhằm đáp ứng nhu cầu tương tác và chia sẻ thông tin giữa người dùng. Với giao diện thân thiện, hệ thống hướng đến việc cung cấp các tính năng cơ bản thường thấy trong các mạng xã hội phổ biến.

**\* Nền tảng phát triển:**

Ứng dụng được thiết kế dựa trên JavaScript và React Native, đảm bảo khả năng tương thích.

**\* Yêu cầu về chức năng:**

a. Tạo tài khoản và đăng nhập:

+ Người dùng có thể đăng ký, đăng nhập và chỉnh sửa thông tin cá nhân.

b. Tương tác bài viết:

+ Đăng bài viết, hình ảnh hoặc video.

+ Thả cảm xúc, bình luận và chia sẻ bài viết.

c. Quản lý bài viết:

+Chỉnh sửa bài viết, xóa bài.

d. Quản lý theo dõi:

+ Theo dõi bạn bè để hiện thị bài viết của nhau.

+ Hiển thị danh sách bạn bè và đề xuất bạn mới.

e. Nhắn tin:

+ Cho phép gửi tin nhắn trực tiếp giữa các người dùng.

+ Hỗ trợ hiển thị lịch sử trò chuyện.

f. Thông báo:

+ Thông báo cho người dùng về lượt thích, bình luận hoặc chia sẻ bài viết.

+ Cập nhật thông báo về yêu cầu theo dõi hoặc tin nhắn mới.

g. Đăng khoảnh khắc:

+Khoảnh khắc sẽ hiển thị của người dùng sẽ hiển thị 24h rồi biến mất.

**\*Yêu cầu về phi chức năng:**

a. Hiệu năng:

+Tối ưu hóa tốc độ tải trang và xử lý thao tác của người dùng.

b. Khả dụng:

+ Hoạt động ổn định trên cả nền tảng Android và iOS.

c. Khả năng mở rộng:

+ Có thể tích hợp thêm các tính năng như nhóm, sự kiện, hoặc livestream trong tương lai.

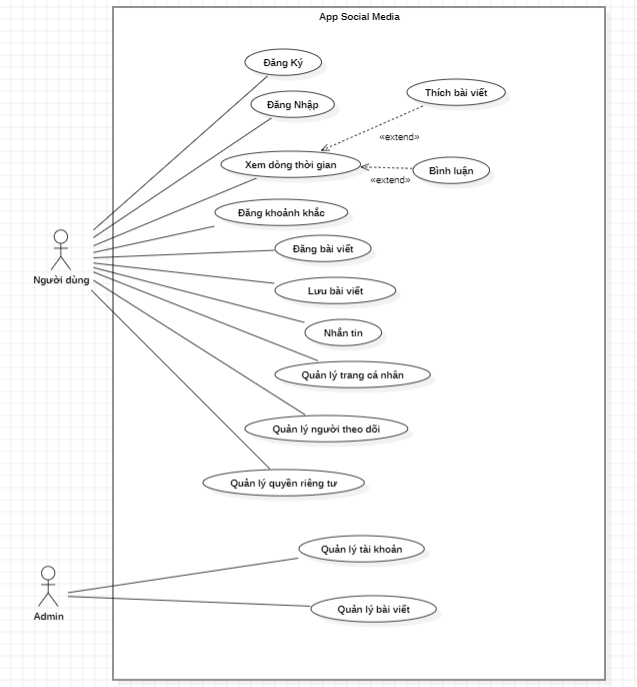
d. Thân thiện với người dùng:

+ Giao diện trực quan, dễ sử dụng cho mọi đối tượng.

Hệ thống đáp ứng các chức năng cơ bản của một mạng xã hội, đồng thời đặt nền móng để phát triển thêm trong tương lai.

**2.3. Các biểu đồ về chức năng của hệ thống**

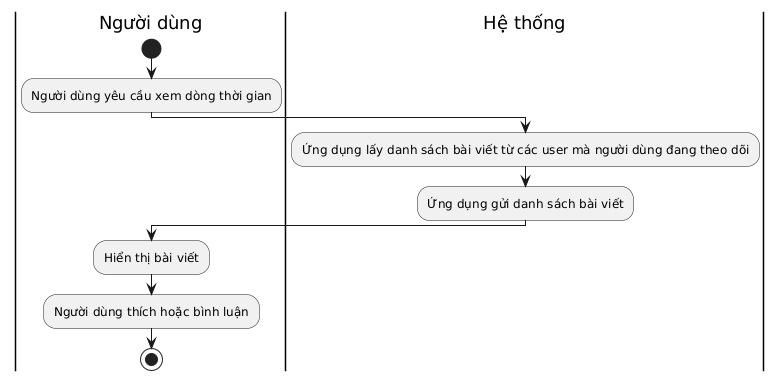
### *2.3.1.* *Use Case Diagram toàn hệ thống*



Hình 2.6. Use Case tổng quát

### *2.3.2. Activity Diagram toàn hệ thống*

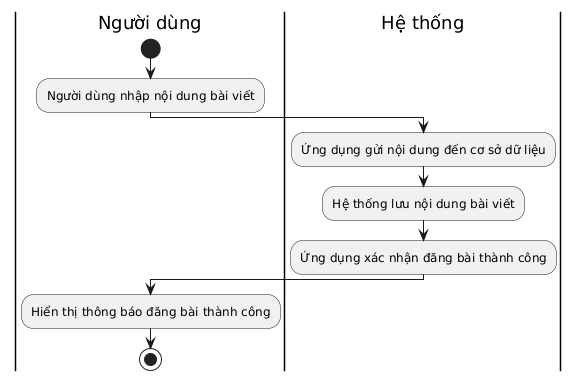
### 2.3.2.1. Activity Diagram dòng thời gian



Hình 2.7. Activity Diagram dòng thời gian

Chức năng này cho phép người dùng xem dòng thời gian các bài viết từ những người mà họ đang theo dõi. Hệ thống sẽ lấy và hiển thị các bài viết mới nhất, sau đó người dùng có thể tương tác bằng cách thích hoặc bình luận vào các bài viết này.

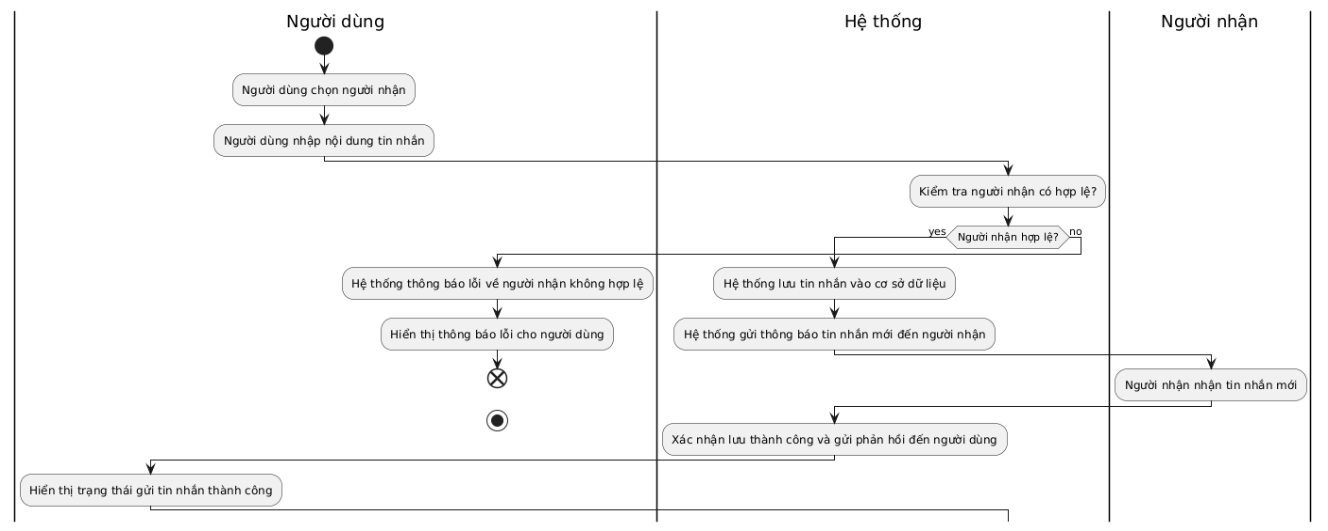
### 2.3.2.2. Activity Diagram đăng bài viết



Hình 2.8. Activity Diagram đăng bài viết

Chức năng đăng bài viết cho phép người dùng tạo và chia sẻ nội dung trên hệ thống. Người dùng sẽ nhập nội dung bài viết, có thể bao gồm văn bản, hình ảnh hoặc video. Sau khi hoàn tất, ứng dụng gửi bài viết đến cơ sở dữ liệu để lưu trữ. Hệ thống sẽ lưu bài viết vào cơ sở dữ liệu và xác nhận quá trình lưu trữ thành công. Sau khi bài viết được đăng thành công, ứng dụng sẽ hiển thị thông báo cho người dùng, thông báo rằng bài viết đã được đăng lên thành công và có thể được xem hoặc tương tác bởi người khác.

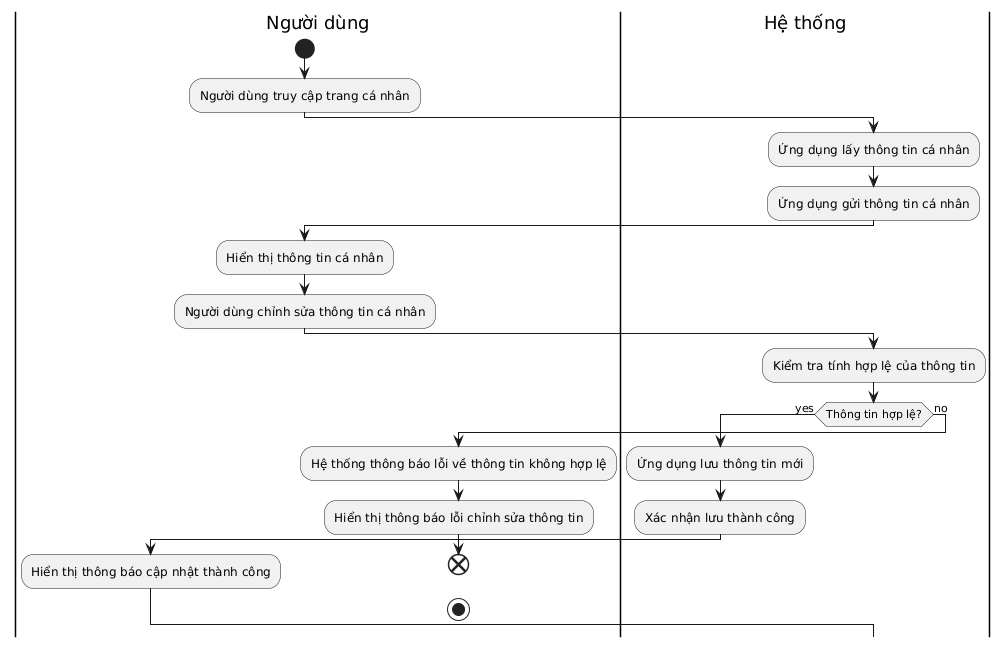
### 2.3.2.3. Activity Diagram nhắn tin



Hình 2.9. Activity Diagram nhắn tin

Chức năng gửi tin nhắn cho phép người dùng gửi tin nhắn đến một người nhận đã được chọn. Người dùng sẽ chọn người nhận và nhập nội dung tin nhắn. Hệ thống sẽ kiểm tra xem người nhận có hợp lệ không (ví dụ như người nhận có tài khoản hoặc có thể nhận tin nhắn). Nếu người nhận hợp lệ, hệ thống sẽ lưu tin nhắn vào cơ sở dữ liệu và gửi thông báo tin nhắn mới đến người nhận. Sau đó, người nhận sẽ nhận được tin nhắn và hệ thống sẽ xác nhận quá trình lưu tin nhắn thành công, gửi phản hồi về trạng thái tin nhắn cho người gửi. Người gửi sẽ nhận thông báo rằng tin nhắn đã được gửi thành công. Nếu người nhận không hợp lệ, hệ thống sẽ thông báo lỗi và người gửi sẽ nhận được thông báo rằng người nhận không hợp lệ, tin nhắn không thể gửi đi.

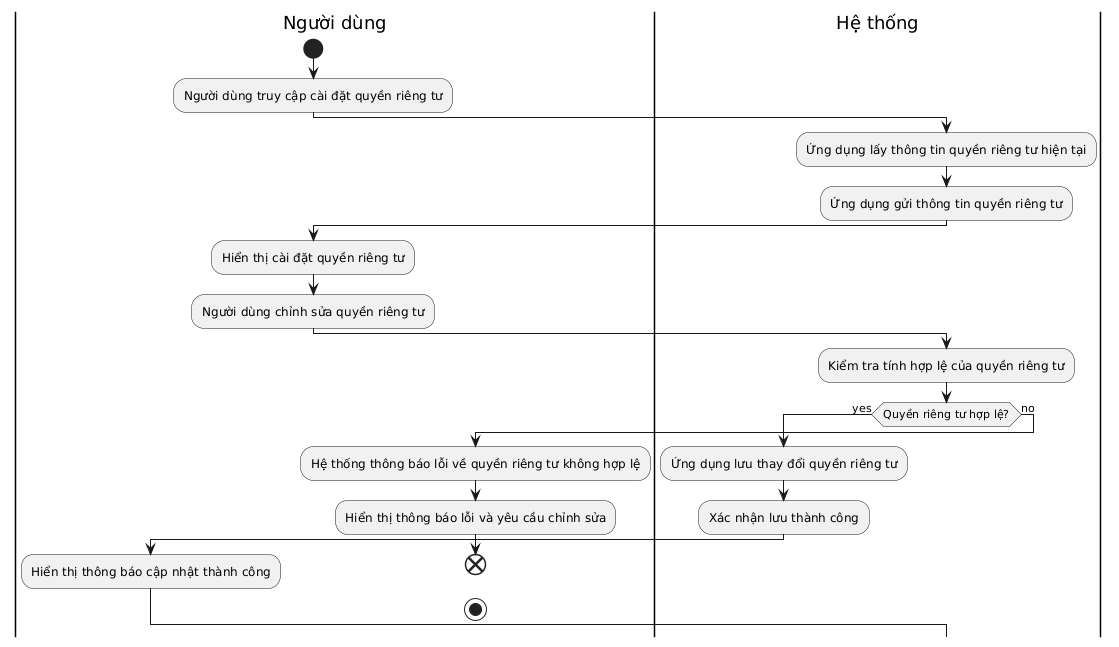
### 2.3.2.4. Activity Diagram quản lý trang cá nhân



Hình 2.10. Activity Diagram quản lý trang cá nhân

Chức năng quản lý trang cá nhân cho phép người dùng truy cập và chỉnh sửa các thông tin cá nhân của mình trên hệ thống. Khi người dùng truy cập vào trang cá nhân, ứng dụng sẽ lấy và hiển thị các thông tin cá nhân hiện tại của họ, bao gồm tên, ảnh đại diện, địa chỉ email, và các thông tin liên hệ khác. Người dùng có thể chỉnh sửa các thông tin này theo ý muốn. Sau khi người dùng thực hiện thay đổi, hệ thống sẽ kiểm tra tính hợp lệ của các thông tin mới (ví dụ: kiểm tra định dạng email, kích thước ảnh, v.v.). Nếu thông tin hợp lệ, hệ thống sẽ lưu các thay đổi và gửi phản hồi xác nhận việc cập nhật thành công. Người dùng sẽ nhận được thông báo xác nhận rằng thông tin của mình đã được cập nhật. Nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu người dùng chỉnh sửa lại thông tin, giúp đảm bảo dữ liệu luôn chính xác và đầy đủ.

### 2.3.2.5. Activity Diagram quản lý quyền riêng tư

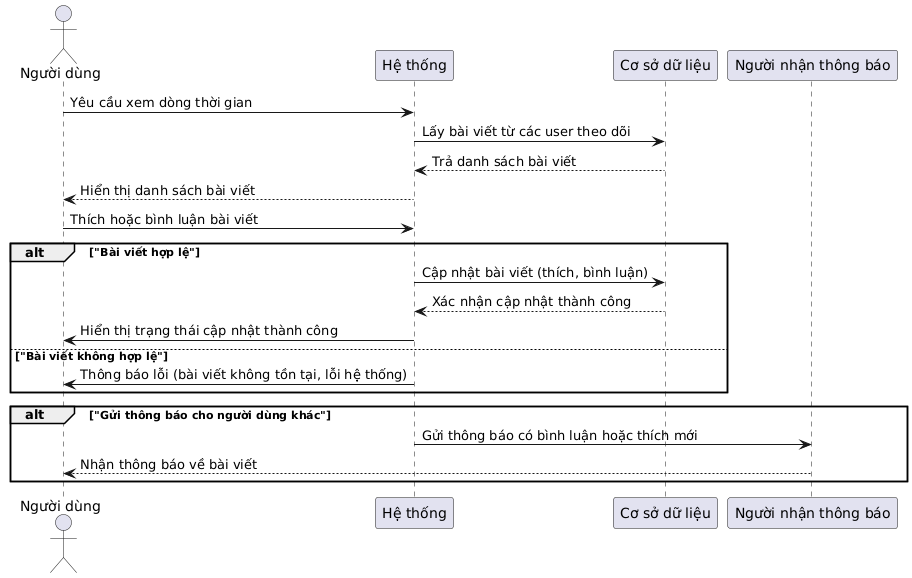


Hình 2.11. Activity Diagram quản lý quyền riêng tư

Chức năng quản lý cài đặt quyền riêng tư cho phép người dùng truy cập và điều chỉnh các cài đặt liên quan đến bảo mật và quyền riêng tư trong ứng dụng. Khi người dùng vào phần cài đặt quyền riêng tư, hệ thống sẽ lấy và hiển thị các cài đặt hiện tại, bao gồm các lựa chọn như ai có thể xem thông tin cá nhân của họ, ai có thể liên hệ với họ, và các tùy chọn khác liên quan đến bảo mật. Người dùng có thể thay đổi các cài đặt này theo ý muốn để kiểm soát mức độ chia sẻ thông tin và bảo vệ dữ liệu cá nhân của mình. Sau khi người dùng chỉnh sửa các cài đặt quyền riêng tư, hệ thống sẽ kiểm tra tính hợp lệ của các thay đổi, đảm bảo chúng không vi phạm các quy định bảo mật. Nếu các cài đặt hợp lệ, hệ thống sẽ lưu chúng và xác nhận cho người dùng biết rằng cài đặt đã được cập nhật thành công. Nếu có lỗi hoặc các cài đặt không hợp lệ, hệ thống sẽ thông báo lỗi và yêu cầu người dùng chỉnh sửa lại các thông tin để đảm bảo tính bảo mật. Chức năng này giúp người dùng kiểm soát quyền riêng tư của mình một cách dễ dàng và an toàn.

### *2.3.3. Sequence Diagram toàn hệ thống*

### 2.3.3.1. Sequence Diagram dòng thời gian



Hình 2.12. Sequence Diagram dòng thời gian

Chức năng này mô tả quy trình người dùng tương tác với dòng thời gian và bài viết trên hệ thống. Khi người dùng yêu cầu xem dòng thời gian, hệ thống sẽ truy xuất các bài viết từ những người mà người dùng đang theo dõi trong cơ sở dữ liệu và hiển thị danh sách bài viết. Người dùng có thể tương tác với bài viết bằng cách thích hoặc bình luận.

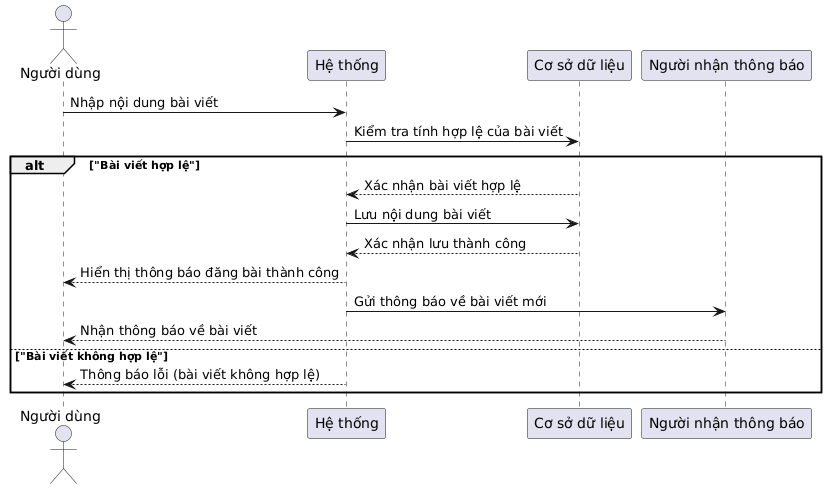
Nếu bài viết hợp lệ (ví dụ, vẫn tồn tại và có thể tương tác), hệ thống sẽ cập nhật bài viết trong cơ sở dữ liệu với thông tin thích hoặc bình luận của người dùng và trả về xác nhận việc cập nhật thành công. Hệ thống sẽ hiển thị trạng thái cập nhật thành công cho người dùng.

Trong trường hợp bài viết không hợp lệ (ví dụ, bài viết đã bị xóa hoặc có lỗi hệ thống), hệ thống sẽ thông báo lỗi cho người dùng.

Ngoài ra, nếu có hành động tương tác (thích hoặc bình luận), hệ thống có thể gửi thông báo đến người dùng khác (người nhận thông báo) để họ biết về bình luận hoặc lượt thích mới trên bài viết của mình. Người nhận thông báo sẽ nhận được thông báo về sự tương tác này.

Quy trình này giúp người dùng tương tác với các bài viết trên dòng thời gian của mình, đồng thời cung cấp các phản hồi thích hợp về trạng thái của bài viết và thông báo cho những người liên quan.

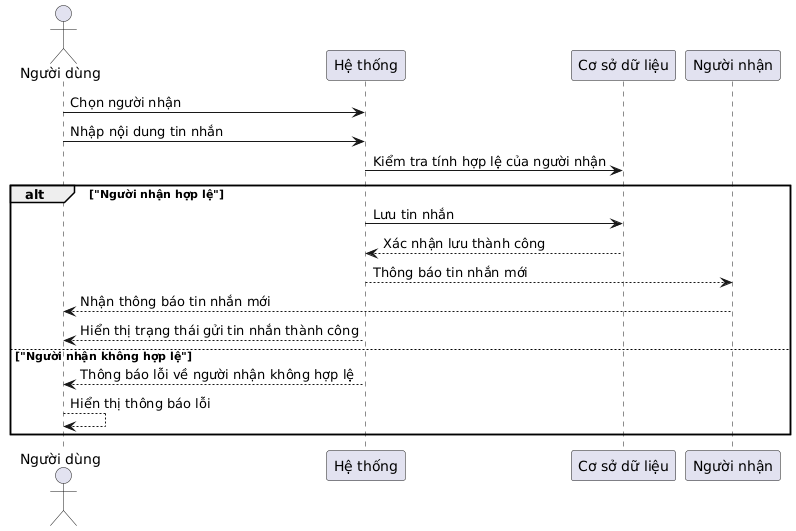
### 2.3.3.2. Sequence Diagram đăng bài viết

  
Hình 2.13. Sequence Diagram đăng bài viết

Chức năng đăng bài viết cho phép người dùng tạo và chia sẻ nội dung lên hệ thống. Khi người dùng nhập nội dung bài viết, hệ thống sẽ kiểm tra tính hợp lệ của bài viết, đảm bảo rằng nó không vi phạm các quy định của hệ thống, chẳng hạn như không chứa nội dung không phù hợp hoặc spam. Nếu bài viết hợp lệ, hệ thống sẽ lưu nội dung bài viết vào cơ sở dữ liệu và xác nhận việc lưu thành công. Sau khi bài viết được đăng thành công, người dùng sẽ nhận được thông báo xác nhận rằng bài viết của họ đã được đăng.

Bên cạnh đó, hệ thống cũng sẽ gửi thông báo về bài viết mới tới những người dùng khác (những người theo dõi hoặc có liên quan), giúp bài viết nhanh chóng được chia sẻ và tiếp cận với nhiều người hơn. Nếu bài viết không hợp lệ, hệ thống sẽ thông báo lỗi cho người dùng, yêu cầu họ chỉnh sửa bài viết trước khi thử đăng lại. Quy trình này giúp người dùng có thể chia sẻ nội dung dễ dàng, đồng thời bảo vệ hệ thống khỏi các bài viết vi phạm chính sách.

### 2.3.3.3. Sequence Diagram nhắn tin

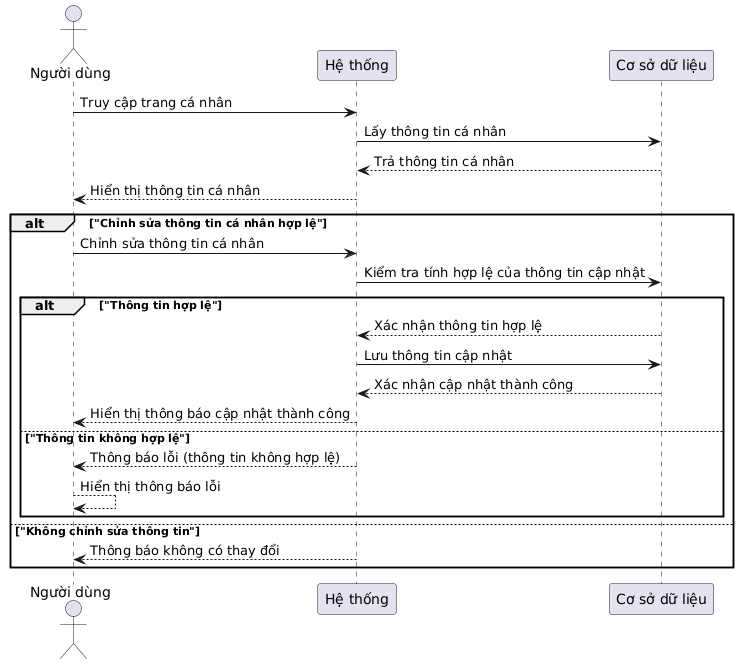


Hình 2.14. Sequence Diagram nhắn tin

Chức năng gửi tin nhắn cho phép người dùng gửi tin nhắn đến một người nhận cụ thể. Quá trình bắt đầu khi người dùng chọn người nhận và nhập nội dung tin nhắn. Hệ thống sẽ kiểm tra tính hợp lệ của người nhận (ví dụ, người nhận có tài khoản hợp lệ, có thể nhận tin nhắn hay không). Nếu người nhận hợp lệ, hệ thống sẽ lưu tin nhắn vào cơ sở dữ liệu và gửi thông báo tin nhắn mới đến người nhận. Người nhận sẽ nhận được thông báo về tin nhắn mới và có thể đọc hoặc phản hồi.

Sau khi tin nhắn được lưu thành công, hệ thống sẽ hiển thị trạng thái gửi tin nhắn thành công cho người dùng, xác nhận rằng tin nhắn đã được gửi đi. Trong trường hợp người nhận không hợp lệ (ví dụ, người nhận không có tài khoản hoặc không thể nhận tin nhắn), hệ thống sẽ thông báo lỗi cho người gửi và yêu cầu họ chọn lại người nhận hợp lệ. Người dùng sẽ nhận được thông báo lỗi và có thể sửa lại thông tin người nhận để gửi lại tin nhắn. Quy trình này đảm bảo rằng người dùng có thể gửi tin nhắn một cách chính xác và an toàn, đồng thời cung cấp phản hồi kịp thời về trạng thái của tin nhắn.

### 2.3.3.4. Sequence Diagram quản lí trang cá nhân



Hình 2.15. Sequence Diagram quản lí trang cá nhân

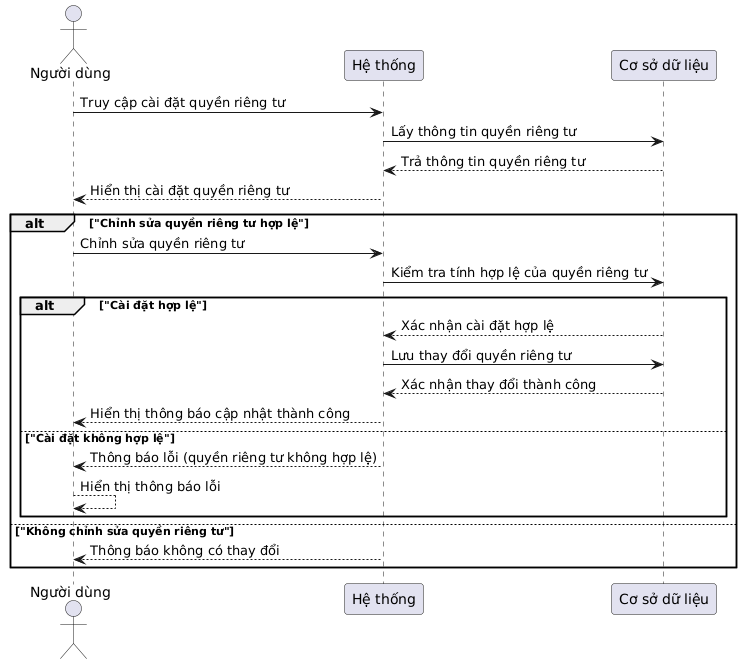
Chức năng quản lý trang cá nhân cho phép người dùng xem và chỉnh sửa thông tin cá nhân của mình trên hệ thống. Quá trình bắt đầu khi người dùng truy cập trang cá nhân, hệ thống sẽ lấy thông tin cá nhân hiện tại từ cơ sở dữ liệu và hiển thị cho người dùng. Nếu người dùng quyết định chỉnh sửa thông tin, hệ thống sẽ kiểm tra tính hợp lệ của các thay đổi.

Nếu thông tin người dùng nhập vào hợp lệ (ví dụ: định dạng email đúng, ảnh đại diện đúng kích thước, v.v.), hệ thống sẽ lưu các thay đổi vào cơ sở dữ liệu và thông báo cho người dùng rằng việc cập nhật thông tin cá nhân đã thành công.

Trong trường hợp thông tin không hợp lệ (ví dụ: email không đúng định dạng, ảnh không phù hợp), hệ thống sẽ thông báo lỗi và yêu cầu người dùng chỉnh sửa lại thông tin. Nếu người dùng không thay đổi bất kỳ thông tin nào, hệ thống sẽ thông báo rằng không có thay đổi nào được thực hiện.

Quy trình này giúp người dùng dễ dàng quản lý và cập nhật thông tin cá nhân của mình, đồng thời bảo đảm rằng các thay đổi phải hợp lệ và được xử lý đúng cách.

### 2.3.3.5. Sequence Diagram quản lý quyền riêng tư



Hình 2.16. Sequence Diagram quản lý quyền riêng tư

Chức năng quản lý cài đặt quyền riêng tư cho phép người dùng truy cập và điều chỉnh các cài đặt bảo mật liên quan đến thông tin cá nhân và quyền truy cập của những người khác trong hệ thống. Quá trình bắt đầu khi người dùng truy cập phần cài đặt quyền riêng tư, lúc này hệ thống sẽ lấy thông tin quyền riêng tư hiện tại từ cơ sở dữ liệu và hiển thị cho người dùng.

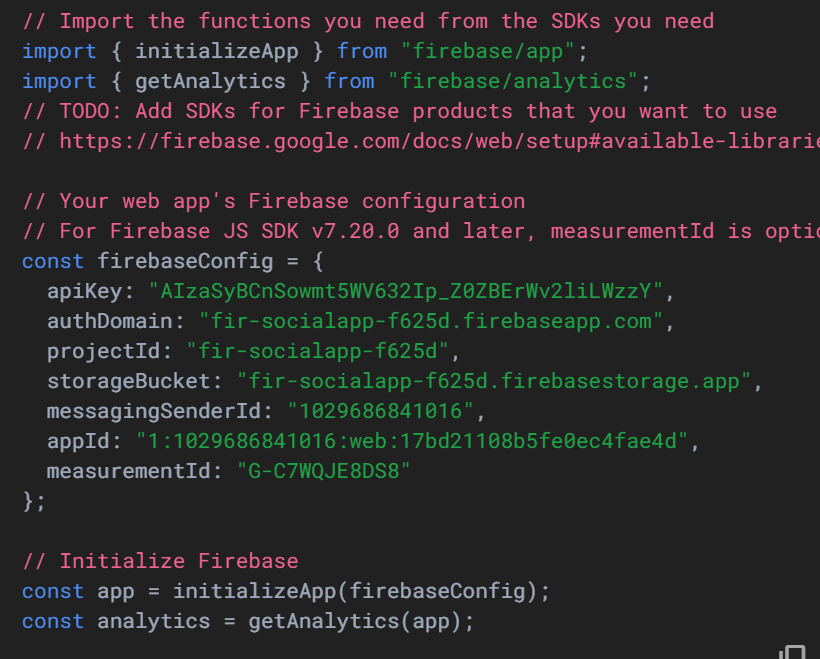
Nếu người dùng quyết định thay đổi các cài đặt quyền riêng tư, hệ thống sẽ kiểm tra tính hợp lệ của các cài đặt mới, đảm bảo rằng chúng không vi phạm các chính sách bảo mật của hệ thống. Nếu các thay đổi hợp lệ (ví dụ: người dùng chỉ định đúng mức độ chia sẻ thông tin, chọn quyền truy cập đúng cho các nhóm người), hệ thống sẽ lưu các thay đổi vào cơ sở dữ liệu và xác nhận việc cập nhật thành công. Sau đó, hệ thống sẽ hiển thị thông báo cho người dùng rằng các cài đặt quyền riêng tư đã được cập nhật thành công.

Nếu cài đặt quyền riêng tư không hợp lệ (ví dụ: cài đặt không hợp lệ hoặc không được phép), hệ thống sẽ thông báo lỗi cho người dùng và yêu cầu họ chỉnh sửa lại các thông tin. Nếu người dùng không thực hiện bất kỳ thay đổi nào đối với quyền riêng tư, hệ thống sẽ thông báo rằng không có thay đổi được thực hiện.

Chức năng này giúp người dùng có thể dễ dàng quản lý và bảo vệ quyền riêng tư của mình, đồng thời đảm bảo rằng các thay đổi đều hợp lệ và không ảnh hưởng đến bảo mật thông tin cá nhân.

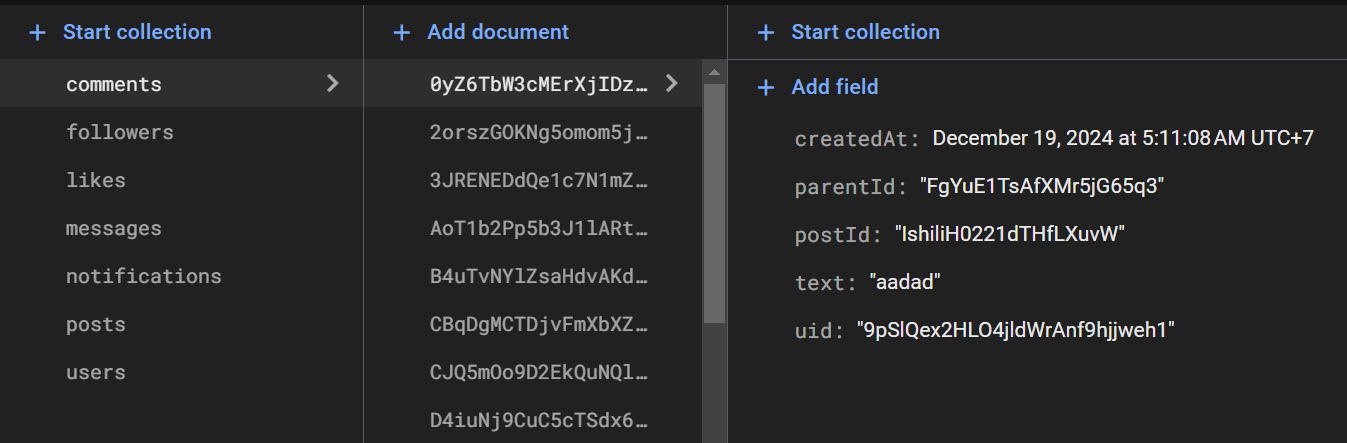
## 2.4. Cơ sở dữ liệu FireBase

### *2.4.1. API của FireBase*



Hình 2.17. API FireBase

### *2.4.2. Collection*



Hình 2.18. Collection

* Users (Người dùng): Lưu trữ thông tin cá nhân của người dùng.
  + uid (String): ID người dùng (do Firebase cấp).
  + username (String): Tên người dùng (unique).
  + fullName (String): Họ và tên người dùng.
  + email (String): Email người dùng.
  + profilePicture (String): URL ảnh đại diện.
  + bio (String): Tiểu sử người dùng.
  + followersCount (Number): Số người theo dõi.
  + followingCount (Number): Số người mà người dùng đang theo dõi.
  + createdAt (Timestamp): Thời gian tạo tài khoản.
  + updatedAt (Timestamp): Thời gian cập nhật hồ sơ.
  + isPrivate (Boolean): Trạng thái tài khoản (công khai/riêng tư).
  + lastLogin (Timestamp): Thời gian người dùng cuối cùng đăng nhập.
* Posts (Bài viết): Chứa các bài viết của người dùng.
  + postId (String): ID của bài viết (unique).
  + uid (String): ID người dùng đăng bài.
  + content (String): Nội dung bài viết.
  + imageUrl (String): URL của ảnh bài viết (nếu có).
  + createdAt (Timestamp): Thời gian bài viết được đăng tải.
  + updatedAt (Timestamp): Thời gian bài viết được cập nhật.
  + likesCount (Number): Số lượt thích bài viết.
  + commentsCount (Number): Số bình luận bài viết.
* Comments (Bình luận): Lưu trữ các bình luận của người dùng.
  + commentId (String): ID bình luận (unique).
  + postId (String): ID bài viết mà bình luận thuộc về.
  + uid (String): ID người dùng bình luận.
  + content (String): Nội dung bình luận.
  + createdAt (Timestamp): Thời gian bình luận được tạo.
  + updatedAt (Timestamp): Thời gian bình luận được cập nhật.
* Likes (Lượt thích): Lưu trữ các lượt thích đối với bài viết.
  + likeId (String): ID lượt thích (unique).
  + postId (String): ID bài viết được thích.
  + uid (String): ID người dùng đã thích bài viết.
  + createdAt (Timestamp): Thời gian thích bài viết.
* Followers (Người theo dõi): Lưu trữ mối quan hệ theo dõi giữa người dùng.
  + followerId (String): ID người theo dõi.
  + followedId (String): ID người được theo dõi.
  + createdAt (Timestamp): Thời gian theo dõi.
* Messages (Tin nhắn): Lưu trữ thông tin các tin nhắn giữa người dùng.
  + messageId (String): ID tin nhắn (unique).
  + senderId (String): ID người gửi.
  + receiverId (String): ID người nhận.
  + content (String): Nội dung tin nhắn.
  + createdAt (Timestamp): Thời gian gửi tin nhắn.
* Notifications (Thông báo): Lưu trữ các thông báo hoạt động trong mạng xã hội.
  + notificationId (String): ID thông báo (unique).
  + uid (String): ID người nhận thông báo.
  + type (String): Loại thông báo (lượt thích, bình luận, theo dõi, v.v.).
  + message (String): Nội dung thông báo.
  + createdAt (Timestamp): Thời gian tạo thông báo.

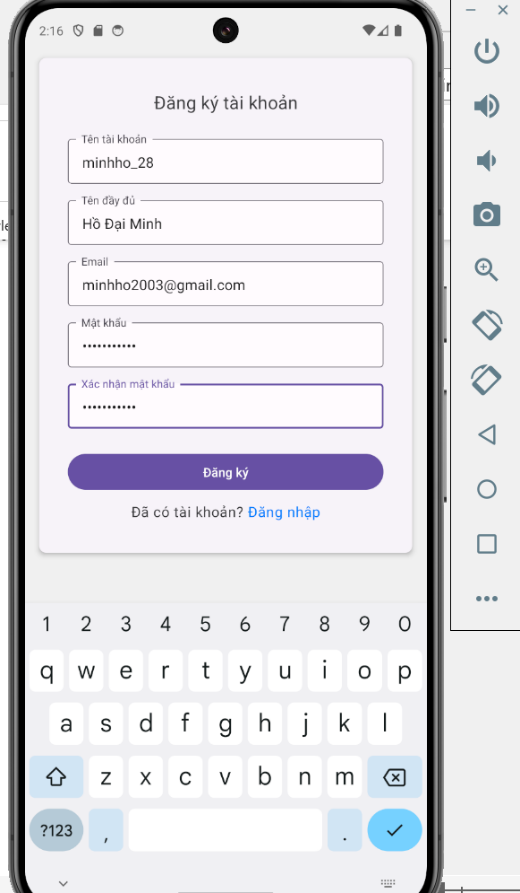
# **CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM**

## 3.1. Các thành phần chức năng của hệ thống

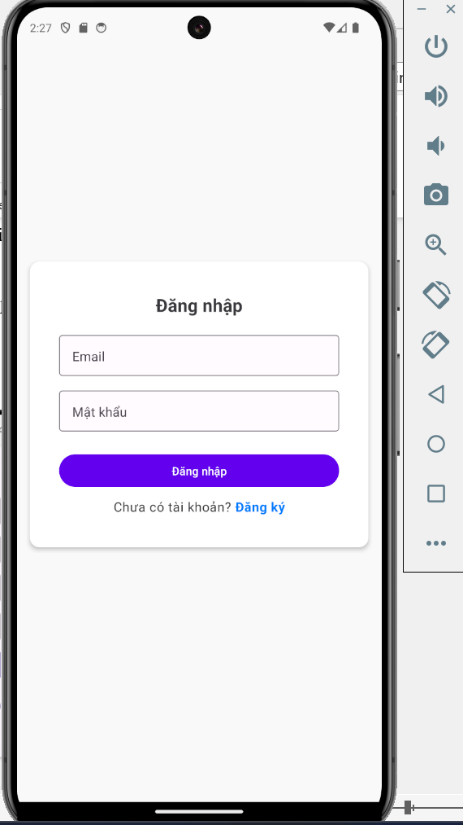
* Chức năng đăng nhập hệ thống:
  + Người dùng có thể đăng nhập bằng tài khoản cá nhân để truy cập vào các chức năng của mạng xã hội.
* Đăng xuất:
  + Người dùng có thể đăng xuất khỏi tài khoản của mình khi không còn sử dụng mạng xã hội.
* Trang chủ/Dòng thời gian:
  + Hiển thị các bài viết từ người dùng mà bạn đang theo dõi.
  + Cho phép người dùng thích, bình luận, chia sẻ các bài viết.
* Quản lý hồ sơ cá nhân:
  + Người dùng có thể xem và chỉnh sửa thông tin cá nhân như tên, ảnh đại diện, tiểu sử và các thông tin khác.
* Quản lý người theo dõi:
  + Tìm kiếm và theo dõi người dùng khác.
  + Quản lý danh sách bạn bè, theo dõi và hủy theo dõi.
* Chức năng đăng bài:
  + Người dùng có thể tạo và đăng bài viết mới, bao gồm nội dung văn bản và ảnh.
  + Người dùng có thể chỉnh sửa và xóa bài viết đã đăng.
* Chức năng tin nhắn:
  + Gửi và nhận tin nhắn trực tiếp giữa các người dùng.
* Chức năng thông báo:
  + Người dùng nhận được thông báo về các hoạt động mới liên quan đến mình, như lượt thích, bình luận, yêu cầu theo dõi, tin nhắn mới.
* Cài đặt tài khoản:
  + Người dùng có thể thay đổi các cài đặt liên quan đến tài khoản của mình, bao gồm cài đặt quyền riêng tư, thông báo, mật khẩu, ảnh đại diện.
* Liệt kê danh sách bài viết:
  + Hiển thị danh sách bài viết của người dùng và những người mà họ theo dõi.
* Cấu hình quyền riêng tư:
  + Người dùng có thể thay đổi cài đặt quyền riêng tư của tài khoản, xác định ai có thể xem bài viết và thông tin cá nhân của mình.

**3.2. Demo**

### *3.2.1. Giao diện đăng ký và đăng nhập*

   
Hình 3.1. Giao diện đăng ký

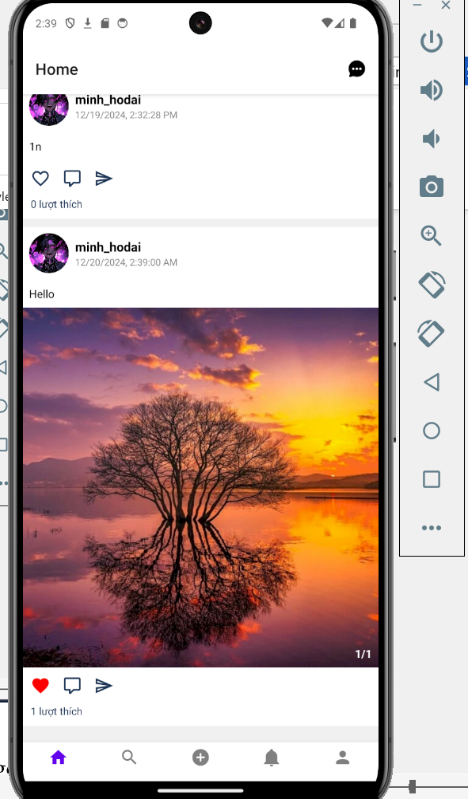
Mô tả: Người dùng chưa có tài khoản cần thực hiện đăng ký để truy cập hệ thống. Quá trình đăng ký yêu cầu cung cấp đầy đủ các thông tin: tên tài khoản, họ tên đầy đủ, email, mật khẩu và xác nhận mật khẩu. Tên tài khoản tối đa 29 ký tự, chỉ chấp nhận chữ thường, số, dấu chấm, gạch dưới và phải là duy nhất. Email được sử dụng để tạo một tài khoản duy nhất, không được trùng lặp. Mật khẩu phải đảm bảo an toàn với tối thiểu 8 ký tự, bao gồm ít nhất một chữ cái in hoa, một chữ cái in thường, một chữ số và một ký tự đặc biệt. Hệ thống kiểm tra tính hợp lệ của thông tin trước khi xác nhận đăng ký thành công.



Hình 3.2. Giao diện đăng nhập

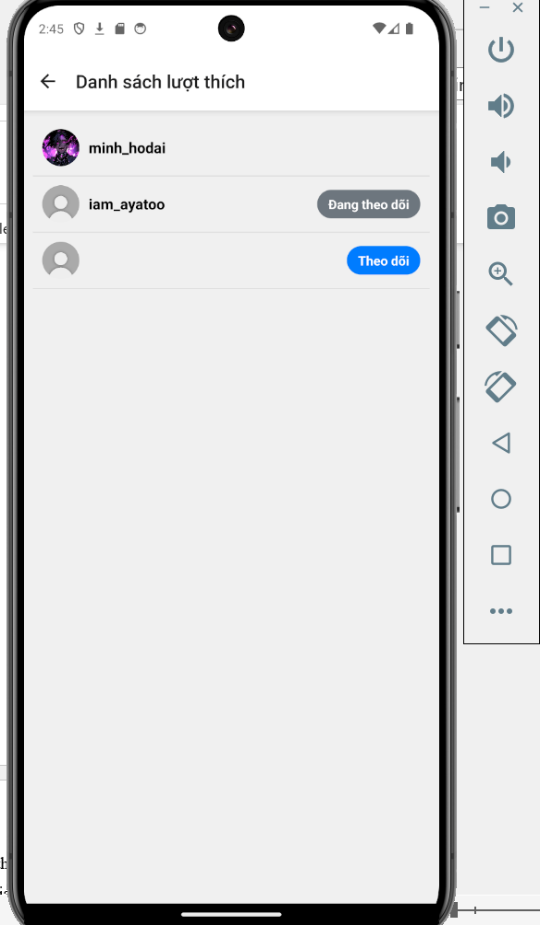
Mô tả: Giao diện đăng nhập được thiết kế đơn giản, gồm hai trường chính để người dùng nhập thông tin: Email và Mật khẩu. Trường Email yêu cầu nhập địa chỉ email hợp lệ đã đăng ký, trong khi trường Mật khẩu cho phép nhập mật khẩu tương ứng và hỗ trợ tùy chọn hiển thị mật khẩu.

### *3.2.2. Giao diện trang chủ*



Hình 3.3. Giao diện trang chủ

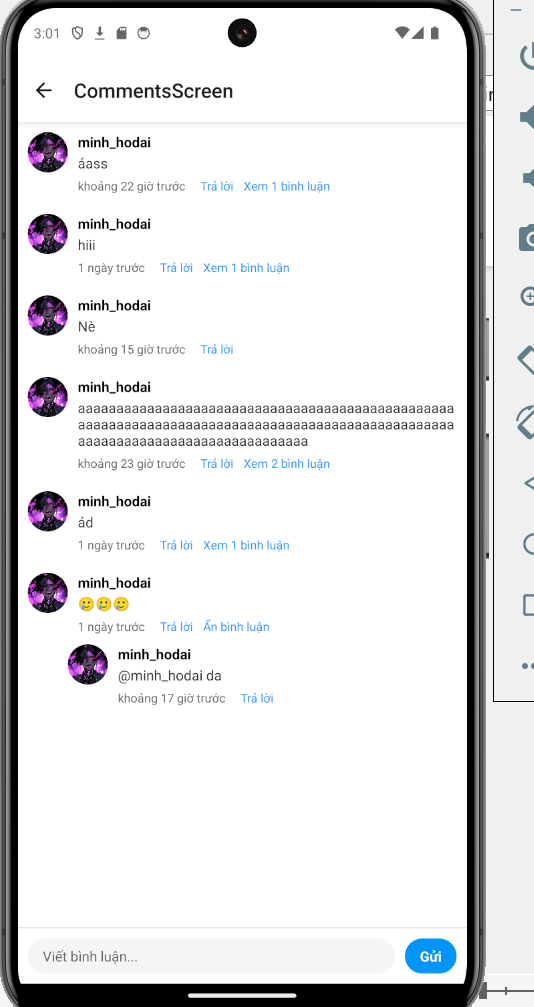
Mô tả: Giao diện trang chủ được thiết kế hiện đại và thân thiện, bao gồm dòng thời gian hiển thị các bài viết từ người dùng mà bạn theo dõi, sắp xếp theo thứ tự thời gian hoặc mức độ tương tác. Phần đầu trang có thanh điều hướng với các mục như Trang chủ, Tìm kiếm, Thông báo, Tin nhắn, và Hồ sơ cá nhân. Người dùng có thể tạo bài viết mới thông qua khu vực nhập nội dung nằm trên cùng của dòng thời gian. Người dùng có thể tim, bình luận những bài viết mà họ thích.



Hình 3.4. Giao diện danh sách lượt thích

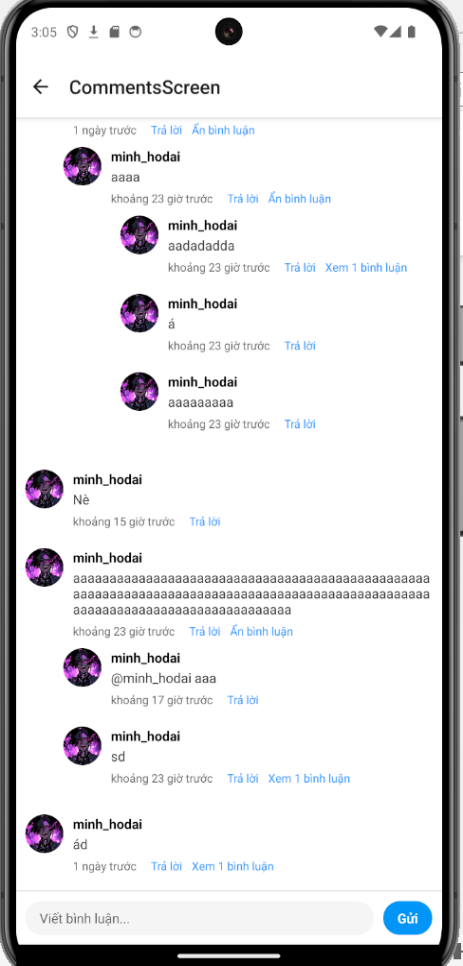
Mô tả: Hiển thị danh sách những người đã thích bài viết mà người dùng đã chọn xem.

### *3.2.3. Giao diện bình luận*



Hình 3.5. Giao diện bình luận

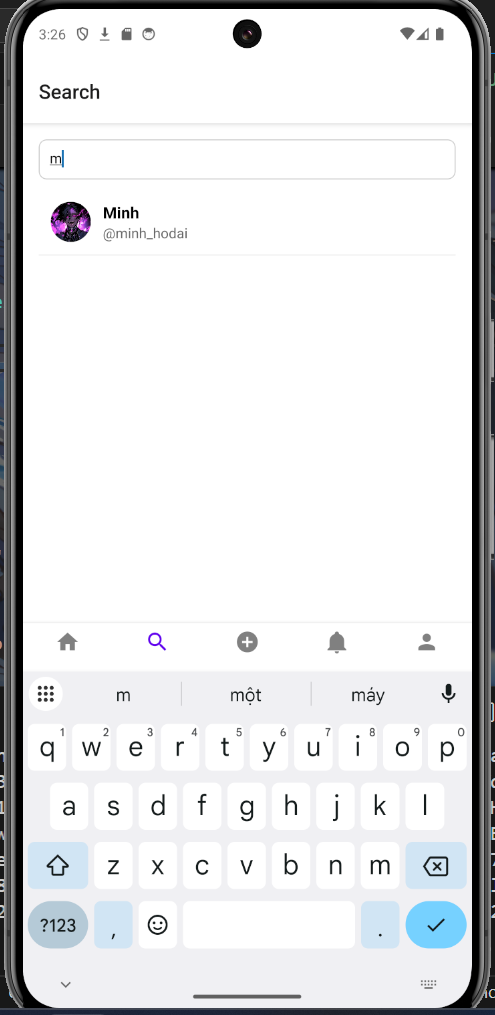
Mô tả: Khi người dùng nhấn vào biểu tượng bình luận, hệ thống chuyển sang giao diện bình luận, nơi hiển thị toàn bộ các bình luận liên quan đến bài viết. Người dùng có thể nhập nội dung bình luận mới vào ô nhập liệu, với tùy chọn tag bạn bè bằng ký tự **@** và chọn từ danh sách gợi ý. Các bình luận hiện tại được sắp xếp theo thứ tự thời gian hoặc mức độ tương tác, và người dùng có thể trả lời trực tiếp các bình luận cụ thể. Giao diện này được thiết kế tối ưu để hỗ trợ thảo luận dễ dàng và tăng cường sự kết nối giữa các thành viên.



Hình 3.6. Xem các bình luận ẩn bớt

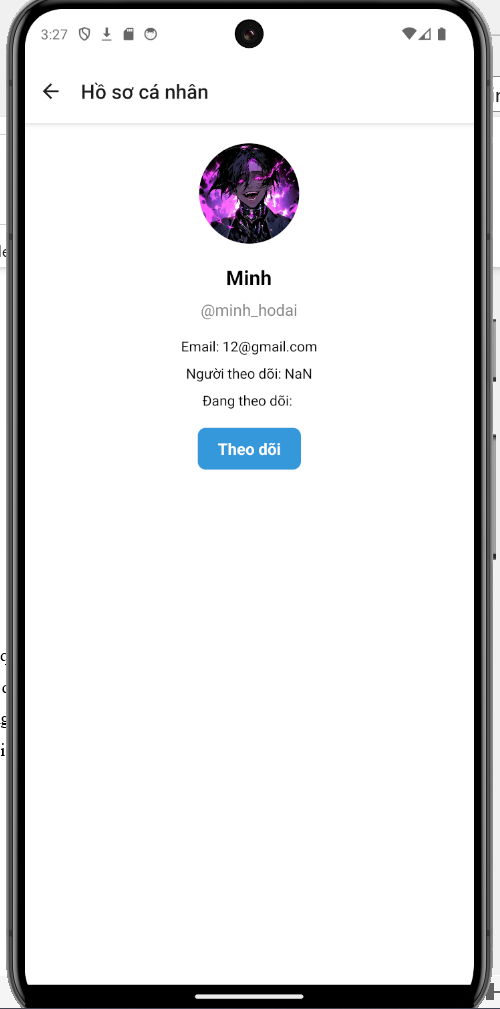
Mô tả: Khi người dùng nhấn vào tùy chọn Xem bình luận, giao diện sẽ mở rộng và hiển thị toàn bộ các bình luận đã bị ẩn trước đó, cho phép người dùng đọc thêm và tương tác đầy đủ. Ngược lại, khi người dùng chọn Ẩn, danh sách bình luận sẽ được thu gọn, chỉ giữ lại một số bình luận gần đây hoặc nổi bật, giúp giao diện trở nên gọn gàng hơn. Tính năng này được thiết kế nhằm tối ưu hóa trải nghiệm người dùng, vừa đảm bảo hiển thị thông tin cần thiết, vừa duy trì sự ngăn nắp trên giao diện.

### *3.2.4. Giao diện tìm kiếm người dùng*



Hình 3.7. Giao diện tìm kiếm người dùng

Mô tả: Giao diện tìm kiếm người dùng được thiết kế đơn giản và trực quan. Ở phần trên, có thanh tìm kiếm cho phép người dùng nhập tên hoặc thông tin liên quan để tìm kiếm. Khi nhập, hệ thống tự động gợi ý danh sách người dùng phù hợp ngay bên dưới, bao gồm hình đại diện, tên đầy đủ, và tên tài khoản. Kết quả tìm kiếm chi tiết được hiển thị trong một danh sách.



Hình 3.8. Giao diện tìm kiếm người dùng

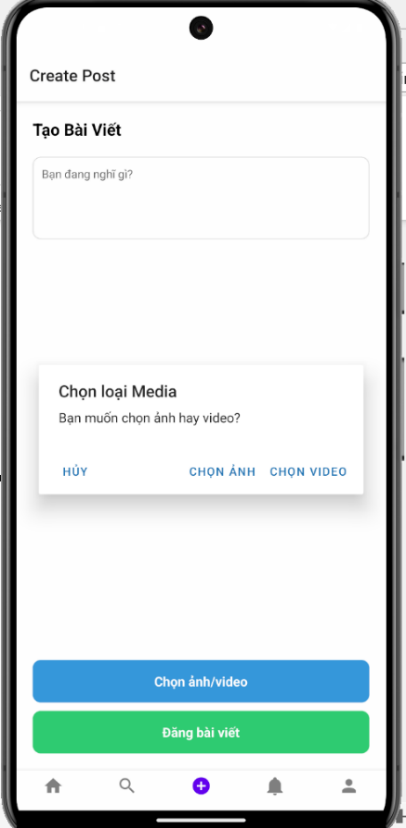
Mô tả: Sau khi thực hiện tìm kiếm, khi người dùng nhấn vào một kết quả trong danh sách, hệ thống sẽ mở hồ sơ cá nhân của người dùng được chọn. Giao diện hồ sơ hiển thị đầy đủ thông tin như ảnh đại diện, tên đầy đủ, tên tài khoản, số lượng người theo dõi và đang theo dõi, cùng với phần giới thiệu ngắn gọn (nếu có). Ngoài ra, người dùng có thể xem các bài viết đã đăng, nhấn nút Theo dõi hoặc Hủy theo dõi, và gửi tin nhắn trực tiếp. Giao diện này cho phép người dùng tương tác nhanh chóng và khám phá thêm về người dùng khác trong hệ thống.

### *3.2.5. Giao diện tạo bài viết*

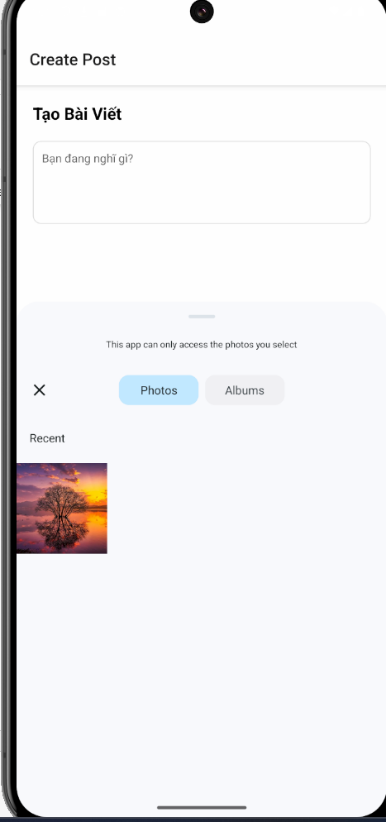


Hình 3.9. Giao diện tạo bải viết

Mô tả: Giao diện tạo bài viết được thiết kế đơn giản và dễ sử dụng, với ô nhập nội dung lớn ngay trung tâm để người dùng có thể dễ dàng viết và chia sẻ cảm nghĩ. Phía dưới ô nhập nội dung là các tùy chọn như Chọn ảnh.



Hình 3.10. Thông báo chọn ảnh hay video

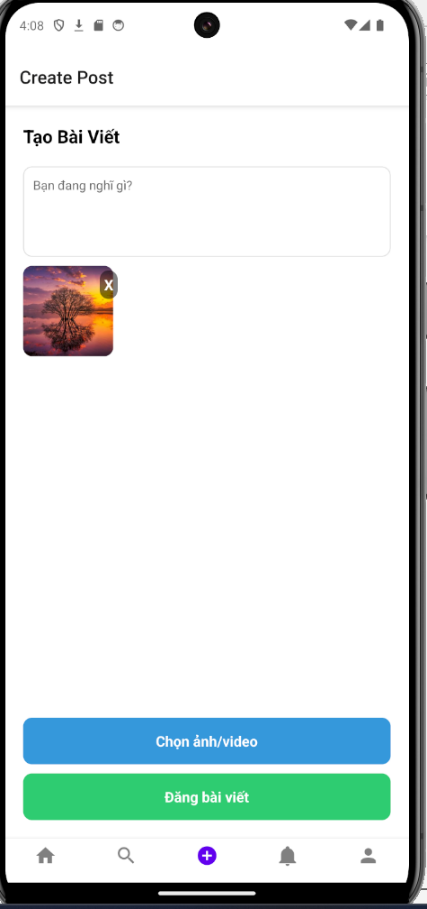


Hình 3.11. Mở thư viện ảnh để chọn ảnh



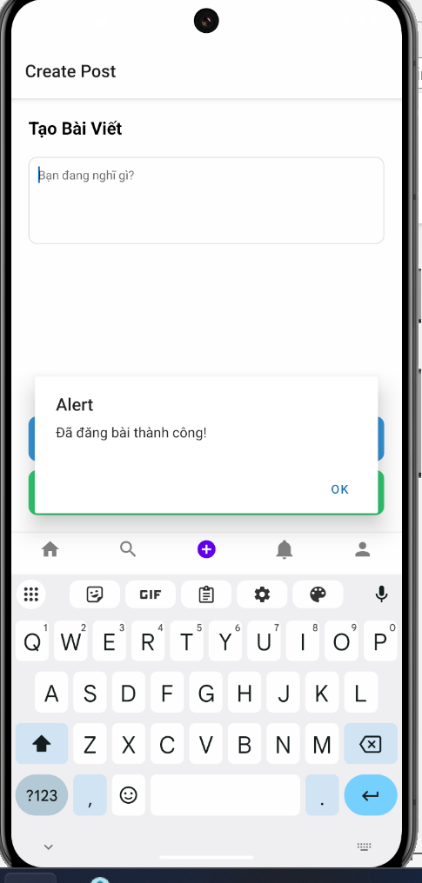
Hình 3.12. Cắt ảnh 3x3

Mô tả: Cắt ảnh 3x3 để các ảnh của bài viết đồng bộ với nhau.

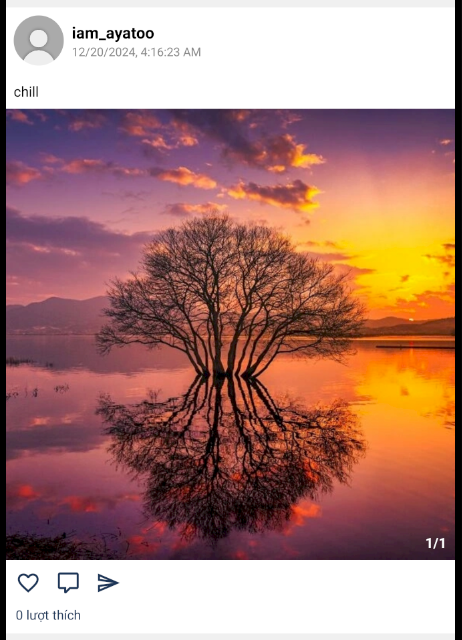


Hình 3.13. Giao diện sau khi chọn xong ảnh

Mô tả: Ảnh chọn xong sẽ nằm dưới phần cảm nghĩ để người dùng kiểm xem chọn có đúng ảnh chưa và bấm x để hủy chọn ảnh.



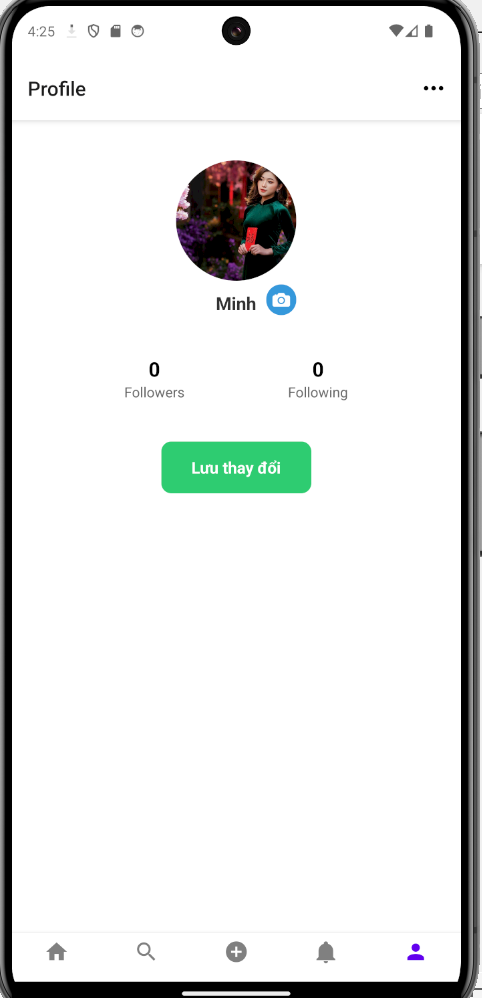
Hình 3.14. Hiện thông báo



Hình 3.15. Giao diện bài viết

Mô tả: Đây là giao diên của bài viết mà người dùng vừa đăng.

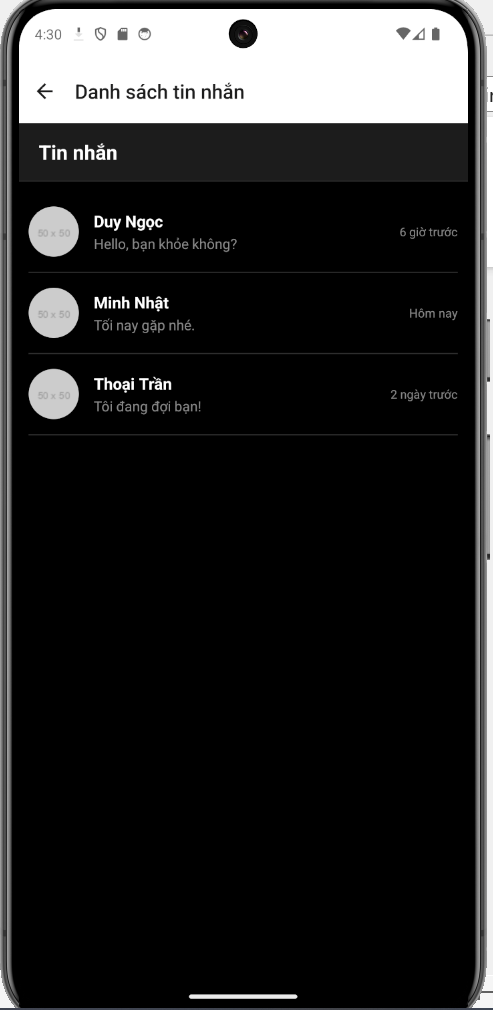
### *3.2.6. Giao diện trang cá nhân*

**

Hình 3.16. Giao diện trang cá nhân

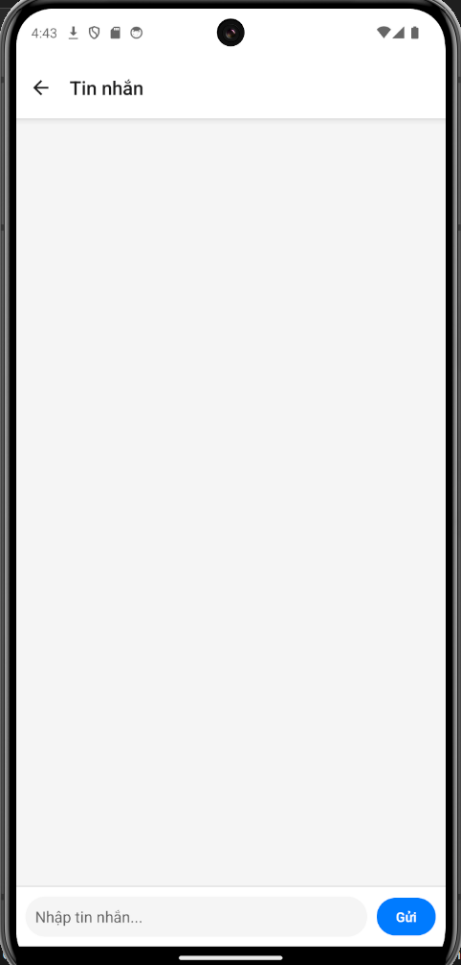
Mô tả: Giao diện Trang cá nhân của người dùng hiển thị các thông tin cơ bản và nội dung mà người dùng chia sẻ. Phần trên cùng của trang bao gồm ảnh đại diện, tên đầy đủ, tên tài khoản, và một đoạn giới thiệu ngắn mà người dùng có thể tùy chỉnh.

### *3.2.6. Giao diện trang tin nhắn*



Hình 3.17. Giao diện danh sách tin nhắn

Mô tả: Giao diện Danh sách tin nhắn hiển thị một danh sách các cuộc trò chuyện của người dùng với các bạn bè. Mỗi mục trong danh sách bao gồm tên người, ảnh đại diện của người đó, và một đoạn tin nhắn gần nhất để người dùng dễ dàng nhận diện cuộc trò chuyện nào là mới nhất.



Hình 3.18. Giao diện tin nhắn

Mô tả: Giao diện Tin nhắn hiển thị cuộc trò chuyện chi tiết giữa người dùng và một người. Đây là nơi người dùng có thể xem và tương tác với các tin nhắn, gửi tin nhắn mới và thực hiện các hành động khác.

# 

# **CHƯƠNG 4. KẾT LUẬN VÀ ĐÁNH GIÁ**

## 4.1. Kết luận

Trong ngữ cảnh hiện nay, mạng xã hội đóng vai trò quan trọng trong việc kết nối cộng đồng, chia sẻ thông tin và tạo cơ hội giao lưu giữa các cá nhân và tổ chức. Nhu cầu xây dựng các ứng dụng mạng xã hội không chỉ đòi hỏi sự tối ưu hóa về mặt hiệu suất mà còn cần đảm bảo tính bảo mật và trải nghiệm người dùng.

Việc sử dụng các công nghệ như React Native, Firebase, và Node.js để phát triển ứng dụng mạng xã hội đã chứng minh được tính khả thi và hiệu quả. React Native cung cấp khả năng phát triển nhanh chóng, giao diện người dùng thân thiện, và khả năng triển khai đa nền tảng. Firebase hỗ trợ quản lý cơ sở dữ liệu thời gian thực, xác thực người dùng và thông báo đẩy, giúp ứng dụng trở nên linh hoạt và hiện đại. Tuy nhiên, cần lưu ý rằng mỗi công nghệ đều có những hạn chế nhất định, như khả năng mở rộng hay việc xử lý dữ liệu phức tạp khi ứng dụng tăng trưởng quy mô.

Mục tiêu chính của ứng dụng mạng xã hội là xây dựng một nền tảng trực quan, dễ sử dụng, cung cấp các tính năng kết nối, chia sẻ và tương tác hiệu quả. Đồng thời, ứng dụng còn nhằm đảm bảo hiệu suất ổn định, bảo mật thông tin người dùng và hỗ trợ quản lý nội dung một cách mạch lạc.

## 4.2. Kết quả đạt được

Trong quá trình học và triển khai dự án, chúng tôi đã tiếp thu và áp dụng nhiều kiến thức và kỹ năng quan trọng trong lĩnh vực phát triển phần mềm. Chúng tôi đã học cách lập kế hoạch, mô hình hóa dự án thông qua các sơ đồ như sơ đồ use case, sơ đồ sequence, và sơ đồ class để tổ chức các thành phần hệ thống. Điều này giúp chúng tôi hiểu rõ hơn về cách thức hoạt động và cấu trúc của ứng dụng mạng xã hội.

Ngoài ra, nhóm đã nắm được cách tích hợp các công nghệ hiện đại như Firebase để xử lý cơ sở dữ liệu và xác thực người dùng, cũng như sử dụng Node.js để xây dựng API backend hiệu quả. Việc thử nghiệm ứng dụng trên cả hai nền tảng Android và iOS cũng mang lại cái nhìn thực tế về quy trình phát triển và triển khai phần mềm đa nền tảng.

## 4.3. Những thứ chưa làm được

Dù đạt được nhiều tiến bộ, nhưng chúng tôi vẫn còn đối mặt với một số khó khăn và hạn chế trong quá trình thực hiện dự án. Báo cáo của chúng tôi chưa hoàn thiện ở một số khía cạnh, bao gồm:

* Chưa thể áp dụng đầy đủ kiến thức đã học vào việc triển khai các tính năng phức tạp của ứng dụng.
* Một số sơ đồ như sơ đồ sequence và sơ đồ activity chưa thể hiện đầy đủ các trường hợp sử dụng của hệ thống.
* Một vài tính năng trong ứng dụng chưa được hoàn thiện hoặc chưa đạt mức kỳ vọng.
* Khả năng diễn đạt thông tin và ý tưởng trong báo cáo có thể còn thiếu sót và cần cải thiện.

# **TÀI LIỆU THAM KHẢO**

[1] Ray, P. (2018). *Learning React Native: Building Native Mobile Apps with JavaScript*. O'Reilly Media, Inc., Sebastopol, CA, USA.

[2] Jones, B. (2011). The Evolution of Networking. HarperCollins, New York, USA

[3] <https://firebase.google.com/docs>

[4] <https://www.youtube.com/c/TraversyMedia/videos>

[5] <https://nodejs.org/en/docs>

[6] <https://reactnative.dev/docs>

[7] <https://developer.android.com/studio/intro>

[8] <https://www.youtube.com/c/Fireship/videos>