Shape, square

Description automatically generated

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

-----🙞🙜🕮🙞🙜-----

  
  
  
  
  
  
  
BÁO CÁO ĐỒ ÁN

**THIẾT KẾ KIẾN TRÚC PHẦN MỀM**

NGÀNH CÔNG NGHỆ THÔNG TIN

**Đề tài**

**Website Mạng xã hội**

Nhóm thực hiện: Nhóm 2

|  |  |
| --- | --- |
| Danh sách sinh viên thực hiện |  |
| 1. Phạm Thị Minh Huyền 2. Bùi Hoài Nam | **0192166** **S138665** |

Giảng viên: **ThS. Phạm Hữu Tùng**

Hà Nội, 04/2025

**MỤC LỤC**

[**DANH MỤC HÌNH ẢNH** 5](#_Toc199189797)

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 7](#_Toc199189798)

[1. Lý do chọn đề tài 7](#_Toc199189799)

[2. Mục tiêu của đồ án 7](#_Toc199189800)

[3. Phạm vi và giới hạn của đồ án 8](#_Toc199189801)

[4. Phương pháp thực hiện 8](#_Toc199189802)

[CHƯƠNG II: PHÂN TÍCH HỆ THỐNG 10](#_Toc199189803)

[1. Mô tả bài toán 10](#_Toc199189804)

[2. Yêu cầu chức năng 10](#_Toc199189805)

[2.1. Đăng bài viết và tương tác 10](#_Toc199189806)

[2.2. Quản lý tài khoản và hồ sơ cá nhân 11](#_Toc199189807)

[2.3. Kết bạn và quản lý mối quan hệ 11](#_Toc199189808)

[2.4. Tin nhắn cá nhân 11](#_Toc199189809)

[2.5. Tham gia và quản lý nhóm 11](#_Toc199189810)

[2.6. Trang fanpage 11](#_Toc199189811)

[2.7. Hệ thống thông báo 11](#_Toc199189812)

[3. Yêu cầu phi chức năng 12](#_Toc199189813)

[3.1. Tính khả dụng (Availability) 12](#_Toc199189814)

[3.2. Hiệu suất (Performance) 12](#_Toc199189815)

[3.3. Khả năng mở rộng (Scalability) 12](#_Toc199189816)

[3.4. Bảo mật (Security) 12](#_Toc199189817)

[3.5. Tính dễ sử dụng (Usability) 12](#_Toc199189818)

[3.6. Khả năng bảo trì (Maintainability) 12](#_Toc199189819)

[4. Sơ đồ Usecase diagram tổng quát 13](#_Toc199189820)

[5. Mô tả các Usecase chính 14](#_Toc199189821)

[5.1. Use case đăng nhập 14](#_Toc199189822)

[*5.2. Use case đăng xuất.* 15](#_Toc199189823)

[*5.3. Use case đăng ký* 15](#_Toc199189824)

[*5.4. Use case Quản lý người dùng* 16](#_Toc199189825)

[*5.5. Use case Quản lý bài đăng* 18](#_Toc199189826)

[*5.7. Use case Quản lý tin nhắn* 22](#_Toc199189827)

[*5.8. Use case Quản lý bạn bè* 24](#_Toc199189828)

[6. Biểu đồ hoạt động 27](#_Toc199189829)

[6.1. Biểu đồ đăng bài và tương tác 27](#_Toc199189830)

[6.2. Quản lý tài khoản và hồ sơ cá nhân 29](#_Toc199189831)

[6.3. Kết bạn và quản lý mối quan hệ 34](#_Toc199189832)

[6.4. Tin nhắn cá nhân 36](#_Toc199189833)

[6.5. Tham gia và quản lý nhóm 37](#_Toc199189834)

[6.8. Hệ thống thông báo 39](#_Toc199189835)

[CHƯƠNG III: THIẾT KẾ HỆ THỐNG 40](#_Toc199189836)

[1. Lựa chọn mô hình kiến trúc 40](#_Toc199189837)

[1.1. Kiến trúc tổng thể 40](#_Toc199189838)

[1.2. Chi tiết các tầng 40](#_Toc199189839)

[1.3. Luồng hoạt động tiêu biểu 41](#_Toc199189840)

[1.4. Ưu điểm kiến trúc 41](#_Toc199189841)

[2. Sơ đồ kiến trúc tổng thể 42](#_Toc199189842)

[2.1. Design Pattern 42](#_Toc199189843)

[3. Biểu đồ tuần tự 44](#_Toc199189844)

[3.1. Sequence Đăng nhập 44](#_Toc199189845)

[3.2. Sequence Đăng xuất 45](#_Toc199189846)

[3.3. Sequence Cập nhật thông tin tài khoản 46](#_Toc199189847)

[3.4. Sequence Sửa bài viết 47](#_Toc199189848)

[3.5. Sequence Sửa nhóm 48](#_Toc199189849)

[3.6. Sequence Thêm bài viết 49](#_Toc199189850)

[3.7. Sequence Tạo nhóm 50](#_Toc199189851)

[3.8. Sequence xóa thông tin bài viết 51](#_Toc199189852)

[*3.*9. Sequence Gửi lời mời kết bạn 51](#_Toc199189853)

[3.10. Sequence Chấp nhận lời mời kết bạn 52](#_Toc199189854)

[3.11. Sequence Hủy lời mời kết bạn 52](#_Toc199189855)

[3.12. Sequence Bình luận 52](#_Toc199189856)

[3.13. Sequence Xóa bình luận 53](#_Toc199189857)

[3.14. Sequence Bày tỏ cảm xúc 53](#_Toc199189858)

[3.15. Sequence Cập nhật bày tỏ cảm xúc bài viết 54](#_Toc199189859)

[3.16. Sequence Hủy bỏ cảm xúc bài viết 54](#_Toc199189860)

[3.17. Sequence Nhắn tin 55](#_Toc199189861)

[4. Class 55](#_Toc199189862)

[5. Thiết kế cơ sở dữ liệu (ERD, bảng, mối quan hệ) 56](#_Toc199189863)

[6. Thiết kế giao diện người dùng 61](#_Toc199189864)

[6.1. Giao diện đăng nhập 61](#_Toc199189865)

[6.2. Giao diện đăng ký tài khoản 62](#_Toc199189866)

[6.3. Giao diện trang chủ 63](#_Toc199189867)

[6.4. Giao diện quản lí tin nhắn 63](#_Toc199189868)

[6.5. Giao diện nhắn tin riêng 64](#_Toc199189869)

[6.6. Giao diện tạo cuộc hội thoại mới 65](#_Toc199189870)

[6.7. Giao diện tạo cuộc trò chuyện nhóm 66](#_Toc199189871)

[6.8. Giao diện tìm kiếm cuộc trò chuyện 66](#_Toc199189872)

[6.9. Giao diện Quản lý tài khoản cá nhân 67](#_Toc199189873)

[6.10. Giao diện tạo bài viết 68](#_Toc199189874)

[6.11. Giao diện lựa chọn chế độ hiển thị khi tạo bài viết 69](#_Toc199189875)

[6.12. Giao diện Quản lý bạn bè cá nhân 70](#_Toc199189876)

[6.13. Giao diện tìm kiếm bạn bè 70](#_Toc199189877)

[6.14. Giao diện xem trang cá nhân của bạn bè 71](#_Toc199189878)

[6.15. Giao diện xem chi tiết trang cá nhân bạn bè 71](#_Toc199189879)

[6.16. Giao diện kết bạn 72](#_Toc199189880)

[6.17. Giao diện sau khi nhấn gửi kết bạn 72](#_Toc199189881)

[6.18. Giao diện khi kết bạn thành công 73](#_Toc199189882)

[6.19. Giao diện xem danh sách bạn bè khác của bạn bè 73](#_Toc199189883)

[6.20. Giao diện thông báo 74](#_Toc199189884)

[6.21. Giao diện cài đặt tài khoản và đăng xuất 75](#_Toc199189885)

[6.22. Giao diện sửa thông tin tài khoản 75](#_Toc199189886)

[CHƯƠNG IV: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG 76](#_Toc199189887)

[1. Môi trường triển khai 76](#_Toc199189888)

[2. Cài đặt hệ thống 76](#_Toc199189889)

[2.1. Yêu cầu hệ thống 76](#_Toc199189890)

[2.2. Các bước cài đặt chương trình 76](#_Toc199189891)

[CHƯƠNG V: KẾT QUẢ VÀ ĐÁNH GIÁ 78](#_Toc199189892)

[1. Kết quả thử nghiệm hệ thống 78](#_Toc199189893)

[2. Đánh giá hiệu quả hệ thống 78](#_Toc199189894)

[CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 79](#_Toc199189895)

[1. Kết luận 79](#_Toc199189896)

[2. Hạn chế 79](#_Toc199189897)

[3. Hướng phát triển 79](#_Toc199189898)

[TÀI LIỆU THAM KHẢO 80](#_Toc199189899)

# **DANH MỤC HÌNH ẢNH**

[Hình 1: Usecase Tổng quát 13](#_Toc199189900)

[Hình 2: Use case Quản lý người dùng 16](#_Toc199189901)

[Hình 3: Usecase Quản lý bài đăng 18](#_Toc199189902)

[Hình 4: Usecase Quản lý tin nhắn 22](#_Toc199189903)

[Hình 5: Usecase Quản lý bạn bè 24](#_Toc199189904)

[Hình 6: Biểu đồ Đăng bài viết và tương tác 28](#_Toc199189905)

[Hình 7: Biểu đồ Quản lý tài khoản 30](#_Toc199189906)

[Hình 8: Biểu đồ Đăng xuất 32](#_Toc199189907)

[Hình 9: Biểu đồ đăng kí, đăng nhập 33](#_Toc199189908)

[Hình 10: Biểu đồ Kết bạn và quản lý mối quan hệ 35](#_Toc199189909)

[Hình 11: Biểu đồ Tin nhắn cá nhân 36](#_Toc199189910)

[Hình 12: Biểu đồ Tham gia và quản lý nhóm 38](#_Toc199189911)

[Hình 13: Biểu đồ Quản lý thông báo 39](#_Toc199189912)

[Hình 14: Sơ đồ kiến trúc tổng thể 42](#_Toc199189913)

[Hình 15: Sequence Đăng nhập 44](#_Toc199189914)

[Hình 16: Sequence Đăng xuất 45](#_Toc199189915)

[Hình 17: Sequence Cập nhật tài khoản 46](#_Toc199189916)

[Hình 18: Sequence Sửa bài viết 47](#_Toc199189917)

[Hình 19: Sequence Sửa nhóm 48](#_Toc199189918)

[Hình 20: Sequence Thêm bài viết 49](#_Toc199189919)

[Hình 21: Sequence Tạo nhóm 50](#_Toc199189920)

[Hình 22: Sequence Xóa bài viết 51](#_Toc199189921)

[Hình 23: Sequence Gửi lời mời kết bạn 51](#_Toc199189922)

[Hình 24: Sequence Chấp nhận lời mời kết bạn 52](#_Toc199189923)

[Hình 27: Sequence Hủy lời mời kết bạn 52](#_Toc199189924)

[Hình 25: Sequence Bình luận 52](#_Toc199189925)

[Hình 26: Sequence Xóa bình luận 53](#_Toc199189926)

[Hình 27: Sequence Bày tỏ cảm xúc 53](#_Toc199189927)

[Hình 28: Sequence Cập nhật bày tỏ cảm xúc bài viết 54](#_Toc199189928)

[Hình 29: Sequence Hủy bỏ cảm xúc bài viết 54](#_Toc199189929)

[Hình 30: Sequence Nhắn tin 55](#_Toc199189930)

[Hình 31: Class 55](#_Toc199189931)

[Hình 32: Sơ đồ ERD 56](#_Toc199189932)

[Hình 33: Giao diện đăng nhập 61](#_Toc199189933)

[Hình 34: Giao diện đăng ký tài khoản 62](#_Toc199189934)

[Hình 35: Giao diện trang chủ 63](#_Toc199189935)

[Hình 36: Giao diện quản lí tin nhắn 63](#_Toc199189936)

[Hình 37: Giao diện nhắn tin riêng 64](#_Toc199189937)

[Hình 38: Giao diện tạo cuộc hội thoại mới 65](#_Toc199189938)

[Hình 39: Giao diện tạo cuộc trò chuyện nhóm 66](#_Toc199189939)

[Hình 40: Giao diện tìm kiếm cuộc trò chuyện 66](#_Toc199189940)

[Hình 41: Giao diện Quản lý tài khoản cá nhân 67](#_Toc199189941)

[Hình 42: Giao diện tạo bài viết 68](#_Toc199189942)

[Hình 43: Giao diện lựa chọn chế độ hiển thị khi tạo bài viết 69](#_Toc199189943)

[Hình 44: Giao diện Quản lý bạn bè cá nhân 70](#_Toc199189944)

[Hình 45: Giao diện tìm kiếm bạn bè 70](#_Toc199189945)

[Hình 46: Giao diện xem trang cá nhân của bạn bè 71](#_Toc199189946)

[Hình 47: Giao diện xem chi tiết trang cá nhân bạn bè 71](#_Toc199189947)

[Hình 48: Giao diện kết bạn 72](#_Toc199189948)

[Hình 49: Giao diện sau khi nhấn gửi kết bạn 72](#_Toc199189949)

[Hình 50: Giao diện sau khi kết bạn thành công 73](#_Toc199189950)

[Hình 51: Giao diện xem danh sách bạn bè khác của bạn bè 73](#_Toc199189951)

[Hình 52: Giao diện thông báo 74](#_Toc199189952)

[Hình 53: Giao diện cài đặt tài khoản và đăng xuất 75](#_Toc199189953)

[Hình 54: Giao diện sửa thông tin tài khoản 75](#_Toc199189954)

# 

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

## Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, mạng xã hội đã trở thành công cụ thiết yếu phục vụ giao tiếp, chia sẻ và kết nối giữa con người với nhau. Từ đời sống cá nhân đến hoạt động doanh nghiệp, mạng xã hội đóng vai trò quan trọng trong truyền thông, xây dựng thương hiệu, trao đổi thông tin và tạo lập cộng đồng.

Tuy nhiên, các nền tảng mạng xã hội phổ biến hiện nay như Facebook, Instagram, hay Zalo là những hệ thống phức tạp với kiến trúc lớn, khó tiếp cận và tùy biến đối với sinh viên hoặc người mới học lập trình web. Do đó, việc xây dựng một mạng xã hội mini với các chức năng cơ bản sẽ giúp người học hiểu rõ hơn về kiến trúc web, cơ chế xử lý dữ liệu, mô hình MVC cũng như tương tác giữa người dùng và hệ thống.

Với mong muốn được tiếp cận gần hơn với thực tế phát triển phần mềm và nâng cao năng lực chuyên môn, nhóm chúng em quyết định thực hiện đồ án xây dựng một hệ thống mạng xã hội đơn giản phục vụ mục đích học tập, nghiên cứu và ứng dụng.

## Mục tiêu của đồ án

Mục tiêu chính của đồ án là xây dựng một hệ thống mạng xã hội mini mô phỏng các chức năng cơ bản của một nền tảng mạng xã hội hiện đại, từ đó giúp người học có thể tiếp cận, thực hành và hiểu rõ hơn về quy trình xây dựng một hệ thống web hoàn chỉnh.

Cụ thể, đồ án hướng tới các mục tiêu sau:

* Phát triển một ứng dụng mạng xã hội cơ bản với các chức năng: đăng ký, đăng nhập, tạo bài viết, like, bình luận, kết bạn và trò chuyện trực tiếp (chat 1:1).
* Ứng dụng Laravel cho backend và Vue.js cho frontend để xây dựng ứng dụng web theo mô hình SPA (Single Page Application), giúp rèn luyện kỹ năng lập trình hiện đại.
* Thiết kế cơ sở dữ liệu logic, chặt chẽ, bảo đảm khả năng mở rộng và quản lý dữ liệu người dùng, bài viết, mối quan hệ bạn bè, tin nhắn.
* Tích hợp tính năng chat trực tiếp thời gian thực, hỗ trợ giao tiếp cá nhân giữa các người dùng, từ đó hiểu sâu hơn về WebSocket hoặc các giải pháp realtime như Laravel Echo, Pusher hoặc socket.io.
* Nâng cao kỹ năng làm việc nhóm, quản lý tiến độ và quy trình phát triển phần mềm, bao gồm các bước từ phân tích, thiết kế, triển khai, kiểm thử đến đánh giá hệ thống.
* Tạo nền tảng cho việc mở rộng hoặc phát triển các chức năng nâng cao khác như thông báo, nhóm cộng đồng, tương tác đa phương tiện.

## Phạm vi và giới hạn của đồ án

Do giới hạn về thời gian thực hiện, tài nguyên và mục tiêu tập trung vào việc mô phỏng và học tập, đồ án sẽ chỉ triển khai các chức năng cơ bản của một mạng xã hội, cụ thể như sau:

**a. Phạm vi thực hiện**

* Cho phép người dùng đăng ký, đăng nhập và chỉnh sửa thông tin cá nhân.
* Tạo bài viết văn bản có thể kèm hình ảnh, hiển thị theo dòng thời gian trên trang chủ.
* Tương tác với bài viết thông qua chức năng Like và Comment.
* Tính năng kết bạn: gửi, nhận và hủy lời mời kết bạn.
* Xem danh sách bạn bè và truy cập trang cá nhân của bạn bè.
* Chức năng chat trực tiếp (chat 1:1) với bạn bè theo thời gian thực, sử dụng công nghệ WebSocket hoặc các thư viện hỗ trợ.
* Giao diện người dùng hiện đại, dễ sử dụng, hỗ trợ hiển thị trên trình duyệt máy tính.

**b. Giới hạn hệ thống**

* Không triển khai các tính năng nâng cao như chat nhóm, video call, hoặc thông báo đẩy (push notification).
* Không có chức năng quản trị viên, quản lý người dùng hay kiểm duyệt nội dung.
* Chưa tối ưu về hiệu năng hoặc bảo mật cao cấp như xác thực đa yếu tố (MFA), mã hóa tin nhắn đầu cuối.
* Hệ thống hoạt động chủ yếu trên môi trường cục bộ (localhost) sử dụng XAMPP, chưa triển khai lên server thực tế hoặc cloud.
* Không thực hiện kiểm thử bảo mật chuyên sâu, chỉ dừng ở mức thử nghiệm chức năng.

## Phương pháp thực hiện

**Backend**

* **Laravel** (PHP Framework): Xây dựng API, xử lý logic nghiệp vụ, xác thực, phân quyền.
* **Laravel Eloquent ORM:** Quản lý và truy vấn dữ liệu.
* **Composer:** Quản lý các thư viện PHP.

**Frontend**

* **Vue.js:** Xây dựng giao diện người dùng động, hiện đại.
* **Inertia.js:** Kết nối giữa Laravel và Vue.js, xây dựng SPA.
* **Bootstrap:** Thiết kế giao diện, responsive.
* **Axios:** Gửi HTTP request từ frontend đến backend.
* **npm:** Quản lý các thư viện JavaScript.

**Cơ sở dữ liệu**

* **MySQL:** Lưu trữ dữ liệu người dùng, bài viết, nhóm, tin nhắn, v.v.

**Môi trường phát triển & triển khai**

* **XAMPP:** Chạy Apache, MySQL, PHP trên máy tính cá nhân (localhost).
* **Node.js:** Chạy môi trường JavaScript cho frontend build.
* **Git:** Quản lý mã nguồn.

**Khác**

* **Inertia Form:** Hỗ trợ gửi form với file từ Vue đến Laravel.
* **Middleware Laravel:** Bảo vệ route, xác thực người dùng.
* **CSRF Protection:** Bảo vệ khỏi tấn công CSRF.
* **Session/Cookie:** Quản lý phiên đăng nhập.

# CHƯƠNG II: PHÂN TÍCH HỆ THỐNG

## 1. Mô tả bài toán

Trong thực tế, sự phát triển của mạng xã hội đã làm thay đổi hoàn toàn cách con người tương tác với nhau trong môi trường số. Người dùng ngày nay có nhu cầu chia sẻ thông tin, kết nối bạn bè, trò chuyện trực tuyến, theo dõi các hoạt động của người thân, bạn bè và các cộng đồng mà họ quan tâm. Các nền tảng như Facebook, Twitter hay Zalo đã trở thành những ví dụ điển hình cho xu thế đó.

Tuy nhiên, việc xây dựng và vận hành một mạng xã hội không hề đơn giản. Đằng sau giao diện người dùng trực quan là một hệ thống phức tạp gồm nhiều thành phần kỹ thuật cần phối hợp chặt chẽ: quản lý người dùng, dữ liệu bài viết, tương tác xã hội, truyền thông tin thời gian thực, và đảm bảo bảo mật cho toàn bộ quá trình xử lý.

Do đó, bài toán đặt ra là **làm thế nào để xây dựng một hệ thống mạng xã hội mô phỏng các chức năng cơ bản nhất của một nền tảng thực tế**, đảm bảo:

* Người dùng có thể tạo tài khoản, chỉnh sửa thông tin cá nhân.
* Thiết lập và quản lý các mối quan hệ bạn bè một cách linh hoạt.
* Cho phép chia sẻ nội dung và tương tác với nội dung người khác tạo ra.
* Cung cấp khả năng giao tiếp qua tin nhắn cá nhân.
* Tổ chức dữ liệu và điều phối xử lý theo cách hiệu quả, bảo mật, và dễ mở rộng.

Bài toán không chỉ nằm ở việc hiện thực giao diện hay lập trình tính năng, mà còn là thách thức về tư duy tổ chức hệ thống, thiết kế dữ liệu, quản lý phiên người dùng, phân chia vai trò frontend – backend, và đảm bảo hệ thống hoạt động ổn định với nhiều người dùng cùng lúc.

## 2. Yêu cầu chức năng

Hệ thống cần đáp ứng các chức năng chính sau để đảm bảo người dùng có thể tương tác xã hội và chia sẻ nội dung một cách hiệu quả:

### 2.1. Đăng bài viết và tương tác

* Người dùng muốn chia sẻ nội dung cá nhân như văn bản, hình ảnh, video để giao lưu và kết nối với bạn bè. Sau khi đăng bài, người dùng có thể nhận được các tương tác từ bạn bè như bình luận, thả cảm xúc (like, love, haha, sad...).
* Bài viết có thể được điều chỉnh quyền riêng tư (công khai, bạn bè, chỉ mình tôi).
* Hệ thống cần hiển thị bài viết theo thứ tự thời gian và mức độ tương tác trên **news feed**.

### 2.2. Quản lý tài khoản và hồ sơ cá nhân

* Người dùng có thể đăng ký tài khoản mới, cập nhật thông tin cá nhân như tên, giới tính, ngày sinh, ảnh đại diện, ảnh bìa, giới thiệu bản thân.

### 2.3. Kết bạn và quản lý mối quan hệ

* Người dùng có thể gửi lời mời kết bạn, chấp nhận hoặc từ chối yêu cầu kết bạn từ người khác.
* Hệ thống cần hiển thị danh sách bạn bè, gợi ý bạn bè dựa trên tương tác và sở thích chung.
* Có thể hủy kết bạn hoặc chặn người dùng khi cần thiết.

### 2.4. Tin nhắn cá nhân

* Người dùng có thể gửi/nhận tin nhắn riêng tư, trò chuyện theo thời gian thực.
* Hệ thống cần hỗ trợ tính năng thông báo tin nhắn mới, cho phép gửi kèm ảnh, emoji, file.

### 2.5. Tham gia và quản lý nhóm

* Người dùng có thể tạo nhóm thảo luận, mời bạn bè tham gia nhóm, đăng bài trong nhóm.
* Quản trị viên nhóm có thể duyệt bài viết, thêm/xóa thành viên, thay đổi thông tin nhóm.

### 2.6. Trang fanpage

* Người dùng có thể tạo trang để đại diện cho doanh nghiệp, thương hiệu hoặc sở thích cá nhân.
* Có thể đăng bài trên fanpage, tương tác với người theo dõi, thống kê lượng tương tác.

### 2.7. Hệ thống thông báo

* Hệ thống cần thông báo khi có người thích, bình luận bài viết, gửi lời mời kết bạn, tin nhắn mới...
* Có thể tùy chỉnh thông báo theo từng loại.

## 3. Yêu cầu phi chức năng

Hệ thống cần đáp ứng các yêu cầu phi chức năng sau nhằm đảm bảo hiệu quả vận hành, bảo mật và trải nghiệm người dùng:

### 3.1. Tính khả dụng (Availability)

* Hệ thống phải sẵn sàng sử dụng bất kỳ lúc nào khi máy chủ hoạt động.
* Hạn chế thời gian downtime ở mức tối thiểu.

### 3.2. Hiệu suất (Performance)

* Thời gian phản hồi cho các thao tác thường xuyên (như đăng nhập, đăng bài, gửi tin nhắn) không vượt quá 1 giây trong điều kiện bình thường.
* Dữ liệu được tải theo phân trang để tránh tình trạng quá tải.

### 3.3. Khả năng mở rộng (Scalability)

* Cấu trúc dữ liệu và kiến trúc hệ thống cho phép mở rộng tính năng (như nhóm, chia sẻ bài viết, thông báo nâng cao) mà không ảnh hưởng đến các chức năng hiện tại.

### 3.4. Bảo mật (Security)

* Mật khẩu người dùng được mã hóa khi lưu trong cơ sở dữ liệu.
* Người dùng chỉ có thể xem hoặc tương tác với bài viết và tin nhắn của bạn bè.
* Tránh các lỗ hổng phổ biến như SQL Injection, XSS, CSRF.

### 3.5. Tính dễ sử dụng (Usability)

* Giao diện người dùng đơn giản, trực quan, dễ thao tác kể cả với người dùng phổ thông.
* Các chức năng chính được đặt ở vị trí dễ truy cập trên giao diện.

### 3.6. Khả năng bảo trì (Maintainability)

* Mã nguồn được tổ chức theo mô hình MVC, giúp dễ dàng quản lý, nâng cấp và sửa lỗi.
* Tách biệt frontend (Vue.js) và backend (Laravel) thông qua API RESTful rõ ràng.

## 4. Sơ đồ Usecase diagram tổng quát

A blue rectangular object with black text

AI-generated content may be incorrect.

Hình 1: Usecase Tổng quát

## 5. Mô tả các Usecase chính

### 5.1. Use case đăng nhập

|  |  |
| --- | --- |
| Use case | Đăng nhập. |
| Use case ID | UC01 |
| Actor | Người dùng. |
| Description | Là người quản lý, tôi muốn đăng nhập vào hệ thống để sử dụng các chức năng. |
| Trigger | Người dùng thực hiện lệnh đăng nhập (bấm nút đăng nhập). |
| Pre-Condition: | * Người dùng đã nhập thông tin đăng nhập chính xác và đầy đủ trên hệ thống. * Tài khoản đăng nhập đã tồn tại trong hệ thống. |
| Basic flow | 1. Tại trang chủ, người dùng truy cập vào trang quản trị hệ thống. 2. Hệ thống hiển thị form đăng nhập chứa username và password. 3. Người dùng nhập:  * Username * Password  1. Người dùng nhấn vào nút “Đăng nhập” 2. Hệ thống kiểm tra thông tin đăng nhập:  * Kiểm tra username và password có khớp với tài khoản trong database.  1. Hệ thống hiển thị thông báo đăng nhập thành công và hiển thị trang quản trị hệ thống. |
| Result | Người dùng đăng nhập được vào hệ thống và sử dụng được các chức năng. |
| Exceptions | * Nếu người dùng không nhập đủ các trường, hiển thị thông báo yêu cầu nhập dưới mỗi trường. * Nếu username hoặc password không nhập đúng, yêu cầu hiển thị lỗi ngay dưới trường đó. |
| Priority | Must have. |

### *5.2. Use case đăng xuất.*

|  |  |
| --- | --- |
| Use case | Đăng xuất. |
| Use case ID | UC02 |
| Actor | Người dùng. |
| Description | Là người dùng, tôi muốn đăng xuất khỏi hệ thống khi kết thúc phiên làm việc. |
| Trigger | Người dùng thực hiện lệnh đăng xuất (bấm nút đăng xuất). |
| Pre-Condition: | * Người dùng đã nhập thông tin đăng nhập chính xác và đầy đủ trên hệ thống. * Tài khoản đăng nhập đã tồn tại trong hệ thống. |
| Basic flow | 1. Tại trang chủ, người dùng truy cập vào trang quản trị hệ thống. 2. Người dùng nhấn vào nút “Đăng xuất” 3. Hệ thống hiển thị thông báo đăng xuất thành công và hiển thị trang quản trị hệ thống. |
| Result | Người dùng đăng xuất thành công tài khoản trên hệ thống. |
| Exceptions | * Nếu người dùng không nhập đủ các trường, hiển thị thông báo yêu cầu nhập dưới mỗi trường. * Nếu username hoặc password không nhập đúng, yêu cầu hiển thị lỗi ngay dưới trường đó. |
| Priority | Must have. |

### *5.3. Use case đăng ký*

|  |  |
| --- | --- |
| Use case | Đăng ký tài khoản. |
| Use case ID | UC03 |
| Actor | Người dùng |
| Description | Người dùng tạo tài khoản mới bằng cách cung cấp Fullname, Username và Password… |
| Trigger | Người dùng truy cập trang đăng ký. |
| Pre-Condition: | * Trang đăng ký phải hoạt động. |
| Basic flow | 1. Người dùng truy cập trang đăng ký. 2. Nhập Fullname, Username và Password, … 3. Hệ thống kiểm tra tính hợp lệ của dữ liệu. 4. Lưu thông tin tài khoản và xác nhận đăng ký thành công. |
| Result | Tài khoản mới được tạo. |
| Priority | Must have. |

### *5.4. Use case Quản lý người dùng*

A blue rectangular object with black text

AI-generated content may be incorrect.

Hình 2: Use case Quản lý người dùng

|  |  |
| --- | --- |
| Use case | Quản lý người dùng |
| Use case ID | UC04 |
| Actor | Người dùng. |
| Description | Là người dùng, tôi muốn thêm tài khoản mới, sửa và xóa tài khoản người dùng để kiểm soát truy cập hệ thống. |
| Trigger | Người dùng truy cập vào chức năng “Quản lý người dùng”. |
| Pre-Condition: | * Người dùng đã đăng nhập và có quyền quản lý người dùng. |
| Basic flow | 1. Người dùng đăng nhập vào hệ thống 2. Người dùng chọn chức năng “Quản lý người dùng”. 3. Hệ thống xử lý và hiển thị danh sách các chức năng. 4. Người dùng có thể thực hiện chức năng “thêm tài khoản mới/ cập nhật/ xóa người dùng”. |
| Result | Người dùng thực hiện quản lý tài khoản thành công trên hệ thống. |
| Priority | Must have. |

#### 5.4.1. Use case Sửa thông tin tài khoản.

|  |  |
| --- | --- |
| Use case | Sửa thông tin tài khoản. |
| Use case ID | UC04.1 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn cập nhật thông tin tài khoản cá nhân. |
| Trigger | |  |  | | --- | --- | |  | Người dùng nhấn vào “Chỉnh sửa tài khoản”. | |
| Pre-Condition: | * Người dùng đã đăng nhập. * Có quyền chỉnh sửa thông tin cá nhân. |
| Basic flow | 1. Người dùng truy cập vào hệ thống. 2. Truy cập giao diện tài khoản. 3. Nhấn “Chỉnh sửa”. 4. Cập nhật thông tin cần thay đổi (email, mật khẩu, ảnh đại diện…). 5. Nhấn “Lưu”. 6. Hệ thống cập nhật thông tin tài khoản. |
| Result | Tài khoản được cập nhật thành công. |
| Exceptions | * Dữ liệu không hợp lệ. * Người dùng không có quyền chỉnh sửa. |
| Priority | Must have. |

#### 5.4.2. Use case Xóa thông tin tài khoản.

|  |  |
| --- | --- |
| Use case | Xóa thông tin tài khoản. |
| Use case ID | UC04.2 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn xóa tài khoản người dùng không còn hoạt động. |
| Trigger | Người dùng thực hiện lệnh xóa tài khoản (bấm nút xóa). |
| Pre-Condition: | * Quản trị viên đã đăng nhập. * Người dùng tồn tại trong hệ thống. |
| Basic flow |  |
| Result | Tài khoản bị xóa khỏi hệ thống. |
| Exceptions | * Tài khoản đang liên kết dữ liệu khác (bài viết, bình luận…) chưa xử lý. * Lỗi hệ thống khi xóa. |
| Priority | Must have. |

### *5.5. Use case Quản lý bài đăng*

A blue rectangular object with white text

AI-generated content may be incorrect.

Hình 3: Usecase Quản lý bài đăng

|  |  |
| --- | --- |
| Use case | Quản lý bài đăng. |
| Use case ID | UC05 |
| Actor | Người dùng. |
| Description | Là người dùng, tôi muốn quản lý bài đăng (thêm/ cập nhật/ xóa/bình luận/ tương tác (like/share) bài đăng) trên hệ thống. |
| Trigger | Người dùng thực hiện lệnh quản lý bài đăng (bấm nút quản lý bài đăng). |
| Pre-Condition: | * Người dùng đã đăng nhập vào hệ thống. |
| Basic flow | 1. Người dùng đăng nhập vào hệ thống 2. Người dùng chọn chức năng “Quản lý bài đăng”. 3. Hệ thống xử lý và hiển thị danh sách các chức năng. 4. Người dùng có thể thực hiện chức năng “thêm/ cập nhật/ xóa/bình luận/ tương tác (like/share) bài đăng”. |
| Result | Người dùng thực hiện quản lý câu hỏi quiz thành công trên hệ thống. |
| Priority | Must have. |

#### 5.5.1. Use Case Tạo bài viết

|  |  |
| --- | --- |
| **Use Case** | Tạo bài viết |
| **Use Case ID** | UC05.1 |
| **Actor** | Người dùng |
| **Description** | Là người dùng, tôi muốn thêm bài viết mới vào hệ thống |
| **Trigger** | Người dùng thực hiện lệnh tạo bài viết (bấm nút thêm). |
| **Pre-Condition** | - Tài khoản người dùng đã đăng nhập thành công. |
| **Basic Flow** | * 1. Người dùng truy cập vào hệ thống.   2. Người dùng vào giao diện tạo bài viết.   3. Nhập tiêu đề và nội dung bài viết.   4. Tải lên hình ảnh hoặc tệp đính kèm nếu cần.   5. Nhấn nút “Đăng bài”.   6. Hệ thống xác nhận và lưu bài viết.   7. Hệ thống hiển thị bài viết trên trang tin. |
| **Result** | Bài viết được tạo và có thể được sử dụng trong hệ thống. |
| **Exceptions** | * Nếu nội dung trống, hệ thống hiển thị thông báo lỗi. * Nếu không có kết nối mạng, hệ thống hiển thị thông báo và không đăng bài. |
| **Priority** | Must have. |

#### 5.5.2. Use case Sửa bài viết.

|  |  |
| --- | --- |
| Use case | Sửa bài viết. |
| Use case ID | UC05.2 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn chỉnh sửa nội dung bài viết đã đăng. |
| Trigger | |  |  | | --- | --- | |  | Người dùng nhấn vào nút “Chỉnh sửa” tại bài viết. | |
| Pre-Condition: | * Người dùng đã đăng nhập. * Bài viết thuộc sở hữu của người dùng. |
| Basic flow | 1. Người dùng truy cập vào hệ thống. 2. Người dùng chọn bài viết cần chỉnh sửa. 3. Hệ thống hiển thị giao diện sửa bài viết. 4. Người dùng chỉnh sửa nội dung hoặc tệp đính kèm. 5. Nhấn “Lưu”. 6. Hệ thống cập nhật và hiển thị nội dung mới. |
| Result | Bài viết được cập nhật thành công. |
| Exceptions | * Người dùng không có quyền chỉnh sửa bài viết. * Nội dung trống hoặc không hợp lệ. |
| Priority | Must have. |

#### 5.5.3. Use case Xóa bài viết.

|  |  |
| --- | --- |
| Use case | Xóa bài viết. |
| Use case ID | UC05.3 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn xóa bài viết không còn phù hợp. |
| Trigger | Người dùng thực hiện lệnh xóa bài viết (bấm nút xóa). |
| Pre-Condition: | * Người dùng đã đăng nhập. * Là chủ sở hữu bài viết. |
| Basic flow | 1. Người dùng truy cập vào hệ thống. 2. Người dùng chọn bài viết muốn xóa. 3. Nhấn nút “Xóa”. 4. Hệ thống hiển thị hộp thoại xác nhận. 5. Người dùng xác nhận xóa. 6. Hệ thống xóa bài viết khỏi hệ thống. |
| Result | Bài viết bị xóa khỏi hệ thống. |
| Exceptions | * Người dùng không có quyền xóa bài viết. * Lỗi hệ thống khi xóa. |
| Priority | Must have. |

#### 5.5.4. Use case Bình luận.

|  |  |
| --- | --- |
| Use case | Cập nhật thông tin câu hỏi. |
| Use case ID | UC05.4 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn bình luận vào bài viết để tương tác và chia sẻ ý kiến. |
| Trigger | Người dùng nhập bình luận và nhấn nút “Gửi”. |
| Pre-Condition: | * Người dùng đã đăng nhập. |
| Basic flow | 1. Người dùng truy cập vào hệ thống. 2. Người dùng đọc bài viết. 3. Nhập nội dung bình luận. 4. Nhấn “Gửi”. 5. Hệ thống lưu và hiển thị bình luận. |
| Result | Bình luận được hiển thị dưới bài viết. |
| Priority | Must have. |

#### 5.5.5. Use case Tương tác (Like, Share).

|  |  |
| --- | --- |
| Use case | Tương tác (Like, Share) |
| Use case ID | UC05.5 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn tương tác với bài viết bằng cách Like hoặc Share. |
| Trigger | Người dùng nhấn nút “Like” hoặc “Chia sẻ”. |
| Pre-Condition: | * Người dùng đã đăng nhập. |
| Basic flow | 1. Người dùng đọc bài viết. 2. Nhấn “Like” hoặc “Chia sẻ”. 3. Hệ thống ghi nhận tương tác. |
| Result | Bài viết được tương tác (hiện số lượt Like/Share). |
| Priority | Should have |

#### 5.5.6. Use case Xem bài viết.

|  |  |
| --- | --- |
| Use case | Xem bài viết |
| Use case ID | UC05.6 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn xem các bài viết hiện có trên hệ thống. |
| Trigger | Người dùng truy cập trang chính hoặc danh sách bài viết. |
| Pre-Condition: |  |
| Basic flow | 1. Người dùng truy cập trang xem bài viết. 2. Hệ thống hiển thị danh sách các bài viết. 3. Người dùng chọn bài viết cụ thể và nhấn “Xem thêm” để đọc chi tiết. |
| Result | Người dùng đọc được bài viết. |
| Priority | Must have |

### *5.7. Use case Quản lý tin nhắn*

A blue rectangular object with text

AI-generated content may be incorrect.

Hình 4: Usecase Quản lý tin nhắn

|  |  |
| --- | --- |
| Use case | Quản lý tin nhắn |
| Use case ID | UC07 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn gửi và nhận tin nhắn với bạn bè để trò chuyện. |
| Trigger | Người dùng truy cập hộp thoại chat và gửi tin nhắn. |
| Pre-Condition: | - Người dùng đã đăng nhập.  - Người dùng và Người nhận có quyền nhắn tin cho nhau. |
| Basic flow | 1. Người dùng truy cập trang nhắn tin.  2. Chọn người nhận trong danh sách bạn bè hoặc tìm kiếm.  3. Nhập nội dung tin nhắn và nhấn “Gửi”.  4. Hệ thống lưu và hiển thị tin nhắn cho cả hai bên. |
| Result | Tin nhắn được gửi thành công và hiển thị trong hội thoại. |
| Priority | Must have |

#### 5.7.1 Usecase Xem tin nhắn

|  |  |
| --- | --- |
| Use case | Xem tin nhắn |
| Use case ID | UC07.1 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn xem lại các tin nhắn đã gửi và nhận để theo dõi cuộc trò chuyện. |
| Trigger | Người dùng truy cập mục tin nhắn hoặc mở hội thoại. |
| Pre-Condition: | - Người dùng đã đăng nhập. |
| Basic flow | 1. Người dùng vào mục “Tin nhắn”. 2. Chọn hội thoại muốn xem. 3. Hệ thống hiển thị lịch sử trò chuyện. |
| Result | Người dùng xem được nội dung các tin nhắn trước đó. |
| Priority | Must have |

#### 5.7.2. Usecase Nhắn tin riêng

|  |  |
| --- | --- |
| Use case | Nhắn tin riêng |
| Use case ID | UC07.2 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn gửi tin nhắn riêng cho một người dùng khác để trò chuyện cá nhân. |
| Trigger | Người dùng chọn bạn bè và gửi tin nhắn. |
| Pre-Condition: | - Người dùng đã đăng nhập. - Hai Người đã kết bạn hoặc được phép nhắn tin. |
| Basic flow | 1. Người dùng chọn bạn bè từ danh sách. 2. Nhập nội dung và nhấn “Gửi”. 3. Hệ thống lưu và gửi tin nhắn đến người nhận. |
| Result | Tin nhắn được gửi thành công đến người nhận và hiển thị trong hội thoại. |
| Priority | Must have |

#### 5.7.3. Usecase nhắn tin nhóm

|  |  |
| --- | --- |
| Use case | Nhắn tin nhóm |
| Use case ID | UC07.3 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn gửi tin nhắn trong nhóm để trò chuyện với nhiều người cùng lúc. |
| Trigger | Người dùng tạo nhóm hoặc truy cập nhóm đã có và gửi tin nhắn. |
| Pre-Condition: | - Người dùng đã đăng nhập. - Người dùng là thành viên của nhóm. |
| Basic flow | 1. Người dùng truy cập nhóm. 2. Nhập nội dung và nhấn “Gửi”. 3. Hệ thống gửi tin nhắn đến tất cả thành viên trong nhóm. |
| Result | Tin nhắn hiển thị trong hội thoại nhóm cho tất cả thành viên. |
| Priority | Should have |

### *5.8. Use case Quản lý bạn bè*

A blue rectangular object with text and black text

AI-generated content may be incorrect.

Hình 5: Usecase Quản lý bạn bè

|  |  |
| --- | --- |
| Use case | Quản lý bạn bè |
| Use case ID | UC08 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn gửi lời mời, chấp nhận lời mời, hoặc hủy kết bạn để kiểm soát danh sách bạn bè của mình. |
| Trigger | Người dùng thực hiện một trong ba thao tác: gửi lời mời, chấp nhận lời mời, hoặc hủy kết bạn. |
| Pre-Condition: | - Người dùng đã đăng nhập. |
| Basic flow | 1. Người dùng truy cập trang cá nhân hoặc danh sách bạn bè. 2. Chọn một trong các hành động:   a. Nhấn “Kết bạn” để gửi lời mời.   b. Nhấn “Chấp nhận” lời mời kết bạn.   c. Nhấn “Hủy kết bạn” nếu đang là bạn bè. 3. Hệ thống xử lý hành động tương ứng và cập nhật quan hệ bạn bè. |
| Result | Mối quan hệ bạn bè được thay đổi: đã gửi lời mời, đã trở thành bạn bè, hoặc đã hủy kết bạn. |
| Priority | Must have |

#### 5.8.1. Usecase Gửi lời mời kết bạn

|  |  |
| --- | --- |
| Use case | Gửi lời mời kết bạn |
| Use case ID | UC08.1 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn gửi lời mời kết bạn để kết nối và nhắn tin với người khác. |
| Trigger | Người dùng nhấn nút “Kết bạn” trên hồ sơ người dùng khác. |
| Pre-Condition: | - Người dùng đã đăng nhập. |
| Basic flow | 1. Người dùng tìm kiếm hoặc truy cập hồ sơ người khác.  2. Nhấn nút “Kết bạn”.  3. Hệ thống ghi nhận và gửi lời mời. |
| Result | Lời mời kết bạn được gửi đi, chờ người kia chấp nhận. |
| Priority | Must have |

#### 5.8.2. Usecase Chấp nhận lời mời kết bạn

|  |  |
| --- | --- |
| Use case | Chấp nhận lời mời kết bạn |
| Use case ID | UC08.2 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn chấp nhận lời mời kết bạn để kết nối với người gửi lời mời. |
| Trigger | Người dùng nhận được thông báo lời mời và nhấn “Chấp nhận”. |
| Pre-Condition: | - Người dùng đã đăng nhập.  - Có lời mời kết bạn đang chờ xử lý. |
| Basic flow | 1. Người dùng mở thông báo lời mời kết bạn.  2. Nhấn “Chấp nhận”.  3. Hệ thống cập nhật trạng thái quan hệ bạn bè. |
| Result | Hai người trở thành bạn bè, có thể nhắn tin hoặc tương tác. |
| Priority | Must have |

#### 5.8.3. Usecase Hủy kết bạn

|  |  |
| --- | --- |
| Use case | Hủy kết bạn |
| Use case ID | UC08.3 |
| Actor | Người dùng |
| Description | Là người dùng, tôi muốn hủy kết bạn để ngừng kết nối với một người dùng khác. |
| Trigger | Người dùng truy cập hồ sơ bạn bè và nhấn “Hủy kết bạn”. |
| Pre-Condition: | - Người dùng đã đăng nhập.  - Đang là bạn bè với người kia. |
| Basic flow | 1. Người dùng vào hồ sơ bạn bè.  2. Nhấn “Hủy kết bạn”.  3. Hệ thống cập nhật trạng thái và xóa quan hệ bạn bè. |
| Result | Người dùng không còn là bạn bè với người kia, mất quyền tương tác như bạn bè. |
| Priority | Should have |

## 6. Biểu đồ hoạt động

### 6.1. Biểu đồ đăng bài và tương tác

**Mô tả luồng xử lý:**

1. **Người dùng đăng nhập** vào hệ thống để xác thực quyền sử dụng.
2. Sau khi đăng nhập thành công, người dùng thực hiện thao tác **tạo bài viết mới**.
3. Hệ thống yêu cầu người dùng **nhập nội dung bài viết**, có thể bao gồm văn bản hoặc hình ảnh.
4. Người dùng chọn **quyền riêng tư** cho bài viết (ví dụ: công khai, bạn bè, chỉ mình tôi).
5. Khi người dùng **nhấn đăng bài viết**, hệ thống sẽ:
   * Lưu nội dung vào cơ sở dữ liệu.
   * **Hiển thị bài viết lên News Feed** của bạn bè hoặc người xem phù hợp với quyền riêng tư đã chọn.
6. **Bạn bè/người xem** truy cập vào News Feed sẽ thấy bài viết mới.
7. Người xem có thể **xem bài viết**, đồng thời thực hiện các hành động **tương tác như thích (Like) hoặc chia sẻ (Share)**.
8. Hệ thống ghi nhận các hành vi tương tác và **hiển thị số lượng tương tác (like/share)** bên dưới bài viết.
9. Quá trình kết thúc.

**Các vai trò tham gia:**

* **Người dùng**: Tác nhân chính khởi tạo bài viết.
* **Hệ thống**: Xử lý đăng bài, lưu trữ, phân phối nội dung.
* **Bạn bè/Người xem**: Tác nhân thụ động, tương tác với bài viết sau khi được hiển thị.

A screenshot of a diagram

AI-generated content may be incorrect.

Hình 6: Biểu đồ Đăng bài viết và tương tác

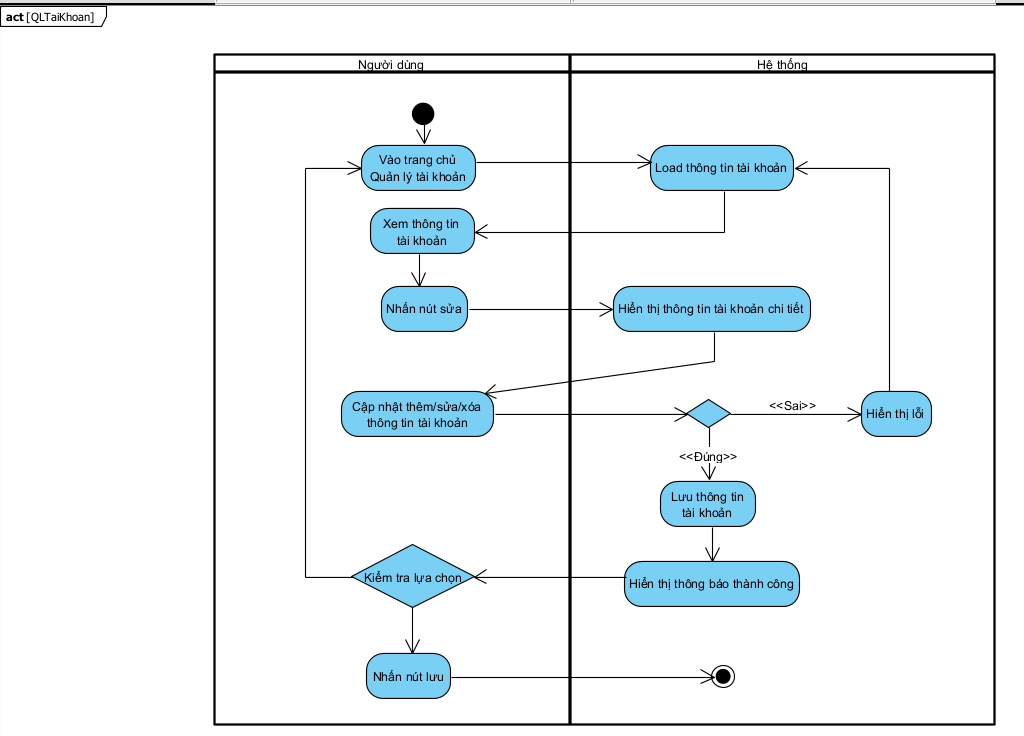
### 6.2. Quản lý tài khoản và hồ sơ cá nhân

**Mô tả luồng xử lý:**

1. **Người dùng** truy cập vào trang **Quản lý tài khoản** từ giao diện trang chủ.
2. Hệ thống thực hiện **load thông tin tài khoản** từ cơ sở dữ liệu.
3. Sau khi dữ liệu được tải, hệ thống **hiển thị thông tin tài khoản chi tiết** lên giao diện cho người dùng xem.
4. Người dùng bấm vào **nút chỉnh sửa** để tiến hành cập nhật lại các thông tin cá nhân như tên, ảnh đại diện, địa chỉ, v.v.
5. Người dùng thực hiện các hành động cập nhật như: **sửa, thêm hoặc xóa** thông tin.
6. Sau khi hoàn tất cập nhật, người dùng **nhấn nút lưu**.
7. Trước khi lưu, hệ thống thực hiện **kiểm tra tính hợp lệ** của dữ liệu nhập vào:
   * Nếu dữ liệu **không hợp lệ (sai)** → Hệ thống sẽ **hiển thị thông báo lỗi** và yêu cầu người dùng sửa lại.
   * Nếu dữ liệu **hợp lệ (đúng)** → Hệ thống sẽ **lưu thông tin tài khoản mới** vào cơ sở dữ liệu.
8. Sau khi lưu thành công, hệ thống **hiển thị thông báo xác nhận** cho người dùng biết đã cập nhật thành công.
9. Quá trình kết thúc.

**Các vai trò tham gia:**

* **Người dùng**: Thực hiện thao tác cập nhật thông tin tài khoản cá nhân.
* **Hệ thống**: Xử lý việc truy xuất, kiểm tra và lưu trữ thông tin người dùng.



Hình 7: Biểu đồ Quản lý tài khoản

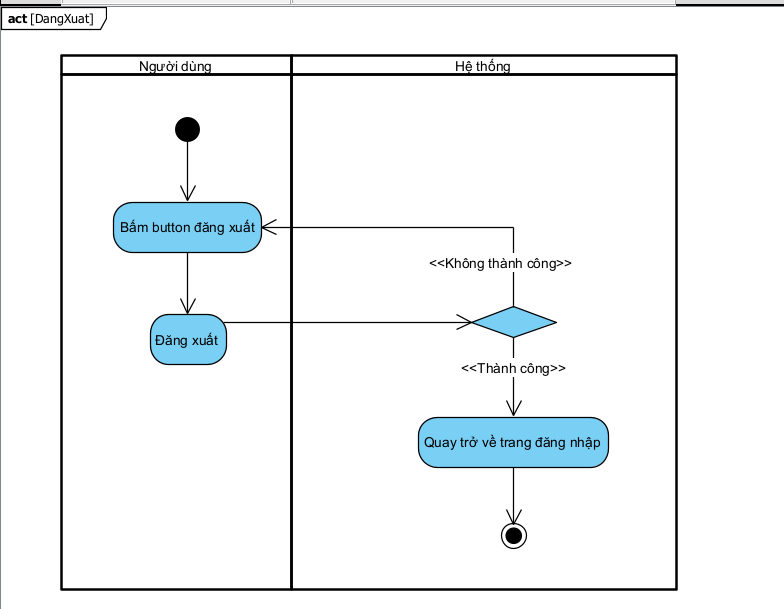
6.3. Đăng xuất

**Mô tả luồng xử lý:**

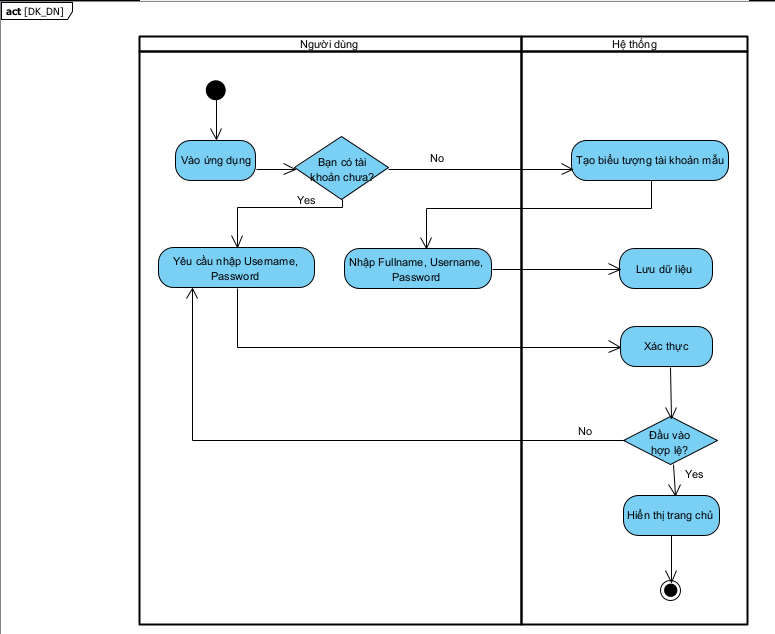
1. Người dùng đang trong trạng thái đã đăng nhập và muốn thoát khỏi hệ thống.
2. Người dùng **nhấn vào nút “Đăng xuất”** trên giao diện.
3. Hệ thống tiếp nhận yêu cầu và thực hiện thao tác **xóa phiên làm việc (session)** hoặc **token đăng nhập** của người dùng.
4. Hệ thống kiểm tra kết quả xử lý:
   * Nếu việc đăng xuất **không thành công** (ví dụ: lỗi máy chủ, session không tồn tại), quá trình dừng lại và có thể hiển thị thông báo lỗi cho người dùng (không thể đăng xuất).
   * Nếu việc đăng xuất **thành công**, hệ thống sẽ **chuyển hướng người dùng về trang đăng nhập**.
5. Quá trình đăng xuất kết thúc.

**Các vai trò tham gia:**

* **Người dùng**: Khởi tạo hành động đăng xuất.
* **Hệ thống**: Xử lý trạng thái đăng xuất và điều hướng phiên làm việc.



Hình 8: Biểu đồ Đăng xuất



Hình 9: Biểu đồ đăng kí, đăng nhập

### 6.3. Kết bạn và quản lý mối quan hệ

**Mô tả luồng xử lý:**

1. **Người dùng (Gửi)** đăng nhập vào hệ thống và truy cập vào mục quản lý bạn bè.
2. Hệ thống thực hiện **lấy danh sách gợi ý bạn bè** dựa trên một số tiêu chí như: bạn chung, tương tác gần đây, ...
3. Danh sách gợi ý được **hiển thị cho người dùng**.
4. Người dùng chọn một người trong danh sách và **gửi lời mời kết bạn**.
5. Hệ thống ghi nhận lời mời và **gửi thông báo** đến người nhận.
6. **Người nhận được lời mời** sẽ:
   * Nhận thông báo từ hệ thống.
   * Truy cập để **xem nội dung lời mời kết bạn**.
   * Có thể chọn **chấp nhận hoặc từ chối** lời mời.
7. Sau khi người nhận phản hồi:
   * Hệ thống **cập nhật trạng thái mối quan hệ** giữa hai người dùng.
   * Người gửi có thể **cập nhật danh sách bạn bè** để phản ánh kết quả (được chấp nhận hoặc bị từ chối).
8. Người dùng có thể **truy cập lại danh sách bạn bè**, chọn một người cụ thể và thực hiện hành động:
   * **Hủy kết bạn**.
   * Hoặc **chặn người dùng** nếu cần thiết.
9. Hệ thống sẽ xử lý và **hiển thị lại danh sách bạn bè đã cập nhật** sau thao tác này.

**Các vai trò tham gia:**

* **Người dùng (Gửi)**: Người khởi tạo lời mời kết bạn.
* **Hệ thống**: Xử lý gợi ý, lưu lời mời, cập nhật trạng thái.
* **Người nhận**: Người phản hồi lời mời kết bạn.

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 10: Biểu đồ Kết bạn và quản lý mối quan hệ

### 6.4. Tin nhắn cá nhân

* Người dùng có thể gửi/nhận tin nhắn riêng tư, trò chuyện theo thời gian thực.
* Hệ thống cần hỗ trợ tính năng thông báo tin nhắn mớiA diagram of a diagram

  AI-generated content may be incorrect.

Hình 11: Biểu đồ Tin nhắn cá nhân

### 6.5. Tham gia và quản lý nhóm

**Mô tả luồng xử lý:**

1. **Người dùng (quản trị viên nhóm)** đăng nhập vào hệ thống và tạo một nhóm mới.
2. Sau khi tạo nhóm, quản trị viên có thể **mời bạn bè tham gia nhóm**.
3. Hệ thống xử lý lời mời và **gửi thông báo đến người dùng được mời**.
4. **Người dùng (khác)** nhận thông báo và có thể **chấp nhận hoặc từ chối** lời mời:
   * Nếu **từ chối** → quá trình dừng lại.
   * Nếu **chấp nhận**, người đó trở thành thành viên nhóm và có quyền tương tác với nội dung trong nhóm.
5. Sau khi có thành viên, quản trị viên có thể:
   * **Đăng, sửa hoặc xóa bài viết** trong nhóm.
   * Với một số nhóm có yêu cầu kiểm duyệt, bài viết sẽ được **gửi đi kiểm duyệt**.
   * Hệ thống **hiển thị thông báo cho quản trị viên** để xử lý.
6. **Quản trị viên duyệt bài viết**:
   * Nếu **từ chối**, bài viết sẽ không hiển thị.
   * Nếu **duyệt**, hệ thống sẽ **hiển thị bài viết lên nhóm** cho các thành viên.
7. Các **thành viên nhóm** có thể:
   * **Đăng bài viết** nếu được quyền.
   * **Tương tác với bài viết** (Like/Share) trong phạm vi nhóm.

**Các vai trò tham gia:**

* **Người dùng (Quản trị viên)**: Tạo nhóm, mời bạn bè, duyệt bài, quản lý nội dung nhóm.
* **Người dùng (Khác)**: Thành viên được mời vào nhóm, có quyền xem hoặc đăng bài tùy chính sách.
* **Hệ thống**: Xử lý thông báo, kiểm duyệt, lưu trữ và hiển thị nội dung nhóm.

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 12: Biểu đồ Tham gia và quản lý nhóm

### 6.8. Hệ thống thông báo

**Mô tả luồng xử lý:**

1. Người dùng thực hiện hành động **nhấn xem thông báo** từ biểu tượng thông báo trên giao diện.
2. Hệ thống tiến hành **hiển thị nội dung thông báo** tương ứng như: có lời mời kết bạn, có bình luận mới, có bài viết được thích, ...
3. Sau khi xem thông báo, người dùng có thể **tùy chỉnh hành động tiếp theo tùy theo loại thông báo**, ví dụ:
   * Truy cập vào bài viết được nhắc đến.
   * Chấp nhận hoặc từ chối lời mời kết bạn.
   * Vào khung chat nếu là thông báo tin nhắn.
4. Hệ thống thực hiện hành động tương ứng, đồng thời **cập nhật nội dung thông báo** như: đánh dấu đã đọc, thay đổi trạng thái, ...

**Các vai trò tham gia:**

* **Người dùng**: Tác nhân khởi tạo hành động xem và xử lý thông báo.
* **Hệ thống**: Hiển thị nội dung thông báo và cập nhật dữ liệu liên quan.

A diagram of a diagram

AI-generated content may be incorrect.

Hình 13: Biểu đồ Quản lý thông báo

# CHƯƠNG III: THIẾT KẾ HỆ THỐNG

## Lựa chọn mô hình kiến trúc

### 1.1. Kiến trúc tổng thể

Ứng dụng được xây dựng theo mô hình **Client-Server** với kiến trúc **MVC (Model-View-Controller)** kết hợp **SPA (Single Page Application)** và **Realtime**:

* **Backend (Server):** Laravel đảm nhiệm xử lý logic nghiệp vụ, truy xuất dữ liệu, xác thực, phân quyền, cung cấp API và WebSocket (Laravel Reverb).
* **Frontend (Client):** Vue.js kết hợp Inertia.js xây dựng giao diện động, tương tác mượt mà như SPA.
* **Realtime:** Laravel Reverb và Laravel Echo giúp truyền dữ liệu thời gian thực (chat, thông báo).

### 1.2. Chi tiết các tầng

#### Tầng trình diễn (Frontend)

* **Vue.js:** Xây dựng các component giao diện, quản lý trạng thái, xử lý sự kiện người dùng.
* **Inertia.js:** Đóng vai trò cầu nối giữa Laravel và Vue, giúp frontend nhận dữ liệu từ backend mà không cần API REST thuần.
* **Bootstrap:** Thiết kế giao diện responsive, thân thiện.
* **Laravel Echo:** Lắng nghe các sự kiện realtime từ server (qua Reverb).

#### Tầng điều phối (Controller)

* **Laravel Controller:** Nhận request từ client, gọi các repository/service để xử lý logic, trả về response hoặc render view qua Inertia.
* **Xử lý xác thực, phân quyền:** Sử dụng middleware của Laravel để bảo vệ route.

#### Tầng truy xuất dữ liệu (Repository)

* **Repository Pattern:** Tách biệt logic truy vấn dữ liệu khỏi controller, giúp code dễ bảo trì, kiểm thử, mở rộng.
* **Eloquent ORM:** Quản lý các model, quan hệ dữ liệu, truy vấn database.

#### Tầng dữ liệu (Database)

* **MySQL:** Lưu trữ dữ liệu người dùng, bài viết, nhóm, tin nhắn, v.v.
* **Migration:** Quản lý cấu trúc bảng dữ liệu.

#### Tầng realtime (WebSocket)

* **Laravel Reverb:** Server WebSocket tích hợp, truyền dữ liệu realtime (chat, thông báo…).
* **Laravel Echo:** Kết nối frontend với WebSocket server, nhận sự kiện realtime.

### 1.3. Luồng hoạt động tiêu biểu

* 1. Người dùng thao tác trên giao diện (Vue.js).
  2. Gửi request (qua Inertia hoặc Axios) đến Laravel Controller.
  3. Controller gọi Repository để truy xuất hoặc cập nhật dữ liệu.
  4. Repository thao tác với Eloquent Model và Database.
  5. Kết quả trả về Controller, rồi trả về Frontend qua Inertia.
  6. Nếu có sự kiện realtime (ví dụ chat), Controller phát sự kiện, Laravel Reverb gửi tới client qua WebSocket, client nhận bằng Laravel Echo.

### 1.4. Ưu điểm kiến trúc

* **Tách biệt rõ ràng các tầng:** Dễ bảo trì, mở rộng, kiểm thử.
* **Realtime mạnh mẽ:** Nhờ Laravel Reverb và Echo.
* **SPA mượt mà:** Nhờ Vue.js và Inertia.js.
* **Bảo mật tốt:** Nhờ middleware, CSRF, xác thực Laravel.
* **Dễ triển khai:** Có thể chạy local (XAMPP) hoặc deploy lên hosting/VPS.

## Sơ đồ kiến trúc tổng thể

A screenshot of a computer

AI-generated content may be incorrect.

Hình 14: Sơ đồ kiến trúc tổng thể

### 2.1. Design Pattern

**Repository Pattern** là một mẫu thiết kế (design pattern) phổ biến trong phát triển phần mềm, đặc biệt với các ứng dụng sử dụng kiến trúc MVC như Laravel. Mục đích của Repository Pattern là tách biệt logic truy xuất dữ liệu (data access logic) khỏi logic nghiệp vụ (business logic), giúp mã nguồn dễ bảo trì, kiểm thử và mở rộng.

#### Khái niệm

* **Repository** đóng vai trò như một “kho lưu trữ” trung gian giữa tầng truy cập dữ liệu (database, API, ...) và tầng nghiệp vụ (service/controller).
* Thay vì truy vấn dữ liệu trực tiếp trong controller hoặc service, ta gọi các phương thức của repository.
* Repository che giấu chi tiết truy xuất dữ liệu, cung cấp một API rõ ràng cho các thao tác như lấy, thêm, sửa, xóa dữ liệu.

#### Cấu trúc Repository Pattern

* **Repository Interface:** Định nghĩa các phương thức mà repository phải triển khai (ví dụ: find, all, create, update, delete...).
* **Repository Implementation:** Hiện thực các phương thức của interface, sử dụng Eloquent, Query Builder hoặc các nguồn dữ liệu khác.
* **Controller/Service:** Chỉ gọi các phương thức của repository, không truy vấn dữ liệu trực tiếp.

## Biểu đồ tuần tự

### 3.1. Sequence Đăng nhập

A diagram of a project

AI-generated content may be incorrect.

Hình 15: Sequence Đăng nhập

### 3.2. Sequence Đăng xuất

A screenshot of a computer

AI-generated content may be incorrect.

Hình 16: Sequence Đăng xuất

### 3.3. Sequence Cập nhật thông tin tài khoản

A blue and white diagram

AI-generated content may be incorrect.

Hình 17: Sequence Cập nhật tài khoản

### 3.4. Sequence Sửa bài viết

A screenshot of a computer

AI-generated content may be incorrect.

Hình 18: Sequence Sửa bài viết

### 3.5. Sequence Sửa nhóm

A screenshot of a computer

AI-generated content may be incorrect.

Hình 19: Sequence Sửa nhóm

### 3.6. Sequence Thêm bài viết

A blue and white diagram

AI-generated content may be incorrect.

Hình 20: Sequence Thêm bài viết

### 3.7. Sequence Tạo nhóm

A diagram of a project

AI-generated content may be incorrect.

Hình 21: Sequence Tạo nhóm

### 3.8. Sequence xóa thông tin bài viết

A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 22: Sequence Xóa bài viết

### *3.*9. Sequence Gửi lời mời kết bạn

A diagram of a diagram

AI-generated content may be incorrect.

Hình 23: Sequence Gửi lời mời kết bạn

### 3.10. Sequence Chấp nhận lời mời kết bạn

A diagram of a diagram

AI-generated content may be incorrect.

Hình 24: Sequence Chấp nhận lời mời kết bạn

### 3.11. Sequence Hủy lời mời kết bạn

A diagram of a diagram

AI-generated content may be incorrect.

Hình 27: Sequence Hủy lời mời kết bạn

### 3.12. Sequence Bình luận

A diagram of a computer

AI-generated content may be incorrect.

Hình 25: Sequence Bình luận

### 3.13. Sequence Xóa bình luận

A diagram of a diagram

AI-generated content may be incorrect.

Hình 26: Sequence Xóa bình luận

### 3.14. Sequence Bày tỏ cảm xúc

A diagram of a diagram

AI-generated content may be incorrect.

Hình 27: Sequence Bày tỏ cảm xúc

### 3.15. Sequence Cập nhật bày tỏ cảm xúc bài viết

A diagram of a diagram

AI-generated content may be incorrect.

Hình 28: Sequence Cập nhật bày tỏ cảm xúc bài viết

### 3.16. Sequence Hủy bỏ cảm xúc bài viết

A diagram of a diagram

AI-generated content may be incorrect.

Hình 29: Sequence Hủy bỏ cảm xúc bài viết

### 3.17. Sequence Nhắn tin

A diagram of a diagram

AI-generated content may be incorrect.

Hình 30: Sequence Nhắn tin

## Class

A diagram of a computer program

AI-generated content may be incorrect.

Hình 31: Class

## 5. Thiết kế cơ sở dữ liệu (ERD, bảng, mối quan hệ)

A computer screen shot of a diagram

AI-generated content may be incorrect.

Hình 32: Sơ đồ ERD

Mô tả chi tiết các bảng cơ sở dữ liệu mạng xã hội

**Bảng: users**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã người dùng, khóa chính, tự tăng |
| name | VARCHAR(255) | Họ và tên người dùng |
| username | VARCHAR(255), UNIQUE | Tên đăng nhập duy nhất |
| email | VARCHAR(255), UNIQUE | Email người dùng, không trùng nhau |
| phone | VARCHAR(20) | Số điện thoại |
| birthday | DATE | Ngày sinh |
| password | VARCHAR(255) | Mật khẩu được mã hóa |
| role | ENUM('admin', 'user') | Phân quyền: admin hoặc user |

**Bảng: posts**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã bài viết |
| user\_id | INT, FK | Người đăng, khóa ngoại tham chiếu bảng users |
| content | TEXT | Nội dung bài viết |
| privacy\_setting | BOOLEAN | Cài đặt riêng tư: công khai hoặc riêng tư |
| allow\_comments | BOOLEAN | Cho phép người khác bình luận không |
| created\_at | DATETIME | Thời gian tạo bài viết |

**Bảng: comments**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã bình luận |
| post\_id | INT, FK | Khóa ngoại tới bài viết |
| user\_id | INT, FK | Người bình luận |
| content | TEXT | Nội dung bình luận |
| created\_at | DATETIME | Thời gian bình luận |

**Bảng: likes**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã lượt tương tác |
| post\_id | INT, FK | Bài viết được tương tác |
| user\_id | INT, FK | Người tương tác |
| reaction\_type | VARCHAR(50) | Loại cảm xúc: like, love, haha... |

**Bảng: media**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã phương tiện |
| post\_id | INT, FK | Bài viết chứa ảnh/video |
| media\_url | VARCHAR(255) | Đường dẫn ảnh/video |
| media\_type | ENUM('image', 'video') | Loại phương tiện |

**Bảng: groups**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã nhóm |
| name | VARCHAR(255) | Tên nhóm |
| description | TEXT | Mô tả nhóm |
| cover\_photo\_url | VARCHAR(255) | Ảnh bìa |
| privacy\_setting | BOOLEAN | Nhóm công khai hay kín |
| approval\_required | BOOLEAN | Có yêu cầu duyệt khi tham gia không |

**Bảng: member\_group**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| user\_id | INT, FK | Người dùng tham gia nhóm |
| group\_id | INT, FK | Nhóm tham gia |
| status | VARCHAR(50) | Trạng thái: pending, joined |
| user\_id | INT, FK | Người dùng tham gia nhóm |
| group\_id | INT, FK | Nhóm được tham gia |
| status | VARCHAR(50) | Trạng thái thành viên: pending, approved |

**Bảng: conversations**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã cuộc trò chuyện |
| name | VARCHAR(255) | Tên cuộc trò chuyện |
| type | VARCHAR(50) | Loại: nhóm hoặc cá nhân |

**Bảng: member\_conversation**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| user\_id | INT, FK | Người tham gia |
| conversation\_id | INT, FK | Cuộc trò chuyện |
| user\_id | INT, FK | Người dùng tham gia cuộc trò chuyện |
| conversation\_id | INT, FK | Cuộc trò chuyện được tham gia |

**Bảng: messages**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã tin nhắn |
| conversation\_id | INT, FK | Cuộc trò chuyện |
| sender\_id | INT, FK | Người gửi |
| content | TEXT | Nội dung tin nhắn |
| sent\_at | DATETIME | Thời gian gửi |

**Bảng: notifications**

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| id | INT, PK | Mã thông báo |
| user\_id | INT, FK | Người nhận thông báo |
| message | TEXT | Nội dung thông báo |
| read\_at | DATETIME | Thời gian đọc |
| is\_read | BOOLEAN | Đã đọc hay chưa |

## 6. Thiết kế giao diện người dùng

### 6.1. Giao diện đăng nhập

A screenshot of a login form

AI-generated content may be incorrect.

Hình 33: Giao diện đăng nhập

### 6.2. Giao diện đăng ký tài khoản

A screenshot of a login form

AI-generated content may be incorrect.

Hình 34: Giao diện đăng ký tài khoản

### 6.3. Giao diện trang chủ

A screenshot of a computer

AI-generated content may be incorrect.

Hình 35: Giao diện trang chủ

### 6.4. Giao diện quản lí tin nhắn

A screenshot of a computer

AI-generated content may be incorrect.

Hình 36: Giao diện quản lí tin nhắn

### 6.5. Giao diện nhắn tin riêng

A screenshot of a computer

AI-generated content may be incorrect.

Hình 37: Giao diện nhắn tin riêng

### 6.6. Giao diện tạo cuộc hội thoại mới

A screenshot of a computer

AI-generated content may be incorrect.

Hình 38: Giao diện tạo cuộc hội thoại mới

### 6.7. Giao diện tạo cuộc trò chuyện nhóm

A screenshot of a computer

AI-generated content may be incorrect.

Hình 39: Giao diện tạo cuộc trò chuyện nhóm

### 6.8. Giao diện tìm kiếm cuộc trò chuyện

A screenshot of a computer

AI-generated content may be incorrect.

Hình 40: Giao diện tìm kiếm cuộc trò chuyện

### 6.9. Giao diện Quản lý tài khoản cá nhân

A screenshot of a computer

AI-generated content may be incorrect.

Hình 41: Giao diện Quản lý tài khoản cá nhân

### 6.10. Giao diện tạo bài viết

A screenshot of a computer

AI-generated content may be incorrect.

Hình 42: Giao diện tạo bài viết

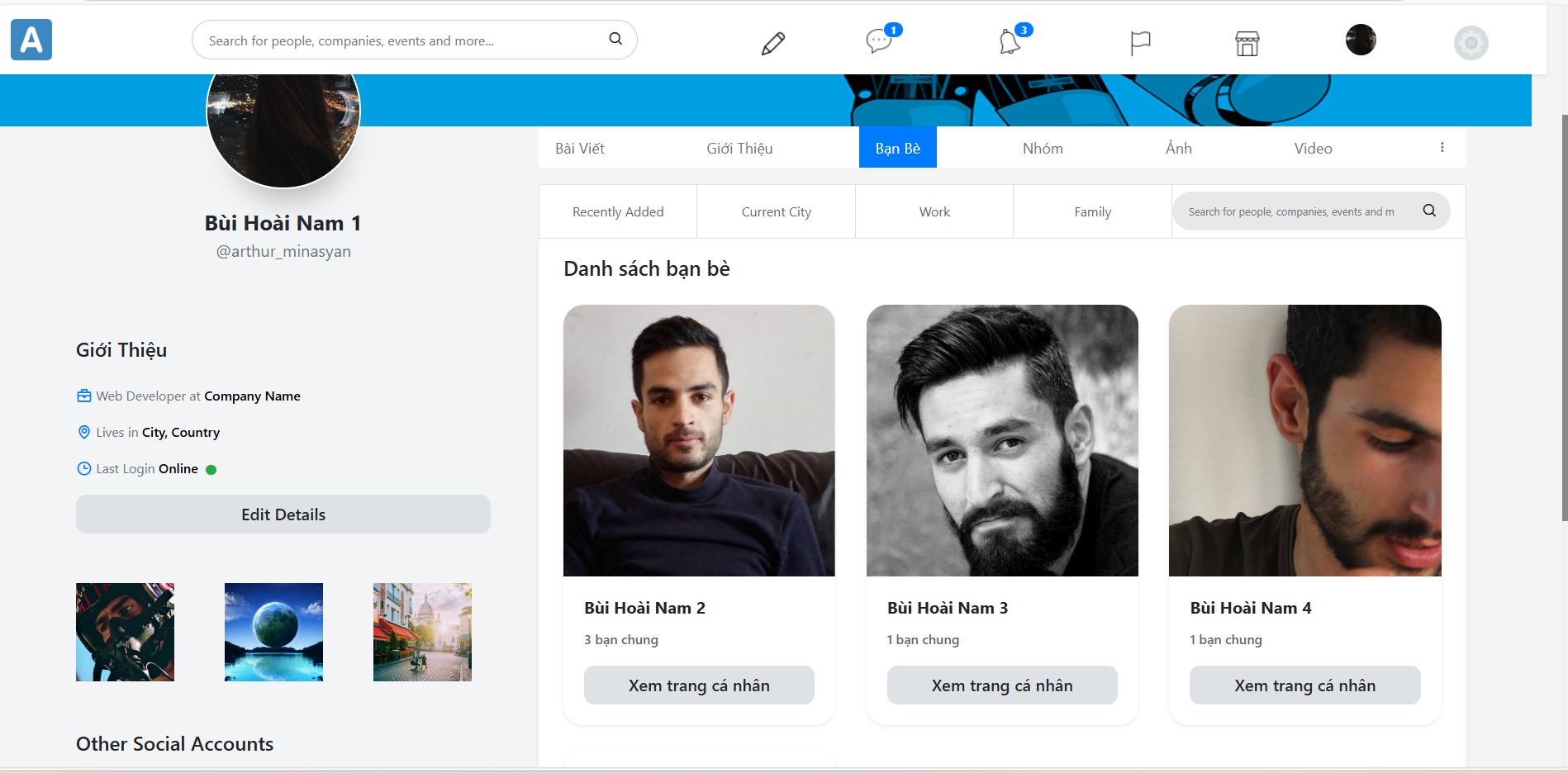
### 6.11. Giao diện lựa chọn chế độ hiển thị khi tạo bài viết

A screenshot of a computer

AI-generated content may be incorrect.

Hình 43: Giao diện lựa chọn chế độ hiển thị khi tạo bài viết

### 6.12. Giao diện Quản lý bạn bè cá nhân



Hình 44: Giao diện Quản lý bạn bè cá nhân

### 6.13. Giao diện tìm kiếm bạn bè

A screenshot of a social media account

AI-generated content may be incorrect.

Hình 45: Giao diện tìm kiếm bạn bè

### 6.14. Giao diện xem trang cá nhân của bạn bè

A screenshot of a chat

AI-generated content may be incorrect.

Hình 46: Giao diện xem trang cá nhân của bạn bè

### 6.15. Giao diện xem chi tiết trang cá nhân bạn bè

A screenshot of a computer

AI-generated content may be incorrect.

Hình 47: Giao diện xem chi tiết trang cá nhân bạn bè

### 6.16. Giao diện kết bạn

A computer screen shot of a blue robot

AI-generated content may be incorrect.

Hình 48: Giao diện kết bạn

### 6.17. Giao diện sau khi nhấn gửi kết bạn

A screenshot of a computer

AI-generated content may be incorrect.

Hình 49: Giao diện sau khi nhấn gửi kết bạn

### 6.18. Giao diện khi kết bạn thành công

A screenshot of a computer

AI-generated content may be incorrect.

Hình 50: Giao diện sau khi kết bạn thành công

### 6.19. Giao diện xem danh sách bạn bè khác của bạn bè

A screenshot of a computer

AI-generated content may be incorrect.

Hình 51: Giao diện xem danh sách bạn bè khác của bạn bè

### 6.20. Giao diện thông báo

A screenshot of a chat

AI-generated content may be incorrect.

Hình 52: Giao diện thông báo

### 6.21. Giao diện cài đặt tài khoản và đăng xuất

A screenshot of a computer

AI-generated content may be incorrect.

Hình 53: Giao diện cài đặt tài khoản và đăng xuất

### 6.22. Giao diện sửa thông tin tài khoản

A screenshot of a computer

AI-generated content may be incorrect.

Hình 54: Giao diện sửa thông tin tài khoản

# CHƯƠNG IV: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

## 1. Môi trường triển khai

Hệ thống được cài đặt và triển khai trên môi trường máy tính cá nhân, sử dụng các công cụ và phần mềm mã nguồn mở phổ biến:

* **Hệ điều hành:** Windows 10 hoặc cao hơn
* **Trình máy chủ web:** XAMPP (Apache + MySQL)
* **Ngôn ngữ backend:** PHP 8.x sử dụng framework Laravel 10
* **Ngôn ngữ frontend:** JavaScript sử dụng framework Vue.js 3
* **Cơ sở dữ liệu:** MySQL
* **Trình duyệt kiểm thử:** Google Chrome, Firefox
* **Công cụ hỗ trợ:** Composer, Node.js & npm, Laravel Mix, VS Code

## 2. Cài đặt hệ thống

### 2.1. Yêu cầu hệ thống

**Phần cứng:**

* RAM: ≥ 4GB
* CPU: ≥ Intel Core i3
* Ổ cứng: ≥ 1GB trống

**Phần mềm:**

* PHP >= 8.x
* Composer
* Node.js & npm
* XAMPP (Apache, MySQL)
* Laravel CLI
* Trình duyệt web

### 2.2. Các bước cài đặt chương trình

**Cài đặt môi trường:**

* Cài đặt XAMPP để khởi động Apache và MySQL.
* Cài đặt Composer, Node.js và npm.

**Tạo cơ sở dữ liệu:**

* Mở phpMyAdmin → tạo cơ sở dữ liệu tên social\_network.

**Clone hoặc copy mã nguồn:**

* Đặt mã nguồn Laravel trong thư mục htdocs/ (ví dụ: htdocs/social-app).

**Cài đặt Laravel:**

composer install

cp .env.example .env

php artisan key:generate

**Cấu hình .env:**

* Thiết lập kết nối cơ sở dữ liệu:

DB\_DATABASE=social\_network

DB\_USERNAME=root

DB\_PASSWORD=

**Chạy migration và seeder:**

php artisan migrate --seed

**Cài đặt frontend (Vue.js):**

npm install

npm run dev

**Chạy hệ thống:**

php artisan serve

* Truy cập tại địa chỉ: http://127.0.0.1:8000/

# CHƯƠNG V: KẾT QUẢ VÀ ĐÁNH GIÁ

## 1. Kết quả thử nghiệm hệ thống

Hệ thống đã được xây dựng và thử nghiệm thành công với các chức năng chính sau:

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Trạng thái** | **Ghi chú** |
| Đăng ký / Đăng nhập | ✅ | Có xác thực đầu vào |
| Tạo / Hiển thị bài viết | ✅ | Có hỗ trợ hình ảnh |
| Gửi / Chấp nhận lời mời kết bạn | ✅ | Tự động cập nhật danh sách bạn bè |
| Thả tim / Bình luận | ✅ | Hiển thị thời gian thực |
| Chat trực tiếp giữa bạn bè | ✅ | Sử dụng WebSocket giả lập |
| Xem trang cá nhân | ✅ | Hiển thị bài viết và thông tin |
| Responsive (giao diện phản hồi) | ✅ | Trải nghiệm tốt trên desktop |

## 2. Đánh giá hiệu quả hệ thống

* Hệ thống hoạt động ổn định trong môi trường giả lập cục bộ (local).
* Các chức năng được triển khai đầy đủ theo yêu cầu ban đầu, đảm bảo tính trực quan và thân thiện với người dùng.
* Việc áp dụng Laravel và Vue.js giúp hệ thống phân tách rõ backend và frontend, dễ bảo trì, mở rộng và học hỏi.
* Tính năng chat trực tiếp mô phỏng tốt mô hình giao tiếp thời gian thực.
* Tuy nhiên, hệ thống hiện vẫn còn giới hạn
* Chưa triển khai được upload ảnh lên máy chủ thực.
* Chưa có cơ chế thông báo (notification).
* Chưa triển khai bảo mật nâng cao như CSRF, JWT cho API.

# CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Ứng dụng mạng xã hội được xây dựng bằng PHP Laravel ở backend và Vue.js ở frontend đã hoàn thiện các chức năng cốt lõi như: đăng ký/đăng nhập, kết bạn, nhắn tin, đăng bài, bình luận và tương tác. Hệ thống đảm bảo tính bảo mật, tốc độ xử lý tốt và giao diện thân thiện với người dùng. Việc kết hợp Laravel và Vue giúp tách biệt rõ giữa xử lý logic và giao diện, tăng khả năng mở rộng trong tương lai. Qua quá trình phát triển, nhóm đã tích lũy được nhiều kinh nghiệm thực tiễn về thiết kế hệ thống, tối ưu hiệu suất và quản lý phiên làm việc của người dùng.

## Hạn chế

Mặc dù ứng dụng đã đáp ứng được các chức năng cơ bản, tuy nhiên vẫn còn tồn tại một số hạn chế nhất định:

* Hệ thống bảo mật mới dừng ở mức cơ bản (sử dụng cơ chế xác thực Laravel mặc định), chưa triển khai xác thực hai lớp hoặc cảnh báo đăng nhập bất thường.
* Chưa có hệ thống phân quyền theo vai trò (role-based access control) cho các đối tượng như quản trị viên, kiểm duyệt viên.
* Hiệu năng và khả năng chịu tải chưa được kiểm thử với dữ liệu và lưu lượng người dùng lớn, chưa áp dụng các kỹ thuật như caching, phân cụm hay queue xử lý nền.

Những hạn chế này là cơ sở để định hướng cải tiến ứng dụng trong các giai đoạn phát triển tiếp theo.

## Hướng phát triển

Để hoàn thiện hơn nữa và đáp ứng được nhu cầu thực tiễn của người dùng, trong tương lai, nhóm định hướng phát triển ứng dụng theo các hướng sau:

* **Hỗ trợ đa ngôn ngữ và mở rộng thị trường** quốc tế bằng cách tích hợp các gói ngôn ngữ và cơ chế nhận diện vị trí địa lý của người dùng.
* **Tích hợp hệ thống phân quyền và quản trị nội dung** để hỗ trợ đội ngũ kiểm duyệt, đảm bảo môi trường sử dụng lành mạnh.

Những định hướng này không chỉ giúp hoàn thiện sản phẩm hiện tại mà còn tạo tiền đề cho việc mở rộng quy mô, nâng cao chất lượng và tăng sức cạnh tranh của ứng dụng trên thị trường mạng xã hội.

# TÀI LIỆU THAM KHẢO

1. Tài liệu chính thức Laravel: <https://laravel.com/docs>

2. Tài liệu chính thức Vue.js: <https://vuejs.org/guide/introduction.html>

3. StackOverflow, Dev.to