**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙞🙞🙞🙞-----

A blue logo with a ball and a black background

Description automatically generated

**TIỂU LUẬN CHUYÊN NGÀNH**

**Chuyên ngành: Hệ Thống Thông Tin**

**XÂY DỰNG HỆ THỐNG MẠNG XÃ HỘI CODE4LIFE**

**(công nghệ ReactJS và Spring Boot)**

**GV hướng dẫn:** : TS. Nguyễn Thành Sơn.

**SV thực hiện:** Tào Việt Đức.

**MSSV:** 21110169.

**Học kỳ**: 1.

**Năm học**: 2024 – 2025.

Thành phố Hồ Chí Minh, 28 tháng 11, năm 2024

**ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH**

**SV thực hiện:** Tào Việt Đức.                     **Mã Số SV:** 21110169.

**Thời gian làm tiểu luận:** từ: 19/08/2024 đến: (10/12/2025)

**Chuyên ngành:** Hệ Thống Thông Tin.

**Đề tài:** XÂY DỰNG HỆ THỐNG MẠNG XÃ HỘI CODE4LIFE (công nghệ ReactJS và Spring Boot).

**GV hướng dẫn:** TS. Nguyễn Thành Sơn.

**NHIỆM VỤ**

**1. Xác định Đề Tài, Mục Tiêu và Phạm Vi.**

**Đề Tài:** Phát triển hệ thống mạng xã hội Code4Life sử dụng công nghệ ReactJS và Spring Boot.

**Mục Tiêu:** Mục tiêu chính của đề tài là xây dựng và phát triển mạng xã hội Code4Life với những tính năng cơ bản nhưng hữu ích nhằm đáp ứng tốt các nhu cầu của cộng đồng IT. Cụ thể, Code4Life sẽ tạo ra một không gian cho việc chia sẻ kiến thức, học hỏi qua các bài viết và diễn đàn hỏi đáp, đồng thời cung cấp kho lưu trữ cho các dự án lập trình. Ngoài ra, nền tảng này hỗ trợ kết nối giữa các thành viên trong cộng đồng, giúp họ xây dựng mạng lưới quan hệ chuyên nghiệp thông qua các mối quan hệ mentor-mentee. Bên cạnh đó, Code4Life sẽ phát triển các tính năng như nhóm học tập và cộng tác, cho phép người dùng cùng nhau học hỏi và làm việc hiệu quả. Đặc biệt, hệ thống sẽ đảm bảo tính bảo mật cao và hiệu suất tối ưu để tạo sự tin tưởng và an toàn cho người dùng.

**Phạm Vi:** Phạm vi của đề tài này sẽ tập trung vào việc phát triển các tính năng cơ bản của mạng xã hội Code4Life, nhằm đáp ứng nhu cầu thiết yếu của cộng đồng IT. Đầu tiên, em sẽ triển khai chức năng đăng ký và đăng nhập, cung cấp phương thức xác thực qua email và mạng xã hội để đảm bảo an toàn cho người dùng. Nền tảng cũng sẽ bao gồm trang cá nhân, cho phép người dùng quản lý thông tin cá nhân, kỹ năng và dự án tham gia. Diễn đàn hỏi đáp sẽ tạo không gian cho việc đặt câu hỏi và thảo luận về công nghệ, trong khi kho lưu trữ dự án lập trình giúp người dùng chia sẻ và tìm kiếm các dự án mã nguồn mở. Tính năng nhóm học tập và cộng tác sẽ hỗ trợ người dùng trong việc tạo và tham gia vào các nhóm học tập, giúp nâng cao kỹ năng lẫn nhau. Cuối cùng, hệ thống sẽ có tính năng tương tác thời gian thực, đảm bảo người dùng nhận thông báo và tương tác nhanh chóng. Nhưng phạm vi này sẽ không bao gồm các tính năng phức tạp như thương mại điện tử hay tích hợp dịch vụ bên thứ ba, mà tập trung vào xây dựng một nền tảng ổn định và dễ sử dụng cho cộng đồng IT.

**2. Nghiên cứu và Phân Tích**

**Phân Tích Nhu Cầu**: Xác định nhu cầu của cộng đồng IT đối với một nền tảng mạng xã hội. Tìm hiểu các tính năng cần thiết và các vấn đề người dùng gặp phải trên các nền tảng hiện tại.

**Nghiên Cứu Công Nghệ:** Đánh giá các công nghệ phù hợp để phát triển ứng dụng. So sánh ưu nhược điểm của ReactJS và Spring Boot để đảm bảo tính khả thi của dự án.

**3. Thiết Kế và Lập Kế Hoạch**

**Thiết Kế Hệ Thống:** Xây dựng kiến trúc tổng thể cho hệ thống, bao gồm thiết kế frontend và backend. Xác định các module chính và cách thức hoạt động của chúng.

**Kế Hoạch Phát Triển:** Lên kế hoạch chi tiết cho quá trình phát triển, bao gồm các bước phát triển, kiểm thử, và triển khai. Xác định các mốc thời gian quan trọng và phân công nhiệm vụ.

**4. Phát Triển và Triển Khai**

**Phát Triển Frontend:** Xây dựng giao diện người dùng bằng ReactJS. Đảm bảo giao diện tương tác tốt và dễ sử dụng.

**Phát Triển Backend:** Xây dựng các dịch vụ và API bằng Spring Boot. Thiết kế cơ sở dữ liệu và tích hợp các tính năng bảo mật.

**Triển Khai Ứng Dụng:** Đưa ứng dụng vào môi trường thực tế. Sử dụng các công cụ và dịch vụ cloud để triển khai và quản lý hệ thống.

**5. Kiểm Thử và Đánh Giá**

**Kiểm Thử Chức Năng:** Thực hiện các bài kiểm tra để đảm bảo tất cả các tính năng hoạt động đúng như dự kiến. Xác định và sửa lỗi nếu có.

**Đánh Giá Hiệu Năng và Bảo Mật:** Đánh giá hiệu suất hệ thống và các cơ chế bảo mật. Đảm bảo ứng dụng có thể xử lý tốt lượng người dùng lớn và bảo vệ dữ liệu người dùng.

**6. Báo Cáo và Trình Bày**

**Viết Báo Cáo:** Tạo báo cáo chi tiết về quá trình phát triển, kết quả đạt được, và những vấn đề gặp phải. Bao gồm cả các hình ảnh, biểu đồ và số liệu minh họa.

**Trình Bày Kết Quả:** Chuẩn bị một bài thuyết trình để trình bày kết quả dự án. Giải thích các tính năng của hệ thống, công nghệ sử dụng, và lợi ích của nền tảng mạng xã hội Code4Life.

**7. Rút Kinh Nghiệm và Đề Xuất Phát Triển**

**Rút Kinh Nghiệm:** Đánh giá các bài học rút ra từ quá trình phát triển. Xem xét những gì đã hoạt động tốt và những điểm cần cải thiện.

**Đề Xuất Phát Triển:** Đưa ra các đề xuất cho các tính năng hoặc cải tiến trong tương lai. Xác định các cơ hội mở rộng và phát triển thêm cho nền tảng.

**MỤC LỤC**

[**I.** **PHẦN MỞ ĐẦU** 13](#_Toc183601789)

[**II.** **PHẦN NỘI DUNG** 14](#_Toc183601790)

[**Chương 1: TỔNG QUAN VỀ CÔNG NGHỆ ÁP DỤNG.** 14](#_Toc183601791)

[**Chương 2: TỔNG QUAN VỀ CÁC MẠNG XÃ HỘI PHỔ BIẾN.** 18](#_Toc183601792)

[**Chương 3: PHÂN TÍCH THIẾT KẾ MẠNG XÃ HỘI.** 23](#_Toc183601793)

[**3.1**  **ĐẶC TẢ ĐỀ TÀI.** 23](#_Toc183601794)

[**3.2** **PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU.** 27](#_Toc183601795)

[**3.2.1** **PHÂN TÍCH THIẾT KẾ MỨC QUAN NIỆM. \** 27](#_Toc183601796)

[**3.2.1.1 CRC User.** 28](#_Toc183601797)

[**3.2.1.2 CRC Post.** 28](#_Toc183601798)

[**3.2.1.3 CRC Comment**. 29](#_Toc183601799)

[**3.2.1.4 CRC Message**. 29](#_Toc183601800)

[**3.2.1.5 CRC Group.** 30](#_Toc183601801)

[**3.2.1.6 CRC Course.** 30](#_Toc183601802)

[**3.2.1.7 CRC Subject**. 31](#_Toc183601803)

[**3.2.1.8 CRC Notification.** 32](#_Toc183601804)

[**3.2.2**  **PHÂN TÍCH THIẾT KẾ MỨC LOGIC.** 32](#_Toc183601805)

[**3.2.2.1 Kiểu Dữ Liệu, Khóa Chính và Khóa Ngoại.** 32](#_Toc183601806)

[**3.2.2.1.1 Collection: User.** 32](#_Toc183601807)

[**3.2.2.1.2 Collection: Post.** 33](#_Toc183601808)

[**3.2.2.1.3 Collection: Comment.** 33](#_Toc183601809)

[**3.2.2.1.4 Collection: Notification.** 33](#_Toc183601810)

[**3.2.2.1.5 Collection: Message.** 34](#_Toc183601811)

[**3.2.2.1.6 Collection: Group.** 34](#_Toc183601812)

[**3.2.2.1.7 Collection: Course.** 34](#_Toc183601813)

[**3.2.2.1.8 Collection: Subject (Lesson).** 35](#_Toc183601814)

[**3.2.2.2 Các Phương Thức.** 36](#_Toc183601815)

[**3.2.2.2.1 Collection: User.** 36](#_Toc183601816)

[**3.2.2.2.2 Collection: Post.** 37](#_Toc183601817)

[**3.2.2.2.3 Collection: Comment.** 38](#_Toc183601818)

[**3.2.2.2.4 Collection: Notification.** 38](#_Toc183601819)

[**3.2.2.2.5 Collection: Message.** 39](#_Toc183601820)

[**3.2.2.2.6 Collection: Group.** 39](#_Toc183601821)

[**3.2.2.2.7 Collection: Course.** 41](#_Toc183601822)

[**3.2.2.2.8 Collection: Subject.** 42](#_Toc183601823)

[**3.2.3**  **PHÂN TÍCH THIẾT KẾ MỨC VẬT LÝ.** 43](#_Toc183601824)

[**3.3** **PHÂN TÍCH THIẾT KẾ THÀNH PHẦN CHỨC NĂNG.** 43](#_Toc183601825)

[**3.3.1 Activity Diagram.** 43](#_Toc183601826)

[**3.3.1.1 Biểu đồ hoạt động đăng ký tài khoản.** 43](#_Toc183601827)

[**3.3.1.2 Biểu đồ hoạt động đăng nhập.** 44](#_Toc183601828)

[**3.3.1.3 Biểu đồ hoạt động đổi mật khẩu.** 44](#_Toc183601829)

[**3.3.1.4 Biểu đồ hoạt động quên mật khẩu.** 45](#_Toc183601830)

[**3.3.1.5 Biểu đồ hoạt động chỉnh sửa thông tin tài khoản.** 46](#_Toc183601831)

[**3.3.1.6 Biểu đồ hoạt động xoá tài khoản.** 46](#_Toc183601832)

[**3.3.1.7 Biểu đồ hoạt động tạo bài viết.** 47](#_Toc183601833)

[**3.3.1.8 Biểu đồ hoạt động chỉnh sửa bài viết.** 47](#_Toc183601834)

[**3.3.1.9 Biểu đồ hoạt động xoá bài viết.** 48](#_Toc183601835)

[**3.3.1.10 Biểu đồ hoạt động private-public bài viết.** 48](#_Toc183601836)

[**3.3.1.11 Biểu đồ hoạt động tạo comment.** 49](#_Toc183601837)

[**3.3.1.12 Biểu đồ hoạt động chỉnh sửa comment.** 49](#_Toc183601838)

[**3.3.1.13 Biểu đồ hoạt động xoá comment.** 49](#_Toc183601839)

[**3.3.1.14 Biểu đồ hoạt động gửi tin nhắn.** 50](#_Toc183601840)

[**3.3.1.15 Biểu đồ hoạt động thu hồi tin nhắn 2 phía.** 50](#_Toc183601841)

[**3.3.1.16 Biểu đồ hoạt động chỉnh sửa tin nhắn.** 51](#_Toc183601842)

[**3.3.1.17 Biểu đồ hoạt động theo dõi bạn bè.** 51](#_Toc183601843)

[**3.3.1.18 Biểu đồ hoạt động theo dõi lại.** 52](#_Toc183601844)

[**3.3.1.19 Biểu đồ hoạt động bỏ theo dõi.** 52](#_Toc183601845)

[**3.3.1.20 Biểu đồ hoạt động tạo Group.** 53](#_Toc183601846)

[**3.3.1.21 Biểu đồ hoạt động chỉnh sửa thông tin Group.** 53](#_Toc183601847)

[**3.3.1.22 Biểu đồ hoạt động xoá Group.** 54](#_Toc183601848)

[**3.3.1.23 Biểu đồ hoạt động join Group.** 54](#_Toc183601849)

[**3.3.1.24 Biểu đồ hoạt động out Group.** 54](#_Toc183601850)

[**3.3.1.25 Biểu đồ hoạt động thêm thành viên vào Group.** 55](#_Toc183601851)

[**3.3.1.26 Biểu đồ hoạt động xoá thành viên khỏi Group.** 55](#_Toc183601852)

[**3.3.1.27 Biểu đồ hoạt động video call.** 55](#_Toc183601853)

[**3.3.1.28 Biểu đồ hoạt động tạo khoá học.** 56](#_Toc183601854)

[**3.3.1.29 Biểu đồ hoạt động chỉnh sửa thông tin khoá học.** 56](#_Toc183601855)

[**3.3.1.30 Biểu đồ hoạt động xoá khoá học.** 56](#_Toc183601856)

[**3.3.1.31 Biểu đồ hoạt động thêm môn học.** 57](#_Toc183601857)

[**3.3.1.32 Biểu đồ hoạt động huỷ môn học.** 57](#_Toc183601858)

[**3.3.1.33 Biểu đồ hoạt động đăng ký khoá học.** 58](#_Toc183601859)

[**3.3.1. 34 Biểu đồ hoạt động huỷ đăng ký khoá học.** 59](#_Toc183601860)

[**3.3.2 Use Case Diagram.** 59](#_Toc183601861)

[**3.3.3 Sequence Diagram.** 59](#_Toc183601862)

[**Chương 4: CÀI ĐẶT MẠNG XÃ HỘI** 59](#_Toc183601863)

[**Chương 5: TRIỂN KHAI MẠNG XÃ HỘI** 59](#_Toc183601864)

[**Chương 6: KẾT LUẬN** 60](#_Toc183601865)

[**III.** **Phần TÀI LIỆU THAM KHẢO** 60](#_Toc183601866)

**KẾ HOẠCH THỰC HIỆN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thời gian** | **Công việc** | **Người thực hiện** | **Trạng thái** |
| 1 | Tuần 1  (19/8-25/8) | Tìm hiểu về đề tài chuẩn bị những công cần thiết. | Tào Việt Đức | DONE |
| 2 | Tuần 2  (26/8-1/9) | Tiến hành lên bố cục sơ lược cho đề tài. | Tào Việt Đức | DONE |
| 3 | Tuần 3  (2/9-8/9) | Thực hiện viết Chương 1: TỔNG QUAN VỀ CÔNG NGHỆ ÁP DỤNG. | Tào Việt Đức | DONE |
| 4 | Tuần 4  (9/9-15/9) | Chỉnh sửa  Chương 1 và viết lại cho hoàn chỉnh. | Tào Việt Đức | DONE |
| 5 | Tuần 5  (16/9-22/9) | Thực hiện viết Chương 2: TỔNG QUAN VỀ MỘT VÀI MẠNG XÃ HỘI PHỔ BIẾN.. | Tào Việt Đức | DONE |
| 6 | Tuần 6  (23/9-29/9) | Chỉnh sửa  Chương 2 và viết lại cho hoàn chỉnh. | Tào Việt Đức | DONE |
| 7 | Tuần 7  (30/9-6/10) | Thực hiện viết Chương 3:  ĐẶC TẢ ĐỀ TÀI. | Tào Việt Đức | DONE |
| 8 | Tuần 8  (7/10-13/10) | Chỉnh sửa  Chương 3 và viết lại cho hoàn chỉnh. | Tào Việt Đức | DONE |
| 9 | Tuần 9  (14/10-20/10) | Hoàn thiện  chương 3 | Tào Việt Đức | DONE |
| 10 | Tuần 10  (21/10-27/10) | Thực hiện  làm & viết Chương 4:  ĐẶC TẢ ĐỀ TÀI. | Tào Việt Đức | DONE |
| 11 | Tuần 11  (28/10-3/11) | Chỉnh sửa  Chương 4 và viết lại cho hoàn chỉnh. | Tào Việt Đức | DONE |
| 12 | Tuần 12  (4/11-10/11) | Chỉnh sửa  Chương 4 và viết lại cho hoàn chỉnh. | Tào Việt Đức | DONE |
| 13 | Tuần 13 (11/11-17/11) | Hoàn thiện  chương 4 | Tào Việt Đức |  |
| 14 | Tuần 14 (18/11-24/11) | Thực hiện  làm & viết Chương 5 | Tào Việt Đức |  |
| 15 | Tuần 15  (25/8-1/12) | Hoàn thiện  chương 5 + 6 | Tào Việt Đức |  |
| 16 | Tuần 16 (8/12 - 10/12) | Làm slide thuyết trình + photo thành cuốn + chuẩn bị trình bày và phản biện | Tào Việt Đức |  |

Ngày 28 tháng 11 năm 2024

**Người Viết Đề Cương**

Tào Việt Đức

**Ý kiến của giáo viên hướng dẫn**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

(Ký và ghi rõ họ tên)

# **PHẦN MỞ ĐẦU**

Trong kỷ nguyên số, việc chia sẻ kiến thức và kết nối cộng đồng đã trở nên dễ dàng hơn bao giờ hết nhờ vào sự phát triển mạnh mẽ của các nền tảng mạng xã hội. Đặc biệt, đối với cộng đồng công nghệ thông tin (IT), nhu cầu tìm kiếm thông tin, học hỏi và trao đổi kinh nghiệm là vô cùng cần thiết. Với mong muốn tạo ra một không gian riêng biệt, kết nối những người đam mê IT và hỗ trợ việc học tập, nghiên cứu, mạng xã hội Code4Life đã ra đời.

Code4Life không chỉ là nơi tập trung các lập trình viên, kỹ sư phần mềm mà còn là nền tảng để các thành viên tham gia vào các nhóm học tập cũng như chia sẻ thông tin trong các lĩnh vực khác nhau. Về mặt công nghệ, đề tài “Xây dựng mạng xã hội Code4Life” sử dụng hai công nghệ chính là ReactJS và Spring Boot. Cụ thể, ReactJS sẽ đóng vai trò chủ đạo trong việc phát triển giao diện người dùng, tạo nên trải nghiệm tương tác linh hoạt. Trong khi đó, Spring Boot đảm nhiệm phần xử lý phía server, mang lại hiệu suất cao cũng như đảm bảo tính bảo mật cho hệ thống.

Qua đề tài này, em sẽ trình bày chi tiết về quá trình xây dựng mạng xã hội, từ việc phân tích yêu cầu, thiết kế cơ sở dữ liệu, phát triển các tính năng chính, đến triển khai và thử nghiệm. Mục tiêu cuối cùng của emlà tạo ra một mạng xã hội không chỉ đáp ứng nhu cầu kết nối cộng đồng IT mà còn hỗ trợ các hoạt động học tập cũng như chia sẻ kiến thức. Hy vọng rằng Code4Life sẽ trở thành một mạng xã, đồng hành cùng sự phát triển của cộng đồng công nghệ thông tin, và góp phần nâng cao hiệu quả trong việc học hỏi và trao đổi kinh nghiệm giữa các thành viên trong cộng đồng IT.

# **PHẦN NỘI DUNG**

## **Chương 1: TỔNG QUAN VỀ CÔNG NGHỆ ÁP DỤNG.**

Trong thời đại công nghệ phát triển mạnh mẽ như hiện nay, việc xây dựng một trang web thường yêu cầu kết hợp nhiều công nghệ phổ biến và tiên tiến, bao gồm frontend (giao diện người dùng), backend (xử lý máy chủ) và cơ sở dữ liệu (để lưu trữ, quản lý, và truy xuất dữ liệu). Sau một thời gian tìm hiểu và nghiên cứu, em đã có những hiểu biết về các công nghệ cần thiết để xây dựng một dự án web hoàn chỉnh.

Đối với phần frontend, có các framework và thư viện nổi bật như React.js, Vue.js, Angular, và nhiều công nghệ khác. Mỗi framework đều cung cấp những giải pháp linh hoạt và hiệu quả, giúp tối ưu hóa quá trình phát triển cũng như mang lại trải nghiệm người dùng tốt hơn.

Đầu tiên, React.js là một thư viện JavaScript mạnh mẽ được phát triển bởi Facebook, và đã trở thành một trong những công nghệ phổ biến nhất trong việc phát triển giao diện người dùng nhờ tính linh hoạt và khả năng tối ưu hóa hiệu suất. Với khả năng tái sử dụng các component, React cho phép nhà phát triển xây dựng các ứng dụng web phức tạp mà vẫn đảm bảo hiệu suất nhanh chóng và dễ bảo trì. Điều này giúp React phù hợp cho cả các dự án nhỏ lẫn các ứng dụng web lớn đòi hỏi khả năng mở rộng và tương tác cao.

Trong khi đó, Vuejs lại được đánh giá cao nhờ tính đơn giản và nhẹ nhàng. Khác với React, Vue.js không yêu cầu phải học quá nhiều khái niệm mới, giúp các nhà phát triển dễ dàng làm quen và tích hợp vào các dự án hiện có. Vì vậy, Vue thường được lựa chọn cho các dự án nhỏ hoặc trung bình, nơi tính dễ dùng và tốc độ triển khai là ưu tiên hàng đầu. Mặc dù nhẹ nhưng Vue.js vẫn cung cấp đủ công cụ để phát triển những ứng dụng có tính tương tác cao.

Ngược lại, Angular, một framework toàn diện được phát triển bởi Google, lại phù hợp hơn cho các dự án lớn và phức tạp. Angular không chỉ cung cấp đầy đủ công cụ để xây dựng giao diện người dùng mà còn tích hợp sẵn các tính năng như dependency injection, routing, và quản lý trạng thái, giúp việc phát triển các ứng dụng có quy mô lớn trở nên dễ dàng hơn. Tuy nhiên, sự phức tạp của Angular đòi hỏi nhà phát triển phải có kiến thức sâu rộng và đầu tư nhiều thời gian hơn để thành thạo, nhưng đổi lại, nó mang lại sức mạnh và tính mở rộng rất cao trong các hệ thống lớn.

Trong lĩnh vực phát triển Backend (xử lý máy chủ), có nhiều công nghệ mạnh mẽ được sử dụng, bao gồm các nền tảng như Node.js, Django (Python), Ruby on Rails (Ruby), Spring Boot (Java), và ASP.NET Core (C#), và mỗi công nghệ đều có những đặc điểm riêng biệt, phục vụ cho các nhu cầu phát triển ứng dụng khác nhau.

Đầu tiên, Node.js là một môi trường chạy JavaScript trên máy chủ, và với tính năng không đồng bộ (asynchronous), nó cho phép phát triển các ứng dụng nhanh chóng cũng như dễ dàng mở rộng quy mô. Do đó, Node.js là lựa chọn phổ biến cho các ứng dụng thời gian thực như chat hay hệ thống theo dõi, và thích hợp cho cả các ứng dụng web đơn giản lẫn phức tạp.

Trong khi đó, Django, một framework nổi tiếng viết bằng Python, được biết đến với khả năng bảo mật cao và sự đơn giản trong việc phát triển ứng dụng. Nhờ vào các công cụ tích hợp sẵn, Django giúp các nhà phát triển nhanh chóng triển khai các ứng dụng có độ phức tạp cao, và đặc biệt phù hợp cho các dự án đòi hỏi sự ổn định cũng như bảo mật như hệ thống quản lý dữ liệu hoặc các ứng dụng thương mại điện tử.

Ngoài ra, Ruby on Rails, một framework dựa trên ngôn ngữ Ruby và tuân theo mô hình MVC (Model-View-Controller), nổi bật với tính dễ học và khả năng phát triển nhanh chóng. Với tư tưởng “Convention over Configuration” (Ưu tiên quy ước thay vì cấu hình), Rails giúp các nhà phát triển tập trung vào logic của ứng dụng thay vì cấu hình chi tiết, do đó nó là lựa chọn tuyệt vời cho các dự án startup hay các ứng dụng cần ra mắt nhanh chóng.

Ngược lại, Spring Boot, viết bằng Java, lại là một framework lý tưởng cho các hệ thống lớn và phức tạp. Với khả năng mở rộng và các tính năng bảo mật tích hợp sẵn, Spring Boot thường được sử dụng trong các hệ thống doanh nghiệp, nơi đòi hỏi tính ổn định và khả năng quản lý hiệu suất cao. Điều này làm cho Spring Boot trở thành một lựa chọn hàng đầu trong các ứng dụng quy mô lớn và yêu cầu xử lý dữ liệu phức tạp.

Cuối cùng, ASP.NET Core, một framework phát triển bởi Microsoft, là lựa chọn phổ biến trong các dự án doanh nghiệp lớn, vì nó cung cấp khả năng tích hợp tốt với các sản phẩm Microsoft khác và hỗ trợ phát triển ứng dụng đa nền tảng. Với hiệu suất cao và tính bảo mật tốt, ASP.NET Core thích hợp cho các ứng dụng doanh nghiệp, trang web thương mại điện tử, và các hệ thống yêu cầu mức độ bảo mật cao.

Về phần cơ sở dữ liệu, công nghệ được sử dụng thường thay đổi tùy thuộc vào yêu cầu cụ thể của từng dự án, bao gồm các công nghệ như MySQL, PostgreSQL, và MongoDB.

Đầu tiên, MySQL là một cơ sở dữ liệu quan hệ phổ biến, được đánh giá cao nhờ tính miễn phí và dễ sử dụng, vì vậy nó thường là lựa chọn hàng đầu cho các dự án nhỏ và trung bình, nơi mà yêu cầu về hiệu suất không quá phức tạp.

Bên cạnh đó, PostgreSQL nổi bật với vai trò là một cơ sở dữ liệu quan hệ mạnh mẽ hơn, và được tối ưu hóa cho các ứng dụng quy mô lớn, đặc biệt thích hợp cho những dự án yêu cầu xử lý phức tạp và khối lượng dữ liệu lớn, giúp đảm bảo tính ổn định và hiệu suất cao trong suốt quá trình hoạt động.

Ngoài ra, MongoDB, một cơ sở dữ liệu NoSQL phổ biến, cung cấp khả năng lưu trữ dữ liệu không có cấu trúc, do đó nó thường được sử dụng trong các ứng dụng có khối lượng dữ liệu lớn hoặc các hệ thống cần khả năng mở rộng nhanh chóng, chẳng hạn như các ứng dụng thời gian thực hoặc dữ liệu phi cấu trúc. Tóm lại, sự lựa chọn giữa các công nghệ cơ sở dữ liệu này phụ thuộc vào các yếu tố như quy mô dự án, loại dữ liệu cần lưu trữ, cũng như yêu cầu về hiệu suất và khả năng mở rộng của ứng dụng.

Giữa một rừng các công nghệ hiện có, em quyết định chọn ReactJS cho phần frontend và Spring Boot cho backend, còn cơ sở dữ liệu thì em sử dụng cả MySQL và MongoDB cho đề tài xây dựng mạng xã hội mang tên Code4Life, đồng thời đây cũng là đề tài tiểu luận chuyên ngành của em trong học kỳ này. Lý do em chọn ReactJS là vì nó cung cấp khả năng tái sử dụng các component, giúp phát triển giao diện người dùng một cách linh hoạt và hiệu quả, cùng với tính năng tối ưu hóa hiệu suất. Trong khi đó, Spring Boot lại là một framework mạnh mẽ cho backend, cung cấp đầy đủ các công cụ cần thiết để xây dựng một mạng xã hội có quy mô lớn và phức tạp, đặc biệt là khả năng bảo mật và tích hợp dễ dàng với các dịch vụ khác.

Về phần cơ sở dữ liệu, em chọn kết hợp cả MySQL và MongoDB. MySQL là sự lựa chọn hoàn hảo cho các dữ liệu có cấu trúc như thông tin người dùng, bài viết, vì nó miễn phí và dễ sử dụng. Còn MongoDB, với khả năng linh hoạt và lưu trữ dữ liệu phi cấu trúc, sẽ hỗ trợ tốt cho việc lưu trữ các nội dung như dự án lập trình, hình ảnh hay các file liên quan.

Tóm lại, trong chương này em đã tìm hiểu và nghiên cứu để đưa ra các đặc điểm của từng công nghệ ứng với từng phần frontend (giao diện người dùng), backend (xử lý máy chủ) và cơ sở dữ liệu (lưu trữ, quản lý, và truy xuất dữ liệu). Cuối cùng, em quyết định chọn sự kết hợp giữa ReactJS, Spring Boot, MySQL và MongoDB vì chúng không chỉ đáp ứng được yêu cầu kỹ thuật trong đề tài tiểu luận chuyên ngành của em mà còn giúp em phát triển kỹ năng lập trình một cách toàn diện.

## **Chương 2: TỔNG QUAN VỀ CÁC MẠNG XÃ HỘI PHỔ BIẾN.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tiêu chí** | **Facebook** | **Instagram** | **TikTok** | **Threads** |
| **Năm thành lập** | 2004 | 2010 | 2016 | 2023 |
| **Người sáng lập** | Mark Zuckerberg cùng cộng sự | Kevin Systrom, Mike Krieger | ByteDance  (Trung Quốc) | Meta Platforms (Facebook) |
| **Số người dùng (2023)** | 3 tỷ người dùng hoạt động hàng tháng | 2 tỷ người dùng hoạt động hàng tháng | Hơn 1 tỷ lượt tải xuống | Mới ra mắt, chưa có số liệu cụ thể |
| **Mục đích chính** | Kết nối, chia sẻ cá nhân, kinh doanh | Chia sẻ hình ảnh, video | Chia sẻ video ngắn, giải trí | Thảo luận văn bản ngắn gọn |
| **Ưu điểm chính** | - Kết nối toàn cầu  - Nền tảng tiếp thị mạnh mẽ  - Hỗ trợ học tập | - Giao diện trực quan  - Khả năng kết nối cộng đồng  - Tiếp thị hiệu quả | - Sáng tạo nội dung dễ dàng  - Tiếp cận đối tượng rộng  - Lan truyền nhanh | - Tích hợp với Instagram  - Giao diện đơn giản  - Tập trung vào văn bản |
| **Nhược điểm chính** | - Quyền riêng tư  - Tin tức giả  - Nghiện mạng xã hội | - Áp lực về hình ảnh cá nhân  - Nghiện mạng xã hội  - Thông tin sai lệch | - Quyền riêng tư  - Gây nghiện  - Nội dung không kiểm duyệt | - Phụ thuộc Instagram  - Cạnh tranh với Twitter  - Nội dung hạn chế |
| **Tính năng chính** | - Bài đăng, tin nhắn  - Facebook Ads  - Marketplace | - Bộ lọc ảnh, video  - Stories  - Reels  - Instagram Ads | - Video ngắn  - Bộ lọc và hiệu ứng  - Livestream | - Chia sẻ văn bản ngắn  - Thảo luận trực tiếp  - Kết nối qua Instagram |
| **Đối tượng sử dụng** | Mọi đối tượng, doanh nghiệp, người dùng cá nhân | Người dùng cá nhân, doanh nghiệp, influencer | Giới trẻ, nhà sáng tạo nội dung, người yêu giải trí | Người dùng Instagram, người thích thảo luận văn bản |
| **Tích hợp quảng cáo** | Facebook Ads | Instagram Ads | Quảng cáo hợp tác với influencer | Chưa rõ ràng |
| **Mức độ bảo mật** | Đã từng có vấn đề về quyền riêng tư và dữ liệu | Gặp một số vấn đề về quyền riêng tư | Bị chỉ trích về bảo mật dữ liệu | Phụ thuộc hệ sinh thái Meta |
| **Khả năng mở rộng cộng đồng** | Rất lớn, kết nối toàn cầu | Lớn, dựa vào hình ảnh và video | Rất lớn, dựa vào nội dung lan truyền nhanh chóng | Đang phát triển, tận dụng lượng người dùng Instagram |

**Bảng so sánh các nền tảng mạng xã hội hiện nay**

Sau quá trình nghiên cứu các mạng xã hội đang có ở hiện tại và ứng dụng vào việc phát triển đề tài Mạng xã hội Code4Life, việc xây dựng một mạng xã hội chuyên biệt dành cho cộng đồng IT như Mạng xã hội Code4Life đòi hỏi phải học hỏi và phát huy những ưu điểm từ các nền tảng mạng xã hội hiện có, đồng thời khắc phục những nhược điểm của chúng để tạo ra một hệ sinh thái mạnh mẽ, hỗ trợ học tập, kết nối và phát triển cá nhân. Qua quá trình nghiên cứu các mạng xã hội như Facebook, Instagram, TikTok, Threads có nhiều điểm sáng có thể ứng dụng vào Mạng xã hội Code4Life, cùng với những bài học quan trọng đã được rút ra.

Đầu tiên về ưu điểm được học hỏi từ 4 mạng xã hội Facebook, Instagram, TikTok, Threads từ đó cần phát huy những ưu điểm này khi xây dựng Mạng xã hội Code4Life:

Tính tương tác cao và trải nghiệm người dùng (UX): Giống như Facebook và Instagram, Code4Life cần chú trọng vào giao diện người dùng thân thiện, dễ sử dụng, và tương tác mượt mà. ReactJS sẽ là công cụ giúp phát triển một giao diện linh hoạt, tương tác cao, giúp người dùng cảm thấy thoải mái và dễ dàng tiếp cận. Việc tối ưu hóa trải nghiệm người dùng (UX) không chỉ giúp thu hút người dùng mới mà còn giữ chân các thành viên hiện tại, đặc biệt là những người trong cộng đồng IT vốn có yêu cầu cao về mặt kỹ thuật và sự tiện lợi.

Hỗ trợ kết nối cộng đồng (Networking): Lấy cảm hứng từ LinkedIn, Code4Life cần đẩy mạnh tính năng kết nối giữa các thành viên trong cộng đồng IT. Việc phát triển các công cụ để xây dựng mạng lưới quan hệ chuyên nghiệp sẽ giúp các lập trình viên, kỹ sư phần mềm và các chuyên gia công nghệ kết nối với nhau một cách dễ dàng. Các tính năng như tạo nhóm học tập, cộng tác dự án, mentor-mentee sẽ hỗ trợ tối đa trong việc trao đổi kiến thức và phát triển kỹ năng, từ đó giúp tăng giá trị cho người dùng.

Tính năng nhóm học tập và chia sẻ kiến thức: Giống như Stack Overflow và GitHub, Code4Life có thể khuyến khích người dùng tham gia vào các diễn đàn hỏi đáp, chia sẻ tài liệu và bài viết hướng dẫn, cũng như kho lưu trữ các dự án lập trình. Điều này sẽ tạo nên một cộng đồng học tập tích cực, nơi mọi người có thể cùng nhau phát triển kỹ năng và kiến thức. Đặc biệt, những tính năng hỗ trợ nhóm học tập theo chuyên đề sẽ giúp người dùng học hỏi sâu hơn trong những lĩnh vực họ quan tâm, từ đó phát triển sự nghiệp trong ngành công nghệ thông tin.

Tính bảo mật và hiệu suất cao: Với nền tảng Spring Boot, Code4Life sẽ đảm bảo hiệu suất hoạt động cao và tính bảo mật tuyệt đối, đặc biệt khi xử lý các dữ liệu nhạy cảm như thông tin cá nhân và dữ liệu học tập. Học hỏi từ các nền tảng như GitHub và LinkedIn, bảo mật và hiệu suất là yếu tố không thể thiếu trong một nền tảng mạng xã hội công nghệ thông tin. Điều này không chỉ giúp xây dựng niềm tin với người dùng mà còn tạo sự khác biệt với các mạng xã hội khác.

Có phát huy về mặt ưu điểm thì cũng phải khắc phục được các nhược điểm đang là hạn chế trong các mạng xã hội hiện nay để mang lại sự tối ưu cho người dùng:

Quảng cáo và quản lý nội dung: Một trong những nhược điểm lớn của các mạng xã hội phổ biến như Facebook là việc nội dung quảng cáo chiếm quá nhiều không gian, gây khó chịu cho người dùng. Hơn nữa, việc quản lý nội dung trên các nền tảng này đôi khi không được hiệu quả, dẫn đến việc phát tán thông tin không phù hợp hoặc không đúng đối tượng. Để tránh những vấn đề này, Code4Life cần xây dựng cơ chế quản lý nội dung rõ ràng, bảo đảm rằng các quảng cáo không ảnh hưởng tiêu cực đến trải nghiệm người dùng. Tập trung vào nội dung liên quan đến học tập và phát triển công nghệ sẽ giúp giữ vững giá trị của nền tảng.

Tính cá nhân hóa: Mặc dù LinkedIn là một nền tảng kết nối chuyên nghiệp, nhưng lại chưa tối ưu trong việc cá nhân hóa trải nghiệm cho từng người dùng. Để tránh điểm yếu này, Code4Life cần phát triển hệ thống cá nhân hóa nội dung, giúp người dùng dễ dàng tìm kiếm các thông tin phù hợp với nhu cầu học tập, công việc, và phát triển kỹ năng của họ. Tính cá nhân hóa cao sẽ giúp người dùng cảm thấy họ được quan tâm và hỗ trợ đúng nhu cầu, từ đó nâng cao trải nghiệm tổng thể trên nền tảng.

Quản lý sự tham gia và giữ chân người dùng: Twitter gặp vấn đề trong việc mất dần người dùng do không cung cấp đủ tính năng để giữ chân họ. Code4Life có thể khắc phục nhược điểm này bằng cách xây dựng các hoạt động cộng đồng hấp dẫn, chẳng hạn như tổ chức hackathon, các cuộc thi lập trình, và hệ thống đánh giá xếp hạng thành viên theo những đóng góp của họ. Những hoạt động này không chỉ tạo ra sự gắn kết giữa các thành viên mà còn khuyến khích họ tham gia tích cực hơn vào cộng đồng.

Qua quá trình nghiên cứu các nền tảng mạng xã hội, Code4Life có thể học hỏi và phát huy nhiều ưu điểm nổi bật như tính tương tác cao, kết nối cộng đồng, bảo mật và hiệu suất vượt trội. Đồng thời, việc khắc phục những nhược điểm như quảng cáo quá tải, thiếu cá nhân hóa và khó khăn trong việc giữ chân người dùng sẽ giúp Code4Life trở thành một nền tảng đáng tin cậy và hữu ích cho cộng đồng IT. Hy vọng rằng, Code4Life không chỉ là nơi để chia sẻ kiến thức mà còn là ngôi nhà chung của các lập trình viên, kỹ sư phần mềm và những người đam mê công nghệ trên toàn thế giới.

## **Chương 3: PHÂN TÍCH THIẾT KẾ MẠNG XÃ HỘI.**

### **3.1 ĐẶC TẢ ĐỀ TÀI.**

Trong thời đại số hóa hiện nay, mạng xã hội đã trở thành cầu nối quan trọng giúp con người chia sẻ kiến thức và kết nối với nhau. Đặc biệt đối với cộng đồng công nghệ thông tin (IT), việc giao lưu, học hỏi và trao đổi kinh nghiệm trở nên vô cùng cần thiết. Với mong muốn tạo ra một không gian chuyên biệt, mạng xã hội Code4Life được phát triển để đáp ứng nhu cầu kết nối của cộng đồng IT, từ lập trình viên đến kỹ sư phần mềm và những người đam mê công nghệ. Code4Life không chỉ giúp liên kết các cá nhân trong cộng đồng, mà còn hỗ trợ mạnh mẽ quá trình học tập, nghiên cứu và phát triển bản thân của các cá nhân có niềm đam mê với Công Nghệ Thông Tin.

Yêu cầu kỹ thuật cho mạng xã hội Code4Life tập trung vào việc sử dụng các công nghệ tiên tiến nhằm tối ưu hóa trải nghiệm người dùng và đảm bảo hiệu suất, bảo mật cao. ReactJS sẽ được sử dụng để phát triển giao diện người dùng, mang lại trải nghiệm mượt mà và linh hoạt, giúp người dùng dễ dàng tương tác với nền tảng. Phần xử lý server sẽ được đảm nhiệm bởi Spring Boot, không chỉ giúp cải thiện hiệu suất mà còn tăng cường tính bảo mật cho hệ thống.

Cơ sở dữ liệu sẽ được quản lý bằng MySQL và MongoDB, đảm bảo việc lưu trữ và quản lý hiệu quả các thông tin của người dùng cũng như các nội dung như bài viết và dự án lập trình. Hệ thống xác thực sẽ được hỗ trợ bởi JWT (JSON Web Token), nhằm đảm bảo an toàn cho người dùng khi đăng nhập và sử dụng nền tảng. Tính năng tương tác thời gian thực giữa các thành viên sẽ được thực hiện thông qua WebSocket, giúp cải thiện trải nghiệm thông báo và trao đổi trực tiếp. Sau đó, hệ thống sẽ được triển khai và mở rộng nhanh chóng trên các nền tảng đám mây như AWS hoặc GCP, giúp đảm bảo tính linh hoạt và khả năng mở rộng của Code4Life trong tương lai.

Code4Life được xây dựng với nhiều tính năng chính nhằm phục vụ cộng đồng IT một cách toàn diện và hiệu quả. Trước hết, người dùng có thể dễ dàng đăng ký và đăng nhập thông qua email, mật khẩu hoặc tích hợp với các mạng xã hội khác. Mỗi thành viên sẽ có một trang cá nhân (Profile) riêng, nơi họ có thể hiển thị thông tin về kỹ năng, kinh nghiệm làm việc, các dự án tham gia, và mục tiêu học tập của mình.

Tính năng kết nối cộng đồng và nhóm học tập cho phép người dùng tạo và tham gia vào các nhóm học tập và dự án cộng tác, giúp mở rộng mạng lưới quan hệ trong cộng đồng IT. Diễn đàn hỏi đáp, tương tự như Stack Overflow, giúp người dùng đăng câu hỏi, trả lời và thảo luận về các chủ đề lập trình, công nghệ, với những câu hỏi hữu ích được cộng đồng bình chọn và đánh giá.

Ngoài ra, Code4Life còn cung cấp một kho lưu trữ mã nguồn và dự án lập trình, tương tự như GitHub, để người dùng chia sẻ và học hỏi từ các dự án thực tế. Người dùng cũng có thể chia sẻ kiến thức qua các bài viết, giúp lan tỏa kinh nghiệm và kỹ năng trong lĩnh vực công nghệ.

Hệ thống thông báo và tương tác thời gian thực sẽ giúp người dùng cập nhật nhanh chóng về các hoạt động như bình luận, trả lời câu hỏi, thích bài viết hoặc cập nhật dự án. Một tính năng khác là hệ thống xếp hạng thành viên, nơi người dùng sẽ được đánh giá và xếp hạng dựa trên những đóng góp cho cộng đồng như bài viết, câu trả lời, dự án, và tương tác tích cực.

Đặc biệt, nền tảng sẽ đảm bảo có các cơ chế quản lý nội dung chặt chẽ và bảo mật cao, đảm bảo thông tin cá nhân và dữ liệu của người dùng được bảo vệ, đồng thời ngăn chặn các nội dung không phù hợp có thể gây ảnh hưởng tiêu cực đến cộng đồng.

Cơ sở dữ liệu của Code4Life được xây dựng với mục tiêu quản lý toàn bộ thông tin liên quan đến người dùng, bài viết, dự án, nhóm học tập và các tương tác trong cộng đồng. Các bảng dữ liệu được tổ chức theo cấu trúc rõ ràng và hợp lý. Bảng User là bảng chính, quản lý các thông tin về tài khoản người dùng, bao gồm tên, email, mật khẩu và các chi tiết cá nhân khác. Tiếp theo, bảng Group lưu trữ thông tin về các nhóm học tập và cộng tác, giúp người dùng tham gia vào các nhóm dễ dàng. Bảng Post được thiết kế để lưu trữ nội dung bài viết, câu hỏi và câu trả lời từ người dùng, nhằm tạo ra một môi trường chia sẻ kiến thức và thảo luận. Bảng Project quản lý thông tin về các dự án lập trình mà người dùng có thể chia sẻ hoặc tham gia, từ đó thúc đẩy sự học hỏi từ các dự án thực tế. Để tăng cường sự tương tác, bảng Comment lưu trữ tất cả bình luận và phản hồi về bài viết, dự án, trong khi bảng Rating quản lý thông tin về điểm số và xếp hạng thành viên dựa trên mức độ đóng góp của họ cho cộng đồng. Với thiết kế cơ sở dữ liệu này, Code4Life sẽ tối ưu hóa việc quản lý thông tin và tạo ra một nền tảng mạnh mẽ, kết nối và hỗ trợ tốt cho cộng đồng IT.

Ngoài ra, cơ sở dữ liệu cũng bao gồm các bảng bổ sung như Project\_members, Assignments, Classes, và Class\_enrollments, hỗ trợ quản lý các dự án lập trình, bài tập và khóa học mà người dùng tham gia. Bảng Course\_materials lưu trữ các tài liệu học tập liên quan đến từng khóa học, trong khi bảng Achievements và User\_achievements giúp theo dõi thành tựu mà người dùng đạt được, khuyến khích sự tham gia tích cực. Bảng Tags và Post\_tags được thêm vào để phân loại và tổ chức nội dung, giúp người dùng dễ dàng tìm kiếm thông tin liên quan. Private\_messages và Activity\_log theo dõi các tin nhắn riêng tư và hoạt động của người dùng, tăng cường sự tương tác và kết nối trong cộng đồng.

Với việc tích hợp tất cả các bảng này, cơ sở dữ liệu sẽ quản lý thông tin toàn diện hơn và dễ dàng mở rộng khi nền tảng phát triển, đồng thời hỗ trợ các tính năng quản lý dự án, khóa học, động lực tham gia, và tương tác giữa các thành viên một cách hiệu quả.

Quy trình phát triển mạng xã hội Code4Life được chia thành nhiều giai đoạn cụ thể nhằm đảm bảo rằng sản phẩm cuối cùng đáp ứng tốt nhu cầu của cộng đồng IT.

Đầu tiên, chúng ta tiến hành phân tích yêu cầu, nơi mà các yêu cầu từ cộng đồng IT được thu thập và xác định, từ đó làm rõ các tính năng cốt lõi cần phát triển. Giai đoạn tiếp theo là thiết kế hệ thống, trong đó kiến trúc tổng quan, giao diện người dùng và cơ sở dữ liệu sẽ được phác thảo chi tiết, tạo nền tảng cho việc phát triển sau này.

Khi thiết kế đã hoàn tất, chúng ta sẽ bước vào giai đoạn phát triển tính năng. Tại đây, các tính năng chính như đăng ký, đăng nhập, tạo nhóm, diễn đàn, chia sẻ dự án lập trình và quản lý nội dung sẽ được lập trình, mang lại giá trị thực tiễn cho người dùng.

Sau khi phát triển, hệ thống sẽ được đưa vào giai đoạn kiểm thử. Các hoạt động kiểm thử sẽ được thực hiện để đảm bảo rằng tất cả các tính năng hoạt động đúng cách, đồng thời đảm bảo rằng hệ thống đạt tiêu chuẩn về bảo mật và hiệu suất cao.

Khi mọi thứ đã sẵn sàng, em triển khai hệ thống lên môi trường đám mây, cho phép người dùng có thể dễ dàng truy cập và sử dụng. Cuối cùng, nhằm hoàn thiện sản phẩm, em sẽ thu thập phản hồi từ một nhóm người dùng thử nghiệm. Những phản hồi này sẽ giúp cải thiện hệ thống trước khi phát hành chính thức. Thông qua quy trình phát triển này, Code4Life mong muốn tạo ra một nền tảng mạnh mẽ và hữu ích cho cộng đồng IT.

Code4Life hứa hẹn sẽ trở thành một mạng xã hội chuyên biệt cho cộng đồng IT, tạo môi trường học tập, chia sẻ và phát triển mạnh mẽ. Với sự hỗ trợ của các công nghệ tiên tiến như ReactJS và Spring Boot, cùng với các tính năng hướng đến cộng đồng như nhóm học tập, diễn đàn hỏi đáp, kho lưu trữ dự án lập trình và hệ thống mentor-mentee, Code4Life sẽ đáp ứng tối đa nhu cầu kết nối và phát triển của các lập trình viên và kỹ sư phần mềm. Hệ thống không chỉ đảm bảo tính bảo mật và hiệu suất mà còn tạo nên một không gian học tập và giao lưu hiệu quả cho cộng đồng IT.

### **3.2 PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU.**

#### **3.2.1 PHÂN TÍCH THIẾT KẾ MỨC QUAN NIỆM. \**

A computer screen shot of a computer

Description automatically generated

**Lược Đồ Class Diagram Hệ Thống Mạng Xã Hội Code4Life**

##### **3.2.1.1 CRC User.**

|  |  |
| --- | --- |
| **Tên lớp: User.** | **ID: 1.** |
| **Mô tả:** Lớp User đại diện cho người dùng trong hệ thống, chịu trách nhiệm quản lý thông tin cá nhân, xử lý đăng nhập và đăng ký tài khoản và các chức năng trong hệ thống. | **Use case liên quan:** |
| **Trách nhiệm:**  **-         Đăng ký.**  **-         Đăng nhập.**  - **Đăng content.**  **- Trao đổi thông tin.** | **Cộng tác:**   * **Post** * **Comment** * **Notification** * **Course** * **Group** * **Message** |
| **Thuộc tính:** userID: String , username: String , password: String , email: String , status: String , createdDate: Date, following: String. | |
| **Các mối quan hệ:**  -         **Chuyên biệt hóa:** Quản trị viên, người dùng. | |

##### **3.2.1.2 CRC Post.**

|  |  |
| --- | --- |
| **Tên lớp: Post.** | **ID: 2.** |
| **Mô tả:** Lớp Post đại diện cho các bài đăng được tạo bởi người dùng trong hệ thống, chịu trách nhiệm quản lý nội dung, thời gian và tác giả của bài đăng. | **Use case liên quan:** |
| **Trách nhiệm:**   * **Tạo bài đăng.** * **Chỉnh sửa bài đăng.** * **Xóa bài đăng.** | **Cộng tác:**   * **Comment.** * **Notification.** * **Course** |
| **Thuộc tính:** postID: String , content: String , timestamp: Date , author: User , postType: String. | |
| **Các mối quan hệ:**  **-     Tổng quát hóa:** Quản trị viên, người dùng.  **-     Kết hợp:** Comment, Notification, Course. | |

##### **3.2.1.3 CRC Comment**.

|  |  |
| --- | --- |
| **Tên lớp: Comment.** | **ID: 3.** |
| **Mô tả:** Lớp Comment đại diện cho các bình luận mà người dùng thêm vào các bài đăng. Chịu trách nhiệm quản lý nội dung, tác giả, và thời gian của bình luận. | **Use case liên quan:** |
| **Trách nhiệm:**   * **Thêm bình luận vào bài đăng.** * **Xóa bình luận khỏi bài đăng.** | **Cộng tác:**   * **Post.** * **Notification.** |
| **Thuộc tính:** commentID: String , content: String , timestamp: Date , author: User , post: Post. | |
| **Các mối quan hệ:**  **-    Tổng quát hóa:** Quản trị viên, người dùng.  **-    Kết hợp:** Post, Notificatio. | |

##### **3.2.1.4 CRC Message**.

|  |  |
| --- | --- |
| **Tên lớp: Message.** | **ID: 4.** |
| **Mô tả:** Lớp Message đại diện cho tin nhắn trong hệ thống, cho phép người dùng hoặc admin gửi và nhận tin nhắn với nhau. | **Use case liên quan:** |
| **Trách nhiệm**   * **Gửi tin nhắn.** * **Xem tin nhắn.** * **Xóa tin nhắn.** | **Cộng tác:**   * **Group.** |
| **Thuộc tính:** messageID: String , sender: User , recipient: User , content: String , timestamp: Date. | |
| **Các mối quan hệ:**  **-     Tổng quát hóa:** Quản trị viên, người dung.  **-     Kết hợp:** Group. | |

##### **3.2.1.5 CRC Group.**

|  |  |
| --- | --- |
| **Tên lớp: Group.** | **ID: 5.** |
| **Mô tả:** Lớp Group đại diện cho các nhóm trong hệ thống, nơi người dùng có thể tham gia để trao đổi, thảo luận hoặc cộng tác. | **Use case liên quan:** |
| **Trách nhiệm**   * **Tạo nhóm.** * **Cập nhật thông tin nhóm.** * **Xóa nhóm.** * **Quản lý thành viên.** * **Xem danh sách nhóm.** | **Cộng tác:**   * **Message.** * **Course.** * **Notification.** |
| **Thuộc tính:** groupID: String , groupName: String , description: String , createdBy: User , members: List<User>. | |
| **Các mối quan hệ:**  **-    Tổng quát hóa:** Quản trị viên, người dùng.  **-    Kết hợp:** Message, Course, Notification. | |

##### **3.2.1.6 CRC Course.**

|  |  |
| --- | --- |
| **Tên lớp: Course.** | **ID: 6.** |
| **Mô tả:** Lớp Course đại diện cho các khóa học trong hệ thống, nơi cung cấp nội dung học tập, bài giảng và quản lý học viên. | **Use case liên quan:** |
| **Trách nhiệm**   * **Tạo khóa học.** * **Cập nhật khóa học.** * **Xóa khóa học.** * **Quản lý học viên.** * **Xem danh sách nhóm.** | **Cộng tác:**   * **Post.** * **Notification.** * **Subject.** * **Group.** |
| **Thuộc tính:** courseID: String , courseName: String , description: String , instructor: User , students: List<User>. | |
| **Các mối quan hệ:**  **-    Tổng quát hóa**:  Quản trị viên, người dùng.  **-    Kết hợp:** Post, Notification, Group, Subject. | |

##### **3.2.1.7 CRC Subject**.

|  |  |
| --- | --- |
| **Tên lớp: Subject.** | **ID: 7.** |
| **Mô tả:** Lớp Subject đại diện cho môn học trong hệ thống. Mỗi môn học chứa thông tin như tên, mã môn học, mô tả và có thể liên kết với các khóa học hoặc bài giảng cụ thể. | **Use case liên quan:** |
| **Trách nhiệm**   * **Tạo môn học.** * **Cập nhật môn học.** * **Xóa môn học.** * **Liên kết với khóa học.** * **Hiển thị danh sách môn học.** | **Cộng tác:**   * **Course.** |
| **Thuộc tính:** lessonID: String , lessonName: String , content: String , course: Course. | |
| **Các mối quan hệ:**  **-         Kết hợp:** Course. | |

##### **3.2.1.8 CRC Notification.**

|  |  |
| --- | --- |
| **Tên lớp: Notification.** | **ID: 8.** |
| **Mô tả:** Lớp Notification đại diện cho các thông báo trong hệ thống. Mỗi thông báo sẽ có các thuộc tính như tiêu đề, nội dung, thời gian gửi và trạng thái (đọc/chưa đọc). Lớp này sẽ giúp hệ thống gửi và quản lý thông báo cho người dùng. | **Use case liên quan:** |
| **Trách nhiệm**   * **Hiển thị danh sách thông báo.** * **Xóa thông báo.** | **Cộng tác:**   * Post. * Comment. * Group. * Course. |
| **Thuộc tính:** notificationID: String , message: String , recipient: User , status: Boolean, timestamp: Date. | |
| **Các mối quan hệ:**  **-     Tổng quát hóa:** Quản trị viên, người dùng.  **-     Kết hợp:** Post, Comment, Group, Course. | |

#### **3.2.2 PHÂN TÍCH THIẾT KẾ MỨC LOGIC.**

##### **3.2.2.1 Kiểu Dữ Liệu, Khóa Chính và Khóa Ngoại.**

###### **3.2.2.1.1 Collection: User.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính, tự động được MongoDB tạo ra nếu không cung cấp userID).
* userID: String (Có thể sử dụng làm \_id nếu cần khóa chính tùy chỉnh).
* username: String.
* password: String.
* email: String (nếu dùng).
* status: String.
* createdDate: Date.
* following: Array of String (chứa danh sách userID của những người dùng mà user đang theo dõi).

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**: Không trực tiếp, nhưng following tham chiếu đến \_id hoặc userID của User.

###### **3.2.2.1.2 Collection: Post.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* postID: String (nếu muốn tự định nghĩa ID).
* content: String.
* timestamp: Date.
* author: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* postType: String.

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**: author tham chiếu đến \_id trong User.

###### **3.2.2.1.3 Collection: Comment.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* commentID: String (nếu cần tùy chỉnh ID).
* content: String.
* author: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* timestamp: Date.
* post: ObjectId (Khóa ngoại tham chiếu đến \_id trong Post).

**Khóa:**

* **Khóa chính:** \_id.
* **Khóa ngoại:**
  + author: tham chiếu đến \_id trong User.
  + post: tham chiếu đến \_id trong Post.

###### **3.2.2.1.4 Collection: Notification.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* notificationID: String (tùy chọn nếu cần ID riêng).
* message: String.
* recipient: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* status: Boolean.
* timestamp: Date.

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**:
  + recipient: tham chiếu đến \_id trong User.

###### **3.2.2.1.5 Collection: Message.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* messageID: String (tùy chọn nếu cần ID riêng).
* sender: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* recipient: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* content: String.
* timestamp: Date.

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**:
  + sender: tham chiếu đến \_id trong User.
  + recipient: tham chiếu đến \_id trong User.

###### **3.2.2.1.6 Collection: Group.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* groupID: String (tùy chọn).
* groupName: String.
* description: String.
* createdBy: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* members: Array of ObjectId (Chứa danh sách \_id của các User trong nhóm).

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**:
  + createdBy: tham chiếu đến \_id trong User.
  + members: danh sách \_id của User.

###### **3.2.2.1.7 Collection: Course.**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* courseID: String (tùy chọn).
* courseName: String.
* description: String.
* instructor: ObjectId (Khóa ngoại tham chiếu đến \_id trong User).
* students: Array of ObjectId (Danh sách \_id của User đã đăng ký).

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**:
  + instructor: tham chiếu đến \_id trong User.
  + students: danh sách \_id của User.

###### **3.2.2.1.8 Collection: Subject (Lesson).**

**Thuộc tính:**

* \_id: ObjectId (Khóa chính).
* lessonID: String (tùy chọn).
* lessonName: String.
* content: String.
* course: ObjectId (Khóa ngoại tham chiếu đến \_id trong Course).

**Khóa:**

* **Khóa chính**: \_id.
* **Khóa ngoại**:
  + course: tham chiếu đến \_id trong Course.

##### **3.2.2.2 Các Phương Thức.**

###### **3.2.2.2.1 Collection: User.**

**Phương thức:**

**+ register()**

* **Mô tả:** Thêm một tài liệu mới vào collection users.
* **Đầu vào**: Thông tin người dùng (username, password, email, status).
* **Đầu ra**: Xác nhận người dùng đã được tạo thành công.
* **Thao tác MongoDB:**

|  |
| --- |
| db.users.insertOne({  userID: "user123",  username: "johndoe",  password: "hashed\_password",  email: "john.doe@example.com",  status: "active",  createdDate: new Date(),  following: []  }); |

**+ login()**

* **Mô tả:** Kiểm tra thông tin đăng nhập.
* **Đầu vào:** username và password.
* **Đầu ra:** Kết quả xác thực (thành công/không thành công).
* **Thao tác MongoDB:**

|  |
| --- |
| const user = db.users.findOne({  username: "johndoe",  password: "hashed\_password"  }); |

**+ updateProfile()**

* **Mô tả:** Cập nhật thông tin người dùng.
* **Đầu vào:** Các thuộc tính cần thay đổi (username, status, email, etc.).
* **Đầu ra:** Kết quả cập nhật.
* **Thao tác MongoDB:**

|  |
| --- |
| db.users.updateOne(  { userID: "user123" },  { $set: { username: "new\_username", status: "inactive" } }  ); |

**+ sendNotification()**

* **Mô tả:** Thêm thông báo vào collection notifications.
* **Đầu vào:** Nội dung thông báo, đối tượng nhận.
* **Đầu ra:** Xác nhận thông báo đã được gửi.
* **Thao tác MongoDB:**

|  |
| --- |
| db.notifications.insertOne({  message: "You have a new message!",  recipient: ObjectId("user\_id"),  status: false,  timestamp: new Date()  }); |

**+ joinGroup(group: Group)**

* **Mô tả:** Thêm userID vào danh sách members của một group.
* **Đầu vào:** Thông tin nhóm (Group ID).
* **Đầu ra:** Kết quả tham gia nhóm.
* **Thao tác MongoDB:**

|  |
| --- |
| db.groups.updateOne(  { groupID: "group123" },  { $addToSet: { members: ObjectId("user\_id") } }  ); |

###### **3.2.2.2.2 Collection: Post.**

**Phương thức:**

+ **createPost()**

* **Mô tả**: Tạo một bài đăng mới.
* **Đầu vào**: Nội dung bài viết, loại bài viết, thông tin tác giả.
* **Đầu ra**: Bài viết mới được lưu vào cơ sở dữ liệu.

|  |
| --- |
| db.posts.insertOne({  postID: "post123",  content: "This is a post",  timestamp: new Date(),  author: ObjectId("user\_id"),  postType: "text"  }); |

+ **updatePost()**

* **Mô tả**: Cập nhật nội dung bài đăng hiện có.
* **Đầu vào**: ID bài viết, nội dung mới.
* **Đầu ra**: Kết quả cập nhật.

|  |
| --- |
| db.posts.updateOne(  { postID: "post123" },  { $set: { content: "Updated content" } }  ); |

+ **deletePost()**

* **Mô tả**: Xóa bài viết khỏi cơ sở dữ liệu.
* **Đầu vào**: ID bài viết.
* **Đầu ra**: Kết quả xóa bài viết.

|  |
| --- |
| db.posts.deleteOne({ postID: "post123" }); |

###### **3.2.2.2.3 Collection: Comment.**

**Phương thức:**

+ **addComment()**

* **Mô tả**: Thêm bình luận vào một bài đăng.
* **Đầu vào**: Nội dung bình luận, thông tin bài đăng, người bình luận.
* **Đầu ra**: Bình luận được thêm vào bài đăng.

|  |
| --- |
| db.comments.insertOne({  commentID: "comment123",  content: "This is a comment",  author: ObjectId("user\_id"),  post: ObjectId("post\_id"),  timestamp: new Date()  }); |

+ **editComment()**

* **Mô tả**: Chỉnh sửa nội dung bình luận.
* **Đầu vào**: ID bình luận, nội dung mới.
* **Đầu ra**: Kết quả cập nhật bình luận.

|  |
| --- |
| db.comments.updateOne(  { commentID: "comment123" },  { $set: { content: "Updated comment" } }  ); |

+ **deleteComment()**

* **Mô tả**: Xóa bình luận khỏi bài đăng.
* **Đầu vào**: ID bình luận.
* **Đầu ra**: Kết quả xóa bình luận.

|  |
| --- |
| db.comments.deleteOne({ commentID: "comment123" }); |

###### **3.2.2.2.4 Collection: Notification.**

**Phương thức:**

+ **sendNotification()**

* **Mô tả**: Gửi thông báo tới người dùng.
* **Đầu vào**: Nội dung thông báo, người nhận.
* **Đầu ra**: Xác nhận thông báo được gửi.

|  |
| --- |
| db.notifications.insertOne({  notificationID: "notif123",  message: "You have a new notification",  recipient: ObjectId("user\_id"),  status: false,  timestamp: new Date()  }); |

+ **markAsRead()**

* **Mô tả**: Đánh dấu thông báo là đã đọc.
* **Đầu vào**: ID thông báo.
* **Đầu ra**: Trạng thái thông báo được cập nhật.

|  |
| --- |
| db.notifications.updateOne(  { notificationID: "notif123" },  { $set: { status: true } }  ); |

###### **3.2.2.2.5 Collection: Message.**

**Phương thức:**

+ **sendMessage()**

* **Mô tả**: Gửi tin nhắn từ một người dùng đến người khác.
* **Đầu vào**: Nội dung tin nhắn, người gửi, người nhận.
* **Đầu ra**: Tin nhắn được lưu và gửi thành công.

|  |
| --- |
| db.messages.insertOne({  messageID: "msg123",  sender: ObjectId("sender\_id"),  recipient: ObjectId("recipient\_id"),  content: "Hello there!",  timestamp: new Date()  }); |

+ **receiveMessage()**

* **Mô tả**: Nhận danh sách tin nhắn được gửi tới người dùng.
* **Đầu vào**: ID người nhận.
* **Đầu ra**: Danh sách tin nhắn.

|  |
| --- |
| const messages = db.messages.find({ recipient: ObjectId("recipient\_id") }); |

###### **3.2.2.2.6 Collection: Group.**

**Phương thức:**

+ **createGroup()**

* **Mô tả**: Tạo một nhóm mới.
* **Đầu vào**: Tên nhóm, mô tả nhóm, người tạo nhóm.
* **Đầu ra**: Nhóm mới được lưu vào cơ sở dữ liệu.

|  |
| --- |
| db.groups.insertOne({  groupID: "group123",  groupName: "Group Name",  description: "Group Description",  createdBy: ObjectId("user\_id"),  members: [ObjectId("user\_id")]  }); |

+ **addMember(user: User)**

* **Mô tả**: Thêm thành viên mới vào nhóm.
* **Đầu vào**: Thông tin thành viên (User ID).
* **Đầu ra**: Thành viên mới được thêm vào nhóm.

|  |
| --- |
| db.groups.updateOne(  { groupID: "group123" },  { $addToSet: { members: ObjectId("new\_user\_id") } }  ); |

+ **removeMember(user: User)**

* **Mô tả**: Xóa thành viên khỏi nhóm.
* **Đầu vào**: ID thành viên.
* **Đầu ra**: Thành viên được xóa khỏi nhóm.

|  |
| --- |
| db.groups.updateOne(  { groupID: "group123" },  { $pull: { members: ObjectId("user\_id") } }  ); |

+ **deleteGroup()**

* **Mô tả**: Xóa nhóm khỏi cơ sở dữ liệu.
* **Đầu vào**: ID nhóm.
* **Đầu ra**: Kết quả xóa nhóm.

|  |
| --- |
| db.group.deleteOne({ groupID: "group123" }); |

+ **updateGroup()**

* **Mô tả**: Cập nhật thông tin nhóm.
* **Đầu vào**: ID nhóm, thông tin cập nhật.
* **Đầu ra**: Kết quả cập nhật nhóm.

|  |
| --- |
| db.groups.updateOne(  { groupID: "group123" }, // Điều kiện tìm kiếm nhóm  {  $set: { // Cập nhật các trường  groupName: "Updated Group Name",  description: "Updated description for the group",  },  }  ); |

###### **3.2.2.2.7 Collection: Course.**

**Phương thức:**

+ **registerCourse(user: User)**

* **Mô tả**: Đăng ký một khóa học.
* **Đầu vào**: ID khóa học, thông tin người dùng.
* **Đầu ra**: Người dùng được thêm vào danh sách sinh viên của khóa học.

|  |
| --- |
| db.courses.updateOne(  { courseID: "course123" },  { $addToSet: { students: ObjectId("user\_id") } }  ); |

+ **unregisterCourse(user: User)**

* **Mô tả**: Hủy đăng ký khóa học.
* **Đầu vào**: ID khóa học, ID người dùng.
* **Đầu ra**: Người dùng được xóa khỏi danh sách sinh viên của khóa học.

|  |
| --- |
| db.courses.updateOne(  { courseID: "course123" },  { $pull: { students: ObjectId("user\_id") } }  ); |

+ **addLesson()**

* **Mô tả**: Thêm một bài học mới vào khóa học.
* **Đầu vào**: Thông tin bài học.
* **Đầu ra**: Bài học được thêm thành công.

|  |
| --- |
| db.lessons.insertOne({  lessonID: "lesson123",  lessonName: "Introduction to MongoDB",  content: "This lesson covers MongoDB basics.",  course: ObjectId("64f8b7a8e4a0f9a7e6a2e123"),  }); |

+ **updateCourse()**

* **Mô tả**: Cập nhật thông tin khóa học.
* **Đầu vào**: ID khóa học, thông tin mới.
* **Đầu ra**: Thông tin khóa học được cập nhật.

|  |
| --- |
| db.courses.updateOne(  { courseID: "course123"  {  $set: {  courseName: "Updated Course Name",  description: "Updated description for the course",  instructor: ObjectId("64fa12b8e4a0f9a7e6a2e123")  },  }  ); |

+ **deleteCourse()**

* **Mô tả**: Xóa khóa học khỏi hệ thống.
* **Đầu vào**: ID khóa học.
* **Đầu ra**: Khóa học bị xóa.

|  |
| --- |
| db.courses.deleteOne({ courseID: "course123" } ); |

###### **3.2.2.2.8 Collection: Subject.**

**Phương thức:**

+ **createLesson()**

* **Mô tả**: Tạo một bài học mới.
* **Đầu vào**: Tên bài học, nội dung, khóa học liên quan.
* **Đầu ra**: Bài học được thêm vào khóa học.

|  |
| --- |
| db.lessons.insertOne({  lessonID: "lesson123",  lessonName: "Lesson Name",  content: "Lesson Content",  course: ObjectId("course\_id")  }); |

+ **updateLesson()**

* **Mô tả**: Cập nhật nội dung bài học.
* **Đầu vào**: ID bài học, thông tin cập nhật.
* **Đầu ra**: Kết quả cập nhật bài học.

|  |
| --- |
| db.lessons.updateOne(  { lessonID: "lesson123" },  { $set: { content: "Updated Content" } }  ); |

+ **deleteLesson()**

* **Mô tả**: Xóa một bài học khỏi khóa học.
* **Đầu vào**: ID bài học.
* **Đầu ra**: Kết quả xóa bài học.

|  |
| --- |
| db.lessons.deleteOne({ lessonID: "lesson123" }); |

#### **3.2.3 PHÂN TÍCH THIẾT KẾ MỨC VẬT LÝ.**

### **3.3 PHÂN TÍCH THIẾT KẾ THÀNH PHẦN CHỨC NĂNG.**

#### **3.3.1 Activity Diagram.**

##### **3.3.1.1 Biểu đồ hoạt động đăng ký tài khoản.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.2 Biểu đồ hoạt động đăng nhập.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.3 Biểu đồ hoạt động đổi mật khẩu.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.4 Biểu đồ hoạt động quên mật khẩu.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.5 Biểu đồ hoạt động chỉnh sửa thông tin tài khoản.**

##### **3.3.1.6 Biểu đồ hoạt động xoá tài khoản.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.7 Biểu đồ hoạt động tạo bài viết.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.8 Biểu đồ hoạt động chỉnh sửa bài viết.**

##### **3.3.1.9 Biểu đồ hoạt động xoá bài viết.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.10 Biểu đồ hoạt động private-public bài viết.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.11 Biểu đồ hoạt động tạo comment.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.12 Biểu đồ hoạt động chỉnh sửa comment.**

##### **3.3.1.13 Biểu đồ hoạt động xoá comment.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.14 Biểu đồ hoạt động gửi tin nhắn.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.15 Biểu đồ hoạt động thu hồi tin nhắn 2 phía.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.16 Biểu đồ hoạt động chỉnh sửa tin nhắn.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.17 Biểu đồ hoạt động theo dõi bạn bè.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.18 Biểu đồ hoạt động theo dõi lại.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.19 Biểu đồ hoạt động bỏ theo dõi.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.20 Biểu đồ hoạt động tạo Group.**

A diagram of a group

Description automatically generated

##### **3.3.1.21 Biểu đồ hoạt động chỉnh sửa thông tin Group.**

##### **3.3.1.22 Biểu đồ hoạt động xoá Group.**

A diagram of a group

Description automatically generated

##### **3.3.1.23 Biểu đồ hoạt động join Group.**

##### **3.3.1.24 Biểu đồ hoạt động out Group.**

A diagram of a group

Description automatically generated

##### **3.3.1.25 Biểu đồ hoạt động thêm thành viên vào Group.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.26 Biểu đồ hoạt động xoá thành viên khỏi Group.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.27 Biểu đồ hoạt động video call.**

##### **3.3.1.28 Biểu đồ hoạt động tạo khoá học.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.29 Biểu đồ hoạt động chỉnh sửa thông tin khoá học.**

##### **3.3.1.30 Biểu đồ hoạt động xoá khoá học.**

##### **3.3.1.31 Biểu đồ hoạt động thêm môn học.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.32 Biểu đồ hoạt động huỷ môn học.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1.33 Biểu đồ hoạt động đăng ký khoá học.**

A diagram of a flowchart

Description automatically generated

##### **3.3.1. 34 Biểu đồ hoạt động huỷ đăng ký khoá học.**

A diagram of a flowchart

Description automatically generated

#### **3.3.2 Use Case Diagram.**

#### **3.3.3 Sequence Diagram.**

## **Chương 4: CÀI ĐẶT MẠNG XÃ HỘI**

## **Chương 5: TRIỂN KHAI MẠNG XÃ HỘI**

**5.1 Triển khai (Deployment).  
5.1.1 Chuẩn bị môi trường triển khai.  
5.1.2 Triển khai mã nguồn.  
5.1.3 Tối ưu hóa và bảo mật.  
5.2 Thử nghiệm (Testing).  
5.2.1 Kiểm thử chức năng (Functional Testing).  
5.2.2 Kiểm thử hiệu năng (Performance Testing).  
5.2.3 Kiểm thử bảo mật (Security Testing).  
5.2.4 Kiểm thử hồi quy (Regression Testing).  
5.2.5 Kiểm thử người dùng (User Acceptance Testing - UAT).  
5.3 Hoàn tất và phát hành.**

**5.4 Thu thập ý kiến và đánh giá  
5.4 Kết luận.**

## **Chương 6: KẾT LUẬN**

# **Phần TÀI LIỆU THAM KHẢO**