*Mẫu 1:* *Quy cách trình bày đồ án/khoá luận tốt nghiệp*

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

(bìa 1, bìa cứng, chữ nhũ)



**ĐỒ ÁN/KHÓA LUẬN TỐT NGHIỆP**

**TÊN ĐỀ TÀI:**

**Giảng viên hướng dẫn:**

**Sinh viên thực hiện:**

**Ngày sinh:**

**Lớp:**

**Ngành:**

**Khoa:**

**Khóa:**

**Mã sinh viên:**

**Bắc Ninh, năm 20…**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**(**trang phụ bìa số 2, bìa mềm)

**NGUYỄN VĂN A**

**TÊN ĐỀ TÀI: …………………….**

**Giảng viên hướng dẫn: TS…….**

**Bắc Ninh, năm 20…**

**File gáy**

|  |  |  |
| --- | --- | --- |
| **Năm 20…** | **NGUYỄN VĂN A** | **Khoa:……..** |

**LỜI CAM ĐOAN**

**(Trình bày trong trang riêng)**

Tôi xin cam đoan rằng đồ án/khoá luận tốt nghiệp với đề tài… là nghiên cứu độc lập của tôi. Đồng thời những số liệu được cung cấp từ báo cáo đều là kết quả nghiên cứu hoàn toàn trung thực, không sao chép từ bất kì một công trình nghiên cứu khác nào.

Bắc Ninh, ngày …. Tháng … năm ….

**Sinh viên**

*(ký và ghi rõ họ tên)*

**(ký tay, không ký bằng mực đen )**

**LỜI CẢM ƠN**

**(Trình bày trong trang riêng)**

**MỤC LỤC**

**(Trình bày trong trang riêng)**

**DANH MỤC CÁC TỪ VIẾT TẮT**

**(trình bày trong trang riêng)**

|  |  |  |
| --- | --- | --- |
| STT | Chữ viết tắt | Giải thích |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

**DANH MỤC BẢNG BIỂU VÀ SƠ ĐỒ**

**(trình bày trong trang riêng)**

|  |  |  |
| --- | --- | --- |
| Số hiệu | Tên | Trang |
| 1.1 |  |  |
|  |  |  |

***Lưu ý***

* Các sơ đồ, hình vẽ, bảng biểu phải có tên và số thứ tự được sắp xếp theo chương.
* Đối với sơ đồ, hình vẽ, đồ thị thì tên được đặt ở dưới
* Đối với bảng số liệu thì tên đặt ở trên.

# CHƯƠNG 1: GIỚI THIỆU

## Lý do chọn đề tài

Trong thời đại số hóa và sự phát triển nhanh chóng của công nghệ, việc chia sẻ kiến thức và kinh nghiệm là một phần không thể thiếu trong hành trình học tập và nghề nghiệp của sinh viên ngành công nghệ thông tin. Tuy nhiên, việc tìm kiếm và trao đổi thông tin trên các nền tàng hiện tại có thể gặp phải nhiều khó khăn, từ việc phân loại thông tin đến sự thiếu hiểu biết về nguồn gốc và độ tin cậy của thông tin.

Để giải quyết vấn đề này, dự án "Xây dựng mạng xã hội chia sẻ kiến thức cho sinh viên ngành công nghệ thông tin" đã ra đời nhằm tạo ra một nền tàng tương tác, trong đó sinh viên có thể chia sẻ, trao đổi và học hỏi từ nhau một cách dễ dàng và hiệu quả. Mạng xã hội này không chỉ là nơi để sinh viên tìm kiếm thông tin và tư vấn từ cộng đồng, mà còn là một không gian thú vị để thảo luận về các vấn đề công nghệ mới nhất, chia sẻ dự án và ý tưởng, cũng như xây dựng mối quan hệ với những người đồng nghiệp và chuyên gia trong ngành.

Hy vọng rằng dự án này sẽ không chỉ tạo ra một cộng đồng mạng mạnh mẽ và sôi nổi, mà còn góp phần vào việc nâng cao chất lượng học tập và sự phát triển cá nhân của sinh viên ngành công nghệ thông tin. Đồng thời, dự án cũng mang trong mình mục tiêu thúc đẩy sự chia se và hợp tác trong cộng đồng người học và những người làm việc trong lĩnh vực này, từ đó góp phần vào sự tiến bộ của ngành công nghiệp công nghệ thông tin trong tương lai.

## Mục tiêu của đề tài

* Áp dụng phương pháp phân tích và thiết kế hệ thống thông tin cho bài toán cụ thể và thực tế.
* Áp dụng các công nghệ lập trình như : MySQL, react.js, Java Spring Boot để xây dụng mạng xã hội chia sẻ kiến thức cho sinh viên ngành công nghệ thông tin, đáp ứng được nhu cầu thực tế, đảm bảo mọi chức năng hoạt động tốt, mang lại hiệu quả trong trải nghiệm người dùng.

## Giới hạn và phạm vi của đề tài

Trong phạm vi đề tài này em sẽ nghiên cứu các vấn đề:

* Tìm hiểu, khảo sát thực tế nhu cầu của sinh viên khi sử dụng mạng xã hội.
* Phân tích thiết kế chi tiết hệ thống các chức năng và yêu cầu của mạng xã hội chia sẻ kiến thức cho sinh CNTT. Thiết kế cấu trúc và chức năng của mạng xã hội, bao gồm các tính năng như tạo và quản lý tài khoản cá nhân, chia sẻ bài viết, thảo luận, đánh giá và phản hồi.
* Nghiên cứu và vận dụng các công cụ (ngôn ngữ lập trình Java Spring Boot, React.js, MySQL) để triển khai xây dựng hệ thống bao gồm truy xuất dữ liệu và quản lý người dùng.

## Dự kiến kết quả đạt được

* Vận dụng các phương pháp phân tích và thiết kế hệ thống thông tin cho bài toán cụ thể.
* Hoàn thành xây dựng mạng xã hội chia sẻ kiến thức cho sinh viên ngành công nghệ thông tin, gồm các chức năng chính sau:
  + **Đăng ký, Đăng nhập:** 
    - Người dùng có thể tạo tài khoản mới và đăng nhập bằng email, số điện thoại hoặc tài khoản mạng xã hội khác.
    - Phải có xác thực và bảo mật đảm bảo tính bảo mật cho người dùng
  + **Hồ sơ cá nhân:**
    - Người dùng có thể tạo và quản lý hồ sơ cá nhân của họ.
    - Cung cấp khả năng tải lên ảnh đại diện, thêm mô tả, và cập nhật thông tin cá nhân.
  + **Bài viết và Hoạt động:**
    - Người dùng có thể tạo, đăng và chia sẻ bài viết, hình ảnh, video.
    - Hỗ trợ tính năng thích, bình luận và chia sẻ bài viết của người dùng khác.
    - Hiển thị hoạt động mới nhất của bạn bè và nhóm mà người dùng tham gia.
  + **Kết nối và Quản lý bạn bè:**
    - Người dùng có thể kết bạn với nhau và quản lý danh sách bạn bè.
    - Cho phép tìm kiếm và gợi ý kết bạn dựa trên sở thích và thông tin cá nhân.
  + **Messenger và Trò chuyện:**
    - Cung cấp tính năng gửi tin nhắn trực tuyến và nói chuyện với bạn bè.
    - Hỗ trợ tạo nhóm trò chuyện và chia sẻ hình ảnh, video, tập tin.
  + **Thông báo và Tương tác Realtime:**
    - Hiển thị thông báo realtime về các hoạt động của bạn bè và nhóm mà người dùng tham gia.
    - Cập nhật realtime cho các tương tác như bình luận, thích, tin nhắn.

## Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau:

* **Chương 2:** Trình bày về lý thuyết cũng như các công cụ sẽ hỗ trợ trong việc phát triển dự án. Chương này sẽ đi qua lý thuyết về ngôn ngữ lập trình, framework, CSDL sử dụng.
* **Chương 3:** Từ những chức năng chính bên trên bắt đầu phân tích sâu hơn từ việc khảo sát, đặc tả nghiệp vụ. Sau đó phân tích chi tiết hệ thống (các thực thể) rồi đến thiết kế hệ thống (các mô hình).
* **Chương 4:** Từ việc phân tích, sẽ nêu lên từng quá trình trong xây dựng từ gốc rễ của dự án. Cơ sở dữ liệu cho đến website.

# CHƯƠNG 2: KIẾN THỨC NỀN TẢNG

## Cơ sở lý thuyết

### 2.2.1. Phân tích thiết kế hệ thống

Phân tích và thiết kế hệ thống thông tin (SAD) là một quá trình có hệ thống nhằm xác định nhu cầu của người dùng, đưa ra giải pháp và xây dựng hệ thống thông tin đáp ứng nhu cầu đó. SAD là một lĩnh vực quan trọng trong ngành công nghệ thông tin, đóng vai trò thiết yếu trong việc tạo ra các hệ thống thông tin hiệu quả, hữu ích cho tổ chức.

***2.1.1.1******Giai đoạn, công cụ hỗ trợ, kỹ năng cần thiết trong quy trình phân tích thiết kế hệ thống thông tin***

* **Các giai đoạn trong quy trình SAD được chia thành các giai đoạn sau:**
* ***Phân tích yêu cầu:*** Xác định nhu cầu của người dùng và các chức năng cần thiết của hệ thống.
* ***Phân tích hệ thống hiện tại:*** Xác định cách thức hoạt động của hệ thống hiện tại, những điểm mạnh và điểm yếu của hệ thống.
* ***Thiết kế hệ thống mới:*** Thiết kế các thành phần chính của hệ thống mới, bao gồm phần mềm, phần cứng, mạng lưới và cơ sở dữ liệu.
* ***Triển khai hệ thống:*** Lắp đặt và cấu hình phần mềm, phần cứng, mạng lưới và cơ sở dữ liệu.
* ***Kiểm thử hệ thống:*** Kiểm tra hệ thống để đảm bảo đáp ứng các yêu cầu và không có lỗi.
* ***Bảo trì hệ thống:*** Cập nhật và bảo trì hệ thống sau khi triển khai.
* **Các công cụ hỗ trợ phân tích và thiết kế hệ thống thông tin**
* ***Công cụ CASE (Computer-Aided Software Engineering):*** Hỗ trợ các hoạt động phân tích, thiết kế và triển khai hệ thống.
* ***Công cụ mô hình hóa dữ liệu:*** Hỗ trợ tạo và chỉnh sửa các biểu đồ ERD.
* ***Công cụ thiết kế giao diện người dùng:*** Hỗ trợ thiết kế giao diện người dùng của hệ thống.
* **Kỹ năng cần thiết khi phân tích và thiết kế hệ thống thông tin**
* ***Kỹ năng phân tích:*** Khả năng thu thập và phân tích thông tin, xác định nhu cầu của người dùng và đưa ra giải pháp.
* ***Kỹ năng thiết kế:*** Khả năng thiết kế các thành phần chính của hệ thống, bao gồm phần mềm, phần cứng, mạng lưới và cơ sở dữ liệu.
* ***Kỹ năng kỹ thuật:*** Kiến thức về ngôn ngữ lập trình, CSDL và mạng lưới.
* ***Kỹ năng giải quyết vấn đề:*** Khả năng xác định, giải quyết vấn đề phức tạp.

***2.1.1.2. Phân tích thiết kế với ký pháp UML***

UML (Unified Modelling Language) là ngôn ngữ mô hình hoá tổng quát. Được xây dựng để đặc tả, phát triển và viết tài liệu cho các khía cạnh trong phát triển phần mềm hướng đối tượng:

* Giúp người phát triển hiểu rõ và ra quyết định liên quan đến phần mềm cần xây dựng.
* Bao gồm một tập các khái niệm, các ký hiệu, các biểu đồ và hướng dẫn.
* ***UML là ký pháp chứ không phải là phương pháp.***

UML có thể áp dụng cho tất cả các pha của quy trình phát triển phần mềm

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Nhiều màu sắc

Nội dung do AI tạo ra có thể không chính xác.

UML cung cấp 5 góc nhìn khác nhau nhằm đặc tả một hệ thống:

* **Use Case View:** Là góc nhìn phía bên ngoài hệ thống. Mô tả các chức năng lớn và các quy trình của hệ thống và là góc nhìn của người dùng cuối, người phân tích, người kiểm định.
* **Logical View:** Là góc nhìn phía bên trong hệ thống. Mô tả cấu trúc tĩnh (lớp, đối tượng, quan hệ) và cấu trúc động (chuyển trạng thái) của các thành phần trong hệ thống và là góc nhìn chủ yếu của người thiết kế, người phát triển.
* **Process View:** Là góc nhìn cho thấy các quá trình hoạt động, trình tự hoạt động (song hành hay đồng bộ) trong hệ thống. Là góc nhìn chủ yếu của người thiết kế, người phát triển.
* **Implementation View:** Là góc nhìn cho thấy các thành phần (packages, files, modules,..) được lắp ráp và tương tác với nhau để tạo ra hệ thống. Là góc nhìn chủ yếu của người thiết kế, người phát triển.
* **Deployment View:** Là góc nhìn cho thấy mô hình phần cứng mà trên đó hệ thống được thực hiện. Chỉ rõ sự phân bố, sự sắp đặt các phần của hệ thống vật lý trên các đơn vị phần cứng (servers, network) và là góc nhìn chủ yếu của người thiết kế, người triển khai.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, vòng tròn

Nội dung do AI tạo ra có thể không chính xác.

Các biểu đồ của UML:

* **Biểu đồ cấu trúc:**
  + Package diagram (biểu đồ gói)
  + Class diagram (biểu đồ lớp)
  + Object diagram (biểu đồ đối tượng)
  + Component diagram (biểu đồ thành phần)
  + Deployment diagram (biểu đồ triển khai)
* **Biểu đồ hành vi:**
  + Activity diagram (biểu đồ hoạt động)
  + Use Case diagram (biểu đồ trường hợp sử dụng)
  + Sequence diagram (biểu đồ tuần tự)
  + State machine diagram (biểu đồ trạng thái)
  + Communication/collaboration diagram (biểu đồ cộng tác)

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

### 2.2.2 Ngôn ngữ lập trình JavaScript và React

* **Cú pháp JavaScript:** Hiểu cú pháp cơ bản của JavaScript là điều quan trọng để có thể hiểu và sử dụng các framework. Điều này bao gồm cách khai báo biến, tạo hàm, sử dụng điều kiện và vòng lặp, xử lý sự kiện, và làm việc với mảng và đối tượng.
* **Xử lý bất đồng bộ (Asynchronous Programming):** JavaScript là một ngôn ngữ lập trình không đồng bộ (asynchronous), điều này có nghĩa là nó có khả năng thực hiện nhiều công việc cùng một lúc mà không chờ đợi kết quả của công việc trước. Để làm việc hiệu quả với các framework như Next.js và React.js, cần hiểu về xử lý bất đồng bộ, bao gồm sử dụng callbacks, promises, async/await.
* **DOM Manipulation (Document Object Model):** DOM là cách mà các trang web và ứng dụng web được biểu diễn và tương tác trong JavaScript. Hiểu cách làm việc với DOM là quan trọng để có thể tạo và điều khiển các phần tử trên trang web hoặc ứng dụng.
* **Closures và Scoping:** Closures là một khái niệm quan trọng trong JavaScript, cho phép truy cập biến từ một phạm vi khác trong mã của mình. Scoping cũng là một khái niệm quan trọng, xác định phạm vi của biến trong mã.
* **ES6 và Các tính năng mới:** ES6 (ECMAScript 2015) là phiên bản quan trọng của JavaScript, giới thiệu nhiều tính năng mới như let và const, arrow functions, classes, template literals, và nhiều tính năng khác. Hiểu và sử dụng các tính năng này sẽ giúp viết mã JavaScript hiệu quả hơn.
* **Các khái niệm cơ bản về React:** Hiểu cách React hoạt động là quan trọng để sử dụng chúng hiệu quả. Điều này bao gồm việc hiểu về components, props, state, routing, SSR (Server-Side Rendering), và các khái niệm khác liên quan đến việc phát triển ứng dụng web.

### 2.2.3 Cơ sở dữ liệu MySQL

**MySQL** là một hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến được sử dụng rộng rãi trên toàn thế giới. Nó cung cấp một môi trường lưu trữ dữ liệu hiệu quả, linh hoạt và mạnh mẽ.

* **Cấu trúc dữ liệu:** Trong MySQL, dữ liệu được tổ chức trong các bảng (tables). Mỗi bảng có các cột (columns) đại diện cho các trường dữ liệu khác nhau, và các hàng (rows) đại diện cho các bản ghi riêng lẻ.
* **Ngôn ngữ truy vấn SQL:** SQL (Structured Query Language) là ngôn ngữ được sử dụng để tương tác với cơ sở dữ liệu. Trong MySQL, sử dụng SQL để thực hiện các thao tác như truy vấn dữ liệu, chèn dữ liệu, cập nhật dữ liệu và xóa dữ liệu.
* **Bảo mật và phân quyền:** MySQL cung cấp các cơ chế bảo mật để bảo vệ dữ liệu khỏi truy cập trái phép. Có thể thiết lập các quyền truy cập cho người dùng và nhóm người dùng khác nhau, giúp kiểm soát quyền truy cập vào các bảng và dữ liệu.
* **Tối ưu hóa và hiệu suất:** MySQL cung cấp các công cụ và kỹ thuật tối ưu hóa để tăng cường hiệu suất của cơ sở dữ liệu. Điều này bao gồm việc sử dụng chỉ số, tối ưu hóa truy vấn, và cấu hình hệ thống phù hợp.

## Công cụ sử dụng

### 2.2.1. Visual Studio Code (VS Code):

Ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Có khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng.

* + ***Đa nền tảng:*** VS Code có sẵn cho Windows, macOS và Linux, giúp đảm bảo một trải nghiệm nhất quán trên các hệ điều hành khác nhau.
  + ***Giao diện sáng tạo:*** Nó có giao diện người dùng sạch sẽ và dễ sử dụng, với các tính năng như gợi ý mã, định dạng mã tự động và ghi chú thông minh.
  + ***Hỗ trợ nhiều ngôn ngữ:*** VS Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến như JavaScript, Python, Java, C++, và nhiều ngôn ngữ khác nữa thông qua các tiện ích mở rộng.
  + ***Tiện ích mở rộng phong phú:*** Cộng đồng phát triển tiện ích mở rộng của VS Code rất lớn, cho phép tùy chỉnh và mở rộng khả năng của trình soạn thảo theo nhu cầu cụ thể của mình.
  + ***Hỗ trợ git nâng cao:*** VS Code tích hợp sâu với Git, cho phép quản lý mã nguồn, thực hiện các thao tác nhánh và hợp nhất mã một cách dễ dàng.
  + ***Tích hợp terminal***: Có thể mở một cửa sổ terminal trực tiếp trong VS Code, giúp tăng cường quản lý dự án và thực hiện các lệnh hệ thống một cách thuận tiện.
  + ***Tích hợp gỡ lỗi mạnh mẽ:*** VS Code đi kèm với công cụ gỡ lỗi tích hợp mạnh mẽ cho nhiều ngôn ngữ, giúp tìm và sửa lỗi mã một cách hiệu quả.
  + ***Tích hợp đám mây***: Có thể kết nối với các dịch vụ đám mây như Azure, AWS và Google Cloud để phát triển và triển khai ứng dụng trực tiếp từ VS Code.

### 2.2.2. Postman

Ứng dụng giúp các nhà phát triển xây dựng, thử nghiệm và tương tác với các API.

* + ***Xây Dựng và Kiểm Tra API:*** Postman cho phép tạo ra các yêu cầu HTTP đến các API khác nhau và kiểm tra phản hồi từ chúng. Điều này giúp trong việc phát triển, kiểm thử và xác minh tính đúng đắn của các API.
  + ***Giao diện dễ sử dụng:*** Giao diện của Postman thân thiện và dễ sử dụng, cho phép dễ dàng tạo mới, chỉnh sửa và tổ chức các yêu cầu và bộ kiểm tra.
  + ***Quản Lý Biến và Môi Trường:*** Postman cho phép quản lý các biến và môi trường, giúp tự động hóa các yêu cầu và thử nghiệm cho nhiều môi trường khác nhau như phát triển, kiểm thử và sản xuất.
  + ***Kiểm tra nâng cao:*** Có thể tạo các bộ kiểm tra tự động trong Postman để kiểm tra tính đúng đắn của các API. Các bộ kiểm tra này có thể được tùy chỉnh linh hoạt và chạy tự động.
  + ***Tương Tác với Đám Mây:*** Postman tích hợp với các dịch vụ đám mây như Postman Cloud để chia sẻ và quản lý các bộ kiểm tra và môi trường làm việc.
  + ***Tích Hợp Git và Tài Nguyên Mở:*** Có thể lưu trữ các bộ kiểm tra và môi trường trong kho lưu trữ Git và chia sẻ chúng với đồng nghiệp hoặc cộng đồng bằng cách sử dụng các bộ kiểm tra công cộng và các mẫu.

### 2.2.3. IntelliJ

Ứng dụng giúp các nhà phát triển xây dựng server và API trao đổi trực tiếp với CSDL

* **Refactoring và Tái Cấu Trúc Code**: IntelliJ cung cấp nhiều công cụ hỗ trợ refactoring như đổi tên biến, tách phương thức, hợp nhất class, … Điều này giúp duy trì mã nguồn luôn sạch sẽ và dễ bảo trì ngay cả khi dự án lớn trở nên phức tạp.
* **Hỗ Trợ Plugin Dành Riêng Cho Spring**: IntelliJ có các plugin được xây dựng chuyên biệt cho Spring và Spring Boot, cho phép tự động nhận diện cấu hình, bean, và dependency injection. Tính năng này giúp giảm thiểu thời gian cấu hình thủ công và tăng tính nhất quán của dự án.
* **Hỗ Trợ Maven/Gradle**: IntelliJ tích hợp sẵn các công cụ quản lý dự án như Maven và Gradle, giúp tự động hóa việc quản lý dependency và xây dựng (build) dự án. Điều này giúp giảm thời gian cấu hình và cập nhật dự án.
* **Debugger và Hỗ Trợ Test Nâng Cao:** Việc debug trực tiếp trong IntelliJ với giao diện trực quan giúp bạn theo dõi các luồng xử lý, kiểm tra biến và điểm dừng (breakpoints) một cách dễ dàng. Điều này rất hữu ích trong việc phát hiện và sửa lỗi trong ứng dụng Spring Boot.
* **Tích Hợp với Các Công Cụ và Framework Khác:** Hỗ Trợ Docker và Kubernetes: Nếu bạn triển khai ứng dụng Spring Boot qua container hay chạy trên nền tảng cloud, IntelliJ hỗ trợ tích hợp với Docker và các công cụ quản lý container khác.
* **Giao Diện Người Dùng Thân Thiện và Tùy Biến Cao**: Với giao diện trực quan, dễ sử dụng và khả năng tùy biến cao, IntelliJ mang đến một môi trường làm việc hiệu quả, giúp bạn tập trung vào việc viết và tối ưu hóa code.

## Công nghệ sử dụng Socket.IO và triển khai ứng dụng

Khi truy cập vào một trang Web hoặc một ứng dụng bất kỳ thì việc giao tiếp giữa Client với Server, cũng như việc kết nối giữa các Client với nhau là việc rất quan trọng. Để Client và Server nhận biết được sự thanh đổi của đối phương thì cần sử dụng cách giao tiếp như AJAX, long-polling, short-polling, … Việc giao tiếp bằng công cụ kể trên tồn tại rất nhiều nhược điểm khi Server hoặc Client luôn tục thay đổi khi người dùng sử dụng.

Để khắc phục điều ấy, Socket.io ra đời. Mục đích lớn nhất của Socket.io chính là để tạo môi trường giao tiếp thuận tiện giúp trả về các giá trị thực một cách ngay tức khắc tại một thời điểm giao tiếp giữa các bên với nhau. Đơn giản như một ứng dụng nhắn tin thông thường, nếu không có Socket.io thì phía các Client khác nhau sẽ phải liên tục gửi đến Server các yêu cầu về việc Client đối diện có sự thay đổi hay không, điều này rất dễ dẫn đến việc quá tải khi mà Server sẽ không chỉ trả lời cho 2 Client tham gia vào phòng chat. Khi Socket.io ra đời, các ứng dụng nhắn tin trở nên đơn giản hơn nhiều: khi một Client có sự thay đổi, tức là gửi một tin nhắn đến Client khác, server sẽ nhận biết được và sau đó báo cho Client kia biết, 2 bên sẽ trao đổi với nhau thông qua sự thay đổi của Client 1, vì vậy sẽ không phải cần yêu cầu Server xem lại thay đổi nữa.

Trong ứng dụng mạng xã hội, khi mà việc kết nối trực tuyến giữa các người dùng với nhau được đặt lên hàng đầu, thì Socket.io chính là điều quan trọng nhất dẫn tới thành công của việc kết nối trực tuyến. Cơ bản phải kể đến các chức năng như nhắn tin đã nói phía trên, chức năng nhận thông báo trực tuyến.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Khảo sát hệ thống

### 3.1.1 Giới thiệu chung

**Mô tả dự án:**

Dự án mạng xã hội được thiết kế nhằm cung cấp một nền tảng kết nối, chia sẻ thông tin và tương tác giữa người dùng. Hệ thống bao gồm giao diện người dùng thân thiện, đa chức năng, xây dựng trên nền tảng web với công nghệ React cho phía trình duyệt và Spring Boot cho phía máy chủ.

**Mục tiêu:**

Xây dựng hệ thống có khả năng mở rộng và dễ bảo trì.

Đảm bảo hiệu suất, bảo mật và trải nghiệm người dùng tối ưu.

Hỗ trợ các chức năng tương tác như đăng bài, bình luận, kết bạn, nhắn tin trực tiếp, và thông báo.

### 3.1.2. Kiến trúc hệ thống

**Phần Frontend (React):**

Thư viện & Framework: Sử dụng React kết hợp với Redux (hoặc Context API) để quản lý trạng thái, React Router để định tuyến.

Tính năng:

Tương tác người dùng mượt mà, phản hồi thời gian thực thông qua WebSocket (nếu cần).

Thiết kế responsive hỗ trợ đa thiết bị.

Tích hợp các component UI hiện đại (sử dụng Ant Design, Material-UI hoặc Bootstrap tùy vào lựa chọn của đồ án).

Phần Backend (Spring Boot):

**API RESTful:** Cung cấp các endpoint cho các tính năng chính của hệ thống (xác thực, quản lý người dùng, bài đăng, tương tác, thông báo, …).

Xử lý nghiệp vụ: Tách biệt rõ ràng giữa các tầng (controller, service, repository) nhằm đảm bảo tính module hóa, dễ dàng bảo trì và mở rộng.

Bảo mật: Áp dụng các biện pháp bảo mật như JWT (JSON Web Token) cho việc xác thực và phân quyền truy cập.

Giao tiếp: Hỗ trợ giao tiếp với các dịch vụ ngoài (ví dụ: dịch vụ gửi email, dịch vụ lưu trữ file ảnh/video).

**Cơ sở Dữ liệu:**

SQL/NoSQL: Sử dụng cơ sở dữ liệu SQL (MySQL, PostgreSQL) cho việc lưu trữ dữ liệu quan hệ như thông tin người dùng, bài đăng, kết nối bạn bè, cũng như có thể kết hợp NoSQL (MongoDB, Redis) để tối ưu hiệu năng cho các thao tác dữ liệu không quan hệ, cache.

**Tích hợp ORM**: Sử dụng JPA/Hibernate để thao tác dữ liệu từ Spring Boot.

Các thành phần phụ trợ:

**Server & Deployment:** Triển khai hệ thống trên Cloud (AWS, Azure, Google Cloud) hoặc các giải pháp container hóa (Docker, Kubernetes) nhằm tối ưu khả năng mở rộng và quản lý hạ tầng.

Các dịch vụ tích hợp: Hỗ trợ tính năng upload file, xử lý ảnh (sử dụng các dịch vụ xử lý ảnh từ bên thứ ba nếu cần), và thông báo đẩy tới thiết bị di động hoặc trình duyệt.

## Đặc tả nghiệp vụ

### 3.2.1 Mục tiêu nghiệp vụ

Cung cấp nền tảng giao lưu, chia sẻ thông tin giữa các thành viên nhằm tạo nên một cộng đồng kết nối.

Cho phép người dùng đăng bài, chia sẻ hình ảnh, video và tương tác thông qua các hoạt động như like, comment và share.

Hỗ trợ tính năng nhắn tin (chat trực tiếp) và thông báo các sự kiện, cập nhật mới.

### 3.2.2 Phân tích các yêu cầu chức năng

*3.2.2.1. Quản Lý Người Dùng*

* Đăng ký & Đăng nhập:

Người dùng mới có thể đăng ký tài khoản thông qua username.

Hệ thống sử dụng xác thực bảo mật ( JWT).

* Quản lý thông tin cá nhân:

Cho phép người dùng cập nhật thông tin, hình ảnh đại diện, và thiết lập quyền riêng tư.

Chức năng khôi phục mật khẩu thông qua email.

*3.2.2.2. Quản Lý Nội Dung*

* Bài đăng và Feed tin tức:

Cho phép người dùng đăng văn bản, hình ảnh, video và chia sẻ cảm xúc.

Hệ thống hiển thị Feed theo thuật toán ưu tiên tương tác (mối quan hệ, thời gian đăng, tương tác).

* Tương tác với bài đăng:

Người dùng có thể like, comment bài đăng.

Giao diện phản hồi thời gian thực khi có tương tác mới.

* Quản lý bài đăng:

Cho phép người dùng chỉnh sửa hoặc xóa bài đăng của mình.

Hệ thống báo cáo nội dung vi phạm và xử lý dựa trên chính sách cộng đồng.

## Phân tích hệ thống

### 3.3.1. Biểu đồ phân rã chức năng (BFD)

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 1 Biểu đồ Phân rã chức năng

### 3.3.2 Tác nhân

Bảng số liệu 1 Các tác nhân và mục tiêu của các tác nhân

|  |  |
| --- | --- |
| **Tác nhân** | **Mục tiêu** |
| Người dùng | Đăng ký tài khoản  Đăng nhập/Đăng xuất  Quản lý thông tin cá nhân  Quản lý mật khẩu  Nhắn tin cá nhân  Thông báo  Bình luận và phản hồi  Thích nội dung  Đăng bài viết  Chỉnh sửa/Xóa bài viết  Tìm kiếm người dùng, bài viết |

### 3.3.3. Biểu đồ use-case và đặc tả

* **Chức năng quản lý người dùng**

Ảnh có chứa văn bản, biểu đồ, vòng tròn, hàng

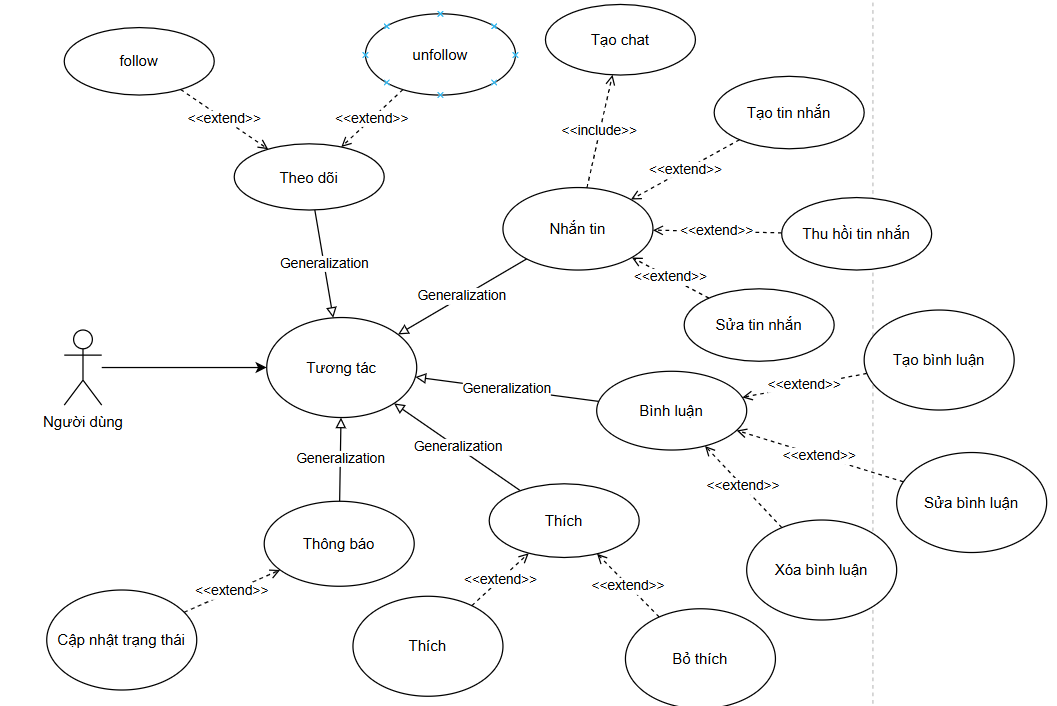
Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 2 Usecase quản lý người dùng

Bảng số liệu 2 Đặc tả usecase quản lý người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên usecase** | | | *Đăng nhập tài khoản* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Không có | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Nhập thông tin tài khoản ( username) và password |
| 2 | Hệ thống | Hệ thống sẽ xác thực thông tin người dùng. |
| 3 | Hệ thống | Xác thực thành công sẽ chuyển người dùng vào trang chính |
| **Tên usecase** | | | *Đăng xuất tài khoản* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Nhấn vào biểu tưởng đăng xuất ở phần Footer icon của mình bên trái. |
| 2 | Hệ thống | Sẽ xóa thông tin token đăng nhập của người dùng đi. |
| 3 | Hệ thống | Chuyển người dùng đến trang sign-in. |
| **Tên usecase** | | | *Đăng ký tài khoản* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Không có | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Vào trang sign-up. Chọn phương thức tạo tài khoản và điền thông tin theo mẫu. |
| 2 | Hệ thống | Tạo và lưu trữ thông tin người dùng vào CSDL |
| 3 | Hệ thống | Chuyển người dùng đến trang chính |
| **Tên usecase** | | | *Quên mật khẩu* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Không có | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Tại trang sign-in, người dùng chọn vào đường dẫn quên mật khẩu |
| 2 | Hệ thống | Hiển thị form nhập thông tin |
| 3 | Người dùng | Nhập thông tin email để nhận mã OTP |
| 4 | Hệ thống | Gửi mã OTP về email người dùng nhận và chuyển người dùng đến trang nhập mã OTP |
| 5 | Người dùng | Nhập mã OTP gồm 6 chữ số được gửi trong email |
| 6 | Hệ thống | Xác thực mã OTP và chuyển người dùng đến trang đặt lại mật khẩu |
| 7 | Người dùng | Nhập mật khẩu mới và nhập lại mật khẩu mới 1 lần nữa |
| 8 | Hệ thống | Xác thực thông tin và cập nhật lại mật khẩu vào trong CSDL |
| **Tên usecase** | | | *Quản lý thông tin cá nhân* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Vào trang cá nhân. Gửi yêu cầu cập nhật thông tin |
| 2 | Hệ thống | Hiển thị form nhập thông tin |
| 3 | Người dùng | Nhập thông tin và xác nhận |
| 4 | Hệ thống | Xác thực thông tin và cập nhật lại thông tin cho người dùng |
| 5 | Hệ thống | Hiển thị thông báo cho người dùng |

* **Chức năng Tương tác**



Hình 3. 3 Usecase Tương tác

Bảng số liệu 3 Đặc tả Usecase Tương tác

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên usecase** | | | *Theo dõi* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng (1) | Gửi yêu cầu Theo dõi đến người dùng (2) |
| 2 | Hệ thống | Thêm người dùng (2) vào danh sách đang theo dõi của người dùng (1) và thêm người dùng (1) vào danh sách theo người theo dõi của người dùng (2) |
|  | 3 | Hệ thống | Gửi thông báo đến người dùng (2) |
| **Tên usecase** | | | *Nhắn tin* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng (1) | Vào trang nhắn tin và gửi tin nhắn cho người dùng (2) |
| 2 | Hệ thống | Tạo và lưu trữ tin nhắn vào CSDL. Gửi thông báo đến người dùng (2) |
| 3 | Người dùng (2) | Đọc thông báo và phản hồi |
| **Mã use case** | *UC22* | **Tên usecase** | *Thông báo* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Mở danh mục thông báo |
| 2 | Hệ thống | Lấy danh sách thông báo từ CSDL |
| 3 | Người dùng | Xác nhận các thông báo đã được gửi đến |
| 4 | Hệ thống | Cập nhật tình trạng xác thực của thông báo tương ứng |
| **Tên usecase** | | | *Thích* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng (1) | Thả tim vào một bài viết |
| 2 | Hệ thống | Thêm thông tin người dùng (1) vào danh sách đã thích của bài viết. Hiển thị thông báo cho người dùng (2) |
| 3 | Người dùng (2) | Nhận thông báo có người dùng (1) tương tác |
| 4 | Người dùng (2) | Xác thực thông báo và phản hồi |
| **Tên usecase** | | | *Bình luận* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng (1) | Tạo bình luận từ một bài viết |
| 2 | Hệ thống | Tạo, lưu trữ thông tin vào CSDL và gửi thông báo đến người dùng (2) |
| 3 | Người dùng (2) | Xác thực thông báo |
| 4 | Hệ thống | Cập nhật trạng thái xác thực thông báo |

* **Chức năng Quản lý nội dung**

Ảnh có chứa biểu đồ, văn bản, vòng tròn, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 4 Usecase quản lý nội dung

Bảng số liệu 4 Đặc tả usecase quản lý nội dung

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên usecase** | | | *Đăng bài viết* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Nhập thông tin nội dung muốn đăng kèm theo file (nếu có) |
| 2 | Hệ thống | Tạo và lưu trữ bài viết vào CSDL và lưu file vào bộ nhớ (nếu có) |
|  | 3 | Hệ thống | Gửi thông báo đến người dùng và người dùng khác đang theo dõi người dùng |
| **Tên usecase** | | | *Chỉnh sửa bài viết* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Vào chi tiết 1 bài viết. Gửi thông tin nội dung muốn cập nhật, kèm theo file (nếu có) |
| 2 | Hệ thống | Cập nhật thông tin cho bài viết vào CSDL và lưu file vào bộ nhớ (nếu có) |
| 3 | Hệ thống | Gửi thông báo đến người dùng và người dùng khác đang theo dõi người dùng |

* **Chức năng tìm kiếm**

Ảnh có chứa văn bản, biểu đồ, vòng tròn, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 5 Usecase Tìm kiếm

Bảng số liệu 5: Đặc tả usecase tìm kiếm

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên usecase** | | | *Tìm kiếm* |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** | Đăng nhập thành công vào website | | |
| **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** |
| 1 | Người dùng | Nhận vào nút tìm kiếm |
| 2 | Hệ thống | Hiển thị trường nhập dữ liệu tìm kiếm |
| 3 | Người dùng | Nhập thông tin muốn tìm kiếm |
| 4 | Hệ thống | Gửi danh sách dữ liệu tìm kiếm được từ CSDL |

### 3.3.4 Biểu đồ hoạt động

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 6 Chức năng đăng nhập

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 7 Chức năng đăng ký

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 8 Chức năng đăng xuất

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 9 Chức năng quên mật khẩu

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 10 Chức năng theo dõi

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 11 Chức năng nhắn tin

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 12 Chức năng cập nhật tin nhắn

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 13 Chức năng thông báo

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 14 Chức năng đăng bài

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 15 Chức năng xóa bài viết

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 16 Chức năng tìm kiếm

## Thiết kế hệ thống

### 3.4.1. Thiết kế lớp

* *Sơ đồ lớp thực thể*

Ảnh có chứa văn bản, biểu đồ, hàng, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Hình 3. 17 Sơ đồ lớp thực thể

* **Mô tả chi tiết các lớp:**

**Lớp User**

Bảng số liệu 6 Mô tả chi tiết lớp User

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | String | * **createUser**(form: FormcreateUser): User * **authenticate**(username: String, password: String): User * **findById**(userId: String): User * **findAll**(): List<User> * **update** (userId: String, form: FormUpdateUser): User * **searchUser**(query: String): List<User> * **followUser**(userId: int, idFollower: int): User * **findUserByJwt**( String jwt) : User |
| 2 | email | String |
| 3 | username | String |
| 4 | password | String |
| 5 | firstName | String |
| 6 | lastName | String |
| 7 | gender | String |
| 8 | avatar | String |
| 9 | backGround | String |
| 10 | ngaySinh | LocalDate |
| 11 | follower | Set<Interger> |
| 12 | following | Set<Interger> |
| 13 | posts | List<Post> |
| **Lớp User** | | | |

**Lớp Post**

Bảng số liệu 7 Mô tả chi tiết lớp Post

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | Int | * **createPost**(form: FormCreatePost, idUser: int): Post * **update** (form: FormUpdatePost, idUser: int, idPost: int): Post * **deletePost** (idPost: int, idUser: int): void * **findAll**(): List<Post> * **findById**(idPost: int): Post * **savePost**(idPost: int, idUser: int): Post * **likePost**(idPost: int, idUser: int): Post * **findByUserId**( idUser: int): List<Post> |
| 2 | caption | String |
| 3 | image | String |
| 4 | video | String |
| 5 | like | List<User> |
| 6 | createdAt | LocalDateTime |
| 7 | updatedAt | LocalDateTime |
| 8 | comments | List<Comment> |
| **Lớp Post** | | | |

**Lớp Comment**

Bảng số liệu 8 Mô tả chi tiết lớp Comment

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | Int | * **create**(postId: Int, userId: Int, form: FormCreateComment): Comment * **updateComment**(commentId: Int, form: FormUpdateComment): Comment * **likeComment**(commentId: Int, idUser: Int): Comment * **deleteComment**(commentId: Int): void * **getComments**(postId: Int): List<Comment> |
| 2 | content | String |
| 3 | image | String |
| 4 | like | List<User> |
| 5 | createdAt | DateTime |
| **Lớp Comment** | | | |

**Lớp Message**

Bảng số liệu 9 Mô tả chi tiết lớp Message

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | Int | * **createMessage**(user: User, chatId: Int, form: FormCreateMesssage): Message * **findChatMessage**(chatId: Int): List<Message> * **getReactions**(postId: Int): List<Reaction> |
| 2 | content | String |
| 3 | timeStamp | DateTime |
| 4 | updatedAt | DateTime |
| **Lớp Message** | | | |

**Lớp Notification**

Bảng số liệu 10 Mô tả chi tiết lớp Notification

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | Int | * **create**(user: User, follower: Set<User>, message: String): Notification * **readNotification**(notificationId: Int): NotificationRecipient * **getNotificationsForUsers**(userId: Int): List<NotificationRecipient> |
| 2 | type | String |
| 3 | message | String |
| 4 | createdAt | DateTime |
| **Lớp Notification** | | | |

**Lớp NotificationRecipient**

Bảng số liệu 11 Mô tả chi tiết lớp NotificationRecipient

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | Int | * **setter**() * **getter**() |
| 2 | notification | Notification |
| 3 | timeStamp | DateTime |
| 4 | recipient | User |
| 5 | isRead | Boolean |
| **Lớp NotificationRecipient** | | | |

**Lớp Reels**

Bảng số liệu 12 Mô tả chi tiết lớp Reels

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | id | Int | * **createReels**(form: FormCreateReels, user: User): Reels * **findAll**(): List<Reels> * **findUserReels**( userId: Int ): List<Reels> |
| 2 | status | Int |
| 3 | reportTo | String |
| 4 | createdAt | DateTime |
| 5 | updatedAt | DateTime |
| 6 | postId | Int |
| 7 | senderId | String |
| 8 | userId | String |
| 9 | commentId | Int |
| 10 | groupId | Int |
| **Lớp Reels** | | | |

**Lớp ForgotPassword**

Bảng số liệu 13 Mô tả chi tiết lớp ForgotPassword

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Phương thức |
| 1 | fpid | Int | * **VerifyEmail**(form: FormForgotPassword): String * **VerifyOtp**(otp: InputOtp): String * **changePassword**(form: ChangePasswordForm): String * **otpGenerator**(): Int |
| 2 | otp | Int |
| 3 | ExpirationTime | DateTime |
| **Lớp ForgotPassword** | | | |

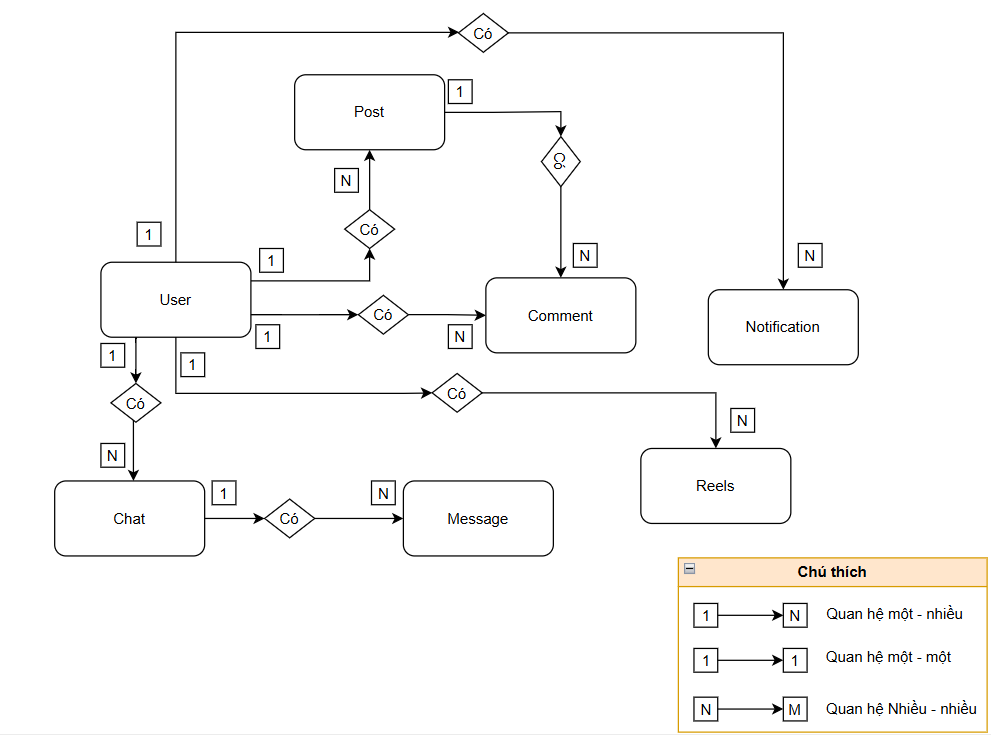
**Lớp Chat**

Bảng số liệu 14 Mô tả chi tiết lớp Chat

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuôc Tính** | **Kiểu Dữ Liệu** | **Phương Thức** |
| 1 | Id | Int | * **createChat**(userReq: int, User2: Int): Chat * **findChatById**(chatId: Int): Chat   **findUserChat**(userId: Int): List<Chat> |
| 2 | Chat | String |
| 3 | timestamp | DateTime |
| 4 | chatImg | String |
| **Lớp Chat** | | | |

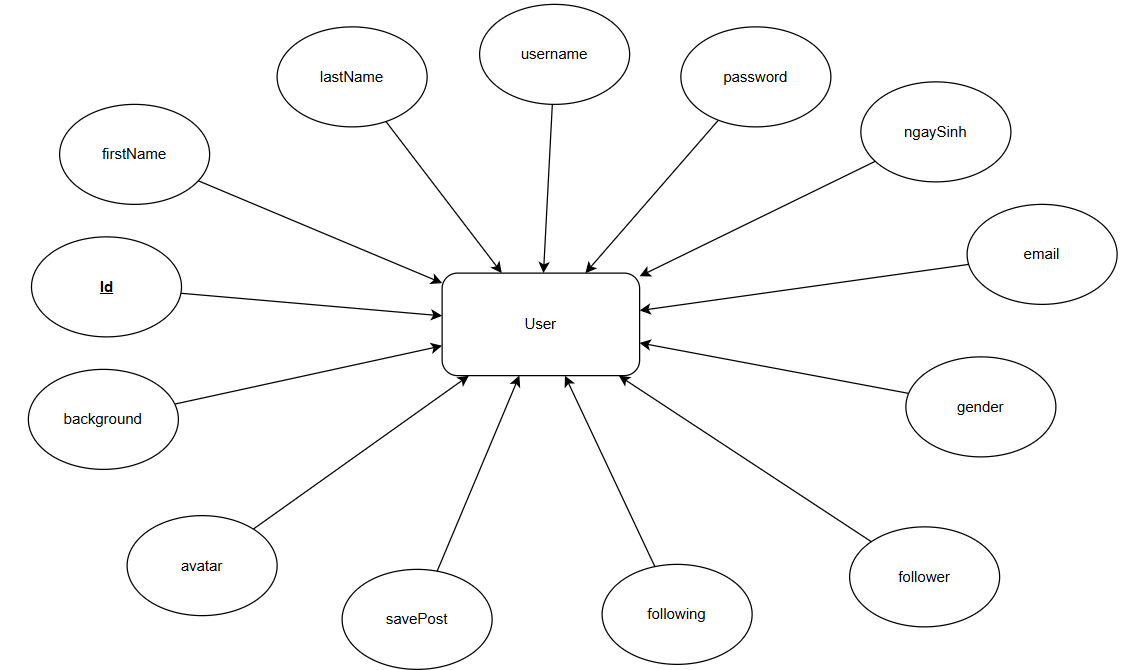
### 3.4.2. Thiết kế database

* ***Biểu đồ thực thể liên kết (ERD):***



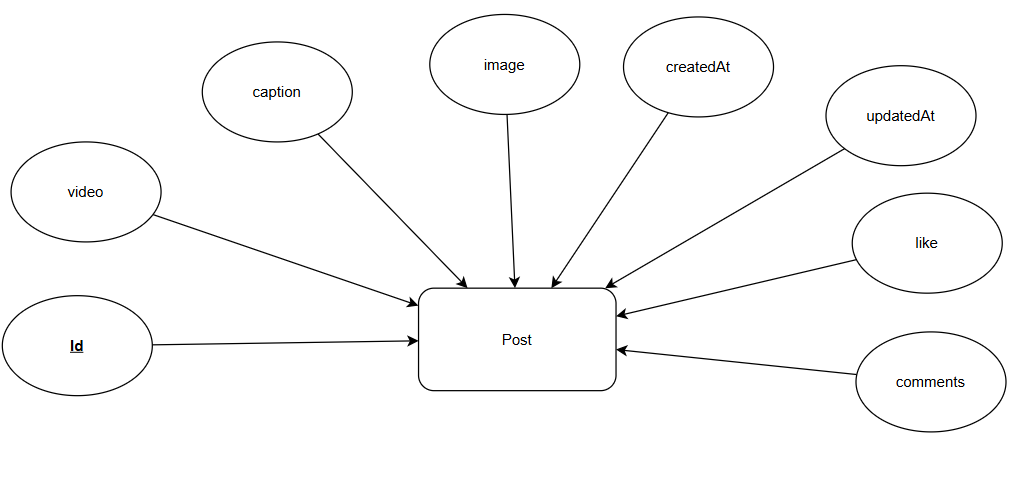
Hình 3. 18 Lược đồ ERD

* ***Mô tả chi tiết các thực thể:***
* Thực thể user



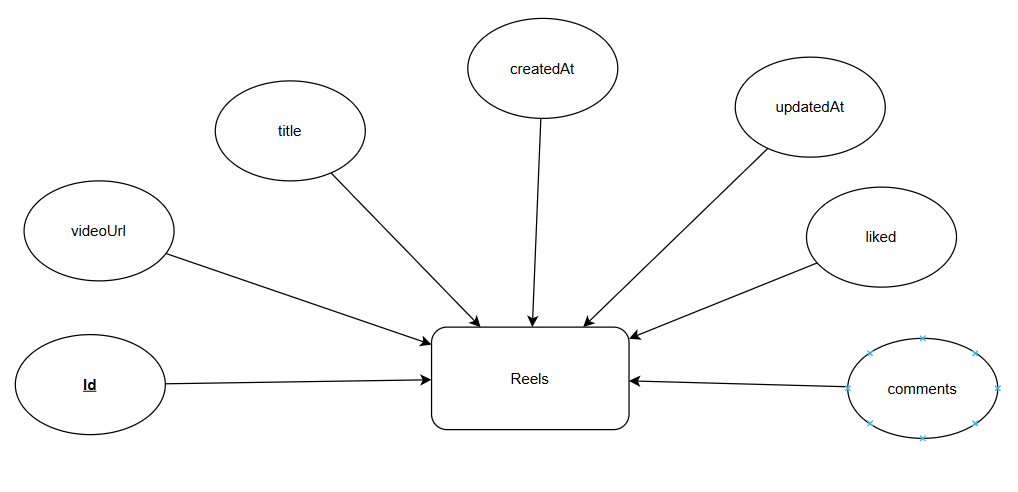
Hình 3. 19 Thực thể User

* Thực thể Post



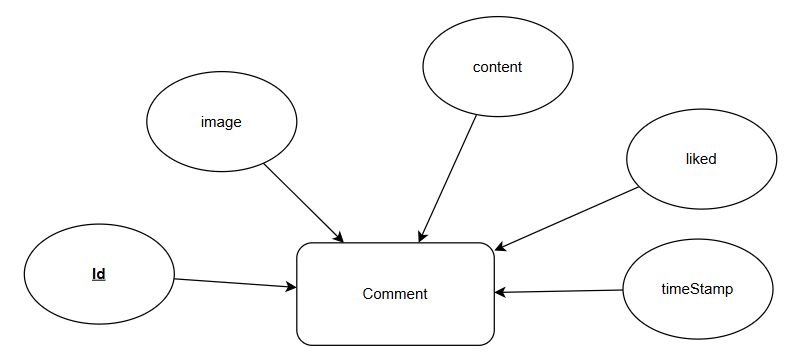
Hình 3. 20 Thực thể Post

* Thực thể Reels



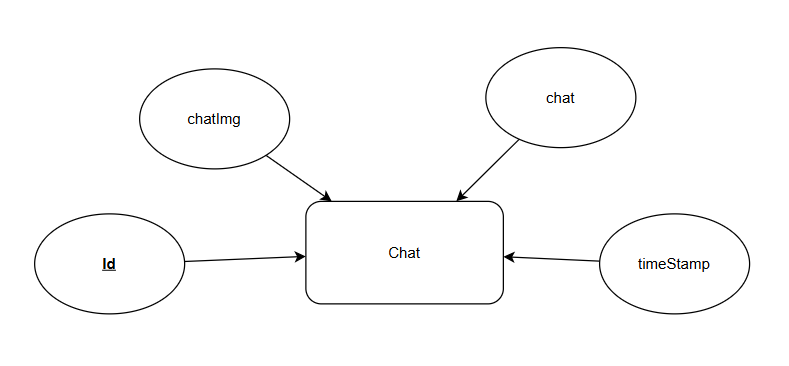
Hình 3. 21 Thực thể Reels

* Thực thể Comment



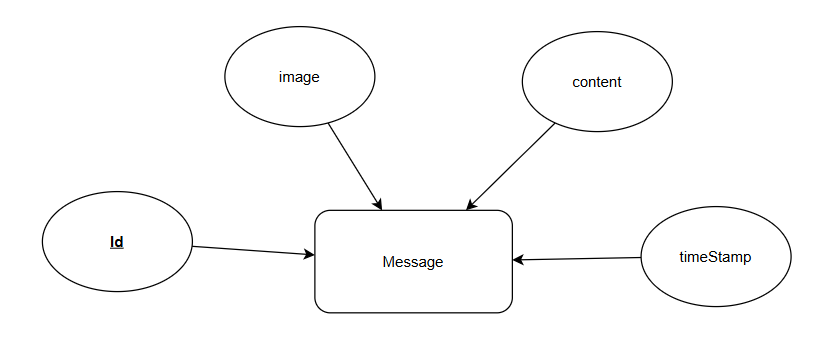
Hình 3. 22 Thực thể Comment

* Thực thể Chat



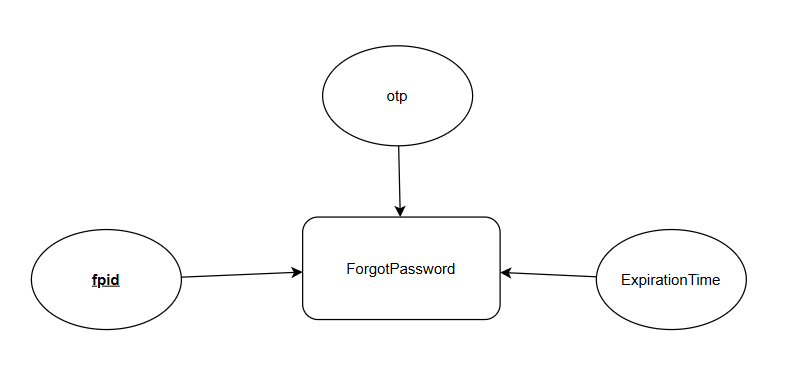
Hình 3. 23 Thực thể Chat

* Thực thể Message



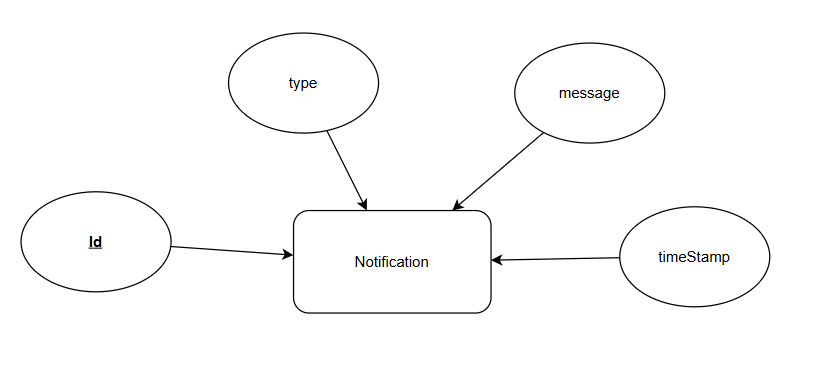
Hình 3. 24 Thực thể Message

* Thực thể ForgotPassword



Hình 3. 25 Thực thể forgotPassword

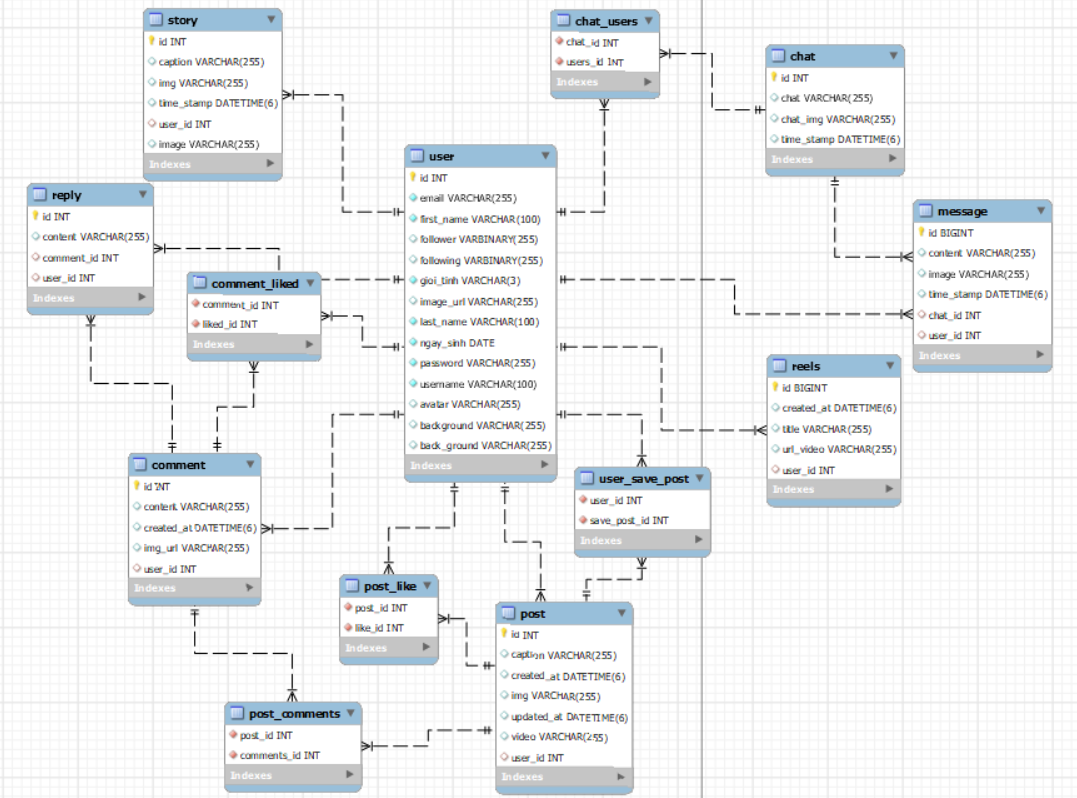
* Thực thể notification



Hình 3. 26 Thực thể Notification

# CHƯƠNG 4: XÂY DƯNG HỆ THỐNG

## Xây dựng cơ sở dữ liệu vật lý



Hình 4. 1 Mô hình quan hệ tables