

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



BÁO CÁO ĐỒ ÁN
KỸ THUẬT LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
ĐỀ TÀI
“XÂY DỰNG ỨNG DỤNG MÔ PHỎNG MẠNG XÃ HỘI”

Giảng viên : TS. HUỖNH TRUNNG TRỰ

Sinh viên : NGUYỄN THANH DỮNG - N19DCCN030

Sinh viên : NGUYỄN MINH KHANG - N19DCCN086

Sinh viên : NGUYỄN THÁI TƯỜNG - N19DCCN184

Lớp : D19CQCP01-N

Khóa : 2019 - 2024

Hệ : ĐẠI HỌC CHÍNH QUY

TP. HCM, tháng 10/2022

LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến tất cả thầy cô giảng dạy đã tận tình dìu dắt và chia sẻ cho chúng em những kiến thức vô cùng quan trọng và quý báu trong quá trình học tập tại Học viện để chúng em có những kỹ năng và kiến thức cần thiết để hoàn thành đồ án này.

Đặc biệt, chúng em vô cùng biết ơn thầy ThS. Huỳnh Trung Trự đã tận tình hướng dẫn, truyền đạt những kiến thức, kinh nghiệm của thầy và dạy bảo chúng em trong quá trình học tập để thực hiện báo cáo này. Chúng em xin chúc thầy và gia đình có thật nhiều sức khỏe và thành công trong cuộc sống. Từ đó mang đến cho chúng em cũng như các bạn khác những giá trị và kiến thức từ thầy.

Cảm ơn tất cả những người bạn luôn đồng hành, chia sẻ, giúp đỡ và cổ vũ tinh thần cho chúng em trong quá trình học tập và rèn luyện các kỹ năng. Nhờ vậy, chúng em đã có thể vượt qua những khó khăn và áp lực trong học tập cũng như trong cuộc sống.

Báo cáo đã được chúng em hoàn thành đúng tiến độ và đảm bảo nội dung của đề tài. Mặc dù, chúng em đã cố gắng hoàn thiện đề tài nhưng sẽ không tránh khỏi những thiếu sót và nhiều điều còn chưa hoàn toàn chính xác. Mong thầy giúp chúng em đóng góp ý kiến để có nhiều hơn những kinh nghiệm và kỹ năng để hoàn thành tốt công tác nghiên cứu, làm việc sau này. Một lần nữa nhóm xin cảm ơn thầy và các bạn rất nhiều.

Tp Hồ Chí Minh, tháng 11 năm 2023

Mục Lục

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	2
1.1 Tìm hiểu “Xây dựng website mô phỏng mạng xã hội:.....	2
1.2 Nền tảng công nghệ:.....	2
1.3 Công cụ lập trình và công nghệ được sử dụng:.....	2
1.4 Giới thiệu ngôn ngữ lập trình Java:	3
1.5 Các thư viện – framework hỗ trợ:	4
1.5.1 Bootstrap :.....	4
1.5.2 JavaScript:.....	4
1.5.3 ReactJS:	4
1.5.4 Tổng quan về SQL:.....	7
1.5.5 PostgreSQL:.....	8
1.5.6 RESTful Web Service:	9
1.5.7 Android studio:	11
1.5.8 Giới thiệu về socket IO:.....	12
CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG.....	7
2.1 Hiện trạng thực tế khi mạng xã hội chưa xuất hiện:	7
2.2 Giá trị mang lại khi xây dựng mạng xã hội:.....	7
2.3 Nghiệp vụ của hệ thống:	8
2.3.1 Người dùng hệ thống:	8
2.3.2 Yêu cầu chức năng:	12
2.3.3 Yêu cầu phi chức năng:	12
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	12
3.1.1 Sơ đồ phân cấp chức năng mức đỉnh:	12
3.1.2 Sơ đồ phân rã chức năng:.....	12
3.2 Mô hình luồng dữ liệu DFD (Data Flow Diagram)	16
3.2.1 Biểu đồ DFD mức ngữ cảnh:	16
3.2.2 Mô hình luồng dữ liệu DFD phân rã cấp 0 (chức năng chính):.....	17
3.3 Mô hình usecase:	18
3.3.1 Xác định Actor:.....	18
3.3.2 Xác định các Use Case:	18
3.3.3 Xây dựng biểu đồ Use Case tổng quát:	18
3.3.4 Đặc tả use case:.....	19
3.3.5 Biểu đồ tuần tự tương ứng với các Use Case:	26
3.4 Xác định thực thể:	29
3.5 Mô hình ERD:	29
3.6 Mô hình cơ sở dữ liệu quan hệ:.....	30
3.7 Mô hình dữ liệu:	34
3.7.1 Xét USER– CONVERSATION:	34

3.7.2 Xét USER– MESSAGE:.....	34
3.7.3 Xét MESSAGE– CONVERSATION:.....	34
3.7.4 Xét USER– POST:.....	34
3.7.5 Xét USER– USER:	34
3.7.8 Xét POST– POST_IMAGE:.....	35
3.7.8 Xét COMMENT– COMMENT:.....	35
3.8 Phân tích và hoàn thiện mô hình quan hệ:	36
CHƯƠNG 4. ỨNG DỤNG WEB APPLICATION.....	50
4.1 Yêu cầu hệ thống:.....	50
4.2 Demo giao diện:	50
4.2.1 Giao diện đăng nhập:	50
4.2.2 Giao diện trang chủ:.....	50
4.2.3 Quản lý bài viết:.....	51
4.2.4 Chức năng tìm kiếm:.....	56
4.2.5 Quản lý tài khoản:.....	57
4.2.6 Kết bạn/theo dõi:.....	62
4.2.7 Quản lý tin nhắn:.....	65
4.2.8 Quản lý thông báo:.....	68
CHƯƠNG 5. RESTFUL WEB SERVICE.....	70
5.1 Công nghệ sử dụng:	70
5.2 Web Service:	70
5.3 Tính chất:	70
5.3.1 Sử dụng các phương thức HTTP một cách rõ ràng:	70
5.3.2 Phi trạng thái:	71
5.3.3 Truyền tải JavaScript Object Notation (JSON), XML hoặc cả hai:	71
5.4 Bảo mật REStful Service sử dụng Auth0 JWT:.....	71
5.4.1 JWT - JSON Web Token:.....	71
5.4.2 Sơ lược về luồng xử lý:.....	72
CHƯƠNG 6. ỨNG DỤNG ANDROID APP	65
6.1 Yêu cầu hệ thống:.....	65
6.2 Demo giao diện:	65
6.2.1 Đăng nhập:.....	65
6.2.2 Trang chủ:.....	65
6.2.3 Bài viết:.....	66
6.2.4 Tìm kiếm.....	73
6.2.5 Trang cá nhân.....	74
6.2.6 Nhắn tin.....	78
6.2.7 Thông báo	80
CHƯƠNG 7. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN	81
7.1 Tổng kết:	81

7.2 Hướng phát triển:	81
<i>DANH MỤC TÀI LIỆU THAM KHẢO</i>	82

KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT

API:	Application Program Interface
URI:	Uniform Resource Identifier
CSDL:	Cơ sở dữ liệu
SQL:	Structured Query Language
UI:	User Interface
HTML:	Hyper Text Markup Language
UML:	Unified Modeling Language
ERD:	Entity Relationship Diagram
ORM:	Object Relational Mapping

LỜI MỞ ĐẦU

Ngày nay sự gia tăng mạnh mẽ của mạng xã hội và vai trò quan trọng của chúng trong giao tiếp và chia sẻ thông tin đã thúc đẩy quan tâm tới việc phát triển ứng dụng mạng xã hội. Điều này tạo nên cơ hội hấp dẫn để nghiên cứu và phát triển các ứng dụng đáp ứng nhu cầu người dùng.

Mục đích nghiên cứu: Phân tích, thiết kế và xây dựng một ứng dụng mạng xã hội, tập trung cung cấp trải nghiệm tốt cho người dùng, khắc phục những hạn chế của các ứng dụng hiện có và tạo ra một không gian trực tuyến an toàn và sáng tạo.

Phạm vi nghiên cứu: Nghiên cứu sẽ tập trung phân tích yêu cầu của người dùng, thiết kế giao diện người dùng thân thiện và dễ sử dụng, xây dựng tính năng bảo mật mạnh mẽ, giúp ngăn chặn các hoạt động độc hại và tạo ra môi trường trực tuyến tích cực.

Phương pháp nghiên cứu: Nghiên cứu sẽ bắt đầu bằng việc tiến hành phân tích nhu cầu người dùng và các vấn đề hiện tại của các ứng dụng mạng xã hội. Sau đó, sẽ thực hiện thiết kế giao diện và tính năng dựa trên các yêu cầu đã xác định. Quá trình phát triển ứng dụng sẽ sử dụng các phương pháp phát triển phần mềm linh hoạt để đảm bảo sự linh hoạt và phản hồi nhanh chóng trong quá trình xây dựng.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Tìm hiểu “Xây dựng website mô phỏng mạng xã hội:

Ngày nay sự gia tăng mạnh mẽ của mạng xã hội và vai trò quan trọng của chúng trong giao tiếp và chia sẻ thông tin đã thúc đẩy quan tâm tới việc phát triển ứng dụng mạng xã hội. Điều này tạo nên cơ hội hấp dẫn để nghiên cứu và phát triển các ứng dụng đáp ứng nhu cầu người dùng.

Mạng xã hội di động đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của con người. Người dùng sử dụng các ứng dụng di động mạng xã hội để kết nối, chia sẻ thông tin và tương tác với nhau. Các ứng dụng di động như Facebook, Instagram và Twitter đã thu hút hàng tỷ người dùng trên toàn thế giới và trở thành nền tảng quan trọng cho việc giao tiếp và kết nối xã hội.

Trong những năm gần đây, sự phát triển của ứng dụng di động mạng xã hội đã đạt đến một tầm cao mới. Các ứng dụng này không chỉ cung cấp các tính năng cơ bản như chia sẻ tin tức, hình ảnh và video, mà còn bổ sung nhiều tính năng mới như livestream, stories và tích hợp trò chơi. Người dùng cũng có thể tạo nhóm, tham gia các cộng đồng chuyên môn và tìm kiếm thông tin quan trọng thông qua các ứng dụng mạng xã hội di động.

Người dùng ứng dụng mạng xã hội di động ngày càng tăng lên với tốc độ nhanh chóng. Họ sử dụng ứng dụng này để kết nối với bạn bè, gia đình và đồng nghiệp, chia sẻ cuộc sống hàng ngày, tìm kiếm thông tin và giải trí. Người dùng đặt nhu cầu cao về tính tương tác, tính đa dạng của nội dung và sự dễ dàng trong việc sử dụng các ứng dụng mạng xã hội di động. Họ cũng quan tâm đến việc bảo mật thông tin cá nhân và quyền riêng tư trên mạng xã hội di động.

Mặc dù ứng dụng mạng xã hội di động đã đạt được sự phát triển lớn, nhưng còn nhiều thách thức đối diện. Vấn đề về bảo mật thông tin cá nhân, tin tức giả mạo và vi phạm quyền riêng tư là những vấn đề nổi cộm cần được giải quyết. Tuy nhiên, cũng có nhiều cơ hội để xây dựng ứng dụng mạng xã hội di động tốt hơn, bao gồm tích hợp trí tuệ nhân tạo, tạo ra trải nghiệm người dùng độc đáo và phát triển các tính năng sáng tạo để thu hút và giữ chân người dùng.

1.2 Nền tảng công nghệ:

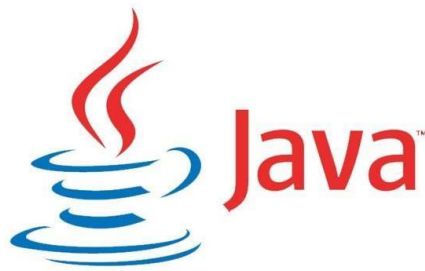
- Web app: Sử dụng reactjs, html, css, javascript, bootstrap,...
- Android app: Sử dụng java, android studio.
- RESTful Web Service: Sử dụng nodejs.

1.3 Công cụ lập trình và công nghệ được sử dụng:

- Hệ quản trị cơ sở dữ liệu: PostgreSQL

- Công cụ lập trình: Visual studio code, android studio.
- Ngôn ngữ lập trình: Java, javascript,
- Xây dựng giao diện front-end: Reactjs, HTML, CSS, Bootstrap, Javascript, java.
- Công nghệ xử lý back-end nodejs.
- Xây dựng Webservice.
- Restfull API NodeJS.
- Lập trình ứng dụng di động Android studio.
- Realtime với socketIO.

1.4 Giới thiệu ngôn ngữ lập trình Java:



Hình 1. 1 Giới thiệu ngôn ngữ lập trình Java

- Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

- Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.
- Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó .

1.5 Các thư viện – framework hỗ trợ:

1.5.1 Bootstrap :

- Bootstrap là một front-end framework miễn phí giúp quá trình phát triển web được nhanh và dễ dàng hơn. Bootstrap bao gồm các mẫu thiết kế dựa trên HTML và CSS như typography, forms, buttons, tables, navigation, modals, image carousels... cũng như các plugin JavaScript tùy chọn. Bootstrap cũng cung cấp cho bạn khả năng tạo ra các responsive designs một cách dễ dàng.

1.5.2 JavaScript:

- JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

1.5.3 ReactJS:

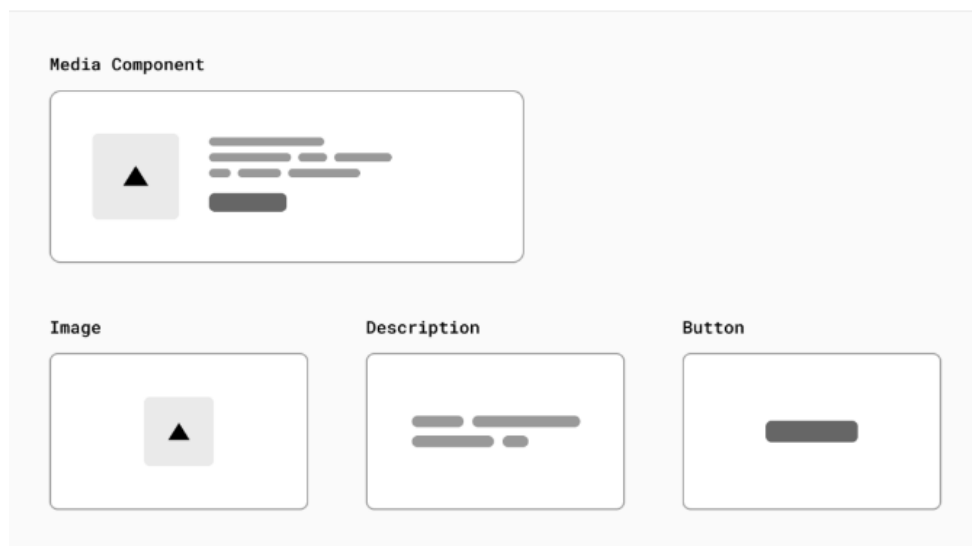
- ReactJS là một thư viện JavaScript phổ biến và mạnh mẽ được sử dụng rộng rãi trong việc phát triển giao diện người dùng động cho ứng dụng web. Nó được tạo ra bởi Facebook và có tính năng tái sử dụng thành phần, hiệu suất cao và khả năng xây dựng ứng dụng đơn trang (single-page applications - SPAs) để tạo ra các ứng dụng web động mà không cần tải lại trang.
- ReactJS sử dụng cú pháp JSX (JavaScript XML) để xây dựng các thành phần giao diện. JSX kết hợp cú pháp JavaScript và HTML, cho phép lập trình viên viết mã gọn gàng và dễ đọc. Cấu trúc ReactJS dựa trên mô hình khối xây dựng thành phần, trong đó mỗi thành phần độc lập và có thể được sử dụng lại trong ứng dụng.

Ví dụ về JSX: `const element = <h1>Hello world!</h1>`

- Có 3 khái niệm cốt lõi của Reactjs cần nắm được khi muốn xây dựng

một ứng dụng

- Reactjs:
- State
- Props
- Component
 - Component đại diện cho một phần giao diện người dùng độc lập và có thể được tái sử dụng trong ứng dụng. Component là một khái niệm quan trọng trong React, giúp tạo ra giao diện linh hoạt và dễ dàng quản lý.

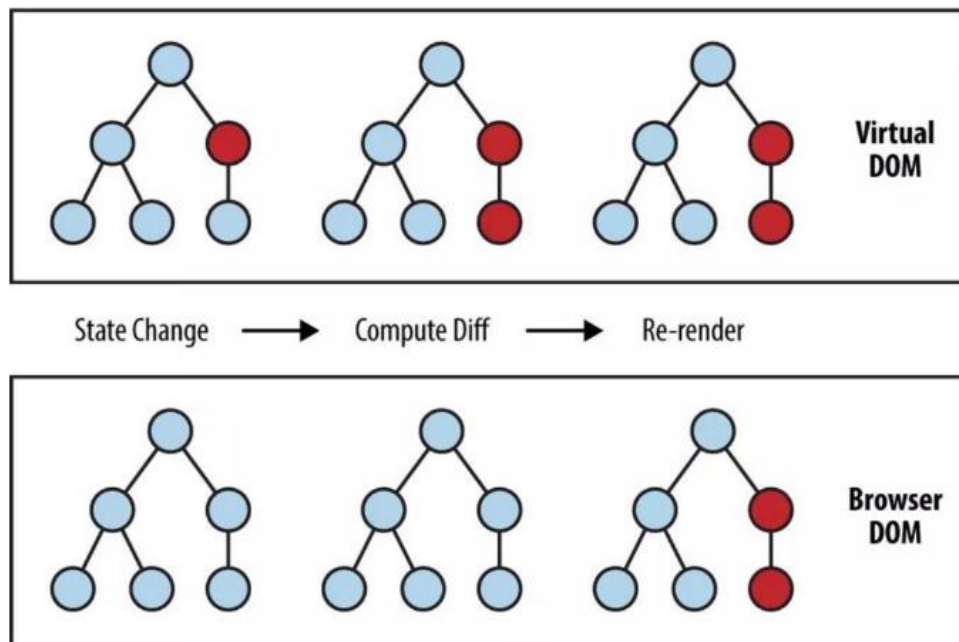


Hình 1. 3 Khái niệm component trong ReactJS

- Props (viết tắt của properties) là một cách để truyền dữ liệu từ component cha đến component con thông qua các thuộc tính. Props là một đối tượng không thay đổi (immutable) và được truyền từ component cha, không thể thay đổi giá trị của props trong component con.
- State là một đối tượng chứa các dữ liệu nội bộ của một component. State được sử dụng để lưu trữ và quản lý các thông tin có thể thay đổi trong quá trình hoạt động của component. State là một khái niệm quan trọng trong React, cho phép quản lý và cập nhật dữ liệu nội bộ của một component. Khi state thay đổi, React tự động cập nhật lại giao diện của component đó, giúp đảm bảo tính tương tác và đồng bộ trong ứng dụng React.
- State và props thường được sử dụng cùng nhau để quản lý và truyền dữ liệu trong ứng dụng React. Thông thường, state được sử dụng để lưu trữ các thông tin nội bộ và có khả năng thay đổi trong một component. Props được sử dụng để truyền dữ liệu từ component cha vào component con, cho phép tái sử dụng và tạo sự tương tác giữa các

component. Component con có thể sử dụng props để hiển thị dữ liệu hoặc thực hiện các logic, và cũng có thể sử dụng state nếu cần thiết để lưu trữ thông tin nội bộ.

- ReactJS có thể được tích hợp vào ứng dụng web sử dụng các công cụ và thư viện hỗ trợ như Babel và Webpack. Babel giúp biên dịch mã JSX và ES6 thành JavaScript thông thường để trình duyệt có thể hiểu. Webpack hỗ trợ quản lý các module và tạo ra các tệp bundle để tải xuống trình duyệt.
- Trong quá trình tích hợp ReactJS vào dự án, ta cần xây dựng các thành phần giao diện, quản lý trạng thái (state) của ứng dụng, và sử dụng các lifecycle methods để thực hiện các tác vụ như khởi tạo, cập nhật và hủy bỏ.
- ReactJS có hiệu suất cao nhờ vào cách làm việc hiệu quả của nó. ReactJS sử dụng Virtual DOM (Document Object Model) để quản lý cập nhật giao diện. Khi có sự thay đổi trong trạng thái của ứng dụng, ReactJS sẽ tạo ra một phiên bản Virtual DOM mới và so sánh nó với phiên bản cũ để xác định những thay đổi cần được áp dụng vào DOM thực tế. Quá trình này giúp giảm tối đa việc truy cập và cập nhật trực tiếp vào DOM, từ đó tăng hiệu suất ứng dụng.
- ReactJS sử dụng quá trình so sánh và cập nhật (reconciliation) để xác định những phần giao diện thực sự cần được cập nhật. React so sánh các thành phần trước và sau khi thay đổi, tìm ra sự khác biệt và chỉ cập nhật những phần thay đổi đó. Việc này giúp giảm bớt thời gian và công sức cần thiết để cập nhật giao diện, cải thiện hiệu suất của ứng dụng.



Hình 1. 4 Cập nhật DOM trong Reactjs

- ReactJS là một công nghệ mạnh mẽ và linh hoạt cho việc xây dựng giao diện người dùng động, với khả năng tái sử dụng thành phần và hiệu suất

cao. Việc tích hợp ReactJS vào dự án cần sử dụng các công cụ hỗ trợ như Babel và Webpack, và áp dụng các nguyên tắc và cấu trúc của ReactJS để tận dụng tối đa lợi ích của nó trong phát triển ứng dụng web.

- NodeJS JavaScript toàn diện: Node.js cho phép sử dụng JavaScript cả phía máy chủ và phía máy khách. Điều này giúp đơn giản hóa việc phát triển và duy trì mã nguồn, vì bạn có thể sử dụng cùng một ngôn ngữ lập trình trên cả hai phía.
- Khả năng xử lý đa luồng: Node.js sử dụng mô hình không đồng bộ (asynchronous) và sự kiện trên một luồng duy nhất (single-threaded) để xử lý các yêu cầu I/O. Điều này giúp tối ưu hóa hiệu suất và độ phản hồi của ứng dụng nhắn tin online, cho phép xử lý hàng nghìn kết nối đồng thời mà không gây chậm trễ.
- Thư viện npm phong phú: Node.js có npm (Node Package Manager), là một kho lưu trữ rất lớn chứa hàng ngàn thư viện và công cụ mở rộng. Có thể tìm thấy các gói npm hữu ích để xây dựng các tính năng nhắn tin, giao tiếp realtime, quản lý người dùng, và nhiều thư viện khác để giúp phát triển ứng dụng nhắn tin online dễ dàng hơn.
- Scalability (khả năng mở rộng): Node.js cho phép xây dựng ứng dụng có khả năng mở rộng. Với cách Node.js xử lý các yêu cầu không đồng bộ và khả năng xử lý đa luồng hiệu quả, có thể xử lý nhiều kết nối cùng một lúc và mở rộng hệ thống theo nhu cầu tải lên.
- Real-time communication (giao tiếp thời gian thực): Node.js có hỗ trợ tốt cho giao tiếp thời gian thực qua WebSockets hoặc các giao thức khác như Socket.io hoặc MQTT.
- Điều này giúp xây dựng tính năng nhắn tin trực tiếp và cập nhật realtime trong ứng dụng nhắn tin online.

1.5.4 Tổng quan về SQL:

- SQL là viết tắt của Structured Query Language là ngôn ngữ truy vấn có cấu trúc. Nó là một ngôn ngữ, là tập hợp các lệnh để tương tác với cơ sở dữ liệu. Dùng để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. Trong thực tế, SQL là ngôn ngữ chuẩn được sử dụng hầu hết cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Postgres và SQL Server... đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.
- SQL bắt đầu từ những năm 1970, khi các kỹ sư của IBM là Donald Chamberlin và Raymond Boyce thiết kế phiên bản đầu tiên để tương tác và 24 lấy dữ liệu được lưu trữ trong hệ thống cơ sở dữ liệu của công ty. Họ gọi nó là SEQUEL, mặc dù sau đó họ buộc phải thay đổi nó do các vấn đề bản quyền. SQL được sử dụng phổ biến vì nó có các ưu điểm sau:
 - o Cho phép truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.

- Cho phép mô tả dữ liệu.
 - Cho phép xác định dữ liệu trong cơ sở dữ liệu và thao tác dữ liệu đó.
 - Cho phép nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL, thư viện
 - và trình biên dịch trước.
 - Cho phép tạo và thả các cơ sở dữ liệu và bảng.
 - Cho phép tạo chế độ view, thủ tục lưu trữ, chức năng trong cơ sở dữ liệu.
 - Cho phép thiết lập quyền trên các bảng, thủ tục và view.
- Hầu như công ty nào lớn cũng cần xây dựng một hệ thống để lưu trữ cơ sở dữ liệu. Mọi thứ trong cơ sở dữ liệu này sẽ được diễn tả ra thành nhiều bảng, có mối quan hệ với nhau. Để truy vấn và lấy dữ liệu từ các bảng này nhằm tổng hợp thành thông tin nào đó, người ta dùng đến SQL thông qua các câu query.
 - PostgreSQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì PostgreSQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả
 - chuyên, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, PostgreSQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên Internet. Người dùng có thể tải về PostgreSQL miễn phí từ trang chủ.

1.5.5 PostgreSQL:

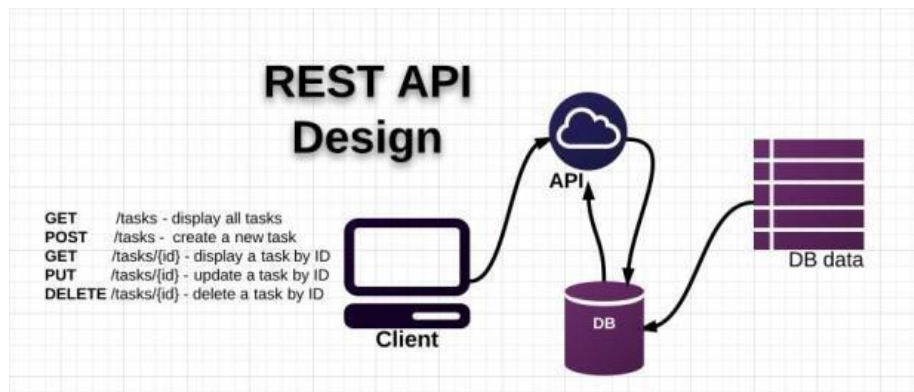
- Độ tin cậy cao: PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ và đáng tin cậy. Nó được xây dựng trên kiến trúc ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính nhất quán và độ tin cậy cao cho dữ liệu của bạn.
- Hiệu suất tối ưu: PostgreSQL có thiết kế và tối ưu hóa cho hiệu suất cao. Nó hỗ trợ các tính năng như đánh chỉ mục (indexing), tối ưu hóa truy vấn (query optimization), và tối ưu hóa I/O để cung cấp truy vấn nhanh chóng và hiệu suất tốt cho ứng dụng nhắn tin online, đặc biệt khi phải xử lý nhiều truy vấn đồng thời.
- Hỗ trợ JSON và dữ liệu phức tạp: PostgreSQL hỗ trợ lưu trữ và truy vấn dữ liệu
- JSON, cho phép bạn lưu trữ thông tin phức tạp như tin nhắn, hình ảnh, và định dạng dữ liệu khác một cách linh hoạt. Điều này rất hữu ích trong việc lưu trữ và truy xuất dữ liệu trong ứng dụng nhắn tin online.
- An ninh và quản lý người dùng: PostgreSQL cung cấp nhiều tính năng bảo mật và quản lý người dùng linh hoạt. Bạn có thể thiết lập quyền truy cập, mã hóa dữ liệu, xác thực người dùng và quản lý phiên làm

việc để đảm bảo an toàn và bảo mật dữ liệu của ứng dụng nhắn tin.

- Hỗ trợ mở rộng: PostgreSQL có khả năng mở rộng tốt. Nó cho phép bạn tăng cường hiệu suất và khả năng chịu tải của hệ thống bằng cách thêm các máy chủ phụ (replication), phân vùng dữ liệu (sharding), và kỹ thuật nhân rộng khác. Điều này rất quan trọng khi ứng dụng nhắn tin online của bạn phải xử lý số lượng người dùng và dữ liệu lớn.

1.5.6 RESTful Web Service:

- RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 1. 5 Giới thiệu RESTful Web Service

- Diễn giải các thành phần:
 - API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.
 - REST (Representational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.
 - RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến

ngày nay để cho các ứng dụng (web, mobile...) khác nhau
giao tiếp với nhau.

- Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API. JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

1.5.7 Android studio:

- Android Studio là một phần mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập emulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.



Hình 1. 5 Giới thiệu Android Studio

- Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiết bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng

tương ứng với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức.

- Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.

1.5.8 Giới thiệu về socket IO:

- Khi truy cập vào 1 trang web hoặc ứng dụng bất kỳ thì việc giao tiếp giữa máy chủ (Server) và máy khách (Client) là việc rất quan trọng. Để máy chủ và máy khách có thể nhận biết được sự thay đổi của đối phương thì cần sử dụng những cách thức như AJAX, long-polling, short-polling, & HTML5 server-sent events,..Việc sử dụng cách giao tiếp bằng những công cụ kể trên tồn tại nhiều nhược điểm trong đó có thể kể đến là kết quả trả về chậm và tốn rất nhiều tài nguyên.
- Để khắc phục những nhược điểm này, công cụ socket.io ra đời để giúp cho việc giao tiếp giữa Server và Client diễn ra tức khắc và chiếm ít tài nguyên nhất.
- **Socket io** được xem là 1 module trong Node.js được nhà sáng chế tạo ra và phát triển từ năm 2010. Mục đích lớn nhất của Socket io là để tạo môi trường giao tiếp thuận lợi trên Internet giúp trả về các giá trị thực ngay tại thời điểm giao tiếp giữa các bên với nhau (thường là giữa server và client).
- Việc giao tiếp 2 chiều giữa máy khách và máy chủ được thực hiện bởi socket io khi và chỉ khi máy khách có module này trong trình duyệt và máy chủ cũng đã tích hợp sẵn gói socket io. Các ứng dụng sử dụng socket io thường đòi hỏi tốc độ phản hồi ngay lập tức. Một số ví dụ điển hình như xổ số, trực tiếp bóng đá, chat...

Ưu điểm của socket IO:

- **Bảo mật cao:** Socket io được xây dựng dựa trên Engine.IO. Nó sẽ khởi chạy phương thức long-polling trước nhất để kết nối. Sau đó nó mới sử dụng các phương thức giao tiếp tốt hơn như là Websocket chẳng hạn. Vì được thiết lập chặt chẽ như vậy nên khi socket io xuất hiện nó sẽ tự động tạo những kết nối bảo mật như là: proxy và cân bằng tải hoặc là tường lửa cá nhân và phần mềm chống vi rút.

- **Kết nối tự động tới server:** Đặc điểm tự động kết nối đến server của socket io là gì? Giả sử trong quá trình khởi chạy bị mất kết nối giữa client và server thì socket io sẽ tự động gắn kết nối mãi mãi cho đến khi nào server phản hồi lại. Và đây là tính năng có thể tùy chỉnh được nên bạn có quyền chọn không kết nối tự động đến bất kỳ server nào mà mình muốn.
- **Mã hóa nhị phân :** Socket io có thể hỗ trợ mã hóa nhị phân như ArrayBuffer và Blob trên trình duyệt hoặc là ArrayBuffer và Buffer trong Node.js.
- **Cho phép tạo kênh và phòng:** Có thể nói đây là 1 tính năng khá nổi bật khi mà socket io có thể tạo ra mối quan hệ giữa các phần hoặc các module riêng lẻ bằng cách tạo ra những kênh riêng biệt khác nhau. Ngoài việc tạo kênh, nó còn hỗ trợ tạo phòng cho các clients tham gia với mục đích gửi thông báo đến 1 nhóm người dùng được kết nối với 1 số thiết bị nào đó chẳng hạn.

CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG

2.1 Hiện trạng thực tế khi mạng xã hội chưa xuất hiện:

- Trước sự ra đời của mạng xã hội, cuộc sống xã hội diễn ra chủ yếu thông qua các hình thức truyền thống. Giao tiếp giữa con người diễn ra qua các phương tiện như điện thoại cố định, thư tín, và gặp trực tiếp. Không có khái niệm về việc kết nối trực tuyến, mọi giao tiếp thường giới hạn trong phạm vi cộng đồng địa phương.
- Trong thời kỳ này, thông tin chủ yếu được truyền tải qua các phương tiện truyền thông truyền thống như radio, truyền hình, báo chí và tạp chí. Thông tin không diễn ra ngay lập tức và thường có sự kiểm soát lớn hơn từ phía các tổ chức truyền thông. Quảng cáo và tiếp thị chủ yếu dựa vào các phương tiện truyền thông truyền thống, đòi hỏi chi phí lớn và không linh hoạt như chiến lược tiếp thị trực tuyến ngày nay.
- Sự kết nối xã hội tập trung vào cộng đồng địa phương, thông qua việc gặp mặt, tham gia sự kiện cộng đồng, và các hoạt động xã hội. Sự tương tác trực tiếp giữa mọi người chiếm vai trò lớn, và không có khái niệm về việc tạo ra mối quan hệ trực tuyến như ngày nay.
- Sự riêng tư cá nhân được bảo vệ chủ yếu thông qua các phương tiện giấy tờ, và thông tin cá nhân được giữ riêng tư hơn so với hiện đại. Sự tìm kiếm thông tin thường phụ thuộc vào thư viện, sách, và nguồn thông tin giới hạn, tạo ra một hạn chế đáng kể so với khả năng truy cập thông tin hiện đại.
- Tổng quan, trước khi có mạng xã hội, cách mọi người tương tác và tiếp cận thông tin có sự hạn chế hơn, và sự kết nối chủ yếu tập trung vào cộng đồng địa phương và các phương tiện truyền thông truyền thống.

2.2 Giá trị mang lại khi xây dựng mạng xã hội:

- Mạng xã hội đã đem đến nhiều giải pháp để giải quyết những hạn chế và nhu cầu xã hội trong bối cảnh trước khi nó xuất hiện. Dưới đây là một số giải pháp mà mạng xã hội mang lại:

Tiện ích	Chi tiết
Kết nối toàn cầu	<ul style="list-style-type: none"> - Mạng xã hội cho phép giao tiếp ngay lập tức qua internet, vượt qua rào cản địa lý và thời gian, tạo ra kết nối toàn cầu - Mọi người có thể tương tác qua văn bản, hình ảnh, video và âm thanh, mang đến trải nghiệm giao tiếp đa dạng và phong phú hơn.
Truy cập thông tin nhanh chóng	<ul style="list-style-type: none"> - Mạng xã hội cung cấp khả năng tìm kiếm thông tin ngay lập tức, không giới hạn và dễ dàng hơn so với thời kỳ trước đó - Người dùng có thể chia sẻ thông tin, kiến thức và kinh nghiệm của họ, mở rộng nguồn thông tin cho cộng đồng

Xây dựng mối quan hệ trực tiếp	- Mạng xã hội tạo điều kiện cho việc xây dựng mối quan hệ trực tuyến qua việc kết nối với người dùng khác trên toàn thế giới.
Bảo vệ quyền riêng tư	- Người dùng có quyền kiểm soát thông tin cá nhân của họ và quyết định mức độ chia sẻ với cộng đồng. - Mạng xã hội liên tục cập nhật và tăng cường các biện pháp bảo mật để bảo vệ thông tin cá nhân người dùng.
Tương tác hiệu quả	- Người dùng có thể cập nhật và chia sẻ thông tin hàng ngày, giữ liên lạc với bạn bè và gia đình. - Sự tương tác qua mạng xã hội không bị hạn chế bởi thời gian và địa lý, mang đến sự linh hoạt và thuận lợi.

Bảng 2. 1 Các tiện ích mang lại khi mạng xã hội ra đời

- Nhìn chung, mạng xã hội đã đưa ra giải pháp toàn diện, kết hợp nhiều khía cạnh của cuộc sống xã hội và cung cấp những tiện ích và cơ hội mới mẻ mà trước đó không có.

2.3 Nghiệp vụ của hệ thống:

2.3.1 Người dùng hệ thống:

- Người dùng (User) : Một người sử dụng được lưu trữ thông tin trong CSDL hệ thống. Mỗi người dùng được sử dụng email và mật khẩu để đăng nhập vào hệ thống.
- Người quản lý (Admin) : Một người có thể truy cập vào CSDL hệ thống, quản lý và xem thông tin user trên hệ thống. Mỗi admin sử dụng email và mật khẩu để đăng nhập vào quản lý hệ thống.

2.3.2 Yêu cầu chức năng:

- Quản lý người dùng, profile.
- Theo dõi, kết bạn với người dùng khác.
- Đăng bài viết, tương tác trên bài viết.
- Nhắn tin, upload hình ảnh, file.
- Gọi điện online.
- Quản lý các cuộc gọi tin nhắn.

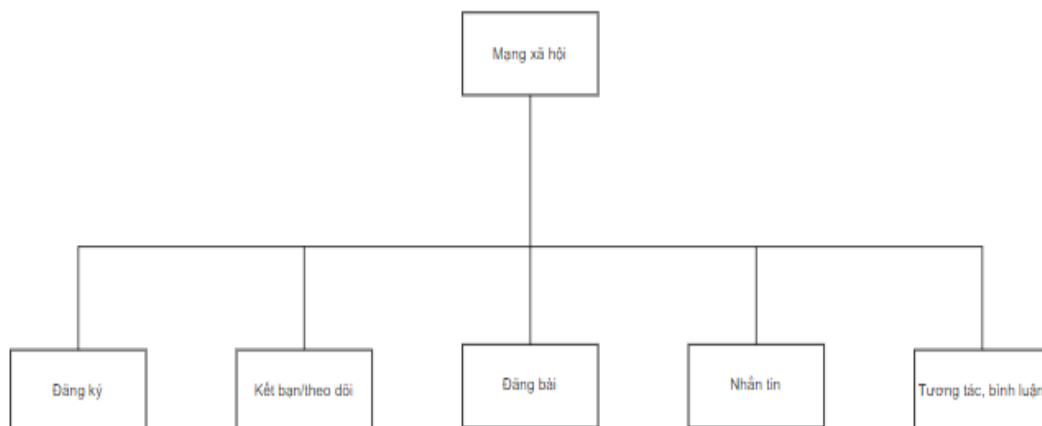
2.3.3 Yêu cầu phi chức năng:

TT	Nội dung	Mô tả chi tiết
1	Bảo mật	Mật khẩu của người dùng được mã hóa.
2	Giao diện thân thiện	Giao diện người dùng dễ sử dụng, trực quan.
3	Tương thích	Tương thích với nhiều trình duyệt phổ biến.

Bảng 2. 1. Yêu cầu phi chức năng

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Sơ đồ phân cấp chức năng:



Sơ đồ 3. 1 Sơ đồ phân rã chức năng mức đỉnh

3.1.1 Sơ đồ phân cấp chức năng mức đỉnh:

- Hệ thống nghiệp vụ được xét gồm các chức năng chính: Phục vụ khách, Pha chế, Quản lý kho, Quản lý sản phẩm, quản lý khu vực, quản lý nhân viên và tổng hợp báo cáo. Mỗi chức này được phân rã thành một số chức năng nhỏ hơn mà thao tác thực hiện nó đủ đơn giản cho việc lập trình.

3.1.2 Sơ đồ phân rã chức năng:

Chức năng quản lý tài khoản:

Mô tả chi tiết các chức năng:



Sơ đồ 3. 2 Sơ đồ phân rã chức năng quản lý tài khoản

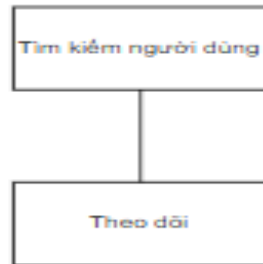
Tạo tài khoản : Tạo tài khoản cho những người dùng mới

Đổi thông tin cá nhân: Thay đổi thông tin cá nhân.

Đổi mật khẩu: Thay đổi mật khẩu.

Chức năng kết bạn:

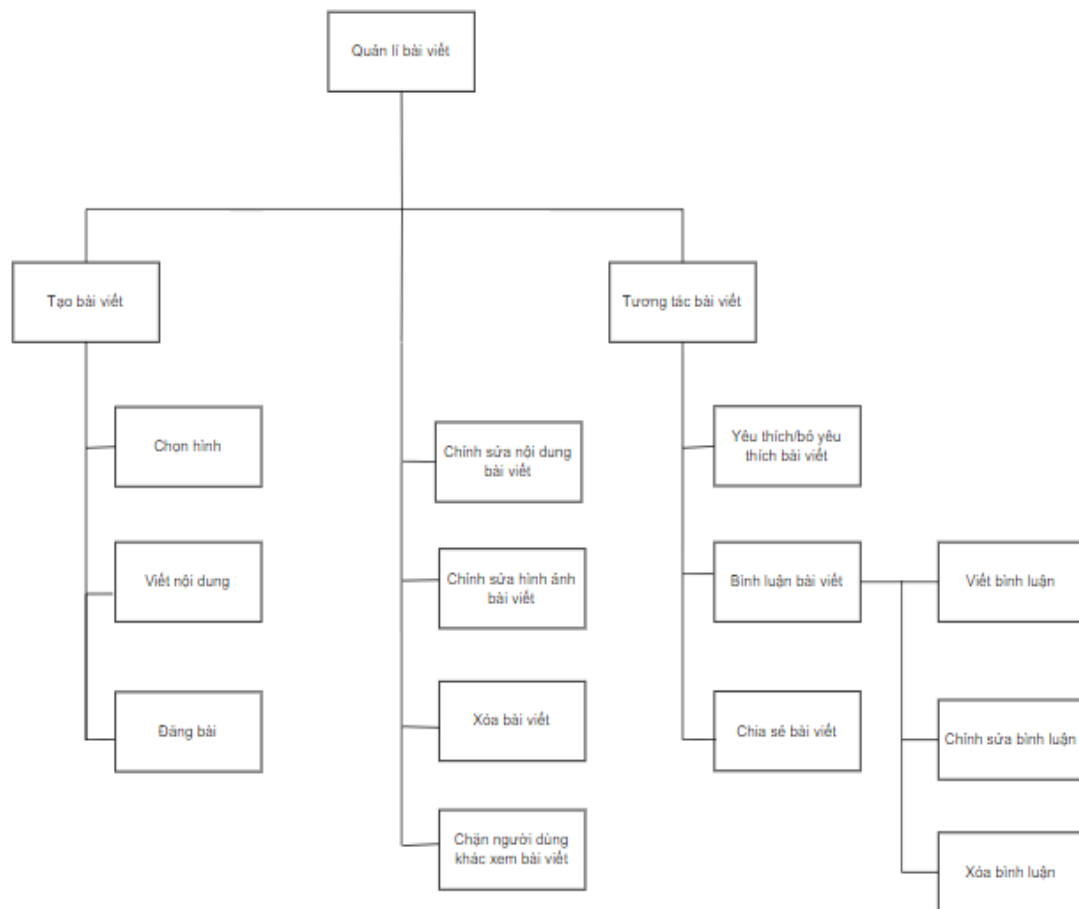
Mô tả chi tiết các chức năng:



Sơ đồ 3. 3 Sơ đồ phân rã chức năng kết bạn

Tìm kiếm người dùng: Tìm kiếm theo tên người dùng.

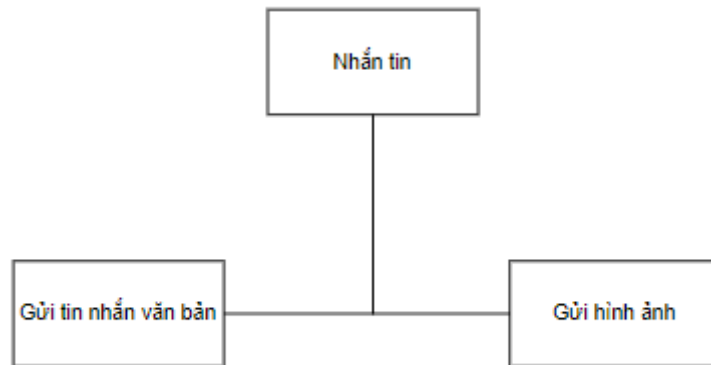
Theo dõi: theo dõi người dùng khác để có thể thấy bài viết, tương tác, nhấn tin,...

Quản lý bài viết:*Sơ đồ 3. 4 Sơ đồ phân rã chức năng quản lý bài viết*

Mô tả chi tiết các chức năng:

- Tạo bài viết: tạo bài viết với nội dung, hình ảnh để chia sẻ cho người khác những khoảnh khắc, cảm xúc của bản thân.
- Chỉnh sửa nội dung bài viết: sau khi tạo bài viết, nếu muốn thay đổi nội dung, người dùng có thể chỉnh sửa nội dung.
- Chỉnh sửa hình ảnh bài viết: sau khi tạo bài viết, nếu muốn thay đổi hình ảnh, người dùng có thể thêm hoặc xóa bớt hình ảnh.
- Xóa bài viết: xóa bài viết đã đăng.
- Yêu thích bài viết: bày tỏ cảm xúc đối với bài viết.
- Bình luận bài viết: bình luận, thảo luận về một bài viết bất kì.

Nhắn tin:



Sơ đồ 3. 5 Sơ đồ phân rã chức năng nhắn tin

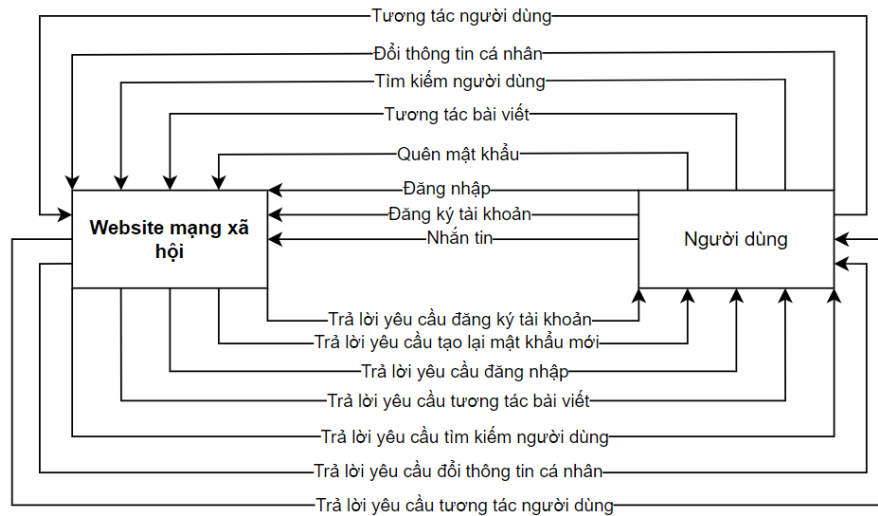
Mô tả chi tiết các chức năng:

Gửi tin nhắn văn bản: Người dùng thực hiện nhắn tin bằng văn bản.

Gửi hình ảnh: Người dùng thực hiện gửi hình ảnh trong tin nhắn riêng tư

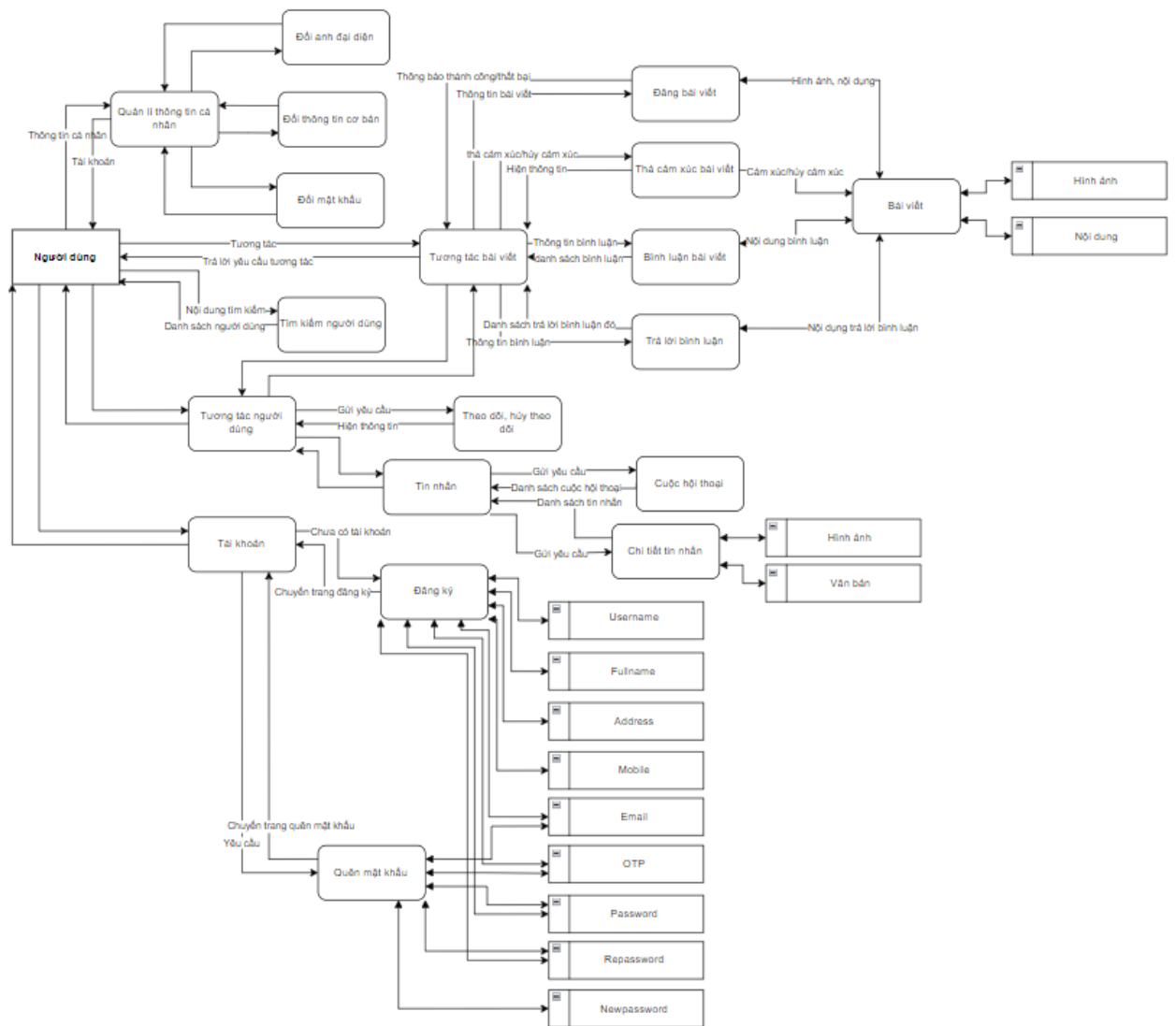
3.2 Mô hình luồng dữ liệu DFD (Data Flow Diagram)

3.2.1 Biểu đồ DFD mức ngữ cảnh:



Sơ đồ 3. 10 Biểu đồ luồng dữ liệu mức ngữ cảnh

3.2.2 Mô hình luồng dữ liệu DFD phân rã cấp 0 (chức năng chính):



Sơ đồ 3. 11 Sơ đồ luồng dữ liệu DFD phân rã cấp 0

3.3 Mô hình usecase:

3.3.1 Xác định Actor:

- Dựa vào yêu cầu của bài toán, ta có các actor: người dùng, khách.

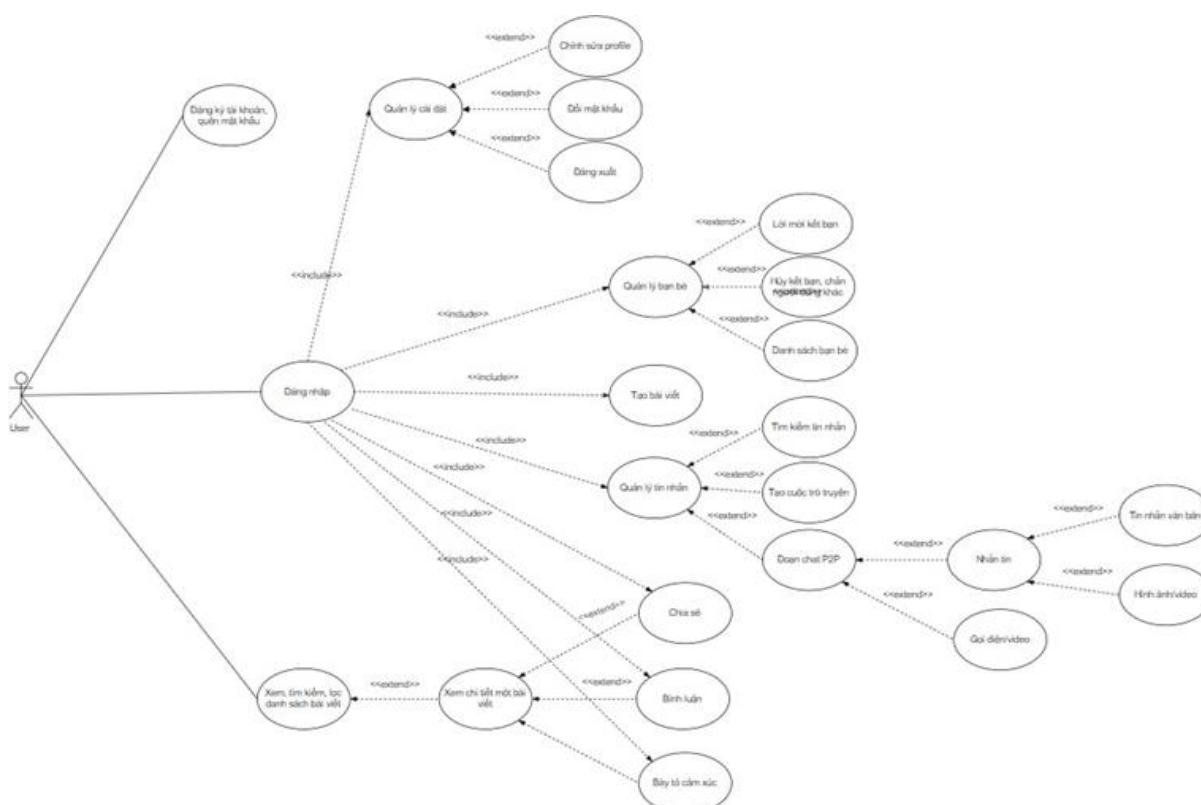
3.3.2 Xác định các Use Case:

- Từ yêu cầu chức năng ứng với từng actor ta có thể xác định được các use case như sau:

Actor	Use Case
Người dùng	Đăng nhập, đăng bài viết, chỉnh sửa bài viết, thích, bình luận trên bài viết, theo dõi bạn bè, nhấn tin, thay đổi thông tin các nhân, quên mật khẩu, ...
Khách	Đăng ký.

Bảng 3. 1 Xác định các use case

3.3.3 Xây dựng biểu đồ Use Case tổng quát:



Sơ đồ 3. 12 Biểu đồ đồ use case tổng quát

3.3.4 Đặc tả use case:**Thao tác dành cho người dùng đã đăng ký:****a. Đăng nhập:**

Mô tả: Chức năng cho phép người dùng đăng nhập vào hệ thống bằng cách cung cấp thông tin đăng nhập hợp lệ, để truy cập vào tài nguyên hệ thống.

Đặc tả:**Pre-condition:**

- Người dùng đã đăng ký tài khoản trong hệ thống và có thông tin đăng nhập hợp lệ.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng truy cập vào giao diện đăng nhập.
 - + Hệ thống hiển thị một giao diện đăng nhập yêu cầu người dùng cung cấp thông tin đăng nhập.
 - + Người dùng nhấn nút "Đăng nhập".
 - + Hệ thống kiểm tra thông tin đăng nhập có hợp lệ hay không.
 - + Nếu thông tin không hợp lệ (ví dụ: tên đăng nhập không tồn tại, mật khẩu không đúng), hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập.
 - + Nếu thông tin hợp lệ, hệ thống cho phép người dùng truy cập vào hệ thống và chuyển hướng tới trang hoặc giao diện chính của ứng dụng.
 - + Người dùng truy cập vào các tính năng và tài nguyên trong hệ thống.
- Ngoại lệ:
 - + Nếu người dùng không có tài khoản, họ cần thực hiện quy trình đăng ký trước khi có thể đăng nhập.
 - + Người dùng quên mật khẩu, nhấn vào nút quên mật khẩu để lấy lại mật khẩu.

b. Đăng xuất:

Mô tả: Chức năng cho phép người dùng đăng xuất khỏi hệ thống, kết thúc phiên làm việc hiện tại và không thể truy cập vào các tính năng và tài nguyên của hệ thống cho đến khi đăng nhập lại.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng truy cập vào giao diện hoặc tùy chọn "Đăng xuất" trang web.
 - + Hệ thống xác nhận ý định đăng xuất của người dùng xóa token của người dùng, đảm bảo không có thông tin được lưu trữ sau khi đăng xuất.
 - + Hệ thống chuyển hướng người dùng đến giao diện đăng nhập
 - + Người dùng không còn có quyền truy cập vào các tính năng và tài nguyên của hệ thống cho đến khi đăng nhập lại.

c. Nhắn tin:

Mô tả: Chức năng cho phép người dùng nhắn tin với người dùng khác.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn chức năng nhắn tin.
 - + Hệ thống hiển thị danh sách các cuộc trò chuyện.
 - + Người dùng chọn cuộc hội thoại để bắt đầu trò chuyện.
 - + Người dùng soạn tin nhắn văn bản/hình ảnh .
 - + Người dùng gửi tin nhắn.
- Dòng sự kiện khác:
 - + Nhấn “Hủy”, không thực hiện việc gộp bàn.

Pos-condition: Không có

Mở rộng: Không có.

d. Xem thông tin cá nhân người nhận tin:

Mô tả: Chức năng cho phép người dùng xem thông tin của người dùng khác.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang ở màn hình đoạn chat.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn chức năng nhắn tin
 - + Người dùng chọn cuộc trò chuyện.
 - + Người dùng click vào avatar người nhận tin nhắn.
 - + Hệ thống hiển thị profile người nhận tin nhắn.

Pos-condition: Không có

e. Chỉnh sửa thông tin cá nhân:

Mô tả: Chức năng cho phép người dùng chỉnh sửa thông tin cá nhân.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang ở trang cá nhân.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn chức năng chỉnh sửa thông tin cá nhân.
 - + Người dùng thay đổi thông tin.
 - + Người dùng xác nhận.

Pos-condition: Không có

f. Tạo bài viết:

Mô tả: Chức năng cho phép người dùng đăng bài viết.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang ở trang chủ.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn chức năng thêm bài viết.
 - + Người dùng chọn hình ảnh muốn đăng và mô tả.
 - + Người dùng chọn “Đăng” để hoàn tất.

Pos-condition: Không có

g. Sửa bài viết:

Mô tả: Chức năng cho phép người dùng sửa bài viết..

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang trang cá nhân của bản thân hoặc trang chủ.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn bài viết của mình.
 - + Người dùng chọn sửa.
 - + Người dùng thay đổi thông tin bài viết như nội dung, hình ảnh.

Pos-condition: Không có

h. Xóa bài viết:

Mô tả: Chức năng cho phép người dùng xóa bài viết..

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang trang cá nhân của bản thân hoặc trang chủ.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn bài viết của mình.
 - + Người dùng chọn xóa.
 - + Người dùng xác nhận.

Pos-condition: Không có

i. Thích bài viết:

Mô tả: Chức năng cho phép người dùng thích bài viết của mình hoặc người dùng khác.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang trang cá nhân của bản thân, bạn bè hoặc trang chủ.

Dòng sự kiện:

- Dòng sự kiện chính:

+ Người dùng chọn thích ở mỗi bài viết..

Pos-condition: Không có

j. Bỏ thích bài viết:

Mô tả: Chức năng cho phép người dùng bỏ thích bài viết mà mình đã thích.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang trang cá nhân của bản thân, bạn bè hoặc trang chủ.
- Người dùng đã thích bài viết

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn bỏ thích ở mỗi bài viết..

Pos-condition: Không có

k. Lấy lại mật khẩu:

Mô tả: Chức năng cho phép người dùng khôi phục mật khẩu của tài khoản đã đăng ký trong hệ thống bằng cách sử dụng liên kết được gửi qua email để nhập mật khẩu mới..

Đặc tả:

Pre-condition:

- Người dùng đã đăng ký tài khoản trong hệ thống và đã quên mật khẩu hiện tại.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng truy cập vào giao diện khôi phục mật khẩu của trang web.
 - + Hệ thống hiển thị một giao diện yêu cầu người dùng cung cấp địa chỉ email được liên kết với tài khoản.
 - + Người dùng nhập thông tin và nhấn nút "Gửi yêu cầu khôi phục mật khẩu".
 - + Hệ thống kiểm tra tính hợp lệ của thông tin và xác nhận rằng tài khoản đã được tìm thấy trong hệ thống.
 - + Nếu thông tin không hợp lệ (email không đúng), hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin.

- + Nếu thông tin hợp lệ, hệ thống tiếp tục quá trình khôi phục mật khẩu.
- + Hệ thống gửi OTP đến email của người dùng.
- + Người dùng truy cập vào email và xác thực OTP khôi phục mật khẩu.
- + Nếu không hợp lệ (ví dụ: đã hết hạn, không khớp với thông tin tài khoản), hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thực hiện lại quá trình khôi phục mật khẩu từ đầu.
- + Nếu hợp lệ, hệ thống chuyển hướng người dùng đến giao diện cho phép nhập mật khẩu mới.
- + Người dùng nhập mật khẩu mới và xác nhận mật khẩu.
- + Hệ thống lưu trữ mật khẩu mới và thông báo cho người dùng rằng mật khẩu đã được khôi phục thành công.
- + Người dùng có thể sử dụng mật khẩu mới để đăng nhập vào hệ thống.

Ngoại lệ:

- Nếu thông tin không hợp lệ hoặc quá trình khôi phục mật khẩu không thành công, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng thực hiện lại quá trình khôi phục mật khẩu.

1. Bình luận bài viết:

Mô tả: Chức năng cho phép người dùng bình luận bài viết của mình hoặc người dùng khác, hoặc trả lời một bình luận khác.

Đặc tả:

Pre-condition:

- Người dùng đã đăng nhập vào hệ thống.
- Người dùng đang trang cá nhân của bản thân, bạn bè hoặc trang chủ.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng chọn bình luận ở mỗi bài viết.
 - + Người dùng nhập nội dung.
 - + Người dùng chọn “Bình luận” để hoàn tất.

Pos-condition: Không có

Thao tác dành cho khách:

a. Đăng ký:

Mô tả: Usecase này cho phép người dùng đăng ký một tài khoản mới trong hệ thống bằng cách cung cấp thông tin cá nhân cần thiết.

Đặc Tả:

Pre-condition:

- Người dùng chưa có tài khoản trong hệ thống và cung cấp thông tin đăng ký hợp lệ.

Dòng sự kiện:

- Dòng sự kiện chính:
 - + Người dùng truy cập vào giao diện đăng ký của ứng dụng hoặc trang web..
 - + Hệ thống hiển thị một giao diện đăng ký yêu cầu người dùng cung cấp thông tin cá nhân.
 - + Người dùng nhập thông tin cá nhân bao gồm tên, email, mật khẩu và các thông tin khác theo yêu cầu của hệ thống.
 - + Hệ thống gửi OTP để xác nhận.
 - + Người dùng nhấn nút "Đăng ký".
 - + Hệ thống kiểm tra tính hợp lệ của thông tin đăng ký.
 - + Nếu thông tin không hợp lệ (ví dụ: email đã được sử dụng, tài khoản đã tồn tại), hệ thống hiển thị thông báo lỗi và yêu cầu người dùng điều chỉnh thông tin.
 - + Nếu thông tin hợp lệ, hệ thống tạo tài khoản mới và lưu trữ thông tin đăng ký.
 - + Sau khi xác thực thành công, người dùng có thể đăng nhập vào hệ thống bằng thông tin đăng nhập đã đăng ký

Ngoại lệ :

- Nếu tài khoản đã tồn tại hoặc thông tin đăng ký không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người dùng điều chỉnh thông tin.

Post-condition:

- Sau khi xác thực thành công, người dùng có thể đăng nhập vào hệ thống bằng thông tin đăng nhập đã đăng ký.

Mở rộng: Không có.

Không có.

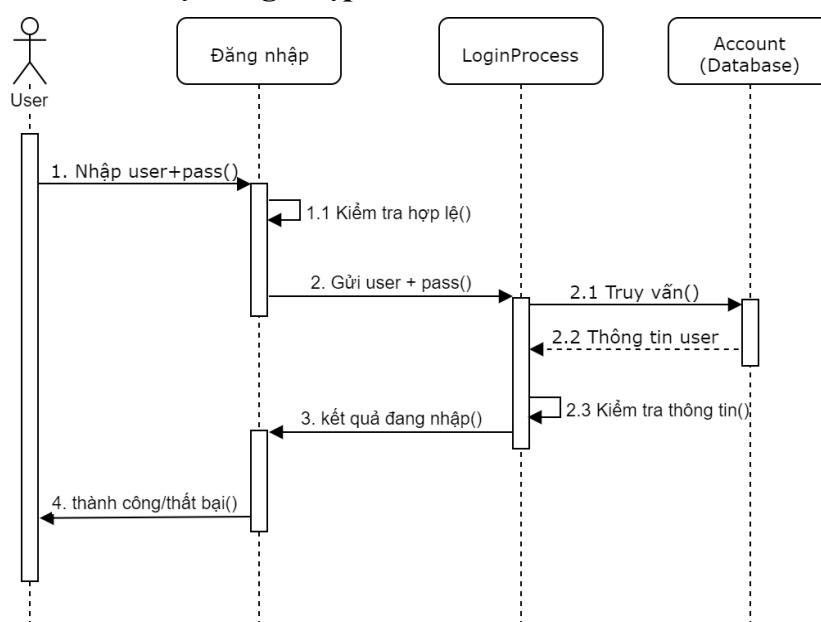
3.3.5 Biểu đồ tuần tự tương ứng với các Use Case:

➤ Use case đăng nhập:



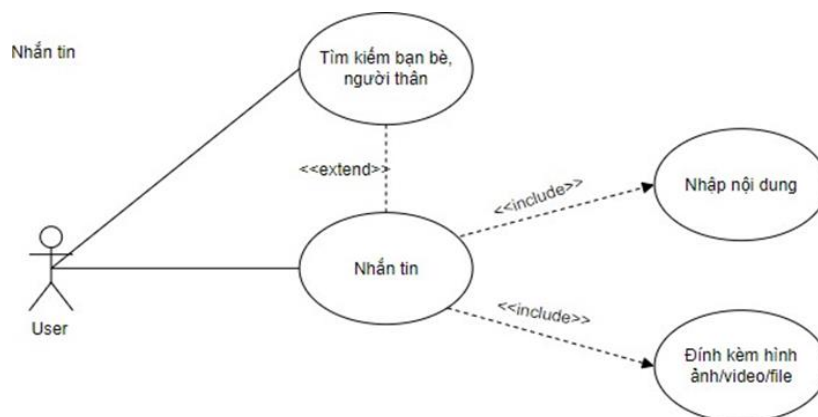
Sơ đồ 3. 16 Use case đăng nhập

Biểu đồ trình tự đăng nhập



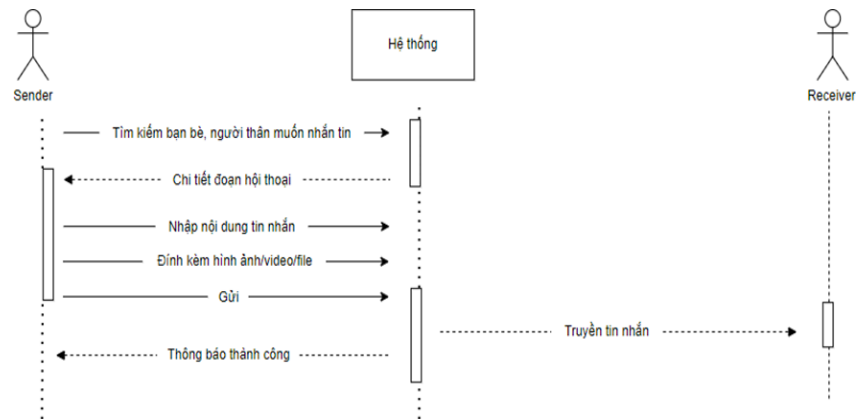
Sơ đồ 3. 17 Biểu đồ trình tự đăng nhập

➤ Use case nhắn tin:



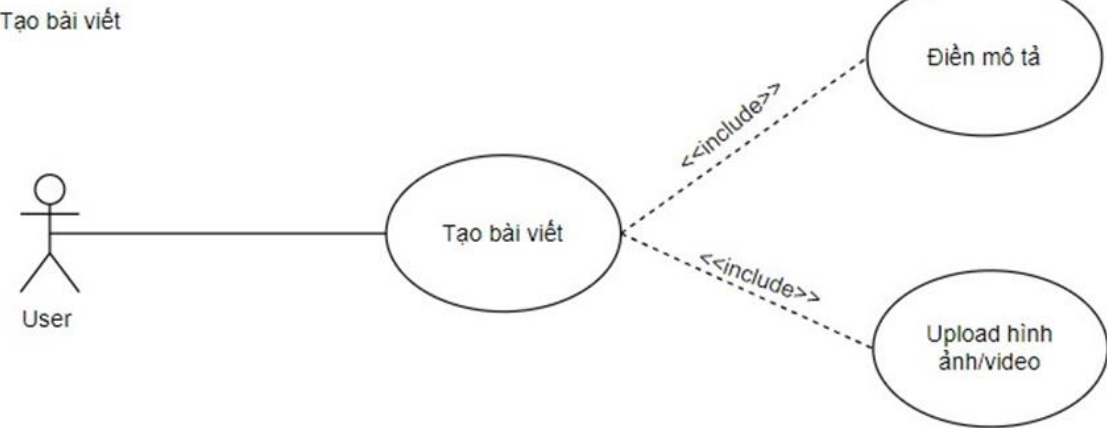
Sơ đồ 3. 18 Use case nhắn tin

Biểu đồ trình tự nhắn tin:



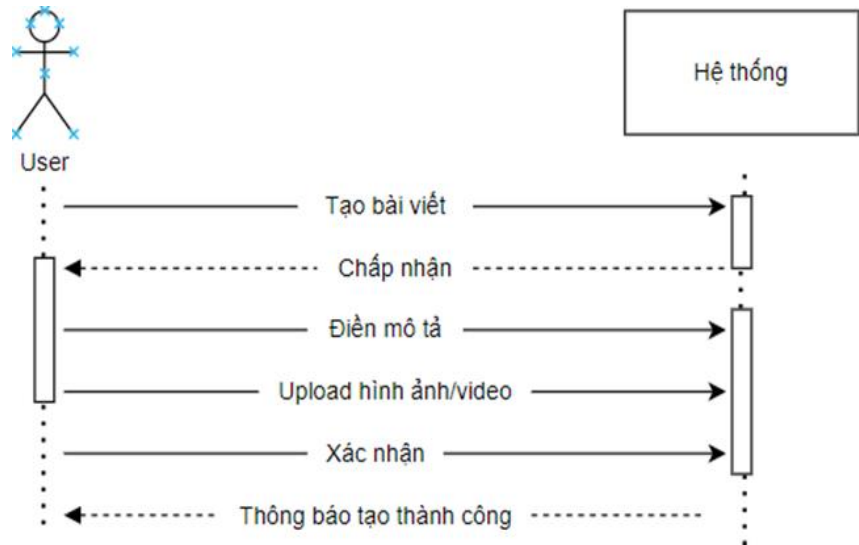
Sơ đồ 3. 19 Biểu đồ trình tự nhắn tin

➤ Use case tạo bài viết:



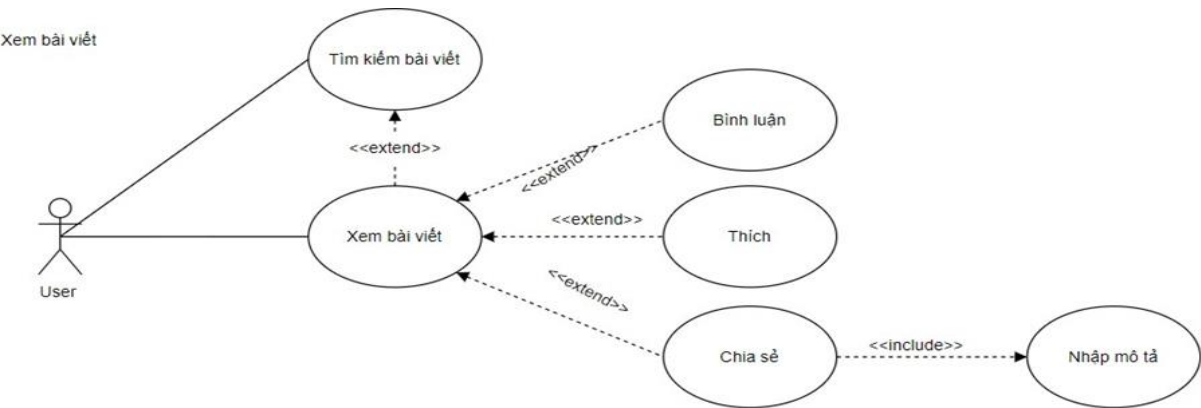
Sơ đồ 3. 21 Use case tạo bài viết

Biểu đồ trình tự tạo bài viết:



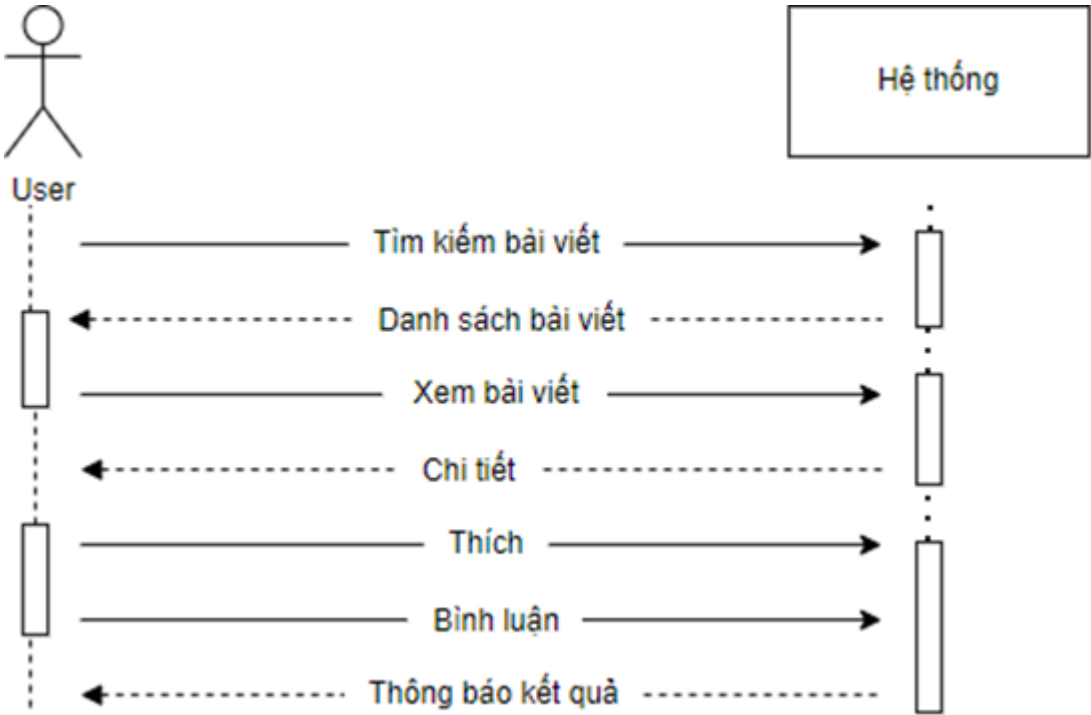
Sơ đồ 3. 22 Biểu đồ trình tự tạo bài viết

➤ Use case xem bài viết:



Sơ đồ 3. 24 Use case xem bài viết

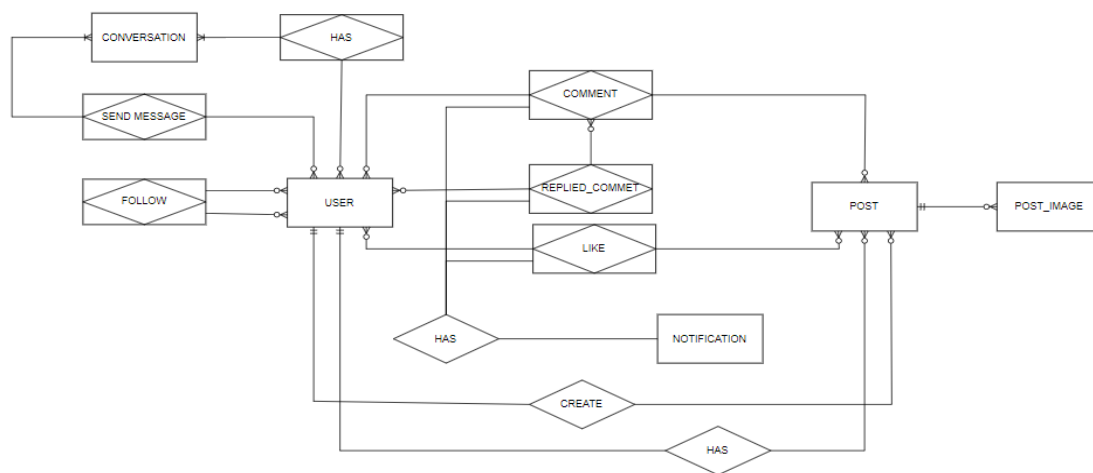
Biểu đồ trình tự tạo bài viết:



Sơ đồ 3. 25 Biểu đồ tuần tự thao tác bài viết

3.4 Xác định thực thể:

- USER(ID, FULLNAME, USERNAME, EMAIL, PASSWORD, GENDER, AVATAR, ADDRESS, DESCRIPTION).
- FOLLOWER(FOLLOWING_USER_ID, FOLLOWED_USER_ID).
- CONVERSATION(ID, TITLE).
- MESSAGE(ID, SEND_USER_ID, CONVERSATION_ID, TYPE, CREATEDAT).
- POST(ID, CREATED_BY, CAPTION).
- POST_IMAGE(ID, POST_ID, IMAGE).

3.5 Mô hình ERD:*Hình 3. 1 Mô hình ERD*

3.6 Mô hình cơ sở dữ liệu quan hệ:

Quy ước: **Primary key:** (gạch chân) MABP, **Foreign key :** (in đậm)

MABP PK_FK: (vừa thuộc khóa chính, vừa là khóa ngoại, gạch chân)

MAPD, MAHP

- USER(ID, FULLNAME, USERNAME, EMAIL, PASSWORD, GENDER, AVATAR, ADDRESS, DESCRIPTION).
- FOLLOWER(**FOLLOWING_USER_ID**, **FOLLOWED_USER_ID**).
- CONVERSATION(ID, TITLE).
- MESSAGE(ID, **SEND_USER_ID**, **CONVERSATION_ID**, TYPE, CONTENT, CREATEDAT)
- USER_CONVERSATION(ID, **USER_ID**, **CONVERSATION_ID**)
- POST(ID, **CREATED_BY**, CAPTION).
- POST_IMAGE(ID, **POST_ID**, IMAGE).
- LIKE(**USER_ID**, **POST_ID**).
- COMMENT(ID, **CREATED_BY**, **POST_ID**, CONTENT, **COMMENT_REPLIED_TO**, CREATEDAT).
- NOTIFICATION(ID, **USER_ID**, **POST_ID**, **R_USER_ID**, TYPE, CREATEDAT).
 - Thực thể USER: Thông tin user

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	id	int	Khóa chính	X
2	FULLNAME	Họ tên	varchar(100)		X
3	USERNAME	Tên đăng nhập	varchar(255)		X
4	EMAIL	Email	varchar(255)		X
5	PASSWORD	Mật khẩu	varchar(12)		X
6	GENDER	Giới tính	varchar(255)		X
7	AVATAR	Ảnh đại diện	double		X
8	ADDRESS	Địa chỉ	varchar(100)		
9	DESCRIPTION	Mô tả bản thân	varchar(100)		

Bảng 3. 2 Bảng thực thể USER

- Thực thể FOLLOWER: **FOLLOWING_USER_ID** + **FOLLOWED_USER_ID** = **Khóa chính**

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	FOLLOWING_USER_ID	Id User	int	Khóa ngoại	X

2	FOLLOWED_USER_ID	Id User follow	int	Khóa ngoại	X
---	------------------	----------------	-----	------------	---

Bảng 3. 3 Bảng thực thể FOLLOWER

- Thực thể LIKE: **USER_ID + POST_ID = Khóa chính**

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	USER_ID	id user	int	Khóa ngoại	X
2	POST_ID	id post	int	Khóa ngoại	X

Bảng 3. 4 Bảng thực thể LIKE

- Thực thể POST: Thông tin bài đăng

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	id	int	Khóa chính	X
2	CREATED_BY	id user	int	Khóa ngoại	X
3	CAPTION	Nội dung bài viết	varchar(max)		

Bảng 3. 5 Bảng thực thể Bills

- Thực thể POST_IMAGE: Hình của bài viết

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	Id	int	Khóa chính	X
2	POST_ID	id bài viết	varchar(100)	Khóa ngoại	X
3	IMAGE	Link hình ảnh	varchar(100)		X

Bảng 3. 6 Bảng thực thể POST_IMAGE

- Thực thể COMMENT:

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	id	int	Khóa chính	X
2	CREATED_BY	id user	int	Khóa ngoại	X
3	POST_ID	id post	int	Khóa ngoại	X
4	CONTENT	nội dung	varchar(max)		X
5	COMMENT_REPLIED_TO	id comment cha	int	Khóa ngoại	
6	CREATEDAT	thời gian tạo	datetime		X

Bảng 3. 7 Bảng thực thể COMMENT

- Thực thể CONVERSATION:

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	id	int	Khóa chính	X
2	TITLE		varchar(20)		

Bảng 3. 8 Bảng thực thể CONVERSATION

- Thực thể USER_CONVERSATION:

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	Id sản phẩm	int	Khóa chính	X
2	USER_ID	Id user	int	Khóa ngoại	X
3	CONVERSATION_ID	Id conversation	int	Khóa ngoại	X

Bảng 3. 9 Bảng thực thể USER_CONVERSATION

- Thực thể MESSAGE:

Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	Id danh mục sản phẩm	int	Khóa chính	X
2	SEND_USER_ID	Id user gửi	int	Khóa ngoại	X
3	CONVERSATION_ID	id đoạn hội thoại	int	Khóa ngoại	X

	ON_ID				
4	CONTENT	nội dung	varchar(max)		
5	TYPE	loại(text, image)	varchar(10)		
6	CREATEDAT	thời gian tạo	datetime		

Bảng 3. 10 Bảng thực thể MESSAGE

- Thực thể NOTIFICATION:

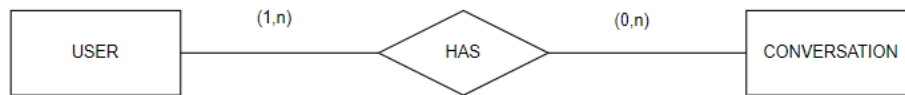
Stt	Thuộc tính	Mô tả	Kiểu dữ liệu	Khóa	Notnull
1	ID	id	int	Khóa chính	X
2	USER_ID	Id user thực hiện	int	Khóa ngoại	X
3	R_USER_ID	Id user nhận	int	Khóa ngoại	X
4	POST_ID	Id bài đăng	int	Khóa ngoại	X
5	TYPE	loại(like, comment, follow)	varchar(10)		X
6	CREATEDAT	thời gian tạo	datetime		X

Bảng 3. 11 Bảng thực thể Recipe

3.7 Mô hình dữ liệu:

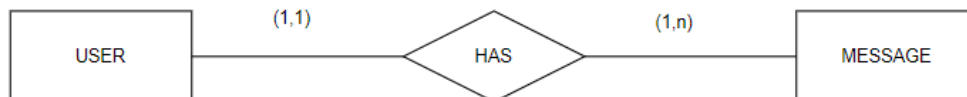
3.7.1 Xét USER– CONVERSATION:

Mỗi User có không hoặc nhiều đoạn hội thoại, một đoạn hội thoại thuộc một hoặc nhiều user.



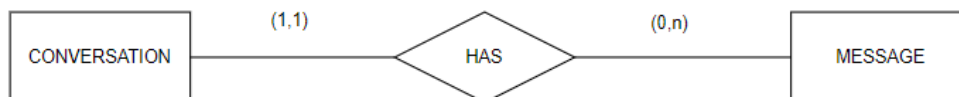
3.7.2 Xét USER– MESSAGE:

Mỗi User có không hoặc nhiều tin nhắn, một tin nhắn thuộc một user.



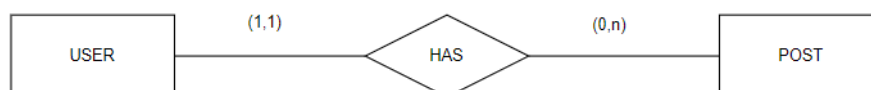
3.7.3 Xét MESSAGE– CONVERSATION:

Mỗi đoạn hội thoại có không hoặc nhiều tin nhắn, một tin nhắn thuộc một đoạn hội thoại.



3.7.4 Xét USER– POST:

Mỗi user có không hoặc nhiều bài đăng, mỗi bài đăng chỉ thuộc một user.



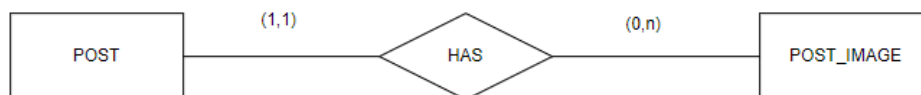
3.7.5 Xét USER– USER:

Mỗi user có không hoặc nhiều người theo dõi, mỗi user cũng theo dõi không hoặc nhiều user khác.



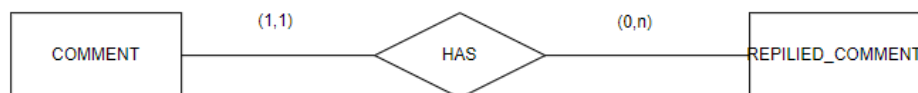
3.7.8 Xét POST– POST_IMAGE:

Mỗi bài đăng có không hoặc nhiều hình ảnh, hình ảnh chỉ thuộc về một bài đăng.

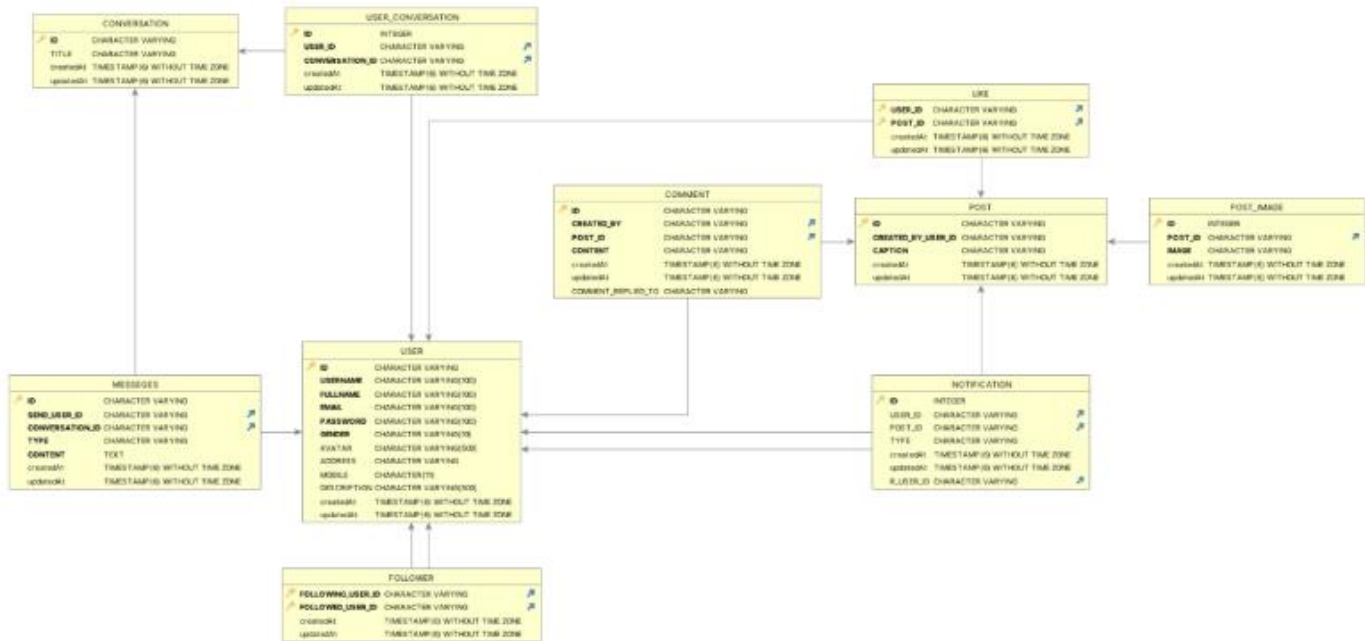


3.7.8 Xét COMMENT– COMMENT:

Mỗi comment có không hoặc nhiều comment con, mỗi comment con chỉ thuộc về một comment.



3.8 Phân tích và hoàn thiện mô hình quan hệ:



Hình 3. 2 Mô hình diagrams

CHƯƠNG 4. ỨNG DỤNG WEB APPLICATION

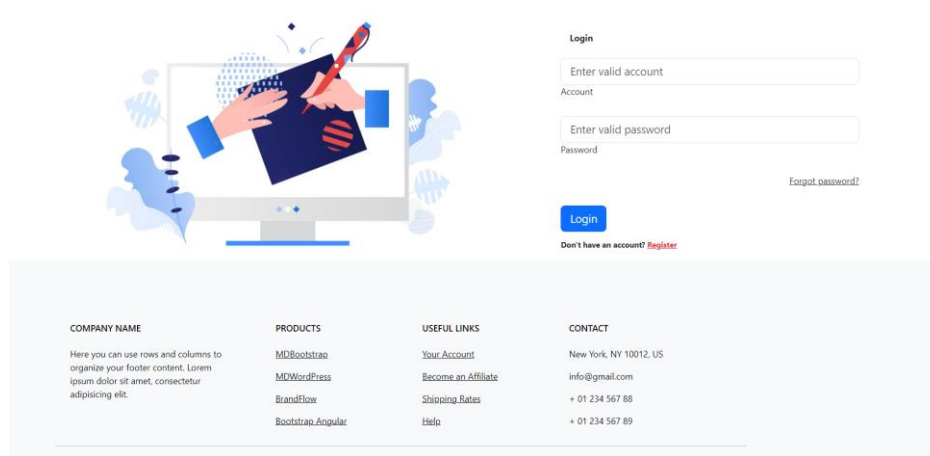
4.1 Yêu cầu hệ thống:

- Hệ quản trị cơ sở dữ liệu: ProgreSQL.
- Công cụ lập trình: Visual studio.
- Trình duyệt web: Chrome, Internet Explorer, Firefox,...

4.2 Demo giao diện:

4.2.1 Giao diện đăng nhập:

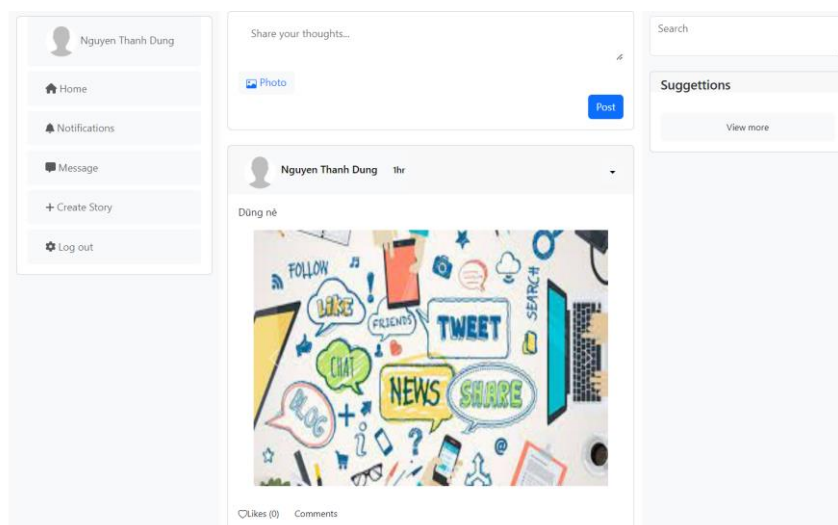
Giao diện được hiển thị đầu tiên khi người dùng vào trang đăng nhập.



Hình 4. 1 Giao diện đăng nhập

4.2.2 Giao diện trang chủ:

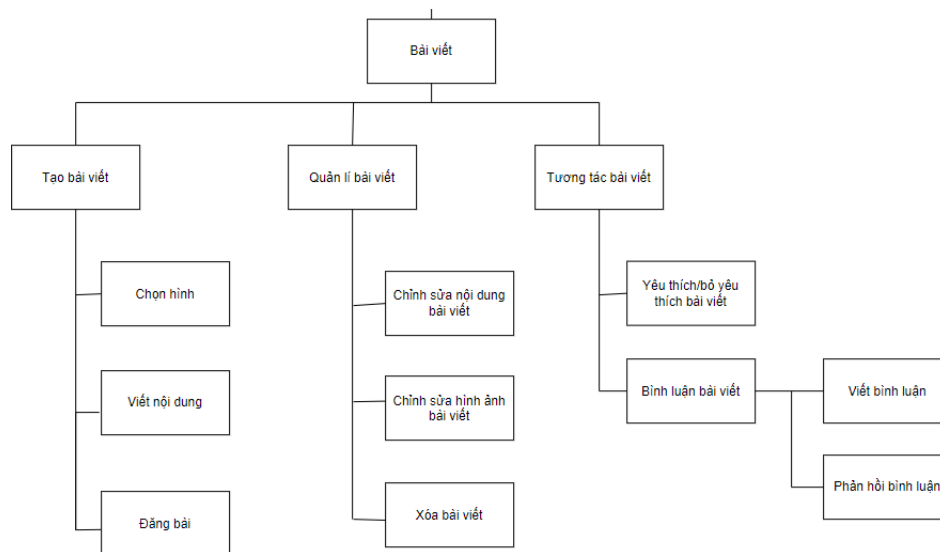
Sau khi đăng nhập thành công, hệ thống sẽ chuyển người dùng vào trang chủ:



Hình 4. 2 Giao diện trang chủ

4.2.3 Quản lí bài viết:

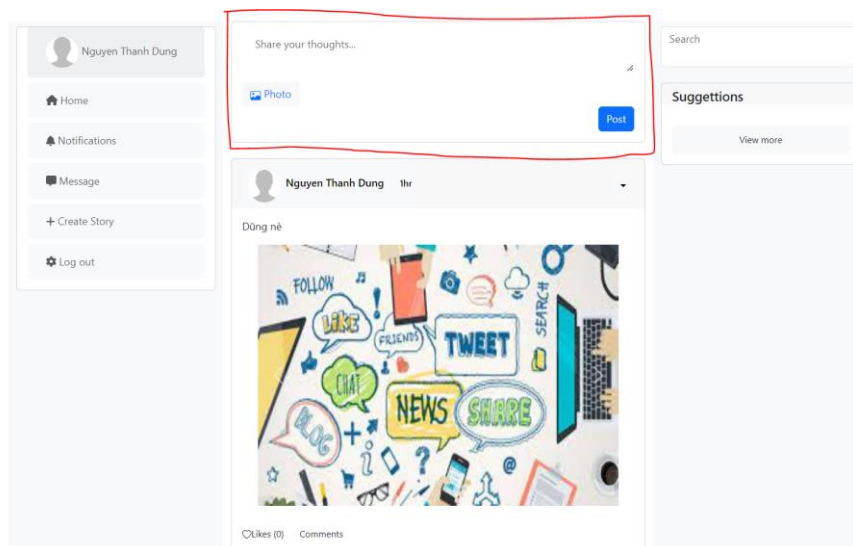
Sơ đồ quản lí bài viết:

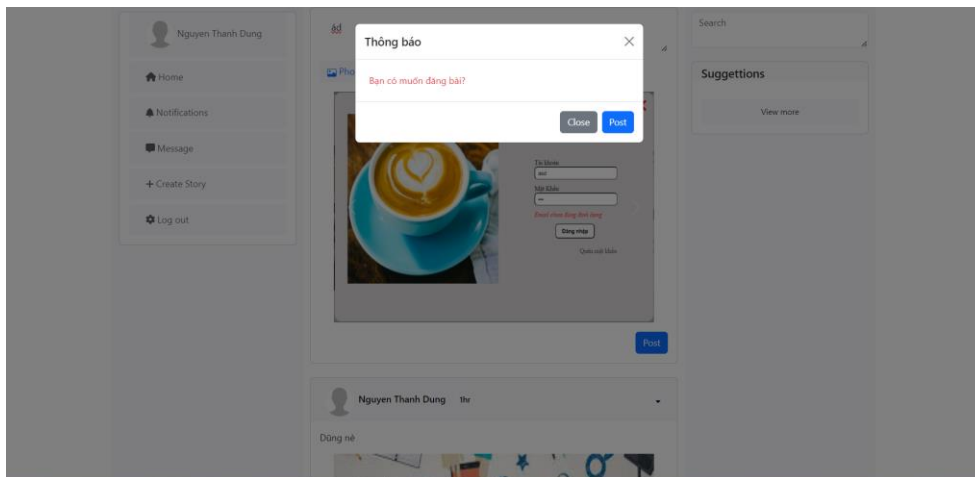


Sơ đồ 4. 1 Sơ đồ quản lí bài viết

Chức năng tạo bài viết:

Sau khi đăng nhập thành công, hệ thống sẽ chuyển người dùng vào trang chủ, trong trang chủ có phần đăng bài viết ở phía trên chính giữa màn hình phần khung đỏ.





Hình 4. 3 Giao diện màn đăng bài viết

Để đăng bài viết nào đó, ta điền nội dung bài viết có thể là chuỗi chữ, có thể là hình ảnh hoặc có thể là cả hai

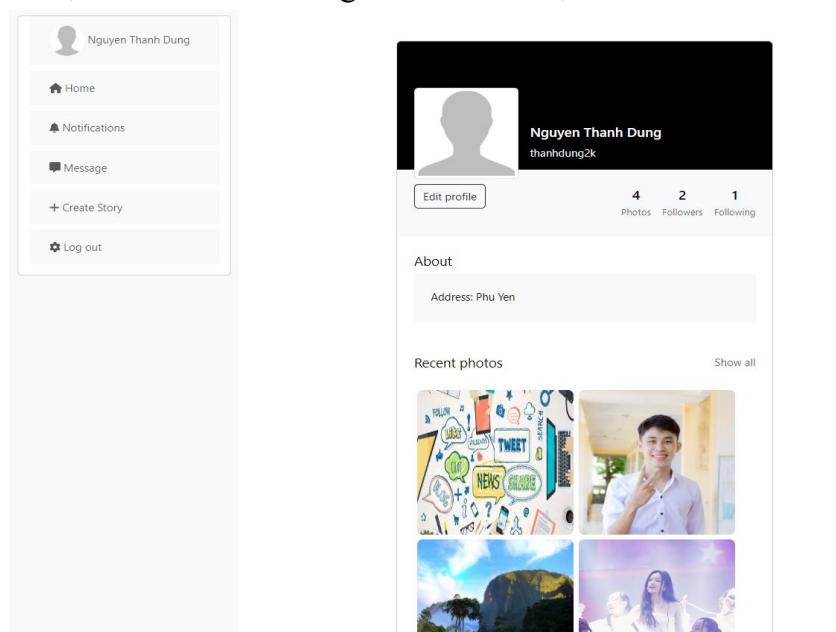
Sau khi chọn ảnh và viết nội dung bài viết, hình ảnh được chọn và nội dung sẽ hiện thị:

Giao diện bao gồm:

- **Hình ảnh:** danh sách hình ảnh đã chọn.
 - **Nội dung bài viết:** Chuỗi kí tự người dùng nhập.
- Khi click “Post”, bài viết sẽ được đăng lên trang chủ:

Chức năng chỉnh sửa bài viết:

Người dùng vào trang cá nhân, có danh sách bài viết, người dùng chọn bài viết để chỉnh sửa, có thể sửa nội dung hoặc hình ảnh, hoặc cả hai.



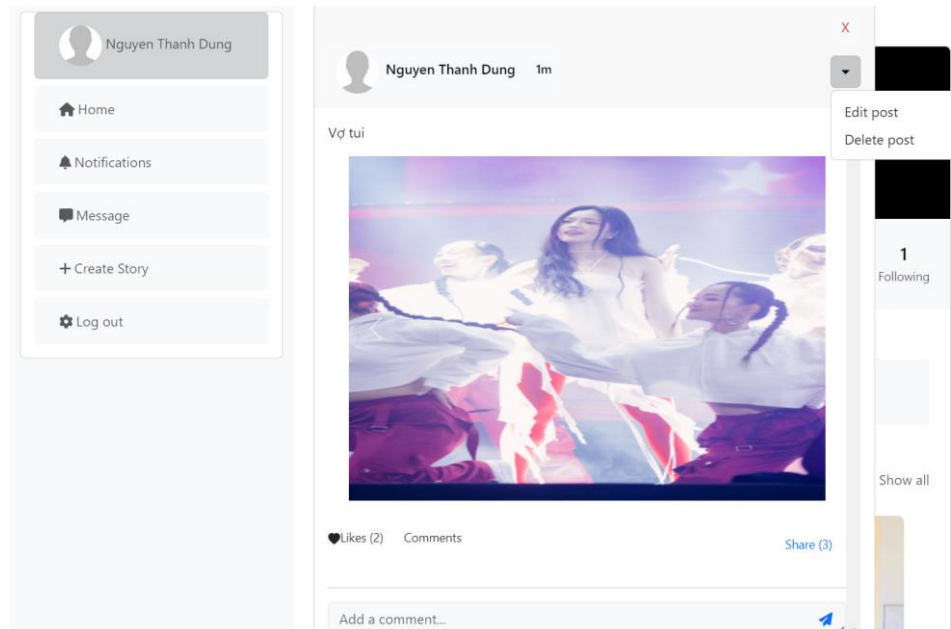
Hình 4. 4 Giao diện trang cá nhân người dùng

Thao tác:

Người dùng chọn 1 bài viết bất kì để xem chi tiết bài viết.

Hệ thống:

Hệ thống sẽ tự động hiển thị chi tiết bài viết người dùng chọn.



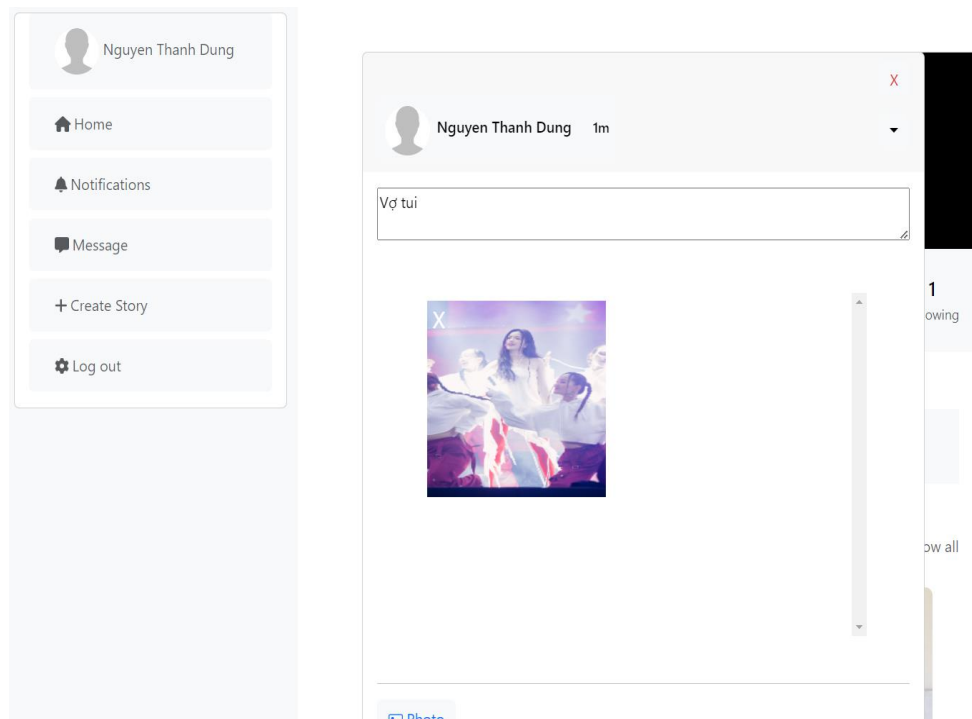
Hình 4. 5 Giao diện chi tiết bài viết (chỉnh sửa)

Thao tác:

Người dùng chọn nút “edit post” để chỉnh sửa bài viết.

Hệ thống:

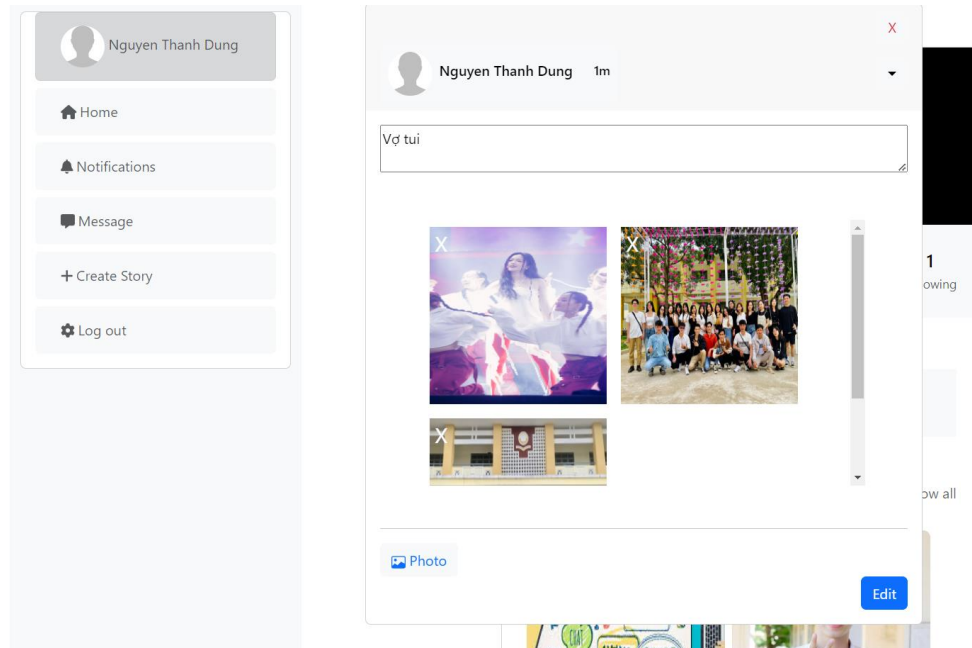
Hệ thống sẽ chuyển người dùng về form chỉnh sửa bài viết.



Hình 4. 6 Giao diện chỉnh sửa bài viết

Giao diện:

Gồm có nội dung và hình ảnh bài viết, người dùng có thể chỉnh sửa nội dung, chỉnh sửa ảnh, thêm ảnh, xóa ảnh trong danh sách ảnh.



Hình 4. 7 Giao diện chỉnh sửa bài viết

Giao diện:

Gồm có nội dung và hình ảnh bài viết.

Thao tác:

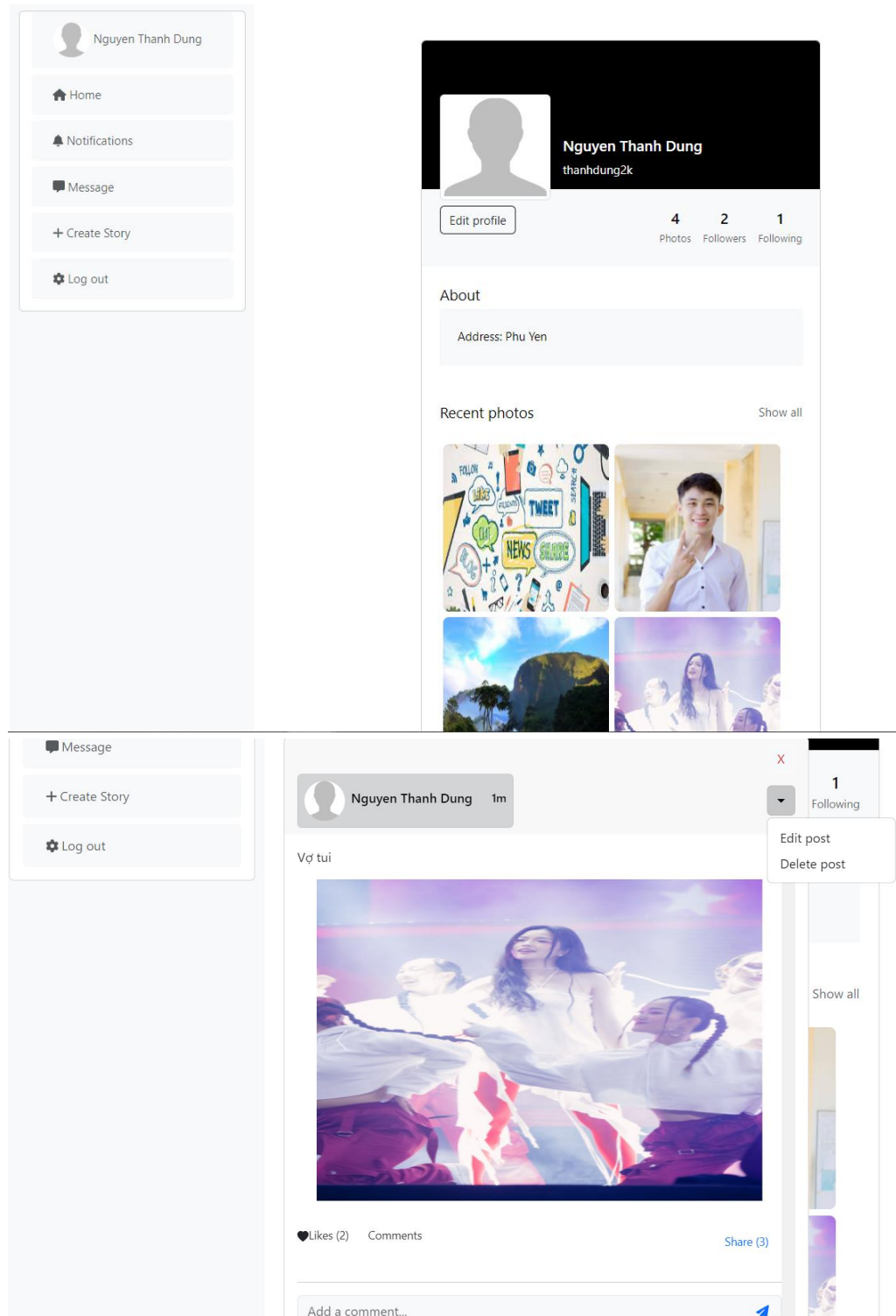
Sau khi chỉnh sửa theo ý muốn, người dùng chọn “Edit” để chỉnh sửa cập nhật bài viết.

Hệ thống:

Hệ thống sẽ cập nhật theo ý muốn của người dùng. Thông báo cho người dùng biết cập nhật thành công hay thất bại.

Chức năng xóa bài viết:

Người dùng vào trang cá nhân, có danh sách bài viết, người dùng chọn bài viết để xóa.



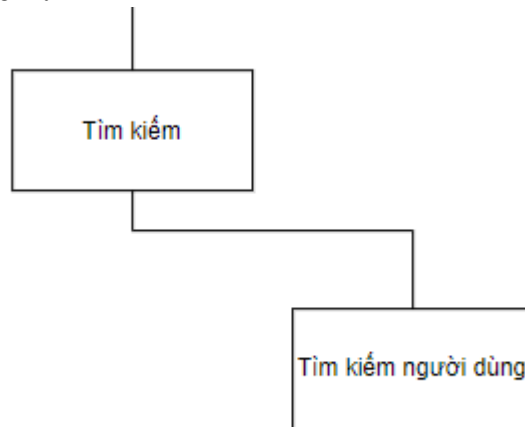
Hình 4. 8 Giao diện chi tiết bài viết(xóa)

Thao tác:

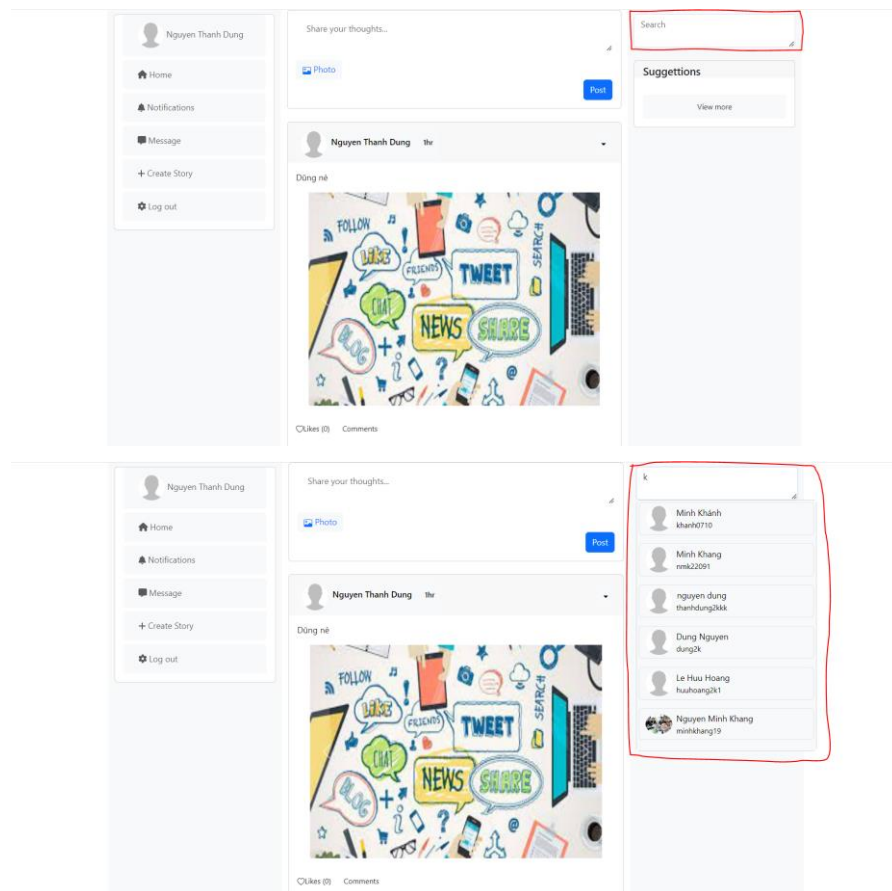
Người dùng chọn "Delete post" để xóa bài viết. Hệ thống sẽ xóa bài viết đó.

Hệ thống:

Hệ thống sẽ tự động xóa bài viết đó, thông báo cho người dùng xóa thành công hay thất bại.

4.2.4 Chức năng tìm kiếm:**Sơ đồ tìm kiếm:***Sơ đồ 4. 2 Sơ đồ chức năng tìm kiếm*

Trong giao diện trang chủ có phần tìm kiếm bên phải phía trên (phần tô đỏ).



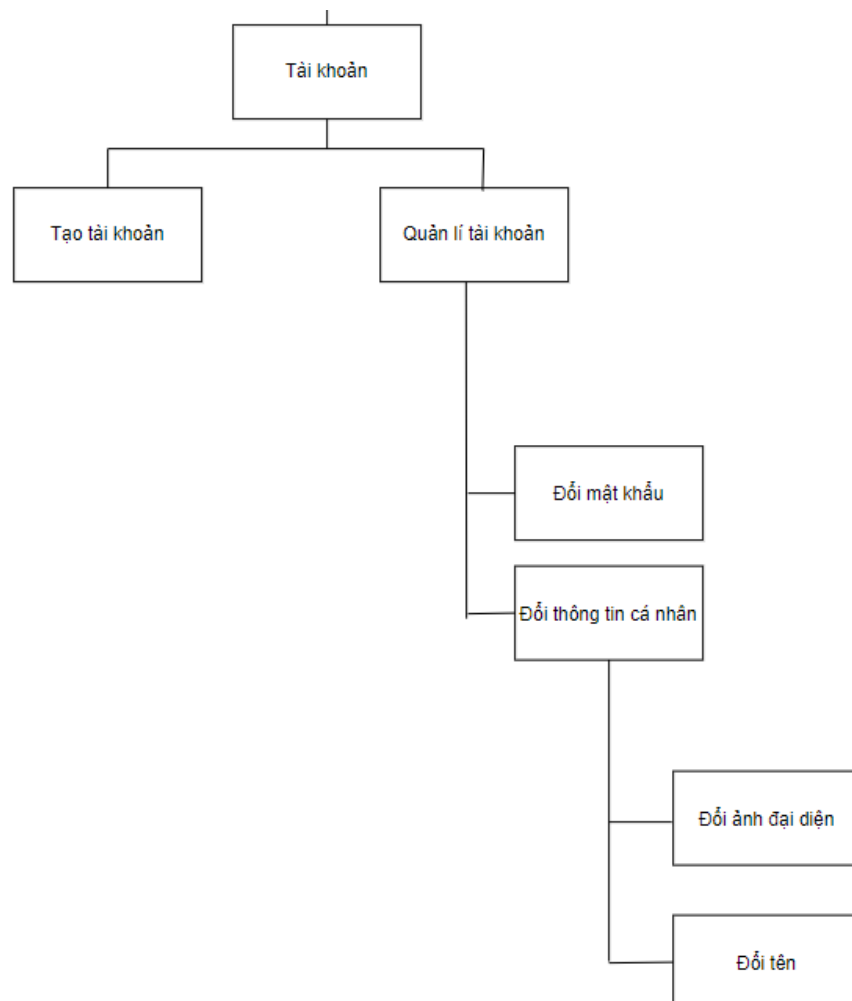
Hình 4. 9 Giao diện tìm kiếm

Thao tác: Người dùng nhập vào ô tìm kiếm, tìm kiếm theo username người dùng khác.

Hệ thống: Lấy dữ liệu danh sách từ database có username chứa kí tự mà người dùng đã nhập, đưa lên màn hình cho người dùng.

4.2.5 Quản lý tài khoản:


Sơ đồ quản lý tài khoản:



Sơ đồ 4. 3 Sơ đồ quản lý tài khoản


Chức năng tạo tài khoản:

Giao diện:




Register

Email

Hình 4. 10 Giao diện đăng ký email

Register

OTP

Hình 4. 11 Giao diện gửi OTP đăng ký

Register

User Name

Full Name

Mobile

Address

PassWord

RePassWord

*Hình 4. 12 Giao diện điền thông tin cá nhân đăng ký***Thao tác:**

Người dùng nhập các trường yêu cầu của hệ thống. Sau khi nhập email, nhận OTP qua email và nhập OTP trong form nhập OTP, nhập thông tin cá nhân và đăng ký.

Hệ thống:

Hệ thống kiểm tra email đã đăng ký chưa, nếu đã đăng ký thì thông báo cho người dùng, nếu chưa sẽ gửi OTP qua email đăng ký.

Hệ thống kiểm tra OTP đã nhập, nếu sai thông báo cho người dùng, nếu đúng chuyển sang form điền thông tin cá nhân.

Hệ thống kiểm tra các trường nhập thông tin cá nhân, nếu chưa điền các trường bắt buộc thì sẽ thông báo cho người dùng. Sau khi điền xong, nếu nhập thông tin username đã tồn tại thì hệ thống sẽ thông báo cho người dùng. Nếu username hợp lệ sẽ thông báo đăng ký thành công và chuyển về trang đăng nhập.

Chức năng quản lý tài khoản:

Giao diện:

Nguyễn Thanh Dung

Home

Notifications

Message

+ Create Story

Log out

Infomation

User name

thanhdung2k

Full name

Nguyễn Thanh Dung

Mobile

0343263672

Address

Phu Yen

Descriptions

Descriptions...

Submit

Security

Password

Current password...

New password

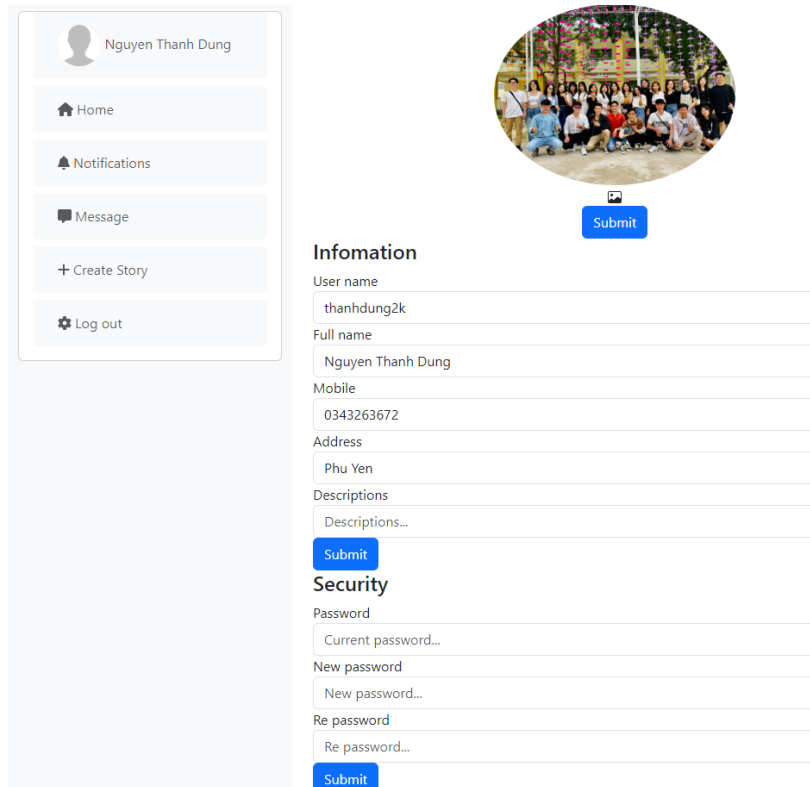
New password...

Re password

Re password...

Submit

Hình 4. 13 Giao diện chỉnh sửa thông tin cá nhân

Đổi ảnh đại diện:**Giao diện sau khi thao tác:**

Nguyen Thanh Dung

Home

Notifications

Message

+ Create Story

Log out

Information

User name
thanhdung2k

Full name
Nguyen Thanh Dung

Mobile
0343263672

Address
Phu Yen

Descriptions
Descriptions...

Submit

Security

Password
Current password...

New password
New password...

Re password
Re password...

Submit

Hình 4. 14 Giao diện sau khi chỉnh sửa ảnh đại diện

Thao tác:

Người dùng chọn biểu tượng ảnh đại diện. Chọn ảnh và xác nhận đổi.

Hệ thống:

Hệ thống cập nhập ảnh đại diện cho người dùng người, và thông báo cập nhập thành công hay thất bại.

Đổi thông tin cá nhân:

Thao tác: người dùng nhập các trường phần “Infomation”, điền các trường bắt buộc điền, sau đó nhấn “Submit”.

Hệ thống: Hệ thống sẽ cập nhật thông tin cá nhân người dùng. Nếu trùng username với người dùng khác thì hệ thống sẽ thông báo cho người dùng, nếu thỏa, hệ thống sẽ tự cập nhật thông tin cá nhân và thông báo thành công cho người dùng.

Đổi mật khẩu:

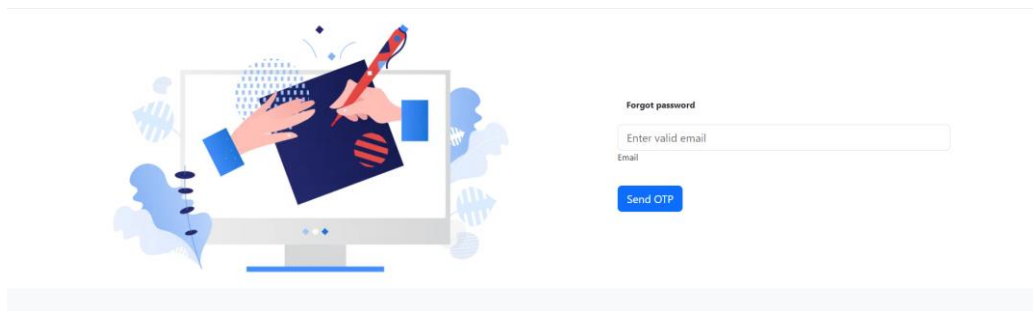
Thao tác: Người dùng nhập các trường phần “Security”, điền các

trường bắt buộc và sau đó nhấn “Submit”.

Hệ thống: Hệ thống sẽ tự động bắt lỗi (mật khẩu mới và nhập lại mật khẩu mới không trùng khớp, chưa nhập các trường phân đổi mật khẩu, mật khẩu cũ không trùng khớp) và thông báo cho người dùng. Sau khi nhập xong, hệ thống sẽ thông báo cho người dùng biết đã cập nhật thành công hay thất bại.

Quên mật khẩu:

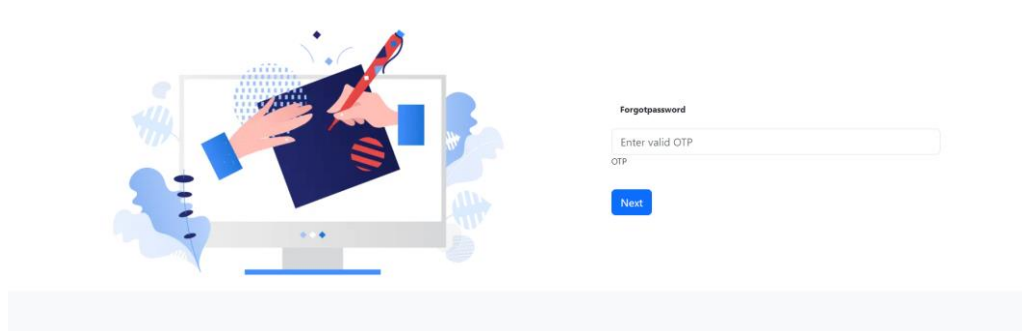
Giao diện:



Hình 4. 15 Giao diện điền email quên mật khẩu

Thao tác: người dùng nhập email đã đăng ký tài khoản. Sau đó nhấn “SendOTP”.

Hệ thống: Hệ thống sẽ kiểm tra email đã nhập đã tồn tại chưa, nếu chưa sẽ thông báo cho người dùng, nếu đã tồn tại thì hệ thống sẽ gửi OTP qua email đã nhập và chuyển sang form nhập OTP.

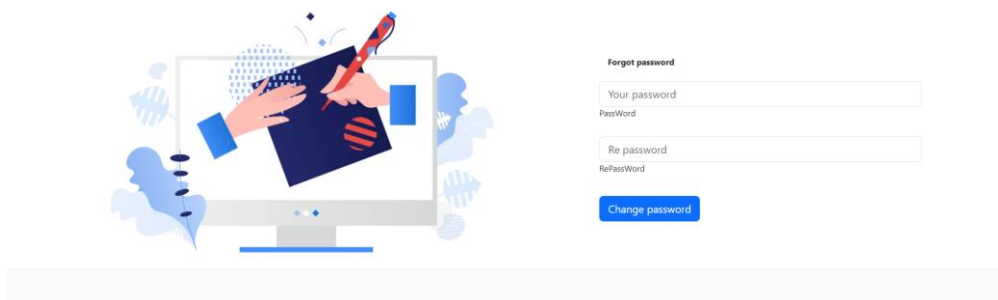


Hình 4. 16 Giao diện điền OTP quên mật khẩu

Thao tác: người dùng nhập OTP đã nhận qua email đã nhập trước đó, sau đó nhấn “Next”.

Hệ thống: hệ thống sẽ kiểm tra OTP đã nhập đủ kí tự hoặc có đúng với OTP hệ thống gửi mới nhất cho người dùng, hệ thống sẽ thông báo lỗi cho người dùng. Nếu hợp lệ, hệ thống sẽ tự động chuyển sang

form tạo lại mật khẩu mới.



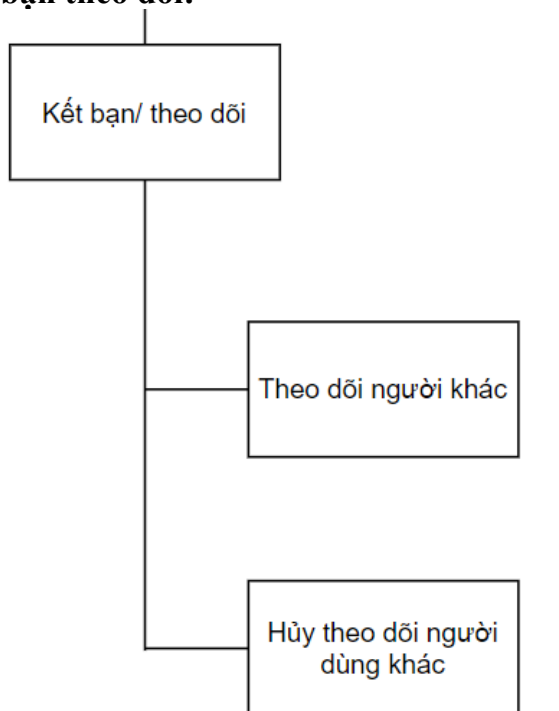
Hình 4.17 Giao diện điện lại mật khẩu mới

Thao tác: Người dùng nhập mật khẩu mới và nhập lại mật khẩu mới, sau đó nhấn “Change password”.

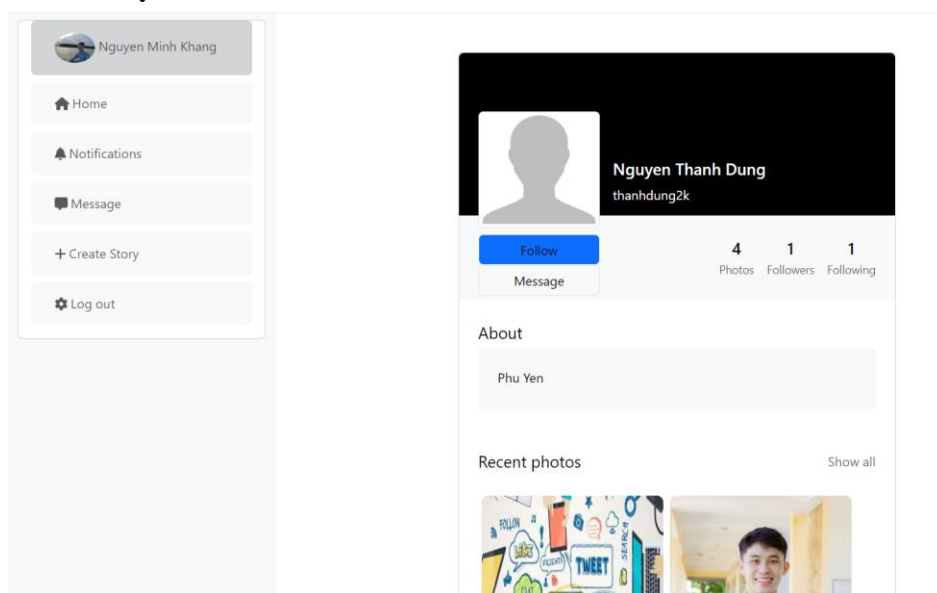
Hệ thống: Hệ thống sẽ kiểm tra người dùng đã nhập đầy đủ và chính xác, nếu chưa hệ thống sẽ thông báo cho người dùng biết. Nếu hợp lệ, hệ thống sẽ thông báo cho người dùng biết cập nhật thành công hay thất bại.

4.2.6 Kết bạn/theo dõi:

Sơ đồ kết bạn theo dõi:



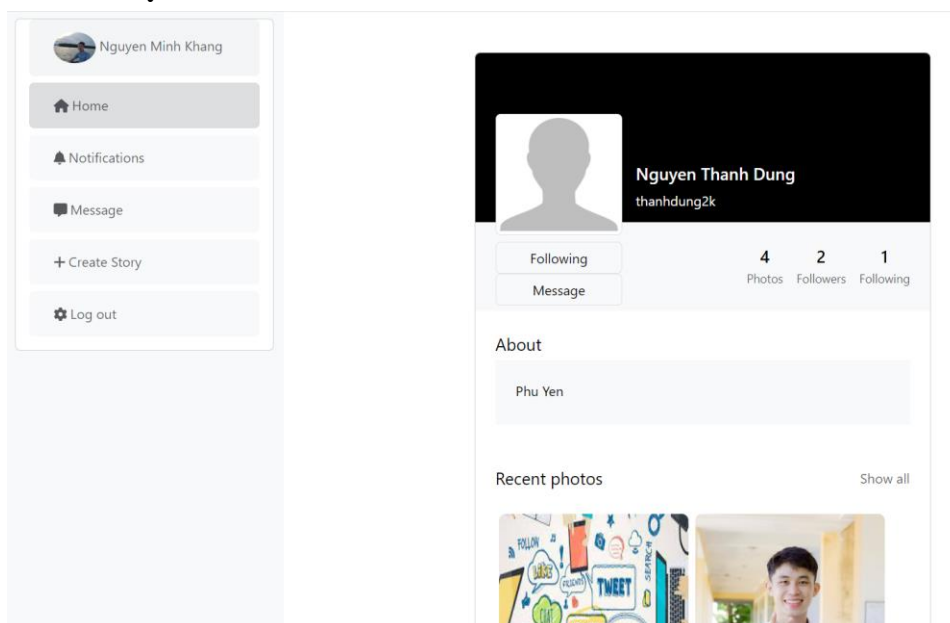
Sơ đồ 4.4 Sơ đồ chức năng kết bạn theo dõi

Theo dõi người khác:**Giao diện:**

Hình 4. 18 Giao diện trang cá nhân người dùng khác chưa theo dõi

Thao tác: Người dùng nhấn nút “Follow” màu xanh trên trang cá nhân của người dùng khác.

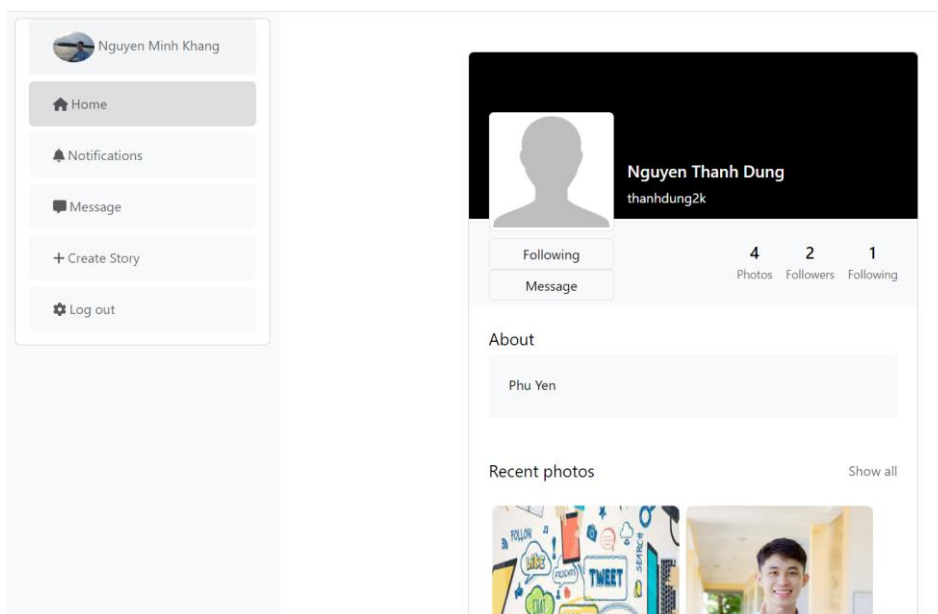
Hệ thống: Hệ thống sẽ cập nhật trạng thái “Follow” thành “Following” và mình có thể xem các bài viết của người dùng vừa theo dõi trên trang chủ.

Giao diện sau khi theo dõi:

Hình 4. 19 Giao diện trang cá nhân người dùng khác đã theo dõi

Chức năng hủy theo dõi: Có chức năng hủy theo dõi người dùng khác.

Giao diện:

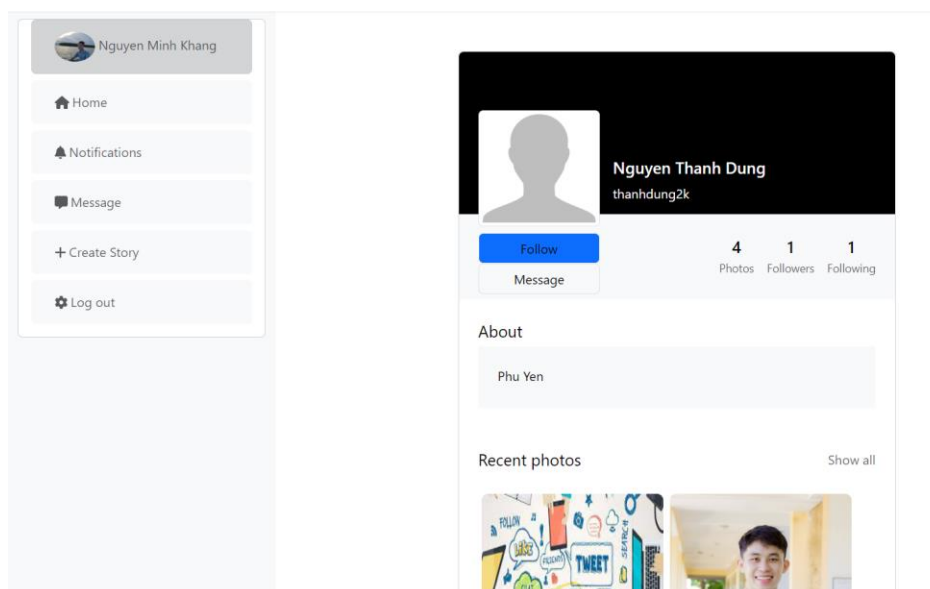


Hình 4. 20 Giao diện đã theo dõi người dùng khác.

Thao tác: Người dùng nhấn nút “Following” để hủy theo dõi người dùng khác.

Hệ thống: Hệ thống sẽ cập nhật lại trạng thái “Following” thành “Follow” và người dùng không thể xem các bài viết của người dùng vừa hủy theo dõi trên trang chủ.

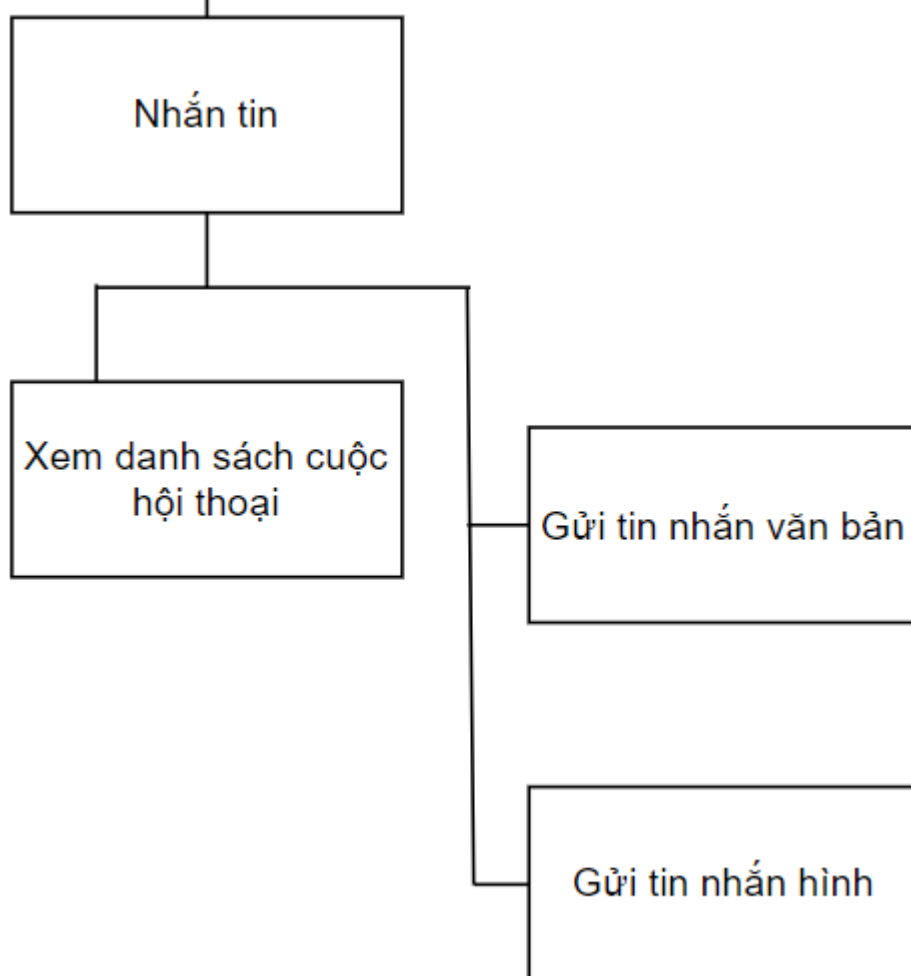
Giao diện sau khi hủy theo dõi:



Hình 4. 21 Giao diện hủy theo dõi người dùng khác.

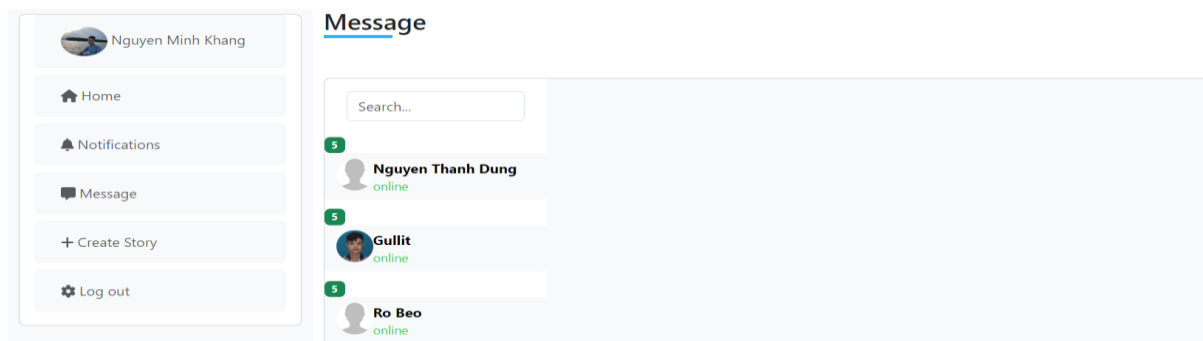
4.2.7 Quản lý tin nhắn:

Sơ đồ quản lý tin nhắn:



Sơ đồ 4. 5 Sơ đồ chức năng quản lý tin nhắn

Giao diện:



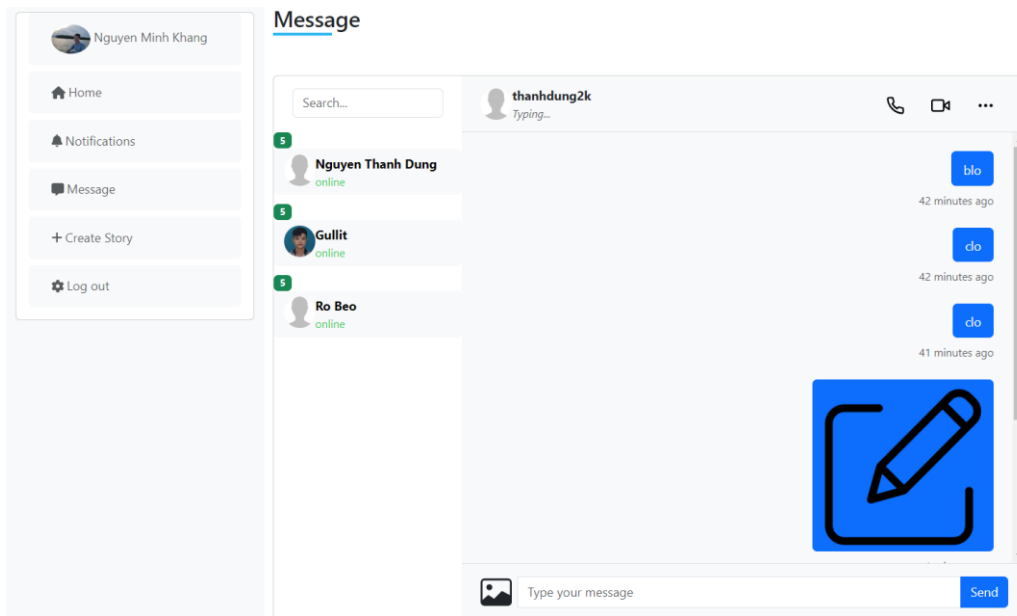
Hình 4. 22 Giao diện xem danh sách cuộc hội thoại

Thao tác: Người dùng vào trang tin nhắn, tiêu đề “Message” sẽ có danh sách các cuộc hội thoại của người dùng đó.

Hệ thống: Hệ thống sẽ đưa ra danh sách các cuộc hội thoại của người đó.

Chức năng xem tin nhắn, nhắn tin, gửi hình:

Giao diện xem chi tiết tin nhắn:

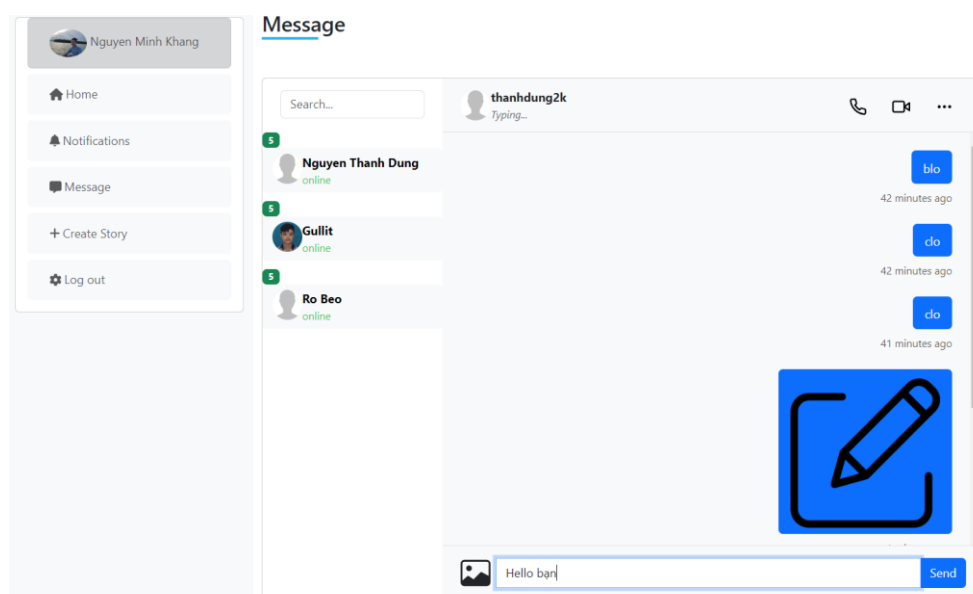


Hình 4. 23 Giao diện xem chi tiết tin nhắn

Thao tác: Người dùng nhấn vào bất kỳ cuộc hội thoại nào để xem các tin nhắn trong cuộc hội thoại đó.

Hệ thống: Hệ thống sẽ tự lấy danh sách tin nhắn của cuộc hội thoại đó và hiện lên màn hình cho người dùng.

Giao diện nhắn tin:

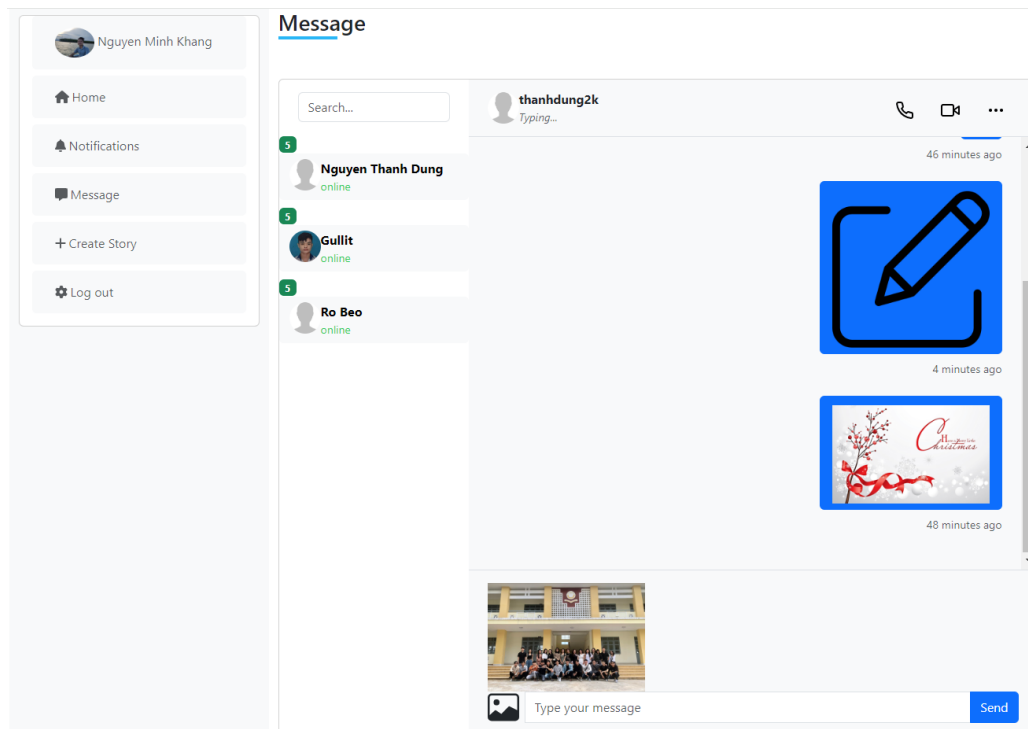


Hình 4. 24 Giao diện nhắn tin văn bản

Thao tác: người dùng nhập tin nhắn vào ô EditText “Type your message”, sau đó nhấn nút “Send” để gửi tin nhắn.

Hệ thống: Hệ thống sẽ gửi tin nhắn realtime cho người dùng bên kia, đồng thời tin nhắn sẽ hiện lên trên phần chi tiết tin nhắn.

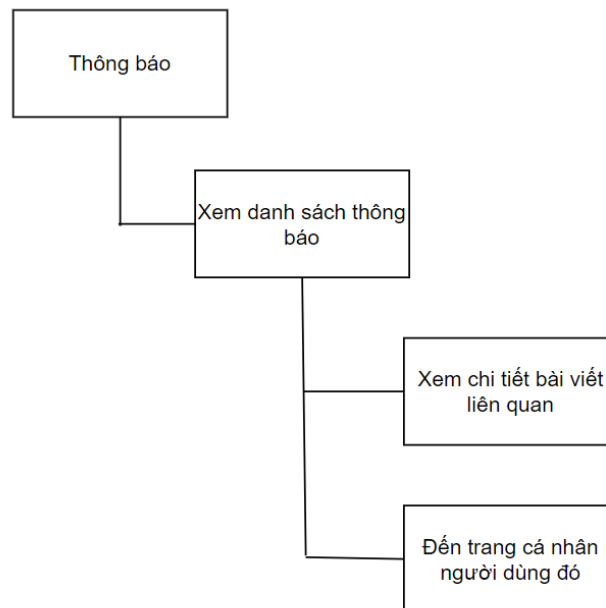
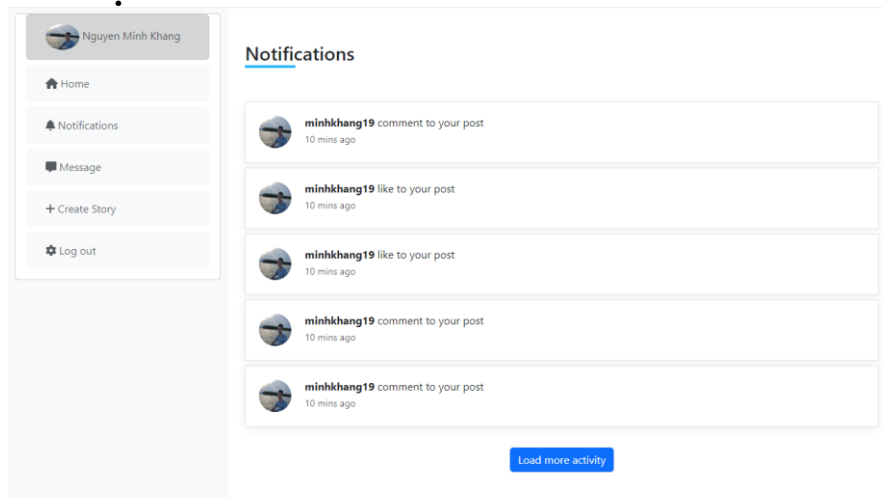
Giao diện gửi tin nhắn hình:



Hình 4. 25 Giao diện gửi tin nhắn hình

Thao tác: người dùng chọn biểu tượng hình ảnh, sau đó chọn hình để gửi, nhấn “Send” để gửi hình.

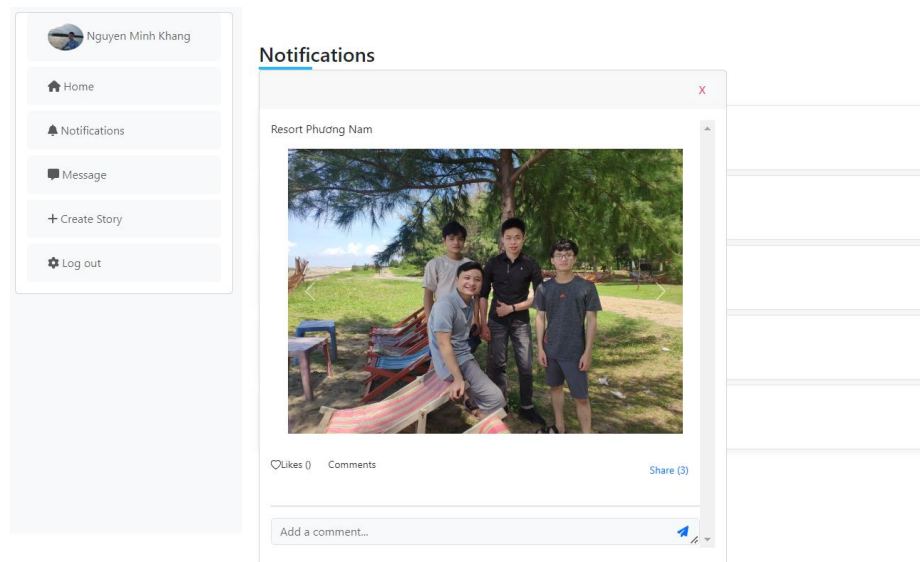
Hệ thống: Hệ thống sẽ tin nhắn hình realtime cho người dùng đó, và đồng thời hiện hình ảnh lên trên cả 2 bên.

4.2.8 Quản lý thông báo:**Sơ đồ quản lý thông báo:***Sơ đồ 4. 6 Sơ đồ chức năng quản lý thông báo***Xem danh sách thông báo:****Giao diện:***Hình 4. 26 Giao diện xem danh sách thông báo*

Thao tác: người dùng vào trang thông báo và có thể xem danh sách thông báo, muốn xem tất cả thông báo thì nhấn “Load more activity”

Hệ thống: Hệ thống lấy danh sách thông báo cho người dùng và hiện lên trên trang thông báo.

Xem chi tiết bài viết liên quan:**Giao diện:**



Hình 4. 27 Giao diện xem chi tiết bài viết liên quan

Thao tác: Người dùng chọn bất kỳ thông báo nào có bài viết.

Hệ thống: Hệ thống sẽ tự động đưa bài viết đó lên trên màn hình để người dùng xem chi tiết bài viết đó.

Đi đến trang cá nhân của người dùng đó:

Thao tác: Trên mỗi thông báo có phần ảnh đại diện của người dùng đó, người dùng nhấn vào ảnh đại diện để đến trang cá nhân của người dùng đó

Hệ thống: Hệ thống sẽ tự động chuyển về trang cá nhân của người dùng sau khi người dùng nhấn vào ảnh đại diện trên bất kỳ thông báo nào

CHƯƠNG 5. RESTFUL WEB SERVICE

5.1 Công nghệ sử dụng:

- Ngôn ngữ: Javascript.
- Framework: Express, NodeJS.
- Thư viện hỗ trợ: Sequelize.
- Bảo mật sử dụng JWT và BrcryptJS.

5.2 Web Service:

- Web Service là một dịch vụ web, nó là một khái niệm rộng hơn so với khái niệm web thông thường, nó cung cấp các thông tin thô, và khó hiểu với đa số người dùng (chúng thường được trả về dưới dạng XML hoặc JSON.), chính vì vậy nó được sử dụng bởi các ứng dụng. Các ứng dụng này sẽ chế biến các dữ liệu thô trước khi trả về cho người dùng cuối cùng.
- RESTful Web Service là các Web Service được viết dựa trên kiến trúc REST. REST đã được sử dụng rộng rãi thay thế cho các Web Service dựa trên SOAP và WSDL. RESTful Web Service nhẹ (lightweight), dễ dàng mở rộng và bảo trì.
- RESTful giống như 1 kiểu kiến trúc gửi/ nhận (client gửi request/dữ liệu tới, server trả dữ liệu tương ứng về)

5.3 Tính chất:

5.3.1 Sử dụng các phương thức HTTP một cách rõ ràng:

- REST đặt ra một quy tắc đòi hỏi lập trình viên xác định rõ ý định của mình thông qua các phương thức của HTTP. Thông thường ý định đó bao gồm lấy dữ liệu, trên dữ liệu, cập nhập dữ liệu hoặc xóa dữ liệu. Vậy khi bạn muốn thực hiện một trong các ý định trên hãy lưu ý các quy tắc sau:
 - Để tạo một tài nguyên trên máy chủ, bạn cần sử dụng phương thức POST.
 - Để truy xuất một tài nguyên, sử dụng GET.
 - Để thay đổi trạng thái một tài nguyên hoặc để cập nhập nó, sử dụng PUT.
 - Để hủy bỏ hoặc xóa một tài nguyên, sử dụng DELETE.

Các nguyên tắc ở trên là không bắt buộc, thực tế bạn có thể sử dụng phương thức GET để yêu cầu lấy dữ liệu, trên, sửa hoặc xóa dữ liệu trên Server. Tuy nhiên REST đưa ra các nguyên tắc ở trên mục đích đưa mọi thứ trở lên rõ ràng và dễ hiểu.

5.3.2 Phi trạng thái:

- Một đặc điểm của REST là phi trạng thái (stateless), có nghĩa là nó không lưu giữ thông tin của client. Chẳng hạn bạn vừa gửi yêu cầu để xem trang thứ 2 của một tài liệu, và bây giờ bạn muốn xem trang tiếp theo (sẽ là trang 3). REST không lưu trữ lại thông tin rằng trước đó nó đã phục vụ bạn trang số 2. Điều đó có nghĩa là REST không quản lý phiên làm việc (Session). Hiện thị cấu trúc thư mục như các URLs:

REST đưa ra một cấu trúc để người dùng có thể truy cập vào tài nguyên của nó thông qua các URL, tài nguyên ở đây là tất cả những cái mà bạn có thể gọi tên được (Video, ảnh, báo cáo thời tiết,..)

5.3.3 Truyền tải JavaScript Object Notation (JSON), XML hoặc cả hai:

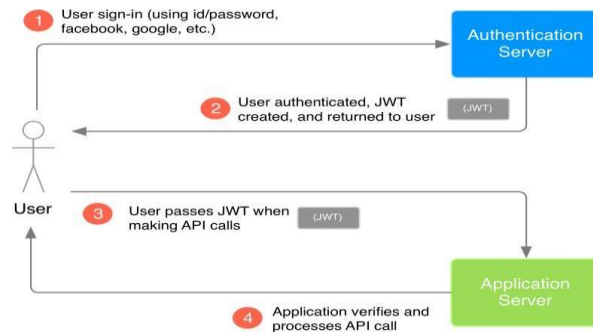
- Khi Client gửi một yêu cầu tới web service nó thường được truyền tải dưới dạng XML hoặc JSON và thông thường nhận về với hình thức tương tự.
- Đôi khi Client cũng có thể chỉ định kiểu dữ liệu nhận về mà nó mong muốn (JSON, hoặc XML,..), các chỉ định này được gọi là các kiểu MINE, nó được gửi kèm trên phần HEADER của request.
- Dưới đây là các kiểu MINE phổ biến thường sử dụng với REST service. JSON: application/json, XML: application/xml.
- XHTML: application/xhtml+xml.

5.4 Bảo mật RESTful Service sử dụng Auth0 JWT:

5.4.1 JWT - JSON Web Token:

- JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

5.4.2 Sơ lược về luồng xử lý:



Hình 5. 1 Sơ lược luồng xử lý xác thực Authentication Server với JWT

- 1) User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (Server xác thực).
- 2) Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
- 3) Người dùng nhận được JWT do Authentication Server vừa mới trả về làm "chìa khóa" để thực hiện các "lệnh" tiếp theo đối với Application Server.
- 4) Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify JWT gửi lên. Nếu OK, tiếp tục thực hiện yêu cầu được gọi.

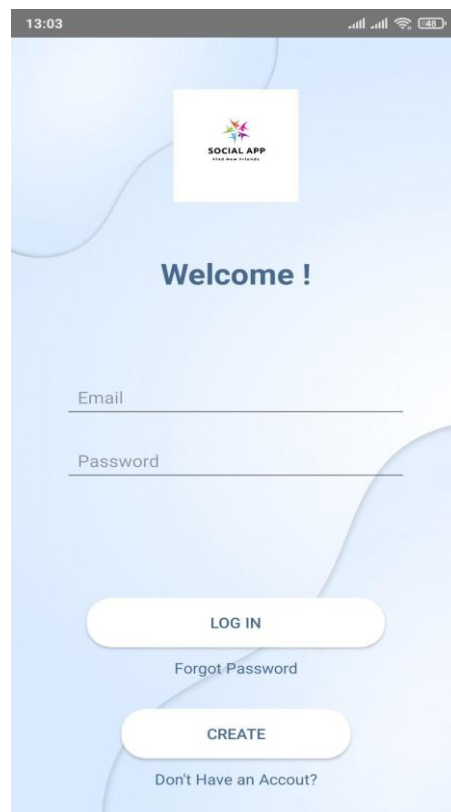
CHƯƠNG 6. ỨNG DỤNG ANDROID APP

6.1 Yêu cầu hệ thống:

- Thiết bị sử dụng nền tảng Android 8.0 trở lên.
- Thiết bị được kết nối internet.

6.2 Demo giao diện:

6.2.1 Đăng nhập:



Hình 6. 1 Giao diện đăng nhập

- Tại màn hình login, người dùng nhập username và password sau đó bấm login để tiến hành đăng nhập.

6.2.2 Trang chủ:

- Sau khi đăng nhập thành công, hệ thống sẽ chuyển người dùng vào trang chủ:

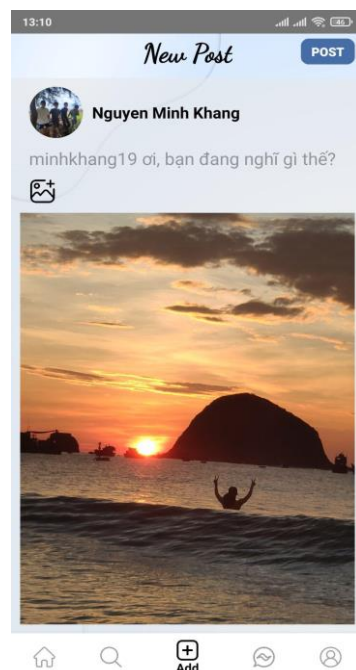


Hình 6. 2 Giao diện trang chủ

6.2.3 Bài viết:

Tạo bài viết

- Để đăng bài viết nào đó, ta điền nội dung bài viết có thể là chuỗi chữ, có thể là hình ảnh hoặc có thể là cả hai



Hình 6. 3 Giao diện tạo bài viết

- Sau khi chọn ảnh và viết nội dung bài viết, hình ảnh được chọn và nội dung sẽ hiện thị:

Giao diện bao gồm:

- **Hình ảnh:** danh sách hình ảnh đã chọn.
- **Nội dung bài viết:** Chuỗi kí tự người dùng nhập.

Khi nhấn button “Post”, bài viết sẽ được đăng lên trang chủ:

Xem và tương tác bài viết (like, comment, edit, delete)

- Người dùng nhấn vào button like (icon trái tim) để like bài viết:



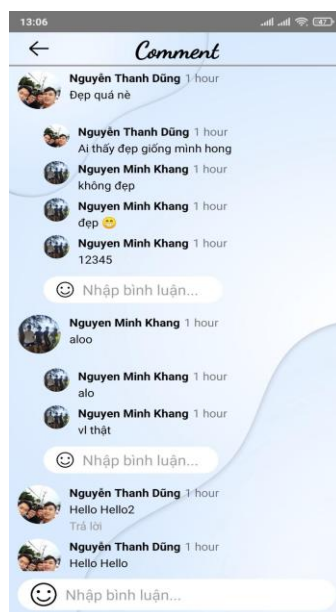
Hình 6. 4 Like bài viết

- Người dùng nhấn vào button comment (icon comment) để bình luận bài viết:



Hình 6. 5 Button Comment bài viết

- Giao diện sẽ chuyển sang màn hình bình luận bài viết
- Tại đây người dùng có thể bình luận hoặc trả lời bình luận của người khác



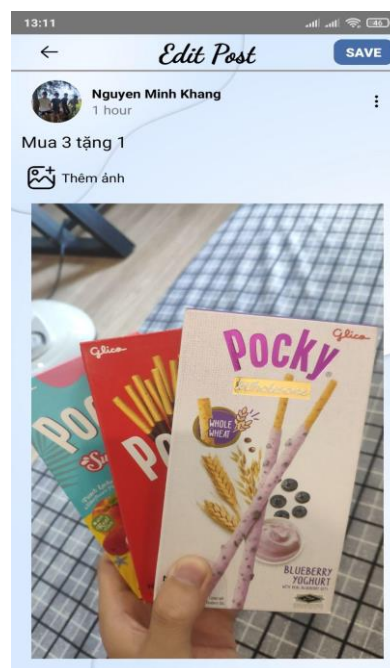
Hình 6. 6 Giao diện Comment bài viết

- Người dùng nhấn vào button edit (icon edit) để chỉnh sửa bài viết:



Hình 6. 7 Button Edit bài viết

- Giao diện sẽ chuyển sang màn hình chỉnh sửa bài viết:
- Tại đây người dùng có thể thay đổi caption, xóa, thêm ảnh cho bài viết
- Nhấn button “SAVE” để cập nhật bài viết



Hình 6. 8 Giao diện Edit bài viết

- Người dùng nhấn và giữ vào bài viết muốn xóa:
- Giao diện sẽ hiện thị lên thông báo xác nhận xóa
- Nhấn button “Xóa” nếu bạn muốn xóa và ngược lại nhấn buttonn “Hủy” để hủy bỏ thay đổi



Hình 6. 9 Giao diện xóa bài viết

6.2.4 Tìm kiếm

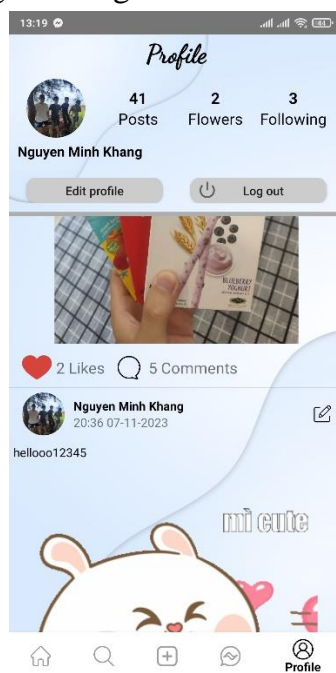
- Người dùng điền từ khóa để tìm kiếm người dùng khác
- Trên giao diện tìm kiếm người dung có thể follow hoặc unfollow người dùng khác
- Người dung có thể nhấn vào để chuyển đến trang cá nhân của user tương ứng



Hình 6. 10 Giao diện tìm kiếm

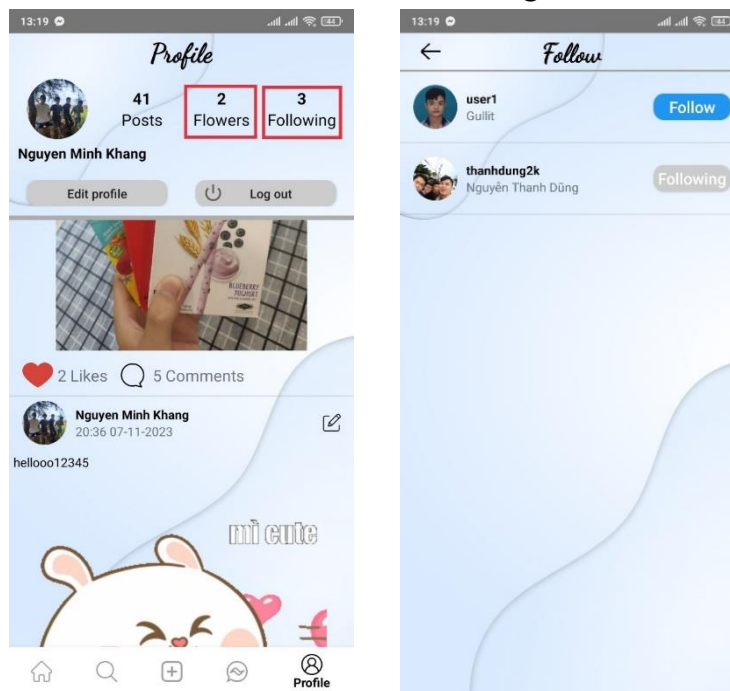
6.2.5 Trang cá nhân

- Trang cá nhân người dùng



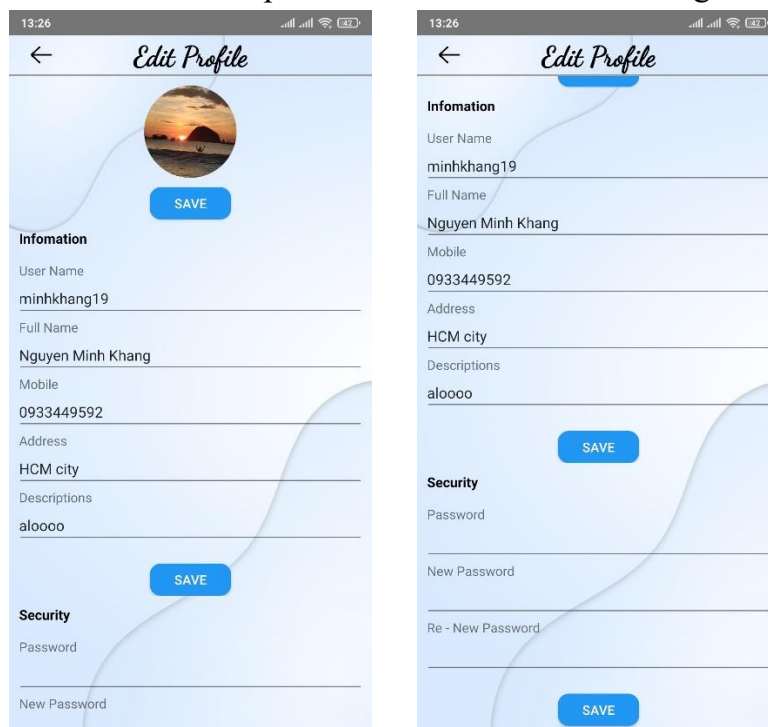
Hình 6. 11 Giao diện người dùng

- Nhấn vào button “follower”, “following” để xem chi tiết



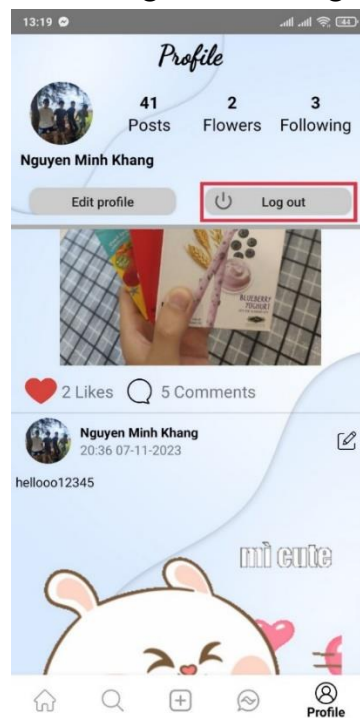
Hình 6. 12 Giao diện xem chi tiết follow

- Nhấn vào button “edit profile” để xem chỉnh sửa thông tin cá nhân



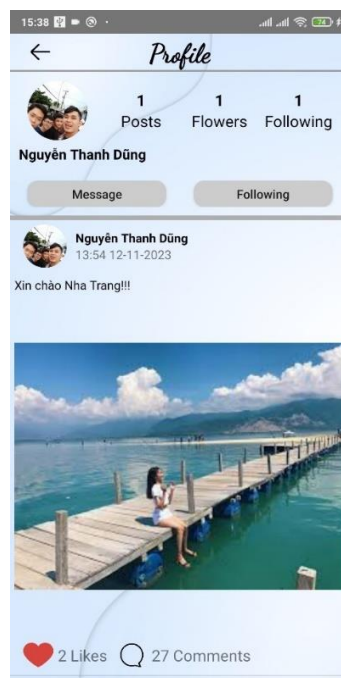
Hình 6. 13 Giao diện chỉnh sửa thông tin cá nhân

- Nhấn vào button “log out” để đăng xuất tài khoản



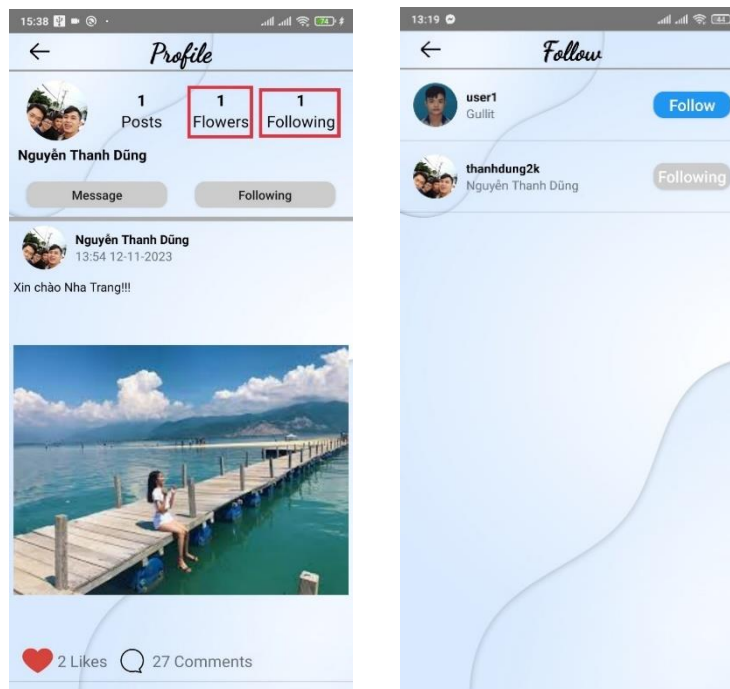
Hình 6. 14 Giao diện logout

- Trang cá nhân khách



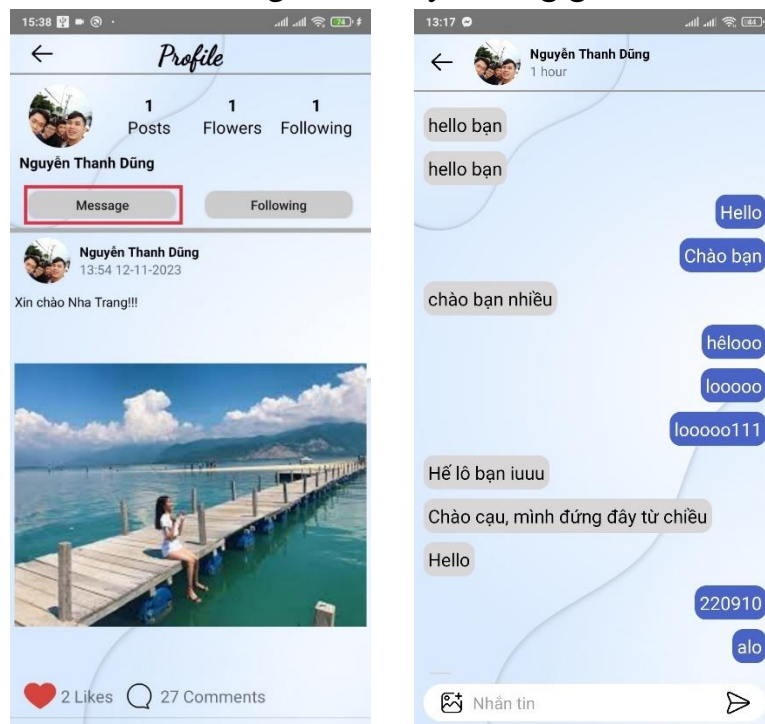
Hình 6. 15 Giao diện người dùng khách

- Nhấn vào button “follower”, “following” để xem chi tiết



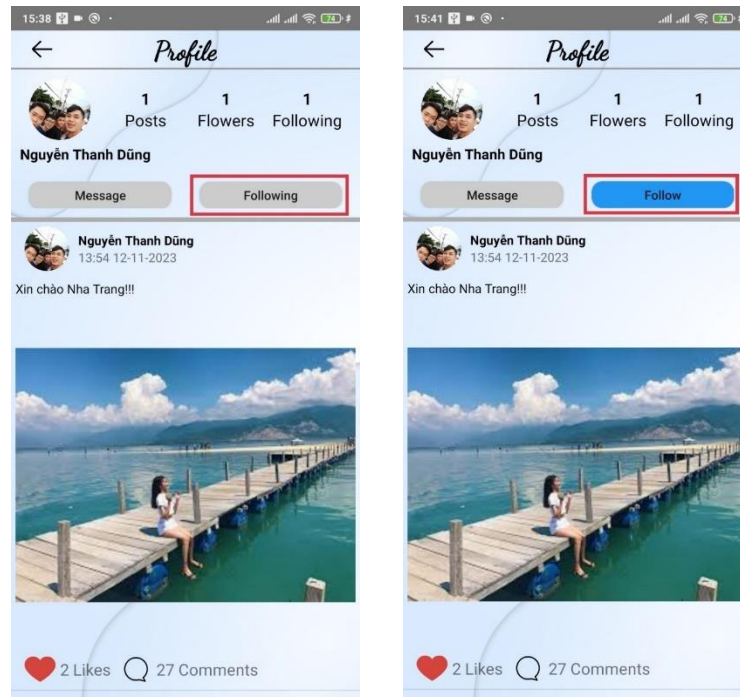
Hình 6. 16 Giao diện xem chi tiết follow

Nhấn vào button “message” để chuyển sang giao diện nhắn tin



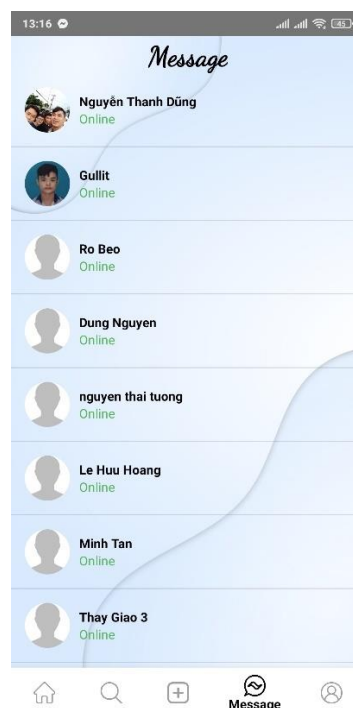
Hình 6. 17 Nhắn tin với người dùng

Nhấn vào button “follow” để follow người dùng



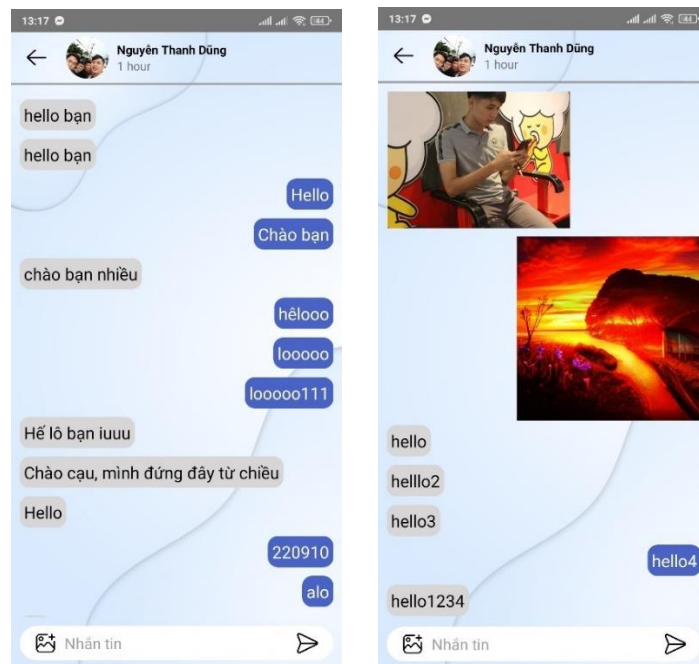
Hình 6. 18 Follow người dùng

6.2.6 Nhắn tin



Hình 6. 19 Giao diện danh sách đoạn hội thoại

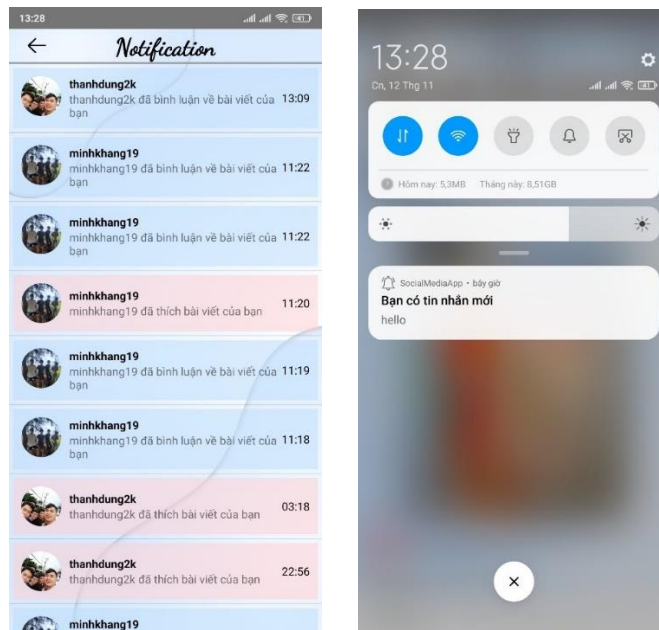
- Giao diện hiển thị danh sách các đoạn hội thoại
- Nhấn vào một người dùng trong danh sách để chuyển sang giao diện nhắn tin



Hình 6. 20 Giao diện nhắn tin

- Giao diện hiển thị nội dung tin cuộc hội thoại
- Người dùng có thể nhắn tin (ký tự hoặc ảnh) cho đối phương
- Tin nhắn được thực hiện realtime và có hiện thị thông báo khi nhận tin nhắn

6.2.7 Thông báo



Hình 6. 6 Giao diện thông báo

- Giao diện hiện thị nội dung thông báo bao gồm like, comment, follow
- Mọi thông báo được nhận real time

CHƯƠNG 7. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

7.1 Tổng kết:

- Mặc dù đã gặp không ít khó khăn nhưng với sự giúp đỡ của các thầy cô giáo, đặc biệt là thầy Huỳnh Trung Trụ, chúng em đã tìm hiểu thực tiễn hoạt động, giải quyết các vấn đề nghiệp vụ của phần mềm mô phỏng mạng xã hội và đã hoàn thành đồ án môn thay thế của mình theo đúng thời gian yêu cầu đề ra. Đồng thời đạt được những mục tiêu đặt ra ở bài báo cáo thực tập tốt nghiệp:

Kết quả đạt được:

- Xây dựng cơ sở dữ liệu cho website mô phỏng mạng xã hội.
- Xây dựng được website mạng xã hội với những chức năng cơ bản:
- Đăng nhập, đăng ký tài khoản.
- Quản lý thông tin cá nhân.
- Tìm kiếm người dùng.
- Nhắn tin
- Đăng bài viết.
- Bình luận.
- Xem trang cá nhân..
- Gợi ý người dùng để kết bạn.
- Sử dụng và học hỏi thêm về NodeJS, ExpressJS, ReactJS, PostgreSQL.

Hạn chế:

- Chưa hoàn thiện hết chức năng cần thiết cho trang web.
- Chưa được thử nghiệm sử dụng thực tế.
- Chưa hỗ trợ đa ngôn ngữ.

7.2 Hướng phát triển:

- Trên cơ sở kế thừa những gì đã đạt được, tiếp tục sửa chữa và khắc phục hạn chế mà đồ án hiện đang có.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Kiến thức về ReactJS: <https://glints.com/vn/blog/hoc-lap-trinh-reactjs/>
2. Kiến thức về PostgreSQL: <https://viblo.asia/p/tim-hieu-he-quan-tri-co-so-du-lieu-postgresql-m68Z0eLdlkG>
3. Kiến thức về NodeJS: <https://glints.com/vn/blog/nodejs-la-gi/>
4. Kiến thức về ExpressJS: <https://itnavi.com.vn/blog/expressjs-la-gi>
5. Android Studio Document. <https://developer.android.com/docs>