**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Xây dựng ứng dụng mạng xã hội kết nối cộng đồng Ntalk**

**PHẠM THÀNH NAM**

nam.pt173266@sis.hust.edu.vn

**Ngành Khoa Học Máy Tính**

**Chuyên ngành Công Nghệ Phần Mềm**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | Ths. Lê Đức Trung |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ thông tin – Truyền thông |
| **HÀ NỘI, 6/2021** | |

# Lời cam kết

Họ và tên sinh viên: Phạm Thành Nam

Điện thoại liên lạc: 0348984167 Email: nam.pt173266@sis.hust.edu.vn

Lớp: KHMT.03-K62 Hệ đào tạo: Cử nhân kĩ thuật

Tôi – *Phạm Thành Nam* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của Ths. Lê Đức Trung. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 2 tháng 6 năm 2021*  Tác giả ĐATN  *Phạm Thành Nam* |
|  |  |

# Lời cảm ơn

Để hoàn thành đồ án tốt nghiệp này đúng hạn và đạt được kết quả tốt nhất, ngoài sự nỗ lực và cố gắng của bản thân, em xin phép được gửi lời cảm ơn chân thành nhất đến ThS. Lê Đức Trung người đã trang bị cho em những kỹ năng cơ bản và kiến thức cần thiết để em có thể hoàn thành được đồ án này. Trong suốt thời gian đồ án, ThS. Lê Đức Trung đã luôn đồng hành, gợi ý đề tài và hướng dẫn cũng như đưa ra các giải pháp và hướng đi thiết thực cho ứng dụng.

Bên cạnh đó, em xin gửi lời cảm ơn chân thành đến gia đình, bạn bè đã luôn động viên, tạo điều kiện và là chỗ dựa tinh thần vững chắc để em có điều kiện tốt nhất và tập trung nhất để hoàn thành đồ án tốt nghiệp trọn vẹn.

Tuy nhiên, trong quá trình làm đồ án do thời gian có hạn và trình độ còn nhiều hạn chế nên không thể tránh khỏi một vài thiếu sót khi trình bày và đánh giá vấn đề. Rất mong nhận được sự góp ý, đánh giá của các thầy cô bộ môn để đề tài của em thêm hoàn thiện hơn.

Em xin chân thành cảm ơn !

# Tóm tắt

Ngày nay, xu thế toàn cầu hóa và sự phát triển nhanh chóng của công nghệ thông tin kéo theo sự phát triển mạnh của mạng internet tại Việt Nam. Sự tham gia của các cá nhân trên không gian mạng ngày càng tích cực và nhu cầu kết nối, chia sẻ thông tin của các cá nhân ngày càng tăng kéo theo sự ra đời và phát triển của hàng loạt các mạng xã hội. Điển hình như các website và ứng dụng mạng xã hội: facebook, zalo, tiktok, gapo, lotus... Tuy đã có nhiều mạng xã hội ra mắt và được đưa vào vận hành nhưng nhu cầu của người dùng vẫn rất cao và tính phân loại ngày càng cần thiết. Nếu như mạng xã hội tiktok là nơi giao lưu clip cho giới trẻ và zalo là mạng xã hội chuyên nhắn tin thì việc tạo ra một mạng xã hội mang nhiều sự kết hợp giữa các loại hình mạng xã hội này nhằm mang lại cho người dùng nhiều sự lựa chọn mới mẻ luôn là một bài toán cần thiết và được đón nhận.

Có rất nhiều cách tiếp cận khi triển khai xây dựng một mạng xã hội như: mạng xã hội phân theo tuổi tác, mạng xã hội việc làm,… nhưng điểm chung giữa các mạng xã hội là người dùng có thể chia sẻ thông tin và tìm kiếm thông tin theo mục đích sử dụng. Trong khuôn khổ đề tài, em chọn cách tiếp cận xây dựng mạng xã hội với các tính năng hữu ích và phổ biến nhất theo xu hướng người dùng hiện nay, đó là tìm kiếm và chia sẻ thông tin, nhằm thổi một luồng gió mới cho sự phát triển của các ứng dụng mạng xã hội ở Việt Nam nói chung và trên thế giới nói riêng.

Trên thế giới đã có nhiều ứng dụng mạng xã hội ra đời. Nhưng thành công nhất có thể kể đến các mạng xã hội chia sẻ thông tin và tìm kiếm thông tin theo mục đích sử dụng chung chung và không đi sâu vào các nghiệp vụ cụ thể nào giống như các ứng dụng facebook và twitter. Ở Việt Nam cũng có một vài sản phẩm liên quan tuy nhiên mới chỉ dừng lại ở mức phong trào nên chưa thực sự được đánh giá cao và sử dụng rộng rãi.

Đứng trước bài toán này thì em đã quyết định thực hiện ĐATN của mình bằng đề tài “Xây dựng ứng dụng mạng xã hội kết nối cộng đồng Ntalk” trên cơ sở việc đề xuất một mạng xã hội mới dựa trên tính năng của các mạng xã hội đã có. Phần mềm xây dựng nhằm góp phần tìm kiếm làn gió mới cho sự phát triển của mạng xã hội Việt nói chung và thế giới nói riêng, bên cạnh đó vẫn đảm bảo đầy đủ việc hiển thị nội dung và tính năng của các mạng xã hội truyền thống đã có.

# Mục lục

[Lời cam kết ii](#_Toc74862388)

[Lời cảm ơn iii](#_Toc74862389)

[Tóm tắt iv](#_Toc74862390)

[Mục lục v](#_Toc74862391)

[Danh mục hình vẽ viii](#_Toc74862392)

[Danh mục bảng x](#_Toc74862393)

[Danh mục các từ viết tắt xii](#_Toc74862394)

[Danh mục thuật ngữ xiii](#_Toc74862395)

[Chương 1 Giới thiệu đề tài 1](#_Toc74862396)

[1.1 Đặt vấn đề 1](#_Toc74862397)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc74862398)

[1.3 Định hướng giải pháp 2](#_Toc74862399)

[1.4 Bố cục đồ án 3](#_Toc74862400)

[Chương 2 Khảo sát và phân tích yêu cầu 4](#_Toc74862401)

[2.1 Khảo sát hiện trạng 4](#_Toc74862402)

[2.2 Tổng quan chức năng 6](#_Toc74862403)

[2.2.1 Biểu đồ use case tổng quan 6](#_Toc74862404)

[2.2.2 Biểu đồ phân rã use case quản lý bài viết 7](#_Toc74862405)

[2.2.3 Biểu đồ phân rã use case quản lý thông tin cá nhân 8](#_Toc74862406)

[2.2.4 Biểu đồ phân rã use case quản lý danh sách bạn bè 8](#_Toc74862407)

[2.2.5 Biểu đồ phân rã use case tìm kiếm 9](#_Toc74862408)

[2.2.6 Quy trình nghiệp vụ 10](#_Toc74862409)

[2.3 Đặc tả chức năng 12](#_Toc74862410)

[2.3.1 Đặc tả use case tạo bài viết 12](#_Toc74862411)

[2.3.2 Đặc tả use case tìm kiếm người dùng 13](#_Toc74862412)

[2.3.3 Đặc tả use case tạo nhóm nhắn tin 14](#_Toc74862413)

[2.3.4 Đặc tả use case nhắn tin 15](#_Toc74862414)

[2.3.5 Đặc tả use case chặn người dùng 16](#_Toc74862415)

[2.4 Yêu cầu phi chức năng 17](#_Toc74862416)

[2.4.1 Yêu cầu đáp ứng đối với cơ sở dữ liệu 17](#_Toc74862417)

[2.4.2 Yêu cầu về đáp ứng bảo mật 17](#_Toc74862418)

[2.4.3 Yêu cầu về giao diện 17](#_Toc74862419)

[2.4.4 Yêu cầu về xử lý logic các trường nhập liệu 18](#_Toc74862420)

[Chương 3 Công nghệ sử dụng 19](#_Toc74862421)

[3.1 Android SDK 19](#_Toc74862422)

[3.1.1 Tổng quan 19](#_Toc74862423)

[3.1.2 Ưu điểm Android SDK 19](#_Toc74862424)

[3.2 Spring Boot Framework 19](#_Toc74862425)

[3.2.1 Tổng quan 19](#_Toc74862426)

[3.2.2 Ưu điểm Spring Boot 20](#_Toc74862427)

[3.3 Mã hóa SHA 21](#_Toc74862428)

[Chương 4 Phát triển và triển khai ứng dụng 22](#_Toc74862429)

[4.1 Thiết kế kiến trúc 22](#_Toc74862430)

[4.1.1 Lựa chọn kiến trúc phần mềm 22](#_Toc74862431)

[4.1.2 Thiết kế tổng quan 24](#_Toc74862432)

[4.1.3 Thiết kế chi tiết gói 26](#_Toc74862433)

[4.2 Thiết kế chi tiết 29](#_Toc74862434)

[4.2.1 Thiết kế giao diện 29](#_Toc74862435)

[4.2.2 Thiết kế lớp 33](#_Toc74862436)

[4.2.3 Thiết kế cơ sở dữ liệu 41](#_Toc74862437)

[4.3 Xây dựng ứng dụng 42](#_Toc74862438)

[4.3.1 Thư viện và công cụ sử dụng 42](#_Toc74862439)

[4.3.2 Kết quả đạt được 43](#_Toc74862440)

[4.3.3 Minh hoạ các chức năng chính 44](#_Toc74862441)

[4.4 Kiểm thử 47](#_Toc74862442)

[4.4.1 Compatibility Testing 47](#_Toc74862443)

[4.4.2 Black box Testing 48](#_Toc74862444)

[4.4.3 Tổng kết kiểm thử 50](#_Toc74862445)

[4.5 Triển khai 50](#_Toc74862446)

[Chương 5 Các giải pháp và đóng góp nổi bật 52](#_Toc74862447)

[5.1 Xây dựng chức năng nhắn tin thời gian thực (real-time) và đẩy thông báo 52](#_Toc74862448)

[5.1.1 Bài toán đặt ra 52](#_Toc74862449)

[5.1.2 Giải pháp 52](#_Toc74862450)

[5.1.3 Kết quả đạt được 53](#_Toc74862451)

[5.2 Xác thực người dùng với giao thức bảo mật OAuth 2.0 và triển khai 55](#_Toc74862452)

[5.2.1 Bài toán đặt ra 55](#_Toc74862453)

[5.2.2 Giải pháp 56](#_Toc74862454)

[5.2.3 Kết quả đạt được 57](#_Toc74862455)

[Chương 6 Kết luận và hướng phát triển 58](#_Toc74862456)

[6.1 Kết luận 58](#_Toc74862457)

[6.2 Hướng phát triển 58](#_Toc74862458)

[Tài liệu tham khảo 60](#_Toc74862459)

# Danh mục hình vẽ

[**Hình 2.1** Biểu đồ use case tổng quan. 6](#_Toc74862460)

[**Hình 2.2** Biểu đồ phân rã use case quản lý bài viết. 7](#_Toc74862461)

[**Hình 2.3** Biểu đồ phân rã use case quản lý thông tin cá nhân. 8](#_Toc74862462)

[**Hình 2.4** Biểu đồ phân rã use case quản lý danh sách bạn bè. 8](#_Toc74862463)

[**Hình 2.5** Biểu đồ phân rã use case tìm kiếm 9](#_Toc74862464)

[**Hình 2.6** Quy trình nghiệp vụ tạo mới bài viết. 10](#_Toc74862465)

[**Hình 2.7** Quy trình nghiệp vụ tìm kiếm người dùng. 11](#_Toc74862466)

[**Hình 3.1** Mã hóa dữ liệu SHA-256. 21](#_Toc74862467)

[**Hình 4.1** Mô hình 3 tầng. 22](#_Toc74862468)

[**Hình 4.2** Mô hình client – server 23](#_Toc74862469)

[**Hình 4.3** Biểu đồ phụ thuộc gói 24](#_Toc74862470)

[**Hình 4.4** Thiết kế chi tiết gói cho chức năng quản lý bài viết 26](#_Toc74862471)

[**Hình 4.5** Thiết kế chi tiết gói cho chức năng quản lý thông tin cá nhân 28](#_Toc74862472)

[**Hình 4.6** Sơ đồ phân cấp nhiệm vụ 29](#_Toc74862473)

[**Hình 4.7** Thiết kế thông báo mặc định. 30](#_Toc74862474)

[**Hình 4.8** Thiết kế hộp thoại xác nhận 30](#_Toc74862475)

[**Hình 4.9** Thiết kế cảnh báo nhập liệu 30](#_Toc74862476)

[**Hình 4.10** Thiết kế mockup màn hình trang chủ 31](#_Toc74862477)

[**Hình 4.11** Thiết kế mockup màn hình nhắn tin 32](#_Toc74862478)

[**Hình 4.12** Thiết kế chi tiết một số lớp cho chức năng quản lý bài viết 33](#_Toc74862479)

[**Hình 4.13** Biểu đồ trình tự cho chức năng xem danh sách bài viết 36](#_Toc74862480)

[**Hình 4.14** Thiết kế chi tiết một số lớp cho chức năng quản lý thông tin cá nhân 37](#_Toc74862481)

[**Hình 4.15** Biểu đồ trình tự cho use case xem thông tin cá nhân. 40](#_Toc74862482)

[**Hình 4.16** Biểu đồ thực thể liên kết (ER). 41](#_Toc74862483)

[**Hình 4.17** Thiết kế cơ sở dữ liệu. 42](#_Toc74862484)

[**Hình 4.18** Trang chủ. 44](#_Toc74862485)

[**Hình 4.19** Trang nhắn tin. 45](#_Toc74862486)

[**Hình 4.20** Trang bạn bè. 46](#_Toc74862487)

[**Hình 4.21** Trang cá nhân. 47](#_Toc74862488)

[**Hình 4.22** Sơ đồ triển khai hệ thống 51](#_Toc74862489)

[**Hình 5.1** Quy trình nhắn tin 53](#_Toc74862490)

[**Hình 5.2** Minh họa nhắn tin thời gian thực 54](#_Toc74862491)

[**Hình 5.3** Minh họa đẩy thông báo 55](#_Toc74862492)

[**Hình 5.4** Quy trình gửi nhận dữ liệu thông qua giao thức OAuth 57](#_Toc74862493)

# Danh mục bảng

[**Bảng 1.1** Yêu cầu đặt ra và định hướng giải pháp 2](#_Toc74862494)

[**Bảng 2.1** Ưu, nhược điểm của những ứng dụng hỗ trợ tìm kiếm, quản lý chặn và quản lý yêu cầu kết bạn. 4](#_Toc74862495)

[**Bảng 2.2** So sánh chức năng của những sản phẩm mạng xã hội 5](#_Toc74862496)

[**Bảng 2.3** Đặc tả use case tạo bài viết. 12](#_Toc74862497)

[**Bảng 2.4** Đặc tả use case tìm kiếm người dùng 13](#_Toc74862498)

[**Bảng 2.5** Đặc tả use case tạo nhóm nhắn tin 14](#_Toc74862499)

[**Bảng 2.6** Đặc tả use case nhắn tin 15](#_Toc74862500)

[**Bảng 2.7** Đặc tả use case chặn người dùng 16](#_Toc74862501)

[**Bảng 4.1** Bảng đặc tả lớp PostResponse 33](#_Toc74862502)

[**Bảng 4.2** Bảng đặc tả lớp PostEntity 34](#_Toc74862503)

[**Bảng 4.3** Bảng đặc tả lớp PostServiceImpl 34](#_Toc74862504)

[**Bảng 4.4** Bảng đặc tả lớp PostRepositoryImpl 35](#_Toc74862505)

[**Bảng 4.5** Bảng đặc tả lớp PostController 35](#_Toc74862506)

[**Bảng 4.6** Bảng đặc tả lớp UserProfileResponse 37](#_Toc74862507)

[**Bảng 4.7** Bảng đặc tả lớp UserEntity 38](#_Toc74862508)

[**Bảng 4.8** Bảng đặc tả lớp UserServiceImpl 38](#_Toc74862509)

[**Bảng 4.9** Bảng đặc tả lớp UserRepositoryImpl 39](#_Toc74862510)

[**Bảng 4.10** Bảng đặc tả lớp UserController 40](#_Toc74862511)

[**Bảng 4.11** Danh sách thư viện và công cụ sử dụng 42](#_Toc74862512)

[**Bảng 4.12** Thống kê thông tin ứng dụng 43](#_Toc74862513)

[**Bảng 4.13** Bảng kiểm thử khả năng tương thích 47](#_Toc74862514)

[**Bảng 4.14** Bảng kiểm thử hộp đen cho chức năng “Đăng nhập” 48](#_Toc74862515)

[**Bảng 4.15** Bảng kiểm thử hộp đen cho chức năng “Đổi mật khẩu” 49](#_Toc74862516)

[**Bảng 4.16** Bảng danh sách yêu cầu thiết bị và công cụ triển khai. 50](#_Toc74862517)

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **EUD** | End-User Development  Phát triển ứng dụng người dùng cuối |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |
| **SDK** | Software Development Kit  Bộ công cụ phát triển phần mềm |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **Interpreter** | Trình thông dịch |
| **Compiler** | Trình biên dịch |
| **Backend** | Dịch vụ máy chủ |
| **OAuth 2.0** | Giao thức xác thực người dùng |
| **SHA** | Thuật toán mã hóa thông tin |

# Giới thiệu đề tài

## Đặt vấn đề

Xuất phát từ nhu cầu tìm kiếm và chia sẻ thông tin của các cá nhân trên không gian mạng ngày càng tăng cao, dẫn đến vấn đề hàng loạt các ứng dụng mạng xã hội ra đời. Mỗi ứng dụng mạng xã hội lại chọn các cách thể hiển nội dung khác nhau đi kèm với hàng loạt tính năng hỗ trợ khác nhau. Có những ứng dụng mạng xã hội chỉ tập chung vào việc tìm kiếm và chia sẻ thông tin, có những mạng xã hội chỉ tập chung cho nghiệp vụ nhắn tin… Điều này có thể dẫn đến sự bất tiện cho người sử dụng khi phải tải nhiều ứng dụng mạng xã hội khác nhau cho mỗi nhu cầu khác nhau.

Như vậy, nhu cầu đặt ra cần một ứng dụng có thể kết hợp các tính năng mạnh và hiệu quả nhất trên các mạng xã hội nổi tiếng đã có và bổ sung các tiện ích tìm kiếm hiệu quả hơn cách nhập liệu thủ công thông thường đem lại nhiều lợi ích và sự chọn lựa cho người sử dụng.

## Mục tiêu và phạm vi đề tài

Trước khi bắt tay vào thực hiện đề tài, em đã thực hiện khảo sát các ứng dụng mạng xã hội trong nước và trên thế giới và có sự đánh giá tổng quan về các ứng dụng mạng xã hội này như sau:

Đầu tiên là ứng dụng Zalo, là một ứng dụng nhắn tin đa nền tảng, thuộc sở hữu của công ty cổ phần VNG, ứng dụng cho phép bạn nhắn tin miễn phí với mạng có sẵn trên thiết bị di động. Zalo đã hỗ trợ cho các nền tảng iOS, Android và những điện thoại trên đều có thể nhắn tin cho nhau,… mặc dù có các chức năng giúp người dùng nhắn tin thuận tiện và chia sẻ thông tin lên nhật ký tuy nhiên chức năng tìm kiếm các yêu cầu kết bạn và quản lý yêu cầu kết bạn chưa có, cũng như chức năng chặn/bỏ chặn bạn bè khó khăn vì chưa có phần quản lý chặn riêng mà lồng ghép vào phần quản lý danh sách bạn bè.

Tiếp theo là ứng dụng Gapo là một sản phẩm của Công ty cổ phần Công Nghệ Gapo. Gapo cũng có những chức năng cơ bản như các mạng xã hội khác gồm kết bạn, đăng bài, bình luận, chia sẻ, trò chuyện trực tuyến. Tuy nhiên chức năng tìm kiếm bạn bè vẫn là cách nhập liệu từ bàn phím thủ công.

Nhận thấy những hạn chế này, em quyết định lựa chọn đề tài “Xây dựng ứng dụng mạng xã hội kết nối cộng đồng Ntalk” bao gồm một bộ chức năng đầy đủ, khắc phục những hạn chế của các ứng dụng ra đời trước đó. Ứng dụng bao gồm những chức năng chính sau:

* Quản lý thông tin cá nhân
* Quản lý danh sách bạn bè
* Quản lý bài viết
* Nhắn tin.

Mục tiêu cuối cùng của ứng dụng là nhằm góp phần nào cải thiện những nhược điểm mà các ứng dụng đi trước mắc phải, trên cơ sở đó là nền tảng và là định hướng cho các nghiên cứu và ứng dụng tương tự.

## Định hướng giải pháp

Từ yêu cầu đã được đề cập trong mục 1.2 là cần xây dựng một ứng dụng mạng xã hội kết nối cộng đồng. Ứng dụng đã phát sinh những vấn đề cần giải quyết, những đề xuất định hướng giải pháp để giải quyết những vấn đề đó được nêu rõ trong Bảng 1.1

**Bảng 1.1** Yêu cầu đặt ra và định hướng giải pháp

|  |  |  |
| --- | --- | --- |
| Yêu cầu | Vấn đề cần giải quyết | Định hướng giải pháp |
| Tìm kiếm bạn bè | Giảm thiểu thao tác nhập liệu từ bàn phím điện thoại | Tích hợp chức năng quét mã QR thay thế nhập liệu thủ công trong ứng dụng |
| Gợi ý gần đúng thông tin tìm kiếm | Tìm kiếm gần đúng từ hệ thống |
| Quản lý chặn | Quản lý chặn thuận tiên, nhanh chóng và chính xác. | Xây dựng màn hình ứng dụng riêng tách biệt với màn hình quản lý bạn bè |
| Quản lý yêu cầu kết bạn | Quản lý yêu cầu kết bạn thuận tiện, nhanh chóng và chính xác. | Xây dựng màn hình ứng dụng riêng tách biệt với màn hình quản lý bạn bè |

Sau khi giải quyết được những vấn đề trên, hệ thống đã giải quyết cơ bản được những vấn đề tồn đọng của những mạng xã hội tiền nhiệm. Ứng dụng được xây dựng trên mô hình client-server, phía client dựa trên nền tảng android, phía server dựa trên nền tảng web sử dụng Java Spring Boot và hệ quản trị cơ sở dữ liệu MySQL.

## Bố cục đồ án

Báo cáo đồ án này được chia làm sáu chương và nội dung của các chương được tóm gọn lại lần lượt như sau:

Chương 1 là chương giới thiệu đề tài, tại chương này chỉ rõ các vấn đề tồn tại của các hệ thống mạng xã hội hiện đang có trên thị trường đồng thời tìm hiểu các giải pháp và công nghệ có sẵn. Trên cơ sở đó chỉ rõ các yếu tố đã giải quyết, các vấn đề còn tồn đọng chưa xử lý được, tại đồ án này đã giải quyết và triển khai được những vấn đề gì. Đồng thời so sánh hệ thống đang xây dựng với các hệ thống có sẵn trên thị trường.

Tại Chương 2 của quyển đồ án, em trình bày chi tiết hiện trạng mà em đã khảo sát được từ các hệ thống thực tế. Từ đó đưa ra những phân tích yêu cầu nhằm giải quyết những vấn đề còn tồn tại trong thực tế. Trong đó, đi từ tổng quan chức năng của phần mềm cho đến quy trình nghiệp vụ quan trọng, đặc tả chi tiết các chức năng và mô tả các yêu cầu phi chức năng của hệ thống.

Với những khảo sát và đánh giá đã thu thập được ở Chương 2, tại Chương 3 em sẽ đưa ra các cơ sở lý thuyết được áp dụng cho đề tài và giới thiệu các công nghệ được sử dụng để xây dựng hệ thống. Đầu tiên là việc lựa chọn bộ công cụ Android SDK trong việc xây dựng ứng dụng di động. Tiếp đó là ứng dụng framework Java Spring Boot để xây dựng và phát triển hệ thống backend. Cuối cùng là thuật toán mã hóa SHA để mã hóa các dữ liệu đảm bảo tính an toàn và bảo mật cho hệ thống.

Ở Chương 4 của quyển đồ án, sẽ trình bày về quá trình phát triển, kiểm thử và triển khai ứng dụng. Trong phần phát triển ứng dụng, đi từ khâu lựa chọn kiến trúc sử dụng để xây dựng hệ thống cho đến các khâu chi tiết như thiết kế gói, thiết kế lớp, thiết kế cơ sở dữ liệu, thiết kế giao diện hệ thống và trình bày về thư viện và công cụ đã sử dụng cho việc xây dựng hệ thống. Trong phần cuối - phần kiểm thử, em trình bày những kỹ thuật kiểm thử đã sử dụng đồng thời nêu các trường hợp kiểm thử chi tiết cho từng kỹ thuật đã chọn lựa.

Trong Chương 5 của quyển đồ án, em xin trình bày đóng góp và giải pháp nổi bật của mình khi thực hiện phát triển xong ứng dụng bao gồm các đóng góp trong việc tìm hiểu và xây dựng ứng dụng.

Cuối cùng, Chương 6 là kết luận về sản phẩm của em với những kinh nghiệm đã học hỏi và đúc kết được trong quá trình xây dựng hệ thống. Đồng thời chỉ ra những thiếu sót trong khi phát triển phần mềm và hướng phát triển trong tương lai để hoàn thiện và phát triển sản phẩm.

# ­­Khảo sát và phân tích yêu cầu

Ở chương này sẽ trình bày về quá trình khảo sát hiện trạng thực tế, sau khi khảo sát hiện trạng xong sẽ tiến hành phân tích yêu cầu. Cuối cùng, đưa ra những phân tích thiết kế hệ thống phù hợp với yêu cầu đặt ra.

## Khảo sát hiện trạng

Từ những vấn đề được nêu ra trong phần 1.1, sau khi thực hiện khảo sát các ứng dụng mạng xã hội để tìm hiểu kỹ hơn về quy trình nghiệp vụ chia sẻ và tìm kiếm thông tin, em đã thu được một số vấn đề xảy ra trong thực tế. Đầu tiên, chức năng tìm kiếm trên một vài ứng dụng tốn rất nhiều công sức trong việc nhập liệu và ghi nhớ nội dung tìm kiếm. Đối với những trường hợp yêu cầy kết quả ngay lập tức thì chưa đáp ứng được. Tiếp theo là quy trình quản lý danh sách chặn bạn bè phức tạp, tốn nhiều bước. Cuối cùng là quản lý yêu cầu kết bạn phức tạp khi số lượng yêu cầu lớn và yêu cầu kết bạn trên các ứng dụng này còn chưa có sự phân loại cụ thể.

Với những vấn đề trên, hiện nay đã có nhiều ứng dụng được xây dựng để đơn giản hóa công việc tìm kiếm, quản lý chặn và quản lý yêu cầu kết bạn. Một số sản phẩm nổi bật được em tìm hiểu được là: Zalo, Facebook, Gapo. Chi tiết hơn về các sản phẩm này được trình bày trong Bảng 2.1

**Bảng 2.1** Ưu, nhược điểm của những ứng dụng hỗ trợ tìm kiếm, quản lý chặn và quản lý yêu cầu kết bạn.

|  |  |  |
| --- | --- | --- |
| Tên ứng dụng | Mô tả | Ưu, nhược điểm |
| Zalo | là một ứng dụng nhắn tin đa nền tảng, thuộc sở hữu của **công ty cổ phần VNG**, cho phép nhắn tin miễn phí với mạng có sẵn trên thiết bị di động. Zalo đã hỗ trợ cho các nền tảng iOS, Android và những điện thoại trên đều có thể nhắn tin cho nhau. | Ưu điểm của Zalo là hỗ trợ tính năng nhắn tin bảo mật, tìm kiếm bạn thuận tiện, mạnh mẽ thông qua tiện ích QrBarcode.  Nhược điểm là chưa hỗ trợ quản lý yêu cầu kết bạn, quản lý danh sách chặn |
| Facebook | là một phương tiện truyền thông xã hội và dịch vụ mạng xã hội trực tuyến thành lập vào năm 2004 của Mỹ có trụ sở tại Califorlia. Nó được Mark Zuckerberg cùng với các đồng nghiệp sáng lập. Đây được coi là một trong bốn công ty công nghệ lớn nhất nước Mỹ cùng với Amazon, Apple và Google | Ưu điểm của Facebook là giao tiếp với bạn bè, xem ảnh & video cũng như chỉnh sửa trang cá nhân của họ tốt và thuận tiện.  Nhược điểm là chức năng tìm kiếm nhập liệu từ bàn phím thủ công. Chức năng nhắn tin chưa có, phải tải thêm ứng dụng phụ đi kèm là facebook message. |
| Gapo | là một dịch vụ mạng xã hội trực tuyến của [Việt Nam](https://vi.wikipedia.org/wiki/Vi%E1%BB%87t_Nam) được thành lập vào năm 2019 bởi Công ty cổ phần Công nghệ Gapo có trụ sở tại Thanh xuân, [Hà Nội](https://vi.wikipedia.org/wiki/H%C3%A0_N%E1%BB%99i). Hướng tới đối tượng người dùng trẻ và hướng tới mục tiêu “thay thế Facebook” trong tương lai. | Ưu điểm Gapo là có những chức năng cơ bản đầy đủ như các mạng xã hội khác gồm kết bạn, đăng bài, bình luận, chia sẻ, trò chuyện trực tuyến.  Nhược điểm là chức năng tìm kiếm nhập liệu từ bàn phím thủ công nên chưa tạo đột phá hơn các mạng xã hội truyền thống. |

**Bảng 2.2** So sánh chức năng của những sản phẩm mạng xã hội

|  |  |  |  |
| --- | --- | --- | --- |
| Chức năng | Zalo | Facebook | Gapo |
| Quản lý thông tin cá nhân | x | x | x |
| Quản lý yêu cầu kết bạn |  | x | x |
| Quản lý danh sách chặn |  | x |  |
| Tìm và kết bạn bằng QrBarcode | x |  |  |
| Bình luận, nhắn tin. | x |  | x |

Dựa vào khảo sát trên, ta có thể nhận thấy những sản phẩm ứng dụng trên chưa thực sự đáp ứng đầy đủ nhu câu người dùng trong thực tế. Những ứng dụng như Facebook, Gapo thì chưa giải quyết được vấn đề giảm bớt công sức về tìm kiếm thông tin. Zalo mặc dù giải quyết được vấn đề về tìm kiếm thông tin qua QrBarcode thì lại không có chức năng quản lý yêu cầu kết bạn, quản lý danh sách chặn.

Từ đó, mục tiêu của em là xây dựng ứng dụng mạng xã hội nhằm giúp giảm tải thời gian, công sức cho việc tìm kiếm thông tin. Giúp hỗ trợ người dùng quản lý danh sách chặn và yêu cầu liên quan đến nghiệp vụ kết bạn.

## Tổng quan chức năng

### Biểu đồ use case tổng quan

Khi chưa đăng nhập, người dùng cần yêu cầu đăng nhập hoặc đăng ký tài khoản để bắt đầu sử dụng các chức năng của phần mềm. Khi người dùng đăng nhập thành công, người dùng sẽ trở thành người dùng hệ thống và hệ thống sẽ hiển thị giao diện tương ứng.

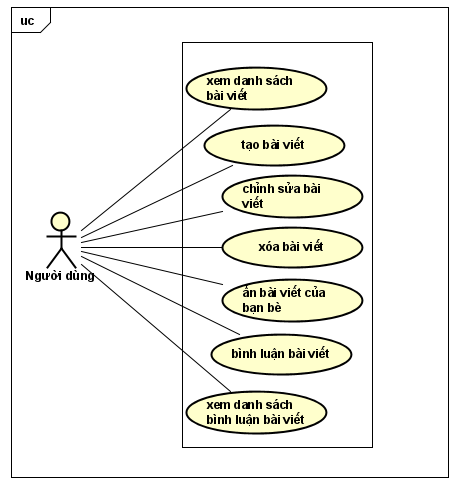
Diagram

Description automatically generated

**Hình 2.1** Biểu đồ use case tổng quan.

Hình 2.1 Thể hiện tổng quan các chức năng của ứng dụng mạng xã hội. Hệ thống có tác nhân chính là người dùng. Trong đó người dùng nói chung muốn sử dụng các chức năng của hệ thống thì bắt buộc phải thực hiện việc đăng nhập. Sau đó người dùng trở thành người dùng hệ thống và có thể truy cập các chức năng khác.

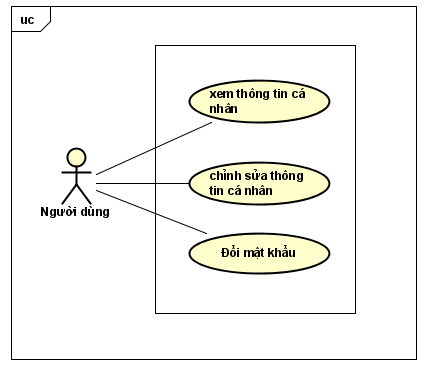
### Biểu đồ phân rã use case quản lý bài viết



**Hình 2.2** Biểu đồ phân rã use case quản lý bài viết.

Hình 2.2 thể hiện biểu đồ phân rã của use case quản lý bài viết. Trong ca sử dụng này, người dùng có thể xem danh sách bài viết, truy cập các bài viết trong danh sách, xem danh sách bình luận và bình luận về bài viết đã truy cập. Người dùng có thể tạo mới thông tin bài viết, chỉnh sửa thông tin bài viết và xóa thông tin bài viết của mình. Ngoài ra, ở ca sử dụng này người dùng có thể ẩn bài viết của bạn bè để loại khỏi danh sách bài viết.

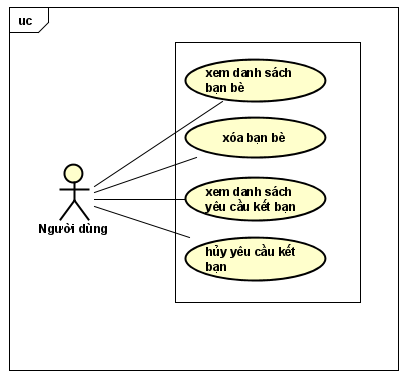
### Biểu đồ phân rã use case quản lý thông tin cá nhân



**Hình 2.3** Biểu đồ phân rã use case quản lý thông tin cá nhân.

Hình 2.3 thể hiện biểu đồ phân rã use case quản lý thông tin cá nhân. Trong ca sử dụng này, người dùng có thể xem thông tin cá nhân, chỉnh sửa thông tin cá nhân như họ tên, địa chỉ, học vấn và đổi mật khẩu.

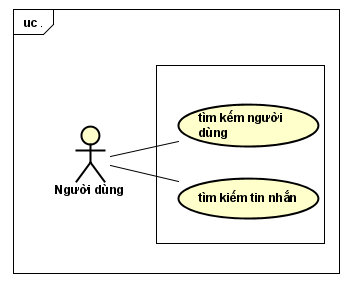
### Biểu đồ phân rã use case quản lý danh sách bạn bè



**Hình 2.4** Biểu đồ phân rã use case quản lý danh sách bạn bè.

Hình 2.4 thể hiện biểu đồ phân rã của use case quản lý danh sách bạn bè. Trong ca sử dụng này, người dùng có thể xem danh sách bạn bè và xóa bạn bè trong danh sách. Ngoài ra người dùng có thể xem các yêu cầu kết bạn được gửi tới mình, đồng thời xóa yêu cầu kết bạn không mong muốn.

### Biểu đồ phân rã use case tìm kiếm



**Hình 2.5** Biểu đồ phân rã use case tìm kiếm

Hình 2.5 thể hiện biểu đồ phân rã của use case tìm kiếm. Trong ca sử dụng này, người dùng có thể tìm kiếm người dùng khác trong hệ thống và tìm kiếm tin nhắn mình đã tham gia trò chuyện cùng người khác.

### Quy trình nghiệp vụ

#### Quy trình nghiệp vụ tạo mới bài viết

Diagram

Description automatically generated

**Hình 2.6** Quy trình nghiệp vụ tạo mới bài viết.

#### Quy trình nghiệp vụ tìm kiếm người dùng

Diagram

Description automatically generated

**Hình 2.7** Quy trình nghiệp vụ tìm kiếm người dùng.

## Đặc tả chức năng

### Đặc tả use case tạo bài viết

**Bảng 2.3** Đặc tả use case tạo bài viết.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã Use case** | UC001 | | | **Tên Use case** | Tạo bài viết | |
| **Tác nhân** | Người dùng | | | | | |
| **Tiền điều kiện** | Đăng nhập hệ thống | | | | | |
| **Luồng sự kiện chính**  **(Thành công)** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 1. | Người dùng | Chọn tạo mới bài viết | |  |
| 2. | Hệ thống | Hiển thị giao diện tạo bài viết | |  |
| 3. | Người dùng | Nhập nội dung bài viết (\*) | |  |
| 4. | Người dùng | Chọn ảnh và video | |  |
| 5. | Hệ thống | Kiểm tra thông tin người dùng nhập vào | |  |
| 6. | Hệ thống | Thông báo thêm mới thành công (\*\*) | |  |
| 7. | Hệ thống | Chuyển về giao diện xem danh sách bài viết | |  |
| **Luồng sự kiện thay thế** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 6a. | Hệ thống | Thông báo lỗi | |  |
| 7a. | Hệ thống | Hiển thị trạng thái lỗi trên ô nhập liệu | |  |
| **Hậu điều kiện** | Không | | | | | |

Dữ liệu đầu vào (\*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | Nội dung bài viết | Nhập vào nội dung bài viết | Không | Không quá 500 ký tự | Trời đẹp, Phong cảnh đẹp,… |

Dữ liệu đầu ra (\*\*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
| 1. | Thông báo kết quả | Thể hiện kết quả khi người dùng nhấn vào lưu thông tin | Chuỗi ký tự | Thêm khách hàng thành công. |

### Đặc tả use case tìm kiếm người dùng

**Bảng 2.4** Đặc tả use case tìm kiếm người dùng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã Use case** | UC002 | | | **Tên Use case** | Tìm kiếm người dùng | |
| **Tác nhân** | Người dùng | | | | | |
| **Tiền điều kiện** | Đăng nhập hệ thống | | | | | |
| **Luồng sự kiện chính**  **(Thành công)** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 1. | Người dùng | Chọn tìm kiếm người dùng | |  |
| 2. | Hệ thống | Hiển thị giao diện màn hình tìm kiếm | |  |
| 3. | Người dùng | Nhập tên người dùng muốn tìm (\*) | |  |
| 4. | Hệ thống | Kiểm tra nội dung người dùng nhập vào | |  |
| 5. | Hệ thống | Chuyển về giao diện xem danh sách bài viết | |  |
| **Luồng sự kiện thay thế** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 5a. | Hệ thống | Thông báo lỗi (\*\*) | |  |
| **Hậu điều kiện** | Không | | | | | |

Dữ liệu đầu vào (\*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | Tìm kiếm | Nhập vào tên người dùng và tên nhóm | Không | Không quá 500 ký tự | Nam Phạm, Nhóm 03, … |

Dữ liệu đầu ra (\*\*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
| 1. | Thông báo kết quả | Thể hiện kết quả khi người dùng nhấn vào lưu thông tin | Chuỗi ký tự | Lỗi kết nối. |

### Đặc tả use case tạo nhóm nhắn tin

**Bảng 2.5** Đặc tả use case tạo nhóm nhắn tin

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã Use case** | UC003 | | | **Tên Use case** | Tạo nhóm nhắn tin | |
| **Tác nhân** | Người dùng | | | | | |
| **Tiền điều kiện** | Đăng nhập hệ thống | | | | | |
| **Luồng sự kiện chính**  **(Thành công)** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 1. | Người dùng | Chọn tạo nhóm mới | |  |
| 2. | Hệ thống | Hiển thị giao diện tạo nhóm gồm trường nhập liệu tên nhóm và danh sách bạn bè của người dùng | |  |
| 3. | Người dùng | Nhập tên nhóm (\*) | |  |
| 4. | Người dùng | Chọn thành viên nhóm | |  |
| 5. | Hệ thống | Kiểm tra nội dung người dùng đã nhập và chọn | |  |
| 6. | Hệ thống | Thông báo thêm mới thành công (\*\*) | |  |
| 7. | Hệ thống | Chuyển về giao diện xem danh sách tin nhắn | |  |
| **Luồng sự kiện thay thế** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 6a. | Hệ thống | Thông báo lỗi | |  |
| 7a. | Hệ thống | Hiển thị trạng thái lỗi trên ô nhập liệu | |  |
| **Hậu điều kiện** | Không | | | | | |

Dữ liệu đầu vào (\*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | Tên nhóm | Nhập vào nội dung tên nhóm | Có | Không quá 30 ký tự, không trùng tên với nhóm đã lập | Nhóm 3, Nhón aoe 19, … |

Dữ liệu đầu ra (\*\*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
| 1. | Thông báo kết quả | Thể hiện kết quả khi người dùng nhấn vào lưu thông tin | Chuỗi ký tự | Thêm nhóm mới thành công. |

### Đặc tả use case nhắn tin

**Bảng 2.6** Đặc tả use case nhắn tin

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã Use case** | UC004 | | | **Tên Use case** | Nhắn tin | |
| **Tác nhân** | Người dùng | | | | | |
| **Tiền điều kiện** | Đăng nhập hệ thống | | | | | |
| **Luồng sự kiện chính**  **(Thành công)** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 1. | Người dùng | Xem thông tin người dùng khác | |  |
| 2. | Người dùng | Chọn nút nhắn tin | |  |
| 3. | Hệ thống | Hiển thị danh sách tin nhắn giữa 2 người | |  |
| 4. | Người dùng | Nhập liệu tin nhắn (\*) | |  |
| 5. | Người dùng | Nhấn nút gửi tin nhắn | |  |
| 6. | Hệ thống | Kiểm tra nội dung người dùng nhập vào | |  |
| 7. | Hệ thống | Hiển thị tin nhắn hợp lệ vừa nhập | |  |
| 8. | Hệ thống | Hiển thị danh sách tin nhắn mới | |  |
| **Luồng sự kiện thay thế** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 7a. | Hệ thống | Thông báo lỗi | |  |
| 8a. | Hệ thống | Hiển thị trạng thái lỗi trên thông báo | |  |
| **Hậu điều kiện** | Không | | | | | |

Dữ liệu đầu vào (\*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1 | Nội dung tin nhắn | Nhập vào nội dung tin nhắn | Có | Không quá 500 ký tự. | Khỏe chứ, làm bài tập đi, … |

### Đặc tả use case chặn người dùng

**Bảng 2.7** Đặc tả use case chặn người dùng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã Use case** | UC005 | | | **Tên Use case** | Chặn người dùng | |
| **Tác nhân** | Người dùng | | | | | |
| **Tiền điều kiện** | Đăng nhập hệ thống | | | | | |
| **Luồng sự kiện chính**  **(Thành công)** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 1. | Người dùng | Xem thông tin cá nhân người dùng khác | |  |
| 2. | Hệ thống | Hiển thị thông tin cá nhân người dùng tương ứng | |  |
| 3. | Người dùng | Chọn nút chặn | |  |
| 4. | Hệ thống | Hiển thị thông báo xác nhận chặn | |  |
| 5. | Người dùng | Nhấn xác nhận chặn | |  |
| 6. | Hệ thống | Thông báo chặn thành công (\*\*) | |  |
| 7. | Hệ thống | Chuyển về giao diện trang chủ | |  |
| **Luồng sự kiện thay thế** |  | **STT** | **Thực hiện bởi** | **Hành động** | |  |
| 6a. | Hệ thống | Thông báo lỗi | |  |
| 7a. | Hệ thống | Hiển thị trạng thái lỗi | |  |
| **Hậu điều kiện** | Không | | | | | |

Dữ liệu đầu ra (\*\*)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
| 1. | Thông báo kết quả | Thể hiện kết quả khi người dùng nhấn vào lưu thông tin | Chuỗi ký tự | Thêm khách hàng thành công. |

## Yêu cầu phi chức năng

### Yêu cầu đáp ứng đối với cơ sở dữ liệu

Phần mềm sử dụng hệ quản trị dữ liệu đáp ứng yêu cầu:

• Hiệu năng: nhiều client có thể truy cập cùng lúc.

• Linh động: có thể hoạt động trên nền tảng máy chủ phổ biến: Linux, Window.

### Yêu cầu về đáp ứng bảo mật

Trong quá trình lưu trữ, truyền tải dữ liệu giữa các thành phần trong môi trường internet phải đảm bảo an toàn được đặt ra.

Các tiêu chuẩn về bảo mật và toàn vẹn dữ liệu cần đáp ứng yêu cầu:

* Mã hóa các dữ liệu có yêu cầu bảo mật cao
* Sử dụng hỗ trợ an toàn bảo mật của các ứng dụng nền

### Yêu cầu về giao diện

Giao diện cần đẹp và tạo ấn tượng cho người xem, bố cục trình bày dễ xem và dễ dùng, giao diện (hệ thống menu, v.v...) nhất quán, dễ sử dụng và thuận tiện cho người dùng. Các sự kiện khi tương tác với giao diện phải có tính phản hồi, khi phải chờ đợi xử lý tiến trình thì cần có thông báo cho người dùng.

### Yêu cầu về xử lý logic các trường nhập liệu

Các trường nhập liệu trên giao diện cần đảm bảo các nguyên tắc:

• Dữ liệu được kiểm tra trước khi người dùng gửi yêu cầu xác nhận.

• Hiển thị cảnh báo trên trường nhập liệu ngay khi người dùng nhập vào dữ liệu không hợp lệ.

• Các trường nhập liệu bắt buộc có ký hiệu hiển thị gợi ý cho người dùng.

# Công nghệ sử dụng

Tổng quan: Với những đặc tả chức năng trong Chương 2, công việc tiếp theo là xác định công nghệ cần áp dụng vào phần mềm để đáp ứng các yêu cầu đã nêu ra. Trong chương này, sẽ trình bày về 3 công nghệ sử dụng là: Android SDK trong việc xây dựng giao diện di động, framework Spring boot trong việc xây dựng hệ thống máy chủ (server).

## Android SDK

### Tổng quan

Hiện nay, Android là một trong những hệ điều hành di động phổ biến nhất thế giới với khoảng 80% người dùng trên thế giới sử dụng các thiết bị chạy android. Android SDK (Software development kit) là bộ công cụ sử dụng ngôn ngữ java được dùng để giúp lập trình viên phát triển các ứng dụng di động trên nền tảng Android. Theo đó, SDK cung cấp mọi thứ như các thư viện, tài liệu, mẫu template, code mẫu để nhà phát triển có thể xây dựng ứng dụng android dễ dàng và nhanh chóng.

### Ưu điểm Android SDK

* Là bộ công cụ duy nhất và toàn diện để giúp lập trình viên xây dựng ứng dụng trên thiết bị di động chạy hệ điều hành android.
* An toàn. Bởi vì Android SDK được viết bằng Java nên nó kế thừa được tính an toàn của Java. Cơ chế tự động dọn rác và việc không sử dụng con trỏ của Java giúp cho ứng dụng đang chạy thoát khỏi nhiều công việc quản lý bộ nhớ. Đồng thời nó xử lý các lỗi rất an toàn theo cơ chế xử lý ngoại lệ của Java.
* Tính linh hoạt. Đóng gói ứng dụng android dưới dạng một file .apk. chúng ta có thể dễ dàng start ứng dụng android từ file .apk ở mọi thiết bị chạy hệ điều hành android.

## Spring Boot Framework

### Tổng quan

Hiện nay, trong lập trình có một xu hướng được tập trung phát triển là xây dựng các chương trình dịch vụ Java ở phía máy chủ (Server). Spring Boot là một là một thành viên trong hệ sinh thái Spring Framework. Nó giúp chúng ta đơn giản trong việc khởi tạo và cấu hình dự án và tập trung vào nghiệp vụ phát triển ứng dụng. Trước đây, để khởi khởi tạo một dịch vụ backend bằng các công nghệ sử dụng java khá phức tạp và tốn nhiều thời gian bao gồm các bước cấu hình từ việc khai báo các dependency trong file pom.xml cho đến cấu hình bằng XML hoặc annotation phức tạp, với Spring Boot, gần như không phải cấu hình hoặc cấu hình rất đơn giản nhằm giúp tổ chức tập chung cho việc phát triển ứng dụng. Spring Boot thường được sử dụng để:

* Tiếp nhận client request và xử lý thông tin lấy từ client request.
* Xử lý nghiệp vụ và phát sinh chuyên môn (bằng cách truy cập database): Quá trình xử lý dữ liệu và tạo ra các kết quả này có thể yêu cầu kết nối với một cơ sở dữ liệu, thực hiện một cuộc gọi RMI để lưu trữ và lưu lại thông tin theo yêu cầu.
* Tạo và gửi request đến client: Không chỉ gửi dữ liệu (tức là tài liệu) tới khách hàng (trình duyệt) dưới nhiều định dạng như văn bản (JSON hoặc XML), nhị phân (hình ảnh GIF),… mà còn gửi phản hồi HTTP ẩn cho khách hàng (trình duyệt). Điều này bao gồm giao tiếp với trình duyệt hoặc các trình khách khác loại tài liệu đang được trả về (ví dụ, HTML), thiết lập cookie và các tham số bộ nhớ đệm, và các tác vụ khác.

### Ưu điểm Spring Boot

* Dễ di chuyển. Spring Boot được viết bằng Java nên nó có tính di động cao, thực hiện được trên nhiều hệ điều hành, trên các Web Server khác nhau. Khái niệm “Viết một lần, chạy ở mọi nơi” cũng rất đúng với Servlet.
* Mạnh mẽ. Spring Boot hỗ trợ rất hiệu quả cho việc sử dụng các giao diện lõi API như lập trình mạng, xử lý đa luồng, xử lý ảnh, nén dữ liệu, kết nối các CSDL, bảo mật, xử lý phân tán. Nó cũng thích hợp để trao đổi tin, truyền thông giữa Client và Server ổn định.
* An toàn. Bởi vì Spring Boot được viết bằng Java nên nó kế thừa được tính an toàn của Java. Cơ chế tự động dọn rác và việc không sử dụng con trỏ của Java giúp cho ứng dụng đang chạy thoát khỏi nhiều công việc quản lý bộ nhớ. Đồng thời nó xử lý các lỗi rất an toàn theo cơ chế xử lý ngoại lệ của Java.
* Tính tích hợp. Cung cấp một loạt các tính năng phổ biến cho các dự án lớn như nhúng trực tiếp web server như Tomcat, Jetty vào ứng dụng, bảo mật và health check....
* Tính linh hoạt. Đóng gói ứng dụng Spring dưới dạng một file JAR (stand-alone application). Chúng ta có thể dễ dàng start ứng dụng Spring ở mọi nơi, mọi thời điểm chỉ với câu lệnh quen thuộc java -jar.

## Mã hóa SHA

SHA (Secure Hash Algorithm) là một thuật toán băm dùng để chuyển một đoạn dữ liệu nhất định thành một đoạn dữ liệu có chiều dài không đổi với xác suất khác biệt cao. Trong hệ thống đã sử dụng chuẩn mã hóa dữ liệu SHA256.

Ảnh có chứa đối tượng

Mô tả được tạo tự động

**Hình 3.1** Mã hóa dữ liệu SHA-256.

Đặc điểm:

* SHA-256 tạo ra một mã băm có cố định 256-bit (32-byte) gần như duy nhất. Một chuỗi băm được tạo ra không thể được tính toán trở lại. Việc tìm được hai đoạn dữ liệu khác nhau có cùng kết quả băm tạo ra bởi một trong những thuật giải SHA là không khả thi. Thuật toán này phù hợp để xác nhận mật khẩu, thách thức xác thực hàm băm, chống giả mạo, chữ ký số.
* SHA-256 là một trong những hàm băm kế tiếp đối sau SHA-1 và là một trong những hàm băm mạnh nhất hiện tại.

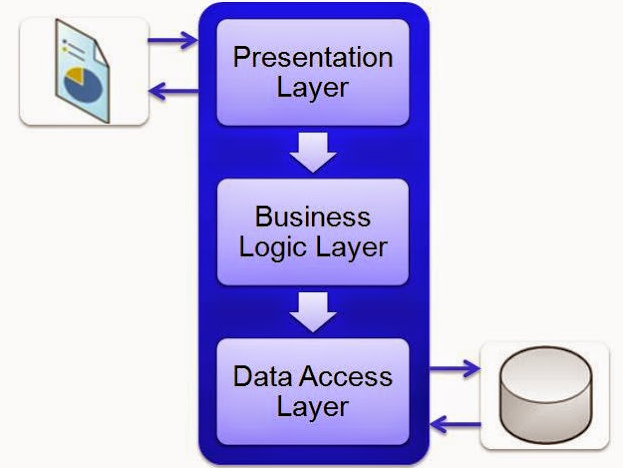
# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

#### Mô hình kiến trúc ba lớp

Để xây dựng phần mềm của mình, em sử dụng mô hình 3 tầng



**Hình 4.1** Mô hình 3 tầng.

Trong mô hình trên:

##### Tầng trình diễn (Presentation Tier):

Lớp này có nhiệm vụ chính là giao tiếp với người dùng. Chứa các thành phần giao diện (web form, button …) để người tương tác như nhập liệu, hiển thị dữ liệu, thực hiện yêu cầu thay đổi dữ liệu. Các yêu cầu sẽ được gửi xuống tầng logic để xử lý.

##### Tầng logic (Business Logic Tier):

Đây là tầng quan trọng nhất của kiến trúc 3 lớp. Mục đích chính của tầng này là để xử lý các nghiệp vụ logic, tăng hiệu năng của hệ thống, tăng cường bảo mật, tăng tính mềm dẻo hơn và tính sử dụng lại.

Tầng này chia làm 2 phần:

* Service Interface: Đây là các interface phục vụ cho việc cài đặt và triển khai các component logic của phân hệ
* Service Implement Component: Cài đặt các service interface, đây là nơi thực thi và xử lý các nghiệp vụ của component logic của phân hệ.

##### Tầng cơ sở dữ liệu (Data Tier):

Tầng này phục vụ cho việc lấy, lưu trữ và cập nhật thông tin từ cơ sở dữ liệu. Tầng này chứa các đối tượng và các lớp có chức năng giao tiếp với cơ sở dữ liệu như thực hiện các truy vấn (tìm, thêm, sửa, xóa…) và trả về tầng business logic.

#### Mô hình client-server

Mô hình client-server là một mô hình nổi tiếng trong [mạng máy tính](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh), được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vài trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.



**Hình 4.2** Mô hình client – server

Client trong mô hình client/server là một máy trạm, một thiết bị bất kì với ý đồ thể hiện tính độc lập của nó. Nó có thể yêu cầu lấy dữ liệu từ một máy chủ (server) hay gửi dữ liệu lên máy chủ (server) đó.

Server còn được định nghĩa như là một máy tính nhiều người sử dụng (multiuser computer). Vì một server phải quản lý nhiều yêu cầu từ các client trên mạng cho nên nó hoạt động sẽ tốt hơn nếu hệ điều hành của nó là đa nhiệm với các tính năng hoạt động độc lập song song với nhau. Các thành phần trên server phải được tách rời nhau để một lỗi của thành phần này không làm hỏng thành phần khác. Tính đa nhiệm đảm bảo một tiến trình không sử dụng hoàn toàn tài nguyên hệ thống.

### Thiết kế tổng quan

Diagram

Description automatically generated

**Hình 4.3** Biểu đồ phụ thuộc gói

Chi tiết hơn về các gói:

Tầng trình diễn (Presentation Layer)

* app-frontend: là phần giao diện trên ứng dụng di động, chứa các file thực hiện việc tạo giao diện cho hệ thống.

Tầng nghiệp vụ (Business Layer):

* controller: thực hiện kết nối với tầng trình diễn, thực hiện nghiệp vụ
* core-service: mô tả phương thức hỗ trợ cho từng nghiệp vụ
* core-service-impl: triển khai các phương thức trong gói core-service

Tầng dữ liệu (Data Layer):

* core-domain: chứa các file dto (data tranfer object) của các entity với các thuộc tính khác nhau tùy thuộc vào nghiệp vụ mong muốn.
* core-repository: mô tả phương thức tương tác với cơ sở dữ liệu riêng cho từng thực thể
* core-repository-impl: triển khai các phương thức đã được mô tả trong gói core-dao cho từng thực thể
* core-data: chứa phương thức tương tác với cơ sở dữ liệu chung cho hệ thống
* core-entity: chứa các thực thể ánh xạ tương ứng với bảng trong cơ sở dữ liệu.

Tầng hỗ trợ (Helper Layer):

* core-utils: chứa các phương thức dùng chung cho toàn bộ phần mềm.
* core-common: chứa các thành phần dùng chung cho toàn bộ phần mềm.

### Thiết kế chi tiết gói

#### Thiết kế chi tiết gói cho thành phần quản lý bài viết

Diagram

Description automatically generated

**Hình 4.4** Thiết kế chi tiết gói cho chức năng quản lý bài viết

Chi tiết hơn về các gói trong hình 4.4:

• Gói core-entity: Chứa các lớp thực thể liên kết với các trường trong CSDL.

• Gói core-domain: Chứa đối tượng với các thuộc tính cần có của thông tin bài viết.

• Gói core-common: Class HibernateUtils hỗ trợ khởi tạo kết nối tới CSDL.

• Gói core-data: Chứa phương thức thực hiện các truy vấn CSDL dùng chung cho cả phần mềm. Với các chức năng quản lý bài viết, sẽ hỗ trợ: Lấy tất cả danh sách, tìm kiếm, chỉnh sửa, tạo mới, xóa thông tin bài viết.

• Gói core-repository, core-repository-impl: Chứa các phương thức truy vấn riêng cho quản lý bài viết.

• Gói core-service, core-service-impl: Thực hiện các yêu cầu nghiệp vụ từ controller

• Gói controller: Chứa các phương thức nhận và xử lý yêu cầu nghiệp vụ.

• Gói app-frontend: Chứa các thành phần giao diện ứng dụng di động cho chức năng quản lý bài viết

#### Thiết kế chi tiết gói cho thành phần quản lý thông tin cá nhân

Diagram

Description automatically generated

**Hình 4.5** Thiết kế chi tiết gói cho chức năng quản lý thông tin cá nhân

Chi tiết hơn về các gói trong Hình 4.5 tương tự như chi tiết gói trong Hình 4.4.

## Thiết kế chi tiết

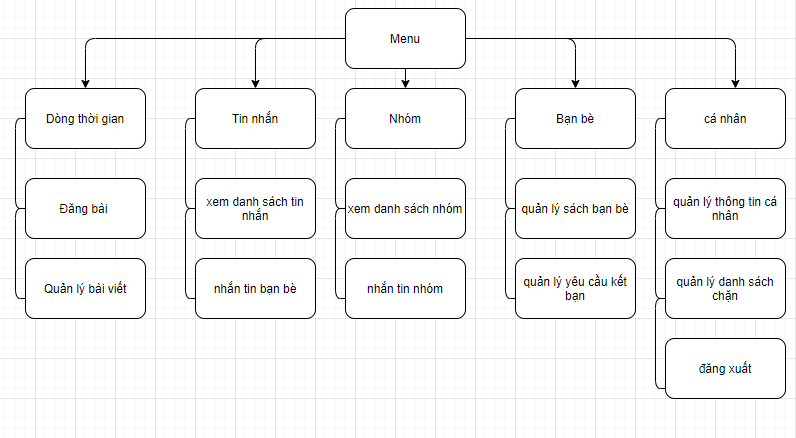
### Thiết kế giao diện

#### Nguyên tắc thiết kế chung

Toàn bộ giao diện của phần mềm sử dụng phong cách thiết kế Material Design nhắm đến những đường nét đơn giản, sử dụng nhiều mảng màu đậm nổi bật, các đối tượng đồ họa trong giao diện uyển chuyển mượt mà. Ngoài ra, nó còn bao gồm cả những hiệu ứng chuyển động tự nhiên khi các nút, menu hiện diện trên màn hình. Các tiêu chuẩn này đề cao tính nhất quán, tăng tính phản hồi cho người dùng sử dụng các màu sắc trang nhã, dễ nhìn không lòe loẹt hoặc dùng các màu sắc quả nổi bật nhằm phù hợp cho phần lớn các nhóm người dùng khác nhau.

#### Thiết kế màn hình

##### Sơ đồ phân cấp chức năng

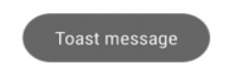


**Hình 4.6** Sơ đồ phân cấp nhiệm vụ

##### Thiết kế thông báo

Thông báo mặc định sẽ được làm nổi bật khi hiển thị:

* Xuất hiện trên hoặc dưới cùng của màn hình.
* Thông báo được đặt trong một thẻ màu nổi bật so với màu nền.



**Hình 4.7** Thiết kế thông báo mặc định.

##### Hộp thoại

Hộp thoại xác nhận xuất hiện khi người dùng thực hiện những thao tác quan trọng.

Graphical user interface, application

Description automatically generated

**Hình 4.8** Thiết kế hộp thoại xác nhận

##### Cảnh báo

Cảnh báo được hiển thị khi người dùng thực hiện sai yêu cầu của phần mềm.

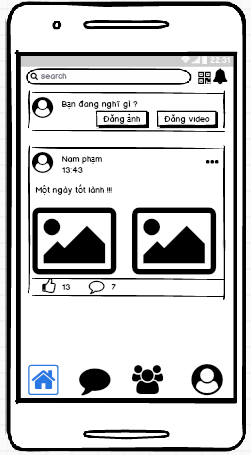
Graphical user interface, application

Description automatically generated

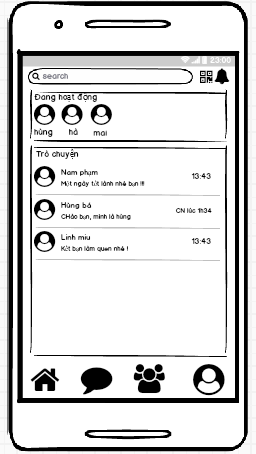
**Hình 4.9** Thiết kế cảnh báo nhập liệu

##### Thiết kế mockup giao diện

Sau đây là thiết kế mockup cho một số màn hình đặc trưng trong phần mềm



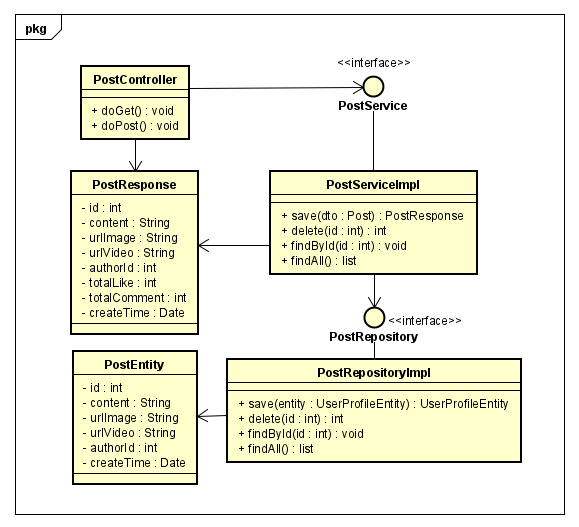
**Hình 4.10** Thiết kế mockup màn hình trang chủ



**Hình 4.11** Thiết kế mockup màn hình nhắn tin

### Thiết kế lớp

#### Thành phần quản lý bài viết



**Hình 4.12** Thiết kế chi tiết một số lớp cho chức năng quản lý bài viết

Hình 4.12 thể hiện một số lớp chủ đạo cho chức năng quản lý bài viết, chi tiết hơn về các phương thức và thuộc tính:

**Bảng 4.1** Bảng đặc tả lớp PostResponse

| Danh sách thuộc tính | | |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| id | int | ID bài viết |
| content | string | Nội dung bài viết |
| urlImage | string | Đường dẫn ảnh bài viết |
| urlVideo | string | Đường dẫn video bài viết |
| authorId | id | ID tác giả |
| totalLike | int | Tổng lượt thích bài viết |
| totalComment | int | Tổng lượt bình luận bài viết |
| createTime | timestamp | Thời gian tạo bài viết |

**Bảng 4.2** Bảng đặc tả lớp PostEntity

| Danh sách thuộc tính | | |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| id | int | ID bài viết |
| content | string | Nội dung bài viết |
| urlImage | string | Đường dẫn ảnh bài viết |
| urlVideo | string | Đường dẫn video bài viết |
| authorId | id | ID tác giả |
| createTime | timestamp | Thời gian tạo bài viết |

**Bảng 4.3** Bảng đặc tả lớp PostServiceImpl

| Danh sách phương thức | | | |
| --- | --- | --- | --- |
| Tên phương thức | Đầu vào | Đầu ra | Ý nghĩa |
| findAll | - | List<PostDto> | Trả về danh sách tất cả bài viết |
| findByID | postId | PostResponse | Tìm Bài viết theo Id |
| save | Post | PostResponse | Lưu thông tin bài viết |
| delete | postId | int | Xóa bài viết  (trả về 1 nếu thành công) |

**Bảng 4.4** Bảng đặc tả lớp PostRepositoryImpl

| Danh sách phương thức | | | |
| --- | --- | --- | --- |
| Tên phương thức | Đầu vào | Đầu ra | Ý nghĩa |
| findAll | - | List<PostEntity> | Trả về danh sách tất cả bài viết |
| findByID | postId | PostEntity | Tìm Bài viết theo Id |
| save | PostEntity | PostEntity | Lưu thông tin bài viết |
| delete | postId | int | Xóa bài viết  (trả về 1 nếu thành công) |

**Bảng 4.5** Bảng đặc tả lớp PostController

| Danh sách phương thức | | | |
| --- | --- | --- | --- |
| Tên phương thức | Đầu vào | Đầu ra | Ý nghĩa |
| doGet | request, repons e | - | Xử lý các yêu cầu gửi tới dạng  GET |
| doPost | request, reponse | - | Xử lý các yêu cầu gửi tới dạng POST |

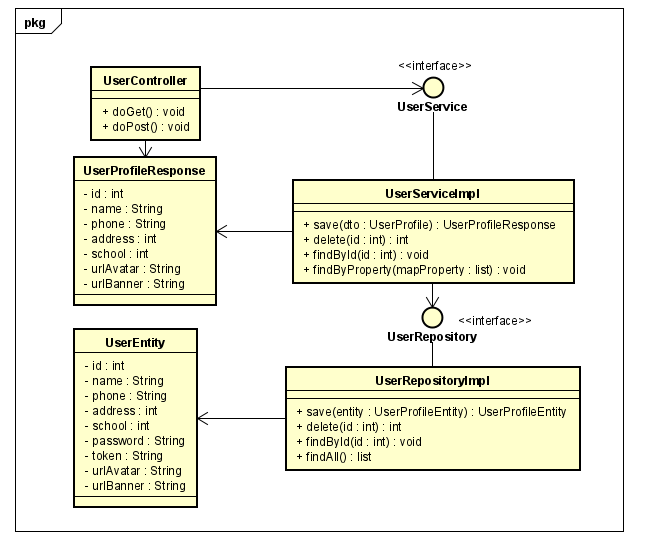
Dưới đây là biểu đồ trình tự cho use case xem danh sách bài viết:

Diagram

Description automatically generated

**Hình 4.13** Biểu đồ trình tự cho chức năng xem danh sách bài viết

#### Thành phần quản lý thông tin cá nhân



**Hình 4.14** Thiết kế chi tiết một số lớp cho chức năng quản lý thông tin cá nhân

Hình 4.14 thể hiện một số lớp chủ đạo cho chức năng quản lý thông tin cá nhân, chi tiết hơn về các phương thức và thuộc tính:

**Bảng 4.6** Bảng đặc tả lớp UserProfileResponse

| Danh sách thuộc tính | | |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| id | int | ID người dùng |
| name | String | Tên người dùng |
| phone | String | Số điện thoại người dùng |
| address | String | Địa chỉ người dùng |
| school | String | Học vấn người dùng |
| urlAvatar | String | Đường dẫn ảnh cá nhân người dùng |
| urlBanner | String | Đường dẫn ảnh bìa người dùng |

**Bảng 4.7** Bảng đặc tả lớp UserEntity

| Danh sách thuộc tính | | |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa |
| id | int | ID người dùng |
| name | String | Tên người dùng |
| phone | String | Số điện thoại người dùng |
| address | String | Địa chỉ người dùng |
| school | String | Học vấn người dùng |
| password | String | Mật khẩu người dùng |
| token | String | Phiên đăng nhập người dùng |
| urlAvatar | String | Đường dẫn ảnh cá nhân người dùng |
| urlBanner | String | Đường dẫn ảnh bìa người dùng |

**Bảng 4.8** Bảng đặc tả lớp UserServiceImpl

| Danh sách phương thức | | | |
| --- | --- | --- | --- |
| Tên phương thức | Đầu vào | Đầu ra | Ý nghĩa |
| findAll | - | List<PostResponse> | Trả về danh sách tất cả người dùng |
| findByProperty | Map<String> | List< PostResponse > | Tìm người dùng theo các thuộc tính (sử dụng phương thức tìm kiểm mặc định) |
| findByID | posrtId | PostResponse | Tìm người dùng theo Id |
| save | Post | PostResponse | Lưu thông tin người dùng |
| delete | postId | int | Xóa người dùng  (trả về 1 nếu thành công) |

**Bảng 4.9** Bảng đặc tả lớp UserRepositoryImpl

| Danh sách phương thức | | | |
| --- | --- | --- | --- |
| Tên phương thức | Đầu vào | Đầu ra | Ý nghĩa |
| findAll | - | List<PostEntity> | Trả về danh sách tất cả người dùng |
| findByID | posrtId | PostEntity | Tìm người dùng theo Id |
| save | PostEntity | PostEntity | Lưu thông tin người dùng |
| delete | postId | int | Xóa người dùng  (trả về 1 nếu thành công) |

**Bảng 4.10** Bảng đặc tả lớp UserController

| Danh sách phương thức | | | |
| --- | --- | --- | --- |
| Tên phương thức | Đầu vào | Đầu ra | Ý nghĩa |
| doGet | request, repons e | - | Xử lý các yêu cầu gửi tới dạng GET |
| doPost | request, reponse | - | Xử lý các yêu cầu gửi tới dạng POST |

Dưới đây là biểu đồ trình tự cho use case xem thông tin cá nhân:

Diagram

Description automatically generated

**Hình 4.15** Biểu đồ trình tự cho use case xem thông tin cá nhân.

### Thiết kế cơ sở dữ liệu

Trong phần mềm này, bao gồm các thực thể:

• Thực thể dùng cho quản lý thông tin người dùng: user.

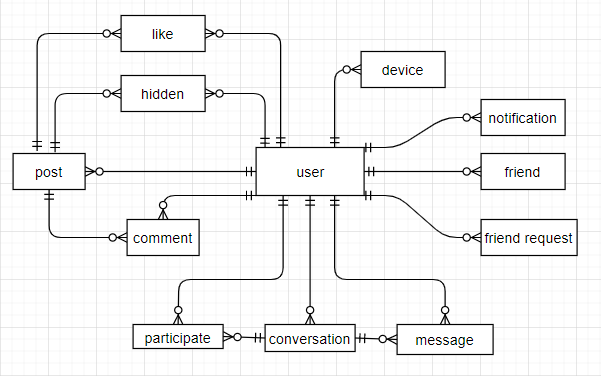
• Thực thể dùng cho nhắn tin, tạo nhóm nhắn tin: conversation, message, participate.

• Thực thể dùng cho quản lý danh sách bạn bè, yêu cầu kết bạn: friend, friend request, notification.

• Thực thể dùng cho quản lý bài viết: post, like, hidden, comment.

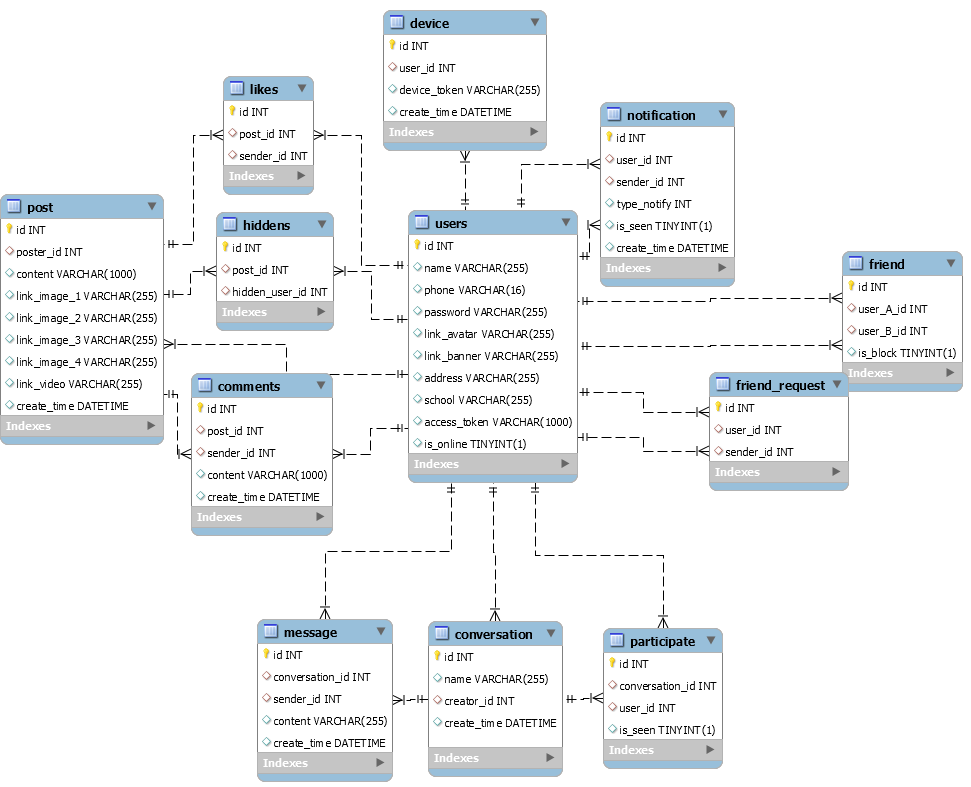
• Thực thể dùng để lưu trữ thông tin thiết bị: device.

Hình 4.16 trình bày chi tiết hơn về các thực thể và mối quan hệ giữa chúng.



**Hình 4.16** Biểu đồ thực thể liên kết (ER).

Từ biểu đồ thực thể liên kết trong Hình 4.16, cơ sở dữ liệu được thiết kế sử dụng MySQL được trình bày trong Hình 4.17



**Hình 4.17** Thiết kế cơ sở dữ liệu.

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

**Bảng 4.11** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | IntelliJ IDEA 2020.3 | https://www.jetbrains.com |
| Ngôn ngữ lập trình | Java 8 | http://www.oracle.com |
| Quản trị CSDL | MySql 8.0.16 | https://www.mysql.com/ |
| Công cụ quản lý | Maven 3.6.0 | https://maven.apache.org/ |
| Framework | Spring Boot 2.2 | https://spring.io/spring-boot |
| Vẽ biểu đồ UML | Astah Uml 8.2 | http://astah.net |
| Vẽ biểu đồ thiết kế | Draw IO | https://www.draw.io |
| Vẽ thiết kế mokup | Balsamiq mockup 3 | https://balsamiq.com/ |

### Kết quả đạt được

Sau thời gian thực hiện đồ án đã xây dựng thành công Ứng dụng mạng xã hội kết nối cộng đồng. Sản phẩm bao gồm:

• Ứng dụng android đóng gói dạng .apk

• Ứng dụng java spring boot đóng gói dạng .jar

• Dữ liệu cơ sở dữ liệu.

• Hướng dẫn sử dụng, triển khai.

Thống kê chi tiết hơn về phần mềm được trình bày trong Bảng 4.12

**Bảng 4.12** Thống kê thông tin ứng dụng

| Thông tin | Thống kê |
| --- | --- |
| Số lớp trong mã nguồn | 81 class java |
| Số gói trong mã nguồn | 15 gói |
| Số file giao diện android (xml) | 30 file xml |
| Dung lượng mã nguồn | 82 Mb |
| Dung lượng file deploy | 44.4 Mb |
| Số bảng trong cơ sở dữ liệu | 12 bảng |

### Minh hoạ các chức năng chính

Graphical user interface, text, application

Description automatically generated

**Hình 4.18** Trang chủ.

Trong giao diện màn hình trang chủ, người dùng sẽ xem được danh sách các bài viết của mình và bạn bè, người dùng có thể quét mã QR tìm bạn bè qua icon góc phải trên cùng hoặc tìm kiếm thủ công qua thanh tìm kiếm.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Hình 4.19** Trang nhắn tin.

Trong giao diện màn hình nhắn tin, người dùng sẽ xem được danh sách các tin nhắn của mình và bạn bè, người dùng có thể chọn một dòng tin nhắn bất kì để gửi tin nhắn đến bạn bè.

Graphical user interface, application

Description automatically generated

**Hình 4.20** Trang bạn bè.

Trong giao diện màn hình bạn bè, người dùng sẽ xem được danh sách bạn bè của mình và danh sách yêu cầu kết bạn. Người dùng có thể thao tác với từng dòng thông qua nút ấn gợi ý sẵn.

Graphical user interface, application, website

Description automatically generated

**Hình 4.21** Trang cá nhân.

Trong giao diện màn hình thông tin cá nhân, người dùng sẽ xem được thông tin cá nhân của mình hoặc bạn bè và danh sách bài viết của tác giả thông tin cá nhân. Người dùng có thể thao tác chỉnh sửa thông tin công khai của mình (nếu trang cá nhân của mình) hoặc yêu cầu kết bạn với cá nhân (nếu trang cá nhân của người dùng khác.)

## Kiểm thử

### Compatibility Testing

Kiểm tương thích (compatibility testing) nhằm đảm bảo ứng dụng được viết ra hoạt động tốt trên nhiều thiết bị di động khác nhau: phần cứng, hệ điều hành android các phiên bản, hãng máy, …

**Bảng 4.13** Bảng kiểm thử khả năng tương thích

|  |  |  |  |
| --- | --- | --- | --- |
| Thiết bị | Phần cứng | Hệ điều hành | Kết quả |
| Nokia 6.1 plus | CPU: qualcomm snapdragon 636  Màn hình: 5.87” fullHD | Android 10 | Đạt |
| Nokia 5.1 plus | CPU: mediatek P60  Màn hình: 5.84” HD+ | Android 8.0 | Đạt |
| Samsung Galaxy S9 | CPU: Exynos 9810 RAM: 4GB  Màn hình: 5.83” 2K | Android 9 | Đạt |
| Xiaomi Redmi Note 2 | CPU: MediaTek MT6795  RAM: 2GB  Màn hình: 5.5” HD | Android 4.4 | Không đạt  (Không nhận thông báo đẩy) |

### Black box Testing

Kiểm thử hộp đen (black box testing) là phương pháp kiểm thử phần mềm mà không quan tâm đến cấu trúc, dòng lệnh chương trình mà dựa vào đặc tả và chức năng đã thiết kế để lên kế hoạch tạo ca kiểm thử phù hợp. Sau đây là các ca kiểm thử hộp đen em đã thiết kế cho một số chức năng quan trọng.

**Bảng 4.14** Bảng kiểm thử hộp đen cho chức năng “Đăng nhập”

|  |  |  |
| --- | --- | --- |
| Giá trị đầu vào | Kết quả mong muốn | Kết quả kiểm thử |
| Bỏ trống số điện thoại | Hiển thị ảnh báo: “số điện thoại chưa nhập” | Đạt |
| Bỏ trống password | Hiển thị cảnh báo: “Mật khẩu chưa nhập” | Đạt |
| Sai định dạng username | Hiển thị cảnh báo: “Số điện thọai sai định dạng” | Đạt |
| Sai định dạng password | Hiển thị cảnh báo: “mật khẩu sai định dạng” | Đạt |
| Đúng định dạng số điện thại và password nhưng user không tồn tại | Hiển thị hộp thoại: “Tài khoản không tồn tại” | Đạt |
| Đúng định dạng username và password và user tồn tại | Chuyển đến trang chủ | Đạt |

**Bảng 4.15** Bảng kiểm thử hộp đen cho chức năng “Đổi mật khẩu”

|  |  |  |
| --- | --- | --- |
| Giá trị đầu vào | Kết quả mong muốn | Kết quả kiểm thử |
| Bỏ trống trường mật khẩu cũ | Hiển thị cảnh báo: “chưa nhập” | Đạt |
| Bỏ trống trường mật khẩu mới | Hiển thị cảnh báo: “chưa nhập” | Đạt |
| Bỏ trống trường xác nhận mật khẩu | Hiển thị cảnh báo: “chưa nhập” | Đạt |
| Sai định dạng mật khẩu cũ | Hiển thị cảnh báo: “sai định dạng” | Đạt |
| Sai định dạng mật khẩu mới | Hiển thị cảnh báo: “sai định dạng” | Đạt |
| Sai định dạng xác nhận mật khẩu | Hiển thị cảnh báo: “sai định dạng” | Đạt |
| Đúng định dạng mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu nhưng mật khẩu cũ không đúng | Hiển thị cảnh báo: “Sai mật khẩu” | Đạt |
| Đúng định dạng mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu nhưng mật khẩu mới trùng mật khẩu cũ | Hiển thị cảnh báo: “Mật khẩu chưa thay đổi” | Đạt |
| Đúng định dạng mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu và mật khẩu cũ đúng | Hiển thị thông báo: “Đổi mật khẩu thành công” | Đạt |

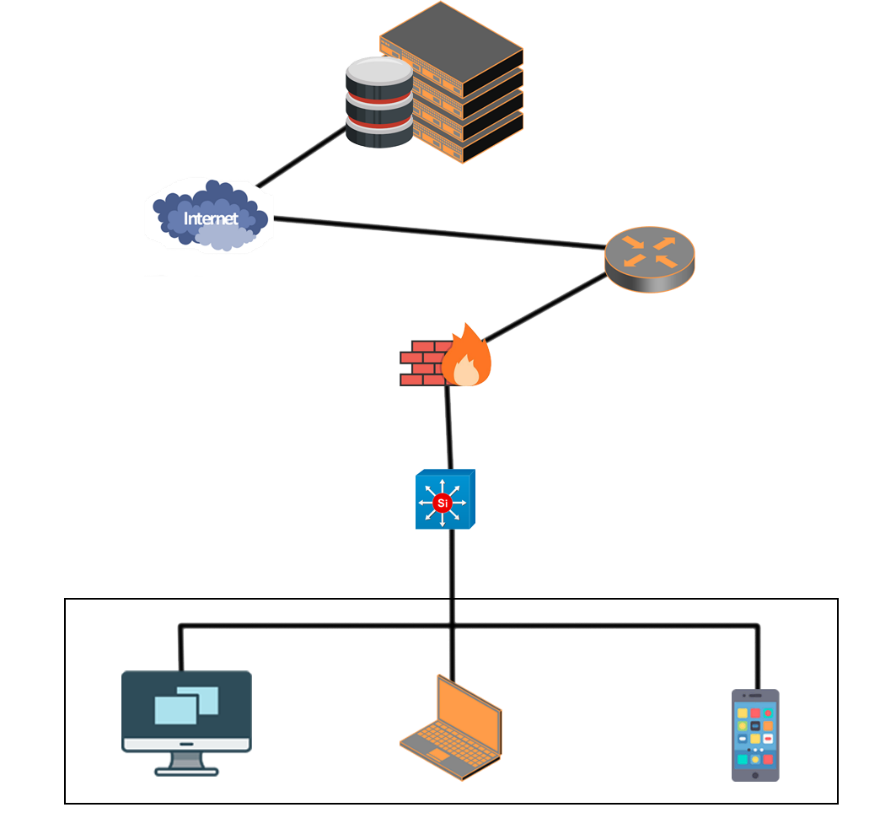
### Tổng kết kiểm thử

Như vậy, trong tất cả các trường hợp kiểm thử sử dụng điện thoại di động android các phiên bản thì phần mềm đều đạt. Với các trường hợp không đạt vì các điện thoại phiên bản android quá cũ so với công nghệ đang sử dụng.

## Triển khai

**Bảng 4.16** Bảng danh sách yêu cầu thiết bị và công cụ triển khai.

|  |  |
| --- | --- |
| Loại thiết bị/ công cụ | Yêu cầu |
| Hệ điều hành | Hệ điều hành mã nguồn mở nhân linux:   * Ubuntu 14.01 LTS trở lên |
| CPU | Bộ xử lý tối thiểu: 2 nhân 1.7GHz |
| GPU | Không yêu cầu |
| Bộ nhớ trong | Ram ít nhất 1GB (khuyến nghị ≥ 1.5GB) |
| Kết nối | Hỗ trợ kết nối băng thông rộng (≥ 8Mbps) |
| Ổ đĩa | Còn trống ít nhất 5GB lưu trữ phần mềm |
| Hệ quản trị cơ sở dữ liệu | MySQL 8 |
| Máy chủ web | Tomcat 8.0 |



**Hình 4.22** Sơ đồ triển khai hệ thống

Hệ thống bao gồm một máy chủ đặt trực tiếp tại chỗ. Dữ liệu sẽ được lưu trữ trong CSDL chung của hệ thống và được sao lưu phục hồi định kỳ thông qua một máy chủ backup

Ưu điểm:

* Tiết kiệm được chi phí hàng tháng do không thuê máy chủ sẵn và thuê cloud.
* Chủ động trong việc sửa chữa bảo trì và nâng cấp.
* Ổn định do không phụ thuộc nhiều vào các yếu tố tác động bên ngoài.
* Có khả năng sao lưu lại và khôi phục các dữ liệu quan trọng.
* Tính bảo mật cao
* Nhược điểm:
* Chi phí xây dựng ban đầu khá cao.
* Có thêm phòng ban chuyên phụ trách cài đặt.
* Tốc độ đường truyền chậm nếu thiết bị/công cụ là tối thiểu.
* Khả năng bảo mật chưa cao.
* Khắc phục:
* Cần thường xuyên cập nhật các công nghệ mới,
* Luôn có một đội ngũ kỹ thuật thức trực có đầy đủ khả năng để bảo trì hệ thống.

# Các giải pháp và đóng góp nổi bật

Tổng quan: trong chương này, sẽ trình bày những đóng góp nổi bật của cá nhân trong việc: xây dựng chức năng nhắn tin thời gian thực và đẩy thông báo, Xác thực người dùng trong ứng dụng với giao thức tiêu chuẩn OAuth 2.0 và triển khai.

## Xây dựng chức năng nhắn tin thời gian thực (real-time) và đẩy thông báo

### Bài toán đặt ra

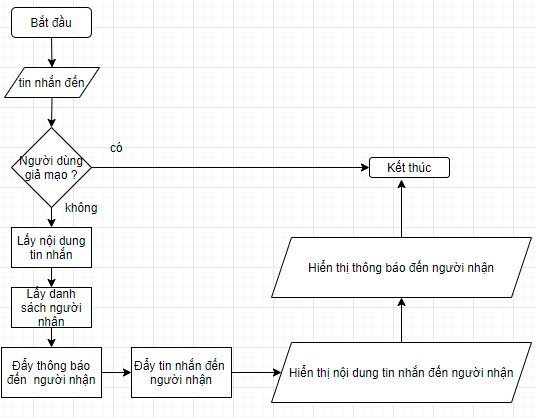
Với chức năng nhắn tin trong ứng dụng, mỗi khi người gửi gửi tin nhắn cho người nhận hoặc nhóm người nhận nào đó thì hệ thống phải đáp ứng chuyển tiếp tin nhắn của người gửi một cách nhanh nhất và đồng thời phải đẩy thông báo về ứng dụng của các thiết bị người nhận để họ có thể biết nội dung tin nhắn nếu máy của họ thoát ứng dụng. Ở đây em có hai vấn đề cần giải quyết: (1) Gửi tin nhắn đáp ứng độ trễ thấp (nhỏ hơn 10 mili giây); (2) đẩy thông báo cho các thiết bị nhận được tin nhắn.

### Giải pháp

Với vấn đề thứ nhất: gửi tin nhắn đáp ứng độ trễ thấp (nhỏ hơn 10 mili giây), em sử dụng lý thuyết về web socket để giải quyết vấn đề. Web soket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào. Dữ liệu truyền tải thông qua giao thức chứa nhiều dữ liệu không cần thiết trong phần header. Một header request/response của HTTP có kích thước khoảng 871 byte, trong khi với WebSocket, kích thước này chỉ là 2 byte (sau khi đã kết nối). Vì vậy nó rất được ưa chuộng trong các ứng dụng đòi hỏi đáp ứng độ trễ thấp như nhắn tin, thu thập dữ liệu, ….

Với vấn đề thứ hai: đẩy thông báo cho các thiết bị nhận được tin nhắn, em sử dụng dịch vụ miễn phí là dịch vụ Nhắn tin qua đám mây - Firebase Cloud Messaging. Với Firebase Cloud Messaging, ta có thể gửi nhận tin nhắn miễn phí. Khái niệm "Gửi tin nhắn" ở đây dùng để chỉ việc đẩy thông báo tới thiết bị nhận.

Quá trình triển khai chức năng nhắn tin được trình bày trong hình 5.1:



**Hình 5.1** Quy trình nhắn tin

### Kết quả đạt được

Xây dựng thành công chức năng nhắn tin ứng dụng web socket và firebase cloud messaging. Nhờ công cụ này đã giải quyết được bài toán nhắn tin đáp ứng thời gian thực và đẩy thông báo cho các thiết bị nhận. Sau đây là một số kết quả đạt được minh họa cho chức năng:

Graphical user interface, text, application

Description automatically generated A picture containing diagram

Description automatically generated

**Hình 5.2** Minh họa nhắn tin thời gian thực

Graphical user interface, text, application

Description automatically generated A picture containing graphical user interface

Description automatically generated

**Hình 5.3** Minh họa đẩy thông báo

## Xác thực người dùng với giao thức bảo mật OAuth 2.0 và triển khai

### Bài toán đặt ra

Với những hệ thống thông thường, nếu bên thứ ba nào đó tấn công hệ thống bằng cách thực hiện yêu cầu đến hệ thống qua những đường dẫn bị lộ, bên thứ ba đó có thể đánh cắp các thông tin trong hệ thống đó. Để tăng tính bảo mật cho hệ thống, bài toán đặt ra là cần một giao thức tiêu chuẩn bảo mật giữa người dùng cơ bản và hệ thống để tránh kẻ tấn công lấy cắp dữ liệu bất cứ lúc nào.

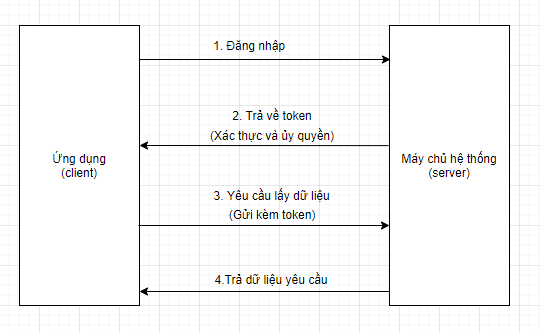
### Giải pháp

Với yêu cầu bảo mật trong hệ thống, sử dụng giao thức OAuth 2.0. Nó là viết tắt của Open với Authentication (xác thực) hoặc Authorization (ủy quyền). OAuth ra đời nhằm giải quyết vấn đề trên và xa hơn nữa, đây là một phương thức chứng thực giúp các ứng dụng có thể chia sẻ tài nguyên với nhau mà không cần chia sẻ thông tin tài khoản và mật khẩu.

Mỗi khi người dùng đăng nhập hệ thống, hệ thống cấp cho người dùng một mã xác thực duy nhất (token) để định danh và ủy quyền cho người dùng đó. Khi thực hiện yêu cầu đến hệ thống, người dùng chỉ cần gửi mã xác thực này để hệ thống biết và trả về kết quả. Vì vậy, nếu bên thứ ba nào đó không có mã xác thực này, sẽ không thể nào truy cập tài nguyên hệ thống. Quy trình xử lý diễn ra như sau:

1. Ứng dụng yêu cầu ủy quyền để truy cập vào tài nguyên hệ thống thông qua chức năng đăng nhập.
2. Nếu người dùng đăng nhập thành công, hệ thống sẽ trả về cho ứng dụng một mã xác thực duy nhất (token). Đến đây quá trình ủy quyền hoàn tất.
3. Để truy cập vào tài nguyên từ hệ thống và lấy thông tin, Ứng dụng sẽ phải đưa ra token để xác thực.
4. Nếu token hợp lệ, hệ thống sẽ trả về dữ liệu của tài nguyên đã được yêu cầu cho ứng dụng.

Quá trình triển khai giao thức OAuth được sử dụng trong hệ thống được trình bày trong hình 5.4 như sau:



**Hình 5.4** Quy trình gửi nhận dữ liệu thông qua giao thức OAuth

### Kết quả đạt được

Xây dựng thành công hệ thống thông qua giao thức chuẩn OAuth. Nhờ giao thức này mà hệ thống đã giải quyết được bài toán bảo mật so với các hệ thống thông thường không sử dụng các giao thức bảo mật tiêu chuẩn.

# Kết luận và hướng phát triển

## Kết luận

Sau một quá trình nghiên cứu làm việc và dưới sự hướng dẫn tận tình của ThS. Lê Đức Trung, em đã xây dựng thành công ứng dụng mạng xã hội kết nối cộng đồng Ntalk. Sản phẩm phần mềm bao gồm những chức năng chính sau: (1) Quản lý thông tin cá nhân của người dùng; (2) Quản lý danh sách bài viết của người dùng; (3) Quản lý danh sách bạn bè của người dùng và (4) Nhắn tin.

Ứng dụng mạng xã hội đã được xây dựng là sự kế thừa và phát triển của các ứng dụng mạng xã hội nổi tiếng đã có trong thị trường. Ứng dụng cung cấp cho người dùng thêm một lựa chọn về một mạng xã hội mới thông minh và tiện lợi hơn so với các mạng xã hội truyền thống đã có.

Với sản phẩm của mình, em tin chắc ứng dụng của mình thực sự cần thiết cho mọi đối tượng người dùng. Ứng dụng giúp giảm thiểu tối đa thời gian cho các nghiệp vụ tìm kiếm thông tin và quản lý bạn bè so với các ứng dụng tiền nhiệm.

Trong quá trình thực hiện ĐATN, em đã đúc rút ra được nhiều kinh nghiệm cho bản thân về xây dựng và phát triên phần mềm, các kỹ năng mềm như: quản lý thời gian, kỹ năng về thiết kế và xây dựng phần mềm, các kỹ năng về lập trình và kiểm thử phần mềm, làm báo cáo và kỹ năng về làm slide thuyết trình.

## Hướng phát triển

Ngoài nhưng chức năng đã thực hiện trong phạm vi của ĐATN thì với yêu cầu của một phần mềm ứng dụng mạng xã hội còn phải bổ sung thêm những chức năng như: chức năng chia sẻ bài viết để tăng lượng tương tác của người dùng, chức năng đổi tên gợi nhớ nhóm nhắn tin, chức năng gợi ý người dùng có thể bạn biết, …

Để tiếp tục hoàn thiện hơn sản phẩm phần mềm của mình thì em cần phải phân tích và tìm hiểu kỹ thêm các sản phẩm phần mềm đã có hiện nay dùng cho việc quản lý nhắn tin, các thuật toán về học máy, hệ gợi ý cho việc gợi ý người dùng. Trực tiếp tới các môi trường sử dụng phần mềm để khảo sát các nghiệp vụ. Từ đó làm tiền đề để phát triển thêm các chức năng chưa có, cải tiến, tinh chỉnh các chức năng đã xây dựng để phù hợp với yêu cầu sử dụng trong thực tế.

Với các chức năng đã xây dựng, phần mềm đã giúp đỡ giải quyết nhiều vấn đề trong việc tìm kiếm thông tin và quản lý bạn bè. Nếu phát triển đúng hướng, sản phẩm sẽ thực sự cần thiết cho bất kỳ ai quan tâm tới một mạng xã hội kiểu mới thông minh và tiện lợi.

# Tài liệu tham khảo

[1] Three-Layer Services Application, [https://docs.microsoft.com](https://docs.microsoft.com/en-us/), last visited May 2021.

[2] Mysql, <https://www.mysql.com/>, last visited May 2021.

[3] OAuth 2.0, <https://oauth.net/2>, last visited May 2021.