



**SEMINAR**

Môn học: CÔNG NGHỆ JAVA

Đề tài: SPRING FRAMEWORK

Giảng viên hướng dẫn: **ThS. Nguyễn Văn Kiệt**

Thành viên:

|  |  |  |
| --- | --- | --- |
| Phạm Nhật Minh | - | 18520102 |
| Nguyễn Thành Đạt | - | 18520580 |

**TP Hồ Chí Minh tháng 5 năm 2021**

MỤC LỤC

[DANH MỤC HÌNH ẢNH 4](#_Toc75616217)

[NHẬN XÉT CỦA GIẢNG VIÊN 5](#_Toc75616218)

[CHƯƠNG 1: MỞ ĐẦU 7](#_Toc75616219)

[TỔNG QUAN ĐỀ TÀI 7](#_Toc75616220)

[1. Giới thiệu đề tài 7](#_Toc75616221)

[2. Lý do chọn đề tài 7](#_Toc75616222)

[3. Mục tiêu của đề tài 7](#_Toc75616223)

[4. Ý nghĩa của đề tài 7](#_Toc75616224)

[5. Nội dung thực hiện 7](#_Toc75616225)

[6. Lời cảm ơn 8](#_Toc75616226)

[II. CƠ SỞ LÝ THUYẾT 9](#_Toc75616227)

[A. Tổng quan về mạng xã hội 9](#_Toc75616228)

[1. Mạng xã hội là gì? 9](#_Toc75616229)

[2. Lịch sử hình thành 9](#_Toc75616230)

[3. Đặc điểm của mạng xã hội 9](#_Toc75616231)

[4. Mục tiêu của mạng xã hội 9](#_Toc75616232)

[5. Một số ứng dụng phân tích mạng xã hội: 10](#_Toc75616233)

[B. Cấu trúc mạng xã hội 10](#_Toc75616234)

[1. Tổng quan về cấu trúc 10](#_Toc75616235)

[2. Dữ liệu trong cấu trúc mạng xã hội 10](#_Toc75616236)

[C. Công nghệ sử dụng cho đồ án 11](#_Toc75616237)

[1. Lý do sử dụng framework và design pattern 11](#_Toc75616238)

[2. Giới thiệu về công nghệ và mẫu thiết kế 12](#_Toc75616239)

[3. Ứng dụng công nghệ vào đồ án 13](#_Toc75616240)

[III. Phân tích và thiết kế 15](#_Toc75616241)

[A. Xây dựng kiến trúc dự án 15](#_Toc75616242)

[B. Xác định chức năng có thể hiện thực hóa, xây dựng use case 16](#_Toc75616243)

[C. Danh sách người dùng và mô tả hệ thống 17](#_Toc75616244)

[1. Danh sách người dùng 17](#_Toc75616245)

[2. Danh sách use-case và mô tả 17](#_Toc75616246)

[D. Các sơ đồ và luồng xử lý hệ thống 19](#_Toc75616247)

[1. Đăng ký 19](#_Toc75616248)

[2. Đăng nhập 20](#_Toc75616249)

[3. Tìm kiếm: 21](#_Toc75616250)

[4. Newfeed 22](#_Toc75616251)

[5. Kết bạn: 22](#_Toc75616252)

[6. Chấp nhận kết bạn: 23](#_Toc75616253)

[7. Tìm bạn: 24](#_Toc75616254)

[8. Lấy danh sách bạn bè 26](#_Toc75616255)

[9. Đăng bài: 27](#_Toc75616256)

[10. Xem thông tin tài khoản 28](#_Toc75616257)

[11. Chỉnh sửa thông tin tài khoản: 29](#_Toc75616258)

[E. Thiết kế dữ liệu: 29](#_Toc75616259)

[F. THIẾT KẾ GIAO DIỆN: 31](#_Toc75616260)

[1. Giao diện trang đăng ký 31](#_Toc75616261)

[2. Giao diện trang đăng nhập 31](#_Toc75616262)

[3. Giao diện Newfeed 32](#_Toc75616263)

[4. Giao diện trang cá nhân 32](#_Toc75616264)

[5. Giao diện thay đổi thông tin tài khoản 33](#_Toc75616265)

[6. Giao diện thay đỏi thông tin cá nhân 33](#_Toc75616266)

[7. Giao diện trang bạn bè 34](#_Toc75616267)

[IV. HIỆN THỰC 35](#_Toc75616268)

[A. Chuẩn viết mã 35](#_Toc75616269)

[B. Cách tổ chức mã nguồn 35](#_Toc75616270)

[1. Mô hình MVC: 36](#_Toc75616271)

[2. Áp dụng mô hình MVC cùng Service và Repository 36](#_Toc75616272)

[3. Sơ đồ luồng đi: 37](#_Toc75616273)

[V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38](#_Toc75616274)

[VI. BẢNG PHÂN CÔNG 38](#_Toc75616275)

[VII. TÀI LIỆU THAM KHẢO 38](#_Toc75616276)

# DANH MỤC HÌNH ẢNH

[Hình II‑1Thị phần của các framework 11](#_Toc75616277)

[Hình II‑2Minh họa Service và Repository 14](#_Toc75616278)

[Hình III‑1Kiến trúc phần mềm 15](#_Toc75616279)

[Hình III‑2 Use-case 17](#_Toc75616280)

[Hình III‑3 Luồng xử lý đăng ký 19](#_Toc75616281)

[Hình III‑4 Luồng xử lý đăng nhập 20](#_Toc75616282)

[Hình III‑5Luồng xử lý tìm kiếm 21](#_Toc75616283)

[Hình III‑6Luồng xử lý Newfeed 22](#_Toc75616284)

[Hình III‑7Luồng xử lý kết bạn 23](#_Toc75616285)

[Hình III‑8Chấp nhận kết bạn 24](#_Toc75616286)

[Hình III‑9 Luồng xử lý tìm bạn 25](#_Toc75616287)

[Hình III‑10Luồng xử lý danh sách bạn bè 26](#_Toc75616288)

[Hình III‑11 Luồng xử lý đăng bài 27](#_Toc75616289)

[Hình III‑12Luồng xử lý xem thông tin tài khoản 28](#_Toc75616290)

[Hình III‑13Luồng xử lý chỉnh sửa thông tin tài khoản 29](#_Toc75616291)

[Hình III‑14Giao diện trang đăng ký 31](#_Toc75616292)

[Hình III‑15Giao diện trang đăng nhập 31](#_Toc75616293)

[Hình III‑16Giao diện Newfeed 32](#_Toc75616294)

[Hình III‑17Giao diện trang cá nhân 32](#_Toc75616295)

[Hình III‑18Giao diện thay đổi thông tin tài khoản 33](#_Toc75616296)

[Hình III‑19Giao diện thay đổi thông tin cá nhân 33](#_Toc75616297)

[Hình III‑20Giao diện trang bạn bè 34](#_Toc75616298)

[Hình IV‑1Cách tổ chức mã nguồn 35](#_Toc75616299)

[Hình IV‑2 Sơ đồ luồng đi 37](#_Toc75616300)

# NHẬN XÉT CỦA GIẢNG VIÊN

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

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

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....

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

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

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

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

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

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

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

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

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

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

# CHƯƠNG 1: MỞ ĐẦU

## TỔNG QUAN ĐỀ TÀI

### Giới thiệu đề tài

Mạng xã hội DM là một mạng xã hội đơn giản gồm một số chức năng cơ bản của mạng xã hội như: Đăng bài, kết bạn, xem trang cá nhân,… người dùng sẽ có thể trải nghiệm và tận hưởng mạng xã hội này.

### Lý do chọn đề tài

Sau thành công của Facebook mạng xã hội phát triển vượt bậc với nhiều start up kỳ lân như Twitter,Youtube,… Ngày nay, các doanh nghiệp đang dần muốn “mạng xã hội hóa” trang web của mình. Việt Nam cũng đang muốn xây dựng mạng xã hội riêng để không phụ thuộc vào các mạng xã hội bên ngoài.

### Mục tiêu của đề tài

Nhóm thực hiện đề tài với những mục tiêu như sau:

* Hiểu được mạng xã hội và cơ chế vận hành cơ bản của nó để có thể giúp cho sự nghiệp của bản thân sau này cũng như nền công nghệ thông tin của đất nước.
* Xây dựng một mạng xã hội đơn giản để làm bước đà tạo 1 framework xây dựng nhanh mạng xã hội cho các doanh nghiệp sau này.

### Ý nghĩa của đề tài

Mạng xã hội giúp cho người dùng có nhiều trải nghiệm hơn so với các website thông thường. Tạo cơ hội cho người dùng với các sở thích khác nhau hay truy cập vào cùng một trang web có thể kết nối và chia sẻ các thông tin hữu ích. Nếu một doanh nghiệp chuyển được trang web thành mạng xã hội sẽ có thể tương tác với người dùng tốt hơn, hiểu người dùng hơn từ đó đưa ra được những hướng đi tốt cho doanh nghiệp của mình.

Đối với riêng cá nhân nhóm:

* Giúp trau dồi và nâng cao kiến thức lập trình
* Ứng dụng những gì đã học để tạo ra sản phẩm hữu ích phù hợp với xu hướng chung của thế giới

### Nội dung thực hiện

Báo cáo này gồm 5 chương chính:

Chương 1: Mở đầu

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích và thiết kế

Chương 4: Hiện thực

Chương 5: Kết luận và hướng phát triển

### Lời cảm ơn

Nhóm xin được cảm ơn thầy Nguyễn Văn Kiệt và hai anh trợ giảng Huỳnh Văn Tín và Nguyễn Thành Luân đã hỗ trợ về mặt thời gian để giúp nhóm có khả năng hoàn thành đồ án.Đồ án sẽ có những thiếu sót, kính mong thầy và hai trợ giảng bỏ qua.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về mạng xã hội

### Mạng xã hội là gì?

Mạng xã hội có thể hiểu là một trang web hay nền tảng trực tuyến với rất nhiều dạng thức và tính năng khác nhau, giúp mọi người dễ dàng kết nối từ bất cứ đâu.

### Lịch sử hình thành

* Mạng xã hội xuất hiện lần đầu tiên năm 1995 với sự ra đời của trang Classmate với mục đích kết nối bạn học, tiếp theo là sự xuất hiện của SixDegrees vào năm 1997 với mục đích giao lưu kết bạn dựa theo sở thích.
* Năm 2002, Friendster trở thành một trào lưu mới tại Hoa Kỳ với hàng triệu thành viên ghi danh. Tuy nhiên sự phát triển quá nhanh này cũng là con dao hai lưỡi: server của Friendster thường bị quá tải mỗi ngày, gây bất bình cho rất nhiều thành viên.
* Năm 2004, MySpace ra đời với các tính năng như phim ảnh (embedded video) và nhanh chóng thu hút hàng chục ngàn thành viên mới mỗi ngày, các thành viên cũ của Friendster cũng dần chuyển qua MySpace và trong vòng một năm, MySpace trở thành mạng xã hội đầu tiên có nhiều lượt xem hơn cả Google và được tập đoàn News Corporation mua lại với giá 580 triệu USD.
* Năm 2006, sự ra đời của Facebook đánh dấu bước ngoặt mới cho hệ thống mạng xã hội trực tuyến với nền tảng lập trình "Facebook Platform" cho phép thành viên tạo ra những công cụ (apps) mới cho cá nhân mình cũng như các thành viên khác dùng. Facebook Platform nhanh chóng gặt hái được thành công vược bậc, mang lại hàng trăm tính năng mới cho Facebook.

### Đặc điểm của mạng xã hội

Mạng xã hội tuy tồn tại dưới nhiều hình mô hình khác nhau nhưng nhìn chung, mạng xã hội đều có những điểm chung sau:

* Mạng xã hội là một ứng dụng được sử dụng trên nền tảng Internet.
* Tất cả nội dung trên mạng xã hội đều do người dùng tự tạo ra, tự chia sẻ.
* Mỗi người dùng trên mạng xã hội đều phải tạo tài khoản, hồ sơ riêng.
* Mạng xã hội sẽ kết nối tài khoản người dùng đến các tài khoản cá nhân, tổ chức khác thông qua các tài khoản ảo do người dùng tạo ra.

### Mục tiêu của mạng xã hội

Mục tiêu của mạng xã hội là tạo ra một hệ thống cho phép người dùng có thể kết nối, giao lưu, chia sẻ những thông hữu ích trên nền tảng Internet.

Ngoài ra, mạng xã hội còn có mục tiêu là tạo nên một cộng đồng có giá trị, nâng cao vai trò của mỗi người dùng trong việc xây dựng các mối quan hệ.

Với những gì mà mục tiêu đưa ra, mạng xã hội đã mang lại cho con người rất nhiều những lợi ích như: Cập nhật tin tức, kết nối các mối quan hệ, kinh doanh, quảng cáo trên mạng xã hội,...

Tuy nhiên, đi kèm với phát triển rộng rãi của mạng xã hội đó lại không tránh khỏi việc nhiều người đang sử dụng mạng xã hội sai cách làm ảnh hưởng xấu đến bản thân như mất ngủ, giảm sức khoẻ, suy nghĩ tiêu cực,...

### Một số ứng dụng phân tích mạng xã hội:

Khai phá dữ liệu, truy hồi thông tin, khoa học web, khoa học xã hội.

Quản lý quan hệ khoa học xã hội (Social CRM), tư vấn xã hội (Social recommendation), khai quan điểm (opinion mining), quản lý danh tiếng (reputation management),….

## Cấu trúc mạng xã hội

### Tổng quan về cấu trúc

Cấu trúc của mạng xã hội gồm 2 thành phần cơ sở để cấu tạo nên:

* Nút: tác nhân xã hội tham gia vào mạng xã hội
* Cạnh là quan hệ/tương tác giữa các tác nhân đó. “là một cấu trúc xã hội bao gồm các cá nhân hay tổ chức, thường được biểu diễn bằng các nút, cùng với các quan hệ xã hội, tương ứng với các liên kết giữa các nút”.

### Dữ liệu trong cấu trúc mạng xã hội

#### Dữ liệu nút:

Dữ liệu nút có thể bao gồm một hoặc nhiều thành phần sau đây:

* Hồ sơ cá nhân:
* Tên, tuổi, giới tính, trình trạng hôn nhân,…
* Nơi làm việc, nghề nghiệp,..
* Gia đình, các thành phần gia đình,..
* Sở thích: Ngắn hạn, trung hạn, dài hạn
* Hành vi/ thói quen: thói quen sinh hoạt, làm việc,…
* Ý định/ nhu cầu
* Cảm xúc
* Tính cách

#### Dữ liệu cạnh:

Dữ liệu cạnh có thể bao gồm một hoặc nhiều thành phần sau đây:

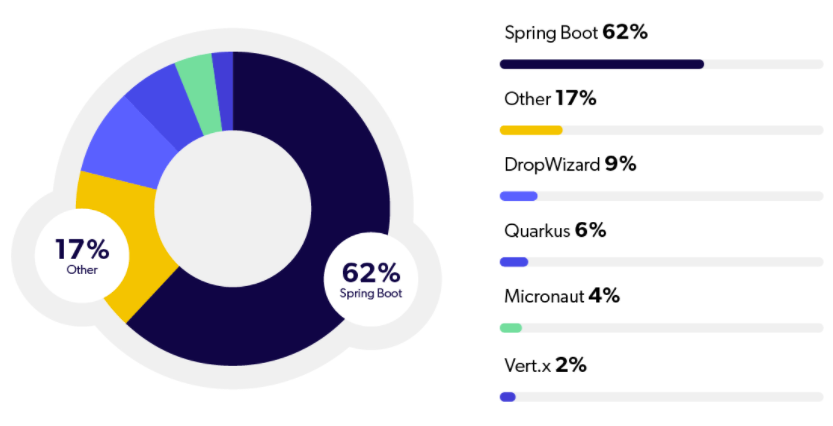
* Quan hệ 1 chiều:
* Fan theo dõi fanpage thần tượng
* Người dùng theo dõi tài khoản khác mà không được phản hồi,..
* Quan hệ 2 chiều:
* Quan hệ bạn bè,…
* Quan hệ đa chiều:
* Quan hệ cá nhân – nhóm/ tổ chức
* Quan hệ giữa các nhóm/ tổ chức với nhau
* Mạng bạn bè,…

## Công nghệ sử dụng cho đồ án

Để tạo mạng xã hội và phù hợp với nhu cầu của môn học, nhóm quyết định sử dụng framework Spring Boot, mẫu thiết kế(design patern) là MVC.

### Lý do sử dụng framework và design pattern

* Spring boot là một trong những framework mạnh mẽ nhất và nhiều người sử dụng nhất của Java (62% lượng ứng dụng java được tạo ra sử dụng Spring Boot – theo <https://www.jrebel.com/blog/2021-java-technology-report>)



Hình II‑1Thị phần của các framework

* Mạng xã hội là một đề tài cần tính nâng cấp và mở rộng vậy nên nếu ko có mẫu thiết kế thì hiện thực hóa việc này rất khó khăn. MVC là một mẫu thiết kế phổ biến cũng như dễ làm quen hơn với người bắt đầu. Sau khi đã thực hiện MVVM và MVC nhóm quyết định chọn MVC.

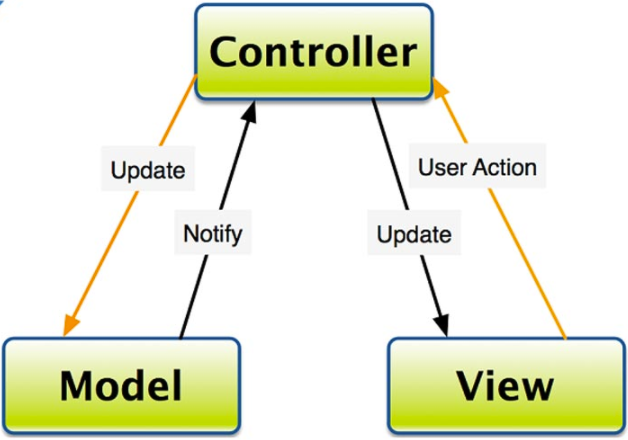
### Giới thiệu về công nghệ và mẫu thiết kế

#### Spring Boot

* Spring là một Java framework khổng lồ, hỗ trợ nhiều chức năng cho việc phát triển phần mềm. Nó được chia thành nhiều module, mỗi module làm một chức năng, ví dụ Spring Core, Web, Data access, AOP,... Spring được xây dựng dựa trên 2 khái niệm nền tảng là Dependency injection và AOP (Aspect Oriented Programming).
* Tuy nhiên Spring quá phức tạp khi phải cấu hình nhiều việc như thêm các thư viện cần thiết, tạo xml cấu hình project, code và build file war, cấu hình tomcat server,…
* Để giải quyết vấn đề này, Spring boot ra đời với các ưu điểm sau:
* Auto config: tự động cấu hình thay cho bạn, chỉ cần bắt đầu code và chạy là được
* Xây dựng các bean dựa trên annotation thay vì XML
* Server Tomcat được nhúng ngay trong file JAR build ra, chỉ cần chạy ở bất kì đâu java chạy được

#### Mẫu thiết kế MVC

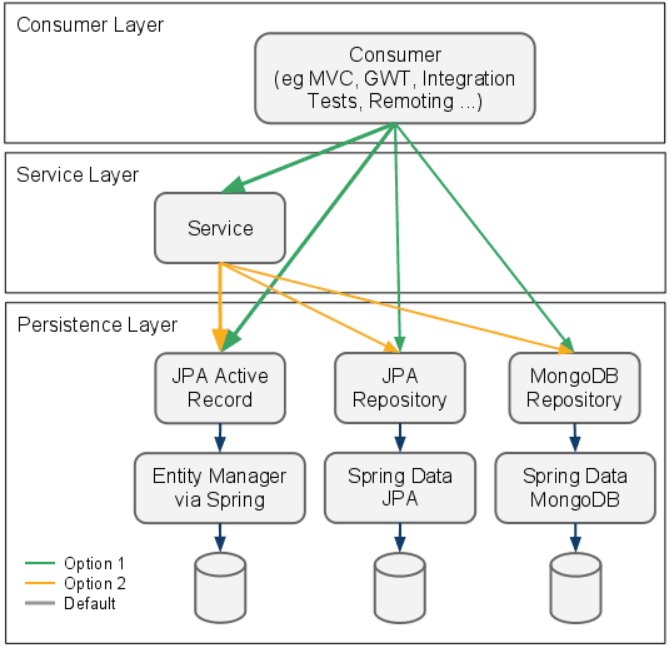
* Model(M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một [file XML](https://monamedia.co/file-xml-la-gi/) bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…
* View(V):
* Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website
* Thông thường, các ứng dụng web sử dụng MVC View như một phần của  hệ thống, nơi các thành phần [HTML](https://monamedia.co/html-va-css-la-gi/) được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của  người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi
* Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong Controller
* Controller (C):
* Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.
* Luồng tương tác giữa các thành phần trong MVC



* Ví dụ: Chức năng thêm bài viết mới trong trang quản trị website. Nơi đây có 2 trường nhập về tiêu đề và nội dung bài viết, thì trong đó:
* View sẽ hiển thị ra phần nhập form tiêu đề và nội dung.
* Controller lấy dữ liệu từ 2 trường và gửi tới Model.
* Model lúc này sẽ nhận dữ liệu từ Controller để lưu trữ trong cơ sở dữ liệu.

### Ứng dụng công nghệ vào đồ án

* Về framework:
* Nhóm ứng dụng Spring security cho việc đăng nhập
* JPA cho xử lý dữ liệu
* Về mẫu thiết kế:
* Thay vì sử dụng mô hình MVC thông thường nhóm tạo ra thêm 2 lớp là Repository và Service
* Khi đó Consumer hay Controller là tầng giao tiếp với bên ngoài; Service thực hiện các nghiệp vụ và logic; Repository chịu trách nhiệm giao tiếp với các DB, thiết bị lưu trữ, xử lý các query và trả về kiểu dữ liệu Service yêu cầu
* Hình minh họa:



Hình II‑2Minh họa Service và Repository

# Phân tích và thiết kế

Chương này sẽ trình bày các bước phân tích và thiết kế mạng xã hội để đáp ứng các mục tiêu được đề ra trong chương I. Phương pháp mô hình hóa UML và tích, thiết kế hướng đối tượng sẽ được sử dụng với các bước tiến hành như sau:

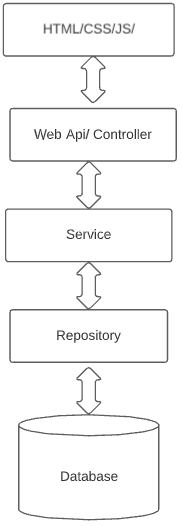
* Xây dựng kiến trúc dự án
* Xác định chức năng có thể hiện thực hóa, xây dựng use-case
* Mô tả cấu trúc bên trong của hệ thống thông qua việc thiết kế dữ liệu, kiến trúc và giao diện.

## Xây dựng kiến trúc dự án

Từ những ý tưởng ở Phần III mục C3, nhóm đã xây dựng kiến trúc phần mềm cho dự án như sau:

* Tầng giao diện sử dụng HTML/CSS/JS
* Tầng Controller
* Tầng Service: chứa các class để xử lý Business Logic
* Tầng Repository: Sử dụng Repository Pattern để ảo hóa code, tương tác với tầng database
* Tầng Database: chứa các class đại diện các bảng trong database

Trực quan hóa kiến trúc:



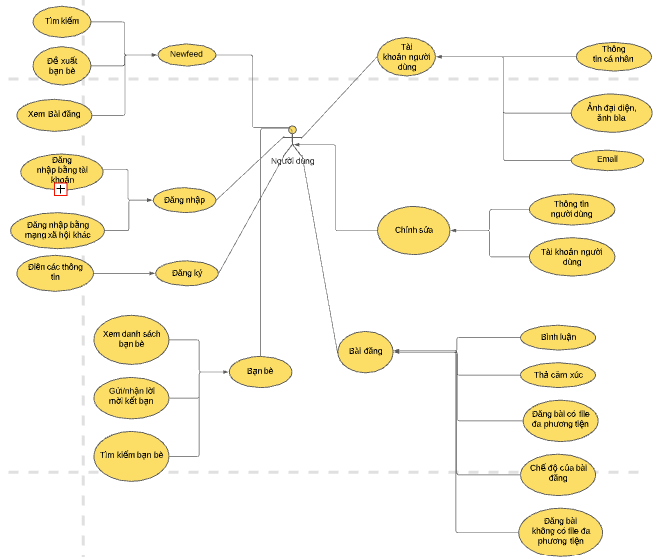
Hình III‑1Kiến trúc phần mềm

## Xác định chức năng có thể hiện thực hóa, xây dựng use case

Sau khi xem xét kỹ lưỡng và thực nghiệm thì nhóm đã xác định được các chức năng có thể hiện thực hóa như sau:

* Đăng ký: Người dùng chưa có tài khoản có thể đăng ký thành viên
* Đăng nhập:
* Đăng nhập bằng tài khoản của riêng mạng xã hội DM
* Đăng nhập bằng tài khoản của các nền tảng khác như facebook, github,..
* Newfeed:
* Tìm kiếm
* Đề xuất bạn bè
* Xem bài đăng
* Bạn bè:
* Xem danh sách bạn bè:
* Gửi lời mời kết bạn
* Nhận lời mời kết bạn
* Tìm kiếm bạn bè
* Bài đăng:
* Đăng bài không có file đa phương tiện
* Đăng bài có file đa phương tiện
* Bình luận, thả cảm xúc bài đăng
* Chế độ của bài đăng
* Tài khoản người dùng:
* Thông tin cá nhân
* Ảnh đại diện, ảnh bìa
* Email
* Chỉnh sửa:
* Chỉnh sửa tài khoản người dùng
* Chỉnh sửa thông tin người dùng

Trực quan hóa use-case của đồ án:



Hình III‑2 Use-case

## Danh sách người dùng và mô tả hệ thống

### Danh sách người dùng

Danh sách người dùng hiện tại chỉ chỉ có người sử dụng mạng xã hội

### Danh sách use-case và mô tả

#### Đăng ký:

* Người dùng chưa có tài khoản có thể đăng ký tài khoản
* Cách đăng ký:
* Người dùng nhập đầy đủ các thông tin cần thiết như: email, tên, ngày sinh,…
* Sau khi hệ thống đánh giá các thông tin đã nhập là hợp lệ, người dùng phải vào email của mình để xác thực lại tài khoản

#### Đăng nhập:

* Người dùng vào trang chủ có yêu cầu đăng nhập
* Người dùng có thể sử dụng tài khoản đã đăng ký hoặc tài khoản của mạng xã hội khác như facebook, google+ :
* Nếu sử dụng các tài khoản này, người dùng phải vào xác thực ngay khi đăng nhập để có thể vào trang chủ của người dùng ở lần đầu tiên.

#### Newfeed

* Người dùng sau khi đăng nhập sẽ vào được Newfeed của mình
* Ở Newfeed có một số chức năng con như:
* Tìm kiếm
* Đề xuất bạn bè
* Xem bài đăng

#### Bạn bè

* Người dùng có thể thực hiện các chức năng con như:
* Nhận lời mời kết bạn
* Gửi lời mời kết bạn
* Xem danh sách bạn bè
* TÌm kiếm bạn bè

#### Bài đăng

* Chức năng bài đăng có thể thực hiện khi người dùng đang ở NewFeed hoặc trang cá nhân của mình
* Các chức năng con của bài đăng:
* Tạo bài đăng (có thể chứa file đa phương tiện như ảnh, video)
* Thiết lập chế độ của bài đăng
* Bình luận
* Thả cảm xúc

#### Tài khoản người dùng

* Chức năng này giúp hiển thị thông tin:
* Thông tin cá nhân như Nơi ở, công việc,…
* Ảnh đại diện, ảnh bìa
* Địa chỉ email, các mạng xã hội khác như facebook, Instagram,…

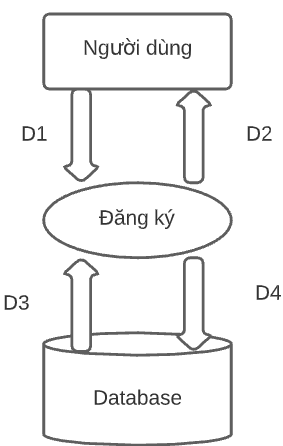
#### Chỉnh sửa:

* Người dùng sử dụng chức năng này để chỉnh sử các thông tin ở trang tài khoản người dùng

## Các sơ đồ và luồng xử lý hệ thống

### Đăng ký

#### Sơ đồ



Hình III‑3 Luồng xử lý đăng ký

D1: Hệ thống tiếp nhận việc đăng ký của người dùng

D2: Thông báo tạo tài khoản thành công

D3: Kiểm tra xem thông tin có hợp lệ không từ dữ liệu được đưa vào

D4: Nếu thông tin hợp lệ thì ghi vào cơ sở dữ liệu

#### Luồng xử kiện chính:

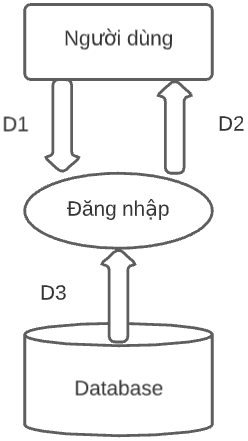
|  |  |
| --- | --- |
| Điều kiện trước: Không | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Chọn tab “Đăng ký” | Hiển thị màn hình đăng ký |
| Nhấn tạo tài khoản | Kiểm tra các thông tin có hợp lệ không ví dụ: Có để trống các trường bắt buộc không, email đã tồn tại chưa… Nếu tất cả hợp lệ thì gửi mã xác thực vào email, nếu không thì xuất ra lỗi. |
| Sau khi vào email để xác thực | Hệ thống thông báo tạo tài khoản thành công |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Đăng nhập

#### Sơ đồ



Hình III‑4 Luồng xử lý đăng nhập

D1: Người dùng tiến hành đăng nhập

D2: Đăng nhập thành công chuyển đến trang người dùng

D3: Kiểm tra xem tài khoản có ở trong cơ sở dữ liệu chưa

#### Luồng xử kiện chính:

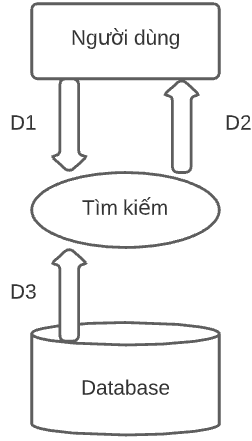
|  |  |
| --- | --- |
| Điều kiện trước: Có tài khoản của mạng xã hội DM hoặc các mạng xã hội khác DM cho phép đăng nhập | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Chọn tab “Đăng nhập” | Hiển thị màn hình đăng nhập |
| Điền thông tin đăng nhập | Kiểm tra tài khoản đã tồn tại chưa. Nếu có thì cho người dùng vào new feed nếu không xuất ra tài khoản không hợp lệ |
| Đăng nhập bằng tài khoản mạng xã hội khác | Nếu các mạng xã hội khác đã được đăng nhập trước, người dùng vào thẳng new feed; nếu chưa được đăng nhập, người dùng phải đăng nhập |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |
| Lần đầu đăng nhập bằng tài khoản mạng xã hội khác | Hệ thống sẽ gửi đến 1 URL yêu cầu xác thực. Sau khi xác thực thành công, hệ thống tự động tạo 1 tài khoản dựa vào mạng xã hội đã chọn |

### Tìm kiếm:

#### Sơ đồ



Hình III‑5Luồng xử lý tìm kiếm

D1: Người dùng gửi một yêu cầu tìm kiếm

D2: Hệ thống trả về kết quả tìm kiếm nếu tìm thấy

D3: Database trả lên dữ liệu nếu có

#### Luồng xử kiện chính:

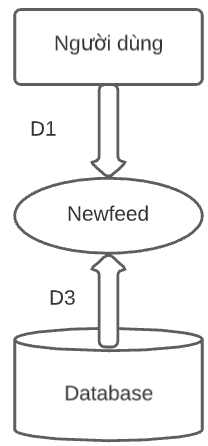
|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng sử dụng thanh tìm kiếm và nhập vào một thông tin bất kỳ | Tìm kiếm thông tin ở cơ sở dữ liệu. Nếu có thì trả lại cho người dùng các kết quả. Nếu không có, không trả về kết quả nào |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Newfeed

#### Sơ đồ



Hình III‑6Luồng xử lý Newfeed

D1: Người dùng truy cập được vào Newfeed

D3: Dữ liệu từ database trả lên

#### Luồng xử kiện chính:

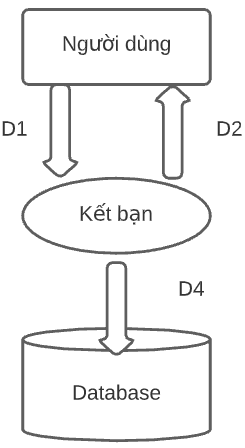
|  |  |
| --- | --- |
| Điều kiện trước: Người dùng đã đăng nhập thành công | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Sau khi đăng nhập | Hiển thị màn hình Newfeed. Ở Newfeed hiện ra các thông tin như: đề xuất bạn bè, bài đăng,.. |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Kết bạn:

#### Sơ đồ



Hình III‑7Luồng xử lý kết bạn

D1: Người dùng gửi yêu cầu kết bạn đến 1 người dùng khác:

D2: Hệ thống thông báo cho người dùng yêu cầu đã được gửi

D4: Yêu cầu được ghi xuống database

#### Luồng xử kiện chính:

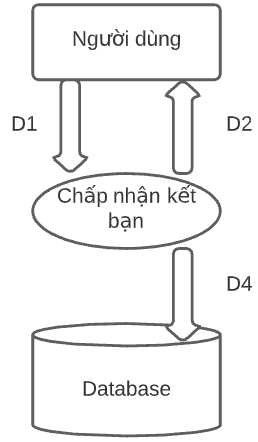
|  |  |
| --- | --- |
| Điều kiện trước: Không | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng thực hiện chức năng kết bạn | Hệ thống ghi nhận yêu cầu của người dùng, ghi yêu cầu kết bạn xuống database |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Chấp nhận kết bạn:

#### Sơ đồ:



Hình III‑8Chấp nhận kết bạn

D1: Người dùng kích hoạt chấp nhận kết bạn

D2: Hệ thống thông báo 2 người đã là bạn bè

D4: Quan hệ bạn bè được ghi xuống database đồng thời xóa yêu cầu kết bạn trong database

#### Luồng xử kiện chính:

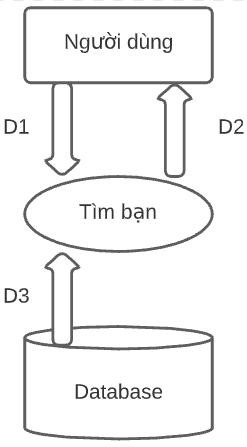
|  |  |
| --- | --- |
| Điều kiện trước: Người dùng nhận được yêu cầu kết bạn | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng chấp nhận yêu cầu kết bạn | Hệ thống ghi nhận, đưa quan hệ bạn bè giữa người gửi và người nhận vào cơ sở dữ liệu, đồng thời xóa yêu cầu kết bạn. |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Tìm bạn:

#### Sơ đồ



Hình III‑9 Luồng xử lý tìm bạn

D1: Người dùng gửi một yêu cầu tìm người khác qua tên

D2: Hệ thống trả về kết quả người dùng nếu tìm thấy

D3: Database trả lên tên của 1 hoặc nhiều người dùng nếu có

#### Luồng xử kiện chính:

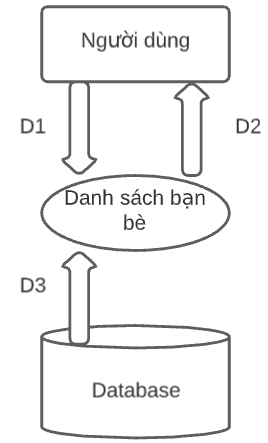
|  |  |
| --- | --- |
| Điều kiện trước: Người dùng nhận được yêu cầu kết bạn | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng sử dụng thanh tìm kiếm và nhập vào tên một người cần tìm. | Tìm kiếm thông tin ở cơ sở dữ liệu. Nếu có thì trả lại cho người dùng các kết quả. Nếu không có, không trả về kết quả nào |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Lấy danh sách bạn bè

#### Sơ đồ



Hình III‑10Luồng xử lý danh sách bạn bè

#### Luồng xử kiện chính:

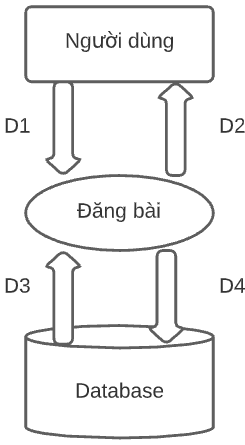
|  |  |
| --- | --- |
| Điều kiện trước: Không | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng nhấn vào button “Danh sách bạn bè”. | Tìm kiếm thông tin ở cơ sở dữ liệu. Trả lại danh sách bạn bè. |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Đăng bài:

#### Sơ đồ



Hình III‑11 Luồng xử lý đăng bài

D1: Người dùng gửi yêu cầu đăng bài

D2: Thông báo bài đăng thành công

D3: Hệ thống xem xét tính hợp lệ của bài đăng

D4: Bài đăng được ghi xuống cơ sở dữ liệu

#### Luồng xử kiện chính:

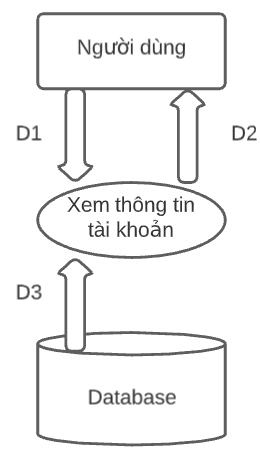
|  |  |
| --- | --- |
| Điều kiện trước: Người dùng phải ở trong Newfeed hoặc trang cá nhân | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng nhập một nội dung kèm file ảnh/video nếu có cùng với đó là quyền riêng tư như: công khai, bạn bè, chỉ mình tôi | Hệ thống kiểm tra tính hợp lệ rồi đưa dữ liệu đã đăng lên. |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Xem thông tin tài khoản

#### Sơ đồ



Hình III‑12Luồng xử lý xem thông tin tài khoản

#### Luồng xử kiện chính:

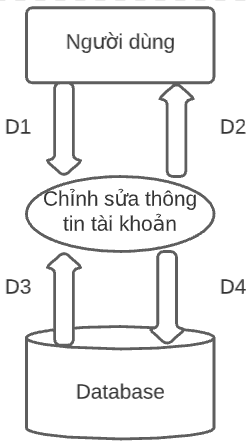
|  |  |
| --- | --- |
| Điều kiện trước: Không | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng nhấp vào tài khoản của mình hoặc một người khác | Nếu đó là tài khoản của người dùng, mọi thông tin sẽ được hiện ra  Nếu đó là tài khoản của người dùng khác, hệ thống sẽ chỉ hiện ra các thông tin dựa trên mối quan hệ giữa hai người |

#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

### Chỉnh sửa thông tin tài khoản:

#### Sơ đồ



Hình III‑13Luồng xử lý chỉnh sửa thông tin tài khoản

#### Luồng xử kiện chính:

|  |  |
| --- | --- |
| Điều kiện trước: Không | |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Người dùng chuyển đến tab “Chỉnh sửa thông tin”. | Người dùng tiến hành chỉnh sửa các thông tin. Hệ thống sẽ ghi nhận tính hợp lệ của thông tin chỉnh sửa. Nếu hợp lệ hệ thống sẽ báo lại chỉnh sửa thành công. Nếu không sẽ xuất ra lỗi tương ứng |

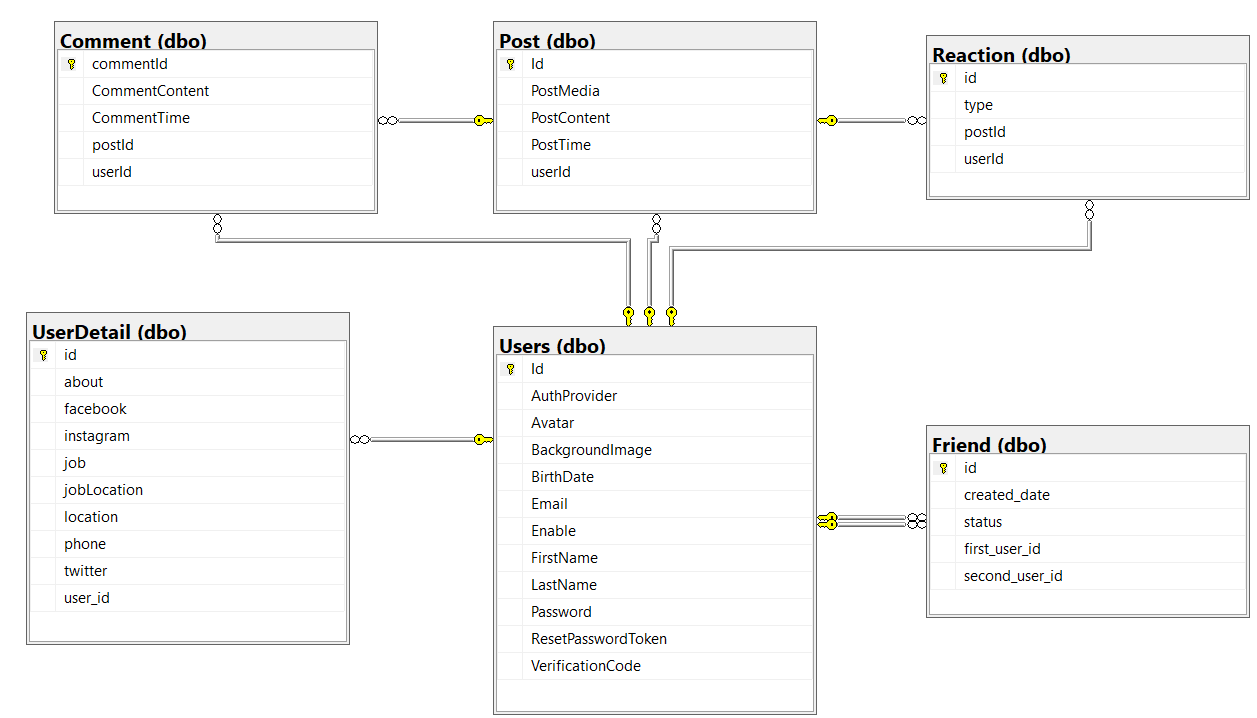
#### Luồng sự kiện ngẫu nhiên phát sinh:

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| Nhấn nhầm nút và thoát | Hệ thống thoát khỏi tab |
| Nhấn nhầm nút và sang trang khác | Hệ thống điều hướng đến trang kia |

## Thiết kế dữ liệu:

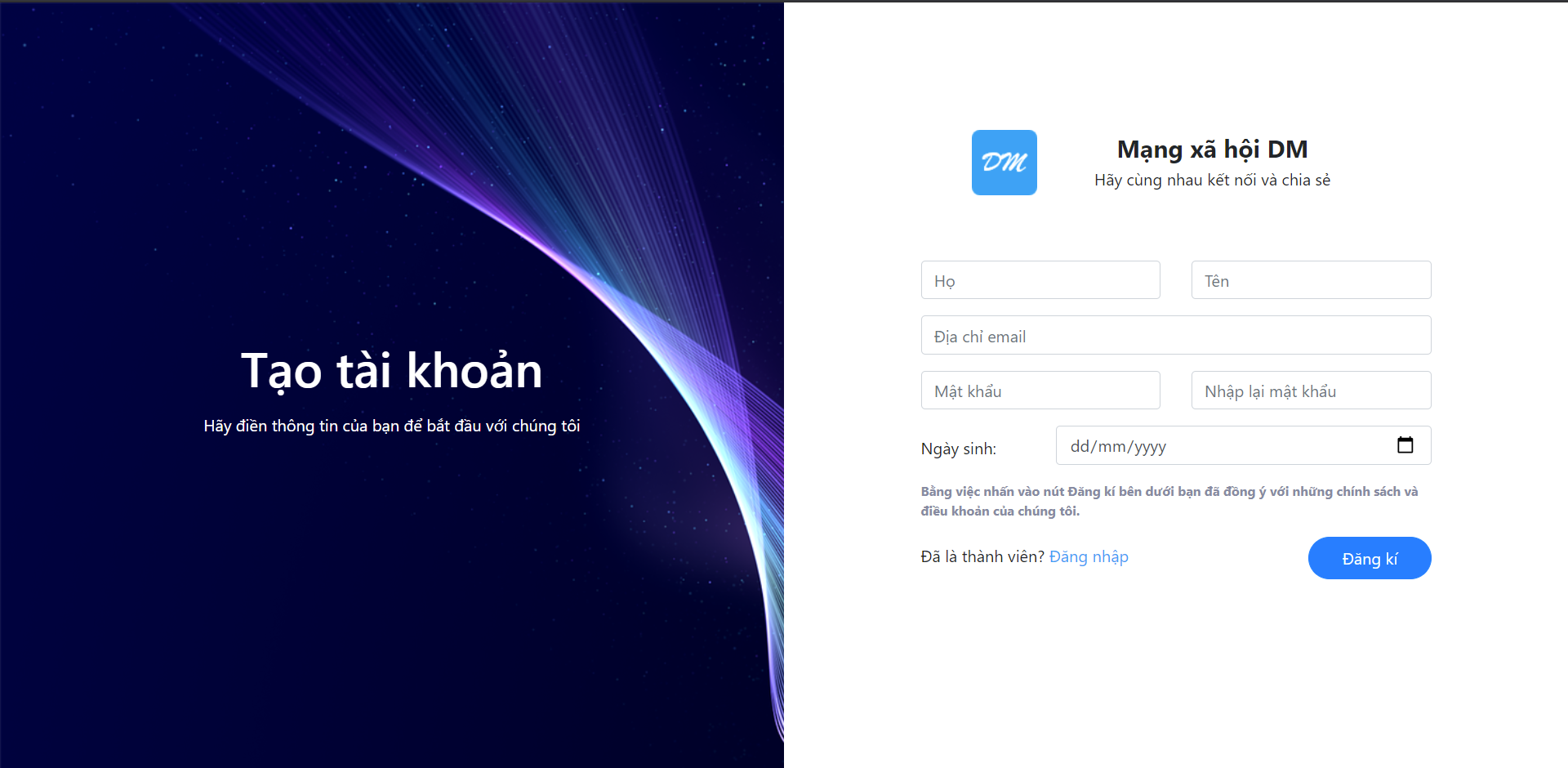
Từ Usecase, các sơ đồ và luồng xử lý đã lập được, nhóm thiết kế sơ đồ dữ liệu như sau:

* Bảng Users chứa các thông tin về tài khoản người dùng
* Bảng UsersDetail chứa các thông tin cá nhân ngoài đời của người dùng
* Bảng Post gồm các bài đăng
* Bảng Comment gồm các bình luận
* Bảng Reaction gồm các cảm xúc
* Bảng bạn bè chứa quan hệ bạn bè giữa các người dùng



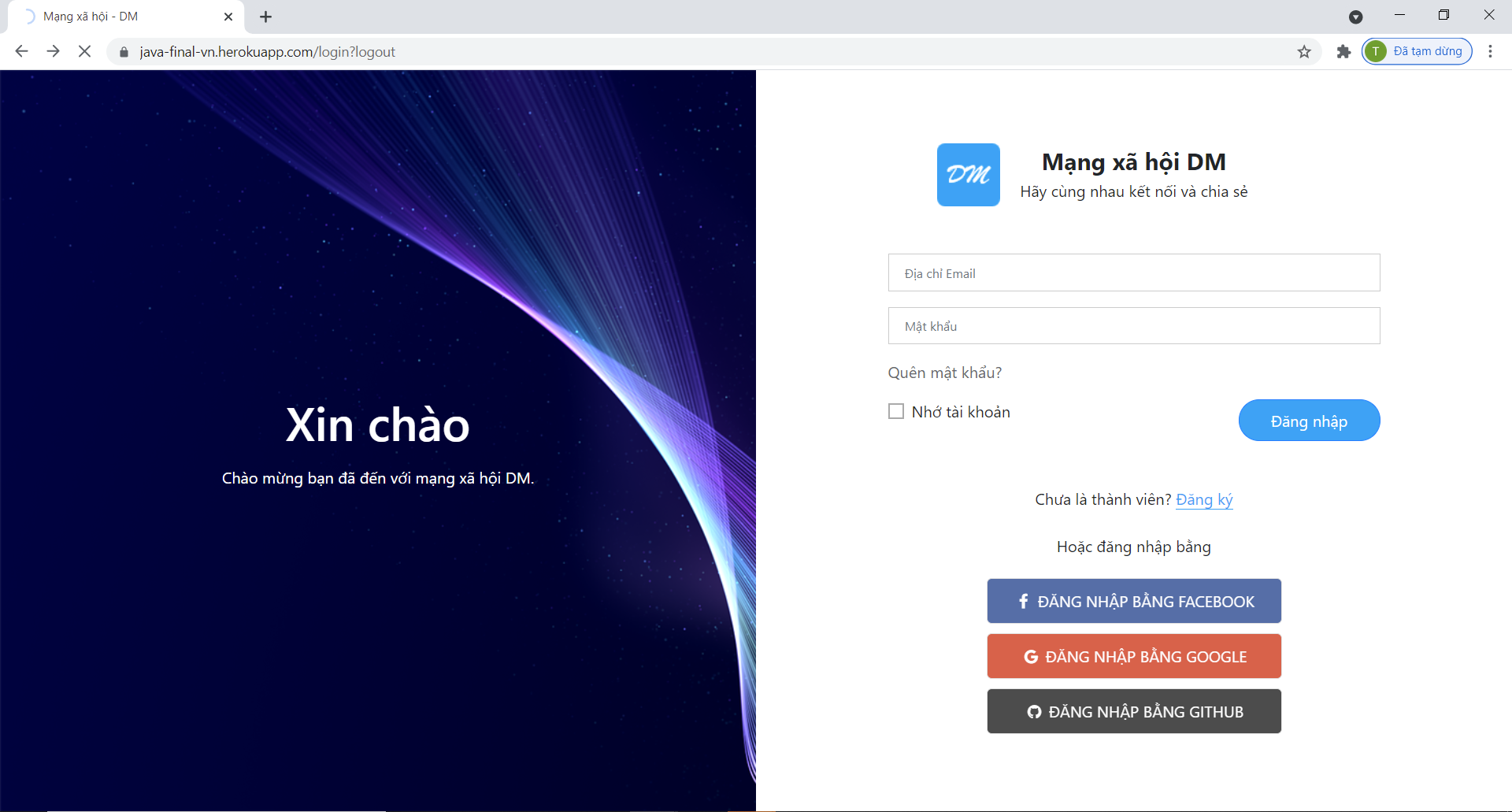
## THIẾT KẾ GIAO DIỆN:

### Giao diện trang đăng ký



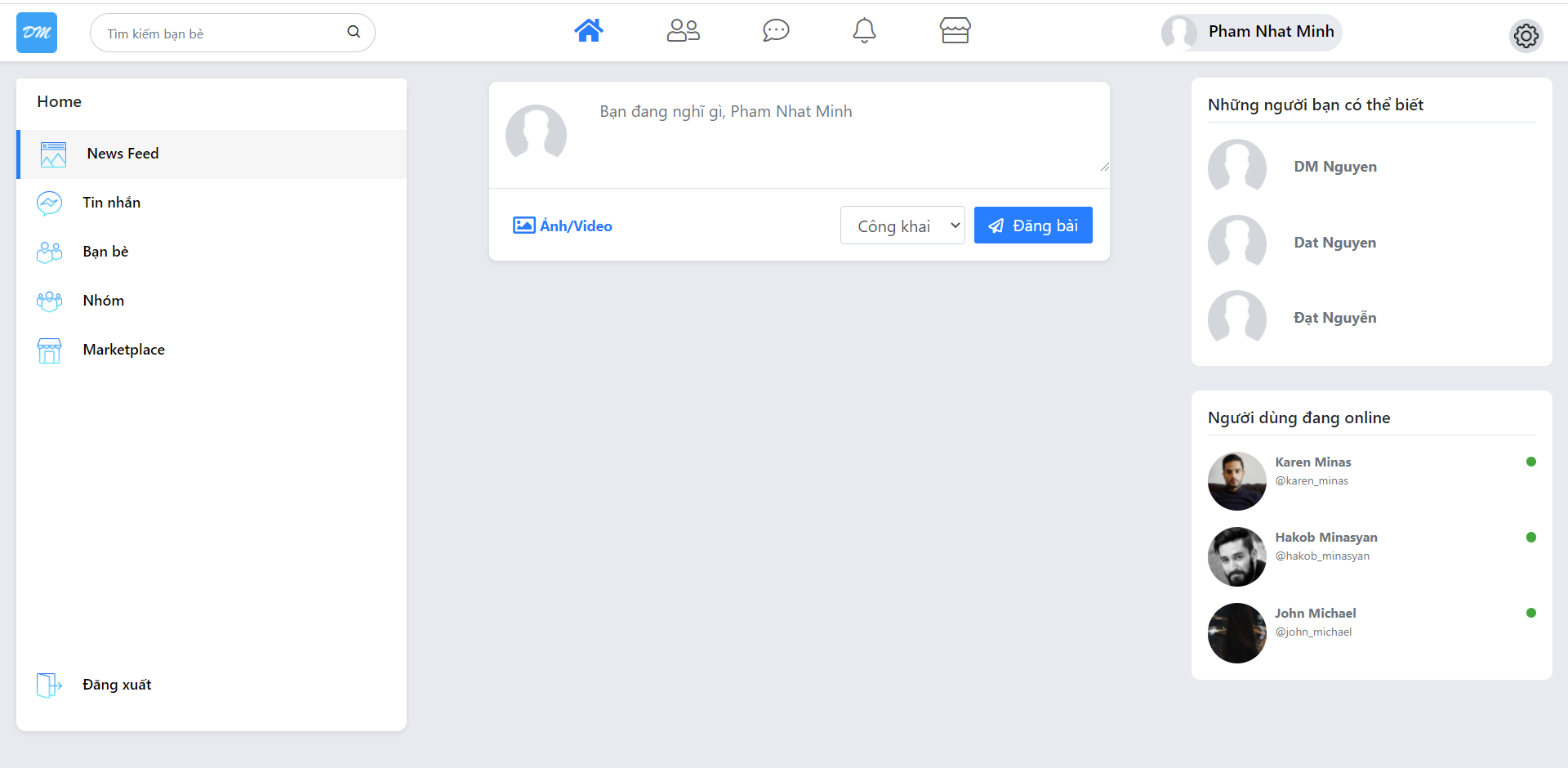
Hình III‑14Giao diện trang đăng ký

### Giao diện trang đăng nhập



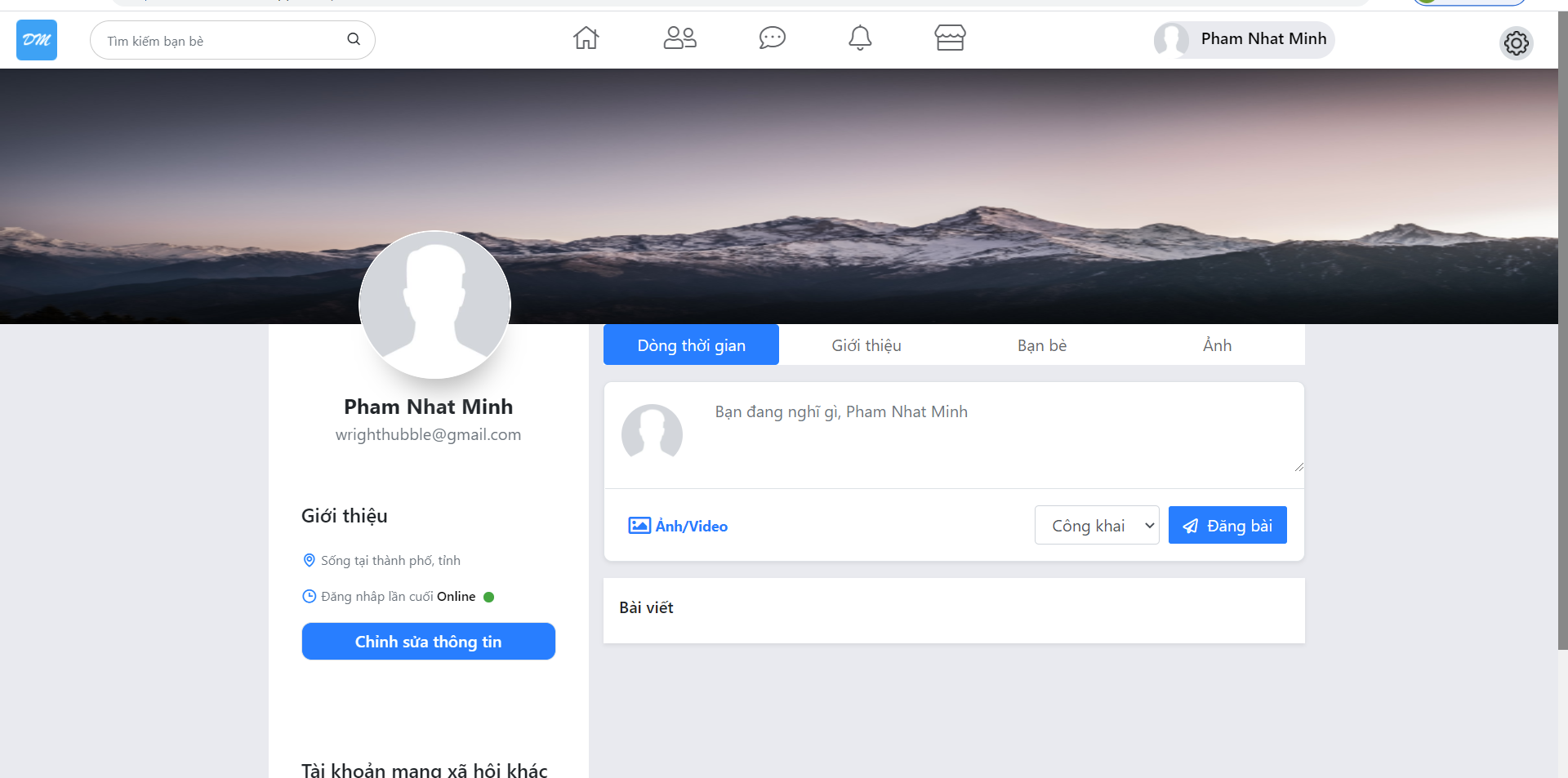
Hình III‑15Giao diện trang đăng nhập

### Giao diện Newfeed



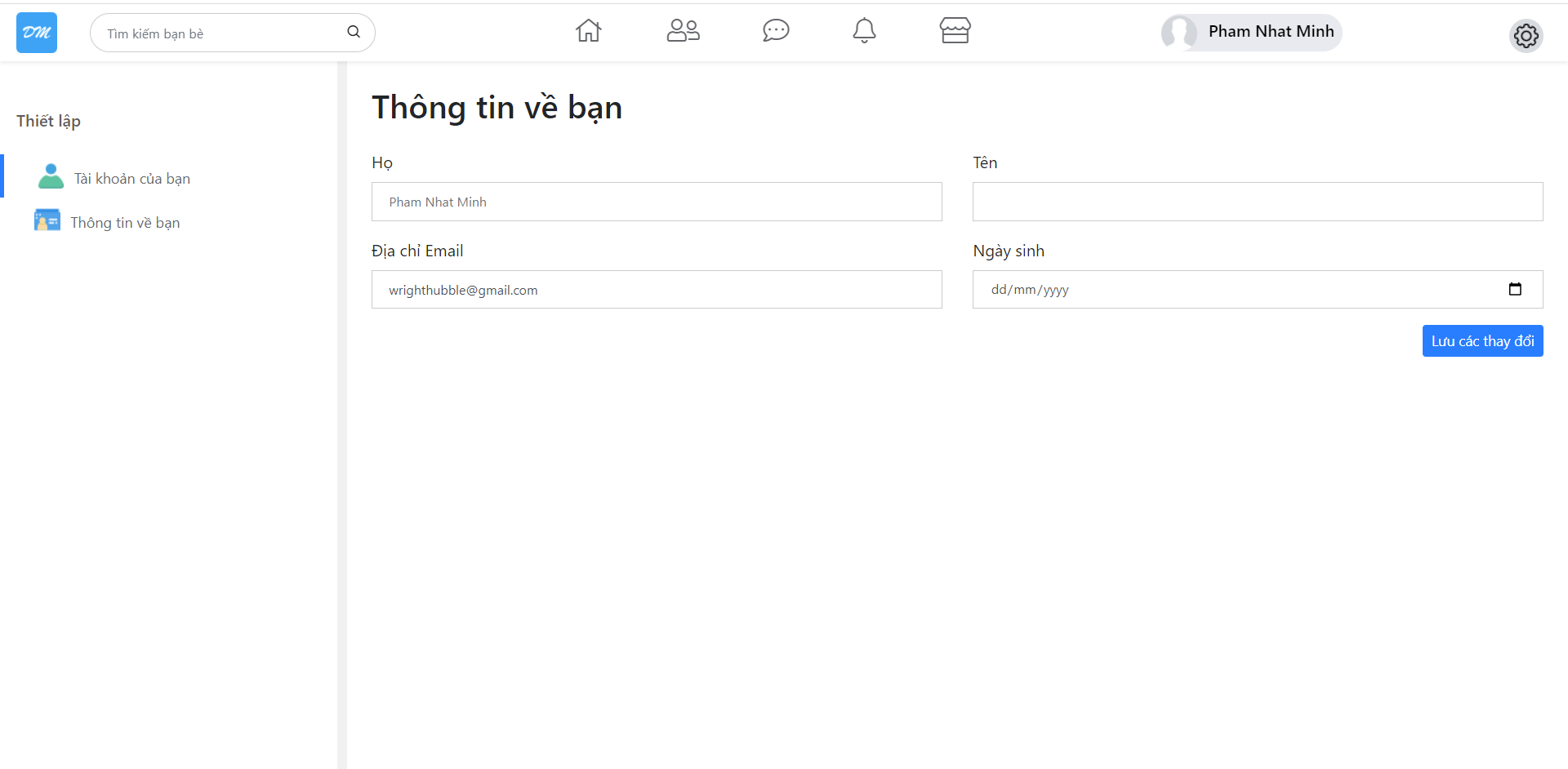
Hình III‑16Giao diện Newfeed

### Giao diện trang cá nhân



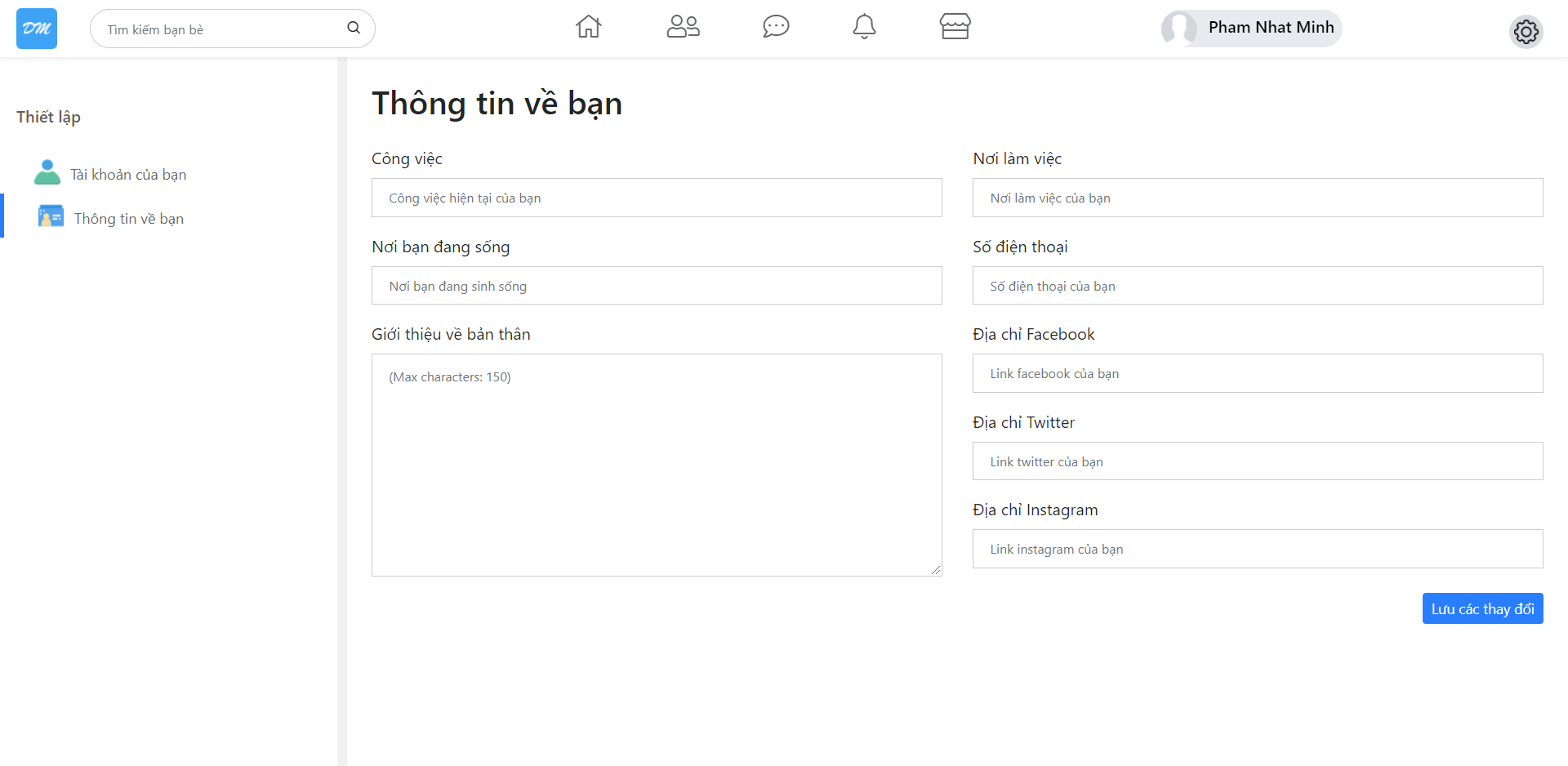
Hình III‑17Giao diện trang cá nhân

### Giao diện thay đổi thông tin tài khoản



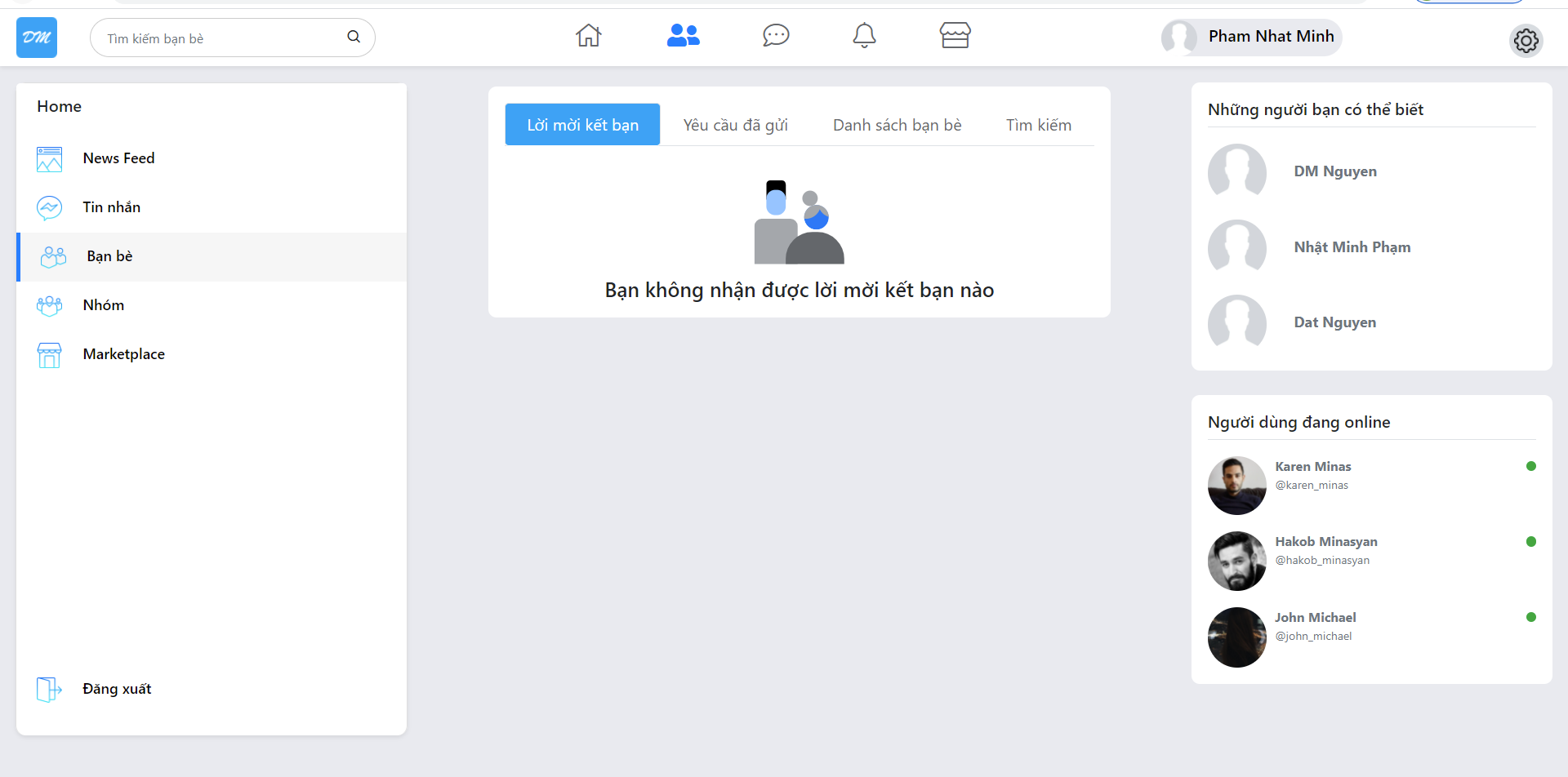
Hình III‑18Giao diện thay đổi thông tin tài khoản

### Giao diện thay đỏi thông tin cá nhân



Hình III‑19Giao diện thay đổi thông tin cá nhân

### Giao diện trang bạn bè



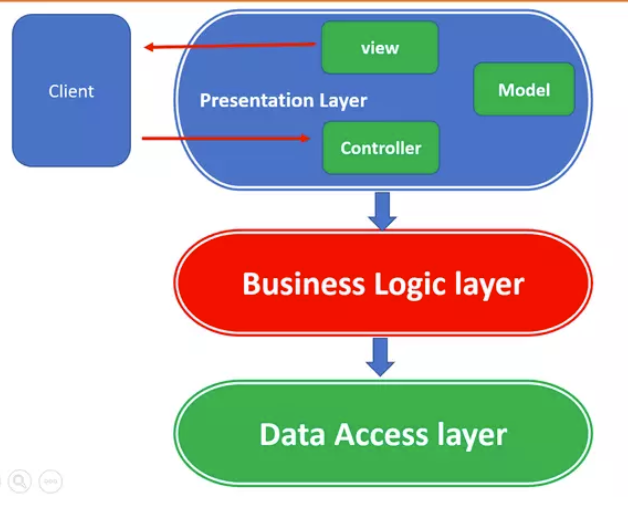
Hình III‑20Giao diện trang bạn bè

# HIỆN THỰC

## Chuẩn viết mã

* Tên class:
* Tên class được viết bằng tiếng anh và viết hoa các kí tự đầu của các thành phần có nghĩa
* VD: UserDetail, Friend,..
* Tên method:
* Tên method phải là động từ viết theo chuẩn camelCase (chữ cái đầu tiên bắt đầu từ từ thứ 2 trở đi được viết hoa)
* VD: Method thêm sản phẩm sẽ được viết với tên: addUsers
* Đối với các biến và hằng số:
* Tên biến phải đặt tên theo kiểu camelCase (chữ cái đầu tiên bắt đầu từ thứ 2 trở đi được viết hoa)
* Tên hằng số viết bằng chữ in hoa cho tất cả các kí tự

## Cách tổ chức mã nguồn



Hình IV‑1Cách tổ chức mã nguồn

### Mô hình MVC:

**Model:** các cấu trúc dữ liệu của toàn chương trình, có thể đại diện cho trạng thái của ứng dụng

**View**: lớp giao diện, dùng để hiển thị dữ liệu ra cho user xem và tương tác

**Controller:** kết nối giữa Model và View, điều khiển dòng dữ liệu

Dữ liệu từ Model qua Controller sau đó được gửi cho View hiển thị ra. Và ngược lại, khi có yêu cầu mới từ View, thì sẽ qua Controller thực hiện thay đổi dữ liệu của Model.

Tuy nhiên, MVC chỉ mô tả luồng đi của dữ liệu, nó không nói rõ như code đặt ở đâu (ở Model, View hay Controller), rồi lưu trữ Model vào database kiểu gì,... Do đó, đối với ứng dụng hoàn chỉnh như Spring Boot thì cần kết hợp cả mô hình MVC Service và Repository

### Áp dụng mô hình MVC cùng Service và Repository

Controller: trả về View (có chứa data sẵn, dạng trang HTML), hoặc Model thể hiện dưới dạng API cho View (View viết riêng bằng React, Vue, hoặc Angular).

Service: chứa các code tính toán, xử lý. Khi Controller yêu cầu, thì Service tương ứng sẽ tiếp nhận và cho ra dữ liệu trả cho Controller (trả về Model). Controller sẽ gửi về View như trên.

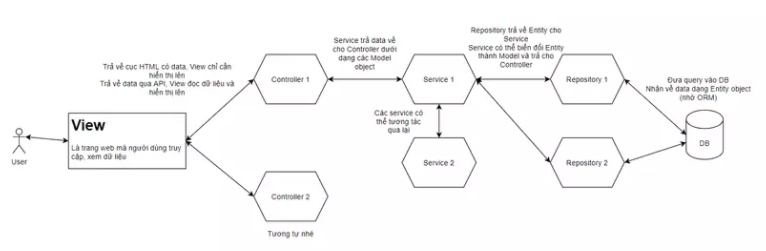
Repository: Service còn có thể tương tác với service khác, hoặc dùng Repository để gọi DB. Repository là thằng trực tiếp tương tác, đọc ghi dữ liệu trong DB và trả cho service.

Model chỉ đơn giản là các đối tượng được Service tính toán xong trả về cho Controller.

View thì có 2 loại, một là dạng truyền thống là trả về 1 cục HTML có data rồi. Lúc này Controller sẽ pass dữ liệu vào View và return về (Spring MVC có JSP hoặc template engine như Thymeleaf làm điều đó).

View dạng 2 là dạng View tách riêng (không thuộc về project Spring boot). Thường có trong các hệ thống dùng API. View sẽ được viết riêng bằng React, Angular,... Controller sẽ đưa dữ liệu Model thông qua API cho View, và cũng nhận lại các yêu cầu qua API luôn.

### Sơ đồ luồng đi:



Hình IV‑2 Sơ đồ luồng đi

#### Chiều từ user đến database:

Đầu tiên, user sẽ vào View để xem, tương tác

Khi user bắt đầu load dữ liệu (ví dụ click nút Reload), thì 1 request từ View gửi cho Controller (kiểu như "ê, cho tao cái danh sách user với")

Controller nhận được yêu cầu, bắt đầu đi hỏi ông Service (trong code là gọi method của Service)

Service nhận được yêu cầu từ Controller, đối với các code đơn giản có thể tính toán và trả về luôn. Nhưng các thao tác cần đụng tới database thì Service phải gọi Repository để lấy dữ liệu trong DB

Repository nhận được yêu cầu từ Service, sẽ thao tác với DB. Data lấy ra trong DB được hệ thống ORM (như JPA hoặc Hibernate) mapping thành các object (trong Java). Các object này gọi là Entity.

#### Chiều từ database trả về user:

Service nhận các Entity được Repository trả về, biến đổi nó. Biến đổi ở đây là có thể thực hiện tính toán, thêm bớt các field,... và cuối cùng biến Entity thành Model. Model sẽ được trả lại cho Controller.

Controller nhận được Model, nó sẽ return cho View. Có 2 cách, một là dùng template engine pass dữ liệu Model vào trang HTML, rồi trả về cục HTML (đã có data) cho client. Cách 2 là gửi qua API, View tự parse ra và hiển thị tương ứng (hiển thị thế nào tùy View).

Khi View hiển thị xong, user sẽ thấy danh sách user hiện lên trang web.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# BẢNG PHÂN CÔNG

|  |  |  |
| --- | --- | --- |
| Thành viên  Công việc | Phạm Nhật Minh | Nguyễn Thành Đạt |
| Làm báo cáo | x | x |
| Làm slide | x | x |
| Làm demo |  | x |

# TÀI LIỆU THAM KHẢO

[1] https://spring.io

[2] https://stackjava.com/category/spring

[3] <https://www.youtube.com/channel/UCYt1sfh5464XaDBH0oH_o7Q> - Java Brain channel on Youtube

[4] <https://www.youtube.com/channel/UC2KfmYEM4KCuA1ZurravgYw> - Amigocode channel on Youtube