

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH CÔNG NGHỆ THÔNG TIN

XÂY DỰNG TRANG WEB HẸN HÒ DATINGAPP  
SỬ DỤNG FRAMEWORK ANGULAR 16 VÀ .NET 8

CBHD: ThS. Nguyễn Thị Hương Lan

Sinh viên: Trần Hoàng Đức

Mã sinh viên: 2020606542

Hà Nội – Năm 2024

BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  
NGÀNH CÔNG NGHỆ THÔNG TIN

XÂY DỰNG TRANG WEB HẸN HÒ DATINGAPP  
XỬ DỤNG FRAMEWORK ANGULAR 16 VÀ .NET 8

CBHD: ThS. Nguyễn Thị Hương Lan  
Sinh viên: Trần Hoàng Đức  
Mã sinh viên: 2020606542

TRẦN HOÀNG ĐỨC

CÔNG NGHỆ THÔNG TIN

Hà Nội – Năm 2024

## LỜI CẢM ƠN

Trong thời đại số hóa ngày nay, việc sử dụng các ứng dụng và website hẹn hò đã trở thành một phần không thể thiếu trong cuộc sống của nhiều người. Với sự tiện lợi và linh hoạt, các nền tảng này không chỉ mở ra cơ hội gặp gỡ mới mà còn tạo ra những trải nghiệm giao tiếp độc đáo. Việc **xây dựng trang web hẹn hò DatingApp sử dụng framework Angular 16 và .Net 8** là kết quả của công việc nghiên cứu và phát triển một website hẹn hò, với mục tiêu cung cấp một nền tảng an toàn và thú vị cho việc kết nối và gặp gỡ giữa các cá nhân.

Để hoàn thành được đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội đã tận tình giảng dạy và trang bị kiến thức cho em trong suốt thời gian em học tập tại trường. Giáo viên hướng dẫn đề tài – **Thạc Sĩ Nguyễn Thị Hương Lan**, giảng viên Khoa Công nghệ Thông tin Trường Đại học Công Nghiệp Hà Nội – đã tận tụy hết lòng giúp đỡ, hướng dẫn, chỉ dẫn tận tình để giúp em hoàn thành được đồ án. Bạn bè đã góp ý và giúp đỡ, bên cạnh động viên và khích lệ em trong suốt thời gian em làm đồ án.

*Em xin chân thành cảm ơn!*

## MỤC LỤC

|  |     |
|--|-----|
| <b>DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT</b> | i   |
| <b>DANH MỤC BẢNG</b>                                       | ii  |
| <b>DANH MỤC HÌNH ẢNH</b>                                   | iii |
| <b>CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG</b>        | 1   |
| <b>1.1. Cơ sở lý thuyết</b>                                | 1   |
| <b>1.1.1. ASP.NET Core</b>                                 | 1   |
| <b>1.1.2. Angular là gì</b>                                | 4   |
| <b>1.1.3. API</b>  | 7   |
| <b>1.1.4. API Testing với Postman</b>                      | 9   |
| <b>1.1.5 Hệ quản trị cơ sở dữ liệu SQL Server</b>          | 10  |
| <b>1.2. Công cụ hỗ trợ lập trình</b>                       | 11  |
| <b>1.2.1. Microsoft Visual Studio</b>                      | 11  |
| <b>1.2.2. Visual studio code</b>                           | 12  |
| <b>CHƯƠNG 2. KHẢO SÁT YÊU CẦU</b>                          | 14  |
| <b>2.1. Lý do chọn đề tài</b>                              | 14  |
| <b>2.2. Mục tiêu của trang web</b>                         | 15  |
| <b>2.3. Khảo sát một số website thực tế</b>                | 16  |
| <b>2.4. Các yêu cầu chức năng</b>                          | 18  |
| <b>2.5. Các yêu cầu phi chức năng</b>                      | 20  |
| <b>CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG</b>               | 21  |
| <b>3.1. Biểu đồ use case</b>                               | 21  |
| <b>3.2. Đặc tả chi tiết usecase</b>                        | 26  |
| <b>3.2.1. Use case đăng nhập</b>                           | 26  |
| <b>3.2.2. Use case đăng ký</b>                             | 27  |
| <b>3.2.3. Use case tìm kiếm người dùng</b>                 | 28  |
| <b>3.2.4. Use case xem chi tiết người dùng</b>             | 29  |
| <b>3.2.5. Use case quản lý thông tin cá nhân</b>           | 30  |

|  |    |
|--|----|
| <b>3.2.6. Use case nhắn tin .....</b>                  | 31 |
| <b>3.2.7. Use case quản lý bài đăng .....</b>          | 32 |
| <b>3.2.8. Use case tương tác bài đăng .....</b>        | 33 |
| <b>3.2.9. Use case quản lý người dùng.....</b>         | 36 |
| <b>3.2.10. Use case quản lý báo cáo bài đăng .....</b> | 37 |
| <b>3.2.11. Use case quản lý thông báo .....</b>        | 38 |
| <b>3.2.12. Use case hẹn hò.....</b>                    | 39 |
| <b>3.3. Phân tích use case.....</b>                    | 41 |
| <b>3.3.1. Use case đăng ký .....</b>                   | 41 |
| <b>3.3.2. Use case đăng nhập .....</b>                 | 43 |
| <b>3.3.3. Use case tìm kiếm người dùng .....</b>       | 45 |
| <b>3.3.4. Use case xem chi tiết người dùng.....</b>    | 47 |
| <b>3.3.5. Use case quản lý thông tin cá nhân .....</b> | 49 |
| <b>3.3.6. Use case nhắn tin.....</b>                   | 51 |
| <b>3.3.7. Use case quản lý thông báo .....</b>         | 53 |
| <b>3.3.8. Use case quản lý bài đăng cá nhân .....</b>  | 55 |
| <b>3.3.9. Use case tương tác bài đăng .....</b>        | 57 |
| <b>3.3.10. Use case quản lý người dùng.....</b>        | 60 |
| <b>3.3.11. Use case quản lý báo cáo bài đăng .....</b> | 62 |
| <b>3.3.12. Use case hẹn hò.....</b>                    | 66 |
| <b>3.4. Mô tả giao diện .....</b>                      | 68 |
| <b>3.5. Thiết kế cơ sở dữ liệu .....</b>               | 71 |
| <b>3.5.1. Sơ đồ thực thể liên kết .....</b>            | 71 |
| <b>3.5.2. Thiết kế bảng.....</b>                       | 74 |
| <b>CHƯƠNG 4: KẾT QUẢ VÀ KIỂM THỬ .....</b>             | 80 |
| <b>4.1. Giao diện .....</b>                            | 80 |
| <b>4.2. Kiểm thử .....</b>                             | 83 |
| <b>4.2.1. Kế hoạch kiểm thử .....</b>                  | 83 |
| <b>4.2.2. Thiết kế ca kiểm thử .....</b>               | 84 |

|                                       |            |
|---------------------------------------|------------|
| <b>4.2.3. Thực thi kiểm thử .....</b> | <b>90</b>  |
| <b>4.2.3. Kết quả kiểm thử .....</b>  | <b>97</b>  |
| <b>KẾT LUẬN .....</b>                 | <b>98</b>  |
| <b>TÀI LIỆU THAM KHẢO .....</b>       | <b>100</b> |

## DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

| Ký hiệu | Giải thích   |
|---------|--|
| API     | Giao diện lập trình ứng dụng (Application programming interface) |
| HTML    | Ngôn ngữ đánh dấu (Hyper Text Markup Language)                   |
| CSS     | Ngôn ngữ tạo phong cách cho trang web (Cascading Style Sheets)   |
| UC      | Use Case   |
| HTTP    | Giao thức truyền tải (Hypertext Transfer Protocol)               |
| CSDL    | Cơ sở dữ liệu  |
| SQL     | Ngôn ngữ truy vấn có cấu trúc                                    |
| Admin   | Quản trị hệ thống  |

**DANH MỤC BẢNG**

|                                      |    |
|--------------------------------------|----|
| Bảng 3.5.1. Bảng thông tin User..... | 74 |
| Bảng 3.5.2. Bảng Role .....          | 74 |
| Bảng 3.5.3. Bảng Role với User ..... | 75 |
| Bảng 3.5.4. Bảng UserPhoto .....     | 75 |
| Bảng 3.5.5. Bảng UserProfile .....   | 75 |
| Bảng 3.5.6. Bảng UserInterest .....  | 76 |
| Bảng 3.5.7. Bảng Message .....       | 76 |
| Bảng 3.5.8. Bảng Post .....          | 76 |
| Bảng 3.5.9. Bảng PostPhoto.....      | 77 |
| Bảng 3.5.10. Bảng Comment .....      | 77 |
| Bảng 3.5.11. Bảng Reaction.....      | 77 |
| Bảng 3.5.12. Bảng PostReport .....   | 78 |
| Bảng 3.5.13. Bảng Connection .....   | 78 |
| Bảng 3.5.14. Bảng Notification ..... | 78 |
| Bảng 3.5.15. Bảng DatingRequest..... | 79 |

## **DANH MỤC HÌNH ẢNH**

|  |    |
|--|----|
| Hình 1.1. ASP.NET Core.....  | 1  |
| Hình 1.2. So sánh build time .Net 7 vs .Net 8.....                 | 2  |
| Hình 1.3. Cơ chế hoạt động của Middleware .....                    | 4  |
| Hình 1.4. Angular Signals.....                                     | 7  |
| Hình 1.5. API .....  | 8  |
| Hình 1.6. Cách hoạt động của API.....                              | 8  |
| Hình 1.7. Postman .....  | 9  |
| Hình 1.8. Giao diện Postman .....                                  | 10 |
| Hình 1.9. SQL Server.....  | 10 |
| Hình 1.12. Visual studio.....                                      | 12 |
| Hình 2.1. Website VietnamCupid.....                                | 16 |
| Hình 2.2. Website dating.com.....                                  | 17 |
| Hình 3.1.1. Biểu đồ usecase tổng quát.....                         | 21 |
| Hình 3.1.2. Sơ đồ use case tương tác bài đăng .....                | 22 |
| Hình 3.1.3. Sơ đồ use case quản lý bài đăng cá nhân .....          | 22 |
| Hình 3.1.4. Sơ đồ use case quản lý thông báo .....                 | 23 |
| Hình 3.1.5. Sơ đồ use case quản lý tin nhắn .....                  | 23 |
| Hình 3.1.6. Sơ đồ use case quản lý hẹn hò .....                    | 24 |
| Hình 3.1.7. Sơ đồ use case quản lý thông tin cá nhân .....         | 24 |
| Hình 3.1.8. Biểu đồ use case admin .....                           | 25 |
| Hình 3.3.1. Biểu đồ trình tự use case đăng ký .....                | 41 |
| Hình 3.3.2. Biểu đồ VOPC use case đăng ký .....                    | 42 |
| Hình 3.3.3. Biểu đồ trình tự use case đăng nhập .....              | 43 |
| Hình 3.3.4. Biểu đồ VOPC đăng nhập.....                            | 44 |
| Hình 3.3.5. Biểu đồ trình tự use case tìm kiếm người dùng .....    | 45 |
| Hình 3.3.6. Biểu đồ VOPC tìm kiếm người dùng.....                  | 46 |
| Hình 3.3.7. Biểu đồ trình tự use case xem chi tiết người dùng..... | 47 |
| Hình 3.3.8. Biểu đồ VOPC use case xem chi tiết người dùng.....     | 48 |

|  |    |
|--|----|
| Hình 3.3.9. Biểu đồ trình tự usecase sửa thông tin cá nhân.....      | 49 |
| Hình 3.3.10. Biểu đồ VOPC use case quản lý thông tin cá nhân .....   | 50 |
| Hình 3.3.11. Biểu đồ trình tự use case nhắn tin .....                | 51 |
| Hình 3.3.12. Biểu đồ VOPC use case nhắn tin .....                    | 52 |
| Hình 3.3.13. Biểu đồ trình tự use case quản lý thông báo .....       | 53 |
| Hình 3.3.14. Biểu đồ VOPC quản lý thông báo.....                     | 54 |
| Hình 3.3.15. Biểu đồ trình tự use case quản lý bài đăng cá nhân..... | 55 |
| Hình 3.3.16. Biểu đồ VOPC use case quản lý bài đăng cá nhân .....    | 56 |
| Hình 3.3.17. Biểu đồ tương tác bài đăng .....                        | 58 |
| Hình 3.3.18. Biểu đồ VOPC use case tương tác bài đăng .....          | 59 |
| Hình 3.3.19. Biểu đồ trình tự use case quản lý người dùng.....       | 60 |
| Hình 3.3.20. Biểu đồ VOPC use case quản lý người dùng .....          | 61 |
| Hình 3.3.21. Biểu đồ trình tự use case quản lý báo cáo bài đăng..... | 62 |
| Hình 3.3.22. Biểu đồ VOPC use case quản lý báo cáo bài đăng .....    | 63 |
| Hình 3.3.23. Biểu đồ trình tự use case quản lý thông báo .....       | 64 |
| Hình 3.3.24. Biểu đồ VOPC use case quản lý thông báo .....           | 65 |
| Hình 3.3.25. Biểu đồ trình tự use case hẹn hò .....                  | 66 |
| Hình 3.3.26. Biểu đồ VOPC use case hẹn hò .....                      | 67 |
| Hình 3.4.1. Mô tả giao diện tin nhắn .....                           | 68 |
| Hình 3.4.2. Mô tả giao diện màn hình chính.....                      | 68 |
| Hình 3.4.3. Mô tả giao diện màn hình bài đăng.....                   | 69 |
| Hình 3.4.4. Mô tả giao diện màn hình trang cá nhân.....              | 69 |
| Hình 3.4.5. Mô tả giao diện màn hình trang hẹn hò .....              | 70 |
| Hình 3.5.1. Mô hình thực thể liên kết .....                          | 73 |
| Hình 4.1. Giao diện màn hình Matches .....                           | 80 |
| Hình 4.2. Giao diện màn hình Posts.....                              | 80 |
| Hình 4.3. Giao diện màn hình tin nhắn .....                          | 81 |
| Hình 4.4. Giao diện màn hình tương tác bài đăng .....                | 81 |
| Hình 4.5. Giao diện màn hình thông báo .....                         | 81 |

|  |    |
|--|----|
| Hình 4.6. Giao diện màn hình trang cá nhân ..... | 82 |
| Hình 4.7. Giao diện màn hình Admin .....         | 82 |
| Hình 4.8. Giao diện màn hình đăng nhập .....     | 82 |
| Hình 4.9. Giao diện màn hình đăng ký .....       | 83 |
| Hình 4.10. Giao diện màn hình hẹn hò .....       | 83 |

## CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ SỬ DỤNG

### 1.1. Cơ sở lý thuyết

#### 1.1.1. ASP.NET Core

ASP.NET Core là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (cloud) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm dẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

Đồng thời nó đã trở thành một mã nguồn mở. Đây là một thay đổi rất lớn và theo mình là quan trọng nhất của ASP.NET Core. Điều mà trước đây khó có một lập trình viên nào có thể nghĩ đến. Có lẽ đó cũng là một xu thế mà các ngôn ngữ lập trình hiện nay đang hướng tới.



Hình 1.1. ASP.NET Core

- *Ưu điểm vượt trội của ASP.NET Core 8*

Microsoft .Net 8 đã được release vào tháng 11 năm 2023 với rất nhiều cải tiến và features mới cho lập trình viên. Điểm nhấn mạnh nhất trong đó là sự

ra đời của ASP.NET Core 8, một bản nâng cấp quan trọng của nền tảng xây dựng websites mã nguồn mở của Microsoft.

ASP.NET Core 8 được xây dựng trên nền tảng của .Net core runtime, qua đó cho phép các nhà phát triển xây dựng ứng dụng Web có thể chạy trên nhiều nền tảng khác nhau từ Windows đến Linux hay MacOS.

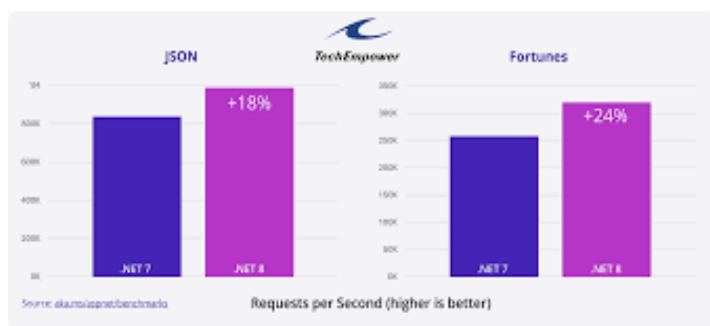
Performance là một phần rất quan trọng khi phát triển ứng dụng hiện đại ngày nay. Về mặt kỹ thuật, tăng performance đồng nghĩa với việc tối ưu hóa tài nguyên server tốt hơn, giảm được chi phí hosting. Mặt khác, performance tốt hơn cũng đồng nghĩa với việc SEO cho website cũng tốt hơn. Có thể nói, ai cũng muốn tốc độ website của mình nhanh nhất có thể.

Với phiên bản ASP.NET core 8, về hiệu năng chúng ta có:

- Throughput của middleware request tăng ~5%.
- MVC trên Linux tăng ~12%
- Kết nối HTTPS sử dụng ít bộ nhớ hơn ~40%
- Protobuf serialization nhanh hơn ~20%

#### - *Tối ưu hóa việc compile Razor View*

.Net 8 tối ưu hóa hiệu năng cho toàn bộ .Net platform, bao gồm việc tăng tốc độ build và tăng hiệu năng khởi động của ứng dụng. Với Razor view của ứng dụng Web, việc dùng engine mới Roslyn Source Generators làm cho việc build MVC Razor Views nhanh gấp 2 lần so với engine build cũ trên .Net 5. Điều này góp phần làm cho thời gian phát triển nhanh hơn, đỡ thời gian chờ đợi build.



Hình 1.2. So sánh build time .Net 7 vs .Net 8

- *Hot reload*

Hot reload là một trong những tính năng đáng mong chờ nhất được thêm vào .NET 8. Với tính năng này, bạn có thể sửa views, pages, components, thay đổi logic hiển thị... khi ứng dụng ASP.NET core 6 đang thực được chạy, lưu thay đổi, và thấy các thay đổi được phản ánh luôn trên ứng dụng mà không cần phải build lại và chạy lại ứng dụng từ đầu.

- *Tách biệt riêng CSS file cho các Page và View*

Với .Net 8, giờ đây bạn có thể sử dụng các file style riêng biệt cho từng page và view, mà không sợ làm ảnh hưởng đến style của toàn hệ thống. Việc này cho phép lập trình viên dễ dàng hơn trong việc tìm lỗi, và tránh các lỗi không đáng có khi có quá nhiều file CSS được thêm/sửa/xóa ở nhiều nguồn khác nhau.

Bằng cách khai báo tên file CSS, ví dụ ProductDetail.cshtml.css trùng khớp với tên View (ProductDetail.cshtml), toàn bộ style của file CSS đó sẽ chỉ được áp dụng cho View cụ thể (ProductDetail.cshtml) mà thôi.

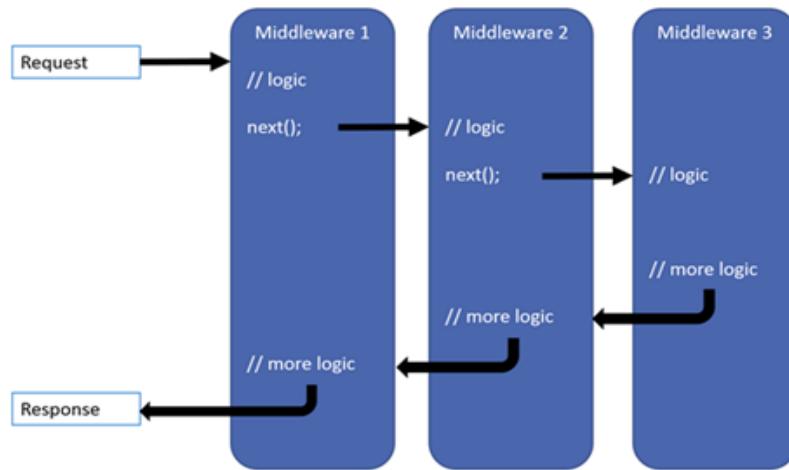
- *Async Streaming*

Trên ASP.NET core 8, Async Streaming đã được hỗ trợ, giúp cho việc truy vấn data với Entity Framework core được tối ưu hơn. Giờ đây response content của một Action sẽ không cần phải buffer trong bộ nhớ (memory) trước khi trả về nữa, dẫn đến việc tối ưu memory hơn khi Action cần trả về một lượng lớn bản ghi.

- *Middleware – bao gồm HTTP Logging middleware*

Chúng ta chắc không còn xa lạ gì với khái niệm request pipeline, một cơ chế xử lý một request đầu vào và trả ra một response ở đầu ra. Với mỗi request được gửi đến web server, nó sẽ được xử lý qua pipeline và trả về client khi đã được xử lý xong. Middleware là các module phần mềm được gắn vào request pipeline để xử lý các requests đầu vào responses đầu ra. Mỗi Middleware sẽ quyết định xử lý một request hoặc response nào đó trước khi chuyển tiếp cho Middleware tiếp theo hoặc bỏ qua hoàn toàn, dựa vào các điều kiện lọc của

middleware đó. Hình dưới đây từ website của Microsoft sẽ giúp bạn dễ hình dung hơn về middleware và cơ chế hoạt động của nó.



Hình 1.3. Cơ chế hoạt động của Middleware

Với sự xuất hiện của HTTP Logging middleware trên ASP.NET core 6, bạn có thể log thông tin về các Http request và response đến server của ứng dụng như:

- Thông tin request
- Thông tin về response
- Thông tin về header, body của các request và response

Ngoài ra bạn cũng có thể viết các custom middleware để phục vụ cho dự án của mình.

### 1.1.2. Angular là gì

Angular là một framework phát triển ứng dụng web và di động do Google phát triển. Được xây dựng dựa trên TypeScript, Angular là một công cụ mạnh mẽ cho việc xây dựng các ứng dụng web động và đa trang, từ các ứng dụng nhỏ đến các ứng dụng doanh nghiệp lớn.

Dưới đây là một số điểm quan trọng về Angular:

- ✓ Framework hoàn chỉnh: Angular không chỉ là một thư viện, mà là một framework hoàn chỉnh cung cấp mọi thứ bạn cần để xây dựng ứng dụng web phức tạp. Nó bao gồm các công cụ cho

routing, quản lý trạng thái ứng dụng, forms, và nhiều tính năng khác.

- ✓ TypeScript: Angular được viết bằng TypeScript, một siêu set của JavaScript với kiểu dữ liệu tinh và nhiều tính năng hiện đại. TypeScript giúp kiểm soát lỗi, cung cấp tính năng lập trình hướng đối tượng mạnh mẽ, và tăng cường tính hiệu suất của ứng dụng.
- ✓ Component-based architecture: Angular sử dụng kiến trúc dựa trên các thành phần (components). Mỗi thành phần đảm nhận một phần nhỏ của giao diện người dùng và quản lý trạng thái của nó. Điều này giúp tổ chức mã nguồn một cách dễ dàng và tái sử dụng linh hoạt.
- ✓ Two-way data binding: Angular cung cấp tính năng two-way data binding, giúp đồng bộ hóa tự động dữ liệu giữa mã nguồn và giao diện người dùng. Điều này giảm sự phức tạp trong quá trình quản lý trạng thái ứng dụng.
- ✓ Dependency Injection (DI): Angular sử dụng hệ thống Dependency Injection để quản lý và cung cấp các thành phần của ứng dụng. Điều này giúp tạo ra mã nguồn dễ kiểm thử, dễ bảo trì và linh hoạt.
- ✓ Cộng đồng và tài liệu mạnh mẽ: Angular có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu, hướng dẫn, và nguồn thông tin để hỗ trợ người phát triển.

Với những đặc điểm này, Angular trở thành một lựa chọn mạnh mẽ cho việc phát triển các ứng dụng web chất lượng cao.

#### - *Phiên bản Angular 16*

Angular do Google phát triển, là một framework mã nguồn mở được sử dụng rộng rãi trong phát triển ứng dụng web. Nó cung cấp một bộ công cụ mạnh

mẽ cùng hàng loạt tính năng, cho bạn tạo app web năng động, đáp ứng và có thể mở rộng.

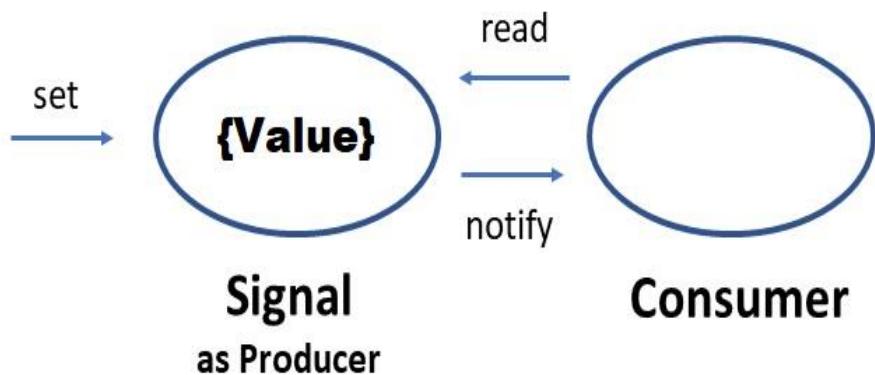
Angular phiên bản 16 giới thiệu những nội dung mới và cải tiến thú vị cho trải nghiệm lập trình cũng như đảm bảo độ ổn định và hiệu suất hoạt động của ứng dụng tốt hơn.

#### - *Angular Signals*

Angular Signals là một thư viện, cho phép định nghĩa các giá trị phản ứng và thiết lập phần phụ thuộc giữa chúng.

Trong Angular 16, một tính năng chính khác được giới thiệu là Tín hiệu. Với những tính năng mới nhất này, chúng ta có thể quản lý cơ chế thay đổi trạng thái trong ứng dụng Angular. Nó được lấy cảm hứng từ Solid.js. Tín hiệu được biểu diễn dưới dạng hàm luôn trả về một giá trị với sự trợ giúp của phương thức get() và đồng thời, hàm tương tự có thể được cập nhật bằng giá trị mới với sự trợ giúp của set(). Ngoài ra, chúng ta có thể định nghĩa tín hiệu là một cấu trúc phụ thuộc. Vì vậy, bất cứ khi nào giá trị thay đổi theo các phần tử phụ thuộc, tín hiệu sẽ tự động cập nhật giá trị của nó. Signal cũng hỗ trợ đối tượng có thể quan sát RxJS có thể được sử dụng trong các ứng dụng dựa trên Angular 16 để phát triển bất kỳ luồng chuỗi dữ liệu khai báo và mạnh mẽ nào. Trong Angular 16, Tín hiệu hoặc Tín hiệu góc được phát hành dưới dạng bản xem trước của nhà phát triển.

Hàm tín hiệu là một đối tượng chấp nhận một giá trị. Người tiêu dùng sử dụng tín hiệu có thể đọc giá trị và cập nhật giá trị nếu được yêu cầu. Ngoài ra, nếu giá trị bị thay đổi do theo bất kỳ cách nào khác thì nó cũng thông báo cho người tiêu dùng tương tự. Nếu bất kỳ thành phần góc nào bị ràng buộc với tín hiệu thì Angular sẽ tự động kích hoạt phát hiện thay đổi bất cứ khi nào tín hiệu thay đổi.



Hình 1.4. Angular Signals

Trong Angular 16, chúng ta có thể sử dụng chức năng tín hiệu bằng cách sử dụng gói `@angular/core`. Hàm này chủ yếu trả về cái gọi là `WritableSignal<T>` sự vật. Ở đây `T` đại diện cho loại giá trị được quản lý.

- *Loại bỏ trình biên dịch tương thích Ngcc hoặc Angular*

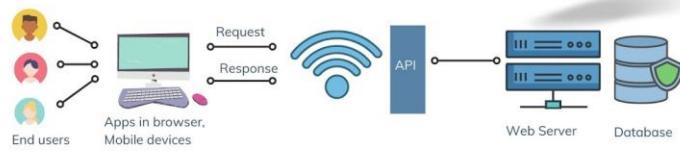
Trước đây, với việc phát hành Angular 9, nhóm Angular đã triển khai công cụ xem mới có tên Ivy liên quan đến quá trình biên dịch liên quan đến quy trình kết xuất thế hệ tiếp theo. Vào thời điểm đó, ngcc đã được giới thiệu để hỗ trợ các thư viện dựa trên công cụ xem cũ.

Trong phiên bản Angular 16 mới nhất, ngcc và tất cả các khôi mã liên quan đến công cụ xem cũ đã bị xóa để giảm kích thước gói ứng dụng. Vì vậy, engine cũ không thể sử dụng được từ Angular 16 trở đi. Vì vậy, bây giờ, chúng ta có thể nói rằng khung Angular đã được chuyển hoàn toàn sang Ivy.

### 1.1.3. API

API được ví như “cánh tay phải” đắc lực không thể thiếu đối với lập trình website và thiết kế các phần mềm ứng dụng di động. Nó có tác dụng liên kết các tính năng của web, app với các cơ sở dữ liệu; tăng tương tác tối đa giữa người dùng và ứng dụng. Bên cạnh đó, công cụ này còn có rất nhiều điều hấp dẫn khác.

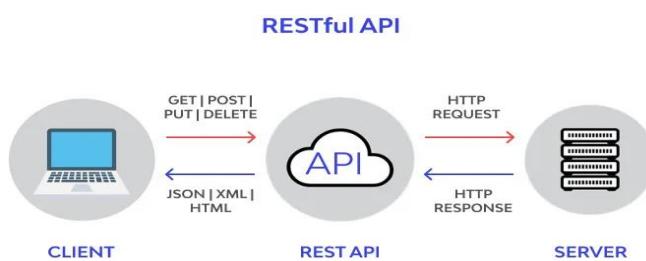
# What is an API?



A Video Tutorial by Ayushi Rawat

Hình 1.5. API

- **Ưu điểm của API**
  - + API được tích hợp trên hầu hết các ứng dụng của desktop, mobile và các ứng dụng trên website.
  - + Có sự linh hoạt trong việc định dạng dữ liệu khác nhau khi được trả về client.
  - + API là một công cụ có mã nguồn mở, việc này giúp bạn có thể sử dụng API mọi lúc bất kể đâu với điều kiện là phải có kết nối với internet. Giúp hỗ trợ chức năng RESTful.
  - + Hỗ trợ giao tiếp hai chiều và được xác nhận trong những giao dịch khác nhau. Điều này giúp đảm bảo được độ tin cậy cao.
  - + Về cấu hình của API khi so sánh với cấu hình của WCF thì vô cùng thân thiện và đơn giản mang lại các trải nghiệm tuyệt vời cho người dùng.



Hình 1.6. Cách hoạt động của API

- + API hỗ trợ đầy đủ các thành phần MVC từ Model Binder, Action cho đến Unit Test,...
- + API giúp người dùng nhanh chóng xây dựng HTTP service điều này giúp công việc lập trình đơn giản hơn.

#### **1.1.4. API Testing với Postman**

- Postman là gì

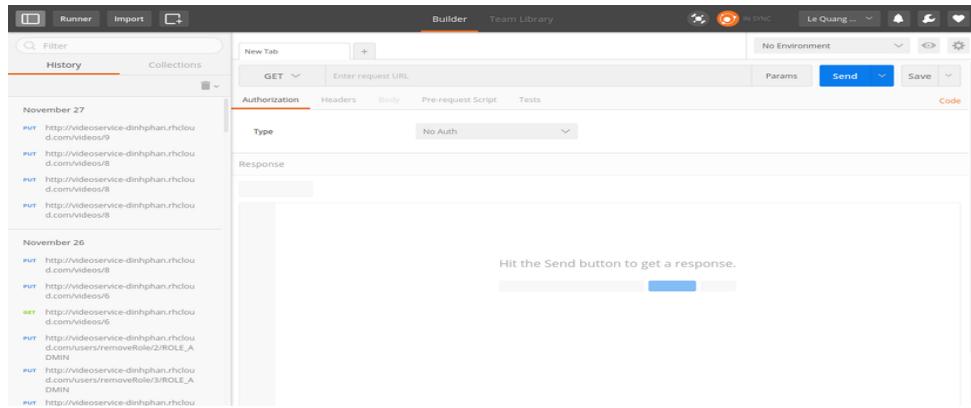
Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.

Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, ...). Bên cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.



Hình 1.7. Postman

- Tại sao sử dụng Postman
  - + Sử dụng Collections (Bộ sưu tập) – Postman cho phép người dùng tạo bộ sưu tập cho các lệnh gọi API của họ. Mỗi bộ sưu tập có thể tạo các thư mục con và nhiều yêu cầu (request). Điều này giúp việc tổ chức các bộ thử nghiệm.
  - + Collaboration – Collections và environment có thể được import hoặc export giúp chia sẻ tệp dễ dàng.
  - + API Testing – Test trạng thái phản hồi HTTP.
  - + Gõ lỗi – Bảng điều khiển Postman giúp kiểm tra dữ liệu nào đã được truy xuất giúp dễ dàng gõ lỗi kiểm tra.



Hình 1.8. Giao diện Postman

### 1.1.5 Hệ quản trị cơ sở dữ liệu SQL Server

- *SQL Server là gì?*

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, database engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.



Hình 1.9. SQL Server

SQL, viết tắt của Structured Query Language, được hiểu đơn giản là ngôn ngữ truy vấn cấu trúc dữ liệu. Đây là một loại ngôn ngữ lập trình được sử dụng để quản lý và truy vấn cơ sở dữ liệu quan hệ. SQL cho phép bạn thực hiện các hoạt động như lưu trữ, cập nhật, xóa, tìm kiếm, truy xuất dữ liệu, quản lý và tối ưu hóa hiệu suất của cơ sở dữ liệu. Không ít người nhầm lẫn giữa hai khái niệm SQL Server và ngôn ngữ SQL.

- *Vì sao sử dụng SQL Server?*

Việc sử dụng SQL Server trong dự án trang web hẹn hò có thể được giải thích với các lý do sau:

- + **Độ Ổn Định và Đáng Tin Cậy:** SQL Server là một hệ quản trị cơ sở dữ liệu phổ biến và được ổn định, được phát triển và hỗ trợ bởi Microsoft. Sự ổn định và đáng tin cậy của SQL Server giúp đảm bảo rằng dữ liệu của người dùng được lưu trữ và quản lý một cách an toàn và hiệu quả.
- + **Hiệu Suất Tốt:** SQL Server cung cấp các công cụ và tính năng tối ưu hóa hiệu suất cho cơ sở dữ liệu. Điều này bao gồm các công cụ tối ưu hóa truy vấn, chỉ mục, và quản lý tài nguyên để đảm bảo rằng trang web hoạt động một cách mượt mà và nhanh chóng, kể cả khi có số lượng lớn người dùng truy cập.
- + **Khả Năng Mở Rộng:** SQL Server có khả năng mở rộng tốt, cho phép mở rộng cơ sở dữ liệu khi cần thiết để đáp ứng nhu cầu tăng trưởng của ứng dụng. Điều này cung cấp một nền tảng linh hoạt cho việc mở rộng và phát triển trang web hẹn hò theo thời gian.
- + **Tích Hợp Với Công Nghệ Microsoft:** SQL Server tích hợp tốt với các công nghệ khác của Microsoft như .NET Framework, ASP.NET và C#. Điều này giúp đơn giản hóa việc phát triển và triển khai ứng dụng, đồng thời tạo ra một môi trường phát triển có tính nhất quán.
- + **Bảo Mật Cao:** SQL Server cung cấp các tính năng bảo mật mạnh mẽ, bao gồm kiểm soát truy cập và mã hóa dữ liệu, giúp bảo vệ thông tin cá nhân và quan trọng của người dùng trên trang web hẹn hò.

## 1.2. Công cụ hỗ trợ lập trình

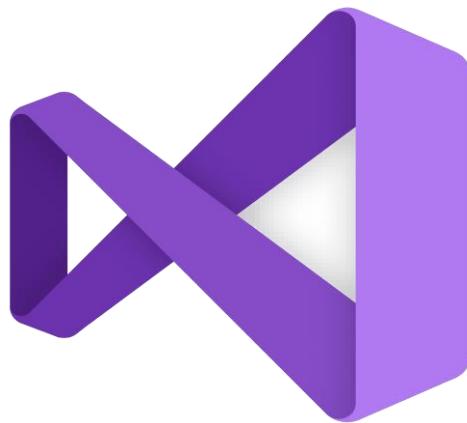
### 1.2.1. Microsoft Visual Studio

Là một môi trường phát triển tích hợp (IDE) từ Microsoft. Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã nhiều người sử dụng nhất thế giới ", được dùng để lập trình C++ và C# là chính. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows

Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio hỗ trợ lập trình trên nhiều nền tảng ngôn ngữ khác nhau từ C/C++, C#, cho đến F#, Visual Basic, HTML, CSS, JavaScript. Thậm chí, phiên bản VS 2015 có hỗ trợ code trên ngôn ngữ Python.

Microsoft cung cấp phiên bản "Express" (đối với phiên bản Visual Studio 2013 trở về trước) và "Community" (đối với bản Visual Studio 2015 trở về sau) là phiên bản miễn phí của Visual Studio.



Hình 1.12. Visual studio

### 1.2.2. Visual studio code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở được phát triển bởi Microsoft. Nó hỗ trợ lập trình cho nhiều ngôn ngữ khác nhau như JavaScript, Python, C++, và nhiều ngôn ngữ khác thông qua việc sử dụng các tiện ích mở rộng. VS Code cung cấp nhiều tính năng hữu ích cho lập trình viên, bao gồm:

- Tính năng IntelliSense: Hỗ trợ tự động hoàn thiện mã, thông tin về các biến, kiểu dữ liệu, hàm...
- Quản lý mã nguồn: Tích hợp hỗ trợ Git cho phép bạn quản lý mã nguồn một cách hiệu quả ngay trong trình soạn thảo.
- Debugging: Cung cấp khả năng gỡ lỗi trực tiếp trong trình soạn thảo.

- Tiện ích mở rộng (Extensions): Cho phép cài đặt thêm các tiện ích để mở rộng chức năng của trình soạn thảo, hỗ trợ nhiều ngôn ngữ lập trình và công cụ khác nhau.
- Terminal tích hợp: Cho phép bạn truy cập và sử dụng terminal/command line ngay trong môi trường soạn thảo.
- Tùy chỉnh giao diện và cấu hình: Cung cấp khả năng tùy chỉnh giao diện và cấu hình môi trường làm việc theo sở thích cá nhân.
- Hỗ trợ đa nền tảng: VS Code có thể chạy trên các hệ điều hành Windows, macOS và Linux.



*Hình 1.13. Visual studio code*

## CHƯƠNG 2. KHẢO SÁT YÊU CẦU

### 2.1. Lý do chọn đề tài

Việc lựa chọn đề tài liên quan đến website hẹn hò là một quyết định được đưa ra sau một quá trình cân nhắc kỹ lưỡng và thảo luận sâu sắc. Đây là một lựa chọn tinh táo và có tính chiến lược, được định hình bởi một số yếu tố quan trọng.

Trước hết, việc nghiên cứu và phát triển một website hẹn hò không chỉ là về việc tạo ra một ứng dụng hoặc trang web công nghệ, mà còn là về việc thấu hiểu sâu sắc về tâm lý và nhu cầu của người sử dụng. Điều này mở ra một cánh cửa rộng lớn về việc khám phá các mô hình hành vi, yếu tố tâm lý, và nền văn hóa ẩn sau hành động online của con người.

Thứ hai, lựa chọn này phản ánh xu hướng xã hội hiện đại, nơi mà sự kết nối qua internet đang trở thành một phần không thể tách rời của cuộc sống hàng ngày. Đặc biệt, trong bối cảnh của cuộc cách mạng kỹ thuật số và tình hình dịch bệnh toàn cầu, nhu cầu về việc tìm kiếm mối quan hệ qua các nền tảng trực tuyến đã trở nên ngày càng quan trọng và phổ biến.

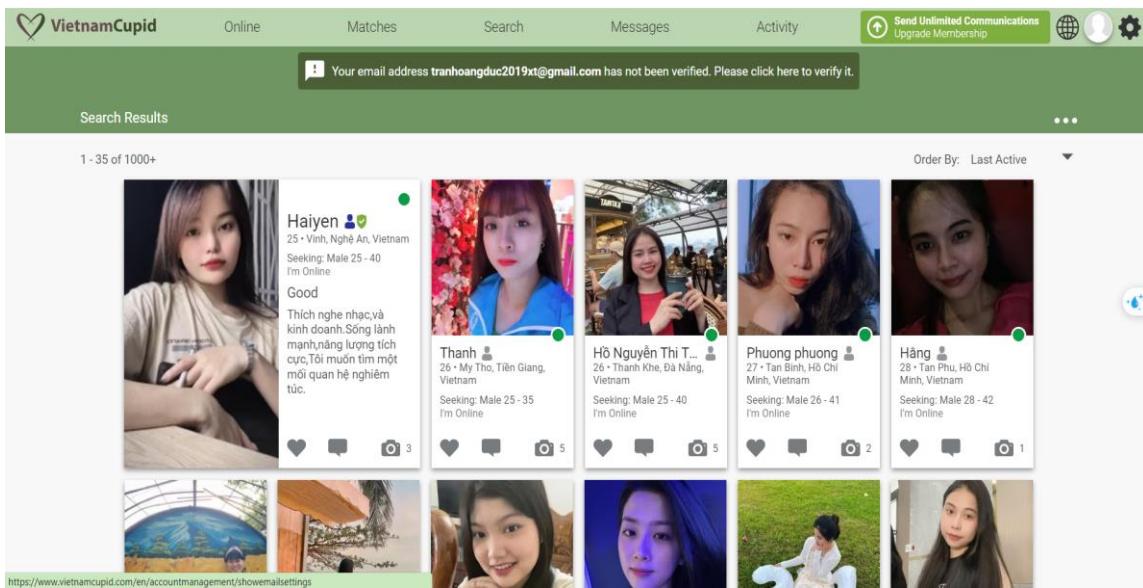
Thứ ba, việc phát triển một website hẹn hò không chỉ là về việc tạo ra một cơ hội gặp gỡ mới mà còn là về việc xây dựng một môi trường an toàn và đáng tin cậy cho người sử dụng. Qua việc tập trung vào các yếu tố như bảo mật thông tin cá nhân, kiểm soát chất lượng và tính đa dạng của người dùng, em mong muốn góp phần tạo ra một không gian trực tuyến lành mạnh và tích cực.

Do đó, việc xây dựng website hẹn hò trực tuyến là một vấn đề thực tế, ứng dụng được và có tiềm năng phát triển trong tương lai. Nhận thấy được nhu cầu “Tìm kiếm tình yêu”, được sự hướng dẫn tận tình của giáo viên **ThS. Nguyễn Hương Lan**, em đã chọn đề tài: “Xây dựng website hẹn hò DatingApp sử dụng framework Angular 16 và .Net 8”.

## 2.2. Mục tiêu của trang web

- **Tạo Ra Một Nền Tảng Giao Tiếp An Toàn:** Một trong những mục tiêu hàng đầu của dự án là xây dựng một nền tảng giao tiếp an toàn và đáng tin cậy. Em cam kết cung cấp một môi trường trực tuyến mà người dùng có thể tin tưởng và cảm thấy an tâm khi sử dụng.
- **Tối Uu Hóa Trải Nghiệm Người Dùng:** Đối với trang web hẹn hò, việc cải thiện trải nghiệm người dùng là một ưu tiên hàng đầu. Website được thiết kế với giao diện thân thiện, dễ sử dụng và linh hoạt để đáp ứng được sự đa dạng và yêu cầu khác nhau của người dùng.
- **Kiểm Soát Chất Lượng và Đa Dạng:** Một trong những mục tiêu quan trọng của trang web là duy trì một cộng đồng đa dạng và tích cực trên nền tảng của. Em cam kết thực hiện các biện pháp kiểm soát chất lượng để đảm bảo rằng nền tảng chỉ chứa những người dùng chân thành và có ý định tìm kiếm mối quan hệ.
- **Cung Cấp Các Tính Năng Đa Dạng và Linh Hoạt:** Để đáp ứng được nhu cầu và mong muốn đa dạng của người dùng, Em đã phát triển một loạt các tính năng và công cụ linh hoạt. Từ việc tìm kiếm đến kết nối và giao tiếp, mỗi tính năng đều được thiết kế để cung cấp một trải nghiệm tốt nhất cho người dùng.

### 2.3. Khảo sát một số website thực tế

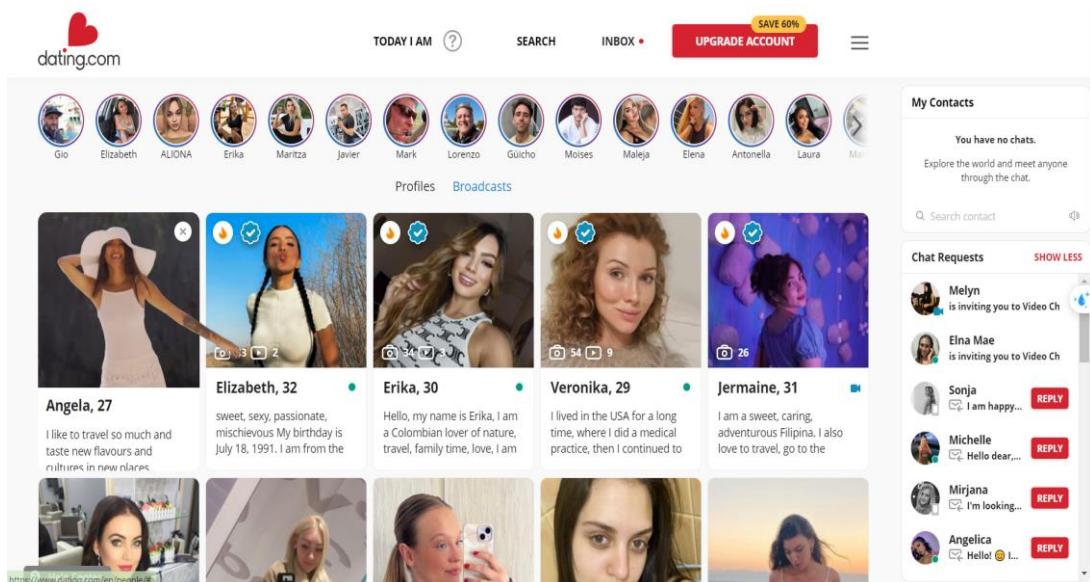


Hình 2.1. Website VietnamCupid

Website VietnamCupid có các chức năng:

- + Đăng nhập: Chức năng cho phép người dùng đăng nhập vào trang web để sử dụng các chức năng của trang web.
- + Đăng ký: Chức năng cho phép người dùng đăng ký tài khoản để có thể truy cập trang web để sử dụng các chức năng của trang web.
- + Tạo hồ sơ: Người dùng có thể tạo một hồ sơ cá nhân với các thông tin như hình ảnh, sở thích, mô tả về bản thân và tiêu chí tìm kiếm.
- + Tìm kiếm và lọc: Cung cấp công cụ tìm kiếm và lọc để người dùng có thể tìm kiếm người khác dựa trên các yêu cầu như độ tuổi, địa điểm, sở thích, v.v.
- + Gửi tin nhắn: Cung cấp tính năng gửi tin nhắn giữa các người dùng để bắt đầu giao tiếp và xây dựng mối quan hệ.
- + Thích và kết nối: Cho phép người dùng dùng thích hoặc thả tim vào hồ sơ của người khác, và nếu có sự quan tâm tương đồng, họ có thể kết nối với nhau.
- + Sự kiện và hoạt động: Cung cấp thông tin về các sự kiện xã hội hoặc hoạt động gặp gỡ trong cộng đồng hẹn hò để người dùng có thể tham gia và gặp gỡ nhau ngoài khung gian trực tuyến.

- + Bảo mật và quản lý hồ sơ: Đảm bảo tính riêng tư và bảo mật cho thông tin cá nhân của người dùng, cung cấp các cài đặt quản lý hồ sơ và kiểm soát quyền truy cập.
- + Matches: Đề xuất những người dùng có độ phù hợp với người dùng hiện tại để hiển thị.
- + Hiển thị trạng thái trực tuyến: Cho phép người dùng xem ai đang trực tuyến trong thời gian thực, giúp họ biết được ai có sẵn và có thể tương tác ngay lập tức.



Hình 2.2. Website dating.com

Website dating.com có các chức năng:

- + Đăng nhập: Chức năng cho phép người dùng đăng nhập vào trang web để sử dụng các chức năng của trang web.
- + Đăng ký: Chức năng cho phép người dùng đăng ký tài khoản để có thể truy cập trang web để sử dụng các chức năng của trang web.
- + Tạo hồ sơ: Người dùng có thể tạo một hồ sơ cá nhân với các thông tin như hình ảnh, sở thích, mô tả về bản thân và tiêu chí tìm kiếm.
- + Xem chi tiết thông tin: Chức năng cho phép người dùng xem chi tiết thông tin về một người dùng khác.

- + Tìm kiếm và lọc: Cung cấp công cụ tìm kiếm và lọc để người dùng có thể tìm kiếm người khác dựa trên các yêu cầu như độ tuổi, địa điểm, sở thích, v.v.
- + Gửi tin nhắn: Cung cấp tính năng gửi tin nhắn giữa các người dùng để bắt đầu giao tiếp và xây dựng mối quan hệ.
- + Thích và kết nối: Cho phép người dùng thích hoặc thả tim vào hồ sơ của người khác, và nếu có sự quan tâm tương đồng, họ có thể kết nối với nhau.
- + Sự kiện và hoạt động: Cung cấp thông tin về các sự kiện xã hội hoặc hoạt động gặp gỡ trong cộng đồng hẹn hò để người dùng có thể tham gia và gặp gỡ nhau ngoài không gian trực tuyến.
- + Bảo mật và quản lý hồ sơ: Đảm bảo tính riêng tư và bảo mật cho thông tin cá nhân của người dùng, cung cấp các cài đặt quản lý hồ sơ và kiểm soát quyền truy cập.
- + Matches: Đề xuất những người dùng có độ phù hợp với người dùng hiện tại để hiển thị.
- + Hiển thị trạng thái trực tuyến: Cho phép người dùng xem ai đang trực tuyến trong thời gian thực, giúp họ biết được ai có sẵn và có thể tương tác ngay lập tức.
- + Đăng status: Đăng tải lên bảng tin các video ngắn và hình ảnh để mọi người có thể tương tác và dễ dàng hơn trong việc tìm đối tượng hẹn hò
- + Đăng bài đăng: Đăng các bài đăng bao gồm content, hình ảnh và video để mọi người có thể tương tác và dễ dàng hơn trong việc tìm đối tượng hẹn hò.

#### **2.4. Các yêu cầu chức năng**

- Khách hàng
  - + Đăng nhập: Chức năng cho phép người dùng đăng nhập vào trang web để sử dụng các chức năng của trang web.

- + Đăng ký: Chức năng cho phép người dùng đăng ký tài khoản để có thể truy cập trang web để sử dụng các chức năng của trang web.
- + Xem chi tiết thông tin: Chức năng cho phép người dùng xem chi tiết thông tin về một người dùng khác.
- + Tạo hồ sơ: Người dùng có thể tạo một hồ sơ cá nhân với các thông tin như hình ảnh, sở thích, mô tả về bản thân và tiêu chí tìm kiếm.
- + Xem chi tiết thông tin: Chức năng cho phép người dùng xem chi tiết thông tin về một người dùng khác.
- + Tìm kiếm và lọc: Cung cấp công cụ tìm kiếm và lọc để người dùng có thể tìm kiếm người khác dựa trên các yêu cầu như độ tuổi, địa điểm, sở thích, v.v.
- + Gửi tin nhắn: Cung cấp tính năng gửi tin nhắn giữa các người dùng để bắt đầu giao tiếp và xây dựng mối quan hệ.
- + Theo dõi: Cho phép người dùng hiện tại theo dõi những người dùng mà họ thấy phù hợp, và nếu có sự quan tâm tương đồng, họ có thể kết nối với nhau.
- + Matches: Đề xuất những người dùng có độ phù hợp với người dùng hiện tại để hiển thị.
- + Hiển thị trạng thái trực tuyến: Cho phép người dùng xem ai đang trực tuyến trong thời gian thực, giúp họ biết được ai có sẵn và có thể tương tác ngay lập tức.
- + Bài đăng: Đăng các bài đăng bao gồm content, hình ảnh và video để mọi người có thể tương tác và dễ dàng hơn trong việc tìm đối tượng hẹn hò.
- + Bình luận và tương tác bài đăng: Mọi người có thể bình luận và trả lời bình luận các bài đăng, like bài đăng và báo cáo bài đăng. Giúp tạo ra các kết nối để mọi người dễ làm quen nhau hơn.

- + Thông báo: Gửi các thông báo đến người dùng khi có người bình luận mới, thả cảm xúc về bài đăng của họ, các bài đăng mới mà người dùng họ đang theo dõi đăng lên.
- + Hẹn hò: Kết nối các cặp đôi với nhau bằng việc đếm số thời gian mà hai người đã hẹn hò với nhau.
- Quản trị viên:
  - + Quản lý tài khoản: Chức năng cho phép quản trị viên phân quyền và khóa tài khoản người dùng.
  - + Quản lý các báo cáo bài đăng: Chức năng cho phép người quản trị quản lý được các bài đăng và xử lý bài đăng nếu bài đăng vi phạm các nguyên tắc cộng đồng.

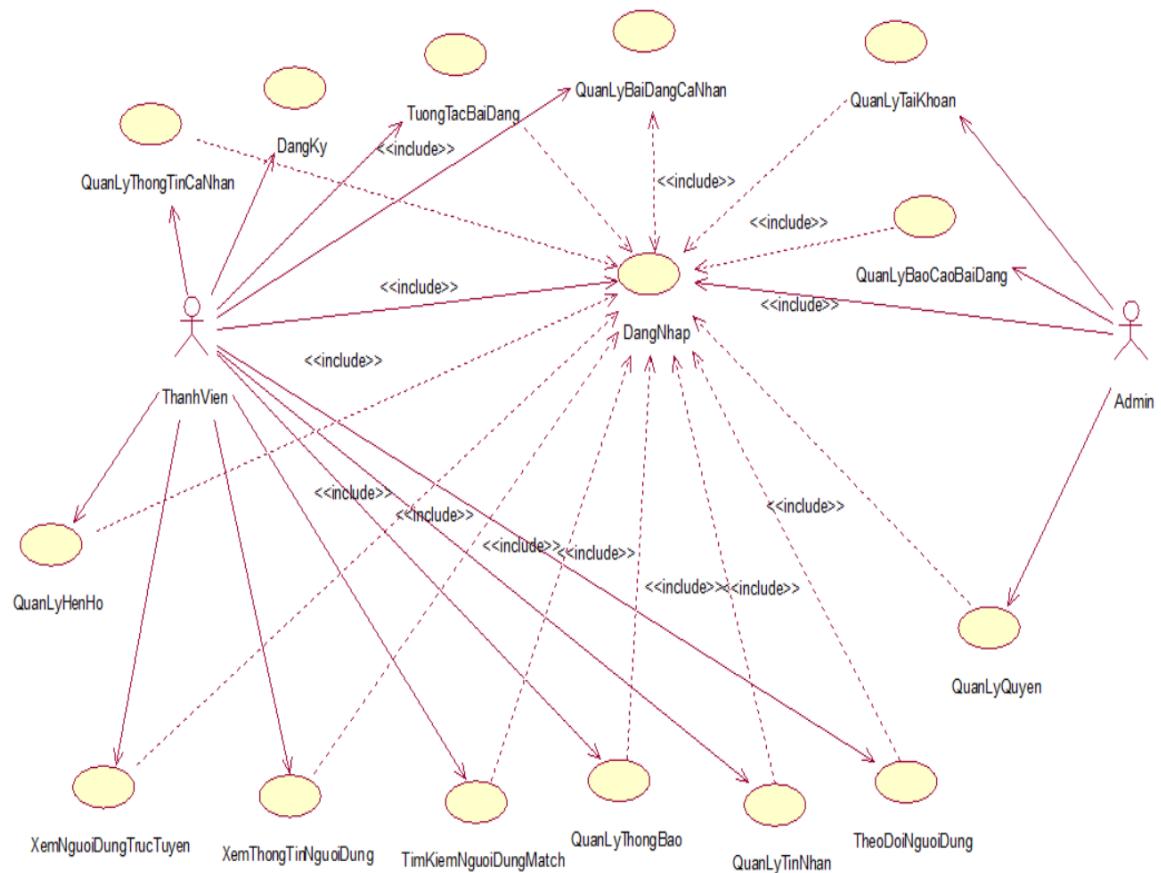
## **2.5. Các yêu cầu phi chức năng**

- Giao diện web được thiết kế dễ sử dụng và có tính thẩm mỹ.
- Hiệu năng: tốc độ, khả năng và độ tin cậy của hệ thống.
- Website bảo mật phân quyền để thực hiện các chức năng của hệ thống.

## CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

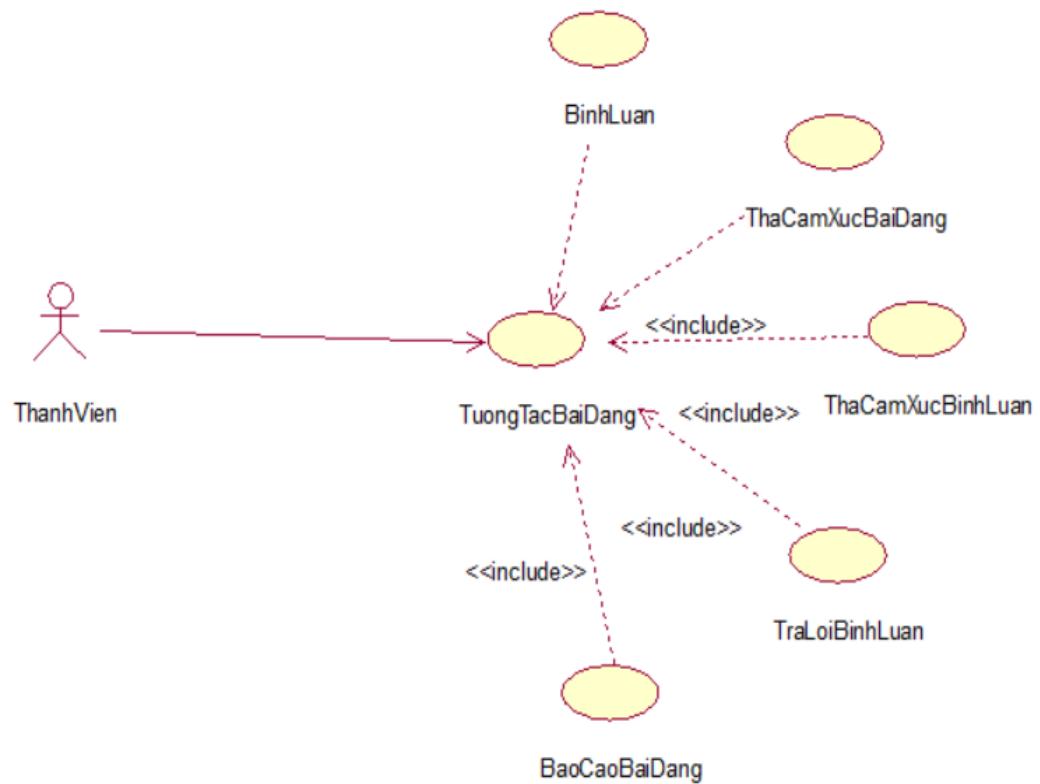
### 3.1. Biểu đồ use case

- Biểu đồ use case tổng quát của hệ thống

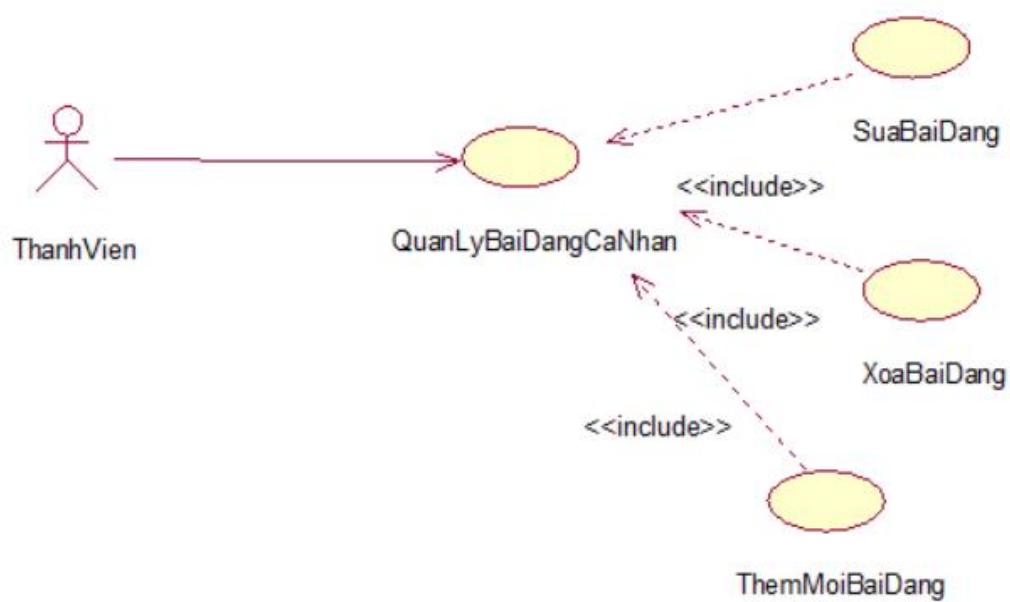


Hình 3.1.1. Biểu đồ usecase tổng quát

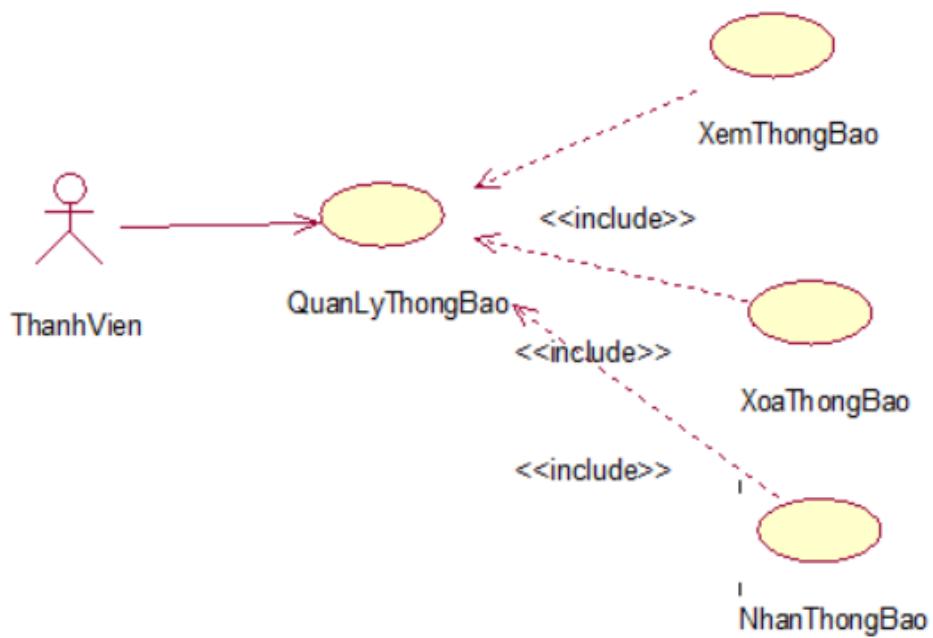
- Biểu đồ use case chi tiết



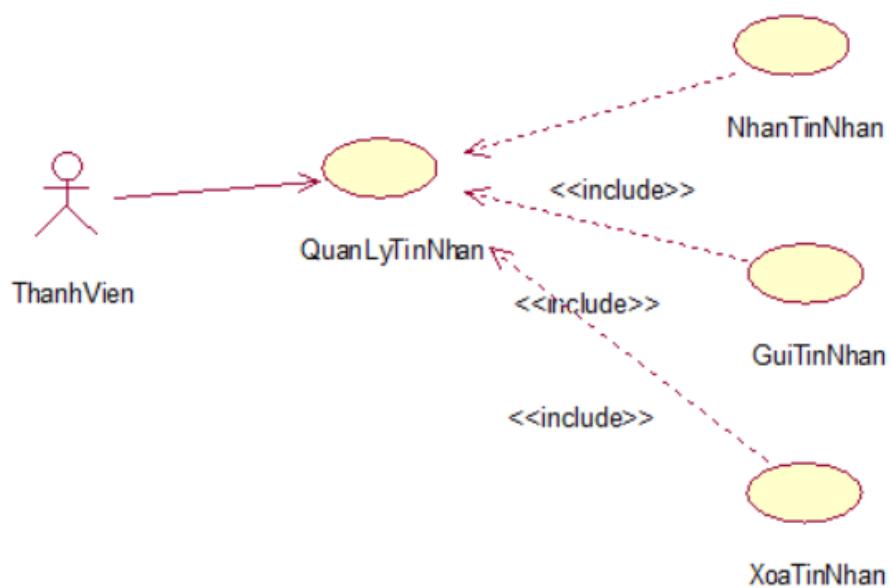
Hình 3.1.2. Sơ đồ use case tương tác bài đăng



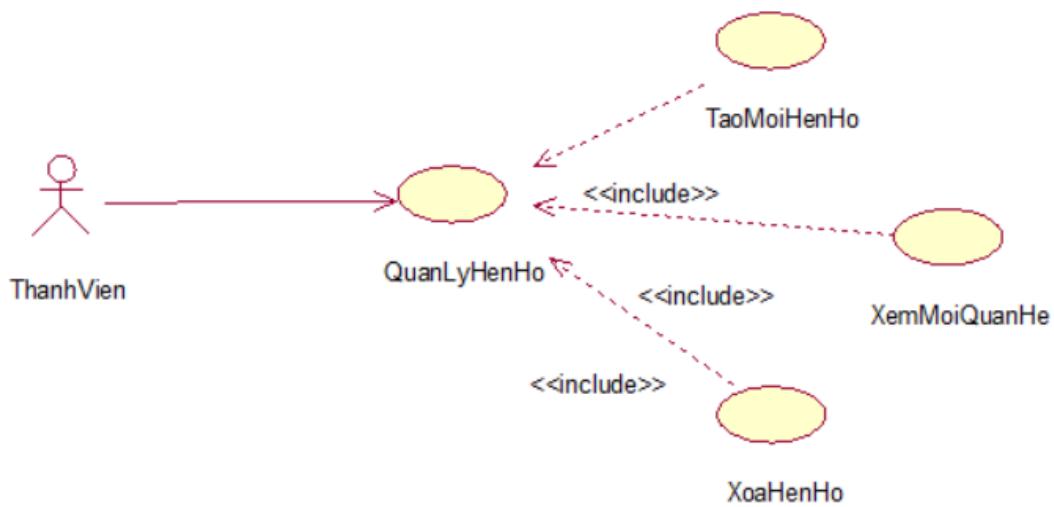
Hình 3.1.3. Sơ đồ use case quản lý bài đăng cá nhân



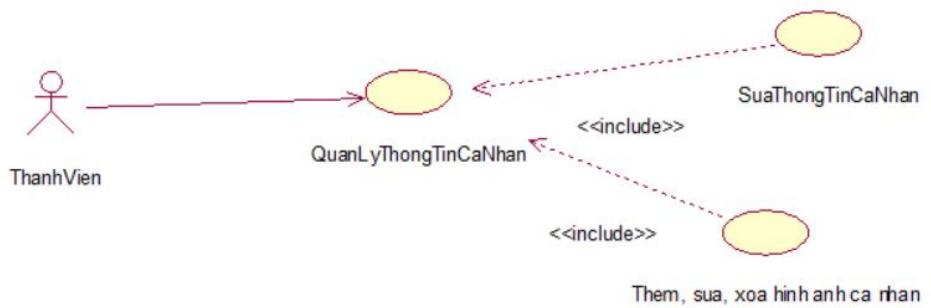
Hình 3.1.4. Sơ đồ use case quản lý thông báo



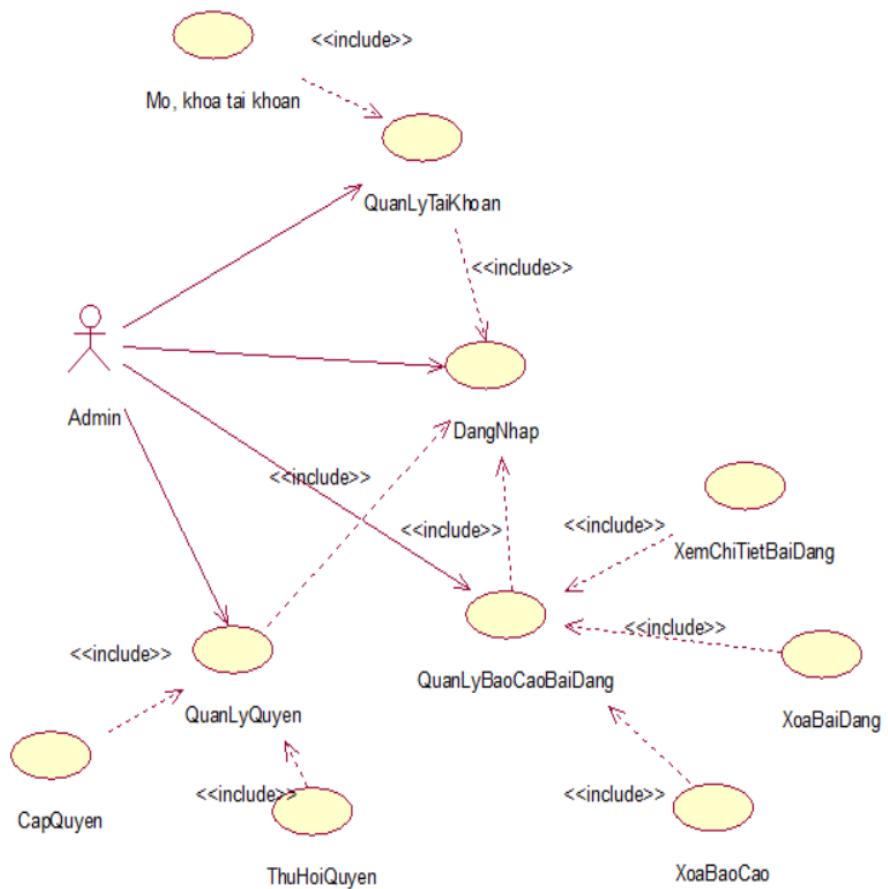
Hình 3.1.5. Sơ đồ use case quản lý tin nhắn



Hình 3.1.6. Sơ đồ use case quản lý hẹn hò



Hình 3.1.7. Sơ đồ use case quản lý thông tin cá nhân



Hình 3.1.8. Biểu đồ use case admin

### 3.2. Đặc tả chi tiết usecase

#### 3.2.1. Use case đăng nhập

**Mô tả văn tắt use case:** Use case này cho phép người dùng đăng nhập để xác định quyền truy cập vào hệ thống.

- **Luồng sự kiện:**

- + Luồng cơ bản:

- 1) Use case này bắt đầu khi người dùng truy cập trang web, hệ thống sẽ hiển thị lên trang đăng nhập.
- 2) Hệ thống yêu cầu người dùng nhập các thông tin bao gồm: tên đăng nhập và mật khẩu.
- 3) Người dùng nhập các thông tin: tên đăng nhập và mật khẩu và bấm nút “Đăng nhập”.
- 4) Hệ thống kiểm tra tên đăng nhập và mật khẩu đã nhập và chuyển hướng sang trang chủ.
- 5) Use case kết thúc.

- + Luồng rẽ nhánh:

- 1) Sai tên đăng nhập hoặc mật khẩu: tại bước 3 trong luồng cơ bản nếu người dùng nhập tên đăng nhập hoặc mật khẩu sai, hệ thống hiển thị thông báo lỗi. Người dùng có thể nhập lại hoặc bỏ qua thao tác, khi đó use case kết thúc.
- 2) Đăng ký: tại bước 3 trong luồng cơ bản, nếu người dùng bấm nút “Đăng ký” use case kết thúc.
- 3) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Không có.

- **Hậu điều kiện:**

Nếu use case thành công, người dùng sẽ đăng nhập được vào hệ thống. Nếu không trạng thái của hệ thống không thay đổi.

- **Điểm mở rộng:**

Không có.

### **3.2.2. Use case đăng ký**

**Mô tả văn tắt use case:** Use case cho phép khách đăng ký tài khoản trên hệ thống:

- **Luồng sự kiện:**

  - + Luồng cơ bản:

- 1) Use case này bắt đầu khi người dùng kích vào nút đăng ký trên màn hình. Hệ thống sẽ hiển thị màn hình đăng ký.
- 2) Người dùng nhập thông tin trên form đăng ký gồm tên hiển thị, tên đăng nhập, sinh nhật, quê quán, mật khẩu rồi kích nút đăng ký. Hệ thống thêm một bản ghi mới vào trong cơ sở dữ liệu.
- 3) Use case kết thúc.

  - + Luồng rẽ nhánh:

- 4) Tại bước 2 trong luồng cơ bản, khi nhập thông tin không thỏa mãn, hệ thống thông báo lỗi. Use case kết thúc.
- 5) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Không có.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### **3.2.3. Use case tìm kiếm người dùng**

**Mô tả vấn tắt use case:** Use case cho phép người dùng tìm kiếm người dùng khác có thông tin cá nhân phù hợp với các thông tin mà người dùng tìm kiếm nhập vào

- **Luồng sự kiện:**

  - + Luồng cơ bản:

- 1) Use case này bắt đầu khi người dùng nhấn vào biểu tượng tìm kiếm trên màn hình. Hệ thống sẽ hiển thị ra các thông tin để người dùng chọn như độ tuổi, cân nặng, đối tượng muốn hẹn hò, nơi muốn hẹn hò.
- 2) Người dùng lựa chọn các thông tin trên và nhấn nút “Tìm kiếm” hệ thống sẽ lấy các thông tin có tiêu đề chứa các từ khóa tìm kiếm từ các bảng USER, DATINGPROFILE, USERINTEREST để lọc ra những người có các thông tin tương đối giống với thông tin mà người dùng đang mong muốn tìm kiếm để hiển thị ra màn hình.
- 3) Use case kết thúc.

  - + Luồng rẽ nhánh:

- 4) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Không có.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### **3.2.4. Use case xem chi tiết người dùng**

**Mô tả văn tắt use case:** Use case cho phép người dùng hiện tại xem thông tin về các thành viên khác.

- **Luồng sự kiện:**

  - + Luồng cơ bản:

- 1) Use case bắt đầu khi người dùng hiện tại nhấn vào biểu tượng “Cá nhân” trên card của người dùng mục tiêu. Hệ thống sẽ lấy thông tin về người dùng bao gồm các sở thích từ bảng USERINTEREST, chiều cao, cân nặng, địa điểm, độ tuổi tìm kiếm muốn hẹn hò từ bảng DatingProfile, mô tả, tuổi, địa chỉ từ bảng USER và hiển thị lên màn hình.

- 2) Use case kết thúc.

  - + Luồng rẽ nhánh:

- 1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Không có.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### 3.2.5. Use case quản lý thông tin cá nhân

**Mô tả vắn tắt use case:** Use case cho phép người dùng quản lý các thông tin cá nhân.

- **Luồng sự kiện:**

- + **Luồng cơ bản:**

- 1) Use case này bắt đầu khi khách hàng kích vào nút profile trên màn hình. Hệ thống sẽ hiển thị màn hình thông tin cá nhân.
- 2) Người dùng nhập thông tin cần chỉnh sửa rồi kích nút “Lưu”. Hệ thống sẽ cập nhật lại thông tin các trường tương ứng với các bảng trong cơ sở dữ liệu. Use case kết thúc
- 3) Cập nhật kho ảnh cá nhân: người dùng kích vào mục hình ảnh hệ thống hiển thị lên màn hình kho ảnh cá nhân.
- 4) Thêm, xóa hình ảnh: Người dùng kích vào “Tải ảnh”, người dùng chọn những hình ảnh ưng ý nhất rồi kích chọn “Tải lên”. Hệ thống sẽ tự động lưu những đường dẫn hình ảnh người dùng vừa tải lên vào cơ sở dữ liệu. Người dùng chọn bức ảnh cần xóa rồi kích “Xóa” hệ thống sẽ xóa hình ảnh người dùng vừa chọn. Use case kết thúc.

- + **Luồng rẽ nhánh:**

- 1) Tại bước 2 trong luồng cơ bản nếu người dùng nhập sai định dạng, hệ thống sẽ thông báo lỗi. Use case kết thúc.
- 2) Tại bước 4 trong luồng cơ bản nếu người dùng tải lên các file không phải file ảnh hệ thống sẽ báo lỗi. Use case kết thúc.
- 3) Tại bước 4 trong luồng cơ bản khi người dùng tải ảnh lên nếu đây là bức ảnh đầu tiên hệ thống sẽ chọn bức ảnh đó làm ảnh đại diện.
- 4) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Không có.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### **3.2.6. Use case nhắn tin**

**Mô tả vắn tắt use case:** Use case cho phép thành viên nhắn tin trực tuyến và nhận các tin nhắn đến với thành viên khác

- **Luồng sự kiện:**

- + Luồng cơ bản

- 1) Use case này bắt đầu khi khách hàng kích vào biểu tượng message trên card của những thành viên khác. Hệ thống sẽ hiển thị màn hình tin nhắn của người dùng hiện tại với người dùng mục tiêu.

- 2) Người dùng nhập các tin nhắn cần gửi đến người dùng mục tiêu và nhấn nút ‘Gửi’. Hệ thống sẽ lưu những tin nhắn giữa người dùng hiện tại và người dùng mục tiêu của họ dữ liệu được lưu vào bảng MESSAGES.

- 3) Tin nhắn được gửi và hiển thị trong giao diện tin nhắn của người nhận và người gửi. Use case kết thúc

- + Luồng rẽ nhánh:

- 1) Tại bước 2 trong luồng cơ bản nếu cả người dùng hiện tại và người dùng mục tiêu đều đang mở giao diện tin nhắn, tin nhắn mới sẽ tự động hiển thị trên cả hai giao diện và cập nhật trạng thái đã đọc.

2) Tại bước 2 trong luồng cơ bản nếu người dùng hiện tại hoặc người dùng mục tiêu không ở trong đoạn chat thì trạng thái tin nhắn được gửi sẽ được cập nhật chưa đọc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Không có.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### **3.2.7. Use case quản lý bài đăng**

**Mô tả vắn tắt use case:** Use case này cho phép người dùng tạo mới bài đăng, sửa, xóa bài đăng.

- **Luồng sự kiện:**

- + Luồng cơ bản:

- 1) Thêm mới bài đăng:

a) Use case này bắt đầu khi người dùng kích vào mục tạo bài đăng trên màn hình bài đăng. Hệ thống sẽ hiển thị lên form bao gồm nội dung, hình ảnh để người dùng tạo mới bài đăng.

b) Người dùng nhập các thông tin (nội dung, hình ảnh) và kích vào nút “Post”. Hệ thống sẽ tạo ra nội dung và hình ảnh và lưu các thông tin(Mã bài đăng,mã người đăng, nội dung, hình ảnh, thời gian tạo, thời gian chỉnh sửa) vào bảng POST và hiển thị danh sách bài đăng vừa cập nhật lên màn hình. Use case kết thúc.

- 2) Sửa thông tin bài đăng:

- a) Người dùng kích vào nút “Sửa” trên bài đăng. Hệ thống sẽ hiển thị lên màn hình với các thông tin cũ bài đăng(nội dung, hình ảnh).
  - b) Người dùng sửa thông tin(nội dung, hình ảnh) của bài đăng và kích vào nút “Sửa”. Hệ thống sẽ cập nhật thông tin bài đăng có trong bảng POST và hiển thị lên bài đăng đã cập nhật. Use case kết thúc.
- 3) Xóa bài đăng:
- a) Người dùng kích vào nút “Xóa” trên bài đăng. Hệ thống sẽ xóa bài đăng khỏi bảng POST và hiển thị danh sách bài đăng đã cập nhật. Use case kết thúc.
- + Luồng rẽ nhánh:
- 1) Tại bước 1b nếu người dùng bấm vào nút “Hủy”. Hệ thống sẽ hiển thị danh sách bài đăng lúc đầu. Use case kết thúc.
  - 2) Tại bước 2b nếu người dùng bấm vào nút “Hủy”. Hệ thống sẽ hiển thị danh sách bài đăng lúc đầu. Use case kết thúc.
  - 3) Tại bước 3b người dùng bấm vào nút “Hủy”. Hệ thống sẽ hiển thị danh sách bài đăng lúc đầu. Use case kết thúc.
- **Các yêu cầu đặc biệt:**  
Không có.
- **Tiền điều kiện:**  
Người dùng đăng nhập thành công.
- **Hậu điều kiện:**  
Không có.
- **Điểm mở rộng:**  
Không có.

### 3.2.8. Use case tương tác bài đăng

**Mô tả văn tắt use case:** Use case này cho phép người dùng tạo mới bình luận, sửa, xóa bình luận bài đăng, thả biểu tượng cảm xúc, báo cáo bài đăng.

- **Luồng sự kiện:**

+ Luồng cơ bản:

1) Tạo bình luận:

- a) Use case này bắt đầu khi người dùng nhấn vào mục bình luận trên bài đăng.
- b) Hệ thống sẽ hiển thị lên danh sách bình luận và form mẫu tạo bình luận.
- c) Người dùng nhập bình luận của mình và nhấn “Gửi” hệ thống sẽ lưu bình luận của người dùng vào cơ sở dữ liệu và hiển thị lên danh sách bình luận vừa tạo.

2) Trả lời bình luận:

- a) Use case này bắt đầu khi người dùng nhấn vào mục “Trả lời bình luận” trên 1 bình luận. Hệ thống sẽ hiển thị ra form trả lời.
- b) Người dùng nhập thông tin bình luận cần phản hồi và nhấn “Gửi”. Hệ thống sẽ lưu bình luận vừa tạo vào cơ sở dữ liệu và hiển thị lên bình luận vừa tạo. Use case kết thúc

3) Sửa bình luận:

- a) Người dùng nhấn vào nút “Sửa” trên bình luận của mình. Hệ thống sẽ hiển thị lên bình luận cũ của người dùng.
- b) Người dùng sửa đổi bình luận của mình và nhấn nút “Gửi”, hệ thống sẽ lưu bình luận vừa sửa vào cơ sở dữ liệu và hiển thị lên bình luận vừa sửa. Use case kết thúc.

4) Xóa bình luận:

- a) Người dùng nhấn vào nút xóa trên bình luận. Hệ thống sẽ hiển thị lên màn hình “Xóa bình luận” yêu cầu xác nhận xóa.
- b) Người dùng nhấn nút “Xóa”. Hệ thống sẽ xóa bình luận và hiển thị danh sách bình luận đã cập nhật. Use case kết thúc.

5) Thả cảm xúc:

- a) Người dùng nhấn vào nút cảm xúc trên bài đăng hoặc bình luận bài đăng. Hệ thống sẽ hiển thị lên màn hình danh sách cảm xúc.
- b) Người dùng chọn một cảm xúc. Hệ thống sẽ lưu biểu tượng cảm xúc của người dùng và hiển thị lên biểu tượng cảm xúc vừa tạo.

6) Báo cáo bài đăng:

- a) Người dùng nhấn vào mục báo cáo trên bài đăng. Hệ thống sẽ hiển thị ra form báo cáo bao gồm các vi phạm bài đăng và mô tả báo cáo vi phạm.
- b) Người dùng chọn một vi phạm và nhập mô tả vi phạm rồi nhấn nút “Gửi”. Hệ thống sẽ lưu báo cáo vào cơ sở dữ liệu và gửi báo cáo lên admin xử lý.

7) Use case kết thúc

+ **Luồng rẽ nhánh:**

- 1) Tại bước 1b, 2a nếu người dùng bấm vào nút “Hủy”. Hệ thống sẽ hiển thị danh sách bình luận lúc đầu. Use case kết thúc.
- 2) Tại bước 3b nếu người dùng bấm vào nút “Hủy”. Hệ thống sẽ hiển thị danh sách bình luận lúc đầu. Use case kết thúc.
- 3) Tại bước 4a người dùng bấm vào nút “Hủy”. Hệ thống sẽ hiển thị danh sách bình luận lúc đầu. Use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Người dùng đăng nhập thành công.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### **3.2.9. Use case quản lý người dùng**

**Mô tả vắn tắt use case:** Use case này cho phép quản trị viên phân quyền, khóa, mở các tài khoản.

- **Luồng sự kiện:**

- + Luồng cơ bản:

- 1) UC bắt đầu khi quản trị viên kích vào quản lý vai trò trong màn hình quản trị viên.
- 2) Gán quyền: Quản trị viên ấn vào nút gán quyền, hệ thống sẽ hiển thị lên popup gồm tất cả các quyền và quản trị viên sẽ chọn quyền gán vào vai trò và ấn lưu. Thông tin sẽ được lưu lại vào trong cơ sở dữ liệu.
- 3) Khóa, mở tài khoản: quản trị viên kích vào nút khóa, mở tài khoản trên dòng tài khoản. Hệ thống sẽ update thông vào cơ sở dữ liệu. Các tài khoản bị khóa sẽ không đăng nhập được cho đến khi được mở khóa. Use case kết thúc

- + Luồng rẽ nhánh:

- 1) Tại thời điểm nào mà Admin không kết nối được server hoặc server không kết nối được cơ sở dữ liệu thì hệ thống sẽ xuất hiện thông báo lỗi và UC sẽ kết thúc.

- **Các yêu cầu đặc biệt:**

Đăng nhập thành công vào hệ thống với vai trò là quản lý hoặc nhân viên.

- **Tiền điều kiện:**

Đăng nhập vào admin và có quyền quản lý vai trò.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### **3.2.10. Use case quản lý báo cáo bài đăng**

**Mô tả ngắn tắt use case:** Use case này cho phép quản trị viên quản lý và xóa các bài đăng vi phạm.

- **Luồng sự kiện:**

- + Luồng cơ bản:

- 1) Use case bắt đầu khi người quản trị kích vào mục “Quản lý báo cáo bài đăng”. Hệ thống sẽ lấy lên danh sách các bài đăng đã được người dùng báo cáo và hiển thị lên màn hình.
- 2) Người quản trị click “Xem” hệ thống sẽ hiển thị lên chi tiết bài đăng bị tố cáo. Nếu bài đăng vi phạm người quản trị sẽ click “Xóa bài đăng” hệ thống sẽ xóa bài đăng bị báo cáo và cập nhật danh sách các bài đăng bị tố cáo và hiển thị lên màn hình. Use case kết thúc.

- + Luồng rẽ nhánh:

- 1) Tại thời điểm nào mà client không kết nối được server hoặc server không kết nối được cơ sở dữ liệu thì hệ thống sẽ xuất hiện thông báo lỗi và UC sẽ kết thúc.

- **Các yêu cầu đặc biệt:**

Đăng nhập thành công vào hệ thống với vai trò là quản lý hoặc nhân viên.

- **Tiền điều kiện:**

Đăng nhập vào quản trị viên và có quyền quản lý vai trò.

- **Hậu điều kiện:**

Không có.

- **Điểm mở rộng:**

Không có.

### 3.2.11. Use case quản lý thông báo

**Mô tả ngắn tắt:** Use case này cho phép người nhận thông báo theo thời gian thực, và quản lý các thông báo.

- **Luồng sự kiện:**

- + Luồng cơ bản:

- 1) Nhận thông báo mới

- a) Use case này được bắt đầu khi có người dùng khác bình luận và thả cảm xúc về bài đăng của người dùng hiện tại, hoặc hệ thống thông báo nhắc nhở người dùng về một sự kiện nào đó.

- b) Nhận số lượng thông báo của icon quả chuông ở header sẽ thay đổi. Người dùng kích vào icon thông báo, hệ thống sẽ hiển thị lên taps “Chưa xem” và “Đã xem”

- c) Khi người dùng chọn “Chưa xem” hệ thống sẽ hiển thị lên danh sách thông báo chưa xem. Người dùng nhấn vào 1 thông báo hệ thống sẽ hiển thị chi tiết về nội dung thông báo. Hệ thống sẽ cập nhật lại thông báo vừa xem đã xem và hiển thị danh sách thông báo vừa cập nhật lên tap “Đã xem”.

Use case kết thúc

- 2) Xóa thông báo

- a) Người dùng kích chọn “Xóa thông báo” trên một thông báo.

Hệ thống sẽ hiển thị xác nhận xóa

- b) Người dùng kích chọn “Đồng ý” hệ thống sẽ xóa thông báo và hiển thị danh sách thông báo vừa cập nhật. Use case kết thúc

- + Luồng rẽ nhánh:

- 1) Người dùng click vào một thông báo của bài đăng hệ thống sẽ chuyển người dùng đến bài đăng đó và thông báo sẽ ẩn mờ đi.

2) Tại bất cứ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được tới server thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

Không có.

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào hệ thống để thực hiện use case này.

- **Hậu điều kiện:**

Không có

- **Điểm mở rộng:**

Khi click vào icon quả chuông ở header thì hệ thống sẽ hiển thị danh sách thông báo

### **3.2.12. Use case hẹn hò**

**Mô tả vắn tắt use case:** Use case cho phép người dùng hẹn hò với nhau.

- **Luồng sự kiện**

- + Luồng cơ bản:

- 1) Đăng ký hẹn hò:

a) Use case này bắt đầu khi người dùng hiện tại kích vào “Dating” trong trang cá nhân của người dùng mục tiêu. Hệ thống sẽ tạo mới thông tin bao gồm SenderId id người gửi, CrushId id người nhận, Status được đặt là Waiting và lưu vào bảng DatingRequest trong cơ sở dữ liệu. Lưu thành công hệ thống gửi thông báo yêu cầu xác nhận hẹn hò đến người dùng mục tiêu.

b) Người dùng mục tiêu kích chọn “Xác nhận” hệ thống sẽ cập nhật cột Status trong bảng DatingRequest thành “Confirmed”. Hệ thống cập nhật lại thông tin và hiển thị thông báo ghép đôi thành công

c) Người dùng hiện tại kích chọn “Dating” trên thanh menu hệ thống sẽ hiển thị thông tin hẹn hò của hai người. Use case kết thúc.

## 2) Kết thúc hẹn hò:

a) Người dùng kích chọn “Kết thúc hẹn hò” trên màn hình hẹn hò của 2 người. Hệ thống sẽ hiển thị form xác nhận.

b) Người dùng chọn xác nhận thông tin về mối quan hệ hẹn hò của 2 người sẽ bị xóa khỏi cơ sở dữ liệu. Hệ thống gửi thông báo đã kết thúc mối quan hệ và hiển thị màn hình mặc định. Use case kết thúc

### + Luồng rẽ nhánh:

1) Tại bước 1a nếu người dùng hiện tại đang trong một mối quan hệ hẹn hò với người khác mà vẫn gửi yêu cầu hẹn hò đến người khác hệ thống sẽ đưa ra thông báo “Không thể gửi yêu cầu hẹn hò vì bạn đang ở trong một mối quan hệ khác”. Use case kết thúc.

2) Tại bước 1b nếu người dùng mục tiêu chọn từ chối hệ thống sẽ gửi thông báo đến người gửi yêu cầu “Yêu cầu hẹn hò của bạn đã bị từ chối”. Use case kết thúc.

### - Các yêu cầu đặc biệt:

Không có.

### - Tiền điều kiện:

Người dùng đã đăng nhập vào hệ thống để thực hiện use case này.

### - Hậu điều kiện:

Không có

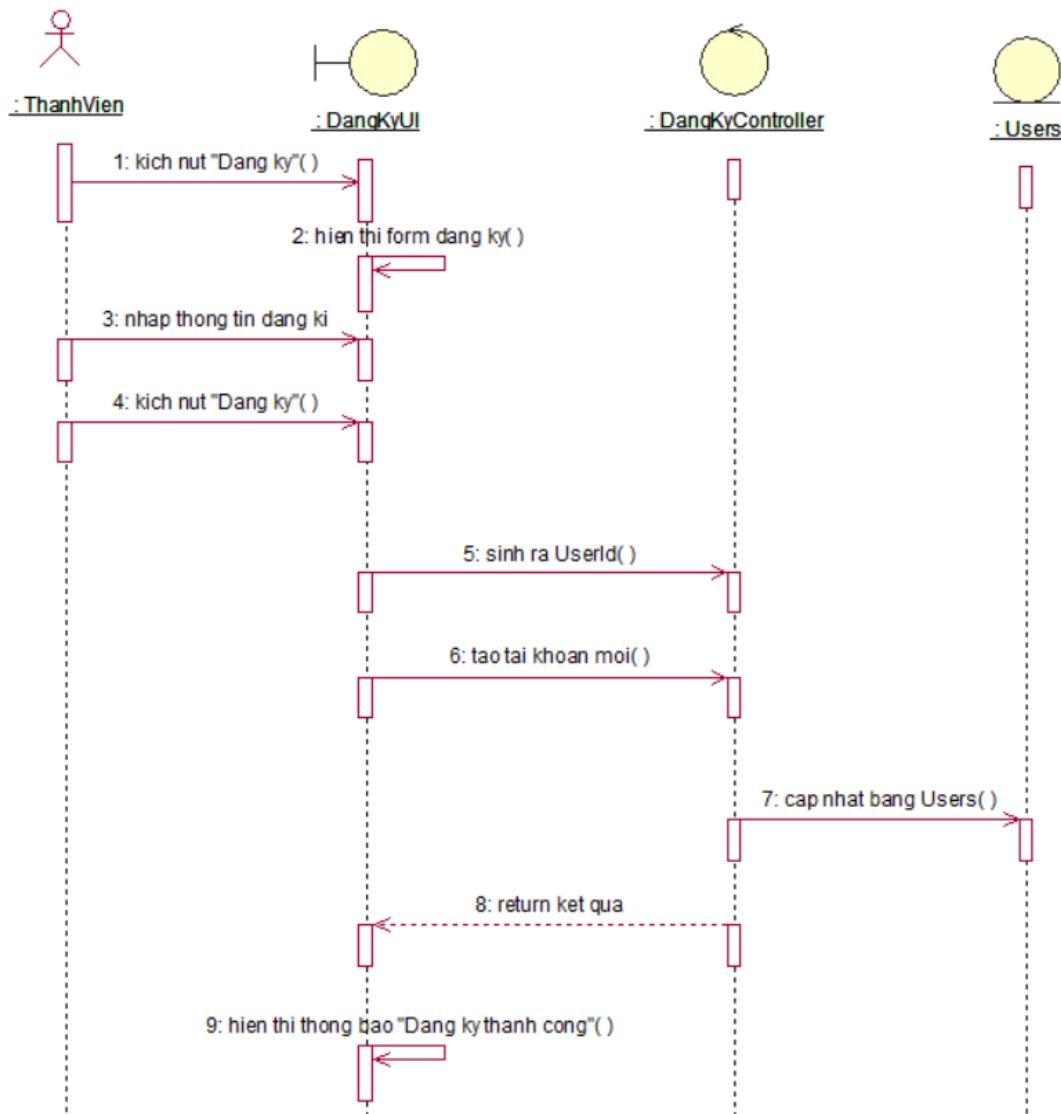
### - Điểm mở rộng:

Khi click vào icon quả chuông ở header thì hệ thống sẽ hiển thị danh sách thông báo.

### 3.3. Phân tích use case

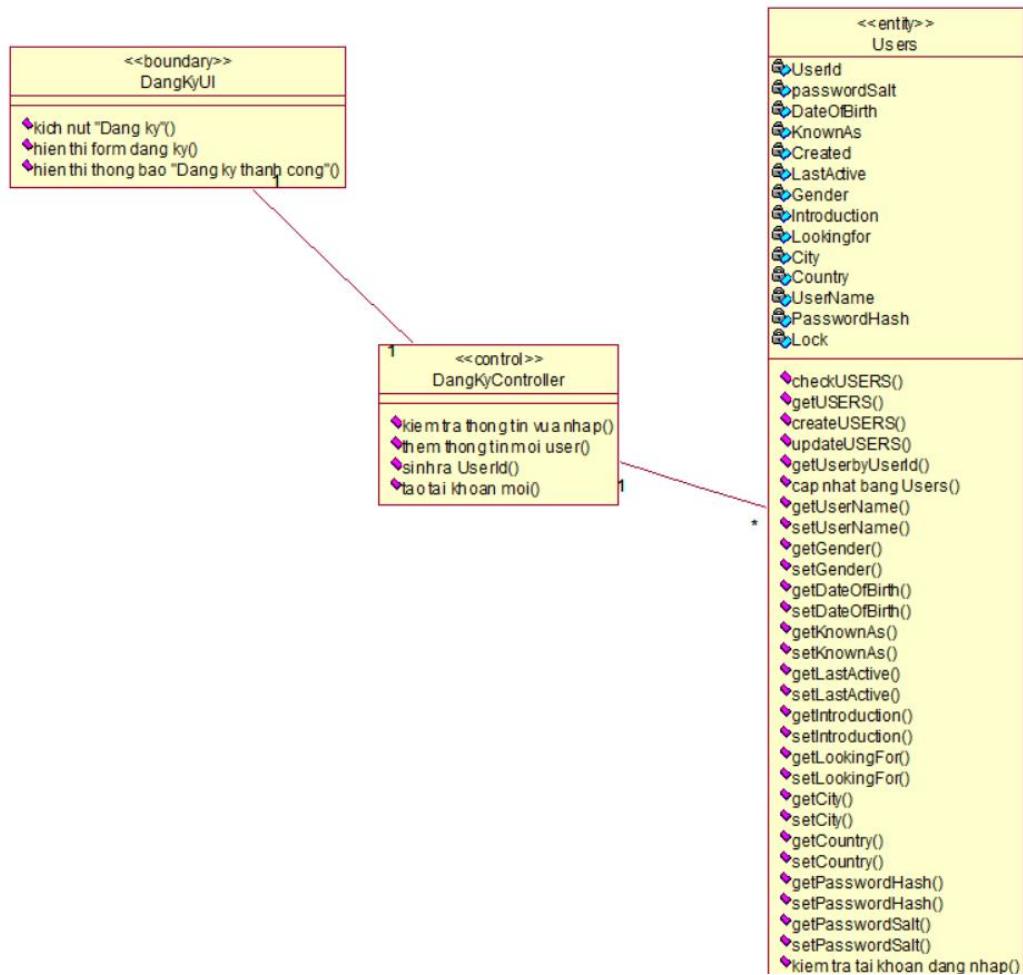
#### 3.3.1. Use case đăng ký

❖ Biểu đồ trình tự:



Hình 3.3.1. Biểu đồ trình tự use case đăng ký

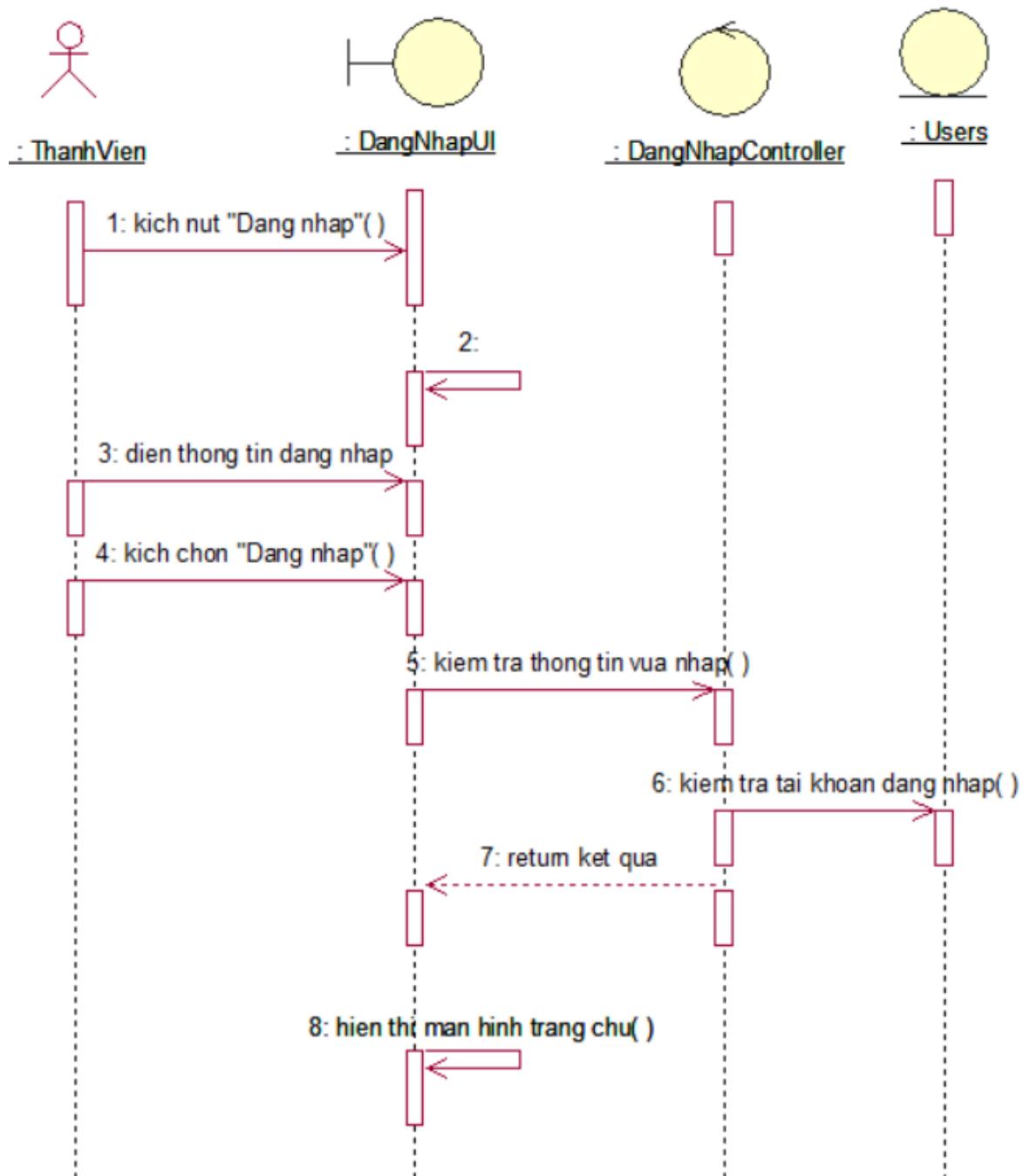
❖ Biểu đồ VOPC



Hình 3.3.2. Biểu đồ VOPC use case đăng ký

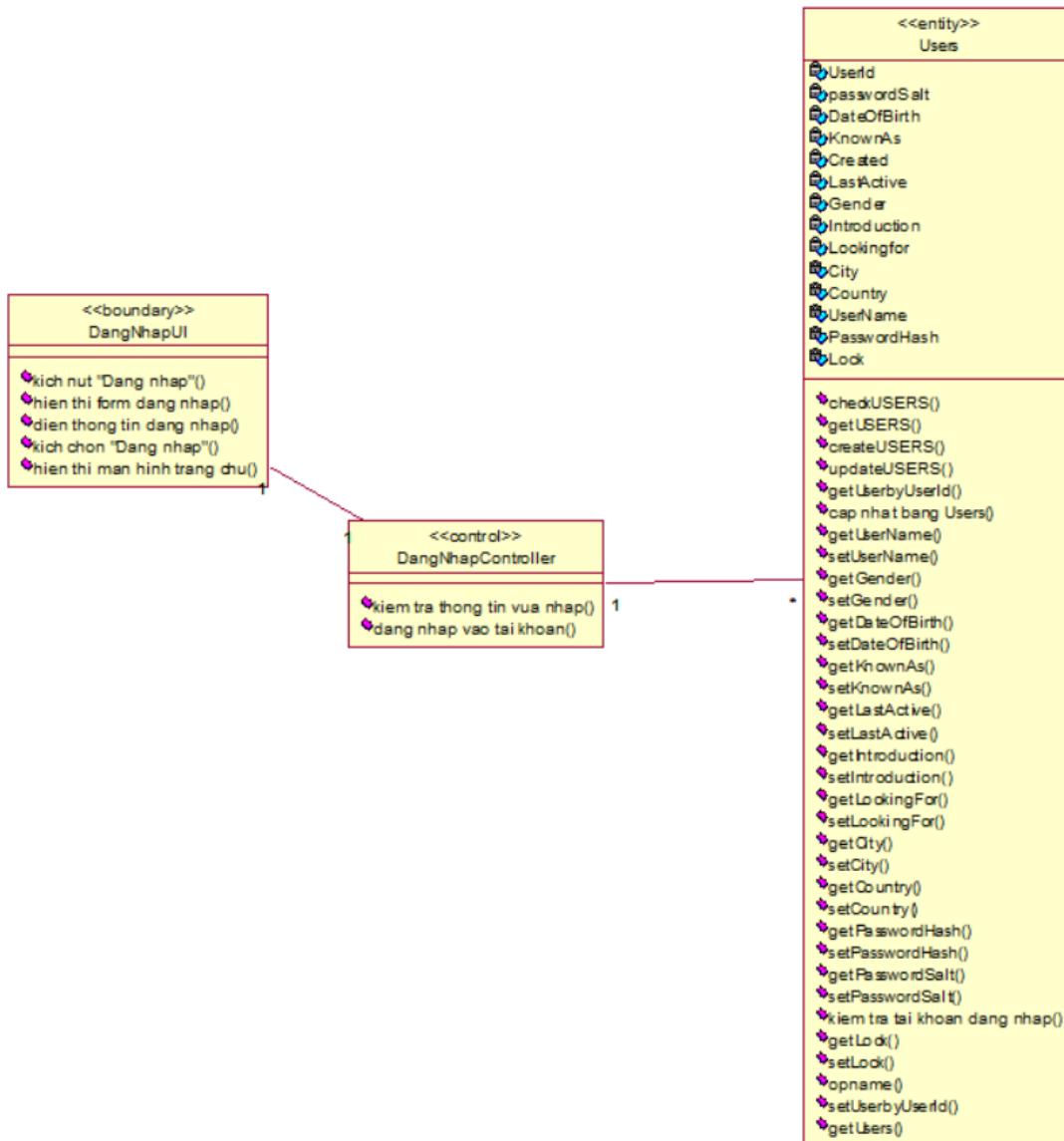
### 3.3.2. Use case đăng nhập

❖ Biểu đồ trình tự



Hình 3.3.3. Biểu đồ trình tự use case đăng nhập

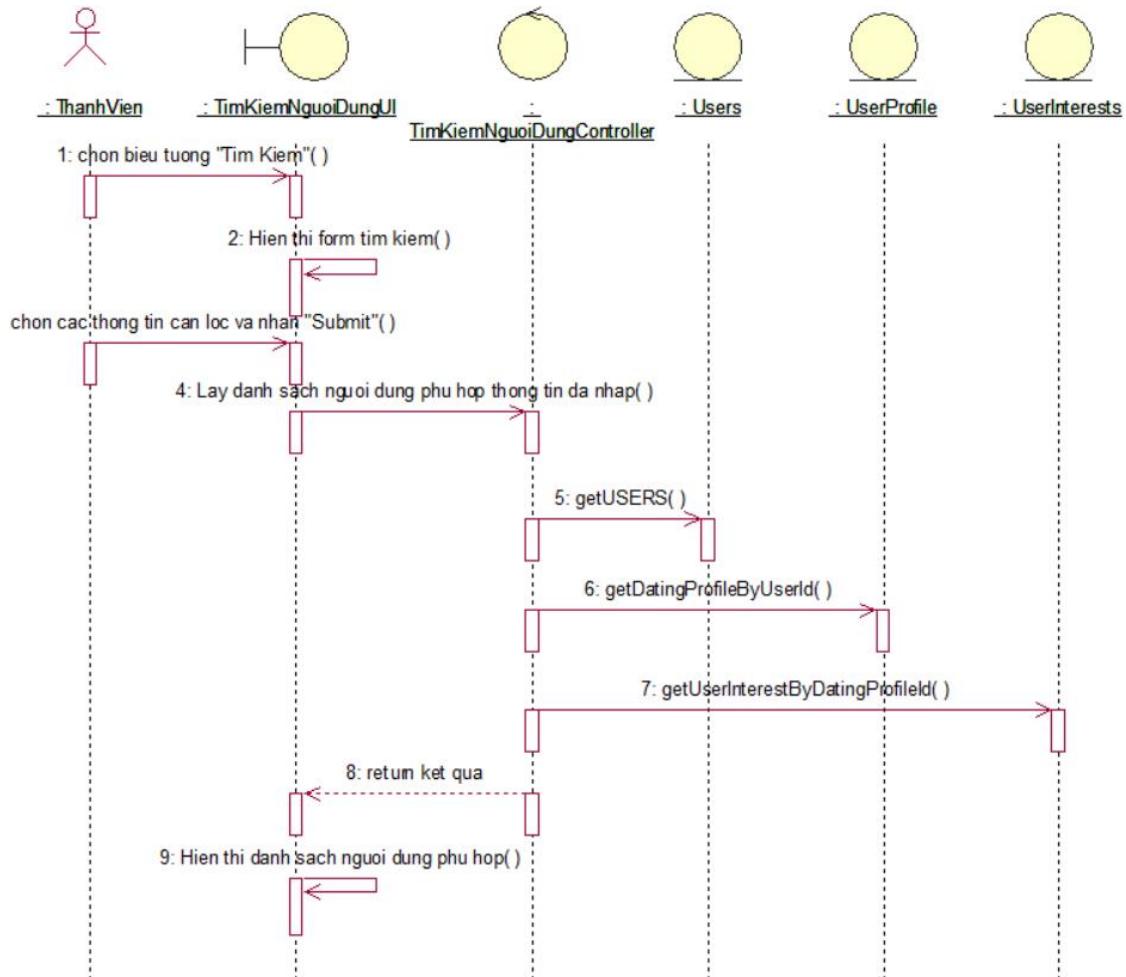
❖ Biểu đồ VOPC



Hình 3.3.4. Biểu đồ VOPC đăng nhập

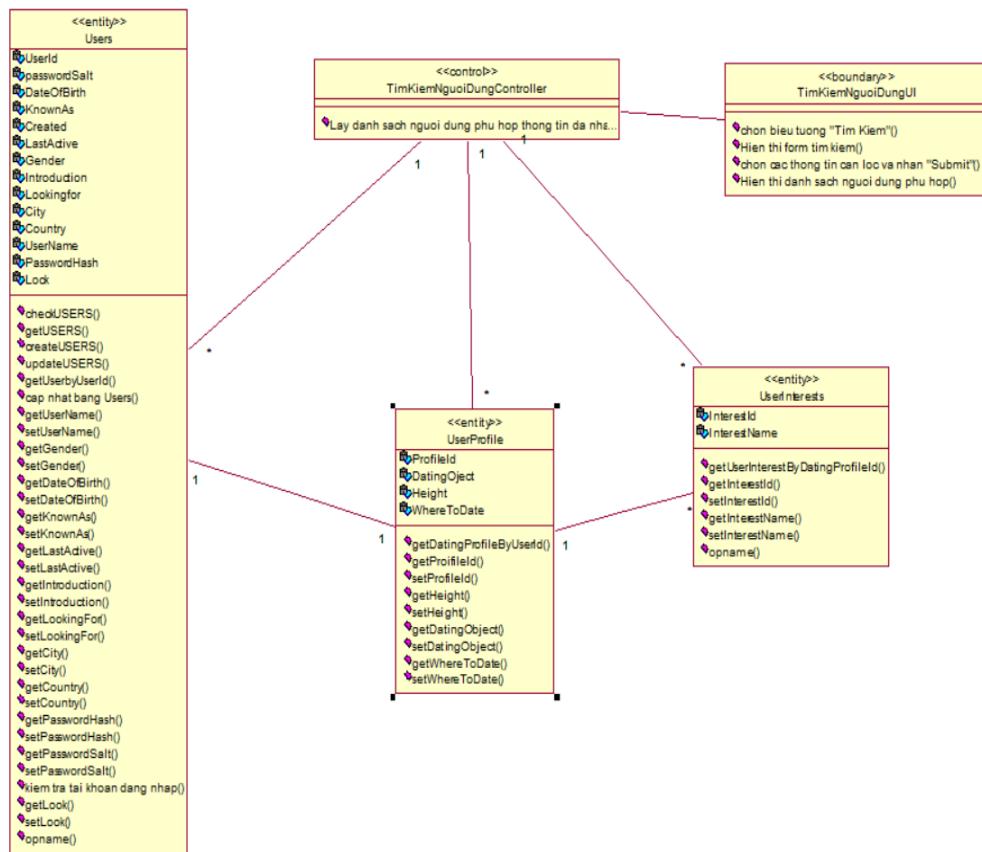
### 3.3.3. Use case tìm kiếm người dùng

❖ Biểu đồ trình tự



Hình 3.3.5. Biểu đồ trình tự use case tìm kiếm người dùng

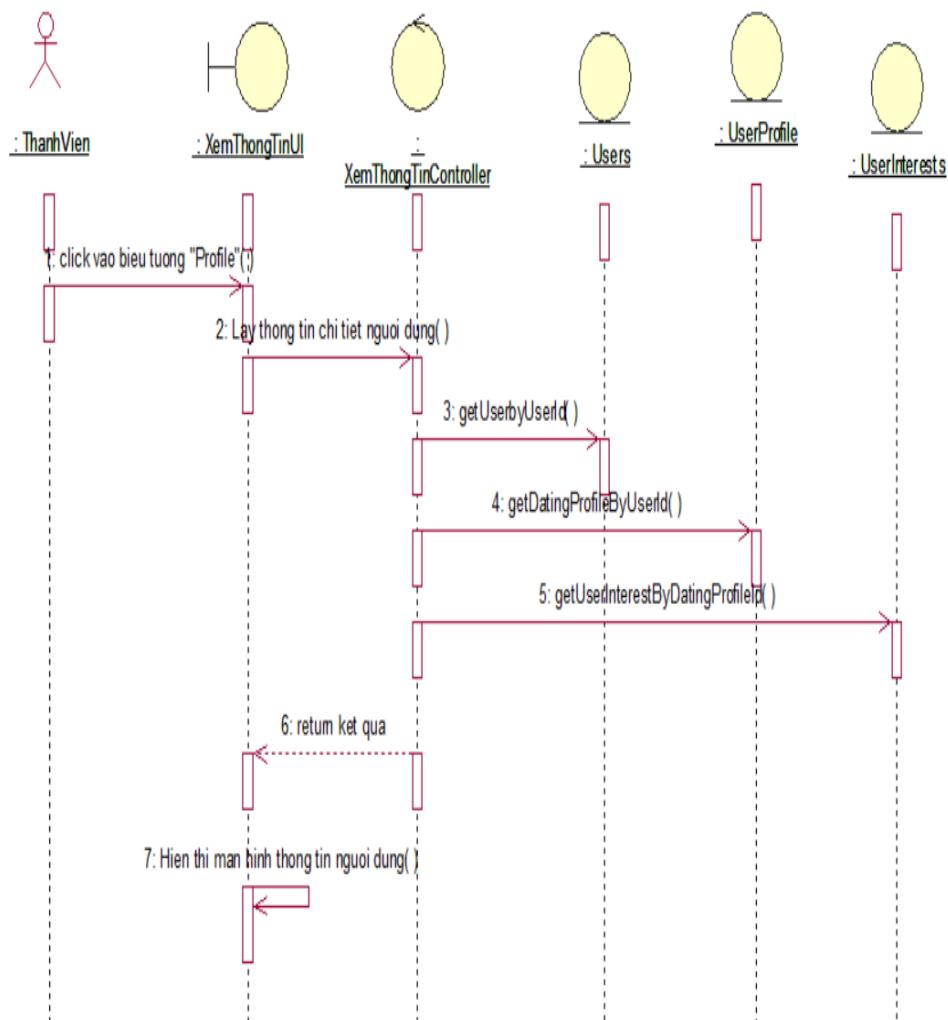
## ❖ Biểu đồ VOPC



Hình 3.3.6. Biểu đồ VOPC tìm kiếm người dùng

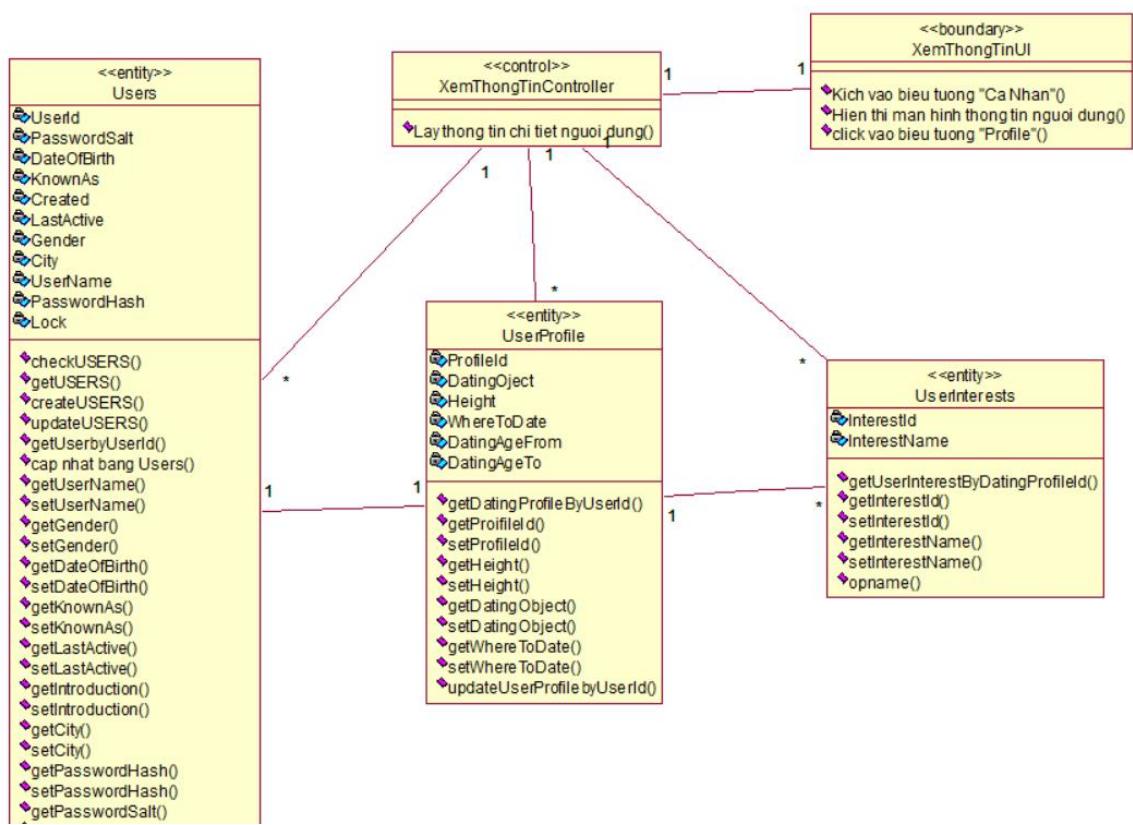
### 3.3.4. Use case xem chi tiết người dùng

❖ Biểu đồ trình tự



Hình 3.3.7. Biểu đồ trình tự use case xem chi tiết người dùng

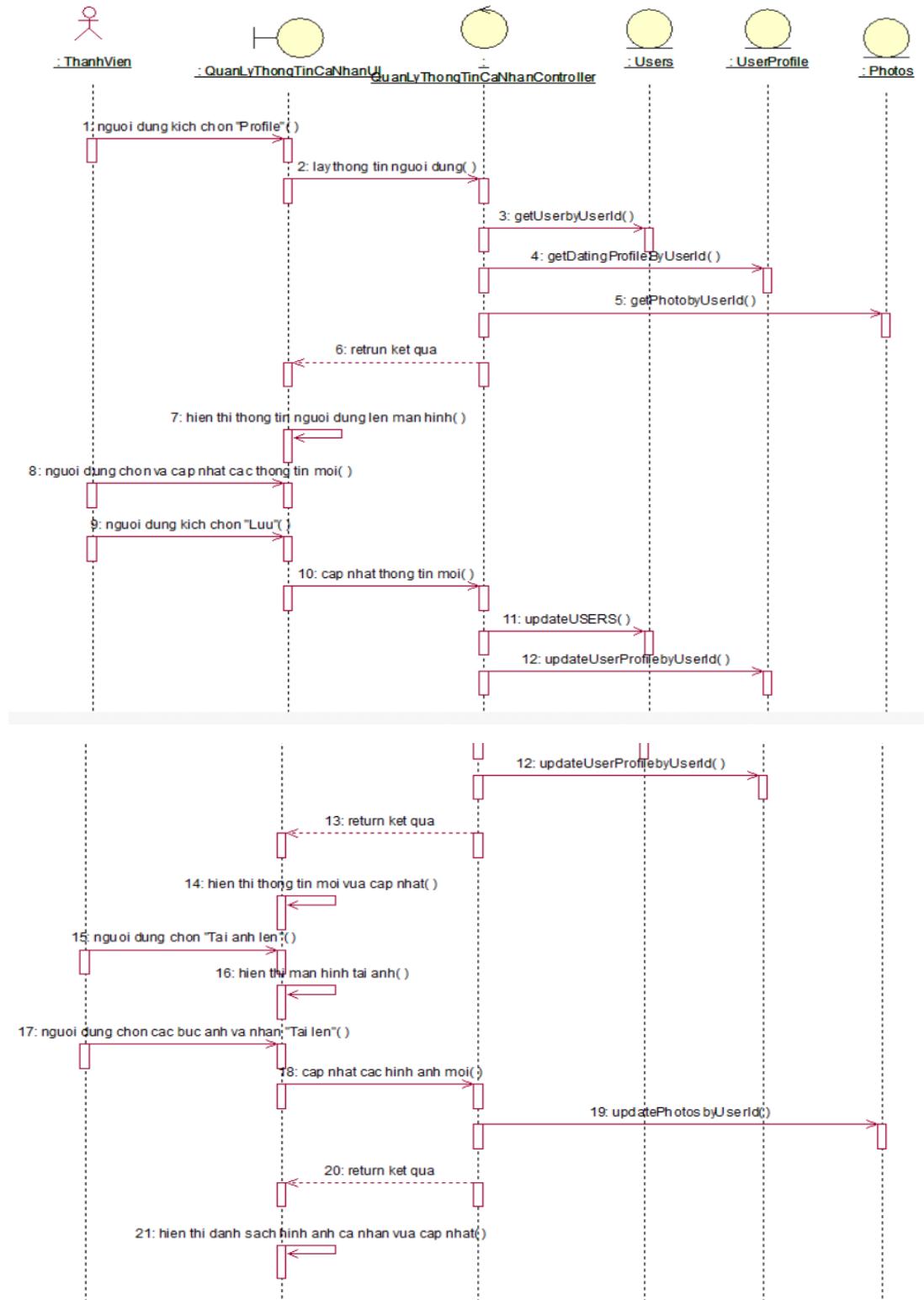
❖ Biểu đồ VOPC



Hình 3.3.8. Biểu đồ VOPC use case xem chi tiết người dùng

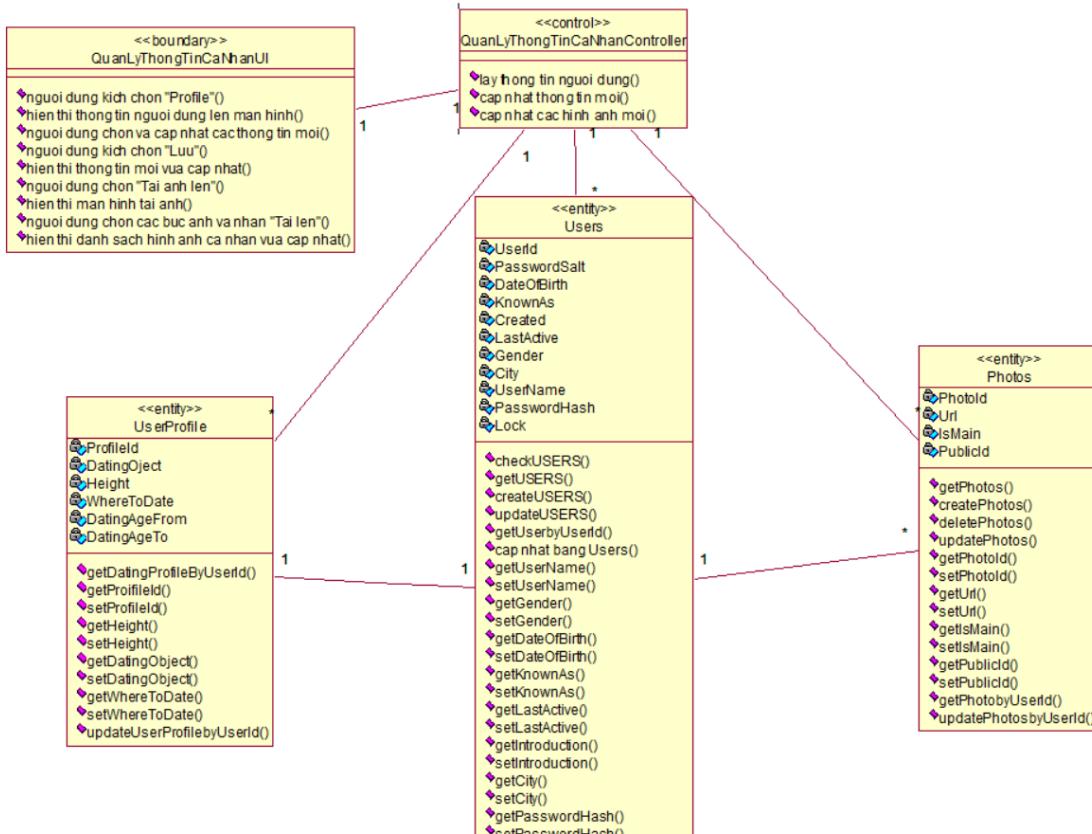
### 3.3.5. Use case quản lý thông tin cá nhân

#### ❖ Biểu đồ trình tự



Hình 3.3.9. Biểu đồ trình tự usecase sửa thông tin cá nhân

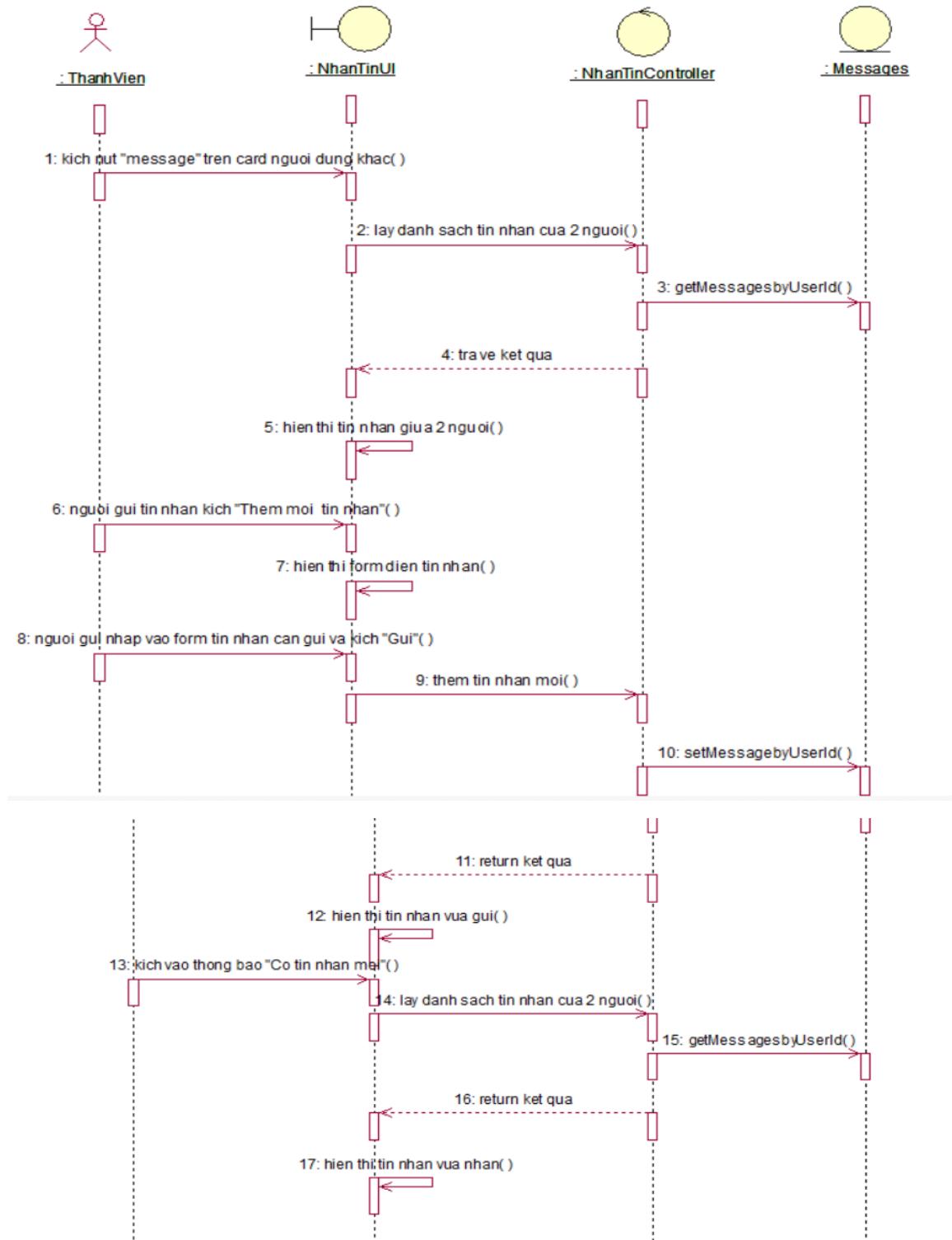
## ❖ Biểu đồ VOPC



Hình 3.3.10. Biểu đồ VOPC use case quản lý thông tin cá nhân

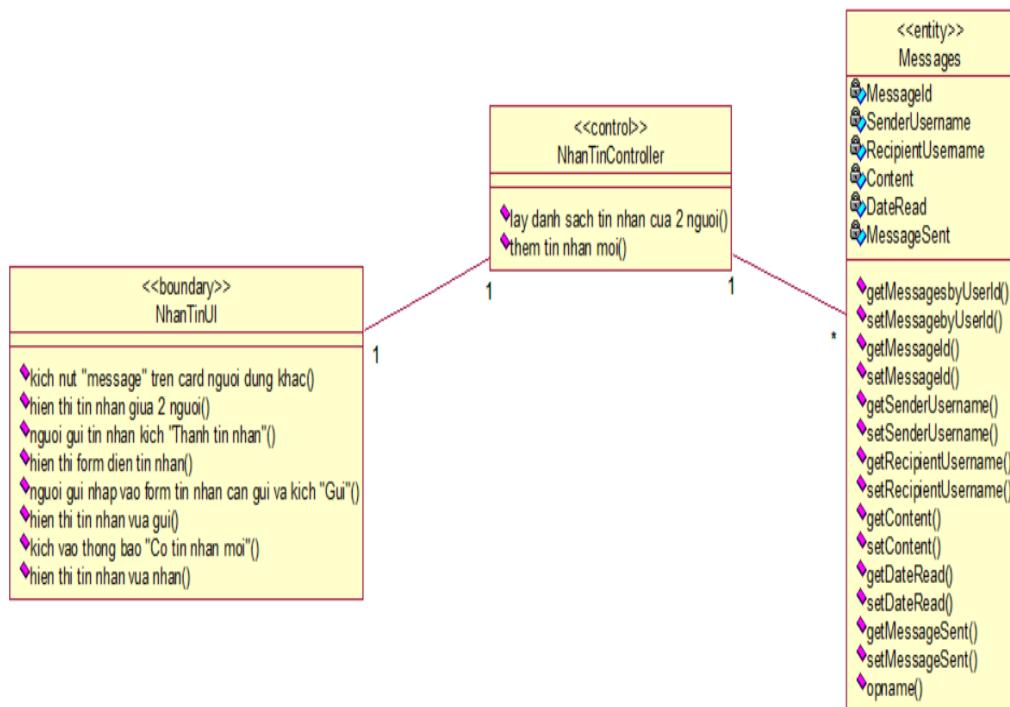
### 3.3.6. Use case nhắn tin

❖ Biểu đồ trình tự



3.3.11. Biểu đồ trình tự use case nhắn tin

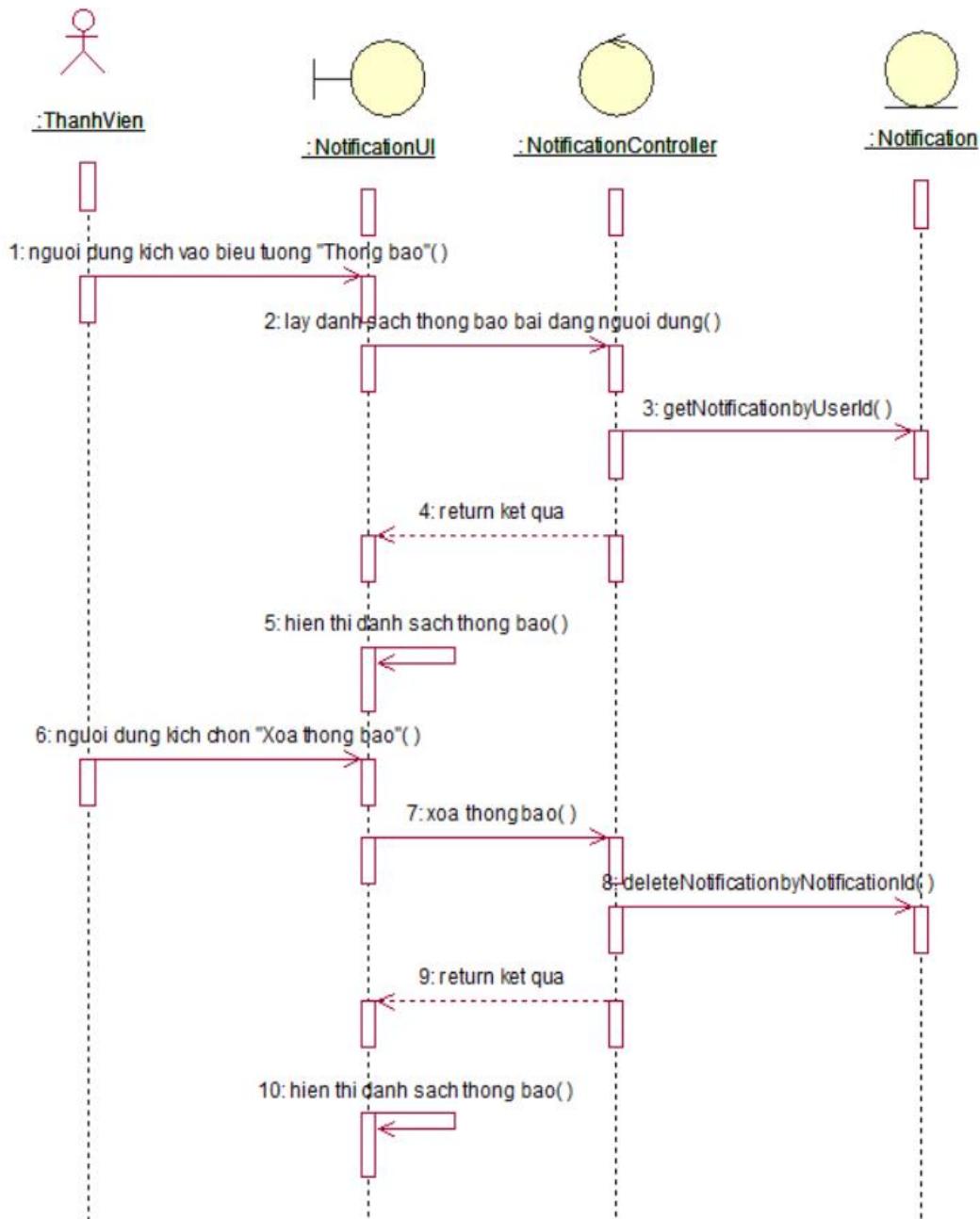
❖ Biểu đồ VOPC



Hình 3.3.12. Biểu đồ VOPC use case nhắn tin

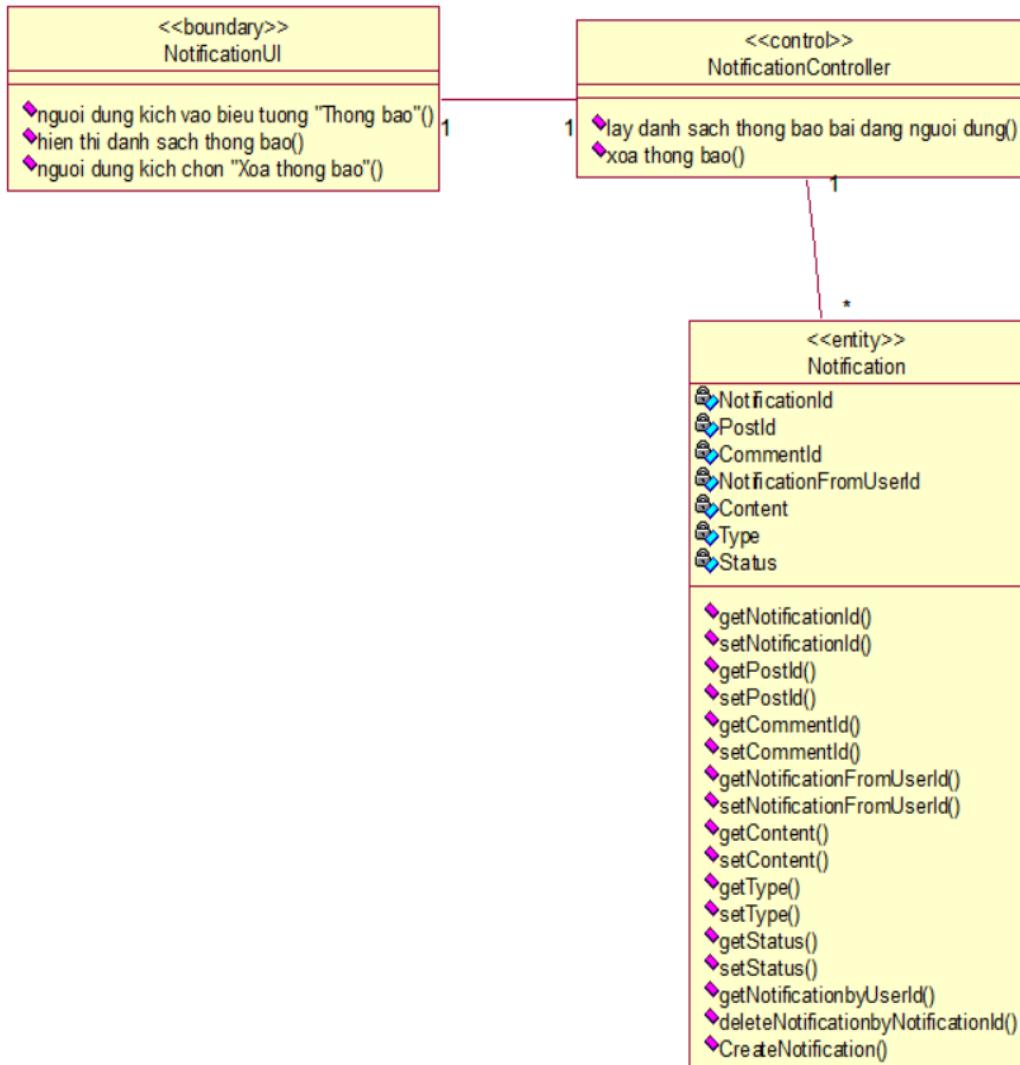
### 3.3.7. Use case quản lý thông báo

❖ Biểu đồ trình tự:



Hình 3.3.13. Biểu đồ trình tự use case quản lý thông báo

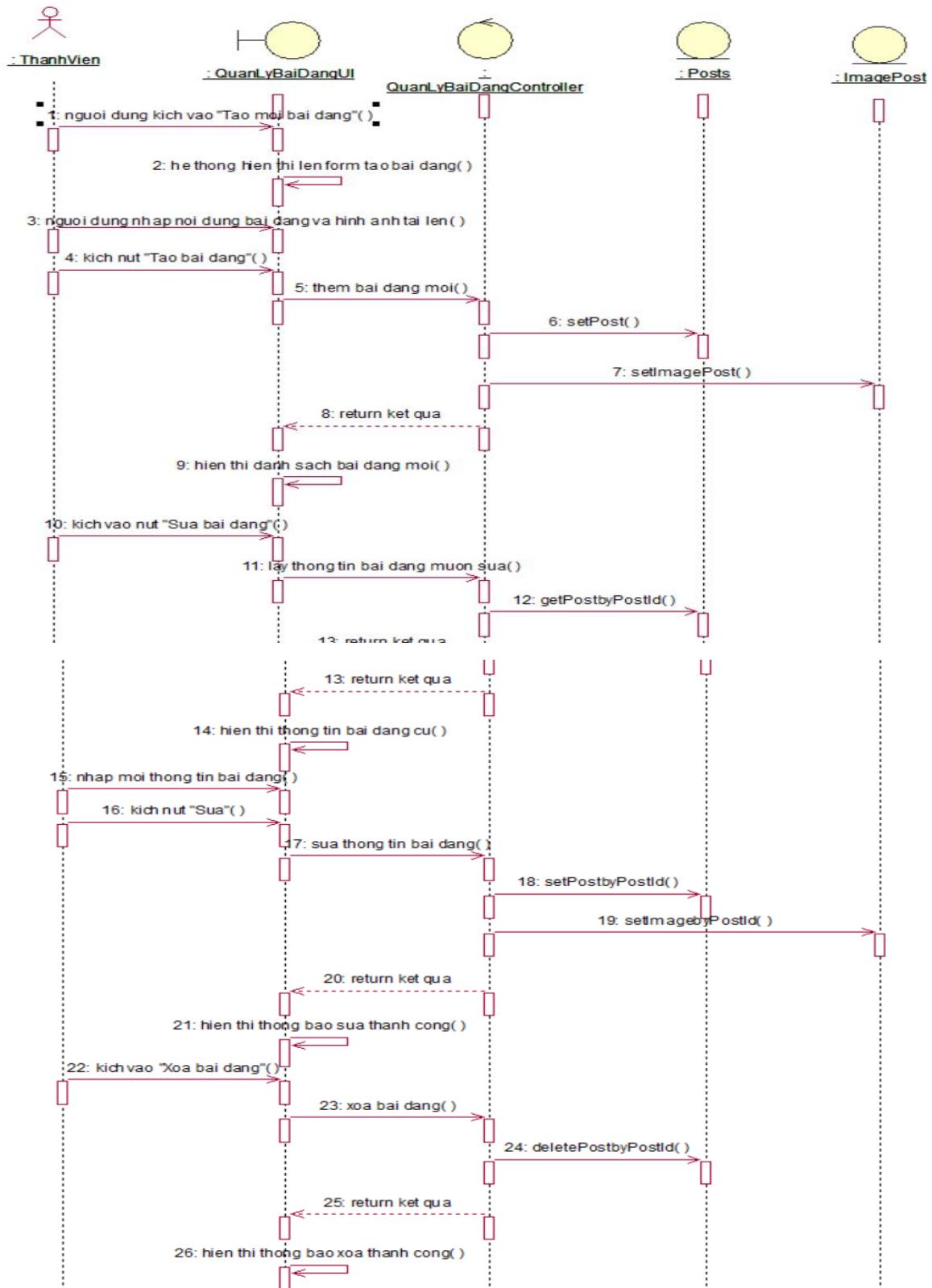
❖ Biểu đồ VOPC



Hình 3.3.14. Biểu đồ VOPC quản lý thông báo

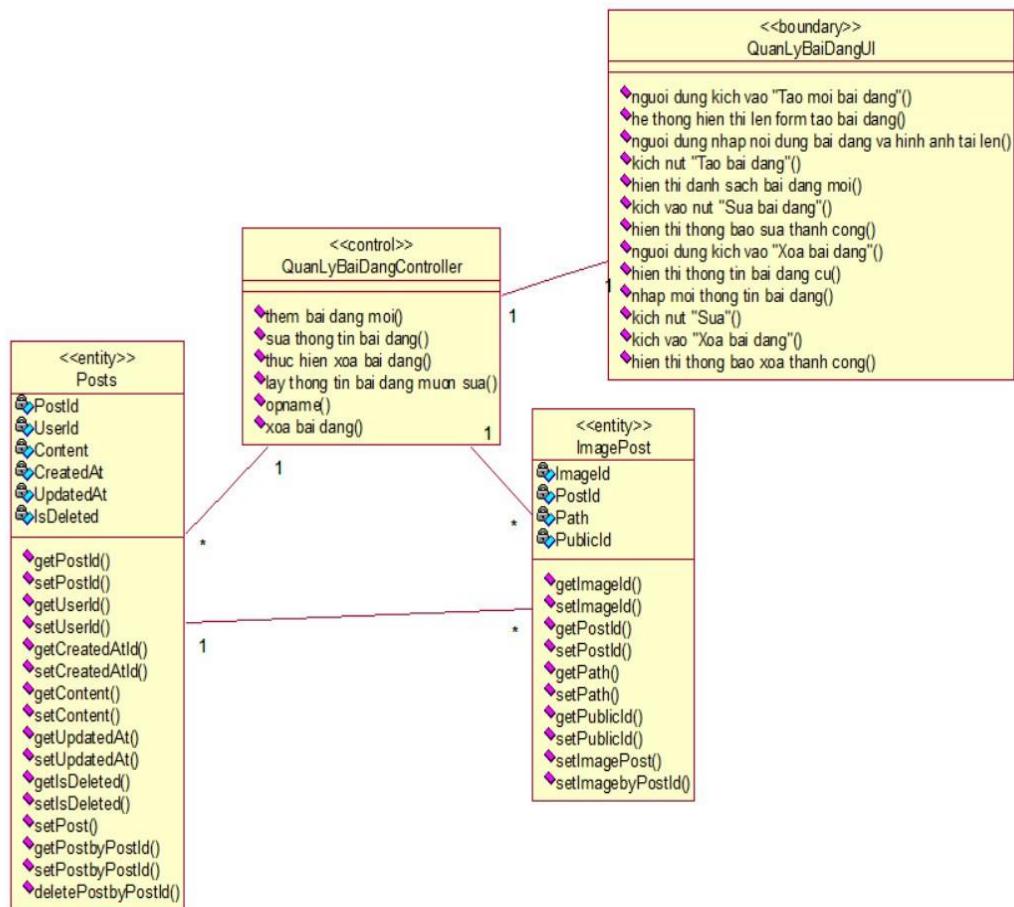
### 3.3.8. Use case quản lý bài đăng cá nhân

❖ Biểu đồ trình tự



Hình 3.3.15. Biểu đồ trình tự use case quản lý bài đăng cá nhân

❖ Biểu đồ VOPC

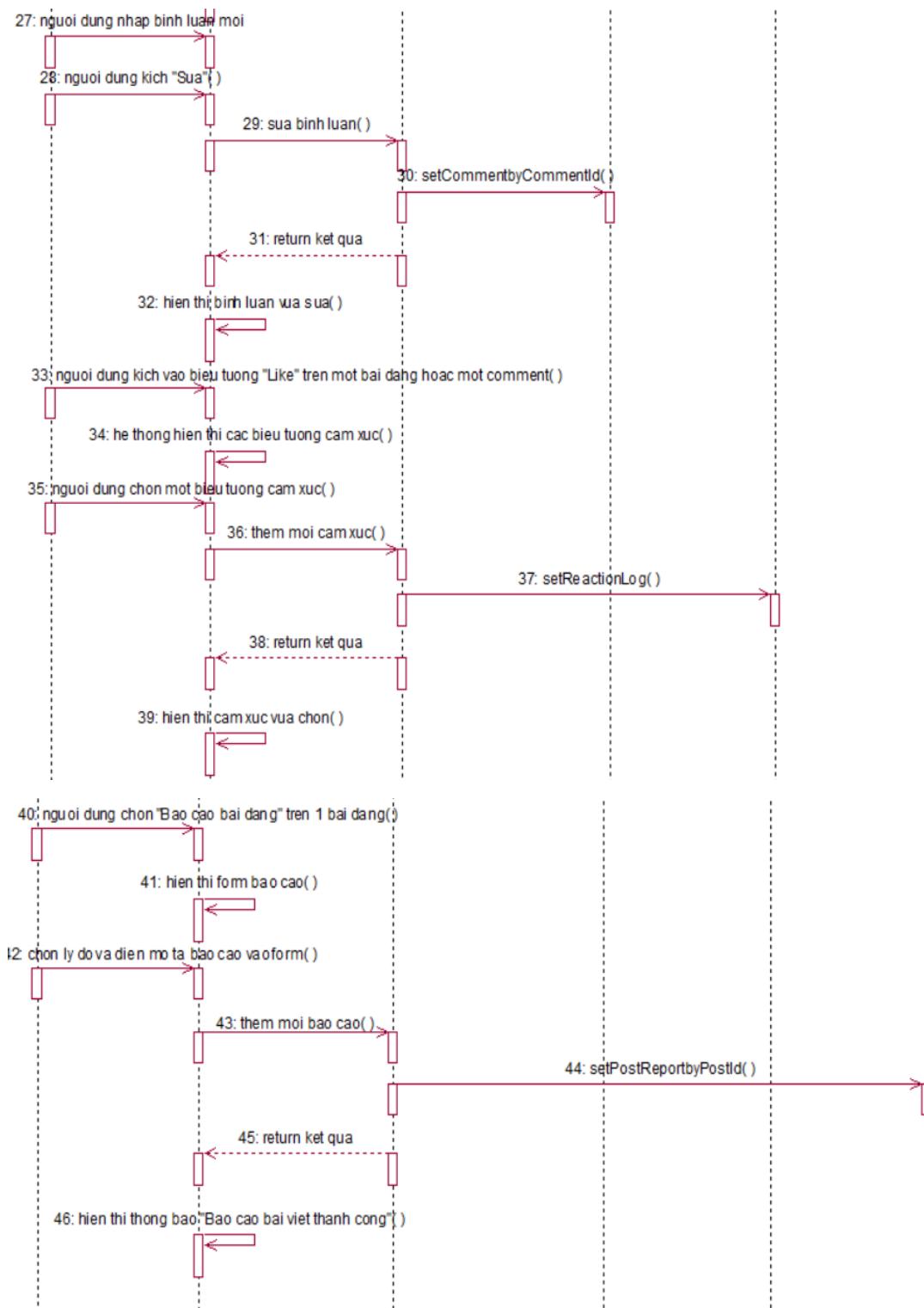


Hình 3.3.16. Biểu đồ VOPC use case quản lý bài đăng cá nhân

### 3.3.9. Use case tương tác bài đăng

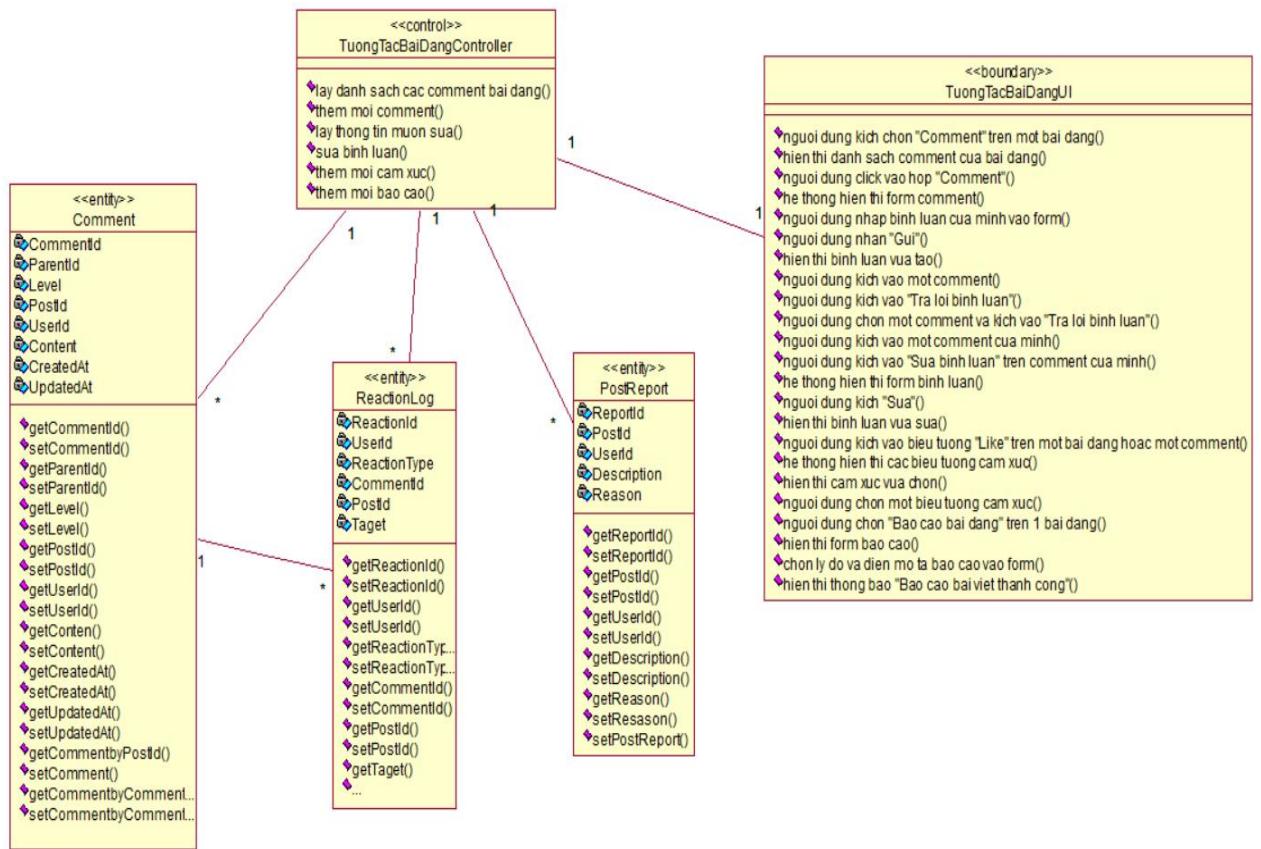
#### ❖ Biểu đồ trình tự





Hình 3.3.17. Biểu đồ tương tác bài đăng

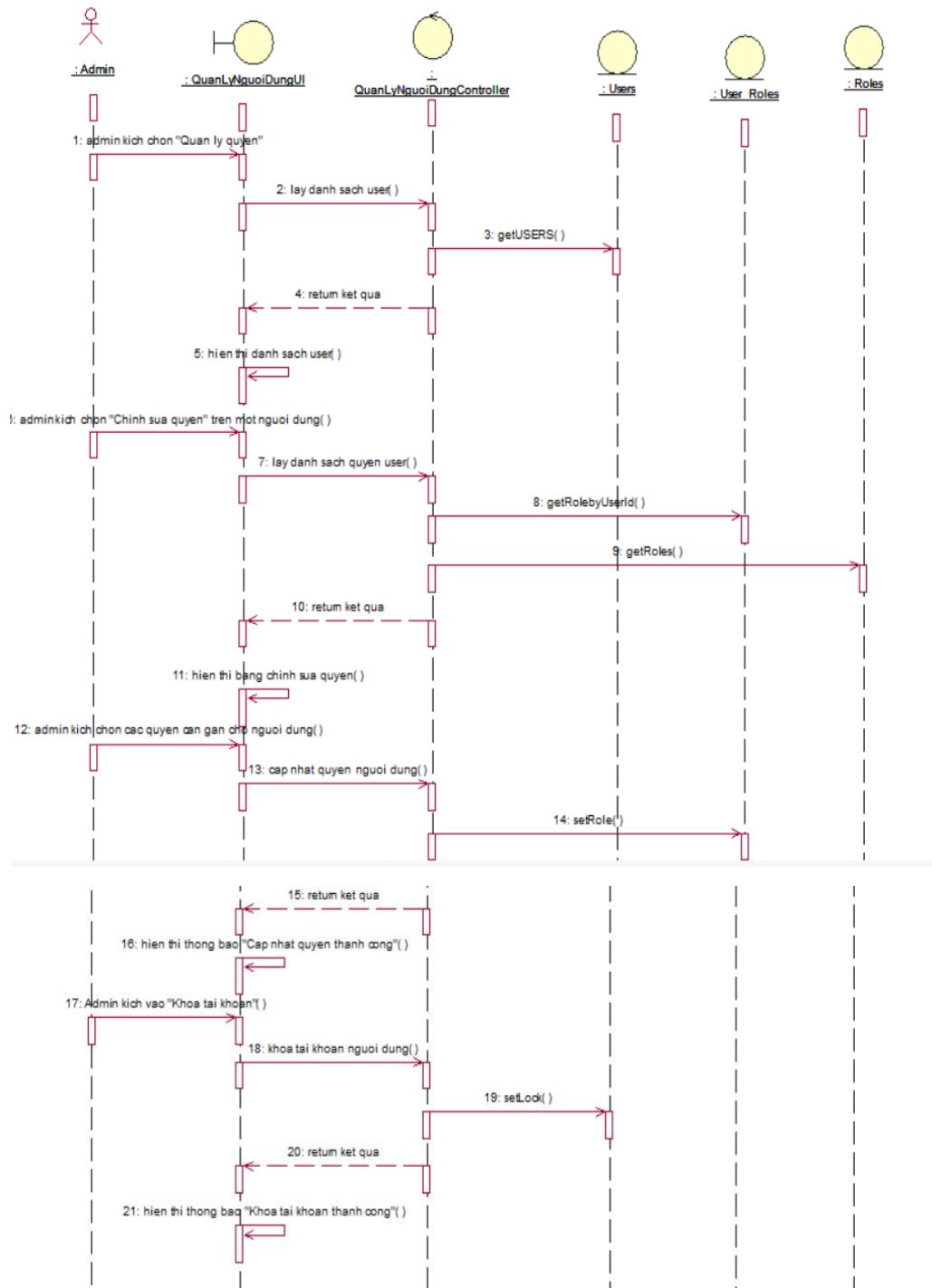
## ❖ Biểu đồ VOPC



Hình 3.3.18. Biểu đồ VOPC use case tương tác bài đăng

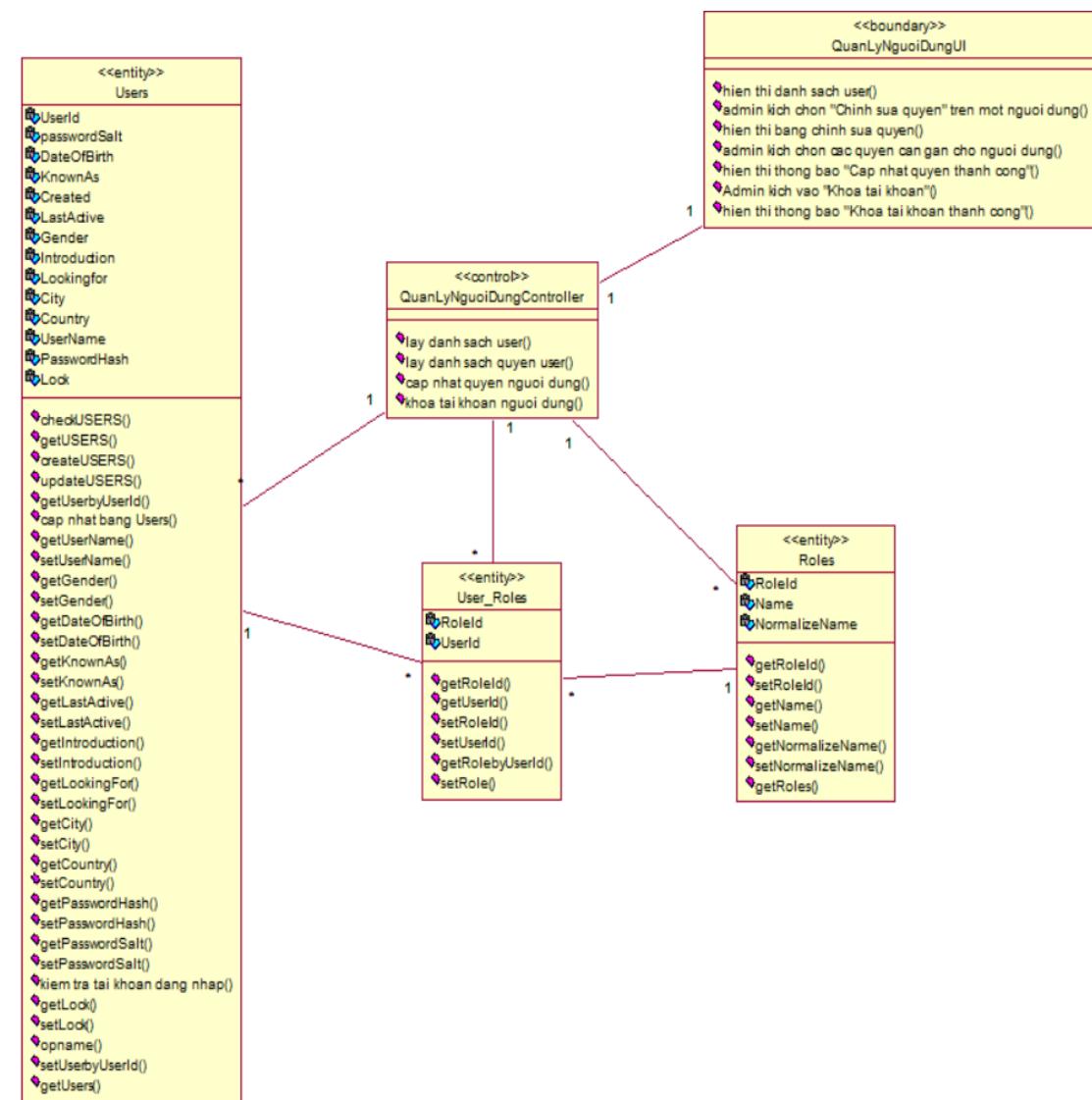
### 3.3.10. Use case quản lý người dùng

❖ Biểu đồ trình tự



Hình 3.3.19. Biểu đồ trình tự use case quản lý người dùng

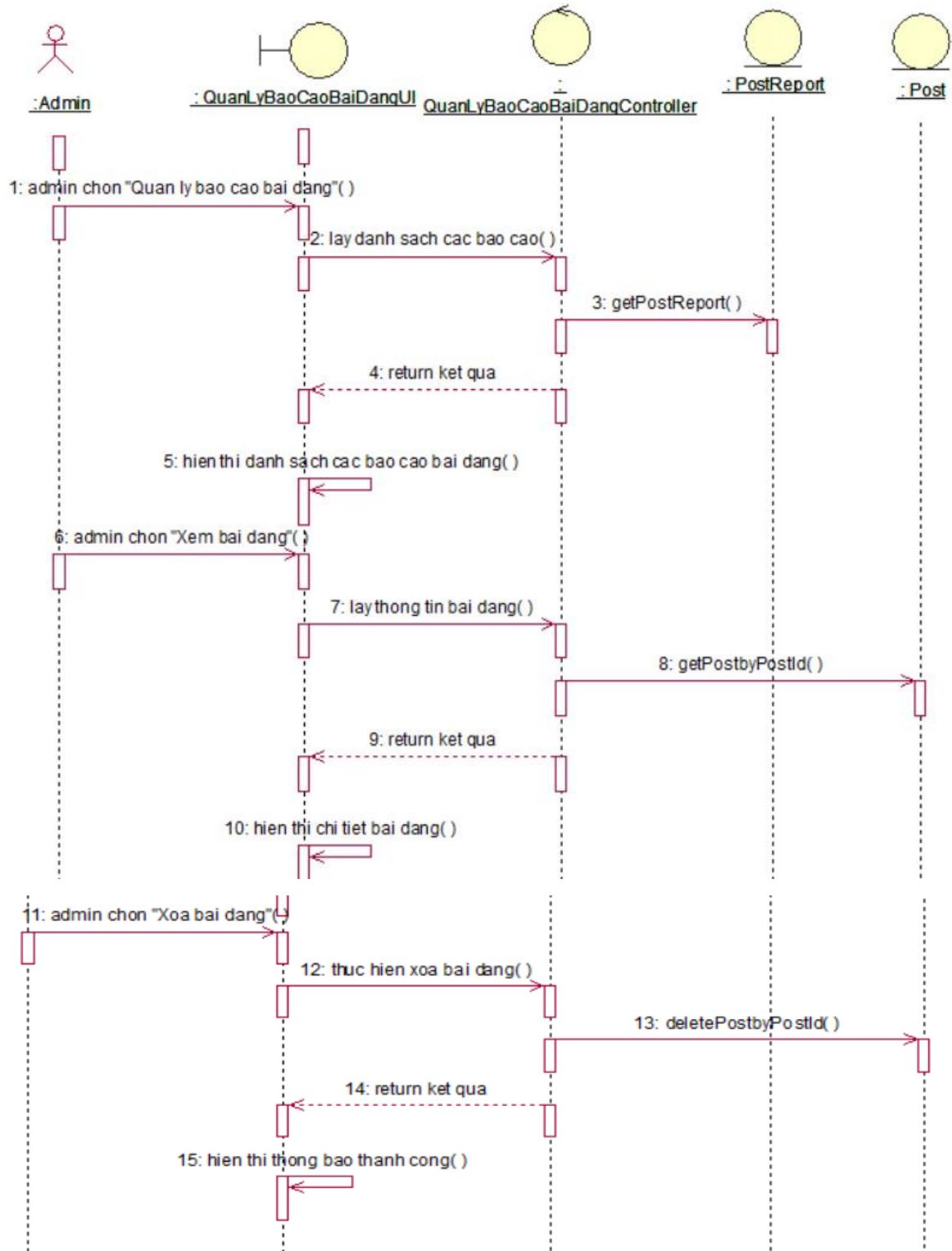
❖ Biểu đồ VOPC



Hình 3.3.20. Biểu đồ VOPC use case quản lý người dùng

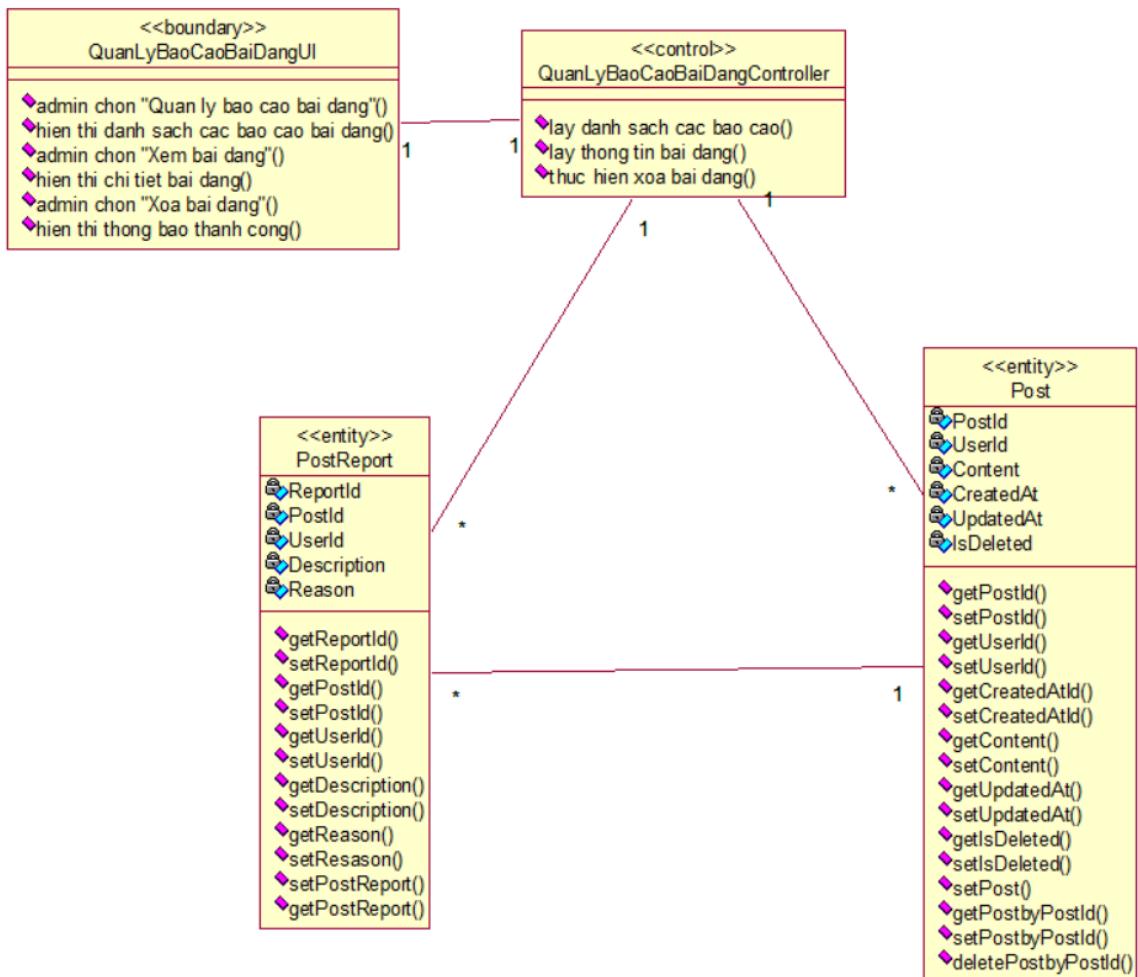
### 3.3.11. Use case quản lý báo cáo bài đăng

❖ Biểu đồ trình tự:



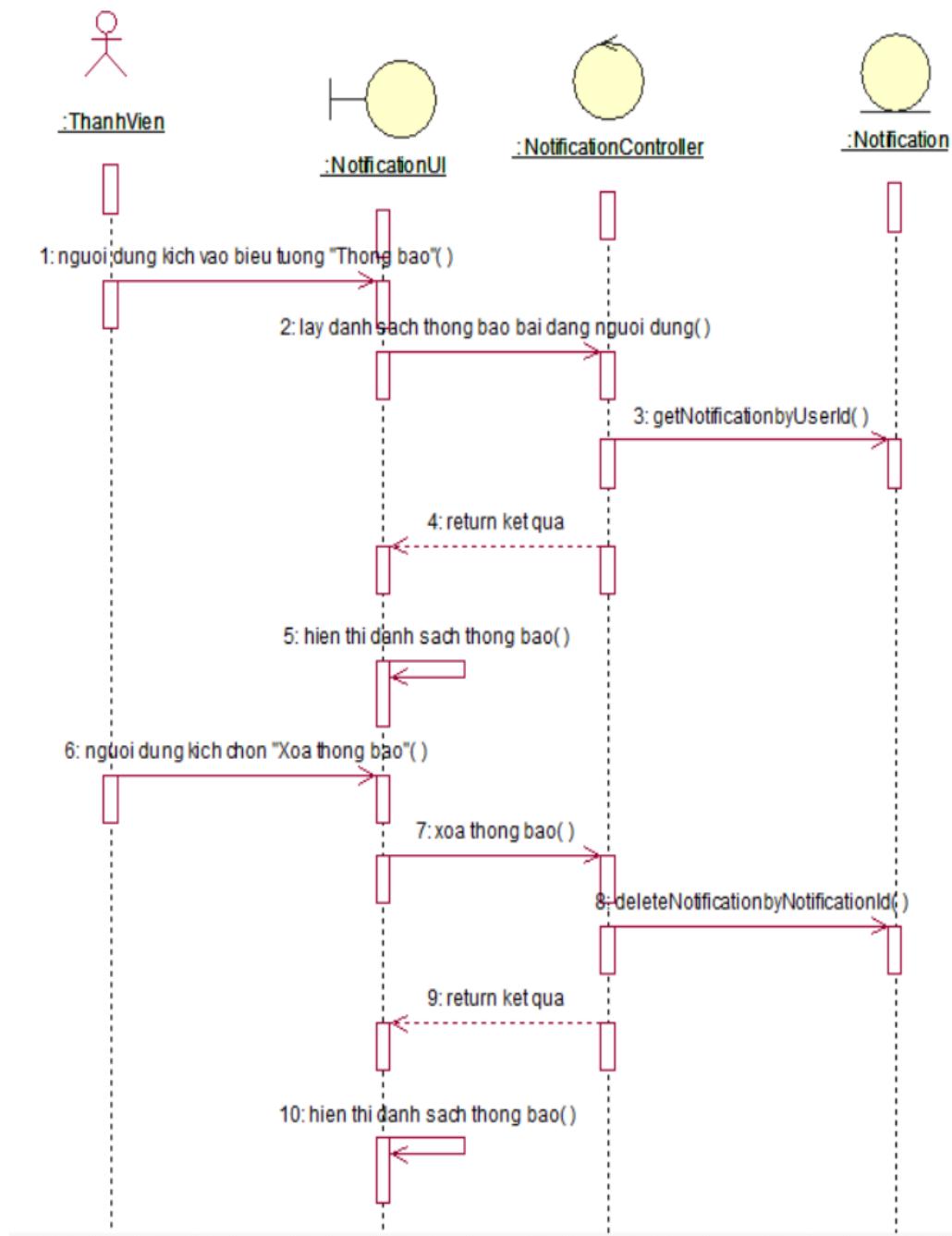
Hình 3.3.21. Biểu đồ trình tự use case quản lý báo cáo bài đăng

❖ Biểu đồ VOPC

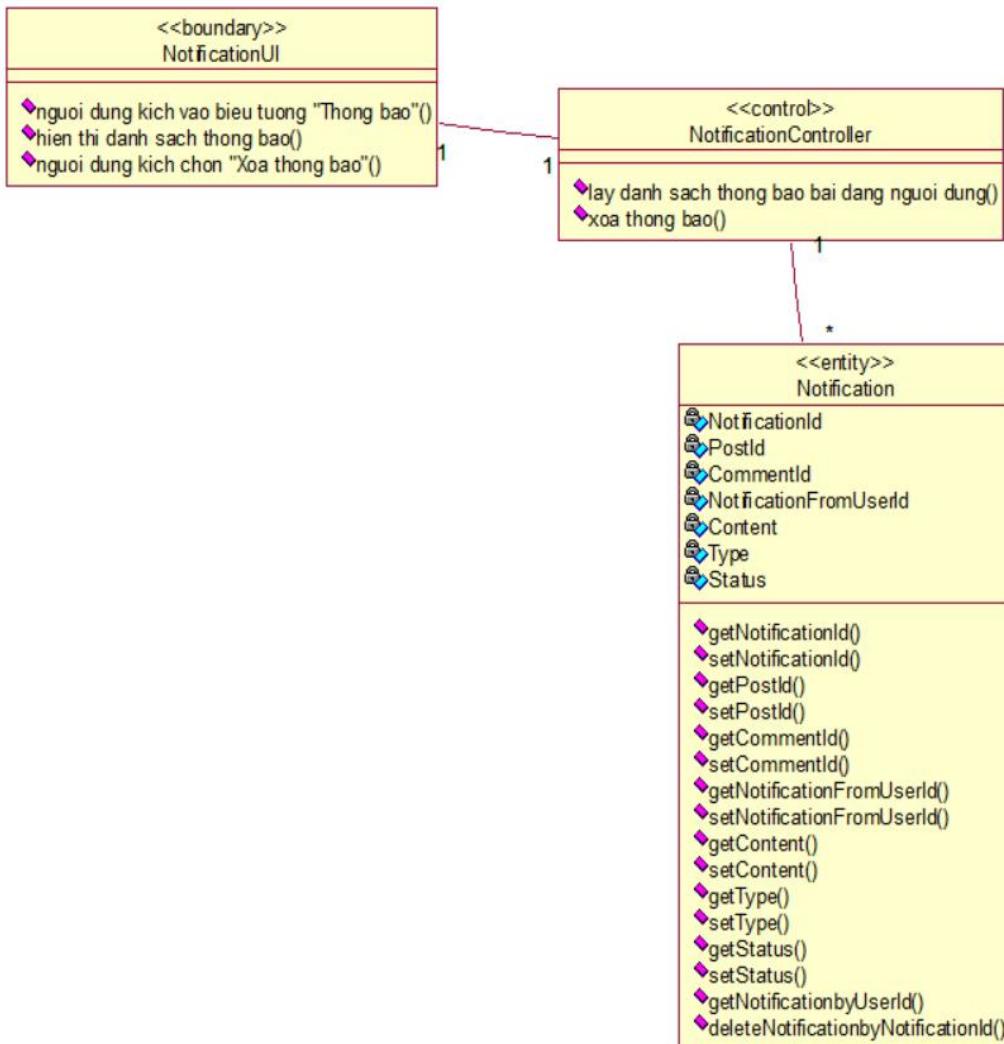


Hình 3.3.22. Biểu đồ VOPC use case quản lý báo cáo bài đăng

### 3.3.12. Use case quản lý thông báo



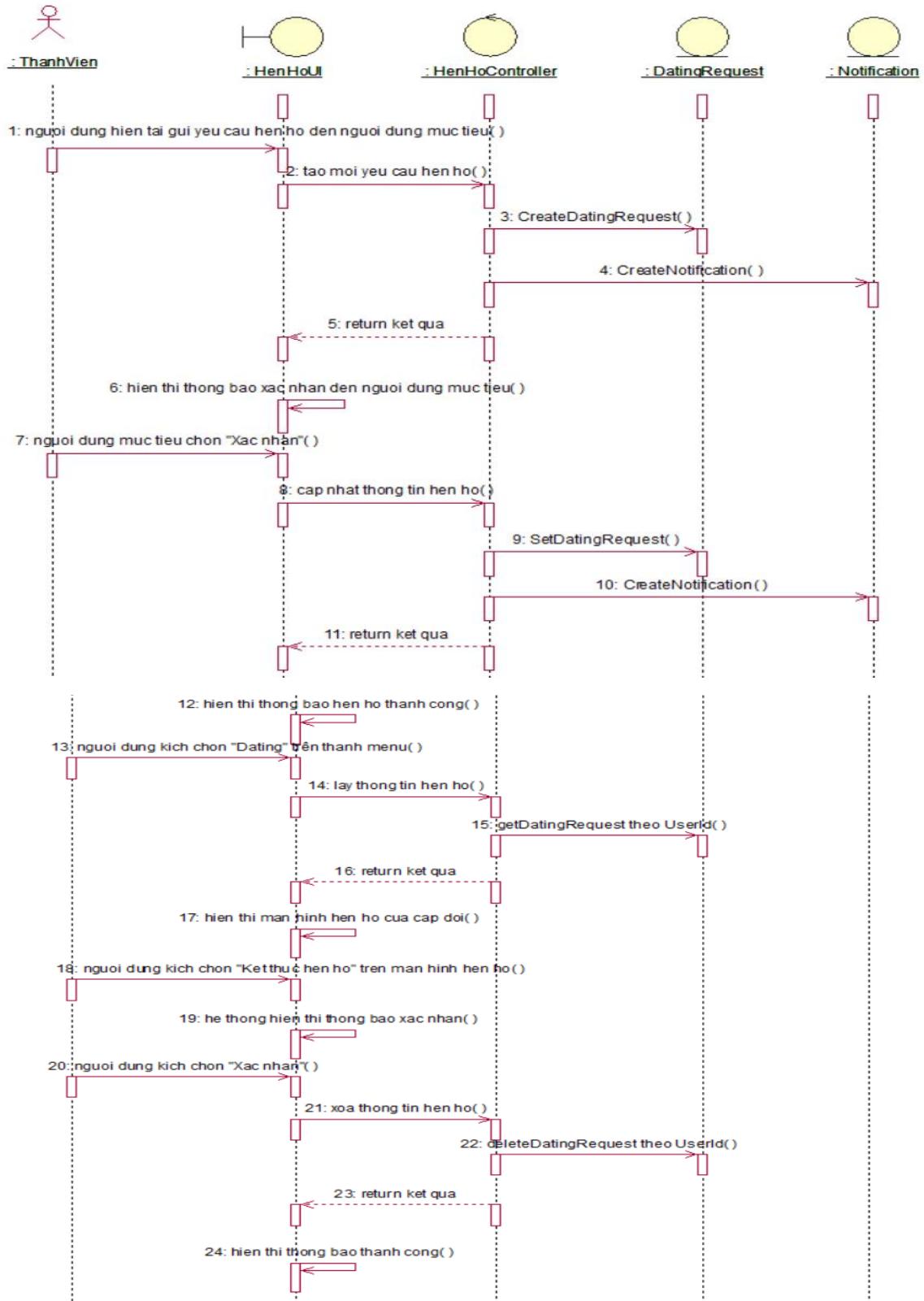
Hình 3.3.23. Biểu đồ trình tự use case quản lý thông báo



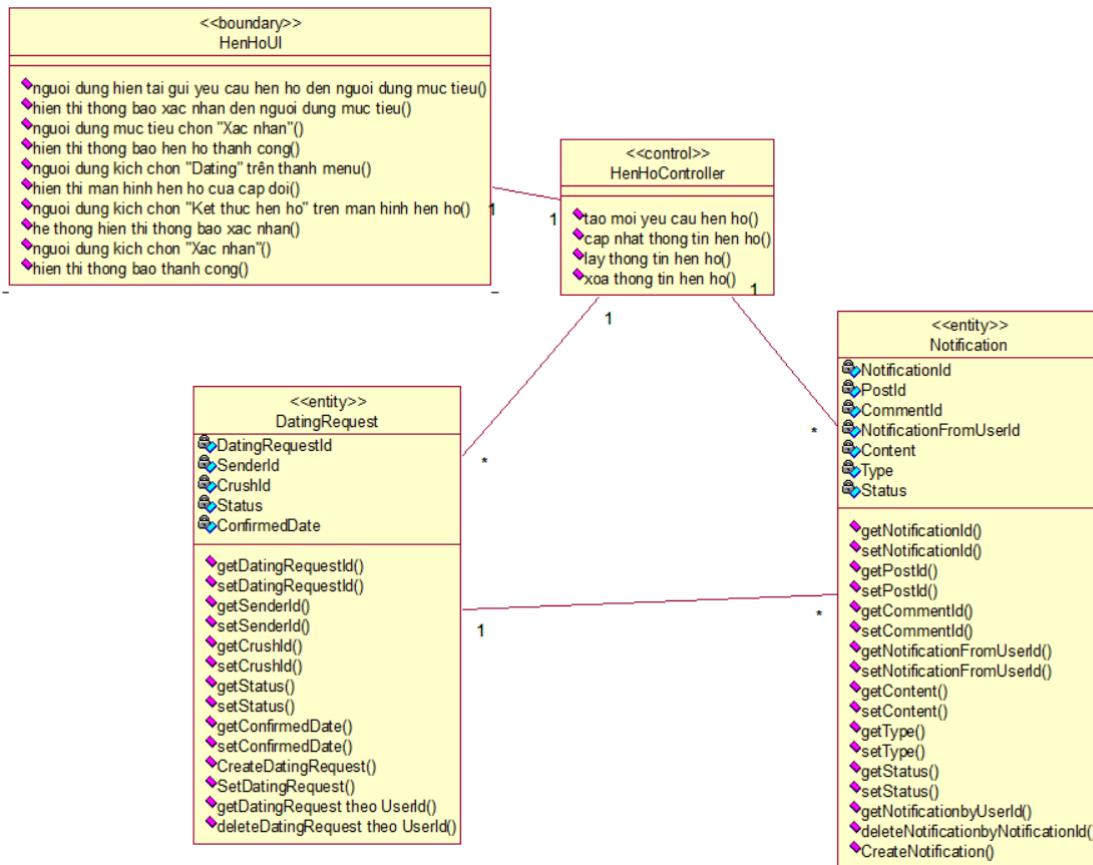
Hình 3.3.24. Biểu đồ VOPC use case quản lý thông báo

### 3.3.12. Use case hẹn hò

❖ Biểu đồ trình tự

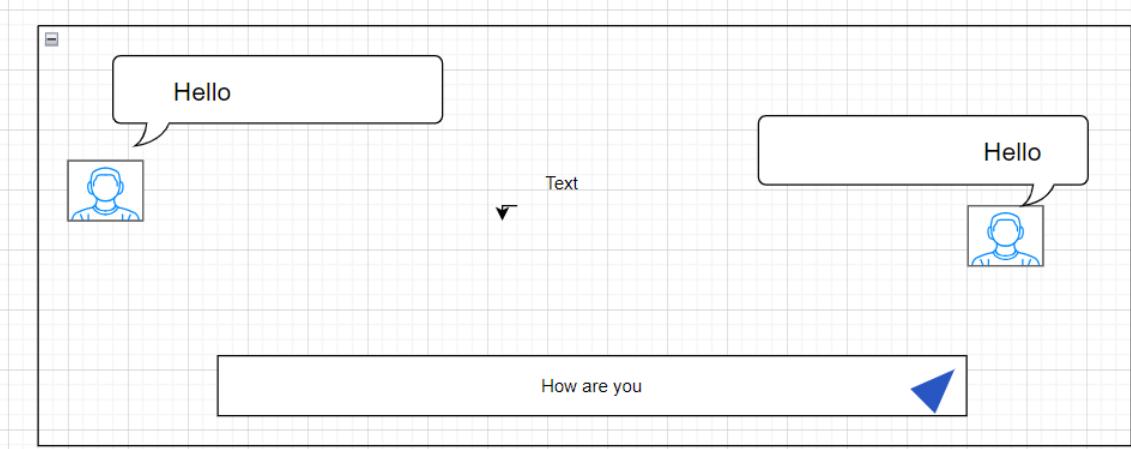


Hình 3.3.25. Biểu đồ trình tự use case hẹn hò

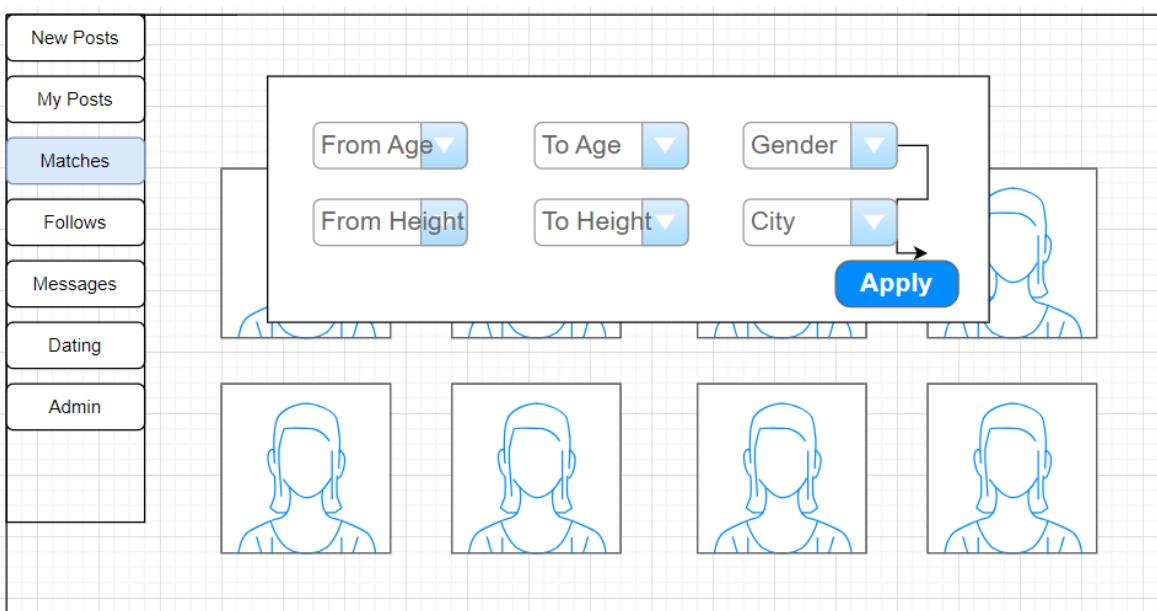


Hình 3.3.26. Biểu đồ VOPC use case hẹn hò

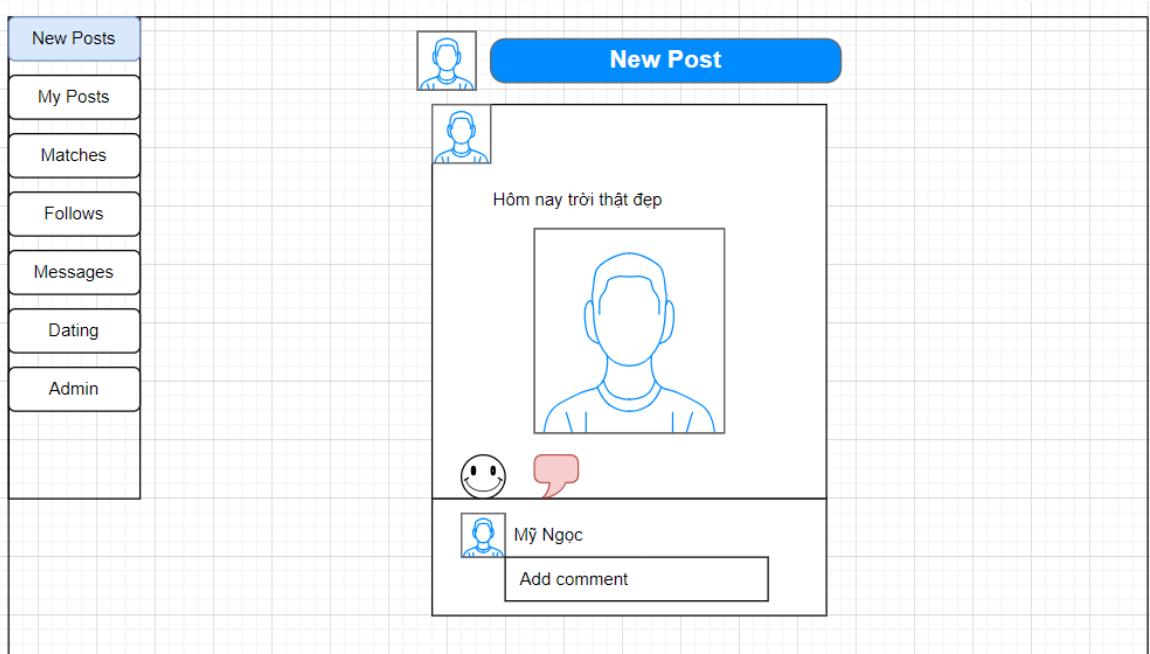
### 3.4. Mô tả giao diện



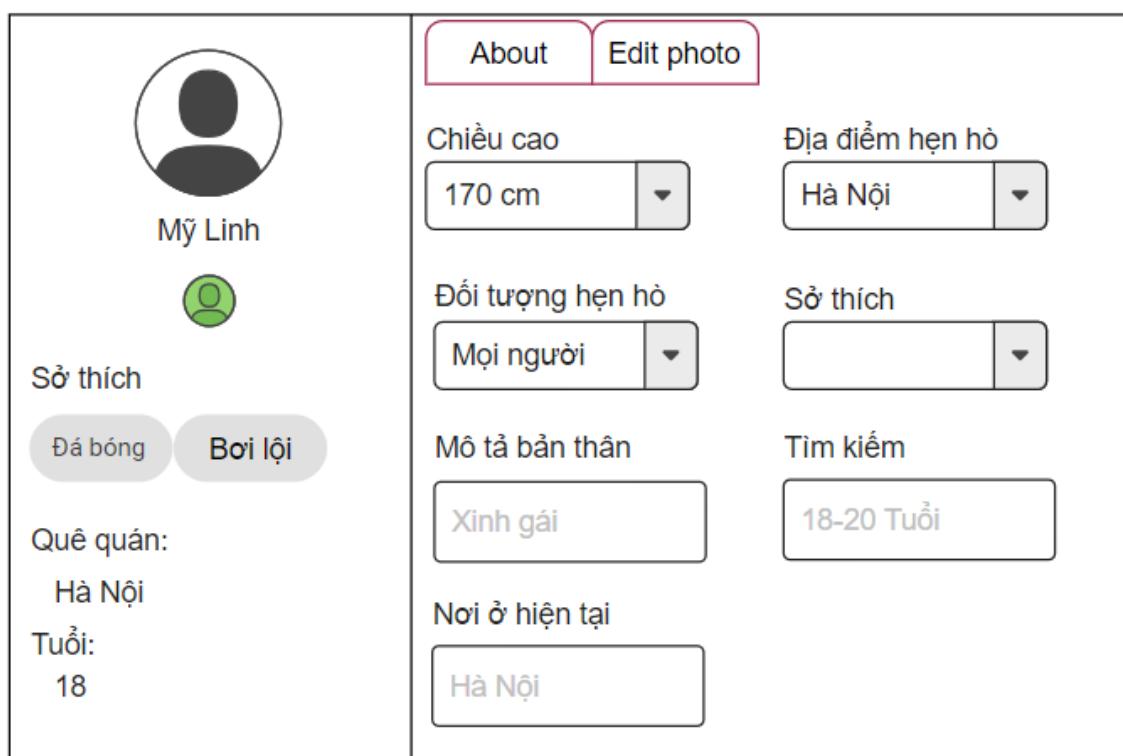
Hình 3.4.1. Mô tả giao diện tin nhắn



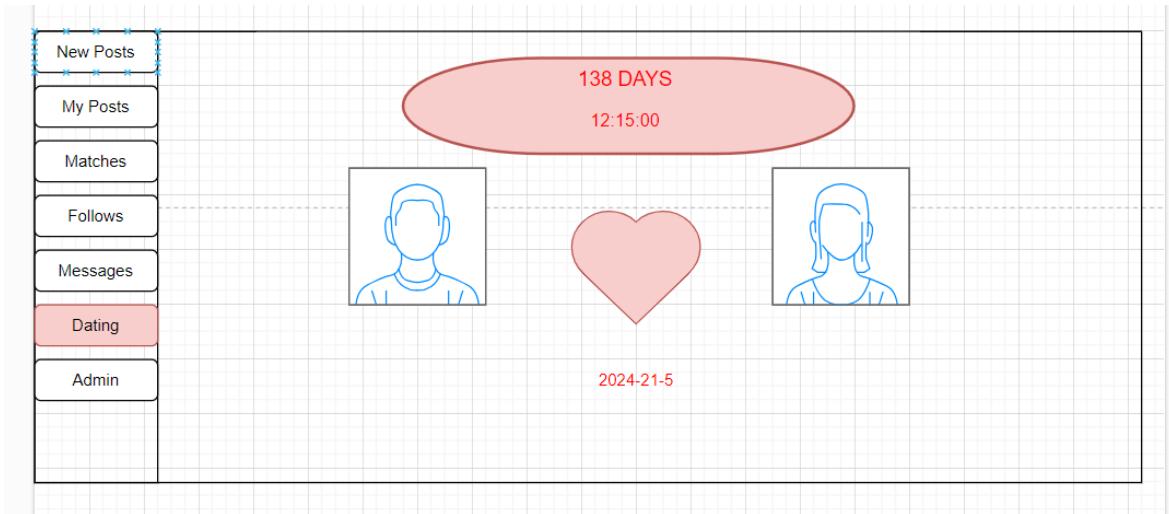
Hình 3.4.2. Mô tả giao diện màn hình chính



Hình 3.4.3. Mô tả giao diện màn hình bài đăng



Hình 3.4.4. Mô tả giao diện màn hình trang cá nhân



Hình 3.4.5. Mô tả giao diện màn hình trang hẹn hò

### 3.5. Thiết kế cơ sở dữ liệu

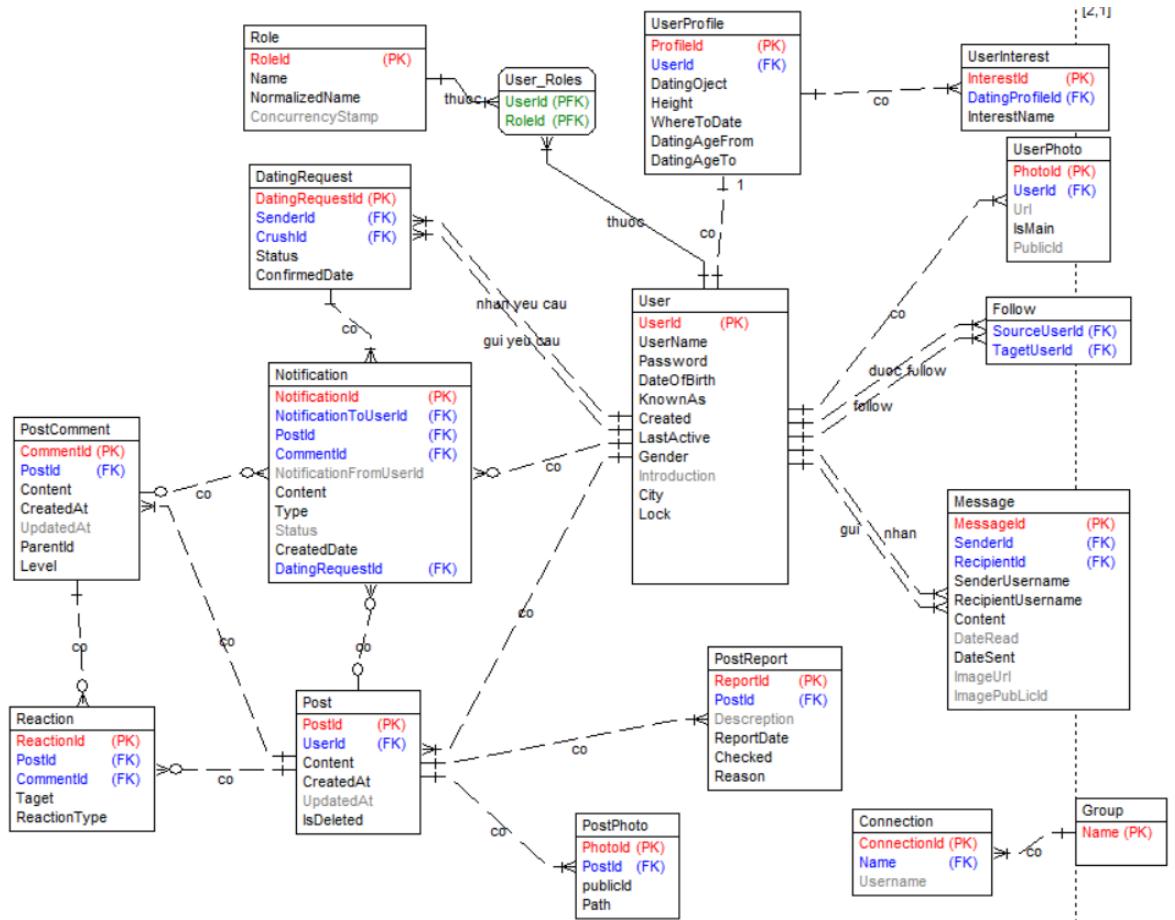
#### 3.5.1. Sơ đồ thực thể liên kết

##### 3.5.1.1. Mô tả dữ liệu

- + **Bảng User:** Bảng lưu trữ các thông tin quan trọng của người dùng như UserName, KnownAs, Password, Gender, Account status, DateOfBirth.
- + **Bảng UserProfile:** Lưu trữ thông tin về Height, Weight, DatingObject, WhereToDate, DatingAgeFrom, DatingAgeTo, Interests của User. Mỗi quan hệ giữa bảng User và UserProfile là mối quan hệ một nhiều, mỗi quan hệ giữa bảng UserProfile và UserInterest là mối quan hệ một nhiều.
- + **Bảng UserPhoto:** Lưu trữ các ảnh cá nhân của người dùng. Một người dùng thì có một hoặc nhiều hình ảnh và một hình ảnh chỉ thuộc về duy nhất 1 người dùng.
- + **Bảng Message:** MessageId, SenderId, RecipientId, RecipientUserName, SenderUserName, Content, DateRead, MessageSent, ImagePath, PublicId. Lưu trữ các tin nhắn của người dùng gửi đi và nhận vào. Mỗi quan hệ giữa người dùng gửi tin nhắn và nhận tin nhắn là mối quan hệ phản thân. Một người có thể gửi tin nhắn cho nhiều người và người đó có thể nhận tin nhắn từ nhiều người.
- + **Bảng Group:** Name. Chứa tên group nhóm chát của người dùng.
- + **Bảng Connection:** ConnectionId, Name, UserName. Chứa tên những người dùng kết nối. Khi 2 người đang ở trong một phòng chát chuỗi kết nối và tên người dùng sẽ được lưu lại trong Group và có bất kỳ sự thay đổi nào trạng thái tin nhắn sẽ tự cập nhật. Quan hệ giữa Group và Connection là quan hệ một nhiều
- + **Bảng Follow:** SourceUserId, TagetUserId. Lưu trữ các lượt follow của người follow và người được follow. Mỗi quan hệ giữa người follow và người được follow là quan hệ phản thân. Một người có thể follow nhiều người và người đó có thể được nhiều người khác follow.

- + Bảng Role: RoleId, Name, NormalizedName. Lưu trữ các quyền mà người dùng có thể được phân. Mỗi quan hệ giữa User và Role là quan hệ nhiều-nhiều. Mỗi người dùng có thể có 1 hoặc nhiều quyền và một quyền có thể được gán cho nhiều người.
- + Bảng Post: PostId, UserId, Content, CreatedAt, UpdatedAt. Post lưu trữ các bài đăng mà người dùng đăng. Một người thì có thể có một hoặc nhiều bài đăng và một bài đăng chỉ thuộc về một người.
- + Bảng PostPhoto: PhotoId, PostId, PublicId, Path. Lưu trữ các hình ảnh của bài đăng. Một bài đăng thì có một hoặc nhiều ảnh, một ảnh thì chỉ thuộc một bài đăng.
- + Bảng Comment: CommentId, PostId, Content, CreatedAt, UpdatedAt, ParentId, Level, UserId. Comment lưu trữ các comment của bài đăng. Quan hệ giữa bảng Post và bảng Comment là quan hệ một-nhiều.
- + Bảng Reaction: ReactionId, PostId, CommentId, UserId, Taget, ReactionType. Reaction lưu trữ các cảm xúc của người dùng của bài đăng và bình luận. Quan hệ giữa bảng Post và bảng Reaction là quan hệ một-nhiều, quan hệ giữa bảng Comment và bảng Reaction là quan hệ một-nhiều.
- + Bảng PostReport: ReportId, PostId, Description, ReportDate, Reason. Report lưu các báo cáo của người dùng về bài đăng. Một Post thì có một hoặc nhiều Report, một Report chỉ thuộc về 1 bài đăng.
- + Bảng Notification: NotificationId, NotificationToUserId, PostId, CommentId, NotificationFromUserId, Content, Type, Status, CreatedDate. Chứa các thông báo của người dùng. User quan hệ 1-n với Notification, Post quan hệ 1-n với Notification, Comment quan hệ 1-n với Notification. Bảng DatingRequest quan hệ 1-n với Notification
- + Bảng DatingRequest: DatingRequestId, SenderId, CrushId, Status, ConfirmedDate. Chứa thông tin về mối quan hệ hẹn hò của người dùng hiện tại và người dùng mục tiêu. Mỗi quan hệ giữa DatingRequest và

User là quan hệ phản thân một người có thể gửi yêu cầu hẹn hò đến nhiều người và cũng có thể nhận yêu cầu hẹn hò từ nhiều người



Hình 3.5.1. Mô hình thực thể liên kết

### 3.5.2. Thiết kế bảng

- Bảng User lưu thông tin người dùng

| Entity Name  |              | Table Name   |               |          |        |             |  |  |  |  |  |
|--|--------------|--------------|---------------|----------|--------|-------------|--|--|--|--|--|
|  |              | User         |               |          |        |             |  |  |  |  |  |
| Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL |              |              |               |          |        |             |  |  |  |  |  |
| Key  | Name         | Column Name  | Datatype      | Not null | Unique | Description |  |  |  |  |  |
| 1  | UserId       | UserId       | Integer       | Not null |        |             |  |  |  |  |  |
| 2  | UserName     | UserName     | NVarChar(255) | Not null |        |             |  |  |  |  |  |
| 3  | Password     | Password     | VarBinary(30) | Not null |        |             |  |  |  |  |  |
| 4  | DateOfBirth  | DateOfBirth  | DateTime      | Not null |        |             |  |  |  |  |  |
| 5  | KnownAs      | KnownAs      | NVarChar(200) | Not null |        |             |  |  |  |  |  |
| 6  | Created      | Created      | DateTime      | Not null |        |             |  |  |  |  |  |
| 7  | LastActive   | LastActive   | DateTime      | Not null |        |             |  |  |  |  |  |
| 8  | Gender       | Gender       | NVarChar(50)  | Not null |        |             |  |  |  |  |  |
| 9  | Introduction | Introduction | NVarChar(50)  |          |        |             |  |  |  |  |  |
| 10   | City         | City         | NVarChar(200) | Not null |        |             |  |  |  |  |  |
| 11   | Lock         | Lock         | Bit           | Not null |        |             |  |  |  |  |  |

Bảng 3.5.1. Bảng thông tin User

- Bảng Role lưu các quyền truy cập hệ thống

| Entity Name  |                  | Table Name       |               |          |        |             |  |  |  |  |  |
|--|------------------|------------------|---------------|----------|--------|-------------|--|--|--|--|--|
|  |                  | Role             |               |          |        |             |  |  |  |  |  |
| Attributes Properties Advanced Indexes Alternate Keys Others Relationship Storage Comments Notes DDL |                  |                  |               |          |        |             |  |  |  |  |  |
| Key  | Name             | Column Name      | Datatype      | Not null | Unique | Description |  |  |  |  |  |
| 1  | RoleId           | RoleId           | Integer       | Not null |        |             |  |  |  |  |  |
| 2  | Name             | Name             | NVarChar(200) | Not null |        |             |  |  |  |  |  |
| 3  | NormalizedName   | NormalizedName   | NVarChar(200) | Not null |        |             |  |  |  |  |  |
| 4  | ConcurrencyStamp | ConcurrencyStamp | NVarChar(200) |          |        |             |  |  |  |  |  |

Bảng 3.5.2. Bảng Role

- Bảng Role với User lưu các quyền của người dùng

| Entity Name  |            | Table Name   |          |          |        |             |
|--|------------|--------------|----------|----------|--------|-------------|
|  |            | > User_Roles |          |          |        |             |
|  |            |              |          |          |        |             |
| <a href="#">Attributes</a> <a href="#">Properties</a> <a href="#">Advanced</a> <a href="#">Indexes</a> <a href="#">Alternate Keys</a> <a href="#">Others</a> <a href="#">Relationship</a> <a href="#">Storage</a> <a href="#">Comments</a> <a href="#">Notes</a> <a href="#">DDL</a> |            |              |          |          |        |             |
| Key  | Name       | Column Name  | Datatype | Not null | Unique | Description |
| 1  | UserRoleId | UserId       | Integer  | Not null |        |             |
| 2  | RoleId     | RoleId       | Integer  | Not null |        |             |

Bảng 3.5.3. Bảng Role với User

- Bảng UserPhoto lưu trữ ảnh người dùng

| Entity Name  |          | Table Name  |               |          |        |             |
|--|----------|-------------|---------------|----------|--------|-------------|
|  |          | > UserPhoto |               |          |        |             |
|  |          |             |               |          |        |             |
| <a href="#">Attributes</a> <a href="#">Properties</a> <a href="#">Advanced</a> <a href="#">Indexes</a> <a href="#">Alternate Keys</a> <a href="#">Others</a> <a href="#">Relationship</a> <a href="#">Storage</a> <a href="#">Comments</a> <a href="#">Notes</a> <a href="#">DDL</a> |          |             |               |          |        |             |
| Key  | Name     | Column Name | Datatype      | Not null | Unique | Description |
| 1  | PhotoId  | PhotoId     | Integer       | Not null |        |             |
| 2  | UserId   | UserId      | Integer       | Not null |        |             |
| 3  | Url      | Url         | NVarChar(300) |          |        |             |
| 4  | IsMain   | IsMain      | Bit           | Not null |        |             |
| 5  | PublicId | PublicId    | NVarChar(200) |          |        |             |

Bảng 3.5.4. Bảng UserPhoto

- Bảng UserProfile lưu trữ các thông tin người dùng

| Entity Name  |               | Table Name    |          |          |        |             |
|--|---------------|---------------|----------|----------|--------|-------------|
|  |               | > UserProfile |          |          |        |             |
|  |               |               |          |          |        |             |
| <a href="#">Attributes</a> <a href="#">Properties</a> <a href="#">Advanced</a> <a href="#">Indexes</a> <a href="#">Alternate Keys</a> <a href="#">Others</a> <a href="#">Relationship</a> <a href="#">Storage</a> <a href="#">Comments</a> <a href="#">Notes</a> <a href="#">DDL</a> |               |               |          |          |        |             |
| Key  | Name          | Column Name   | Datatype | Not null | Unique | Description |
| 1  | ProfileId     | ProfileId     | Integer  | Not null |        |             |
| 2  | UserId        | UserId        | Integer  | Not null |        |             |
| 3  | DatingObject  | DatingObject  | Integer  | Not null |        |             |
| 4  | Height        | Height        | Integer  | Not null |        |             |
| 5  | WhereToDate   | WhereToDate   | Integer  | Not null |        |             |
| 6  | DatingAgeFrom | DatingAgeFrom | Integer  | Not null |        |             |
| 7  | DatingAgeTo   | DatingAgeTo   | Integer  | Not null |        |             |

Bảng 3.5.5. Bảng UserProfile

- Bảng UserInterest lưu trữ sở thích người dùng

Entity

| Entity Name  | Table Name      |                 |          |          |        |             |
|--|-----------------|-----------------|----------|----------|--------|-------------|
| UserInterest   | UserInterest    |                 |          |          |        |             |
| <a href="#">Attributes</a> <a href="#">Properties</a> <a href="#">Advanced</a> <a href="#">Indexes</a> <a href="#">Alternate Keys</a> <a href="#">Others</a> <a href="#">Relationship</a> <a href="#">Storage</a> <a href="#">Comments</a> <a href="#">Notes</a> <a href="#">DDL</a> |                 |                 |          |          |        |             |
| Key  | Name            | Column Name     | Datatype | Not null | Unique | Description |
| 1  | InterestId      | InterestId      | Integer  | Not null |        |             |
| 2  | DatingProfileId | DatingProfileId | Integer  | Not null |        |             |
| 3  | InterestName    | InterestName    | Integer  | Not null |        |             |

Bảng 3.5.6. Bảng UserInterest

- Bảng Message lưu trữ tin nhắn người dùng

Entity

| Entity Name  | Table Name        |                   |               |          |        |             |
|--|-------------------|-------------------|---------------|----------|--------|-------------|
| Message  | Message           |                   |               |          |        |             |
| <a href="#">Attributes</a> <a href="#">Properties</a> <a href="#">Advanced</a> <a href="#">Indexes</a> <a href="#">Alternate Keys</a> <a href="#">Others</a> <a href="#">Relationship</a> <a href="#">Storage</a> <a href="#">Comments</a> <a href="#">Notes</a> <a href="#">DDL</a> |                   |                   |               |          |        |             |
| Key  | Name              | Column Name       | Datatype      | Not null | Unique | Description |
| 1  | MessageId         | MessageId         | Integer       | Not null |        |             |
| 2  | SenderId          | SenderId          | Integer       | Not null |        |             |
| 3  | RecipientId       | RecipientId       | Integer       | Not null |        |             |
| 4  | SenderUsername    | SenderUsername    | NVarChar(50)  | Not null |        |             |
| 5  | RecipientUsername | RecipientUsername | NVarChar(50)  | Not null |        |             |
| 6  | Content           | Content           | NVarChar(200) | Not null |        |             |
| 7  | DateRead          | DateRead          | DateTime      |          |        |             |
| 8  | DateSent          | DateSent          | DateTime      | Not null |        |             |
| 9  | ImageUrl          | ImageUrl          | NVarChar(100) |          |        |             |
| 10   | ImagePublId       | ImagePublId       | NVarChar(100) |          |        |             |

Bảng 3.5.7. Bảng Message

- Bảng Post lưu trữ các bài đăng người dùng

Entity

| Entity Name  | Table Name |             |               |          |        |             |
|--|------------|-------------|---------------|----------|--------|-------------|
| Post   | Post       |             |               |          |        |             |
| <a href="#">Attributes</a> <a href="#">Properties</a> <a href="#">Advanced</a> <a href="#">Indexes</a> <a href="#">Alternate Keys</a> <a href="#">Others</a> <a href="#">Relationship</a> <a href="#">Storage</a> <a href="#">Comments</a> <a href="#">Notes</a> <a href="#">DDL</a> |            |             |               |          |        |             |
| Key  | Name       | Column Name | Datatype      | Not null | Unique | Description |
| 1  | PostId     | PostId      | Integer       | Not null |        |             |
| 2  | UserId     | UserId      | Integer       | Not null |        |             |
| 3  | Content    | Content     | NVarChar(200) | Not null |        |             |
| 4  | CreatedAt  | CreatedAt   | DateTime      | Not null |        |             |
| 5  | UpdatedAt  | UpdatedAt   | DateTime      |          |        |             |
| 6  | IsDeleted  | IsDeleted   | Bit           | Not null |        |             |

Bảng 3.5.8. Bảng Post

- Bảng PostPhoto lưu trữ ảnh bài đăng

| Attributes |          | Properties  | Advanced | Indexes | Alternate Keys | Others | Relationship | Storage  | Comments | Notes       | DDL |
|------------|----------|-------------|----------|---------|----------------|--------|--------------|----------|----------|-------------|-----|
| Key        | Name     | Column Name |          |         | Datatype       |        |              | Not null | Unique   | Description |     |
| 1          | Photoid  | Photoid     |          |         | Integer        |        |              | Not null |          |             |     |
| 2          | PostId   | PostId      |          |         | Integer        |        |              | Not null |          |             |     |
| 3          | publicId | publicId    |          |         | NVarChar(200)  |        |              | Not null |          |             |     |
| 4          | Path     | Path        |          |         | NVarChar(200)  |        |              | Not null |          |             |     |

Bảng 3.5.9. Bảng PostPhoto

- Bảng Comment lưu trữ comment bài đăng

| Attributes |           | Properties  | Advanced | Indexes | Alternate Keys | Others | Relationship | Storage  | Comments | Notes       | DDL |
|------------|-----------|-------------|----------|---------|----------------|--------|--------------|----------|----------|-------------|-----|
| Key        | Name      | Column Name |          |         | Datatype       |        |              | Not null | Unique   | Description |     |
| 1          | CommentId | CommentId   |          |         | Integer        |        |              | Not null |          |             |     |
| 2          | PostId    | PostId      |          |         | Integer        |        |              | Not null |          |             |     |
| 3          | Content   | Content     |          |         | NVarChar(200)  |        |              | Not null |          |             |     |
| 4          | CreatedAt | CreatedAt   |          |         | DateTime       |        |              | Not null |          |             |     |
| 5          | UpdatedAt | UpdatedAt   |          |         | DateTime       |        |              | Not null |          |             |     |
| 6          | ParentId  | ParentId    |          |         | Integer        |        |              | Not null |          |             |     |
| 7          | Level     | Level       |          |         | Integer        |        |              | Not null |          |             |     |

Bảng 3.5.10. Bảng Comment

- Bảng Reaction lưu trữ các cảm xúc của bài đăng và comment

| Attributes |              | Properties   | Advanced | Indexes | Alternate Keys | Others | Relationship | Storage  | Comments | Notes       | DDL |
|------------|--------------|--------------|----------|---------|----------------|--------|--------------|----------|----------|-------------|-----|
| Key        | Name         | Column Name  |          |         | Datatype       |        |              | Not null | Unique   | Description |     |
| 1          | ReactionId   | ReactionId   |          |         | Char(1)        |        |              | Not null |          |             |     |
| 2          | PostId       | PostId       |          |         | Integer        |        |              |          |          |             |     |
| 3          | CommentId    | CommentId    |          |         | Integer        |        |              |          |          |             |     |
| 4          | Taget        | Taget        |          |         | Integer        |        |              | Not null |          |             |     |
| 5          | ReactionType | ReactionType |          |         | Integer        |        |              | Not null |          |             |     |

Bảng 3.5.11. Bảng Reaction

- Bảng PostReport lưu trữ các báo cáo bài đăng

The screenshot shows the Entity Designer interface with the 'Entity' tab selected. The 'Entity Name' is set to 'PostReport' and the 'Table Name' is also 'PostReport'. Below the table name, there is a grid of attributes:

|   | Key | Name        | Column Name | Datatype      | Not null | Unique | Description |
|---|-----|-------------|-------------|---------------|----------|--------|-------------|
| 1 | PK  | ReportId    | ReportId    | Integer       | Not null |        |             |
| 2 | FK  | PostId      | PostId      | Integer       | Not null |        |             |
| 3 |     | Descreption | Descreption | NVarChar(300) |          |        |             |
| 4 |     | ReportDate  | ReportDate  | DateTime      | Not null |        |             |
| 5 |     | Checked     | Checked     | Bit           | Not null |        |             |
| 6 |     | Reason      | Reason      | Bit           | Not null |        |             |

Bảng 3.5.12. Bảng PostReport

- Bảng Connection lưu trữ kết nối của người dùng

The screenshot shows the Entity Designer interface with the 'Entity' tab selected. The 'Entity Name' is set to 'Connection' and the 'Table Name' is also 'Connection'. Below the table name, there is a grid of attributes:

|   | Key | Name         | Column Name  | Datatype      | Not null | Unique | Description |
|---|-----|--------------|--------------|---------------|----------|--------|-------------|
| 1 | PK  | ConnectionId | ConnectionId | NVarChar(200) | Not null |        |             |
| 2 | FK  | Name         | Name         | NVarChar(200) | Not null |        |             |
| 3 |     | Username     | Username     | NVarChar(200) |          |        |             |

Bảng 3.5.13. Bảng Connection

- Bảng Notification lưu trữ các thông báo về bài viết

The screenshot shows the Entity Designer interface with the 'Entity' tab selected. The 'Entity Name' is set to 'Notification' and the 'Table Name' is also 'Notification'. Below the table name, there is a grid of attributes:

|    | Key | Name                 | Column Name            | Datatype      | Not null | Unique | Description |
|----|-----|----------------------|------------------------|---------------|----------|--------|-------------|
| 1  | PK  | NotificationId       | NotificationId         | Integer       | Not null |        |             |
| 2  | FK  | NotificationToUserId | UserId                 | Integer       | Not null |        |             |
| 3  | FK  | PostId               | PostId                 | Integer       |          |        |             |
| 4  | FK  | CommentId            | CommentId              | Integer       |          |        |             |
| 5  |     | NotificationFromUs   | NotificationFromUserId | Integer       |          |        |             |
| 6  |     | Content              | Content                | NVarChar(200) | Not null |        |             |
| 7  |     | Type                 | Type                   | Integer       | Not null |        |             |
| 8  |     | Status               | Status                 | Integer       |          |        |             |
| 9  |     | CreatedDate          | CreatedDate            | DateTime      | Not null |        |             |
| 10 | FK  | DatingRequestId      | DatingRequestId        | Integer       | Not null |        |             |

Bảng 3.5.14. Bảng Notification

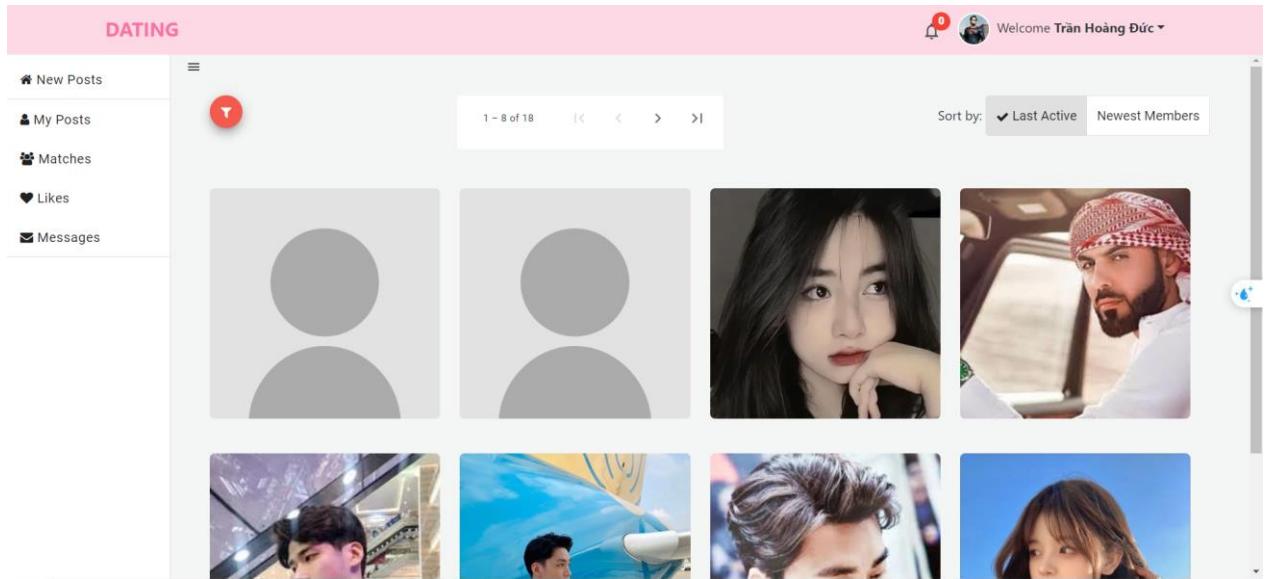
- Bảng DatingRequest được sử dụng để theo dõi và quản lý các yêu cầu hẹn hò giữa các người dùng

| Entity Name   |                 | Table Name      |            |          |         |                |        |              |          |          |       |     |
|---------------|-----------------|-----------------|------------|----------|---------|----------------|--------|--------------|----------|----------|-------|-----|
| DatingRequest |                 | > DatingRequest |            |          |         |                |        |              |          |          |       |     |
|               |                 | Attributes      | Properties | Advanced | Indexes | Alternate Keys | Others | Relationship | Storage  | Comments | Notes | DDL |
| 1             | DatingRequestId | DatingRequestId |            |          |         |                |        |              | Not null | Unique   |       |     |
| 2             | SenderId        | SenderId        |            |          |         |                |        |              | Not null |          |       |     |
| 3             | CrushId         | CrushId         |            |          |         |                |        |              | Not null |          |       |     |
| 4             | Status          | Status          |            |          |         |                |        |              | Not null |          |       |     |
| 5             | ConfirmedDate   | ConfirmedDate   |            |          |         |                |        |              | DateTime | Not null |       |     |

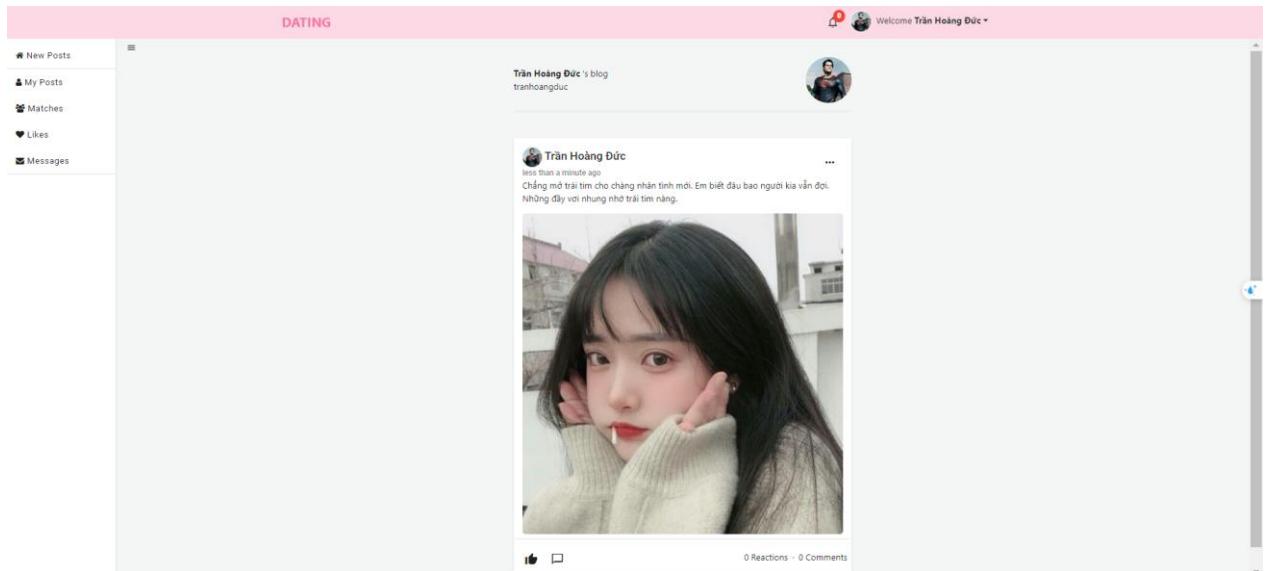
Bảng 3.5.15. Bảng DatingRequest

## CHƯƠNG 4: KẾT QUẢ VÀ KIỂM THỦ

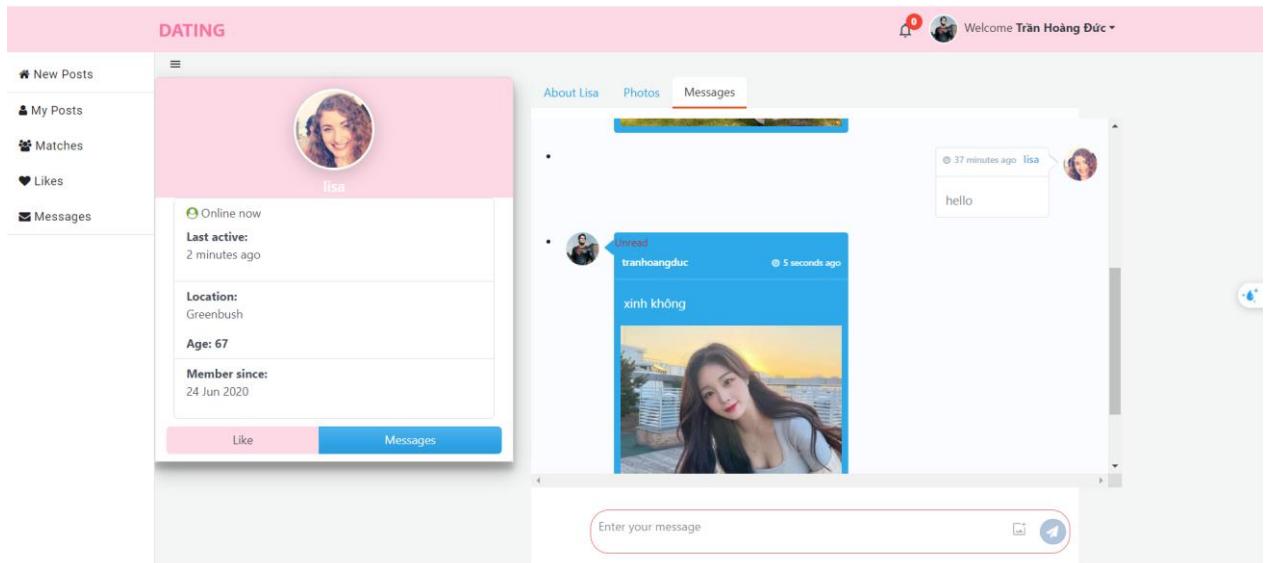
### 4.1. Giao diện



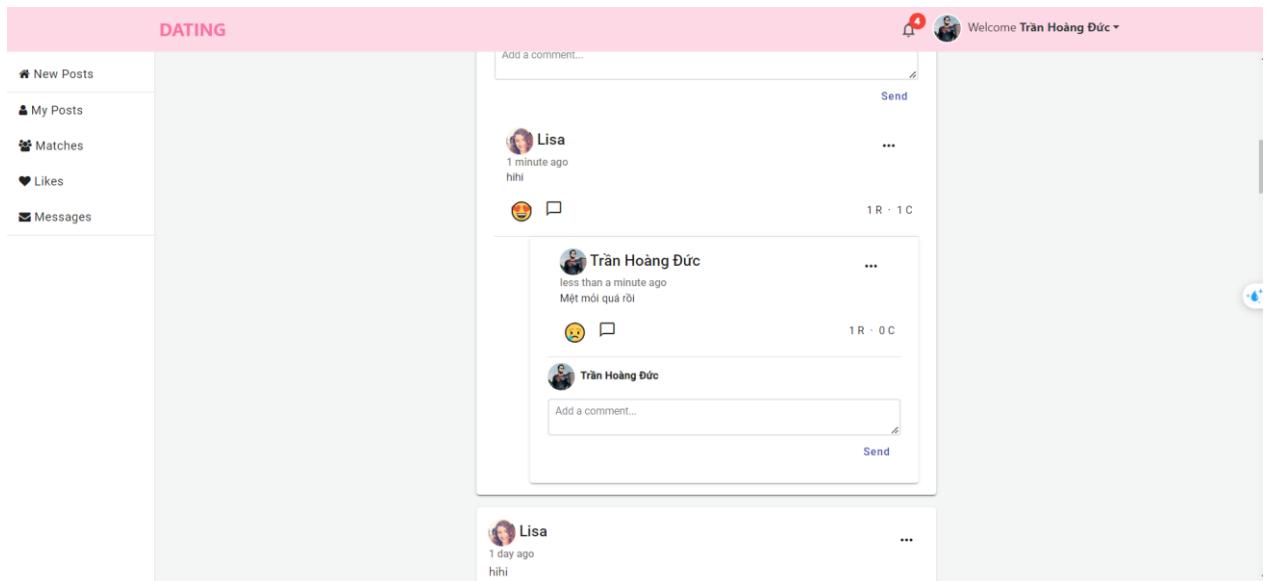
Hình 4.1. Giao diện màn hình Matches



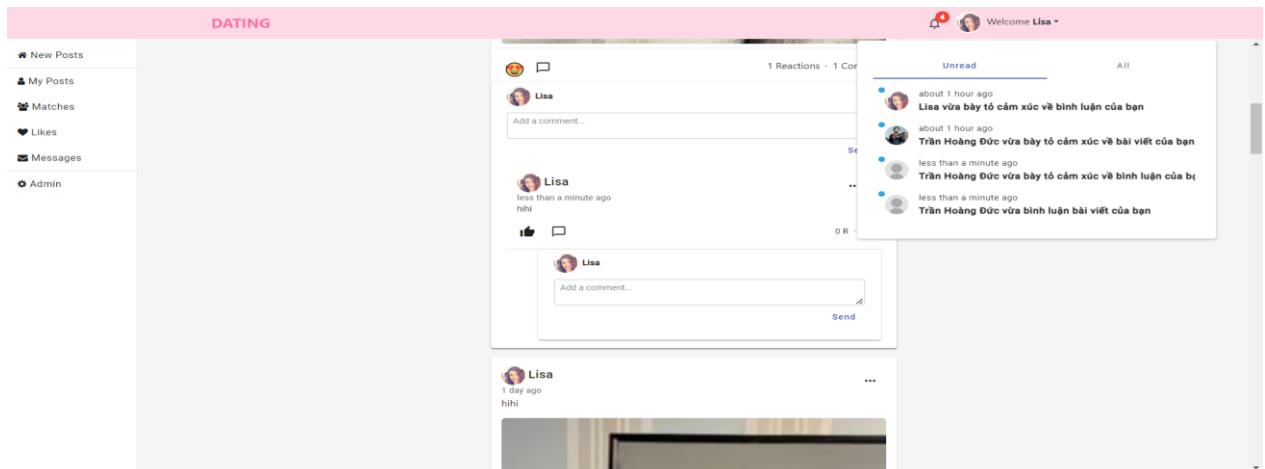
Hình 4.2. Giao diện màn hình Posts



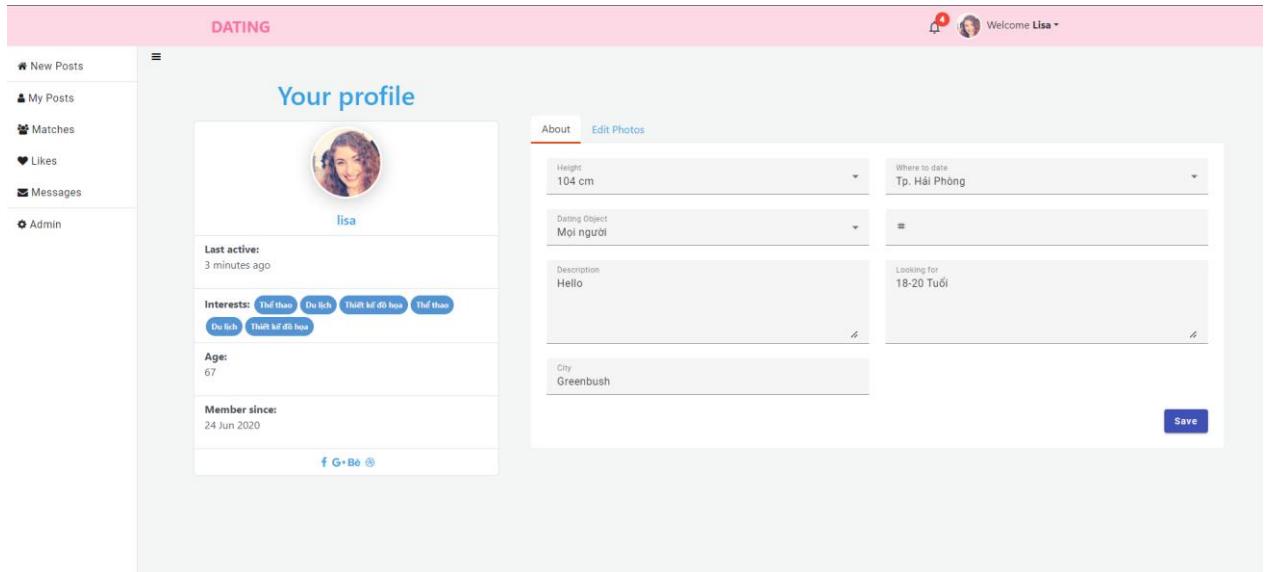
Hình 4.3. Giao diện màn hình tin nhắn



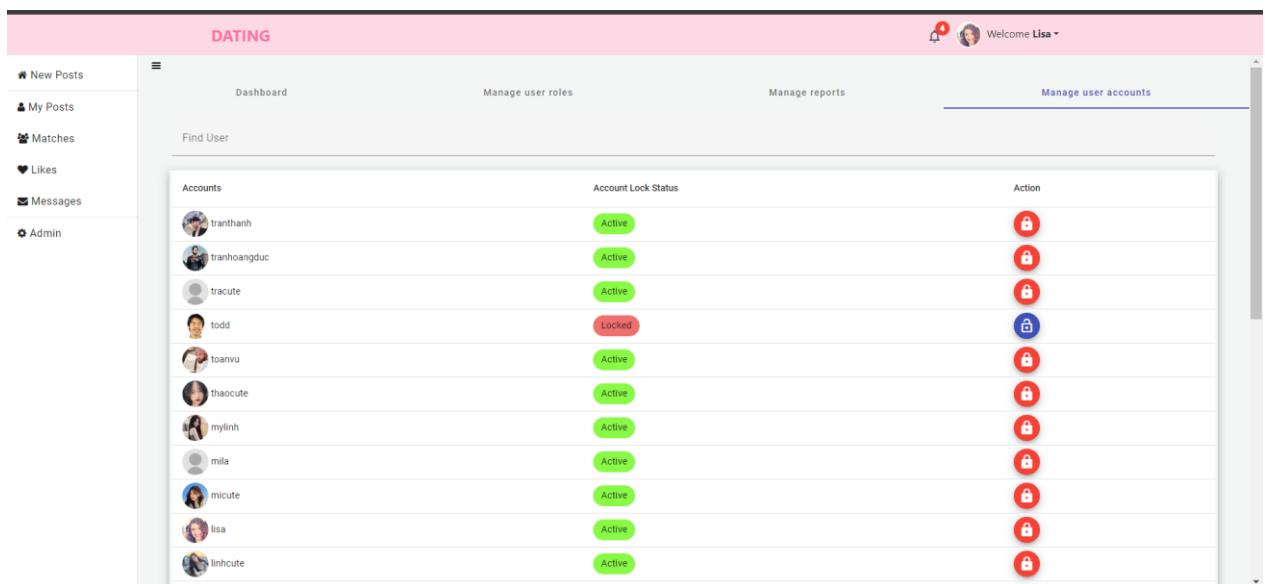
Hình 4.4. Giao diện màn hình tương tác bài đăng



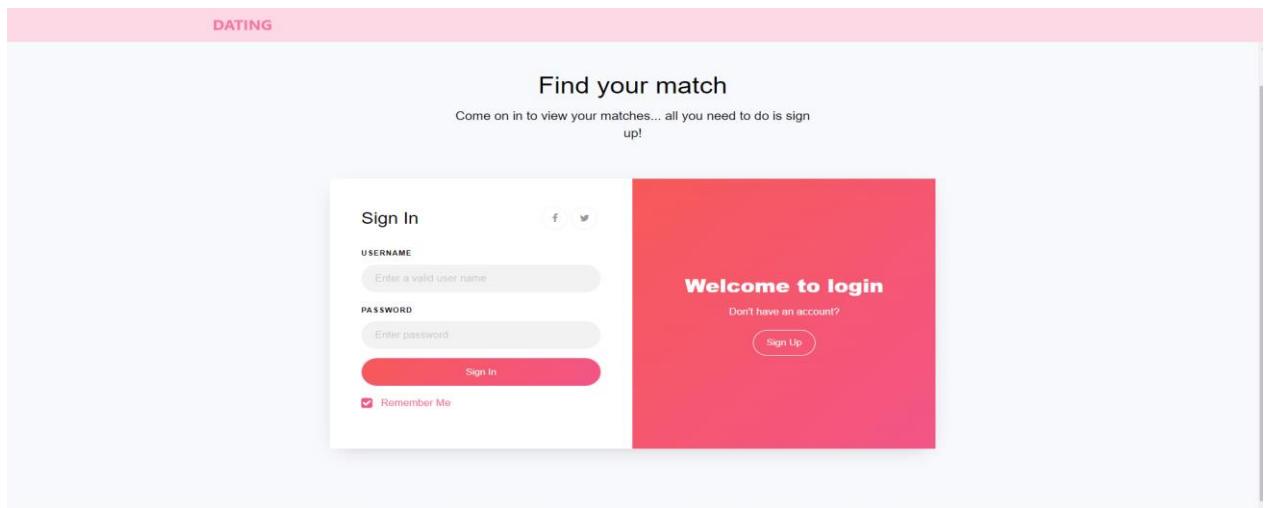
Hình 4.5. Giao diện màn hình thông báo



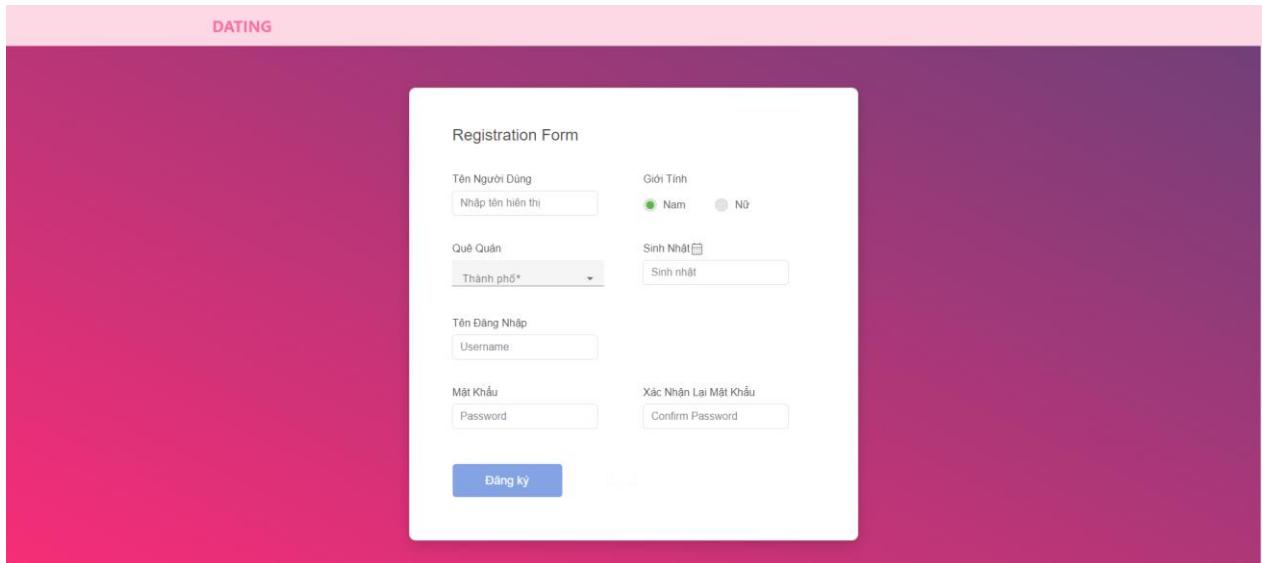
Hình 4.6. Giao diện màn hình trang cá nhân



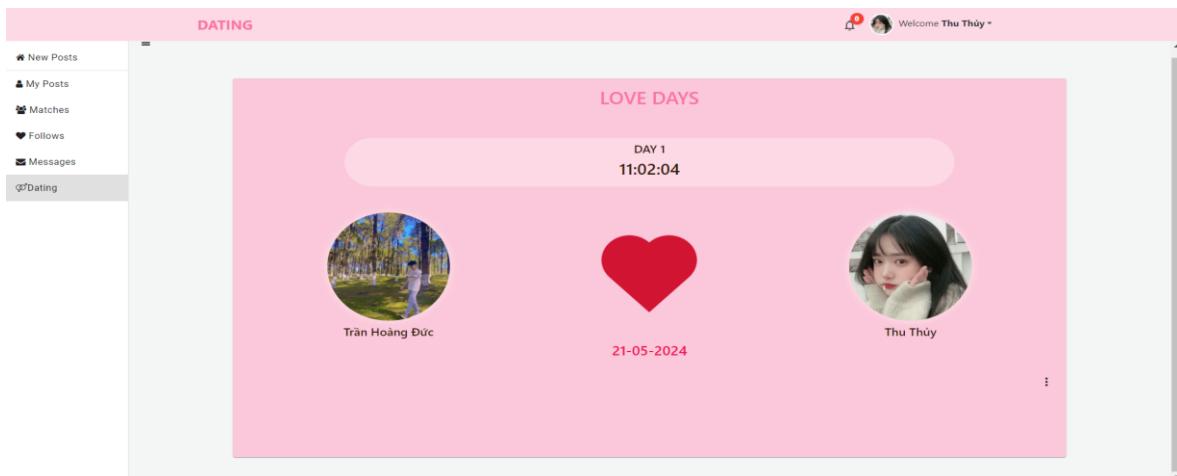
Hình 4.7. Giao diện màn hình Admin



Hình 4.8. Giao diện màn hình đăng nhập



Hình 4.9. Giao diện màn hình đăng ký



Hình 4.10. Giao diện màn hình hẹn hò

## 4.2. Kiểm thử

### 4.2.1. Kế hoạch kiểm thử

- Những chức năng được kiểm thử:
- + Đăng nhập: Người dùng đăng nhập vào hệ thống.
- + Đăng ký: Người dùng đăng ký tài khoản.
- + Quản lý thông tin cá nhân: Quản lý thay đổi thông tin cá nhân.
- + Nhắn tin: Nhắn tin làm quen giữa người dùng hiện tại và người dùng mục tiêu.
- + Hẹn hò: Người dùng có thể hẹn hò với đối tượng phù hợp với họ
- + Quản lý bài đăng: Quản lý có thể thêm sửa xóa các bài đăng cá nhân.

- + Tương tác bài đăng: Người dùng có thể bình luận, thả cảm xúc, báo cáo bài đăng của người khác.
- + Quản lý tài khoản: Quản lý cấp quyền thu hồi quyền, mở và khóa tài khoản người dùng.
- + Tìm kiếm người dùng: Tìm kiếm người dùng với các thông tin được truyền vào.
- + Thông báo: Thông báo về các tương tác và phản hồi của người dùng khác với bài đăng người dùng hiện tại.

#### 4.2.2. Thiết kế ca kiểm thử

| STT | Trường hợp                                     | Đầu vào   | Đầu ra  |
|-----|--|---|---|
| 1   | Kiểm thử chức năng đăng nhập                   | Nhập tên đăng nhập và mật khẩu hợp lệ.                        | Đăng nhập thành công điều hướng về trang chủ.                       |
|     |  | Nhập tên đăng nhập và mật khẩu không hợp lệ.                  | Đăng nhập không thành công và có thông báo ra màn hình.             |
| 2   | Kiểm thử chức năng đăng ký tài khoản           | Nhập thông tin đăng ký hợp lệ, xác nhận lại mật khẩu đã khớp. | Đăng ký thành công rồi điều hướng đến trang chủ.                    |
|     |  | Nhập thông thiếu thông tin.                                   | Thông báo cần phải nhập đủ thông tin mới tạo được tài khoản         |
|     |  | Nhập tên đăng ký đã tồn tại.                                  | Đăng ký không thành công và hiển thị thông báo.                     |
| 3   | Kiểm thử chức năng tìm kiếm người dùng phù hợp | Điền các thông tin vào màn hình tìm kiếm.                     | Hiển thị các người dùng có thông tin phù hợp với thông tin tìm kiếm |

|   |  |  |  |
|---|--|--|--|
|   |  |  |  |
| 4 | Kiểm thử<br>chức năng<br>nhắn tin                    | Không điền tin nhắn và<br>nhắn gửi   | Hiển thị thông báo tin<br>nhắn không được để<br>trống  |
|   |  | Nhập tin nhắn và nhấn<br>gửi   | Hiển thị tin nhắn vừa gửi<br>cho người nhận  |
|   |  | Nhập tin nhắn kèm theo<br>file ảnh và nhấn gửi   | Hiển thị tin nhắn kèm file<br>ảnh cho người nhận   |
| 5 | Kiểm thử<br>chức năng sửa<br>thông tin cá<br>nhân    | Chọn các thông tin cần<br>chỉnh sửa và nhập thông<br>tin chỉnh sửa mong muốn<br>rồi nhấn lưu           | Hiển thị thông báo đã cập<br>nhật thông tin thành công<br>và hiển thị lên thông tin<br>mới vừa chỉnh sửa                 |
|   |  | Chọn các ảnh cần tải lên<br>rồi nhấn upload  | Hiển thị các ảnh cá nhân<br>vừa tải lên, nếu chưa có<br>ảnh đại diện lấy bức ảnh<br>tải lên đầu tiên làm ảnh<br>đại diện |
| 6 | Kiểm thử<br>chức năng<br>quản lý bài<br>đăng cá nhân | Chọn tạo mới bài đăng,<br>nhập nội dung bài đăng<br>và thêm các hình ảnh bài<br>đăng rồi chọn đăng bài | Hiển thị thông báo tạo bài<br>đăng thành công và<br>chuyển hướng đến kho<br>bài đăng cá nhân                             |

|   |  |  |   |
|---|--|--|---|
|   |  | <p>Chọn chỉnh sửa bài đăng.</p> <p>Nhập nội dung và tải lên hình ảnh mới rồi nhấn cập nhật</p>   | Hiển thị thông báo cập nhật bài đăng thành công và hiển thị bài đăng mới vừa cập nhật |
|   |  | Nhập thông tin không hợp lệ.   | Hiển thị thông báo thêm, sửa thất bại.  |
|   |  | Chọn xóa bài đăng  | Hiển thị thông báo xoá thành công và hiển thị danh sách bài đăng cá nhân vừa cập nhật |
| 7 | Kiểm thử chức năng bình luận và thả cảm xúc bài đăng | Chọn bình luận, nhập bình luận và nhấn gửi   | Hiển thị trong khung bình luận bình luận vừa tạo                                      |
|   |  | Bình luận gốc đã tồn tại.<br>Nhập nội dung trả lời bình luận                                     | Bình luận trả lời được thêm vào dưới bình luận gốc                                    |
|   |  | Bình luận gốc đã tồn tại.<br>Nội dung nhập là chuỗi rỗng hoặc chỉ có khoảng trắng                | Thông báo lỗi nội dung bình luận không được để trống                                  |
|   |  | Bình luận đã tồn tại, người dùng hợp lệ và có quyền cập nhật bình luận, nội dung cập nhật hợp lệ | Bình luận được cập nhật với nội dung mới.   |
|   |  | Bình luận đã tồn tại, người dùng hợp lệ và có quyền xóa bình luận                                | Bình luận được xóa khỏi hệ thống. Hiển thị thông báo xóa thành công.                  |

|   |  |   |  |
|---|--|---|--|
|   |  | Bài đăng đã tồn tại.<br>Người dùng hợp lệ và có quyền thả cảm xúc. Loại cảm xúc hợp lệ (ví dụ: thích, yêu thích, ngạc nhiên, buồn, giận dữ)       | Cảm xúc được ghi nhận và hiển thị dưới bài đăng.   |
|   |  | Bình luận đã tồn tại.<br>Người dùng hợp lệ và có quyền thả cảm xúc.   | Cảm xúc được ghi nhận và hiển thị dưới bình luận.  |
| 8 | Kiểm thử chức năng thông báo về bài đăng | Bài đăng đã tồn tại.<br>Người dùng A đã đăng bài. Người dùng B đăng bình luận mới trên bài đăng của người dùng A.                                 | Người dùng A nhận được thông báo về bình luận mới. Thông báo chứa thông tin về bình luận (người bình luận, nội dung bình luận, thời gian bình luận). |
|   |  | Bình luận đã tồn tại trên bài đăng. Người dùng B đã bình luận trên bài đăng của người dùng A.<br>Người dùng C trả lời bình luận của người dùng B. | Người dùng B nhận được thông báo về trả lời bình luận. Thông báo chứa thông tin về người trả lời, nội dung trả lời, và thời gian trả lời.            |
| 9 | Kiểm thử chức năng phân quyền tài khoản  | Tài khoản người dùng đã tồn tại. Quản trị viên thực hiện việc phân quyền. Quyền được phân hợp lệ  | Quyền được phân chính xác cho tài khoản người dùng. Thông báo thành công hiển thị cho quản trị viên.   |

|    |                                      |  |   |
|----|--------------------------------------|--|---|
| 10 | Kiểm thử chức năng quản lý tài khoản | Tài khoản người dùng đã tồn tại. Quản trị viên thực hiện việc khóa tài khoản.        | Tài khoản bị khóa. Người dùng bị khóa không thể đăng nhập hoặc sử dụng các chức năng của hệ thống.  |
|    |                                      | Tài khoản người dùng đã bị khóa. Quản trị viên thực hiện việc mở khóa tài khoản.     | Tài khoản được mở khóa và người dùng có thể đăng nhập và sử dụng lại các chức năng của hệ thống.    |
| 11 | Kiểm thử chức năng quản lý bài đăng  | Bài đăng bị báo cáo vi phạm nguyên tắc. Admin thực hiện xóa bài đăng.                | Bài đăng bị xóa khỏi hệ thống. Thông báo thành công hiển thị cho admin.                             |
| 12 | Kiểm thử chức năng hẹn hò            | Người A gửi yêu cầu hẹn hò đến người B. Người B chấp nhận yêu cầu hẹn hò             | Thông báo cho người A người B đã chấp nhận yêu cầu hẹn hò.<br>Hiển thị màn hình hẹn hò của 2 người. |
|    |                                      | Người A gửi yêu cầu hẹn hò đến người B. Người B không chấp nhận yêu cầu hẹn hò.      | Thông báo cho người A người B không chấp nhận yêu cầu hẹn hò  |
|    |                                      | Người A đang trong một mối quan hệ hẹn hò.<br>Người B gửi yêu cầu hẹn hò đến người A | Thông báo cho người B người A đang trong một mối quan hệ nên không thể hẹn hò với bạn được          |

|  |  |  |  |
|--|--|--|--|
|  |  | <p>Người A và người B đang trong một mối quan hệ. Người A chọn “Kết thúc hẹn hò”</p> | <p>Người A và người B không còn mối quan hệ hẹn hò nữa.</p> <p>Thông báo cho người B là người A đã kết thúc hẹn hò</p> |
|--|--|--|--|

#### 4.2.3. Thực thi kiểm thử

| STT | Nội dung  | Mục đích kiểm thử | Đầu vào  | Đầu ra mong muốn  | Kết quả |
|-----|-----------|-------------------|--|---|---------|
| 1   | Tài khoản | Đăng ký tài khoản | <p>Known As: “Trần Hoàng Đức”</p> <p>User name: tranhoangduc</p> <p>Role: “Member”</p> <p>Password: Duc@12345</p> <p>Confirm password: Duc@12345</p> <p>Date of birth: 01/03/2002</p> <p>Address: Nam Định</p> <p>Dating Object: Male</p> <p>Where to date: Hà Nội</p> <p>Dating age from: 18</p> <p>Dating age to: 30</p> <p>Height: 170 cm</p> <p>Interests: Đá bóng, xem phim</p> | Tạo tài khoản thành công người dùng có thể sử dụng các chức năng của hệ thống | Pass    |
|     |           |                   | Để trống một hoặc nhiều trường dữ liệu   | Đưa ra thông báo yêu cầu nhập đầy đủ thông tin                                | Pass    |

|  |                     |  |  |      |
|--|---------------------|--|--|------|
|  |                     | <p>assword và Confirm password không nhất quán:<br/>Password: Duc@12345<br/>Confirm password:<br/>Duc@123456</p> | Đưa ra cảnh báo yêu cầu hai trường mật khẩu cần giống nhau | Pass |
|  |                     | <p>Username: tranhoangduc<br/>Password: Duc@12345</p>  | Đăng nhập thành công                                       | Pass |
|  | Đăng nhập tài khoản | <p>Username: tranhoangduc<br/>Password: 123456</p>   | Đăng nhập thất bại và đưa ra thông báo                     | Pass |
|  |                     | <p>Thông tin rỗng</p>  | Thông báo yêu cầu điền đầy đủ thông tin                    | Pass |

|   |          |                             |   |   |      |
|---|----------|-----------------------------|---|---|------|
|   |          | Sửa đổi thông tin cá nhân   | Tại Profile điền mới một hoặc nhiều trường dữ liệu:<br>Introduction: Để làm quen<br>Address: Hà Nội<br>Where to date: Tp. Hải Phòng<br>Dating object: Female<br>Dating age from: 25<br>Dating age to: 30<br><br>Chọn Save changes | Thông báo cập nhật thông tin thành công   | Pass |
| 2 | Bài đăng | Hiển thị Danh sách bài đăng | Kích chọn mục New Posts   | Hiển thị toàn bộ các bài đăng của những người dùng hệ thống với thông tin bao gồm: Tên người đăng, nội dung, hình ảnh, thời gian đăng | Pass |

|  |  |                            |  |   |      |
|--|--|----------------------------|--|---|------|
|  |  | Thả reaction bài bình luận | Người dùng nhấn vào biểu tượng reaction trên bài đăng hoặc một bình luận, hệ thống sẽ hiển thị ra các reaction như “thích, yêu thích, ngạc nhiên, buồn, giận dữ”. Người dùng chọn reaction “Yêu thích” | Biểu tượng reaction bài đăng hoặc bình luận sẽ cập nhật thành reaction “Yêu thích”                  | Pass |
|  |  | Bình luận                  | Kích vào bình luận trên bài đăng hệ thống sẽ hiển thị form bình luận. Người dùng nhập bình luận ví dụ “Cô gái này rất đẹp” và chọn “Gửi”   | Hệ thống sẽ lưu bình luận và hiển thị bình luận “Cô gái này rất đẹp” trên mục bình luận bài đăng đó | Pass |

|   |           |                       |   |   |      |
|---|-----------|-----------------------|---|---|------|
|   |           | Trả lời bình luận     | <p>Chọn bình luận “Cô gái này rất đẹp”, chọn trả lời bình luận.</p> <p>Người dùng nhập bình luận trả lời “Cảm ơn” rồi nhấn gửi</p>  | Hệ thống sẽ lưu bình luận và hiển thị bình luận “Cảm ơn” trong mục bình luận con của “Cô gái này rất đẹp”                 | Pass |
|   |           | Báo cáo bài đăng      | <p>Người dùng chọn mục báo cáo bài đăng trên bài đăng.</p> <p>Hệ thống hiển thị form báo cáo. Ví dụ người dùng chọn “Nội dung xấu, xúc phạm, hay kỳ thị” và nhập mô tả “Bài đăng phản cảm” rồi chọn gửi</p> | Hiển thị thông báo thành công và báo cáo về bài đăng đó được hiển thị trong mục quản lý báo cáo bài đăng ở màn hình Admin | Pass |
| 3 | Thông báo | Thông báo về bài đăng | Một người dùng bình luận, thả cảm xúc trên bài đăng của người dùng hiện tại. Ví dụ người dùng A bình luận bài viết người dùng B   | Người dùng B nhận được thông báo “Người dùng A vừa bình luận bài viết của bạn”  | Pass |

|   |          |                             |   |  |      |
|---|----------|-----------------------------|---|--|------|
|   |          | Thông báo về yêu cầu hẹn hò | Người dùng A gửi yêu cầu hẹn hò đến người dùng B.   | Người dùng B nhận được thông báo xác nhận hẹn hò với người dùng A                      | Pass |
| 4 | Nhắn tin | Gửi và nhận tin nhắn        | Người dùng A nhắn tin với người dùng B. Người dùng A nhập tin nhắn “Xin chào” kèm theo ảnh và chọn gửi      | Người dùng B nhận được tin nhắn có nội dung “Xin chào” và hình ảnh đi kèm              | Pass |
| 5 | Hẹn hò   | Gửi yêu cầu hẹn hò          | Người dùng A gửi yêu cầu hẹn hò đến người dùng B. Nhưng người dùng B đang trong một mối quan hệ hẹn hò khác | Hiển thị thông báo “Người dùng B đang ở trong một mối quen hệ hẹn hò, hãy thử lại sau” | Pass |
|   |          |                             | Người dùng A gửi yêu cầu hẹn hò đến người dùng B.   | Mục thông báo ở bên người dùng B hiển thị “Người dùng A vừa gửi yêu cầu hẹn            | Pass |

|  |                                  |  |  |      |  |
|--|----------------------------------|--|--|------|--|
|  |                                  |  | hò đến bạn,<br>vui lòng xác<br>nhận yêu<br>cầu”  |      |  |
|  | Từ chối<br>yêu cầu<br>hẹn hò     | Người dùng B từ chối yêu<br>cầu hẹn hò của người dùng<br>A | Mục thông<br>báo ở bên<br>người dùng<br>A hiển thị<br>“Người dùng<br>B đã từ chối<br>yêu cầu ghép<br>đôi của bạn”  | Pass |  |
|  | Xác<br>nhận<br>yêu cầu<br>hẹn hò | Người dùng B xác nhận<br>ghép đôi với người dùng A         | Thông báo<br>người dùng<br>A hiển thị<br>“Chúc mừng,<br>bạn và người<br>dùng B đã<br>trở thành 1<br>cặp”. Màn<br>hình hẹn hò<br>hiển thị<br>thông tin hẹn<br>của 2 người | Pass |  |

|  |  |                 |   |  |      |
|--|--|-----------------|---|--|------|
|  |  | Kết thúc hẹn hò | Người dùng A chọn kết thúc hẹn hò với người dùng B và nhấn xác nhận | Mục thông báo ở bên người dùng B hiển thị “Người dùng A đã hủy hẹn hò với bạn”, màn hình hẹn hò của người dùng B trở về mặc định | Pass |
|--|--|-----------------|---|--|------|

#### 4.2.3. Kết quả kiểm thử

- Tỉ lệ test case đạt: 100%
- Tỉ lệ test case thất bại: 0%
- Hệ thống chạy ổn định trên các trình duyệt web khác nhau như Firefox, Google Chrome, Microsoft Edge.

## KẾT LUẬN

**Đề tài “Xây dựng trang web hẹn hò DatingApp sử dụng framework Angular 16 và .Net 8”** xuất phát từ những nhu cầu tìm kiếm tình cảm của những người đang độc thân và mang lại tiềm năng phát triển trong tương lai. Qua quá trình thực hiện đề tài, em đã tìm hiểu, tích lũy và vận dụng được những kiến thức công nghệ được học từ trường và đã hoàn thành được những phần sau đây.

### **Kết quả đạt được**

- Cho phép tìm kiếm, hiển thị những người có độ phù hợp với mình
- Cung cấp một nền tảng kết nối mọi người với nhau thông qua việc nhắn tin, đăng tải các suy nghĩ cá nhân, những hình ảnh đẹp của bản thân để mọi người dễ dàng kết nối, tìm cho mình một nửa phù hợp.
- Quản lý người dùng, các bài đăng, các quyền tạo ra một môi trường hoạt động cân đối, an toàn và thuận tiện cho người dùng.

### **Hướng nghiên cứu trong tương lai**

- Tiếp tục hoàn thiện các tính năng mở rộng phần mềm nhằm cải tiến và nâng cấp chương trình.
- Hoàn thiện chương trình, sửa các lỗi và cải thiện tính bảo mật.
- Tích hợp các thuật toán và công nghệ để đề xuất những người dùng phù hợp với người dùng hiện tại
- Tích hợp Single Sign On như facebook, gmail...
- Tối ưu cách viết code và cơ sở dữ liệu.
- Xây dựng thêm quản lý tính lương tự động.

Em muốn bày tỏ lòng biết ơn chân thành nhất đến quý thầy cô trong khoa Công nghệ thông tin vì đã dành tâm huyết giảng dạy và truyền đạt kiến thức quý giá và cần thiết trong những năm học tại trường, đóng góp quan trọng cho sự hoàn thiện của đề tài đồ án tốt nghiệp này. Đặc biệt, em muốn gửi lời cảm ơn đặc biệt tới cô TS. Nguyễn Thị Hương Lan đã hướng dẫn và hỗ trợ tận tâm,

giúp em hoàn thành báo cáo đồ án tốt nghiệp cùng sản phẩm đi kèm một cách hoàn thiện nhất.

Do còn thiếu kinh nghiệm thực tế cũng như thời gian hạn chế, ứng dụng của em vẫn còn nhiều hạn chế. Vì vậy, em mong rằng các quý thầy cô có thể đóng góp ý kiến để giúp em hoàn thiện ứng dụng tốt hơn trong tương lai.

## TÀI LIỆU THAM KHẢO

### Tiếng việt

[1] Hoàng Quang Huy, Phùng Đức Hòa, Trịnh Bá Quý, Nhập môn công nghệ phần mềm, NXB Đại học Công nghiệp Hà Nội.

[2] Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Phạm Thị Kim Phượng, Giáo trình phân tích thiết kế hệ thống, NXB Giáo dục VN.

[3] Giáo trình thiết kế web, Trường đại học Công Nghiệp Hà Nội.

[4]Trang tài liệu lập trình .Net(truy cập lần cuối 10/5/2024)

<https://dotnet.microsoft.com/en-us/download/dotnet/8.0>

[5]Trang tài liệu Sql Server(truy cập lần cuối 10/5/2024)

<https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>

[6]Trang tài liệu Angular(truy cập lần cuối 10/5/2024)

<https://angular.io/docs>