

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



**PBL4: DỰ ÁN HỆ ĐIỀU HÀNH &
MẠNG MÁY TÍNH**

Đề tài

**XÂY DỰNG HỆ THỐNG KẾT BẠN TÂM GIAO
DỰA VÀO ĐỘ TƯƠNG THÍCH CUNG HOÀNG ĐẠO**

SINH VIÊN THỰC HIỆN:

Nguyễn Thị Hồng Diễm LỚP: 21T_DT NHÓM: 04

Nguyễn Phương Hà LỚP: 21T_DT NHÓM: 04

Lê Nguyễn Yến Nhi LỚP: 21T_DT NHÓM: 04

GIẢNG VIÊN HƯỚNG DẪN: Ths Nguyễn Công Danh

Đà Nẵng 12/ 2023

MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	7
1.1. Giao thức TCP	7
1.2. Mô hình Client/Server	7
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	9
2.1. Chức năng của hệ thống	9
2.2. Biểu đồ ca sử dụng và đặc tả	10
2.3. Biểu đồ hoạt động	14
2.4. Thiết kế cơ sở dữ liệu	34
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	35
3.1. Triển khai hệ thống	35
3.2. Kết quả	35
3.2.1 Phía User	35
3.2.2 Phía Admin	56
3.3. Đánh giá	59
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	60

DANH SÁCH HÌNH VẼ

Hình 1 Mô hình Client – Server	8
Hình 2 Sơ đồ usecase tổng quan	10
Hình 3 Sơ đồ usecase quản lý tài khoản	10
Hình 4 Sơ đồ usecase quản lý thông điệp	11
Hình 5 Sơ đồ usecase quản lý report	11
Hình 6 Sơ đồ usecase quản lý bài post.....	12
Hình 8 Sơ đồ usecase kết bạn (matching)	12
Hình 7 Sơ đồ usecase nhắn tin	13
Hình 9 Sơ đồ usecase tìm kiếm tài khoản	13
Hình 10 Sơ đồ usecase xem thông báo	14
Hình 11 Cơ sở dữ liệu	34
Hình 12 Giao diện đăng nhập	35
Hình 13 Giao diện đăng ký	36
Hình 14 Giao diện trang chủ matching	36
Hình 15 Giao diện không có user tương thích.....	37
Hình 16 Giao diện request matching	37
Hình 17 Giao diện nhận thông báo request matching	38
Hình 18 Giao diện chấp nhận kết bạn.....	40
Hình 19 Giao diện đi đến cuộc trò chuyện sau khi matching thành công	41
Hình 20 Giao diện thông báo request matching.....	41
Hình 21 Giao diện xem chi tiết thông báo request matching	42
Hình 22 Giao diện tìm kiếm bạn bè.....	43
Hình 23 Giao diện xem blog bạn bè.....	43
Hình 24 Giao diện chat với bạn bè	44
Hình 25 Giao diện block tin nhắn	46
Hình 26 Giao diện xóa tin nhắn	48
Hình 27 Giao diện tìm kiếm tin nhắn	48
Hình 28 Giao diện nhận thông báo tin nhắn.....	49
Hình 29 Giao diện nhận thông điệp tuần mới từ Admin	51
Hình 30 Giao diện xem chi tiết thông báo thông điệp được gửi từ Admin	53

Hình 31 Giao diện xem trang blog cá nhân.....	53
Hình 32 Giao diện chỉnh sửa nội dung bài viết	54
Hình 33 Giao diện xóa bài viết	54
Hình 34 Giao diện chỉnh sửa thông tin cá nhân.....	55
Hình 35 Giao diện thay đổi mật khẩu	55
Hình 36 Giao diện xem danh sách toàn bộ user.....	56
Hình 37 Giao diện xem chi tiết thông tin user.....	56
Hình 38 Giao diện xóa user	57
Hình 39 Giao diện tạo thông điệp hàng tuần cho 12 cung hoàng đạo.....	57
Hình 40 Giao diện xem danh sách thông điệp cho các cung hoàng đạo	58
Hình 41 Giao diện xem chi tiết thông điệp của từng cung hoàng đạo	58
Hình 42 Giao diện xem danh sách các bài viết bị report	59
Hình 43 Giao diện xem thông tin chi tiết các report và giải quyết	59

DANH SÁCH BẢNG VẼ

Bảng 1 Biểu đồ hoạt động nhấn tin	14
Bảng 2 Biểu đồ hoạt động xóa cuộc trò chuyện	15
Bảng 3 Biểu đồ hoạt động nhấn tin	16
Bảng 4 Biểu đồ hoạt động bỏ chặn nhấn tin.....	17
Bảng 5 Biểu đồ hoạt động tìm kiếm tin nhắn	18
Bảng 6 Biểu đồ hoạt động hiển thị toàn bộ bài post.....	19
Bảng 7 Biểu đồ hoạt động thêm bài post mới	20
Bảng 8 Biểu đồ hoạt động chỉnh sửa bài post	21
Bảng 9 Biểu đồ hoạt động xóa bài post	22
Bảng 10 Biểu đồ hoạt động report bài post	23
Bảng 11 Biểu đồ hoạt động xem danh sách thông điệp.....	23
Bảng 12 Biểu đồ hoạt động xem chi tiết thông điệp	24
Bảng 13 Biểu đồ hoạt động tạo mới thông điệp	25
Bảng 14 Biểu đồ hoạt động xem danh sách bài viết bị report.....	25
Bảng 15 Biểu đồ hoạt động giải quyết bài viết bị report	26
Bảng 16 Biểu đồ hoạt động xem danh sách tài khoản.....	27
Bảng 17 Biểu đồ hoạt động xem chi tiết tài khoản	27
Bảng 18 Biểu đồ hoạt động xóa tài khoản	28
Bảng 19 Biểu đồ hoạt động chỉnh sửa profile	29
Bảng 20 Biểu đồ hoạt động thay đổi mật khẩu.....	30
Bảng 21 Biểu đồ hoạt động kết bạn.....	31
Bảng 22 Biểu đồ hoạt động xem thông báo matching.....	32
Bảng 23 Biểu đồ hoạt động xem thông báo thông điệp gửi từ admin.....	33
Bảng 24 Biểu đồ hoạt động tìm kiếm bạn bè (không bị chặn)	33

GIỚI THIỆU ĐỀ TÀI

Trong bối cảnh xã hội ngày nay, mạng xã hội đã trở thành một phần quan trọng của cuộc sống hàng ngày, giúp mọi người kết nối, chia sẻ và tương tác một cách nhanh chóng. Tuy nhiên, thách thức đặt ra không chỉ là việc kết nối mà còn là sự duy trì và phát triển mối quan hệ thực sự có ý nghĩa. Việc tìm kiếm, xác định và duy trì mối quan hệ trong môi trường trực tuyến vẫn còn là một vấn đề của nhiều ứng dụng mạng xã hội ngày nay.

Vì vậy, nhóm chúng em đã lựa chọn đề tài xây dựng hệ thống kết bạn tâm giao dựa trên độ tương thích cung hoàng đạo, với đặc điểm nổi bật là tăng độ hiệu quả trong việc xây dựng và kết nối các mối quan hệ trực tuyến. Hệ thống gồm các chức năng chính như kết bạn, nhắn tin, truyền file và đăng bài viết trên trang cá nhân, đạt được những yêu cầu chức năng cơ bản của một ứng dụng mạng xã hội.

Hệ thống được hứa hẹn sẽ đem đến những trải nghiệm thú vị cho người dùng, tạo ra một không gian trực tuyến chân thật và ý nghĩa. Đây cũng là cơ hội để ứng dụng công nghệ trong việc xây dựng các mối quan hệ thực sự trong thế giới ảo.

Trong quá trình thực hiện, chắc chắn không thể không tồn tại những thiếu sót, chúng em mong được thầy cô thông cảm và tận tình chỉ dẫn để đề tài của chúng em ngày càng hoàn thiện hơn.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Công Danh đã tận tình giúp đỡ, chỉ bảo và giải đáp thắc mắc của chúng em trong suốt quá trình làm đồ án.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Giao thức TCP

Transmission Control Protocol (TCP) là một giao thức tiêu chuẩn quan trọng đảm bảo trao đổi thành công các gói dữ liệu giữa các thiết bị qua mạng. Giao thức TCP là một trong các giao thức cốt lõi hoạt động cùng với giao thức IP trong bộ giao thức TCP/IP, trong đó giao thức IP sẽ xử lý việc gán địa chỉ và chuyển tiếp các gói tin từ nguồn đến đích trong khi TCP kiểm soát độ tin cậy của truyền dẫn.

Về cách thức hoạt động, TCP đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất. TCP đảm bảo rằng các gói dữ liệu được gửi đi sẽ đến đích một cách chính xác và đúng thứ tự. Nó xác định các kết nối, đồng bộ hóa truyền tải và quản lý lưu lượng dữ liệu.

Ngày nay, bộ giao thức TCP/IP đóng vai trò quan trọng trong việc kết nối thông tin trên Internet. Người dùng có khả năng cung cấp thông tin từ xa, gửi email, truyền file, chia sẻ hình ảnh và triển khai trang web trên mạng Internet thông qua cơ sở hạ tầng mạng được xây dựng dựa trên bộ giao thức TCP/IP.

1.2. Mô hình Client/Server

1.2.1. Khái niệm

Client/Server là một mô hình mạng máy khách - máy chủ bao gồm hai thành phần chính: máy khách (client) - máy chủ (server).

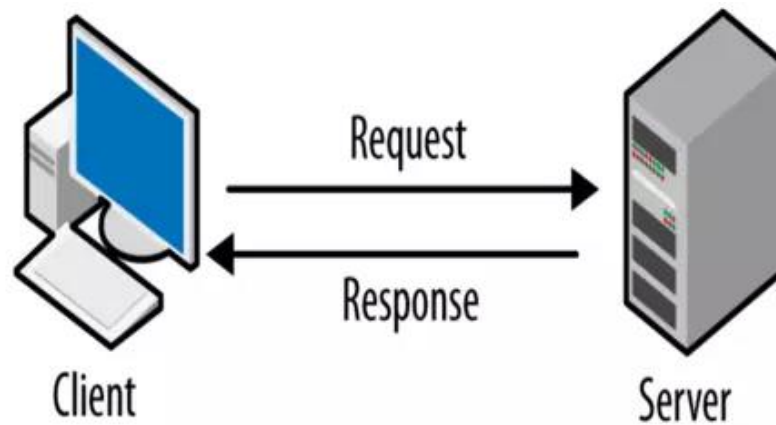
Máy chủ (server) thường là một hệ thống máy lớn, có bộ xử lý mạnh, có khả năng hoạt động đáng tin cậy và lưu trữ dữ liệu lớn, nó chuyên quản lý tài nguyên và cung cấp các dịch vụ mạng cho các máy khách (client) sử dụng. Bình thường nó chạy suốt thời gian thực và sẵn sàng chấp nhận các yêu cầu kết nối và yêu cầu dịch vụ khác từ máy khách.

Máy khách (client) thường được sử dụng bởi người dùng cuối, nó hoạt động dựa trên việc sử dụng dịch vụ mà server cung cấp để thực hiện các công việc mà người dùng cuối mong muốn.

1.2.2. Quy trình hoạt động

Quy trình hoạt động của hai mô hình này lặp lại 2 quá trình như sau:

- Client gửi yêu cầu lên server
- Server nhận được yêu cầu thì sẽ xử lý thích hợp và trả lời lại cho client



Hình 1 Mô hình Client – Server

1.2.3. Nguyên tắc hoạt động

Đầu tiên, khách hàng gửi yêu cầu đến máy chủ, các yêu cầu này có thể là yêu cầu lấy dữ liệu, yêu cầu xử lý một tác vụ nào đó hay là yêu cầu thực hiện một chức năng cụ thể

Sau khi nhận được yêu cầu, máy chủ sẽ xử lý yêu cầu này và trả về kết quả tương ứng cho khách hàng. Việc xử lý này có thể là tìm kiếm, truy vấn dữ liệu hoặc thực hiện tính toán phức tạp, tùy thuộc vào yêu cầu của khách hàng

Sau khi máy chủ xử lý yêu cầu, kết quả sẽ được trả về cho khách hàng. Kết quả này có thể là dữ liệu hoặc kết quả của các tính toán hoặc chức năng được yêu cầu

CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Chức năng của hệ thống

Hệ thống gồm có hai phân quyền là admin và user. Cả hai phân quyền đều cần phải đăng nhập vào hệ thống mới có thể sử dụng được các chức năng tương ứng với từng phân quyền.

2.1.1 Admin

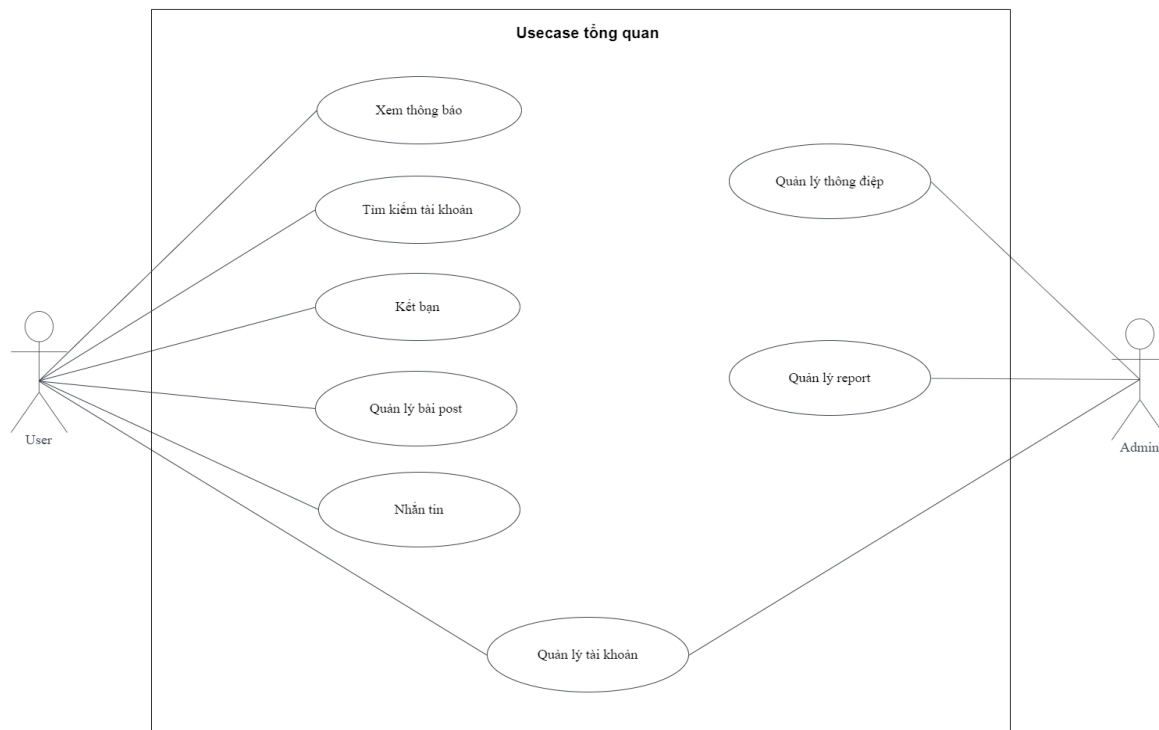
- Quản lý người dùng: admin có thể xem danh sách người dùng của hệ thống với trạng thái online, xem chi tiết thông tin người dùng và xóa người dùng vi phạm quy định
- Quản lý thông điệp: admin có thể xem danh sách các thông điệp đã gửi và tạo ra thông điệp tuần mới cho 12 cung hoàng đạo
- Quản lý report: admin có thể xem danh sách các bài viết cần giải quyết, kiểm tra nội dung bài viết bị report và quyết định có chấp nhận report hay không

2.1.2 User

- Matching: đây là chức năng nổi bật và cốt lõi của hệ thống. Thông qua chức năng này thì người dùng có thể kết bạn trên hệ thống một cách dễ dàng dựa vào độ tương thích cung hoàng đạo
- Message:
 - + Chat: chức năng nhắn tin giữa các người dùng. Sau khi được hệ thống matching thì người dùng có thể nhắn tin với nhau (có thể là text hoặc ảnh)
 - + Gửi file: chức năng cho phép người dùng gửi các tập tin
 - + Block chat: chức năng chặn người dùng trong cuộc trò chuyện. Khi sử dụng chức năng này thì cả hai không thể nhắn tin với nhau và chỉ có bên chặn mới có thể bỏ chặn
 - + Clear messages: chức năng xóa toàn bộ cuộc trò chuyện
 - + Search message: tìm kiếm tin nhắn trong cuộc trò chuyện
- My blog: trang cá nhân của người dùng. Người dùng có thể đăng bài viết mới, chỉnh sửa và xóa bài viết, bên cạnh đó có thể chỉnh sửa thông tin hồ sơ
- Search: chức năng tìm kiếm bạn bè. Người dùng có thể xem blog của những người được hệ thống hiển thị và báo cáo bài viết nếu không hợp lệ
- Settings: chức năng cho phép người dùng có thể thay đổi mật khẩu tài khoản và thông tin cá nhân

2.2. Biểu đồ ca sử dụng và đặc tả

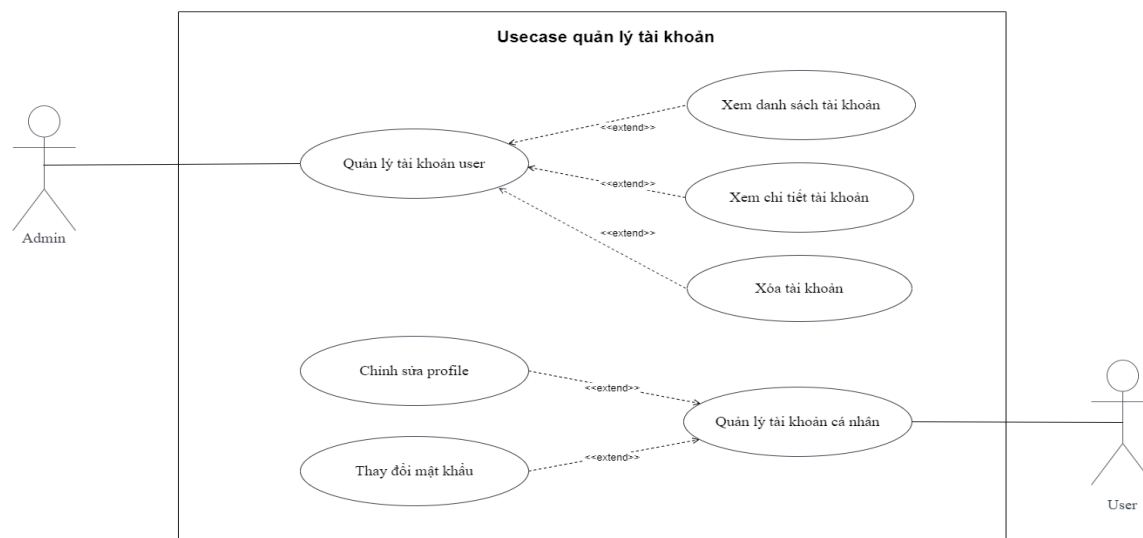
2.2.1 Usecase tổng quan



Hình 2 Sơ đồ usecase tổng quan

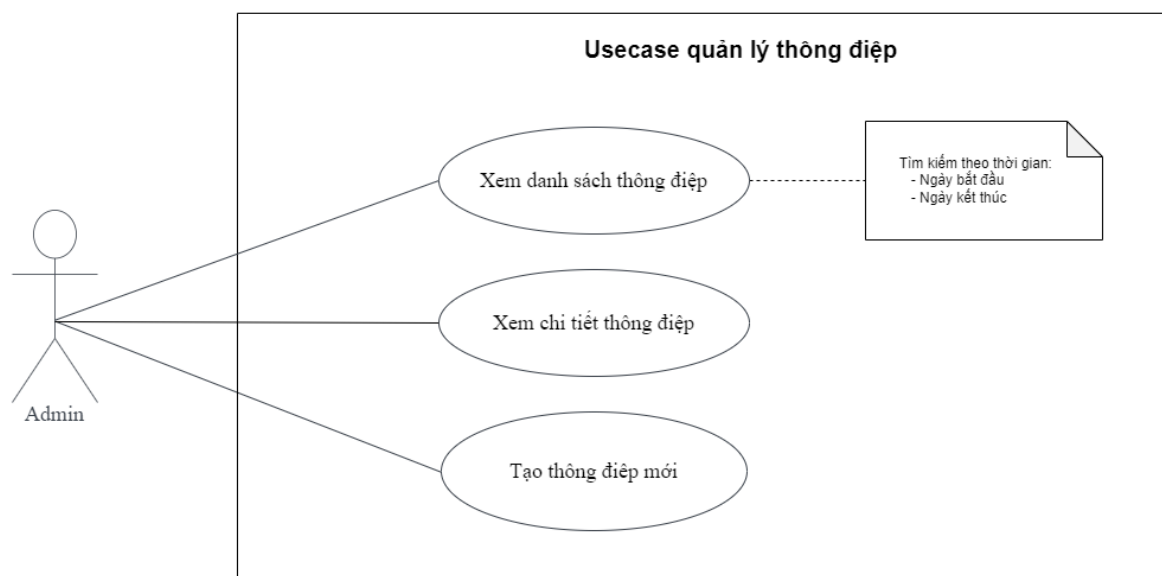
2.2.2 Phân rã usecase

a) Usecase quản lý tài khoản



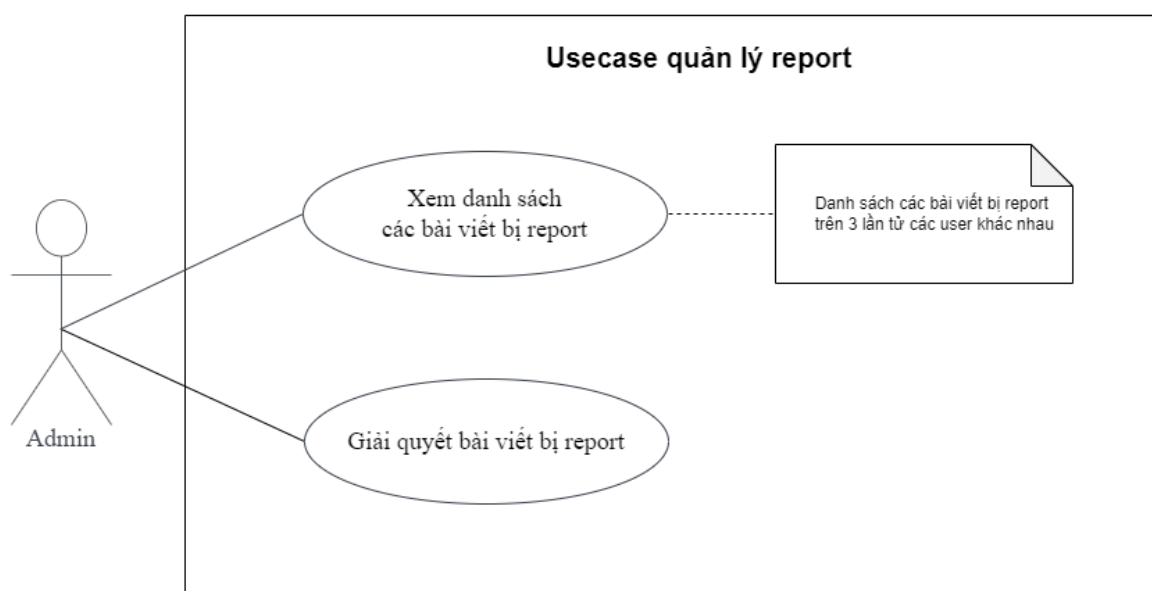
Hình 3 Sơ đồ usecase quản lý tài khoản

b) Usecase quản lý thông điệp



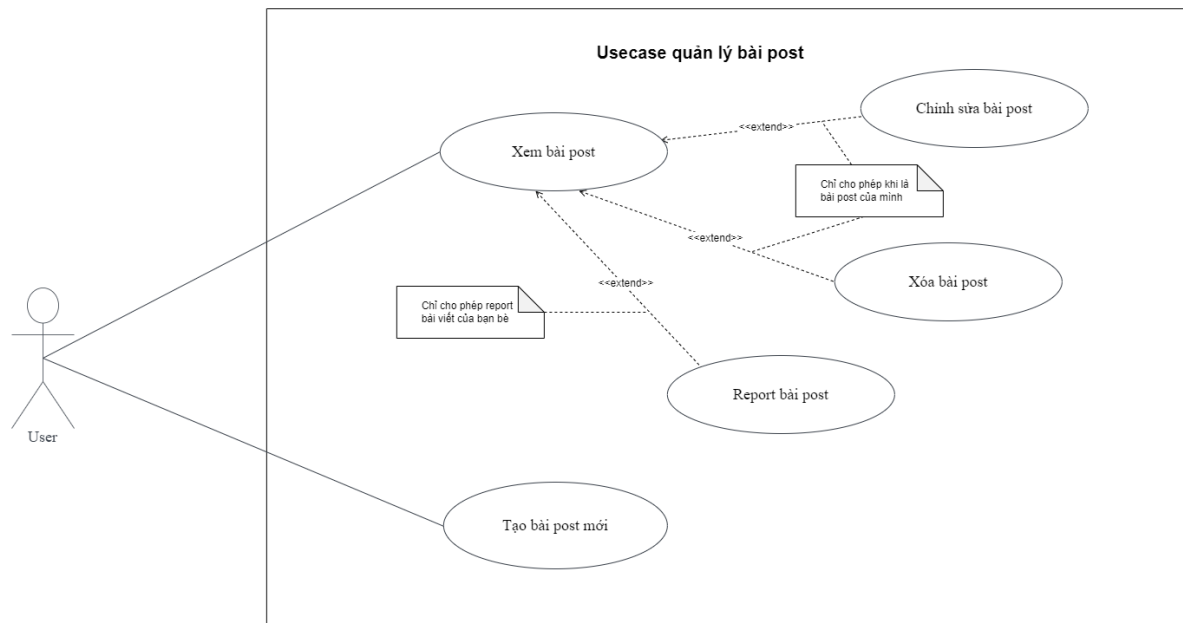
Hình 4 Sơ đồ usecase quản lý thông điệp

c) Usecase quản lý report



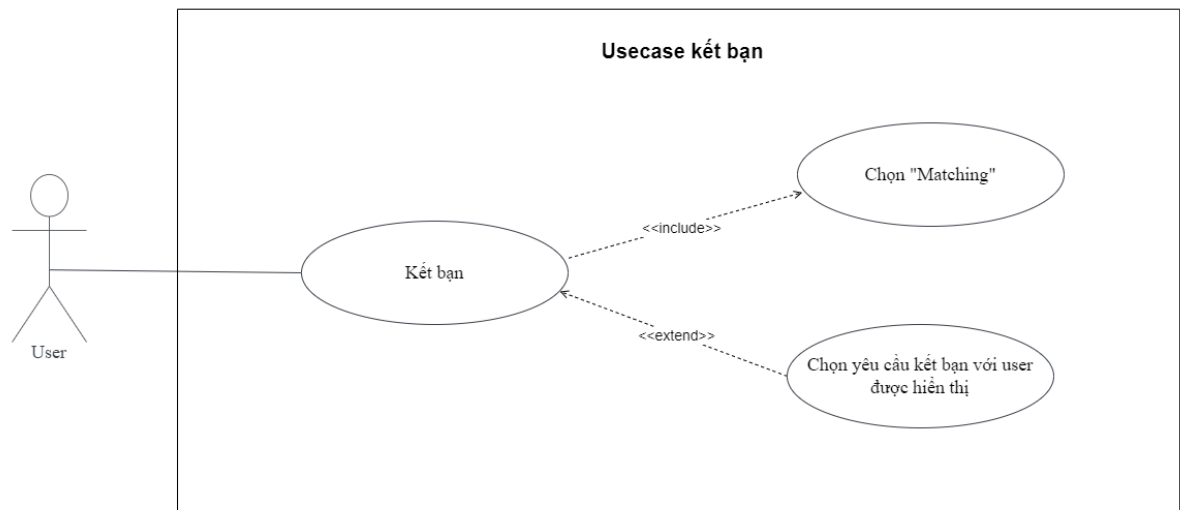
Hình 5 Sơ đồ usecase quản lý report

d) Usecase quản lý bài post



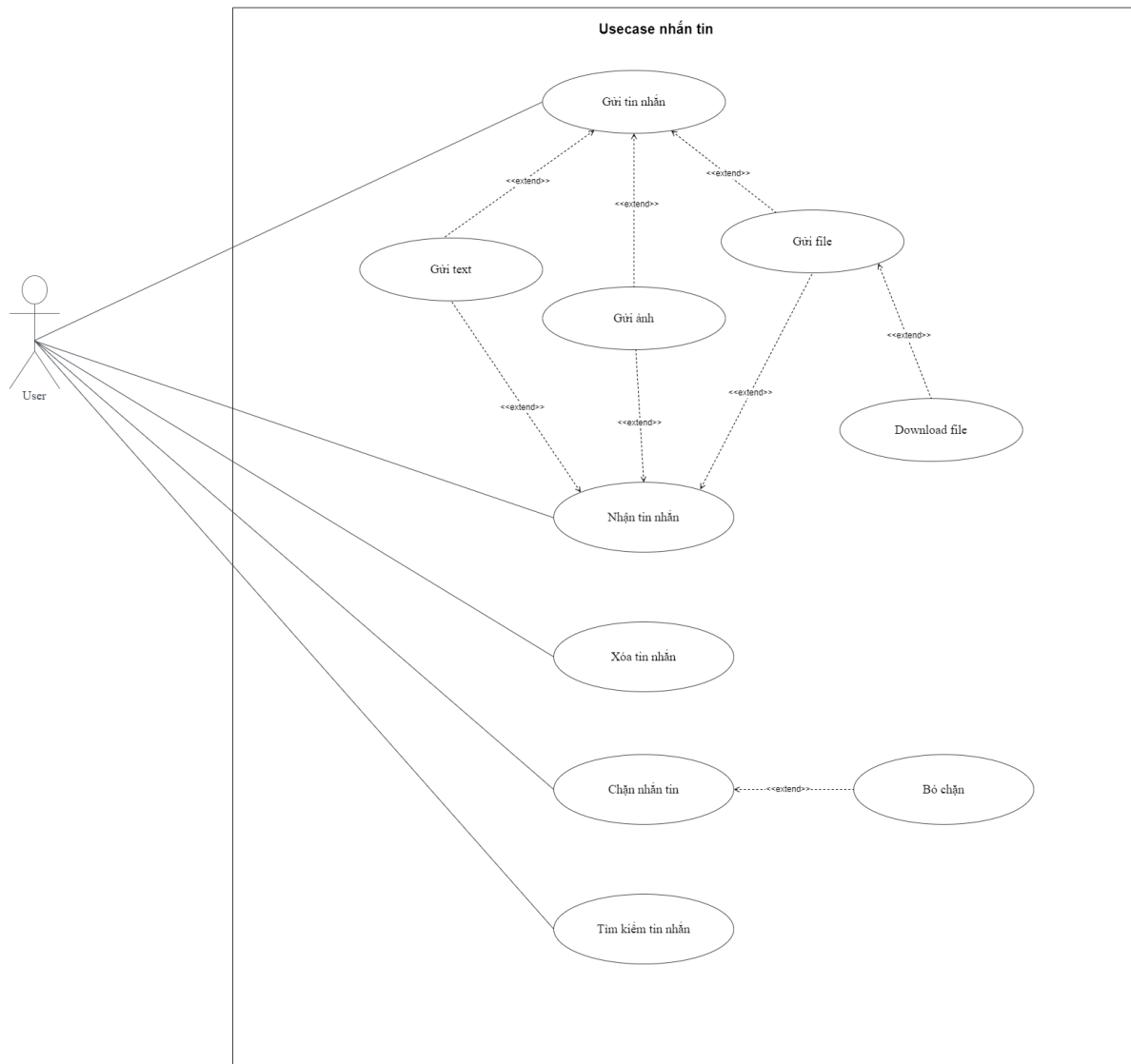
Hình 6 Sơ đồ usecase quản lý bài post

e) Usecase kết bạn



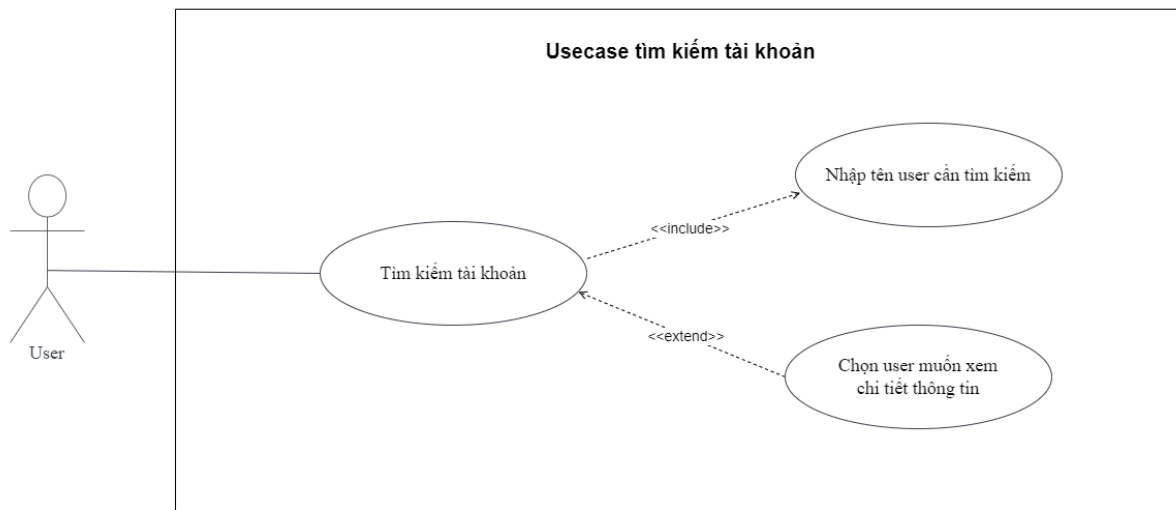
Hình 7 Sơ đồ usecase kết bạn (matching)

f) Usecase nhắn tin



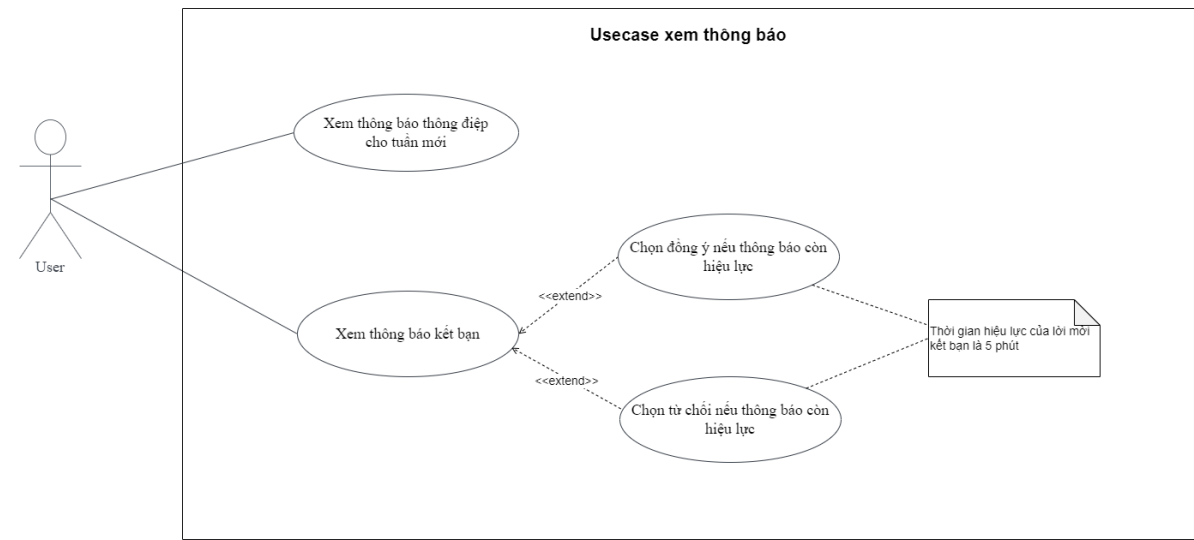
Hình 8 Sơ đồ usecase nhắn tin

g) Usecase tìm kiếm tài khoản



Hình 9 Sơ đồ usecase tìm kiếm tài khoản

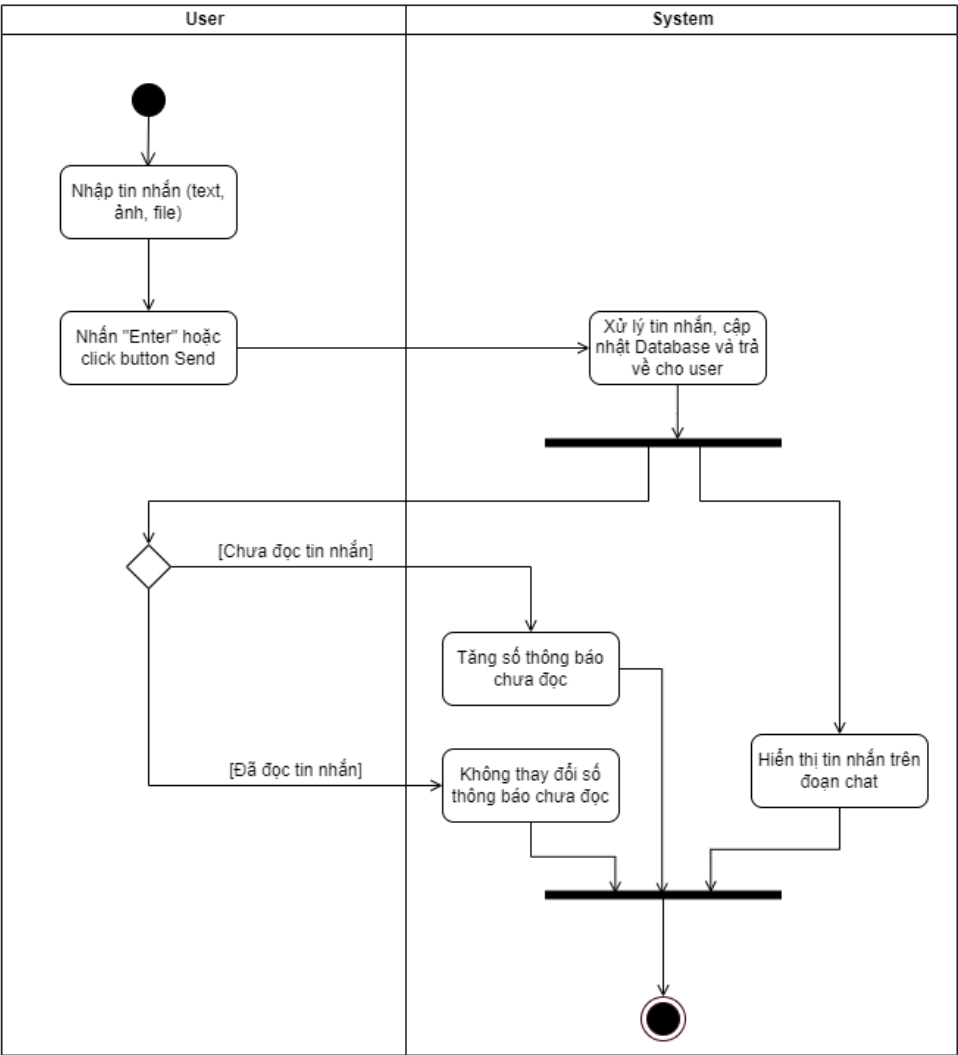
h) Usecase xem thông báo



Hình 10 Sơ đồ usecase xem thông báo

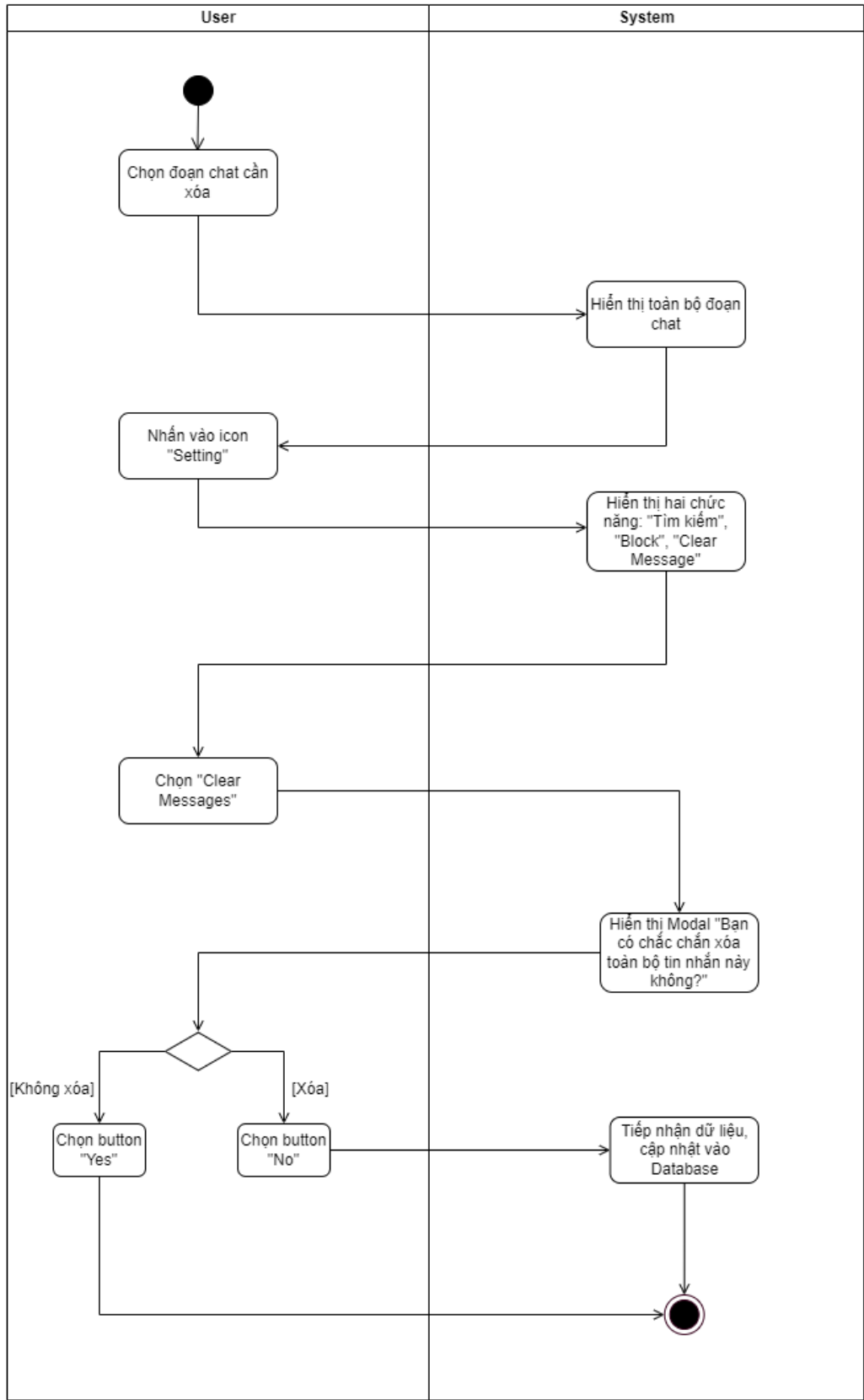
2.3. Biểu đồ hoạt động

2.3.1. Biểu đồ hoạt động nhắn tin



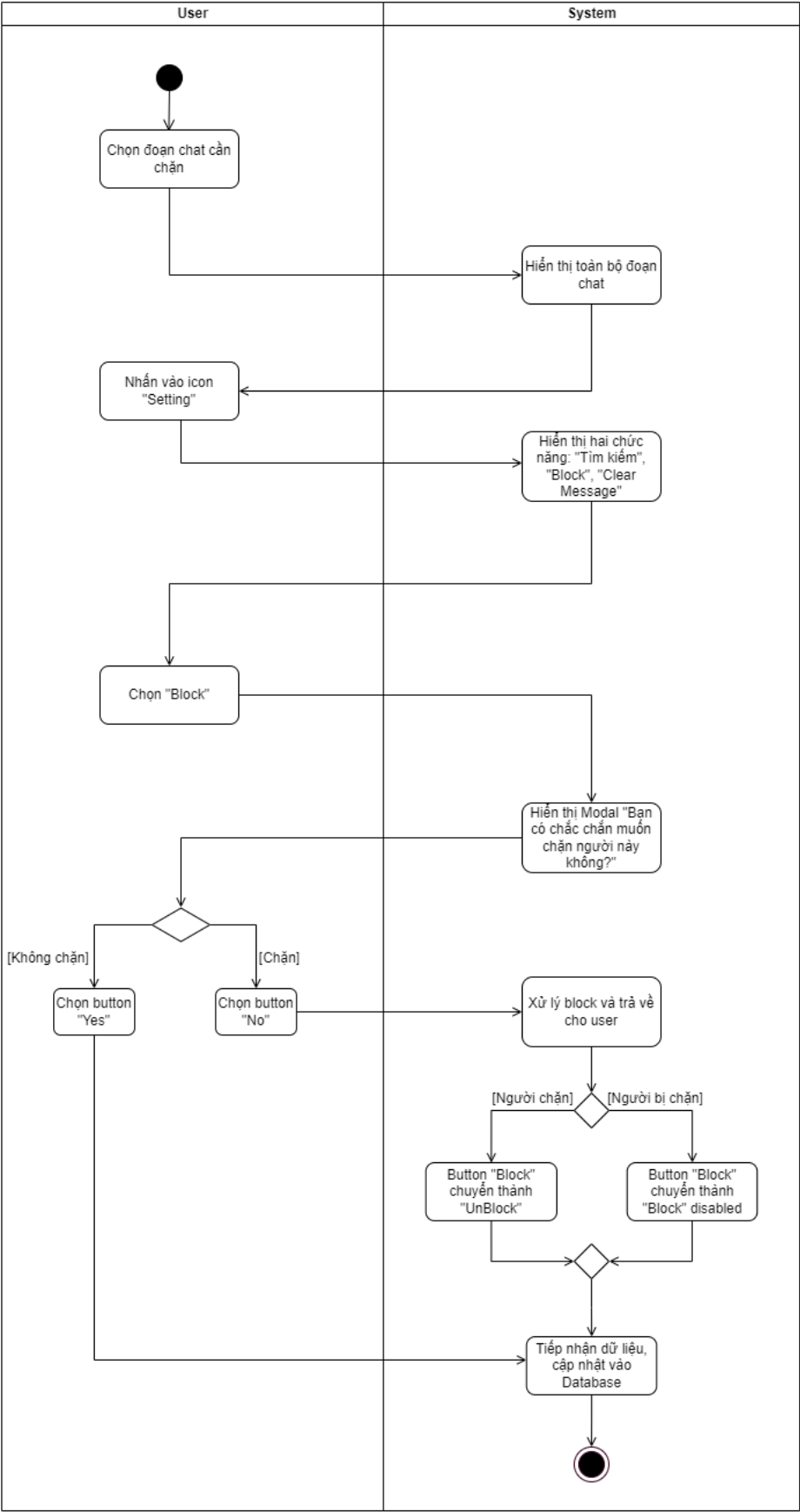
Bảng 1 Biểu đồ hoạt động nhắn tin

2.3.2. Biểu đồ hoạt động xóa cuộc trò chuyện



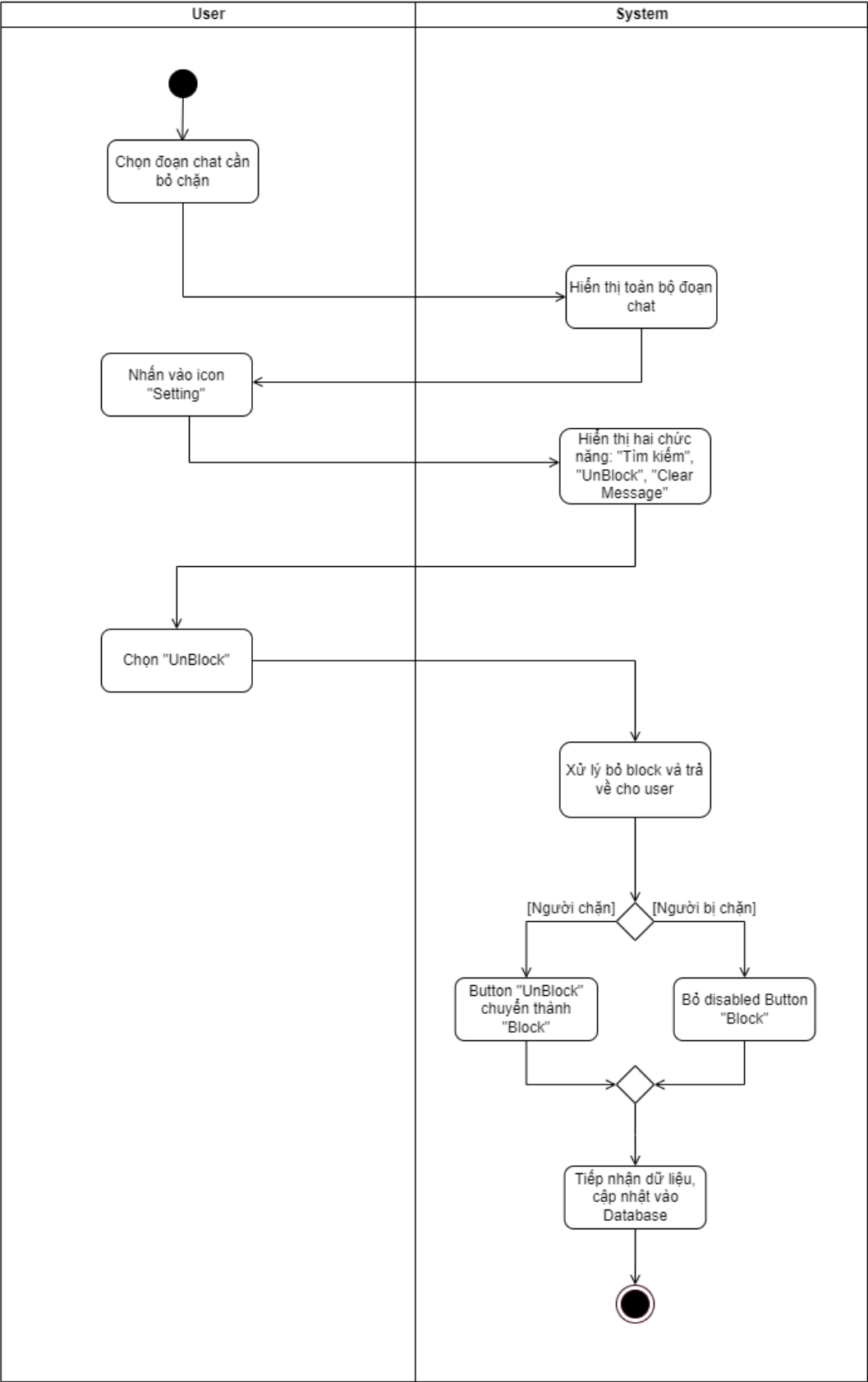
Bảng 2 Biểu đồ hoạt động xóa cuộc trò chuyện

2.3.3. Biểu đồ hoạt động chặn nhắn tin



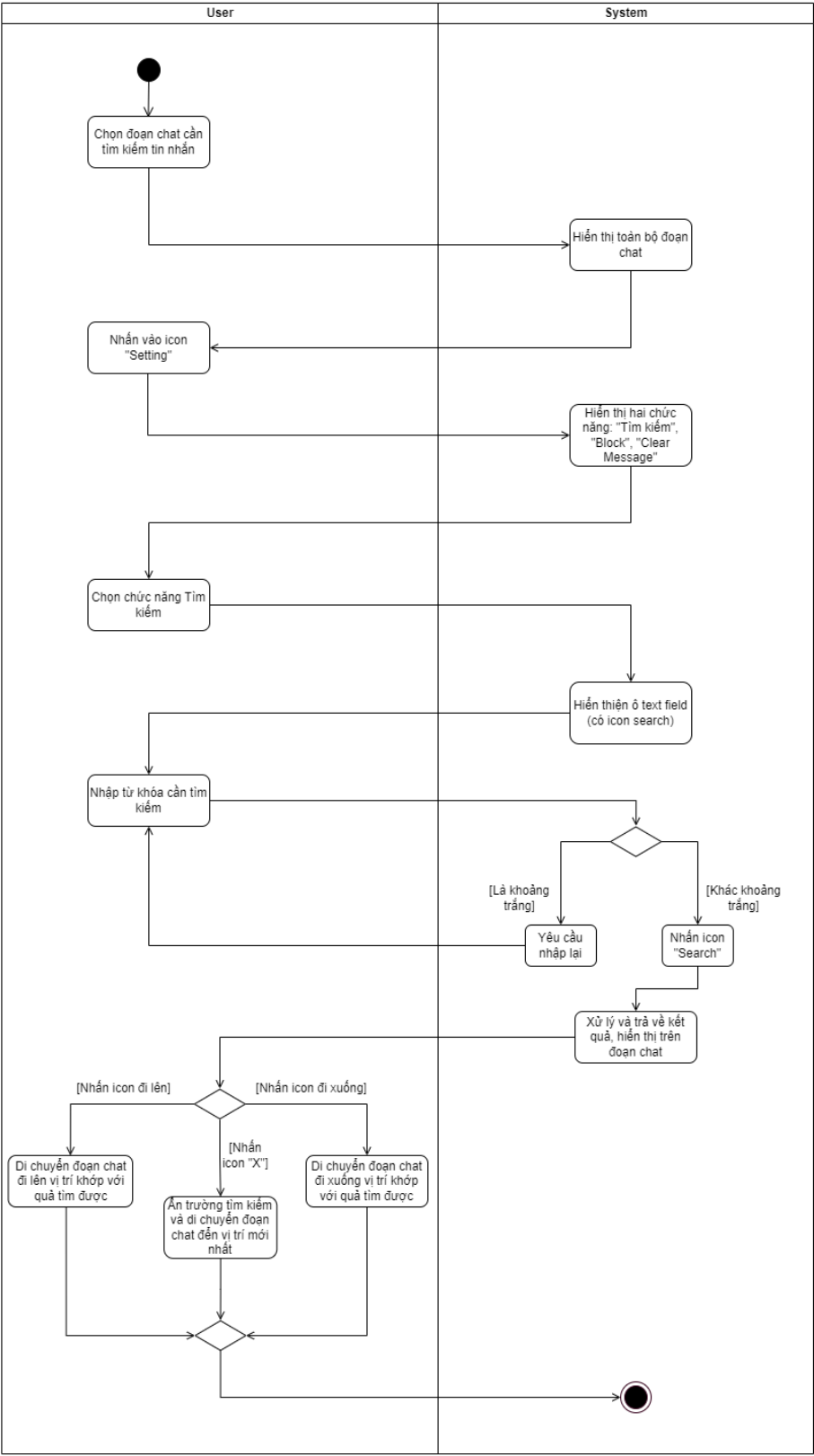
Bảng 3 Biểu đồ hoạt động nhắn tin

2.3.4. Biểu đồ hoạt động bỏ chặn nhắn tin



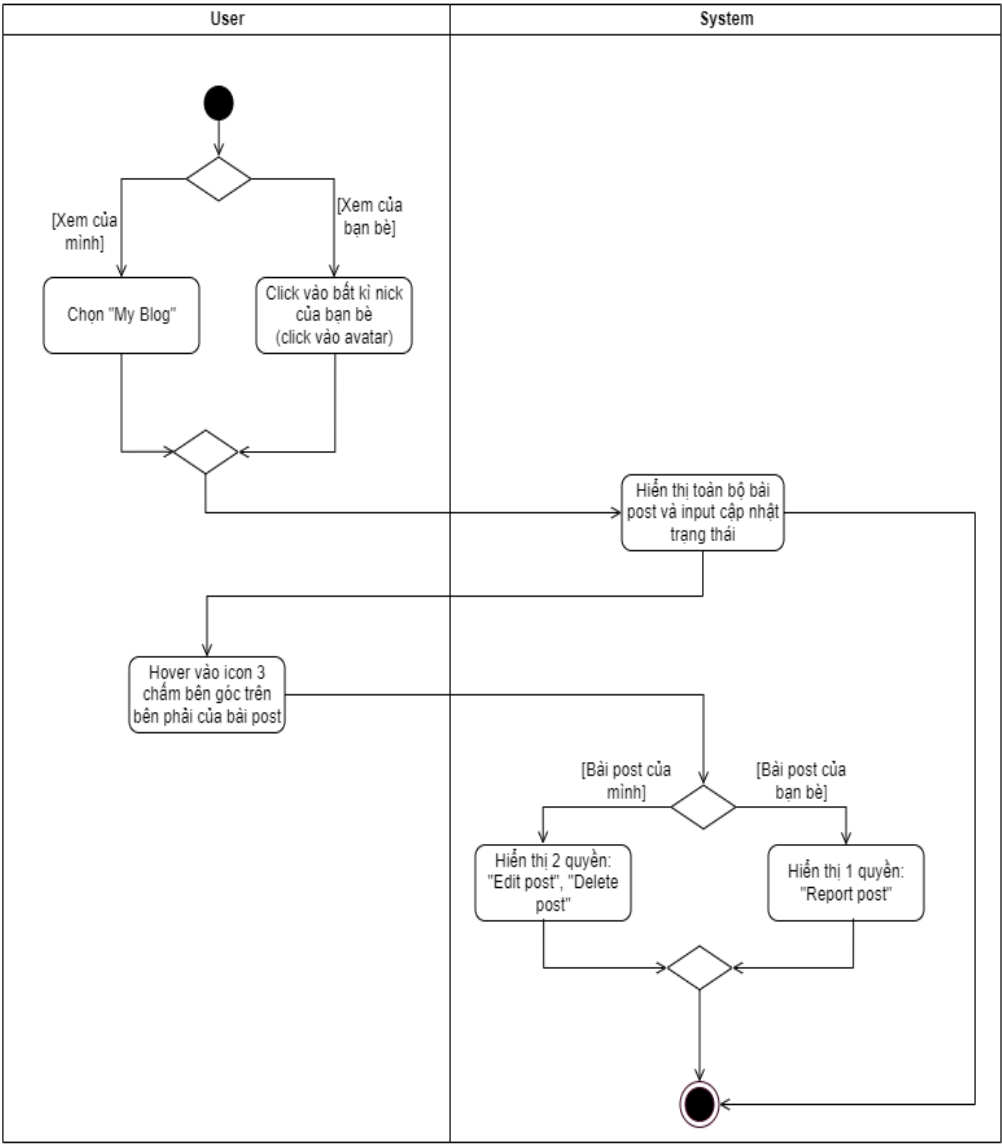
Bảng 4 Biểu đồ hoạt động bỏ chặn nhắn tin

2.3.5. Biểu đồ hoạt động tìm kiếm tin nhắn



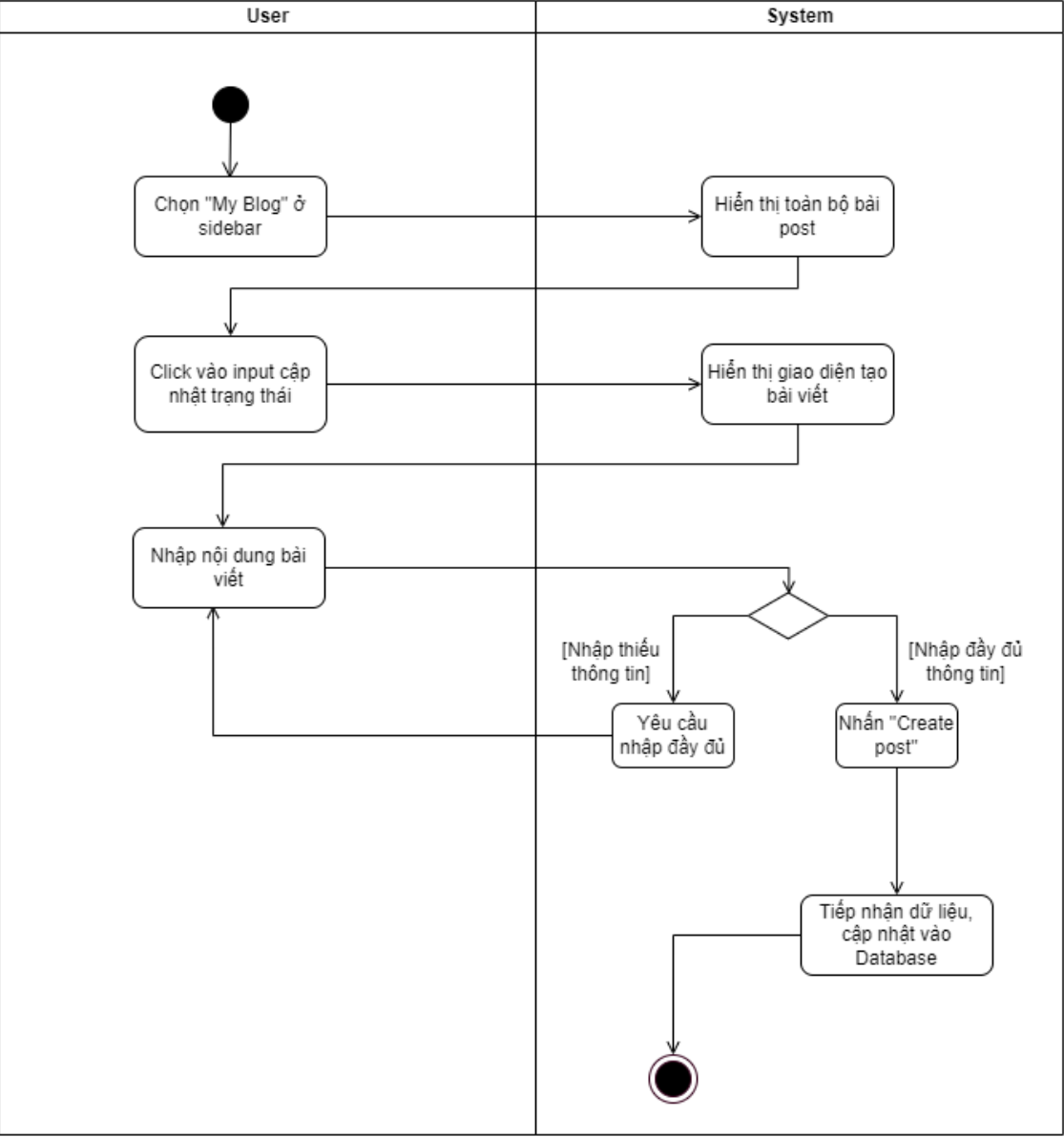
Bảng 5 Biểu đồ hoạt động tìm kiếm tin nhắn

2.3.6. Biểu đồ hoạt động hiển thị toàn bộ bài post



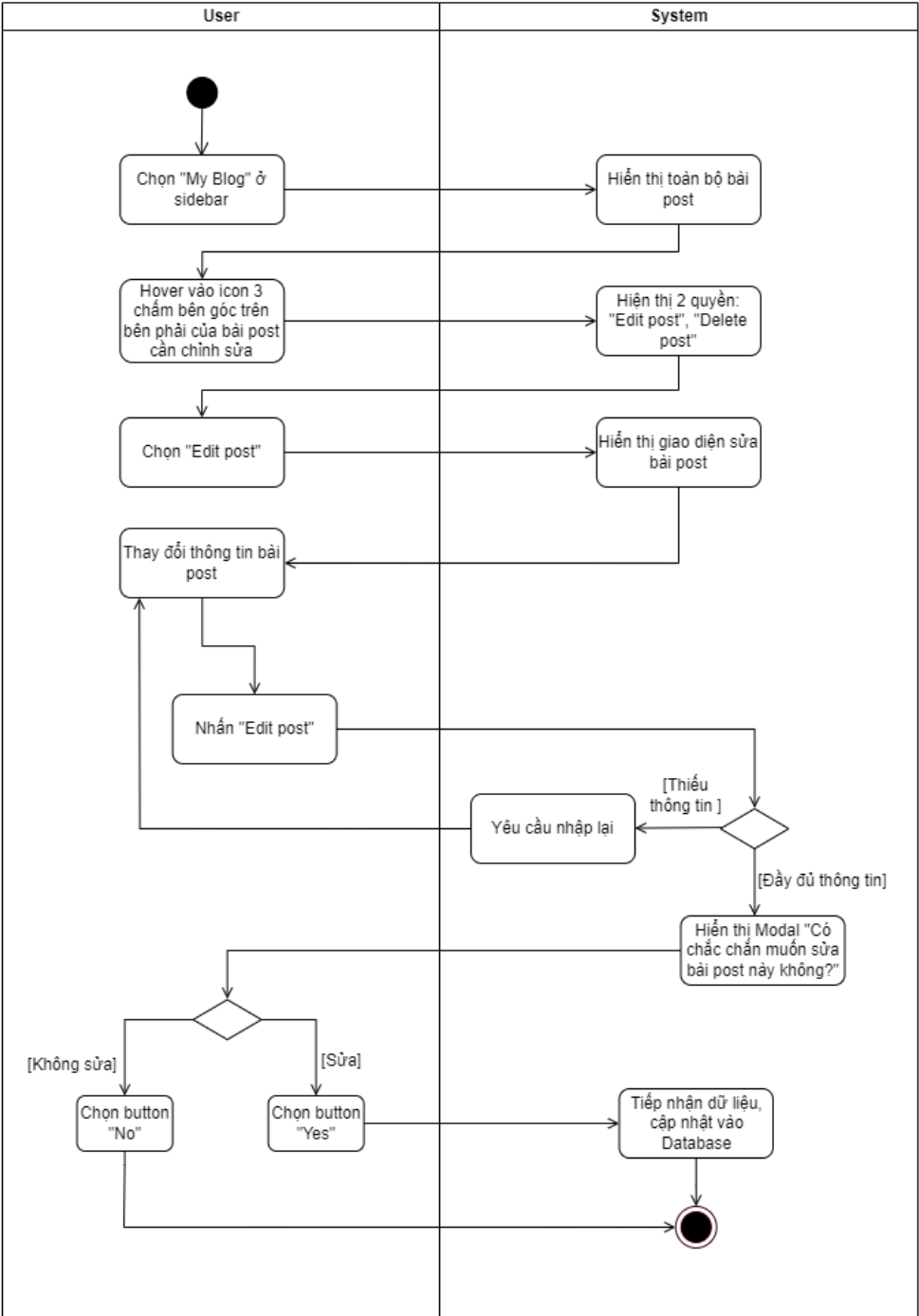
Bảng 6 Biểu đồ hoạt động hiển thị toàn bộ bài post

2.3.7. Biểu đồ hoạt động thêm bài post mới



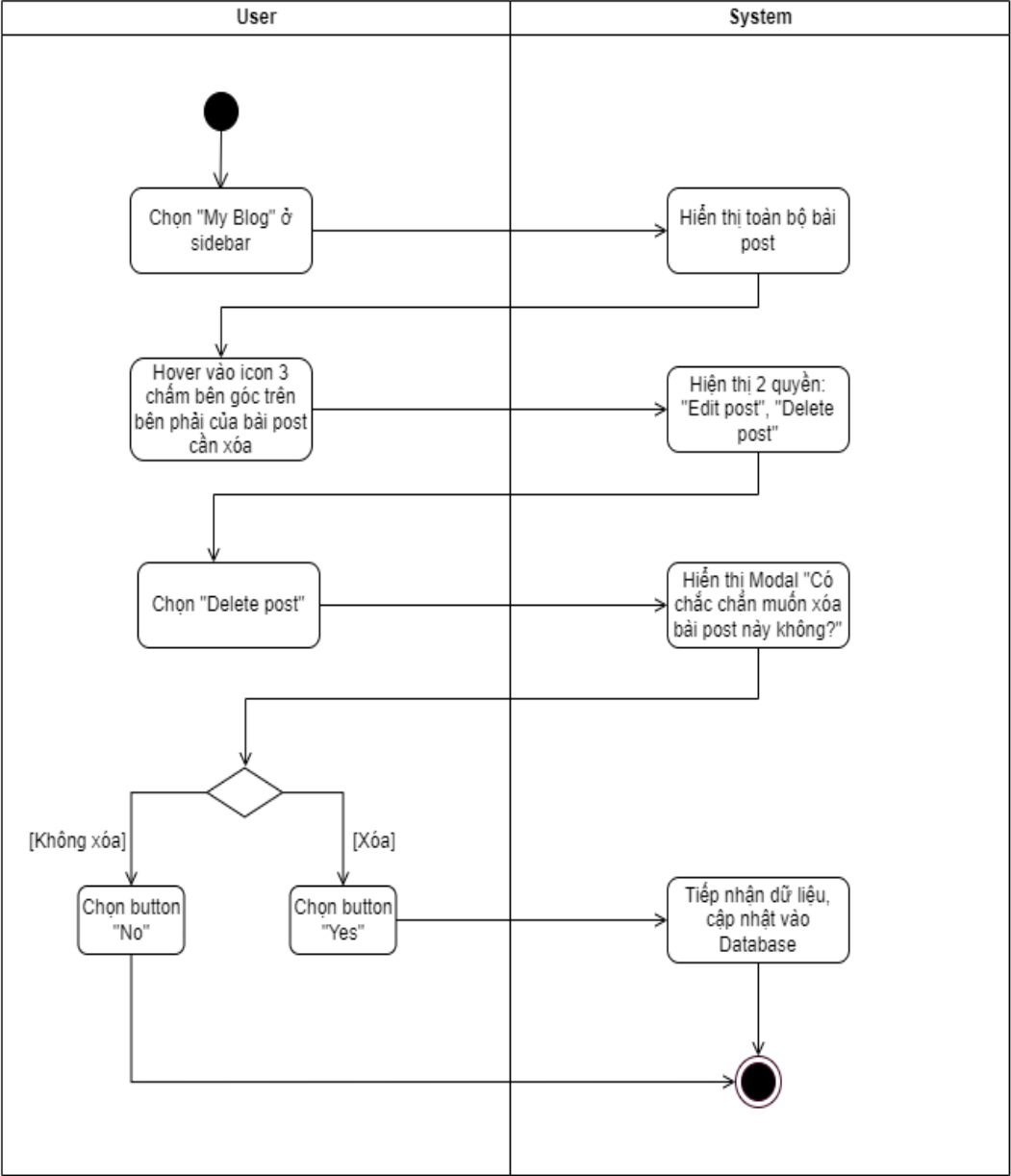
Bảng 7 Biểu đồ hoạt động thêm bài post mới

2.3.8. Biểu đồ hoạt động chỉnh sửa bài post



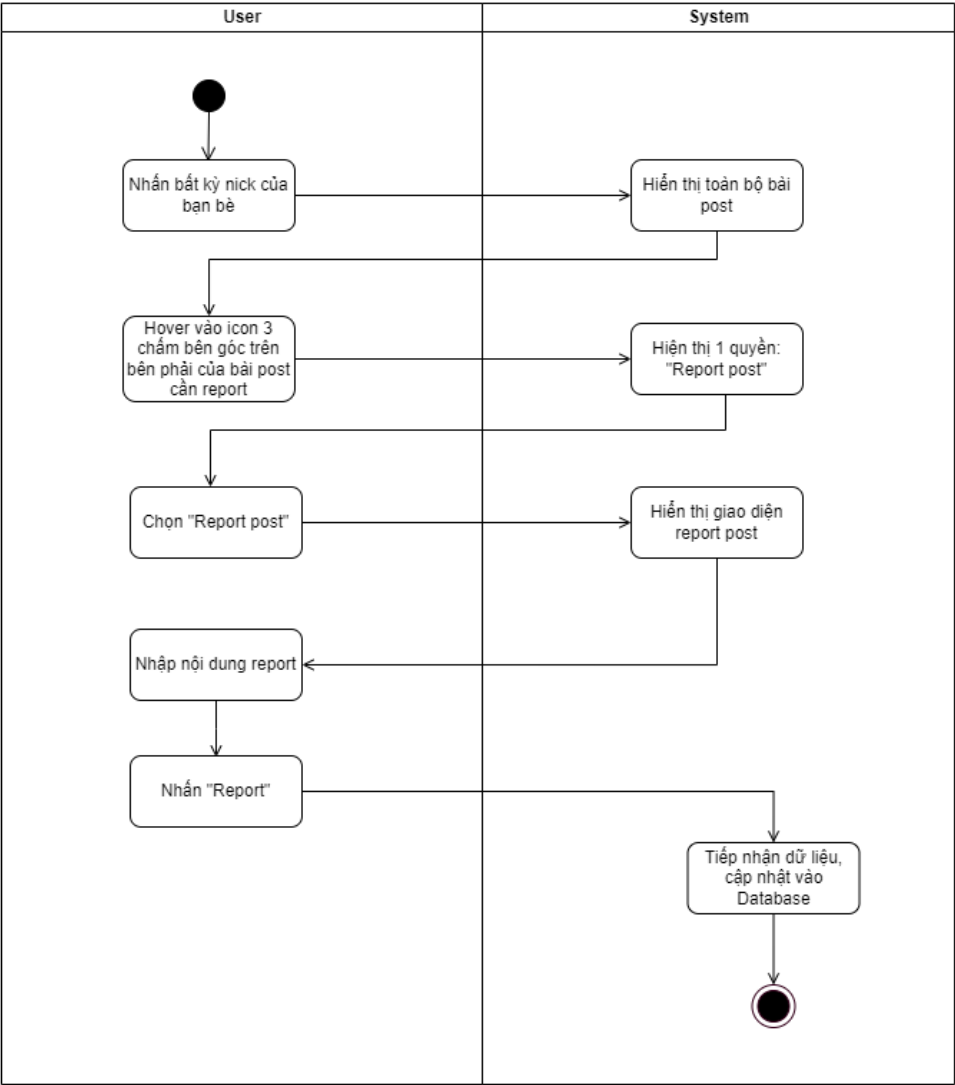
Bảng 8 Biểu đồ hoạt động chỉnh sửa bài post

2.3.9. Biểu đồ hoạt động xóa bài post



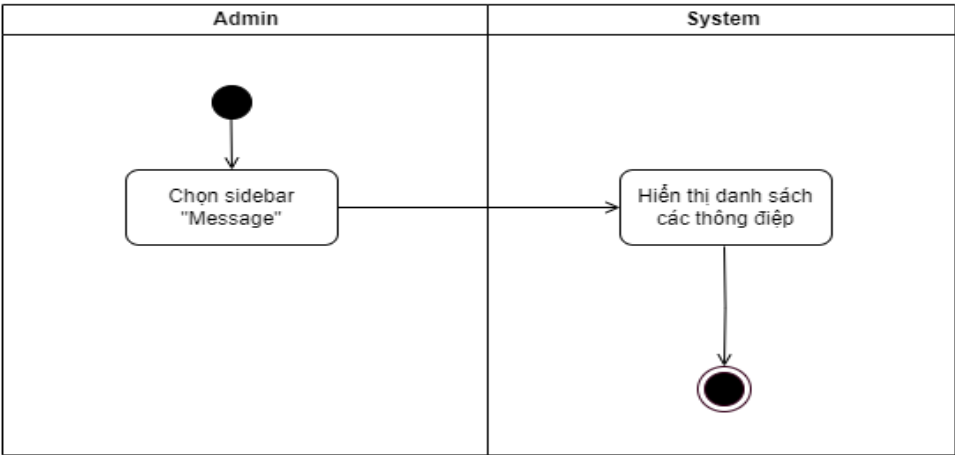
Bảng 9 Biểu đồ hoạt động xóa bài post

2.3.10. Biểu đồ hoạt động report bài post



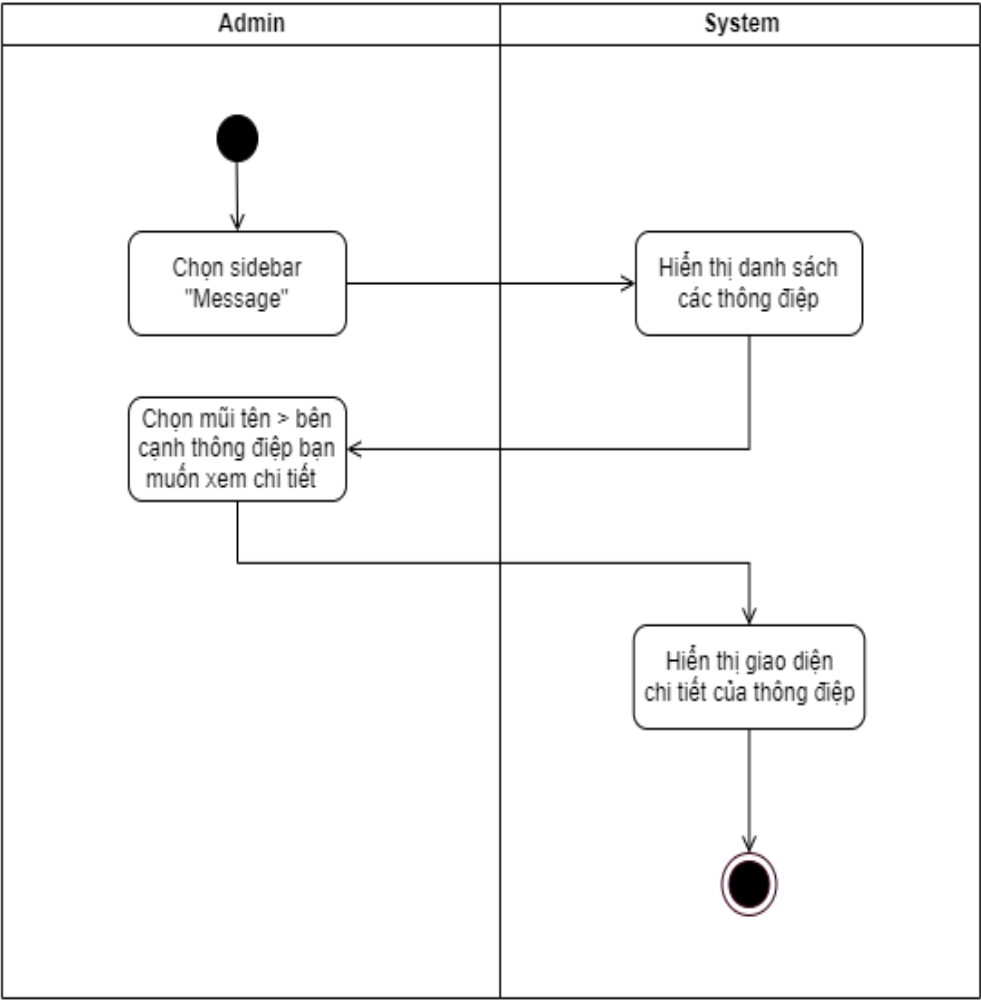
Bảng 10 Biểu đồ hoạt động report bài post

2.3.11. Biểu đồ hoạt động xem danh sách thông điệp



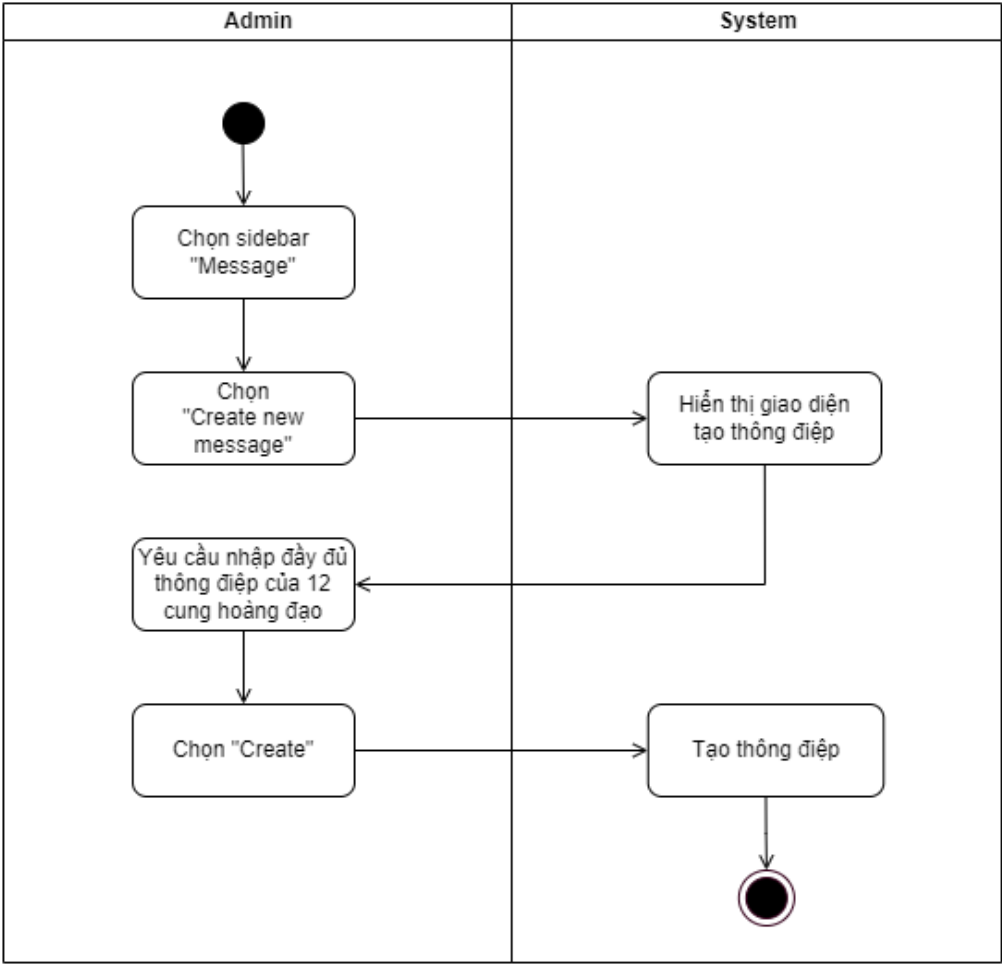
Bảng 11 Biểu đồ hoạt động xem danh sách thông điệp

2.3.12. Biểu đồ hoạt động xem chi tiết thông điệp



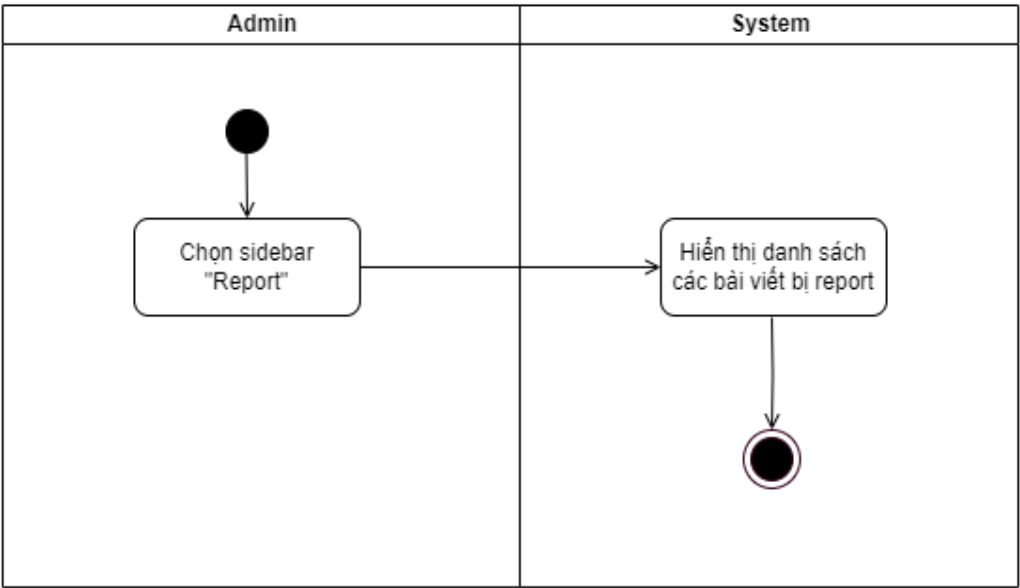
Bảng 12 Biểu đồ hoạt động xem chi tiết thông điệp

2.3.13. Biểu đồ hoạt động tạo mới thông điệp



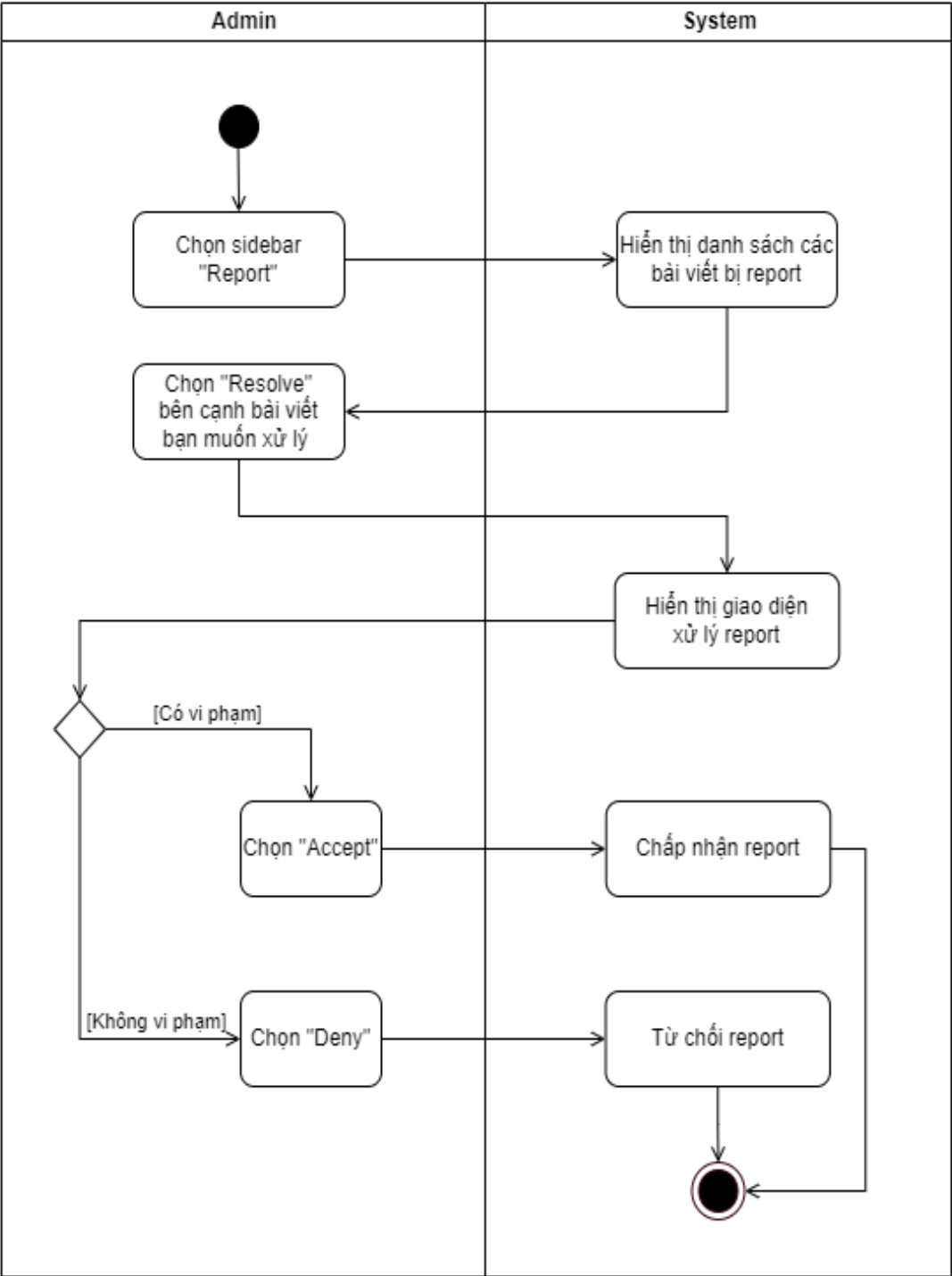
Bảng 13 Biểu đồ hoạt động tạo mới thông điệp

2.3.14. Biểu đồ hoạt động xem danh sách bài viết bị report



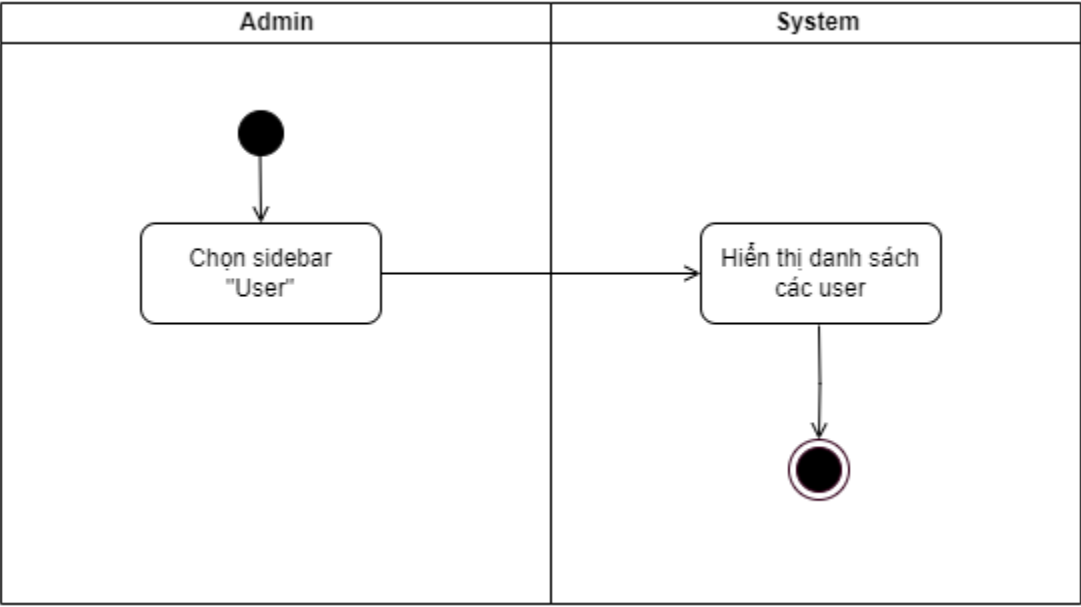
Bảng 14 Biểu đồ hoạt động xem danh sách bài viết bị report

2.3.15. Biểu đồ hoạt động giải quyết bài viết bị report



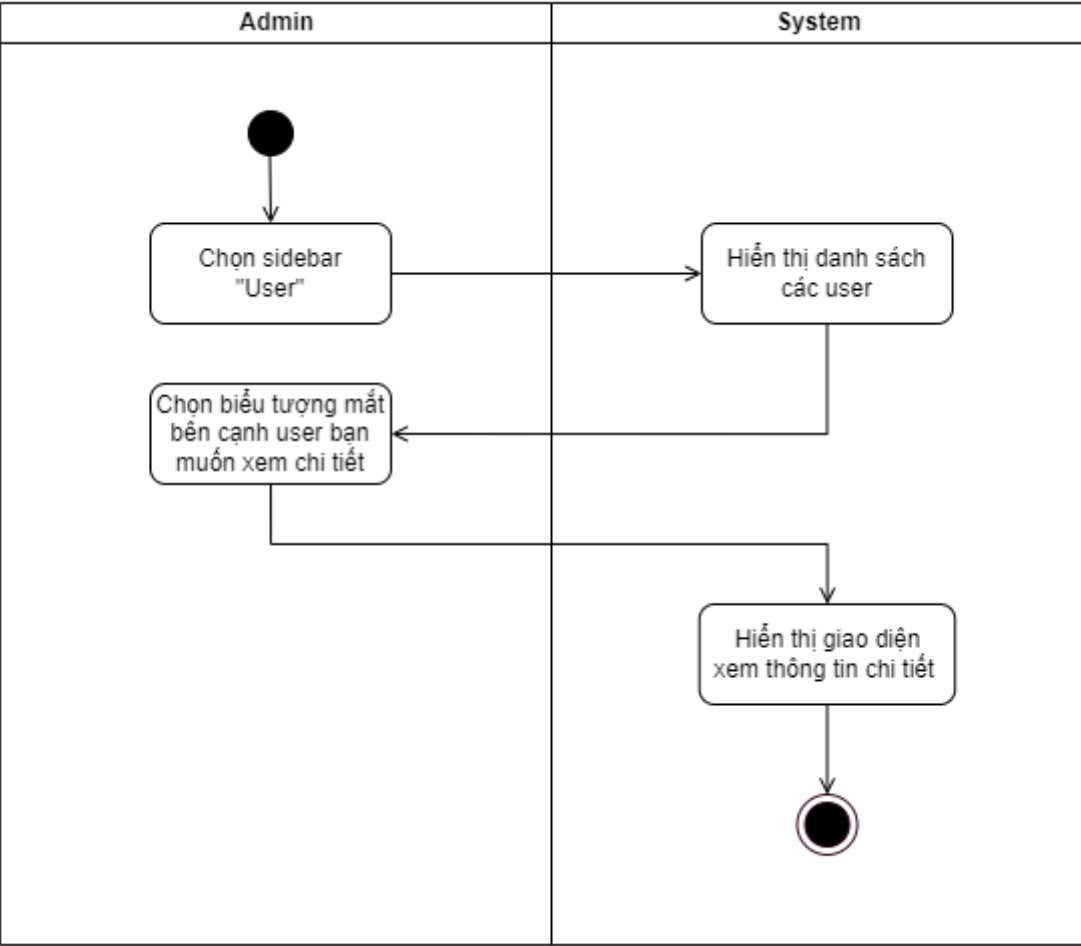
Bảng 15 Biểu đồ hoạt động giải quyết bài viết bị report

2.3.16. Biểu đồ hoạt động xem danh sách tài khoản



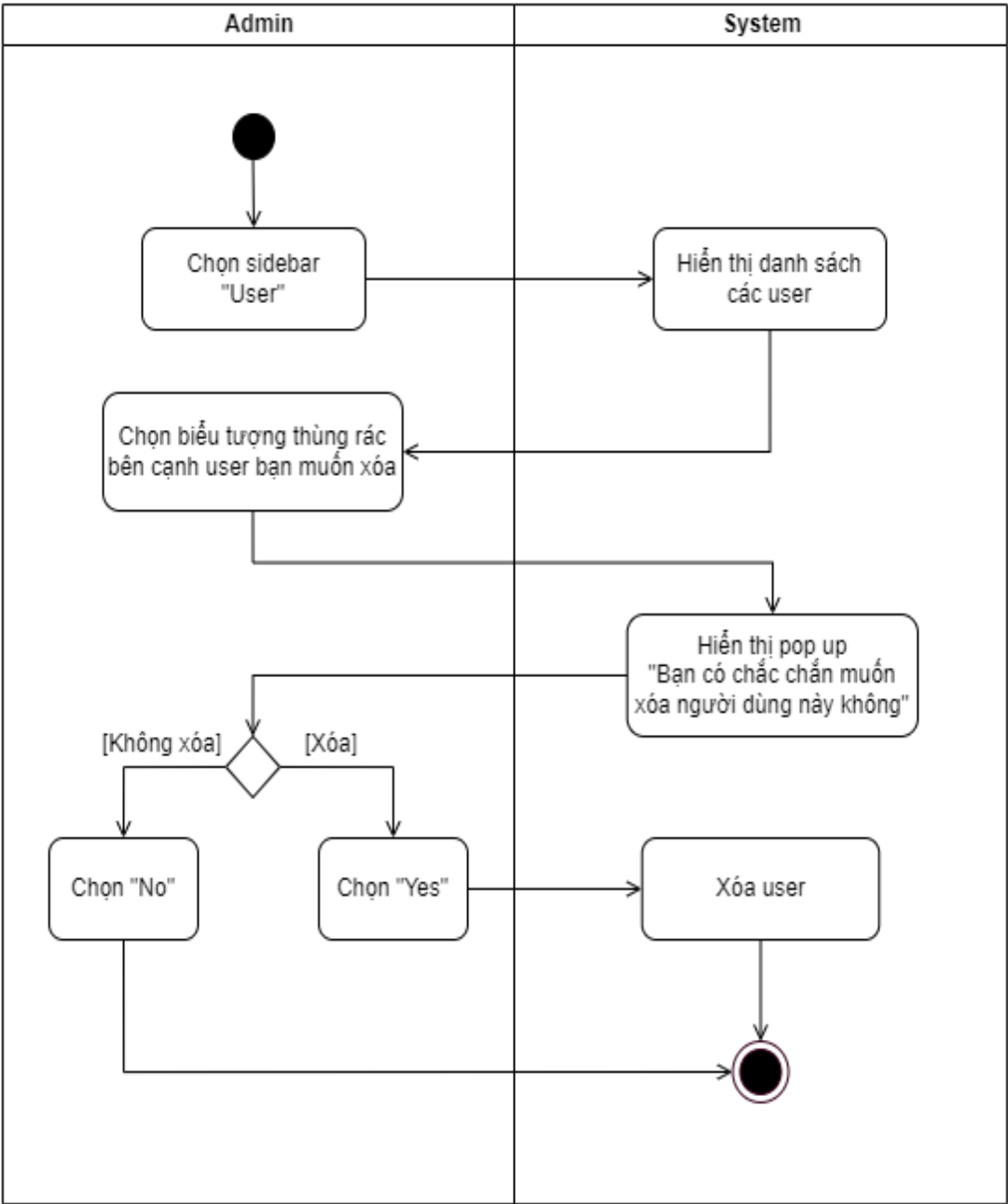
Bảng 16 Biểu đồ hoạt động xem danh sách tài khoản

2.3.17. Biểu đồ hoạt động xem chi tiết tài khoản



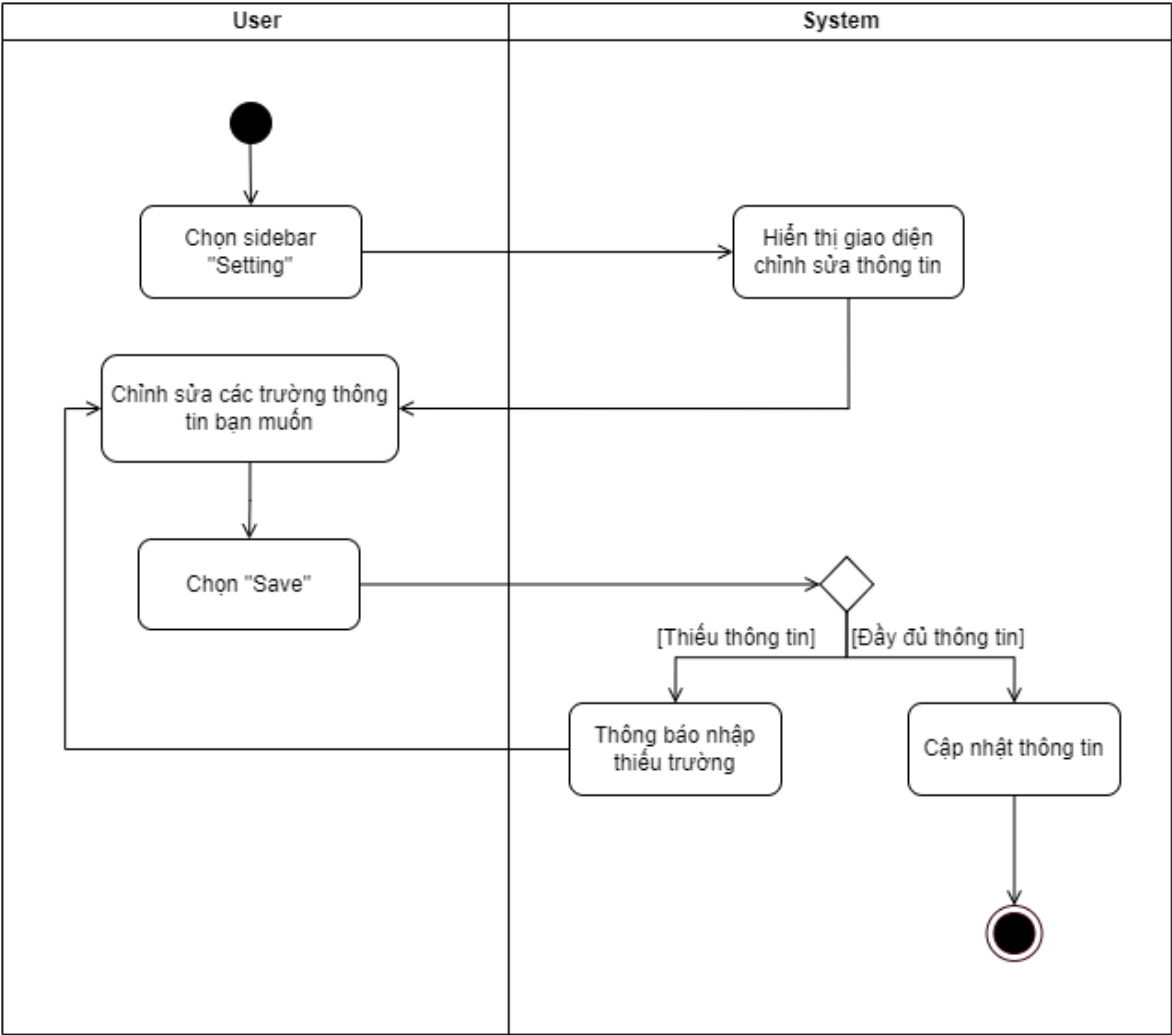
Bảng 17 Biểu đồ hoạt động xem chi tiết tài khoản

2.3.18. Biểu đồ hoạt động xóa tài khoản



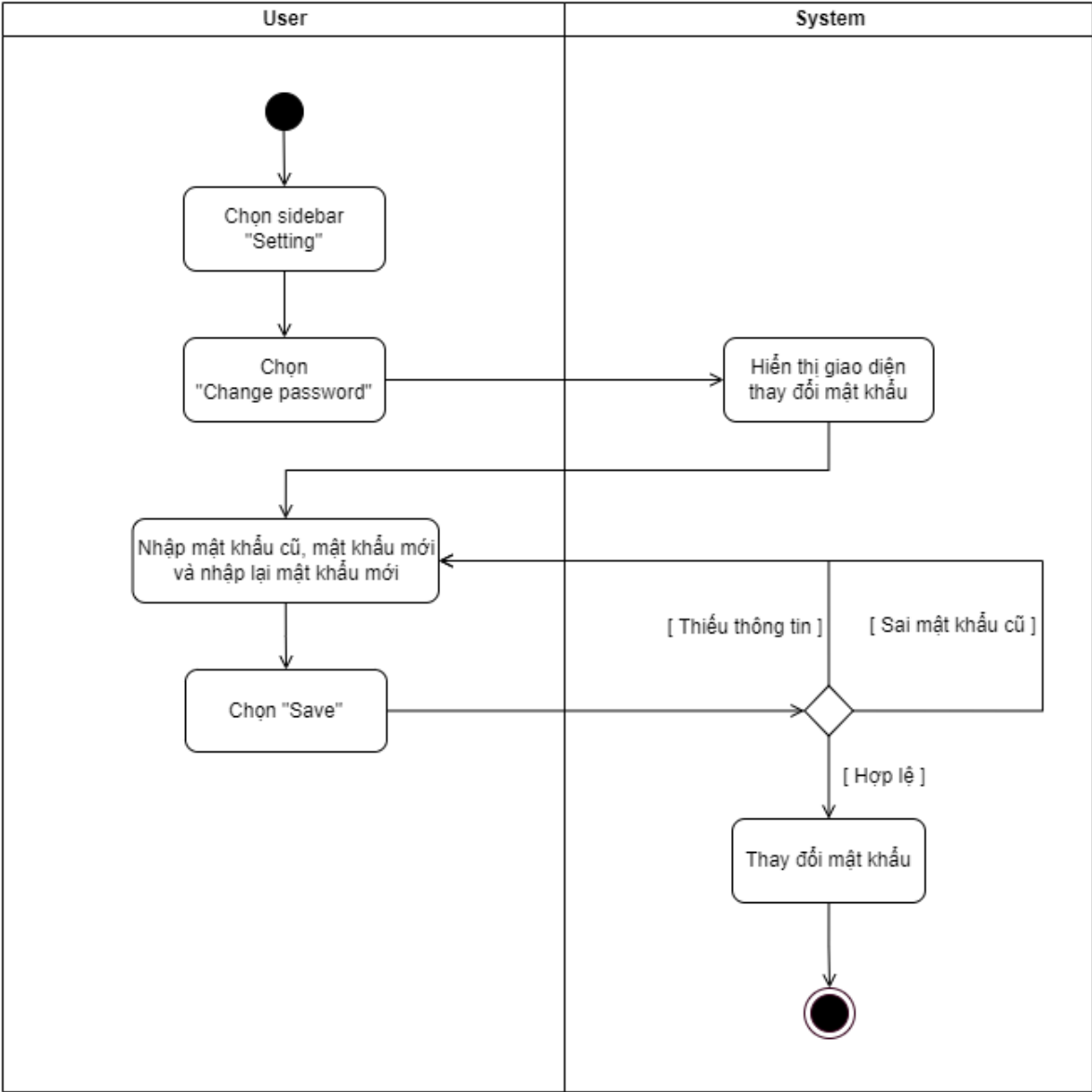
Bảng 18 Biểu đồ hoạt động xóa tài khoản

2.3.19. Biểu đồ hoạt động chỉnh sửa profile



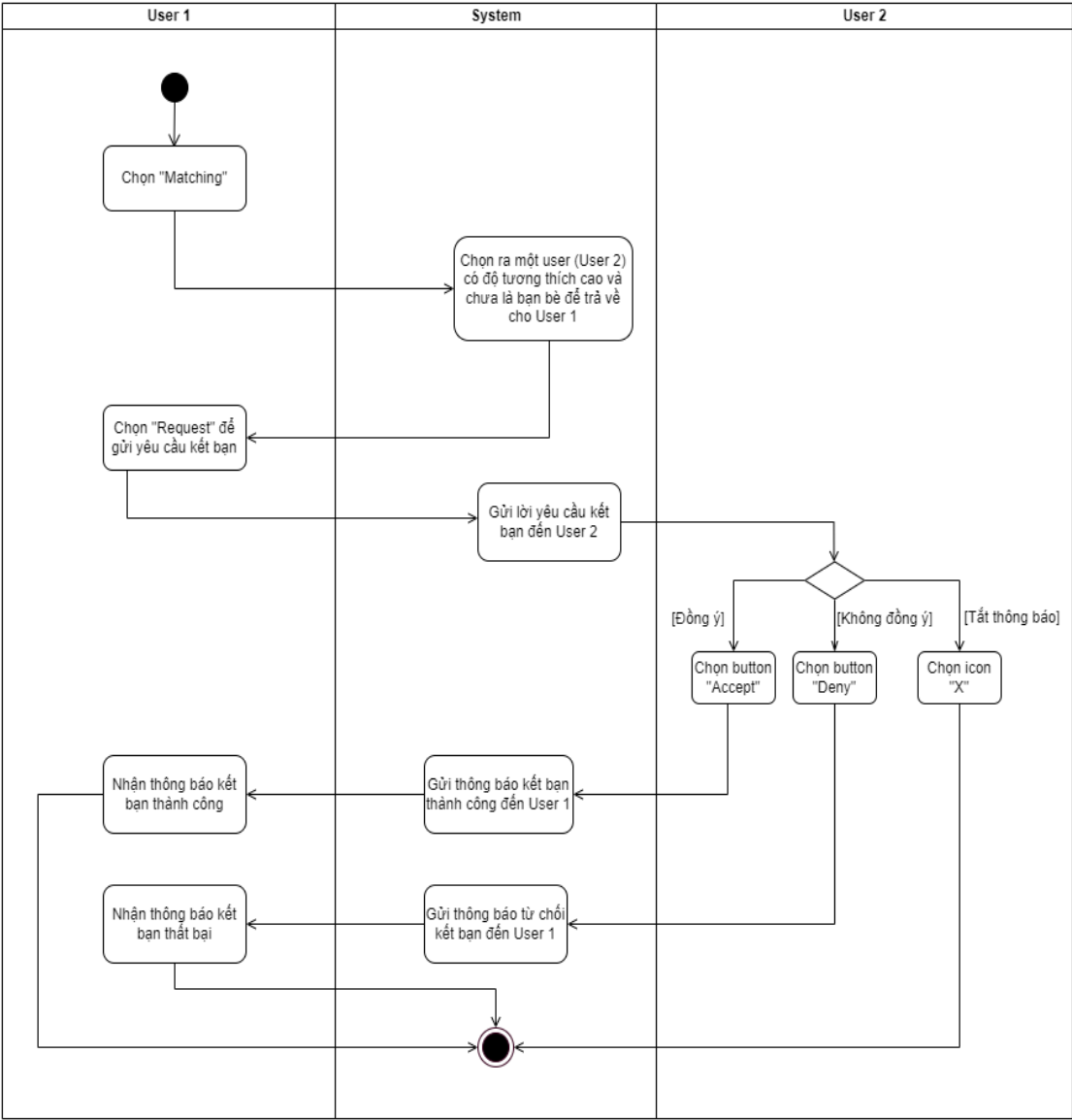
Bảng 19 Biểu đồ hoạt động chỉnh sửa profile

2.3.20. Biểu đồ hoạt động thay đổi mật khẩu



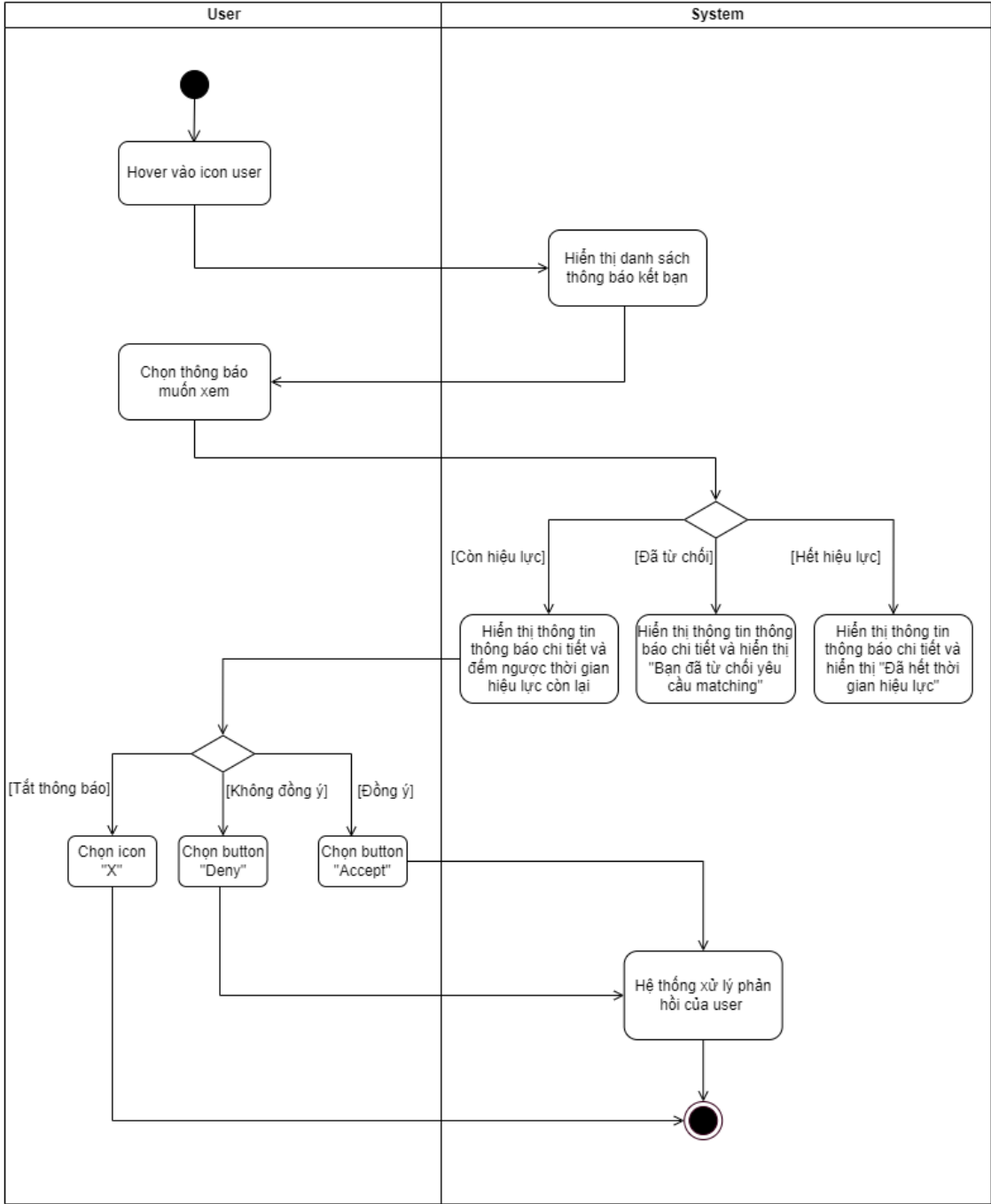
Bảng 20 Biểu đồ hoạt động thay đổi mật khẩu

2.3.21. Biểu đồ hoạt động kết bạn



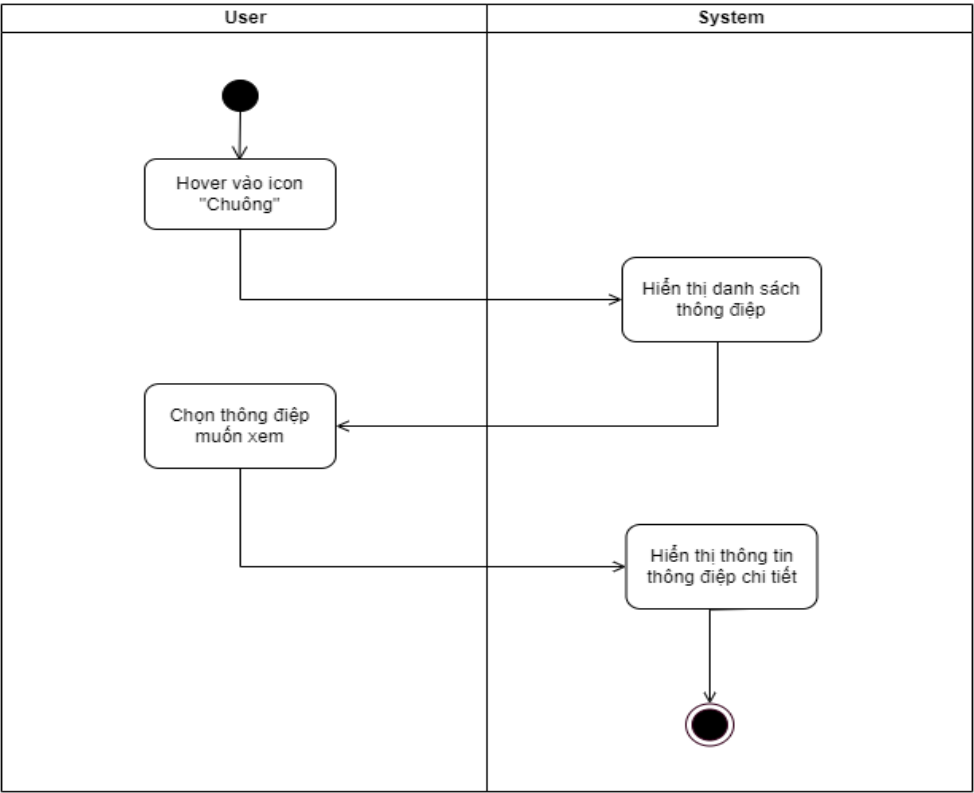
Bảng 21 Biểu đồ hoạt động kết bạn

2.3.22. Biểu đồ hoạt động xem thông báo matching



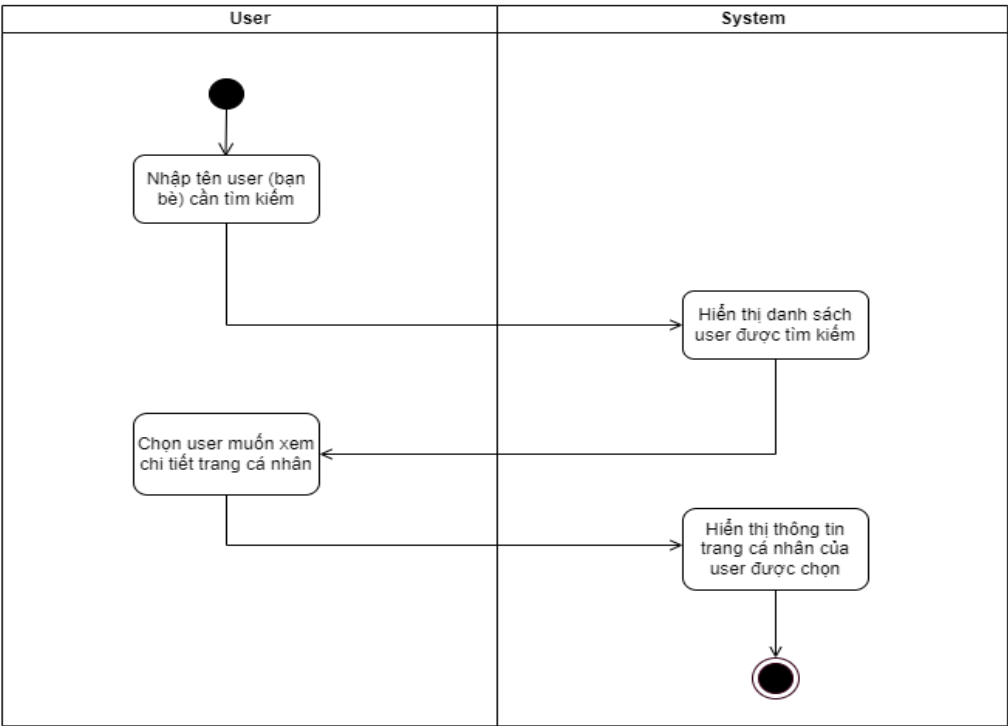
Bảng 22 Biểu đồ hoạt động xem thông báo matching

2.3.23. Biểu đồ hoạt động xem thông báo thông điệp



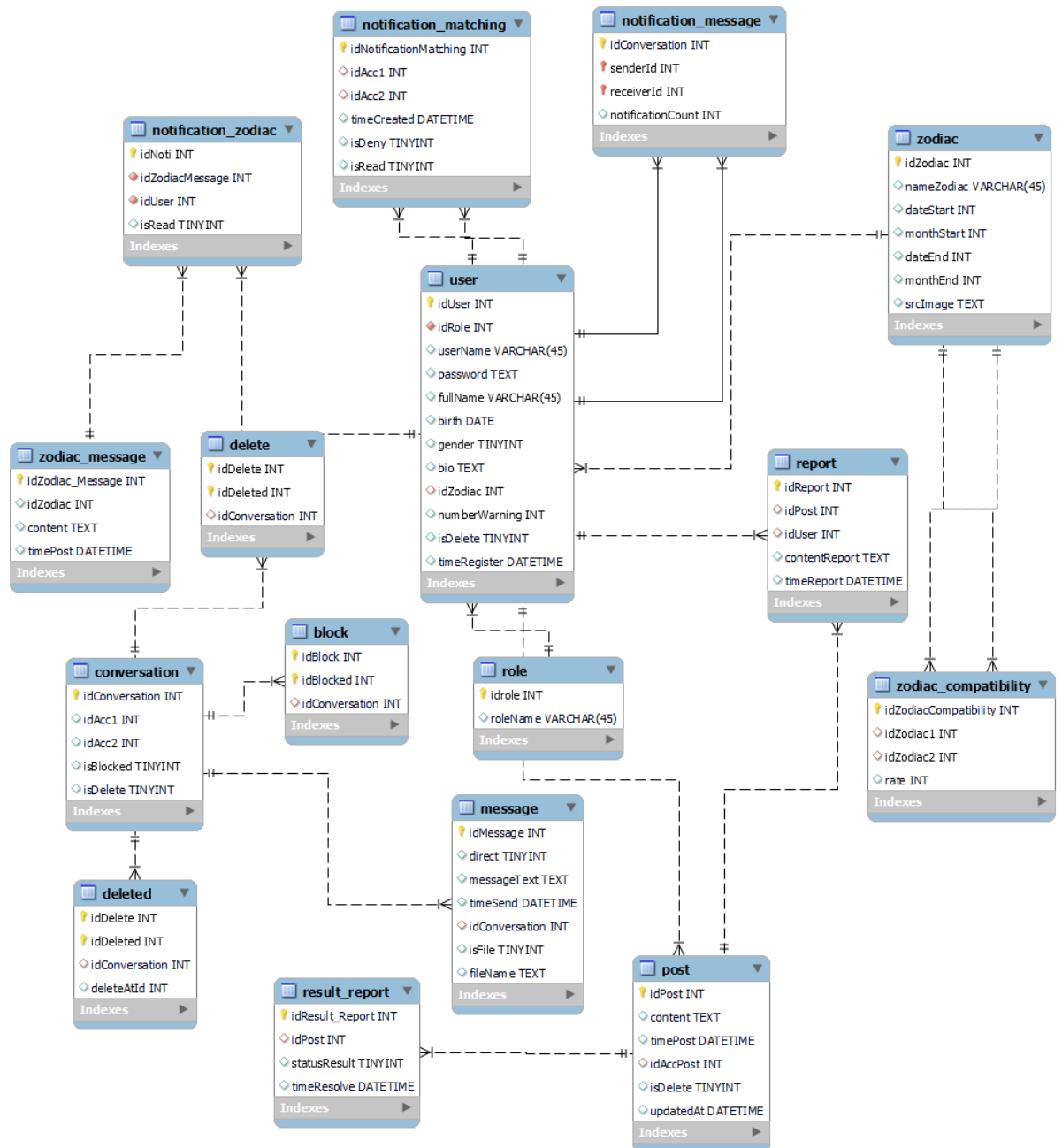
Bảng 23 Biểu đồ hoạt động xem thông báo thông điệp gửi từ admin

2.3.24. Biểu đồ hoạt động tìm kiếm bạn bè (không bị chặn)



Bảng 24 Biểu đồ hoạt động tìm kiếm bạn bè (không bị chặn)

2.4. Thiết kế cơ sở dữ liệu



Hình 11 Cơ sở dữ liệu

CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Triển khai hệ thống

- Ứng dụng xây dựng theo mô hình Client/Server
- Sử dụng giao thức TCP để matching, chat và truyền file trong hệ thống
- Sử dụng GitHub để quản lý mã nguồn
- Công nghệ sử dụng ở Front end: React.js, Restful api
- Xây dựng các module chức năng (Back end): sử dụng Node.js, Express.js
- Sử dụng socket.io để xây dựng những chức năng real-time
- Hoàn thiện hệ thống

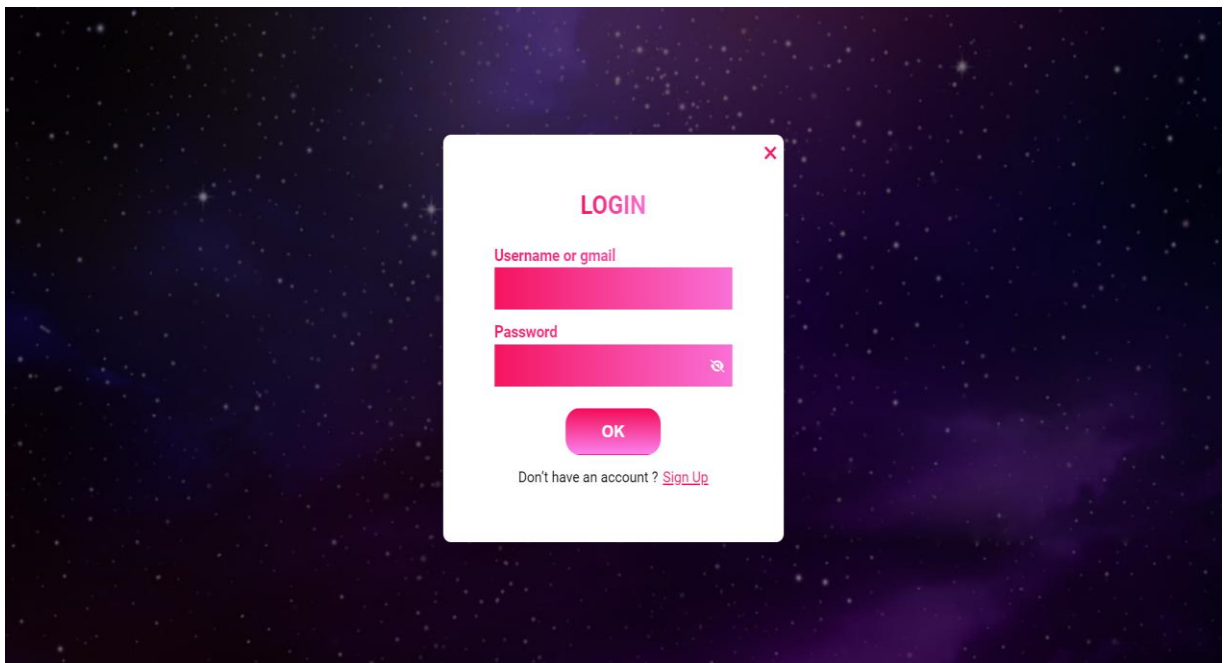
3.2. Kết quả

Các chức năng chính:

3.2.1 Phía User

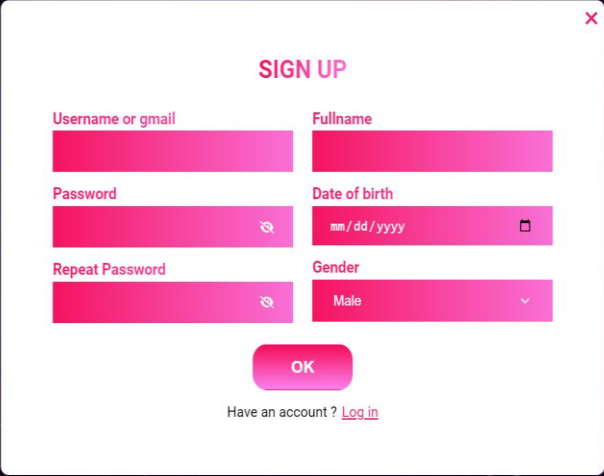
a) Chức năng login/register

➤ Login



Hình 12 Giao diện đăng nhập

➤ **Register**



A registration form titled "SIGN UP" with a close button (X) in the top right corner. The form is set against a dark space background with stars. It contains the following fields:

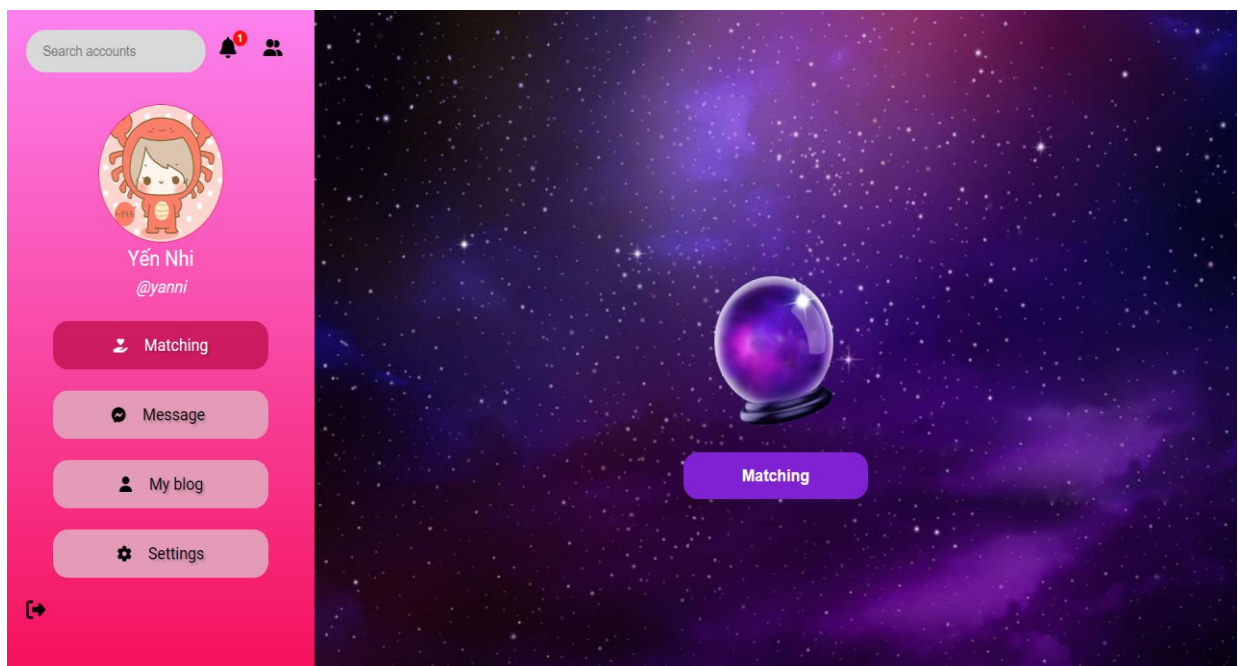
- Username or gmail**: A text input field.
- Fullname**: A text input field.
- Password**: A text input field with a visibility toggle icon.
- Date of birth**: A date picker field showing "mm/dd/yyyy".
- Repeat Password**: A text input field with a visibility toggle icon.
- Gender**: A dropdown menu currently showing "Male".

Below the fields is a blue "OK" button. At the bottom, it says "Have an account ? [Log in](#)".

Hình 13 Giao diện đăng ký

b) Chức năng ở Matching

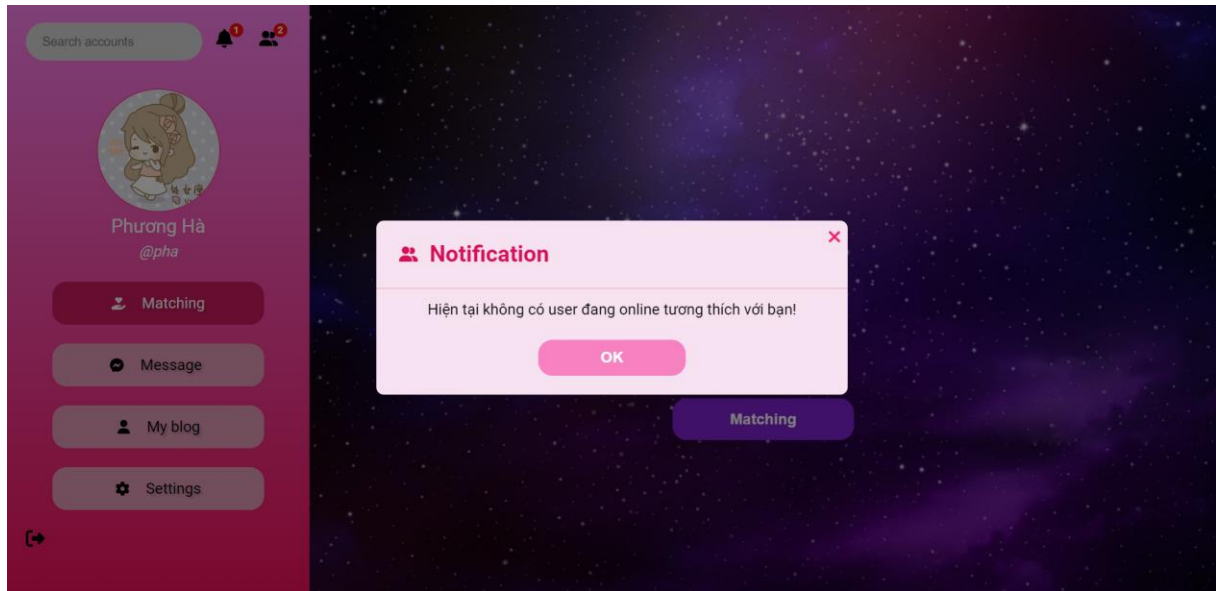
➤ **Trang chính**



Hình 14 Giao diện trang chủ matching

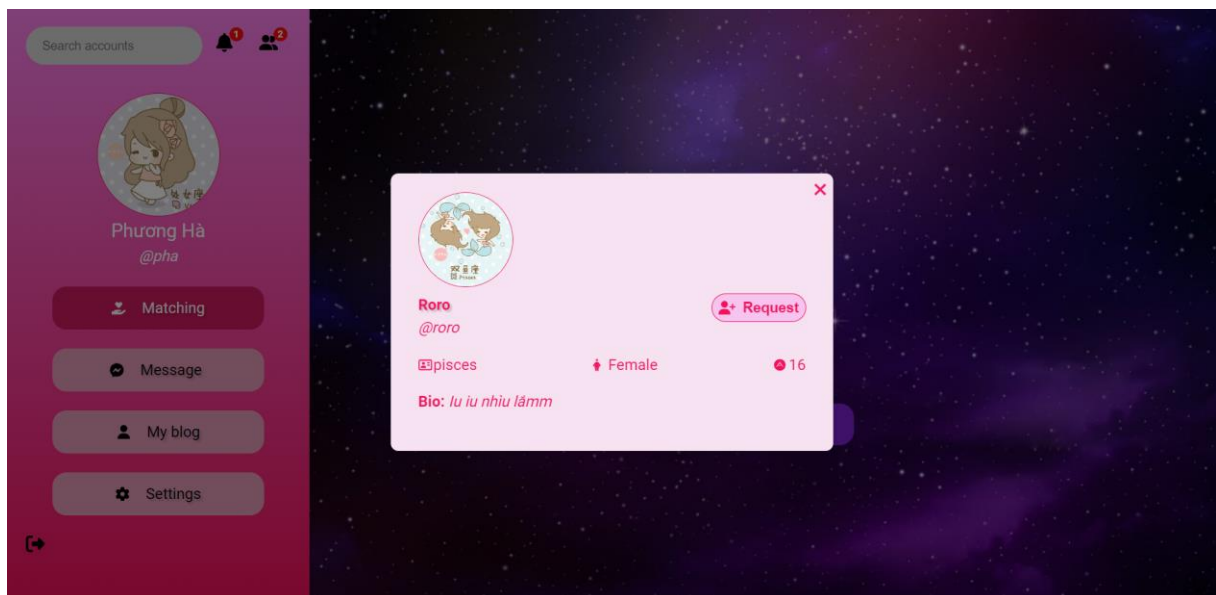
Matching

- Nếu không có user nào đang hoạt động và tương thích:



Hình 15 Giao diện không có user tương thích

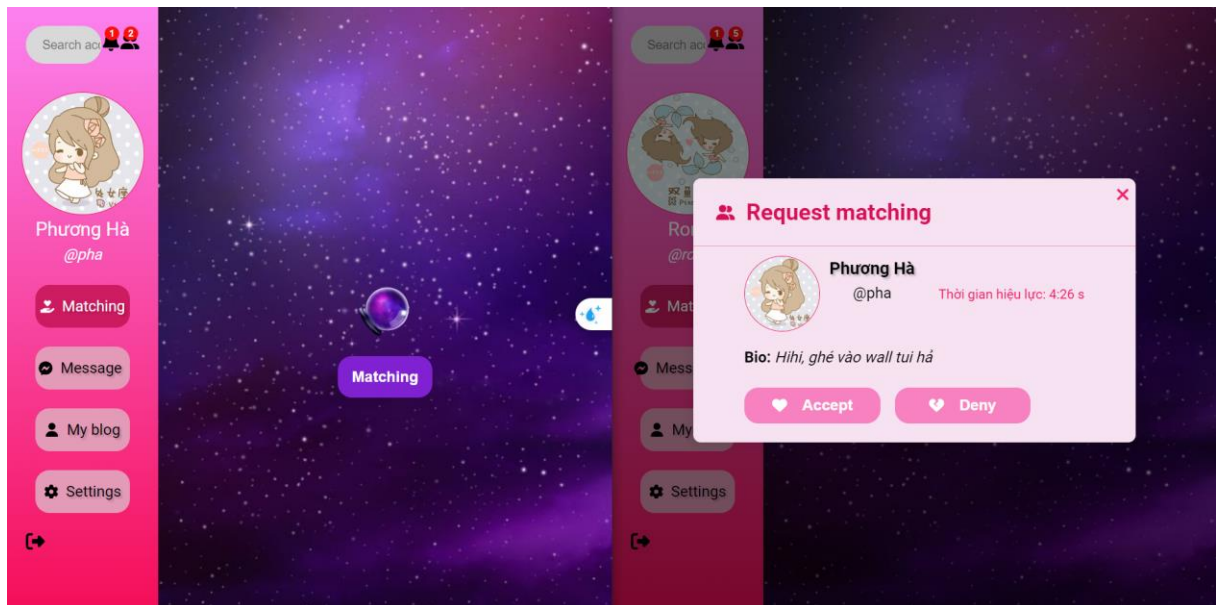
- Tồn tại user đang hoạt động (trả về user có độ tương thích cao)



Hình 16 Giao diện request matching

- Request matching thì phía được request sẽ có thông báo như dưới

- Giao diện



Hình 17 Giao diện nhận thông báo request matching

- Mã nguồn sử dụng socket.io

- Phía client

// Client 1 có id là fromId gửi yêu cầu matching đến Client 2 có id là matchId, emit lên Server xử lý yêu cầu matching

```
socket.emit("send-request-matching", {
    fromId: fromId,
    matchId: matchId
});
```

// Client 2 nhận yêu cầu matching từ Client 1 qua Server

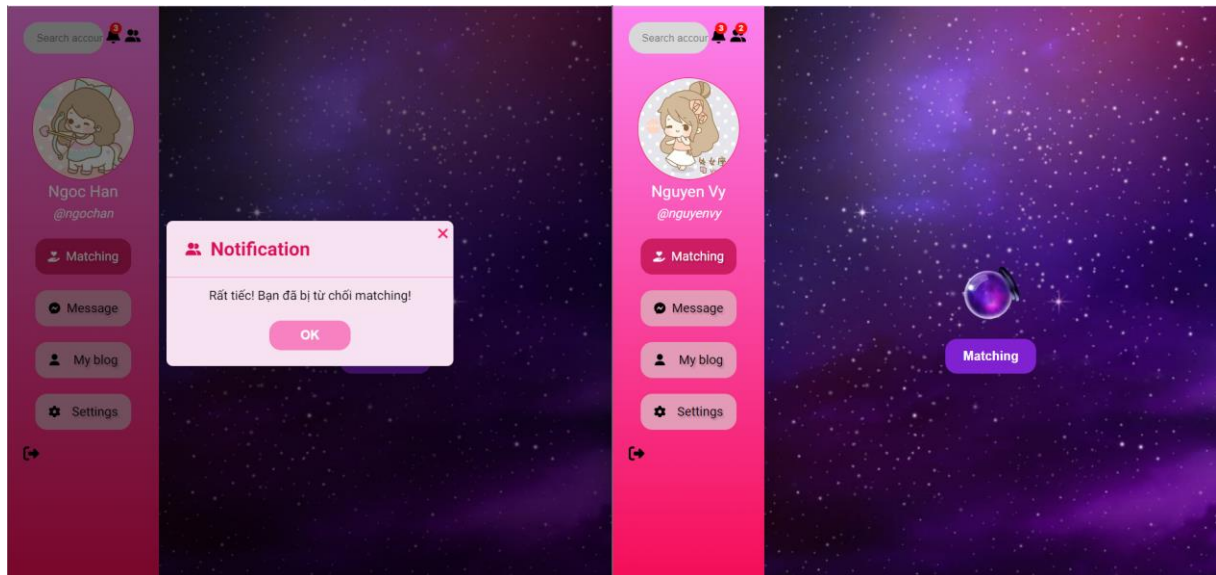
```
socket.on('receive-request-matching', (response) => {
    setShowRequest(!isShowRequest);
    setFromId(response.fromId);
    setMatchId(response.matchId);
});
```

- Phía server

// Server nhận yêu cầu matching từ Client 1

```
socket.on("send-request-matching", (data) => {
    const user = onlineUsers.find((user) => user.userID == data.matchId);
    if (user) {
        socket.to(user.socketID).emit("receive-request-matching", data); // Server gửi yêu cầu matching đến Client 2
    }
});
```


- Request matching và bị từ chối thì phía request sẽ có thông báo như sau
 - Giao diện



- Mã nguồn sử dụng socket.io

- Phía client

// Client 2 gửi thông báo từ chối matching lên Server

```
socket.emit('deny-matching', {  
  fromId: fromId,  
  matchId: matchId,  
});
```

// Client 1 nhận thông báo từ chối matching từ Server

```
socket.on('send-deny-matching', (data) => {  
  setShowDenyMatching(true);  
});
```

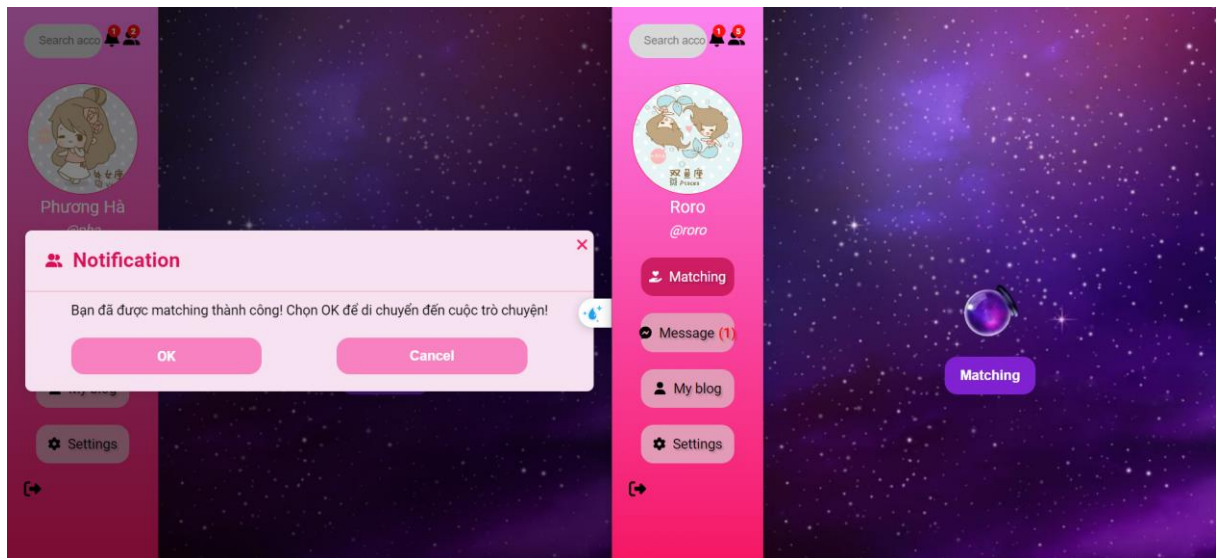
- Phía server

// Server nhận request từ chối matching từ Client 2

```
socket.on("deny-matching", (data) => {  
  const user = onlineUsers.find((user) => user.userID ==  
    data.fromId);  
  if (user) {  
    socket.to(user.socketID).emit("send-deny-matching", data);  
  }  
  // Server gửi thông báo từ chối matching đến Client 1  
});
```

- Request matching và phía được request đồng ý thì phía request sẽ có thông báo như sau

- **Giao diện**



Hình 18 Giao diện chấp nhận kết bạn

- **Mã nguồn sử dụng socket.io**

- **Phía client**

// Client 2 (matchId) đồng ý yêu cầu matching từ Client 1 (fromId), emit lên Server với dữ liệu gồm idConversation là đoạn hội thoại mới được tạo giữa Client 1 và Client 2, fromId và matchId

```
socket.emit('accept-request-matching', {  
    idConversation: data.idConversation,  
    fromId: fromId,  
    matchId: matchId,  
});
```

// Client 1 nhận thông báo kết bạn thành công

```
socket.on('move-to-new-conversation', (data) => {  
    console.log(data);  
    setShowNotifMatching(true);  
});
```

- **Phía server**

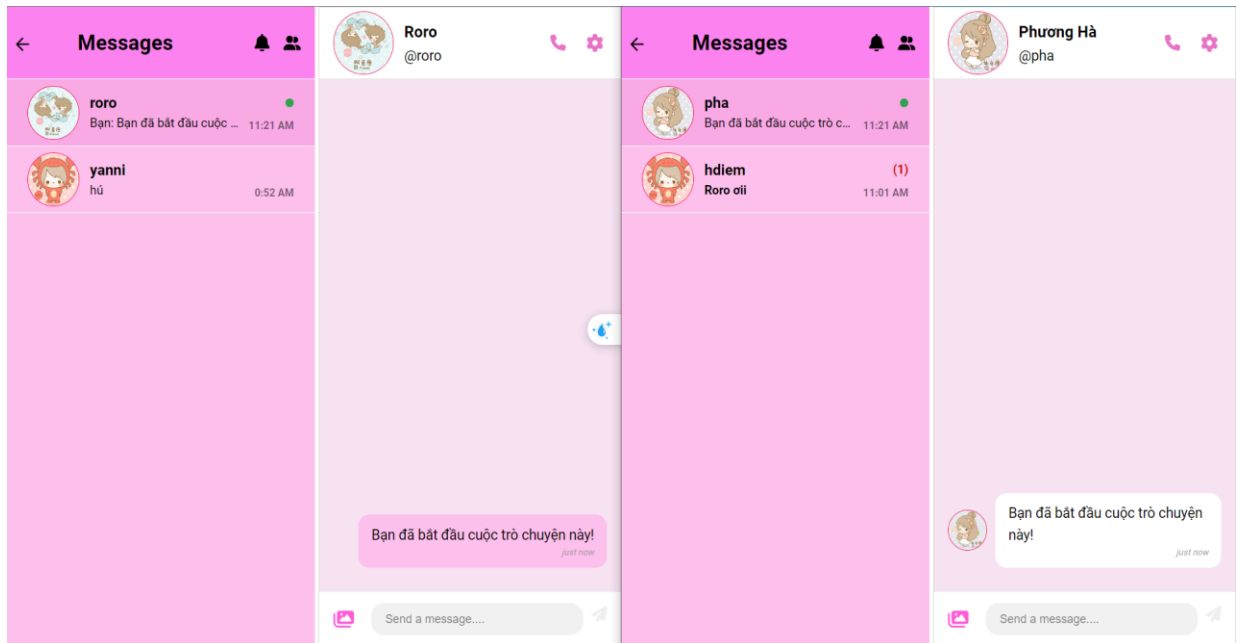
// Server nhận request đồng ý matching từ Client 2

```
socket.on("accept-request-matching", (data) => {  
    const user = onlineUsers.find((user) => user.userID ==  
data.fromId);  
    if (user) {
```



```
socket.to(user.socketID).emit("move-to-new-
conversation", data); // Server emit đến Client 1 kết bạn thành
công
}
});
```

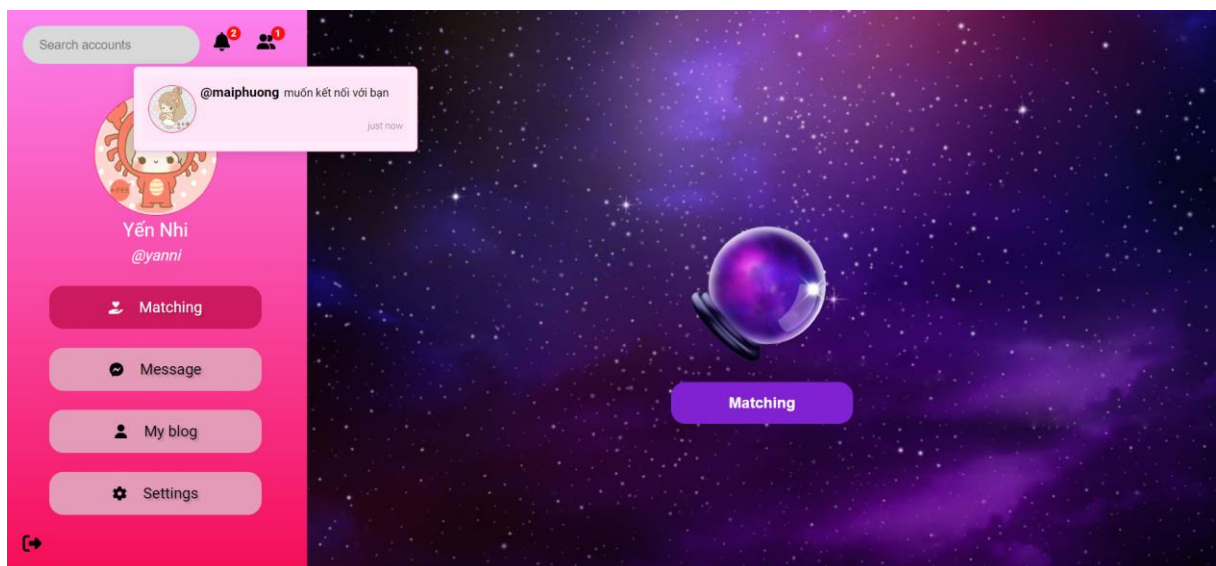
– Đi đến cuộc trò chuyện mới



Hình 19 Giao diện đi đến cuộc trò chuyện sau khi matching thành công

– Thông báo khi có người muốn matching

• Giao diện



Hình 20 Giao diện thông báo request matching

- **Mã nguồn sử dụng socket.io**

- **Phía client**

// Client 1 (fromId) gửi yêu cầu matching đến Client 2 (matchId) sẽ tạo một thông báo mới (idNotificationMatching) và emit đến Server xử lý

```
socket.emit("create-notif-matching", {
    fromId: fromId,
    matchId: matchId,
    idNotificationMatching: notifData.data,
})
```

- **Phía server**

// Server nhận request từ Client 1

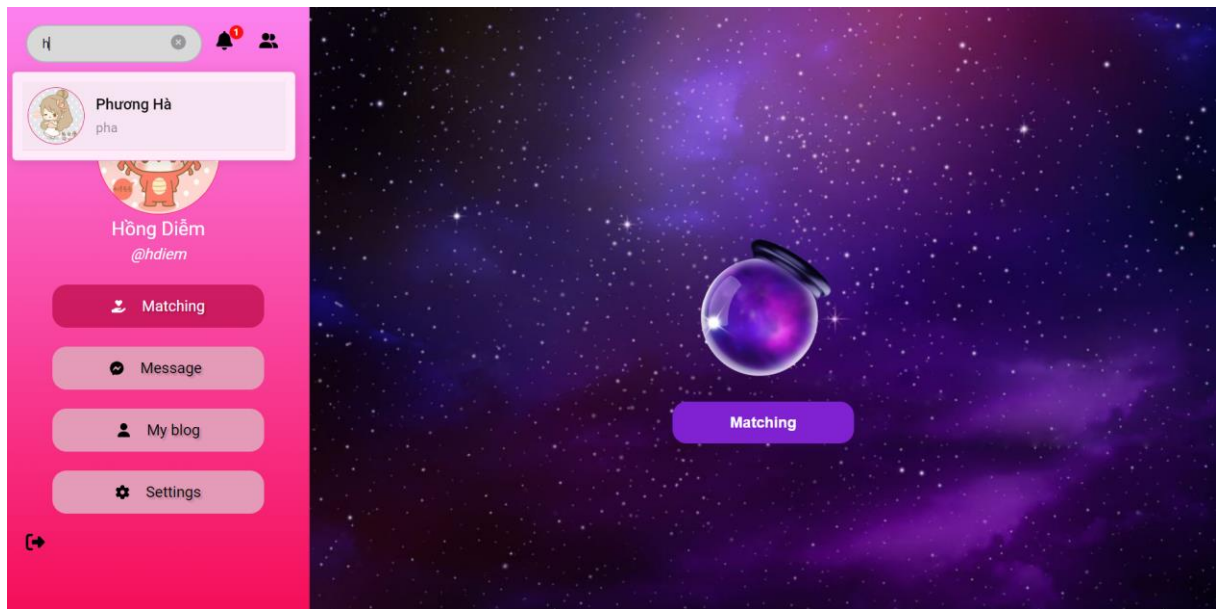
```
socket.on("create-notif-matching", (data) => {
    const user = onlineUsers.find((user) => user.userID == data.matchId);
    if (user) {
        socket.to(user.socketID).emit("receive-notif-matching", data); // Server gửi thông báo matching đến Client 2
    }
});
```

- **Xem chi tiết thông báo matching**



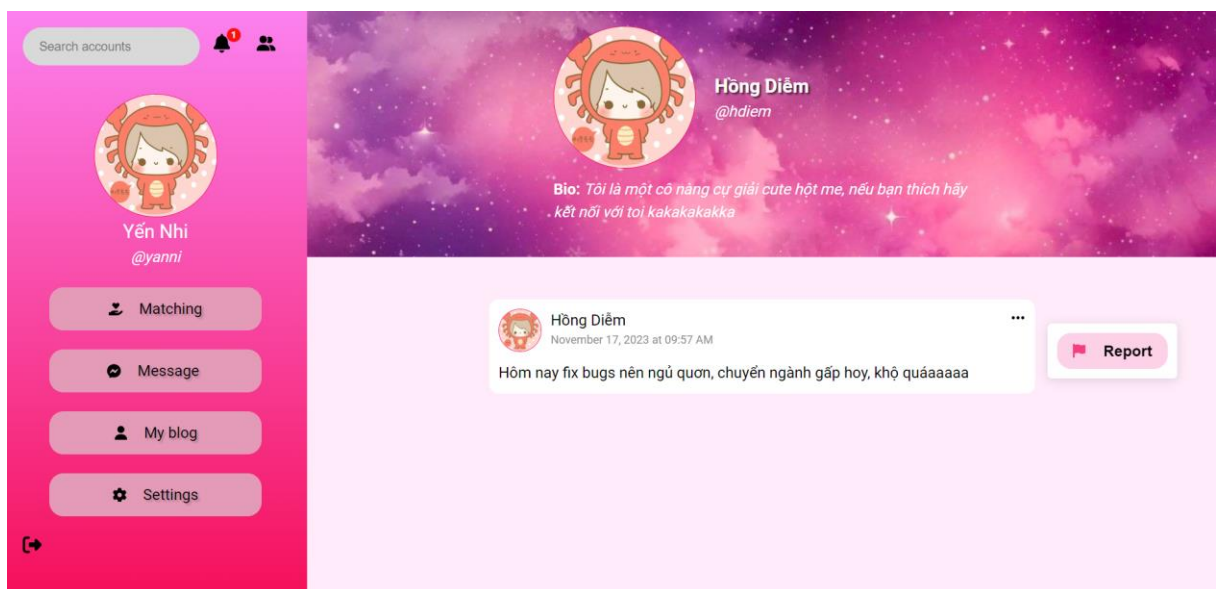
Hình 21 Giao diện xem chi tiết thông báo request matching

c) Chức năng tìm kiếm accounts (là bạn bè và không bị chặn)



Hình 22 Giao diện tìm kiếm bạn bè

d) Chức năng xem blog của bạn bè đã tìm kiếm

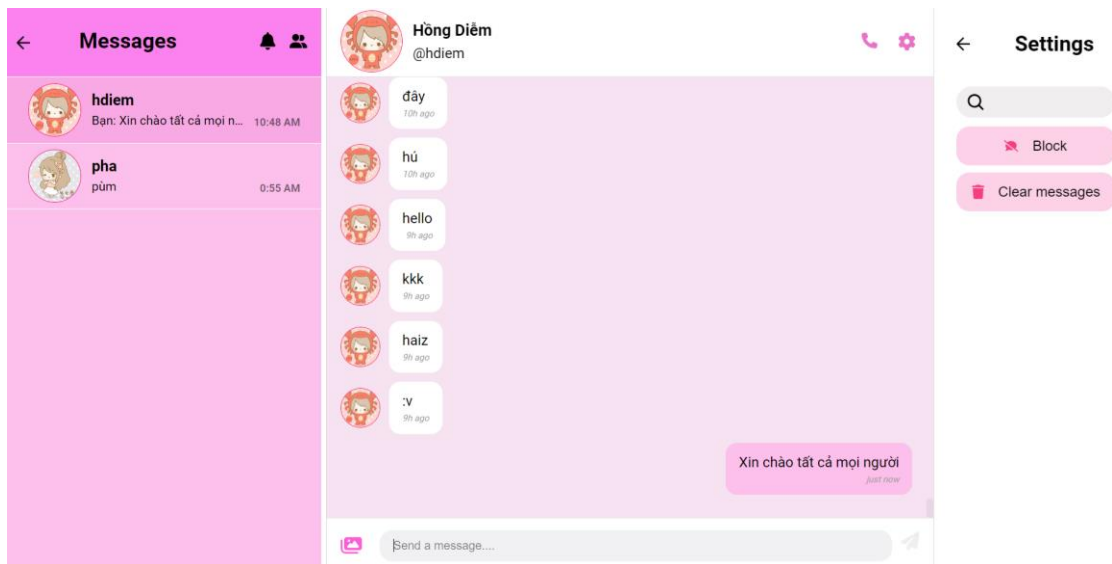


Hình 23 Giao diện xem blog bạn bè

e) Chức năng ở message

➤ Chat

• Giao diện



Hình 24 Giao diện chat với bạn bè

• Mã nguồn sử dụng socket.io

■ Phía client

- Client emit tin nhắn lên để **Server xử lý**:

/* Dữ liệu được gửi lên server là người gửi idSession, người nhận là _idSession, hướng gửi là direct_ để có thể biết là tin nhắn nằm phía người gửi hay người nhận, avatar người nhận, messageText là tin nhắn được gửi, timeSend là thời gian gửi. */

```
const newMessage = {
  _idSession,
  idSession,
  idUser_,
  direct_,
  avatar,
  messageText,
  timeSend,
  idConversation,
  isFile: 0,
};
```

```
socket.emit('send-message', newMessage);
```

- Client nhận tin nhắn được xử lý từ **Server trả về**:

```
if (socket === null) return;
```

```
socket.off('receive-message');
```

```
// Lắng nghe kết quả được gửi từ Server gửi về với dữ liệu là data
// (đã được xử lý)
```

```
socket.on('receive-message', async (data) => {  
    /*Kiểm tra xem ai là người gửi, người nhận để xét giá trị  
    direct và đặt hướng tin nhắn (người nhận là có nền màu  
    hồng, người gửi có nền màu trắng) */  
    if (data.idSession === idUser_ && data.idUser_ ===  
idSession) {  
        data.direct = 0;  
        setLoadMessages((prevLoadMessages) => ({  
            chat: [data, ...(prevLoadMessages?.chat || [])],  
        }));  
    }  
    setUserChat(chatUserResponse.userChatData);  
});
```

■ **Phía server**

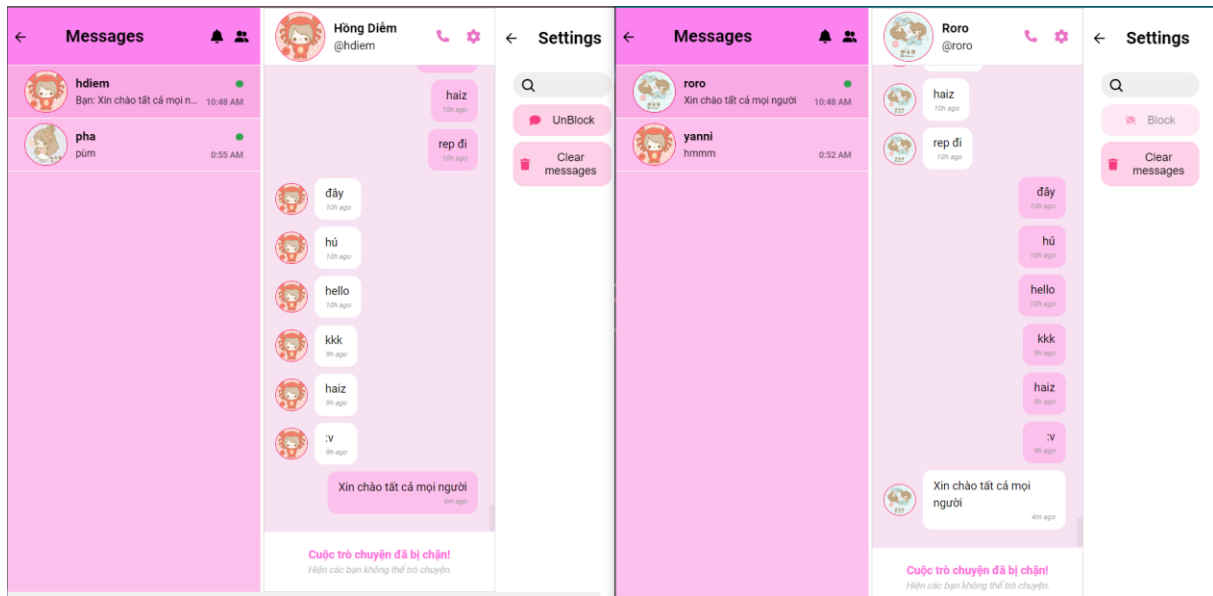
- Server xử lý và trả về Client

// Lắng nghe sự kiện được gửi từ Client

```
socket.on("send-message", (data) => {  
    /* Kiểm tra xem id của người được gửi có đang online hay không,  
    nếu online thì lấy ra socketID của người đó và gửi về */  
    const user = onlineUsers.find((user) => user.userID ===  
data._idSession);  
    if (user) {  
        socket.to(user.socketID).emit("receive-message", data);  
        socket.to(user.socketID).emit("receive-notification", {  
            idConversation: data.idConversation,  
            senderID: data.idSession,  
            receiverID: data._idSession,  
            isRead: false,  
            date: new Date(),  
        });  
    }  
});
```

➤ **Block tin nhắn**

- **Giao diện**



Hình 25 Giao diện block tin nhắn

- **Mã nguồn sử dụng socket.io**

- **Phía client**

- Client emit yêu cầu block chat lên để **Server xử lý**:

/* Dữ liệu được gửi lên server là người gửi idSession, người nhận là _idSession, hướng gửi là value có giá trị true hoặc false để xác định là block hay không, idUser_ là id người bị block và đoạn chat bị block là idConversation_*/

```
const handleBlockConversationChange = async (value) => {
    setIsBlocked(value);
    await socket.emit('block-conversation', { value, idUser_,
idConversation_ });
    const response = await handleGetAccById(idConversation_);
    const a2 = response.accountList.chat[0].idAcc2;
    const a1 = response.accountList.chat[0].idAcc1;
```

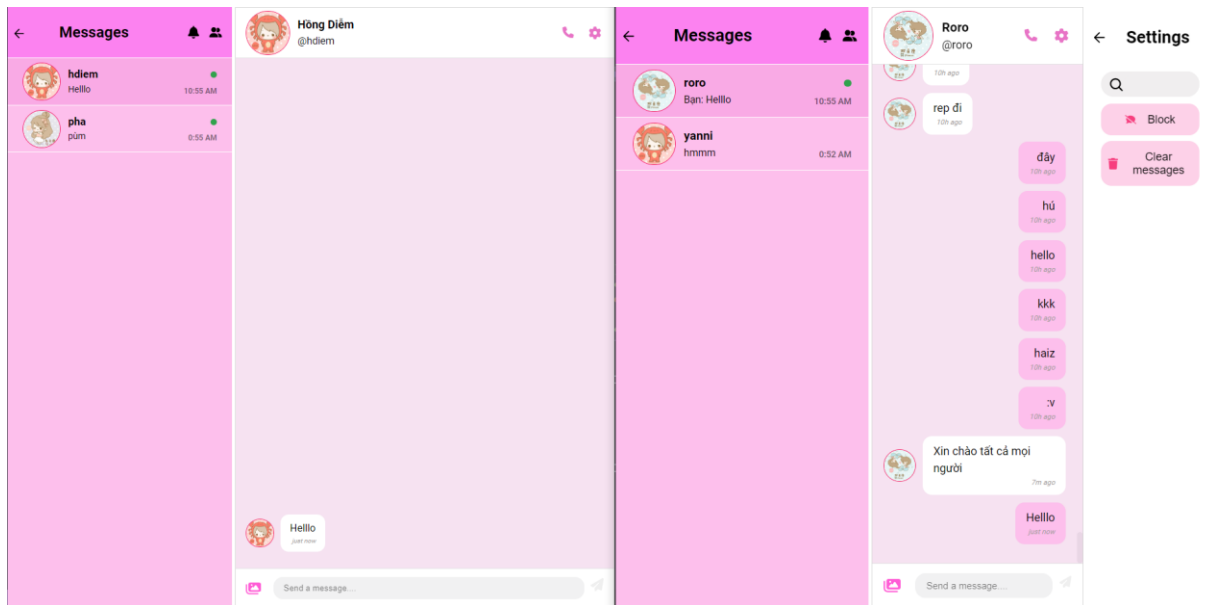
```
let idBlock = null;
if (a2 === idUser_) {
    setUserBlock(a1);
    idBlock = a1;
} else if (a1 === idUser_) {
    setUserBlock(a2);
    idBlock = a2;
}
```

```

        await handlePostBlockInfo(idBlock, idUser_,
idConversation_);
        toggle(false);
        setIsOpenBlock(true);
    };
- Client nhận lại kết quả block chat được xử lý từ Server trả về:
if (socket === null) return;
socket.off('blocked-conversation');
/* Client lắng nghe kết quả được gửi từ Server về là data đã được
xử lý */
socket.on('blocked-conversation', async (data) => {
    setCurrentChatBlock(data.idConversation_);
    setIsDisableBlock(true);
    setIsOpenBlock(false);
});
■ Phía server
- Server xử lý yêu cầu block để trả về cho Client
/* Kiểm tra xem id của người được gửi có đang online hay không,
nếu online thì lấy ra socketID của người đó và trả về kết quả cho
Client */
socket.on("block-conversation", (data) => {
    const user = onlineUsers.find((user) => user.userID ==
data.idUser_);
    if (user) {
        socket.to(user.socketID).emit("blocked-conversation", data);
    }
});

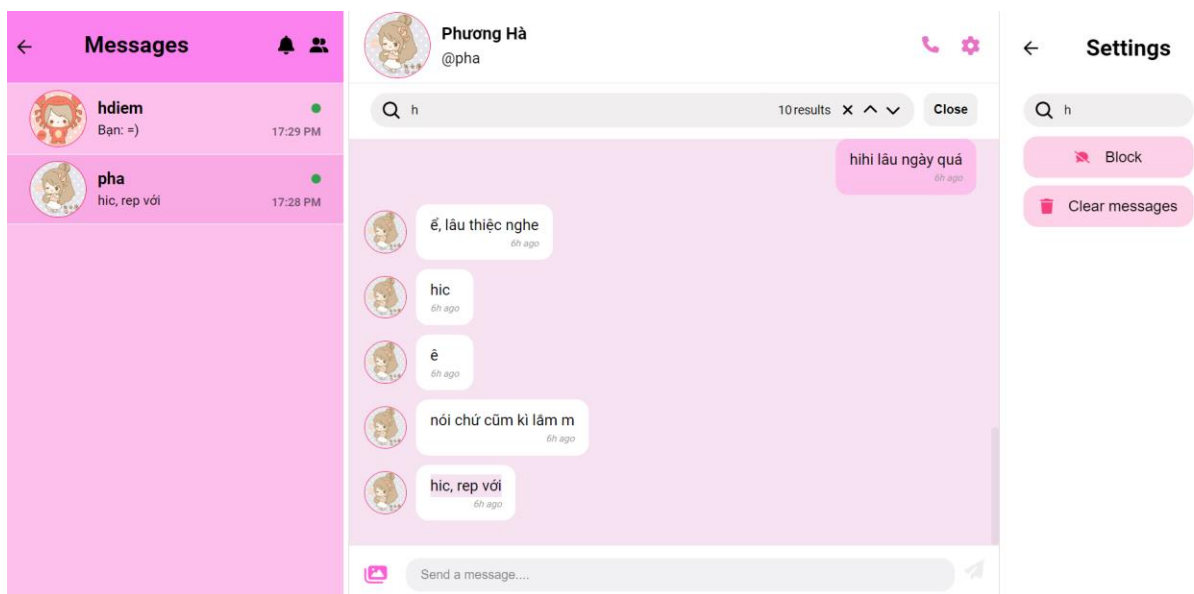
```

➤ Clear messages



Hình 26 Giao diện xóa tin nhắn

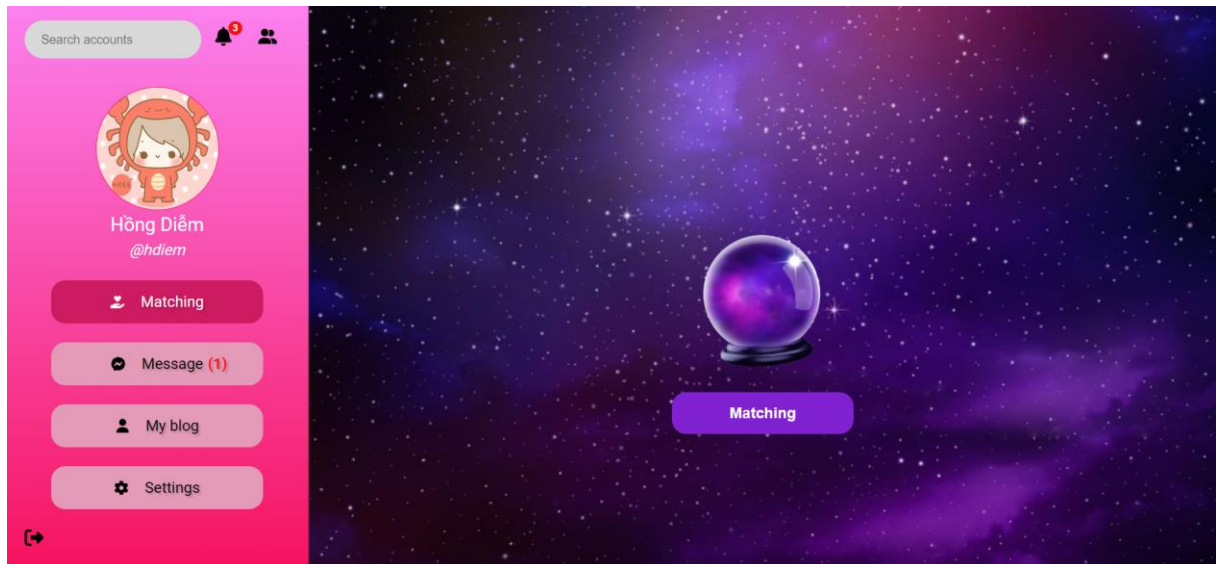
➤ Tìm kiếm tin nhắn



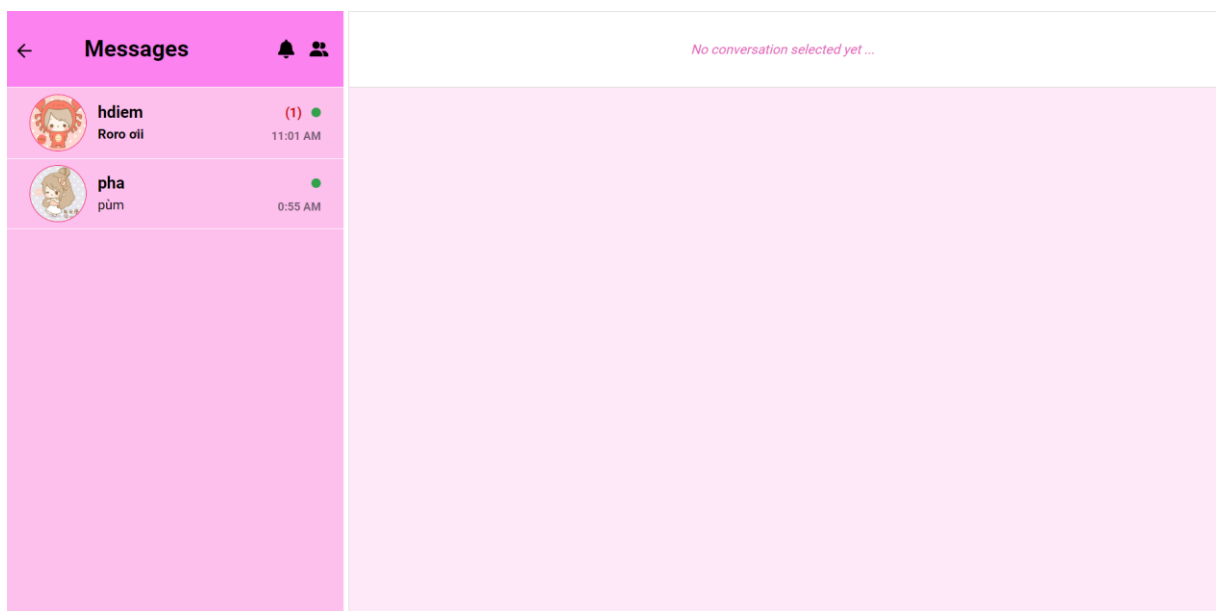
Hình 27 Giao diện tìm kiếm tin nhắn

➤ Thông báo tin nhắn

- Giao diện



Hình 28 Giao diện nhận thông báo tin nhắn



- Mã nguồn sử dụng socket.io

- Phía client

- Client nhận lại kết quả thông báo được xử lý từ **Server trả về**:
/* Kiểm tra xem hiện tại người nhận được tin nhắn có đang ở trong cuộc trò chuyện (đoạn chat) đó không, nếu có thì không cập nhật là số thông báo tin nhắn chưa đọc, ngược lại thì cập nhật notifications và tăng số thông báo chưa đọc là 1 */
if (socket === null) return;
socket.off('receive-notification');
socket.on('receive-notification', async (data) => {

```
const      currentUser      =      await
handleGetAccById(currentConversation);
const idAcc1 = currentUser.accountList?.chat[0]?.idAcc1;
const idAcc2 = currentUser.accountList?.chat[0]?.idAcc2;
setSenderID(data.senderID);
setReceiveID(data.receiverID);

if (
  (data.senderID === idAcc1 && data.receiverID ===
idAcc2) ||
  (data.senderID === idAcc2 && data.receiverID ===
idAcc1) // Đã đọc tin nhắn
) {
  setNotifications((prev) => prev && [{ ...data, isRead:
true }, ...prev]);

handleDeleteNotificationMessageInfo(data.idConversation,
data.senderID);
  } else { // Chưa đọc tin nhắn
    setNotifications((prev) => prev && [data, ...prev]);
  }
});
```

■ **Phía server**

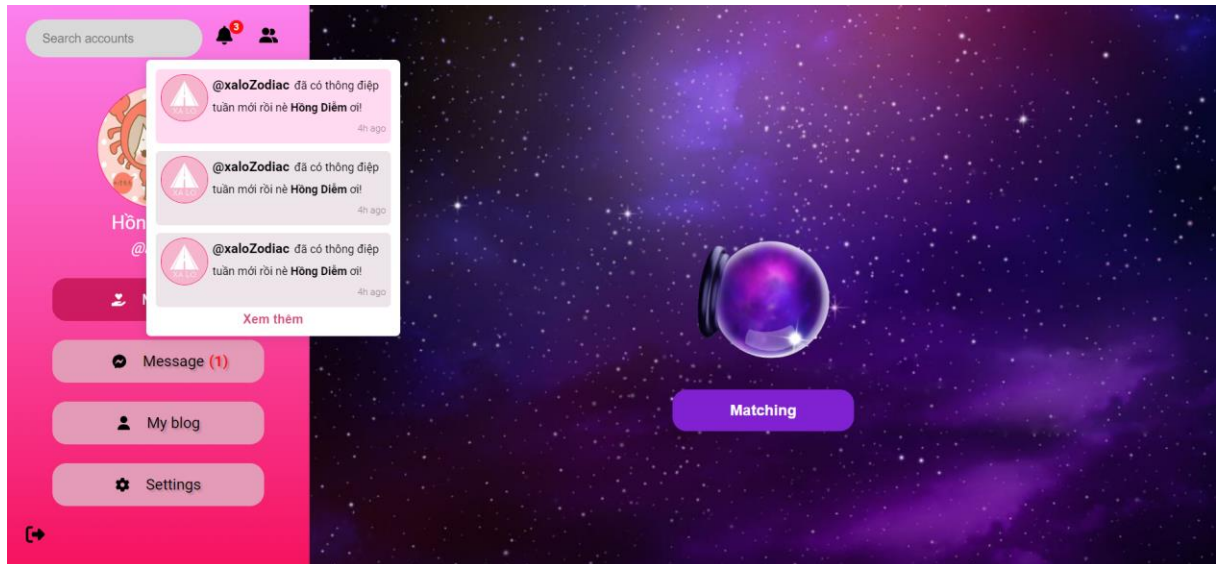
- Server xử lý yêu cầu block để trả về cho Client
/* Kiểm tra xem id của người được gửi có đang online hay không, nếu online thì lấy ra socketID của người đó và trả về kết quả cho Client */

```
socket.on("send-message", (data) => {
  const user = onlineUsers.find((user) => user.userID ==
data._idSession);
  if (user) {
    /* Nhận tin nhắn đồng thời nhận yêu cầu của client về thông
báo */
    socket.to(user.socketID).emit("receive-message", data);
    socket.to(user.socketID).emit("receive-notification", {
      idConversation: data.idConversation,
      senderID: data.idSession,
      receiverID: data._idSession,
```

```
isRead: false,  
date: new Date(),  
});  
}});
```

f) Chức năng nhận thông điệp tuần mới từ admin

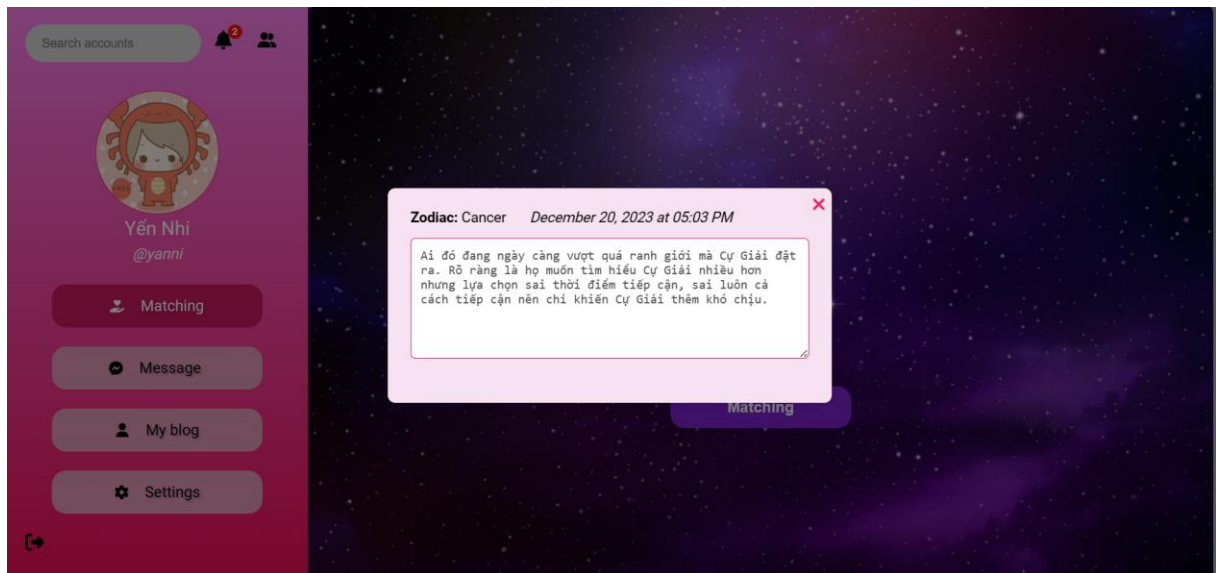
- **Danh sách các thông báo**
 - **Giao diện**



Hình 29 Giao diện nhận thông điệp tuần mới từ Admin

- **Mã nguồn sử dụng socket.io**
 - **Phía client**
 - Client emit gửi thông tin thông điệp được gửi từ Admin lên để Server xử lý:
/* Vì có 12 cung hoàng đạo, để hạn chế dữ liệu được gửi thì chúng ta chỉ lọc ra những cung hoàng đạo của user đang online (zodiacs) để gửi lên Server
/* Nếu zodiacs.length > 0 thì mới gửi bằng socket, ngược lại chỉ cần lưu dữ liệu vào database */
if (zodiacs.length > 0) {
try {
// Duyệt qua từng idZodiac trong danh sách các zodiacs thỏa mãn
for (const zodiac of zodiacs) {
const res = await handleGetIdZodiacMessage
(item.idZodiac, date, zodiac.userID);
newZodiacMessageList.push({
// Lấy ra idUser tương ứng với cung hoàng đạo
và đang online

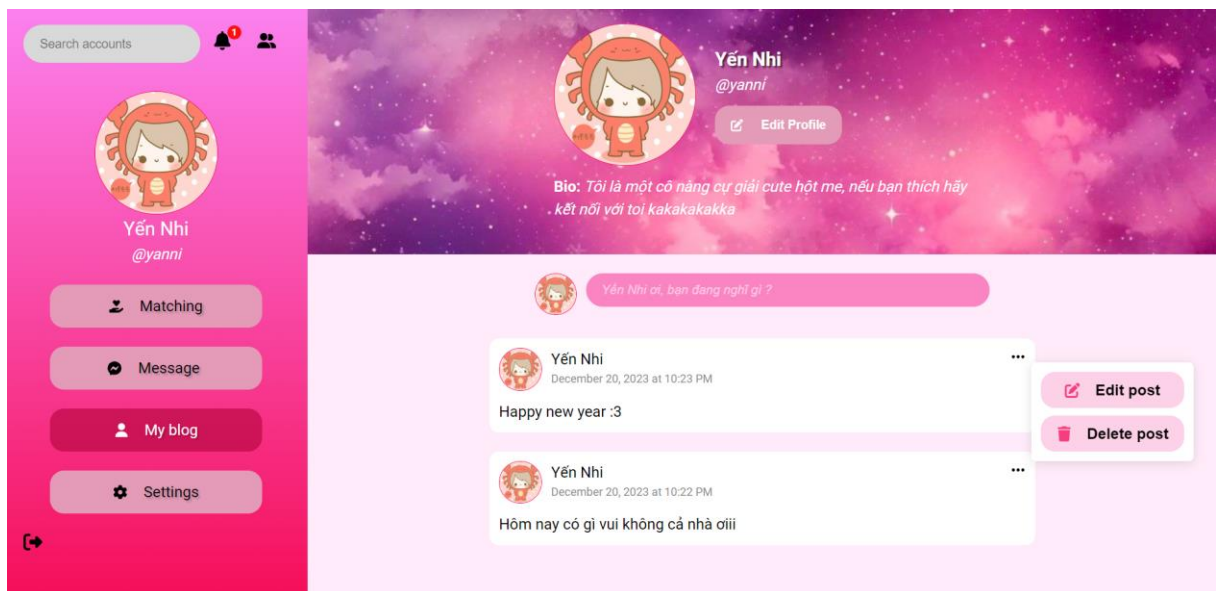
➤ Xem chi tiết thông báo



Hình 30 Giao diện xem chi tiết thông báo thông điệp được gửi từ Admin

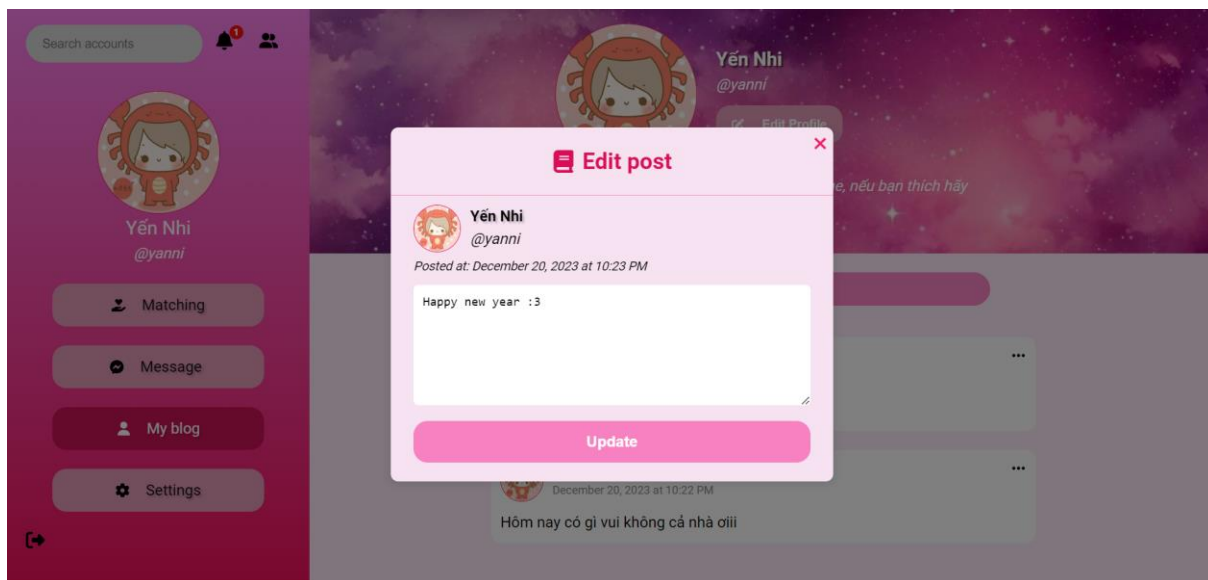
g) Chức năng ở My blog

➤ Xem bài viết đã đăng



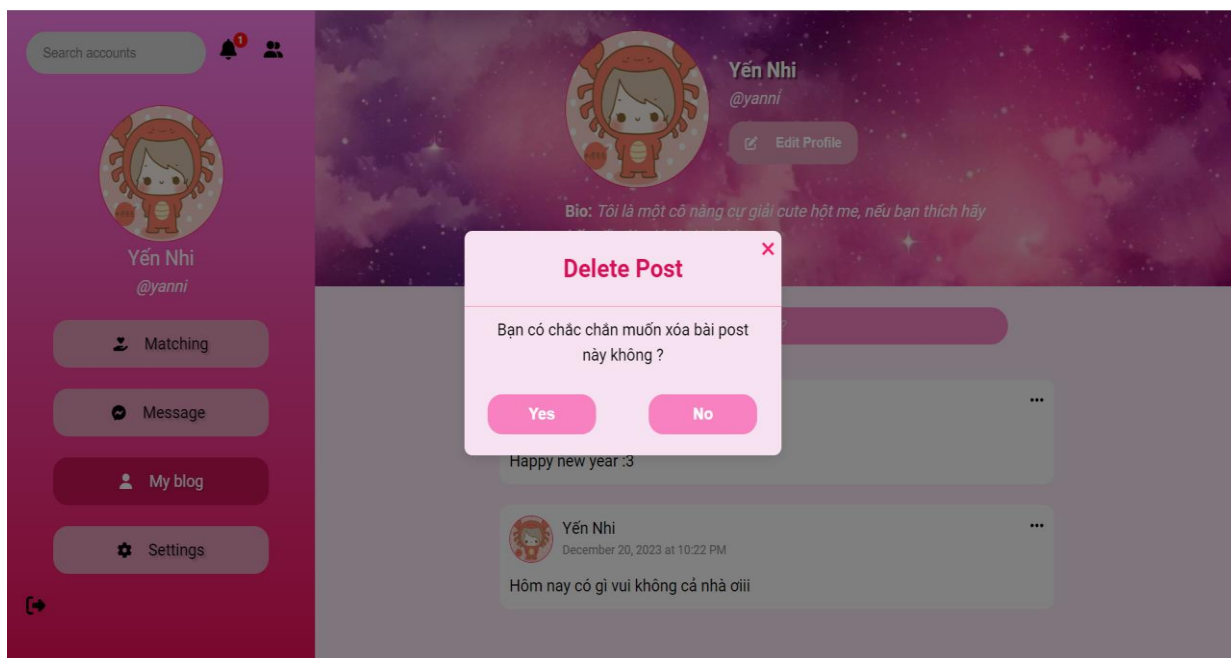
Hình 31 Giao diện xem trang blog cá nhân

➤ **Chỉnh sửa bài viết**



Hình 32 Giao diện chỉnh sửa nội dung bài viết

➤ **Xóa bài viết**



Hình 33 Giao diện xóa bài viết

h) Chức năng ở Settings

➤ Edit profile

Settings

Edit profile

Change password

Edit profile

Username: hdiem
www.website.com/api/profile/@hdiem
Usernames can only contain letters, numbers, underscores, and periods. Changing your username will also change your profile link.

Name*: Hồng Diễm

Bio: Tôi là một cô nàng cực giải cute hột me, nếu bạn thích hãy kết nối với tôi kakakakaka
allow 50 characters

Date of birth*: 07/06/2003

Gender*: Female

Save

Hình 34 Giao diện chỉnh sửa thông tin cá nhân

➤ Change password

Settings

Edit profile

Change password

Change password

Current password: Enter current password

New password: Enter new password

Re-type password: Re-type new password

Your password must be at least 6 characters and should include a combination of numbers, letters and special characters (!\$@%).

Save

Hình 35 Giao diện thay đổi mật khẩu

3.2.2 Phía Admin

a) Chức năng ở User

➤ Xem danh sách toàn bộ user

Admin/User

Tổng số user: 6 Số user đang online: 1

Username	Fullname	Birthday	Gender	Number Warnings	Online	
hdiem	Hồng Diễm	06/07/2003	Nữ	3	Offline	
pha	Phuong Hà	10/09/2003	Nữ	3	Offline	
yanni	Yến Nhi	17/07/2003	Nữ	1	Online	
ngochan	Ngoc Han	05/12/2023	Nam	0	Offline	
nguyenvy	Nguyen Vy	08/09/2023	Nữ	0	Offline	
thuanbui	Thuan Bui	12/05/2023	Nữ	0	Offline	

Previous 1 Next

Hình 36 Giao diện xem danh sách toàn bộ user

➤ Xem chi tiết thông tin user

Detail information

Username: hdiem

Full name: Hồng Diễm

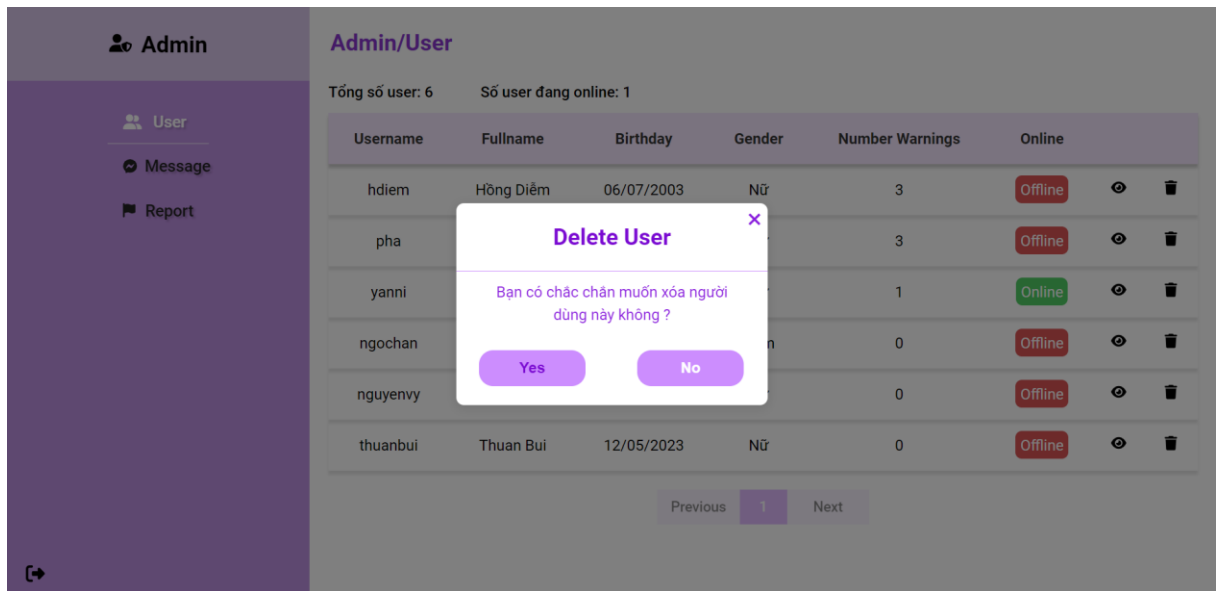
Birthday: 07/06/2003 Gender: Male ☐ Female ☒

Number warning: 3

Bio: Tôi là một cô nàng cực giải cute hột me, nếu bạn thích hãy kết nối với tôi kakakakaka

Hình 37 Giao diện xem chi tiết thông tin user

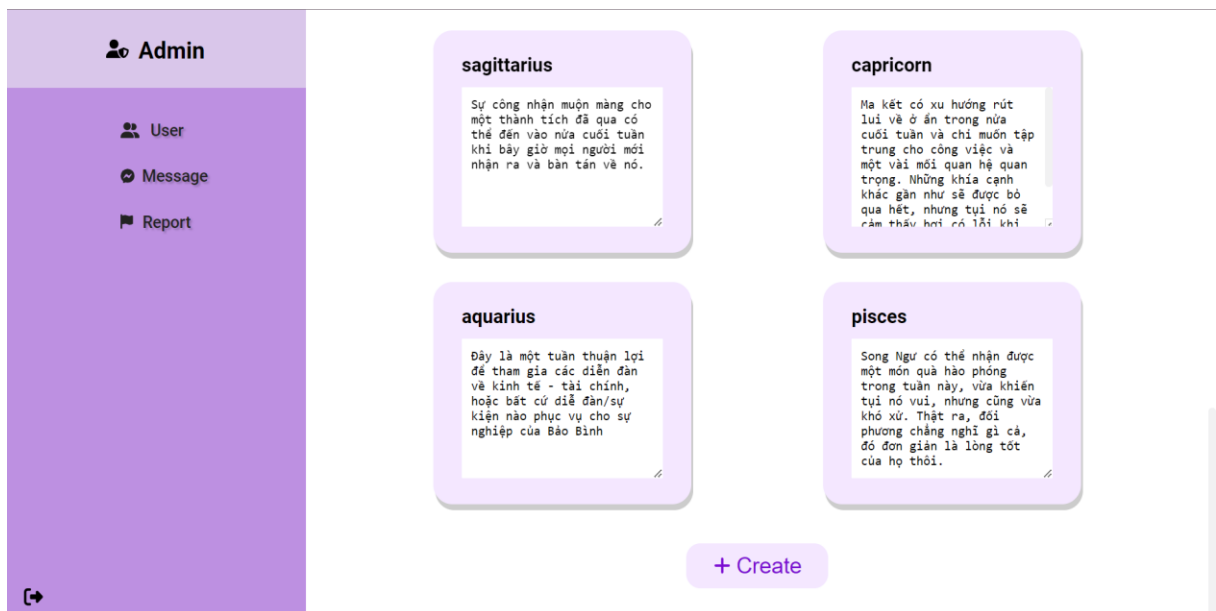
➤ **Xóa user**



Hình 38 Giao diện xóa user

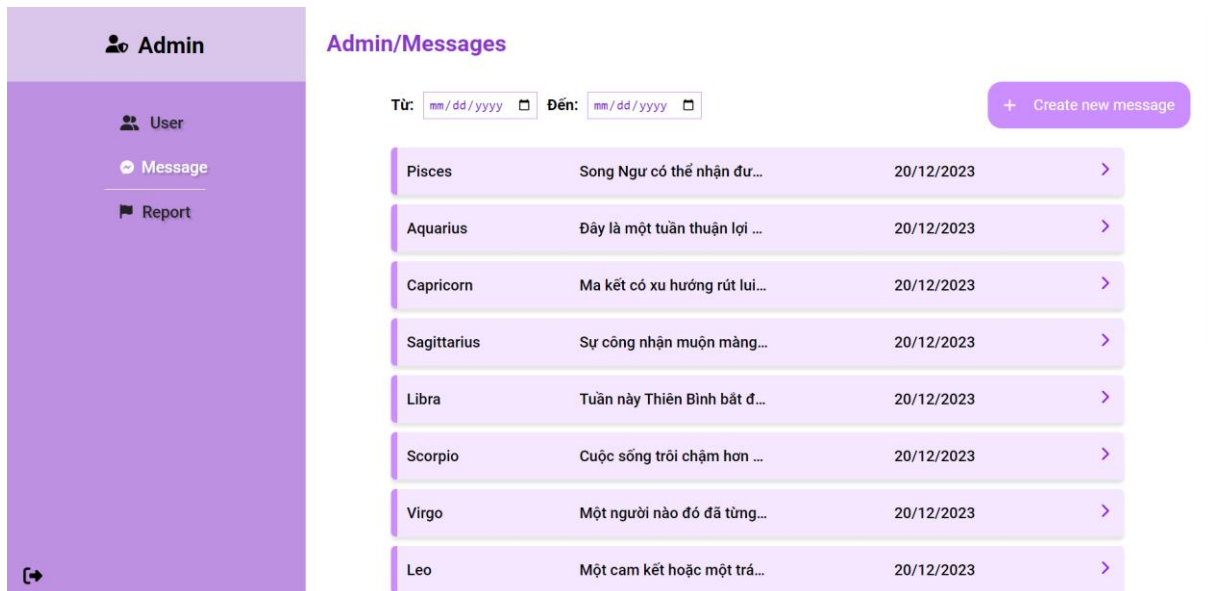
b) Chức năng ở Message

➤ **Tạo thông điệp hàng tuần cho 12 cung hoàng đạo**



Hình 39 Giao diện tạo thông điệp hàng tuần cho 12 cung hoàng đạo

➤ Danh sách thông điệp hàng tuần cho 12 cung hoàng đạo



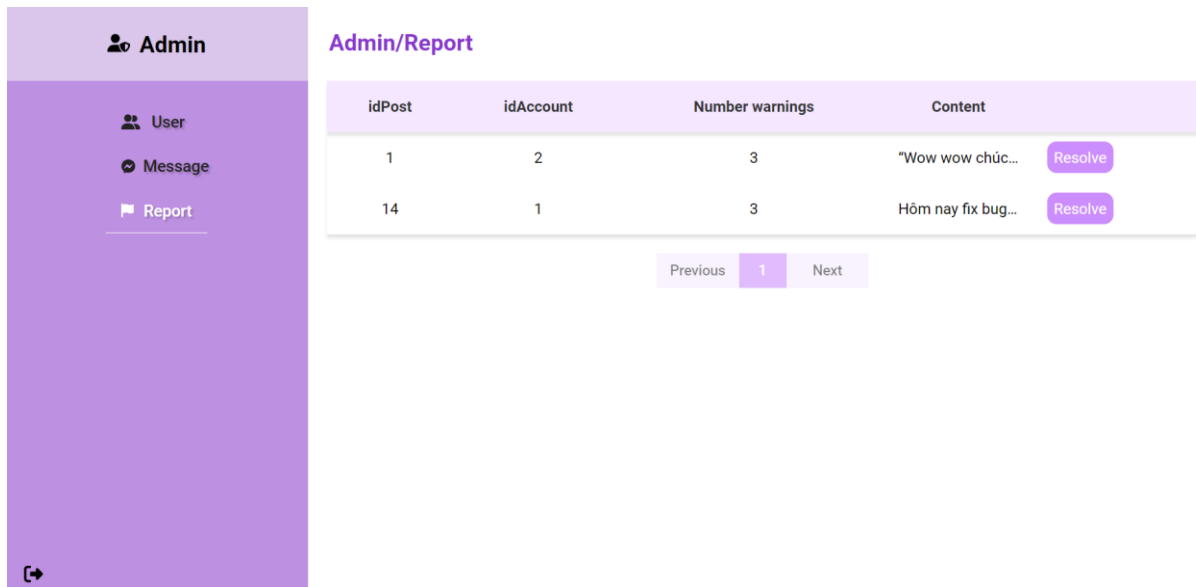
Hình 40 Giao diện xem danh sách thông điệp cho các cung hoàng đạo

➤ Xem chi tiết thông điệp của từng cung hoàng đạo



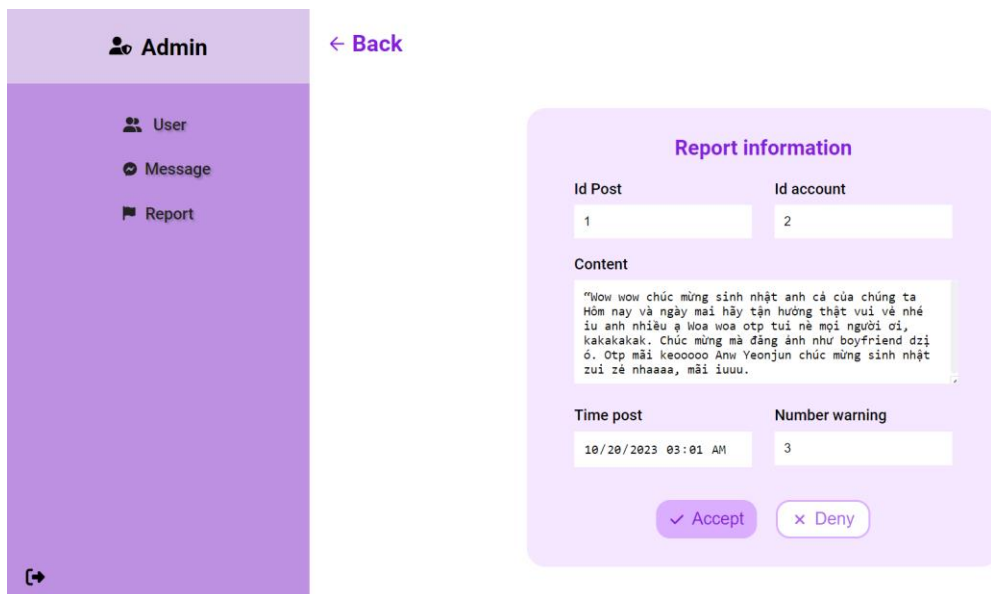
Hình 41 Giao diện xem chi tiết thông điệp của từng cung hoàng đạo

➤ Danh sách các bài viết bị report



Hình 42 Giao diện xem danh sách các bài viết bị report

➤ Xem thông tin chi tiết các report và giải quyết



Hình 43 Giao diện xem thông tin chi tiết các report và giải quyết

3.3. Đánh giá

- Hệ thống đáp ứng đủ những yêu cầu cơ bản của một ứng dụng mạng xã hội
- Giao diện hiển thị thân thiện với người dùng, dễ sử dụng
- Hệ thống đảm bảo việc kết nối mỗi quan hệ có sự chọn lọc dựa trên độ tương thích cung hoàng đạo
- Những điểm cần khắc phục:
 - Hệ thống chưa hoạt động ổn định khi có số lượng lớn user truy cập vào hệ thống
 - Hệ thống chỉ cho truyền file với kích thước nhỏ hơn 1MB

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

- Qua quá trình thực hiện dự án, chúng em đã tìm hiểu và nắm rõ hơn về mô hình client-server và cách mà nó hoạt động trong việc truyền thông và tương tác giữa các thành phần khác nhau của hệ thống, cách thức hoạt động của giao thức TCP,... Bên cạnh đó, chúng em cũng học được cách quản lý cơ sở dữ liệu và lưu trữ thông tin người dùng một cách an toàn và hiệu quả. Việc tối ưu hóa truy vấn trong kết nối cơ sở dữ liệu đã giúp cải thiện hiệu suất của hệ thống
- Nhìn chung, hệ thống đã đạt được các mục tiêu chính về chức năng, gồm kết nối, nhắn tin, truyền file và đăng bài viết. Đặc biệt, việc tích hợp yếu tố cung hoàng đạo đã mang đến những trải nghiệm kết nối độc đáo, đồng thời giúp người dùng tìm kiếm những mối quan hệ có độ tương thích cao hơn.
- Chúng em hy vọng rằng dự án này sẽ đóng góp vào sự phát triển và nâng cao khả năng kết nối của mọi người trong môi trường trực tuyến.

4.2 Hướng phát triển

- Hoàn thiện các tính năng hiện có và tối ưu thuật toán
- Tối ưu trải nghiệm người dùng
- Xử lý được vấn đề khi có số lượng lớn user truy cập vào hệ thống
- Thực hiện truyền file với kích thước lớn hơn
- Tiến hành kiểm thử
- Đưa ứng dụng vào hoạt động
- Thu thập phản hồi của người dùng khi trải nghiệm hệ thống
- Theo dõi hoạt động của hệ thống và tiến hành bảo trì

TÀI LIỆU THAM KHẢO

[1] Ths. Mai Văn Hà, *Slide Chương 3 - Các Giao Thức Cơ Bản*

[2] PGS.TS. Nguyễn Tấn Khôi, *Giáo Trình Mạng Máy Tính*, Đà Nẵng, 2004

PHỤ LỤC

Mã nguồn:

- *Front-end*: <https://github.com/hdroro/PBL4>
- *Back-end*: <https://github.com/hdroro/PBL4-be>
- *Socket.IO*: <https://github.com/hdroro/socket.io>