

**TÀI LIỆU THIẾT KẾ BAN ĐẦU**

**Thiết kế kiến trúc của dự án**

**Lawyer Connect**

**Được soạn bởi:   
Hoàng Đức Dương**

**Hà Tiến Đông  
Đào Thị Bình An**

**Bùi Huyền Tâm**

**Hoàng Duy Hưng**

**Trường Đại học Công nghệ, ĐHQGHN**

**Hà Nội, 26-04-2024**

**Mục lục**

[**1. Kiến trúc hệ thống 3**](#_rikjbek1rb91)

[1.1. Vấn đề đặt ra 3](#_uxlloy6bes3k)

[1.2. Kiến trúc MVC 4](#_of0wcxd7cokt)

[1.3. Biểu đồ thành phần hệ thống 4](#_h6tu7jf2o4si)

[**2. Biểu đồ tuần tự các ca sử dụng chính 6**](#_rwrgnwrd1fc1)

[2.1. Đăng ký 6](#_1nwgcxarcgdn)

[2.2. Đăng nhập 6](#_4jjf5kmui8tb)

[2.3. Tìm kiếm và xem hồ sơ luật sư 7](#_pqyq8ap0vmb4)

[2.4. Quản lý cuộc hẹn 9](#_q95tlhtinm0a)

[2.5. Report 10](#_bwxx6zwyhieu)

[2.6. Đánh giá, phản hồi về luật sư 11](#_w0qs6h3dgy6h)

[2.7. Xem và chỉnh sửa thông tin 12](#_cla9pvg7h92)

[2.8. Thay đổi mật khẩu 13](#_lrkbx2cj4qs8)

[2.9. Admin quản lý người dùng và trang web 14](#_n1xanjkc3ara)

[2.9.1. Tạo tài khoản 14](#_u1kh10mu7r8m)

[2.9.2. Xóa tài khoản 15](#_a9dn6uagjc0m)

[2.9.3. Cập nhật tài khoản 16](#_e7yhh3t5fiz2)

[2.9.4. Xóa cuộc hẹn 17](#_ir6fmoybm3rn)

[2.9.5. Chỉnh sửa cuộc hẹn 18](#_r150khnaz2jn)

[**3. Biểu đồ lớp các ca sử dụng 19**](#_aqgvtopt1xdr)

[3.1. Đăng nhập 19](#_g33g7b7s534r)

[3.2. Đăng ký 20](#_o0784ht0ylwv)

[3.3. Đổi mật khẩu 21](#_m1cr0kk7qwvh)

[3.4. Cập nhật thông tin 22](#_r3wguxaru2ug)

[3.5. Tìm kiếm luật sư 22](#_1w8zow9stfeu)

[3.6. Xem thông tin luật sư 23](#_7xzvxd402l9a)

[3.7. Quản lý cuộc hẹn 23](#_84tuh53o1tym)

[3.8. Report 24](#_sszfdkpft653)

[3.9. Đánh giá, phản hồi về luật sư 24](#_5ej9alp2bo50)

[3.10. Admin quản lý người dùng và trang web 25](#_vl224wi2hub4)

[**4.**](#_z011kp2z763t) [**Thiết kế cơ sở dữ liệu**](#_n3n4jff44s42) [**26**](#_z011kp2z763t)

4[.1. Giới thiệu 2](#_kb0r4qbw1587)6

4[.2. Thiết kế 27](#_3a2u0yt8kaqj)

4[.3. Tổng kết 2](#_ey4jv5lcxw9b)**8**

**5**[**. Mô phỏng giao diện các chức năng 29**](#_rn1rbtozj64u)

5[.1. Đăng ký/Đăng nhập 29](#_aa496y2dpt3y)

5[.2. Tìm kiếm luật sư 31](#_zaob8hrk7t9p)

5[.3. Lịch hẹn 31](#_hiums5cdseso)

5[.4. Gửi phản hồi 33](#_sr8o6z309409)

5[.4.1. Gửi khiếu nại 33](#_kuznqzwkgaks)

5[.4.2. Feedback 33](#_6o8k49rso3yk)

5[.5. Quản lý tài khoản 34](#_e3do30evx059)

5[.6. Admin 34](#_ilyjs17ojqa9)

**6**[**. Giao diện ngoài**](#_n3n4jff44s42) **34**

# **Kiến trúc hệ thống**

## ***Vấn đề đặt ra***

Trong quá trình lựa chọn kiến trúc cho hệ thống của mình, chúng tôi đã xác định các vấn đề chính mà chúng tôi cần giải quyết để đảm bảo hiệu suất, bảo trì và mở rộng của hệ thống đó là:

* Vấn đề phân tách trách nhiệm và quản lý logic: Hệ thống của chúng tôi đang phát triển với nhiều tính năng phức tạp và các tính năng lại có phần trách nhiệm và quản lý logic khác nhau. Vấn đề đặt ra là làm thế nào để phân tách rõ ràng các trách nhiệm và quản lý logic của các thành phần khác nhau. Chúng tôi cần một cách để tổ chức mã nguồn sao cho dễ dàng bảo trì, tái sử dụng và kiểm thử.
* Vấn đề tách biệt giao diện và xử lý logic: Giao diện người dùng của chúng tôi cần phản ánh trạng thái dữ liệu hiện tại và phản hồi tương tác người dùng, Tuy nhiên, chúng tôi muốn tách biệt giao diện và xử lý logic để dễ dàng thay đổi giao diện mà không ảnh hưởng đến logic và ngược lại. Điều này giúp chúng tôi phát triển và bảo trì ứng dụng một cách linh hoạt hơn.
* Vấn đề tương tác dữ liệu đồng bộ và bất đồng bộ: Ứng dụng của chúng tôi tương tác với cơ sở dữ liệu để lấy và cập nhật dữ liệu. Việc quản lý tương tác này một cách hiệu quả và đảm bảo tính nhất quán của dữ liệu trở thành một thách thức. Chúng tôi cần một cơ chế để xử lý tương tác dữ liệu đồng bộ và bất đồng bộ, đồng thời đảm bảo tính toàn vẹn và khả năng phục hồi khi có lỗi xảy ra.

Và với việc sử dụng MERN, chúng tôi áp dụng mô hình Client - Server cho hệ thống của mình và kiến trúc MVC (Model - View - Controller) cho phần backend.

## ***Kiến trúc MVC cho*** *b****ackend***

Mô hình kiến trúc MVC được chia làm 3 phần:

* Model (Mô hình): Model đại diện cho việc quản lý và xử lý logic liên quan đến dữ liệu trong hệ thống. Chúng tôi sử dụng MongoDB để lưu trữ và thực hiện xử lý dữ liệu. Các dữ liệu của hệ thống chúng tôi bao gồm: thông tin đăng nhập người dùng, thông tin về cuộc hẹn, hồ sơ của luật sư . Model cung cấp các phương thức và thao tác để truy xuất, thêm, sửa, xóa dữ liệu từ cơ sở dữ liệu.
* View (Giao diện): Trong backend, không có một thành phần View cụ thể như trong phần frontend. Thay vào đó, các thông tin được trả về từ server thông qua các route có thể được hiển thị lên các trình duyệt thông qua React.
* Controller (Bộ điều khiển): Trong backend, các file controller thường đảm nhận vai trò của Controller trong mô hình MVC. Chúng xử lý các yêu cầu từ client và điều hướng chúng đến các phương thức của model để thực hiện các thao tác trên dữ liệu, sau đó trả về kết quả cho client thông qua các route. Các file route định nghĩa các đường dẫn URL và liên kết chúng với các phương thức trong controller.

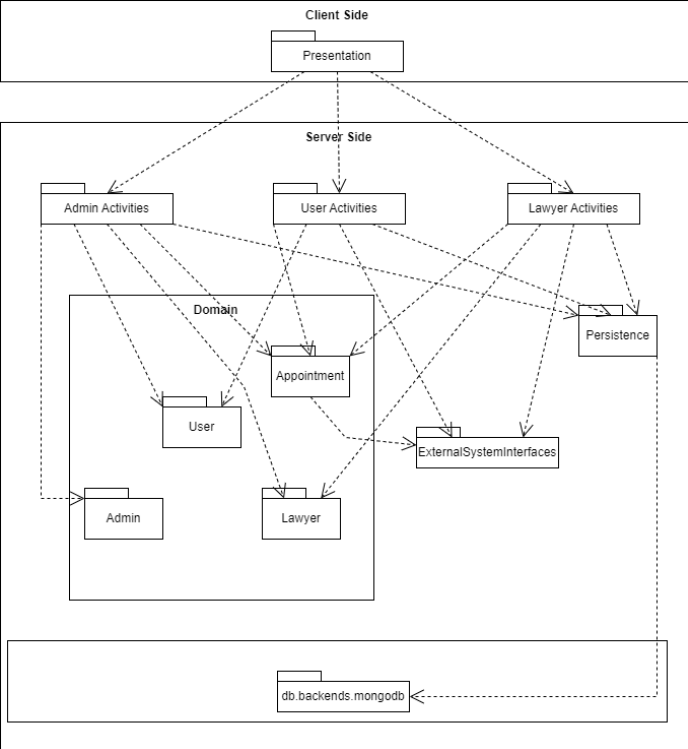
Bằng việc sử dụng kiến trúc MVC với 3 thành phần tách biệt, và phối hợp cùng nhau, từ đó cho phép chúng tôi dễ dàng bảo trì, sửa đổi và mở rộng hệ thống trong tương lai.

## *Kiến trúc Client - Server*

* **Client-Side (Frontend):** React.js: Đây là nơi xây dựng giao diện người dùng (UI) và xử lý các tương tác người dùng. Trong mô hình này, các yêu cầu từ người dùng được gửi đến máy chủ để xử lý, và kết quả được trả về để hiển thị trên trình duyệt.
* **Server-Side (Backend):** Node.js với Express.js: Đây là nơi xử lý các yêu cầu từ phía client, thường là các yêu cầu HTTP. Backend có thể thực hiện các hoạt động như xử lý dữ liệu, truy xuất cơ sở dữ liệu (MongoDB), xác thực người dùng, và trả về dữ liệu cho frontend.
* **Database:** MongoDB: Dùng để lưu trữ và quản lý dữ liệu. Trong trường hợp này, MongoDB thường được sử dụng với Node.js thông qua các thư viện như Mongoose để tương tác với cơ sở dữ liệu.

## *Biểu đồ thành phần hệ thống*

Biểu đồ các thành phần hệ thống của chúng tôi:

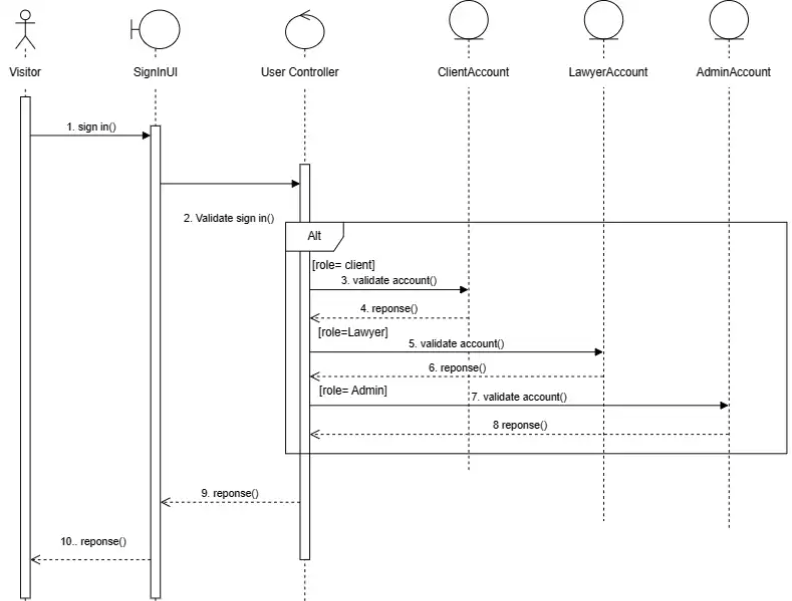


# **Biểu đồ tuần tự các ca sử dụng chính**

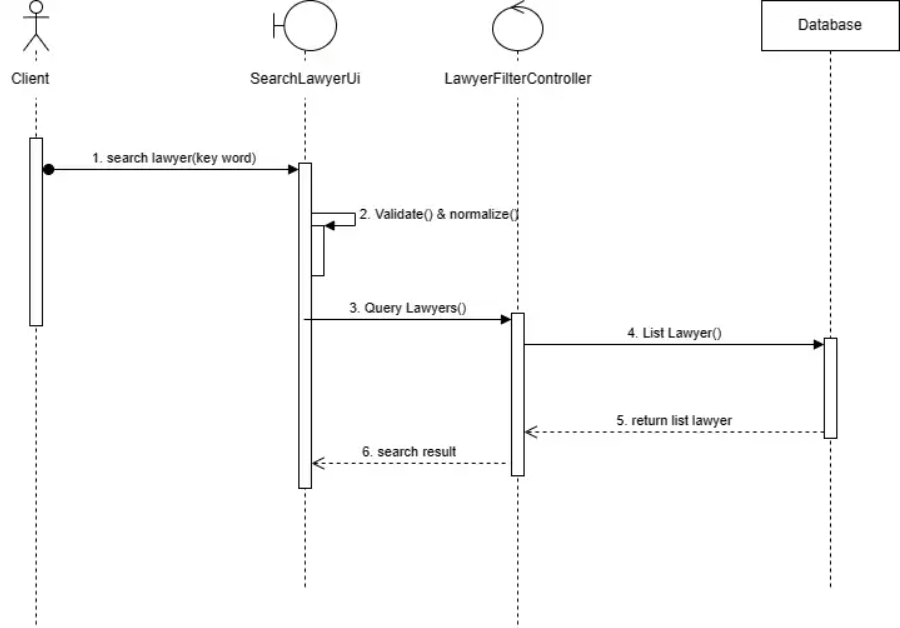
## ***Đăng ký***

## 

## ***Đăng nhập***

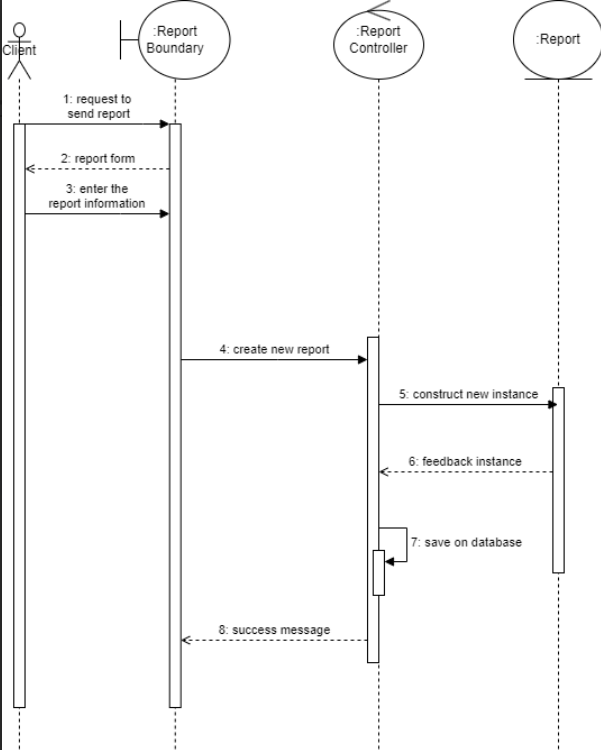


## ***Tìm kiếm và xem hồ sơ luật sư***

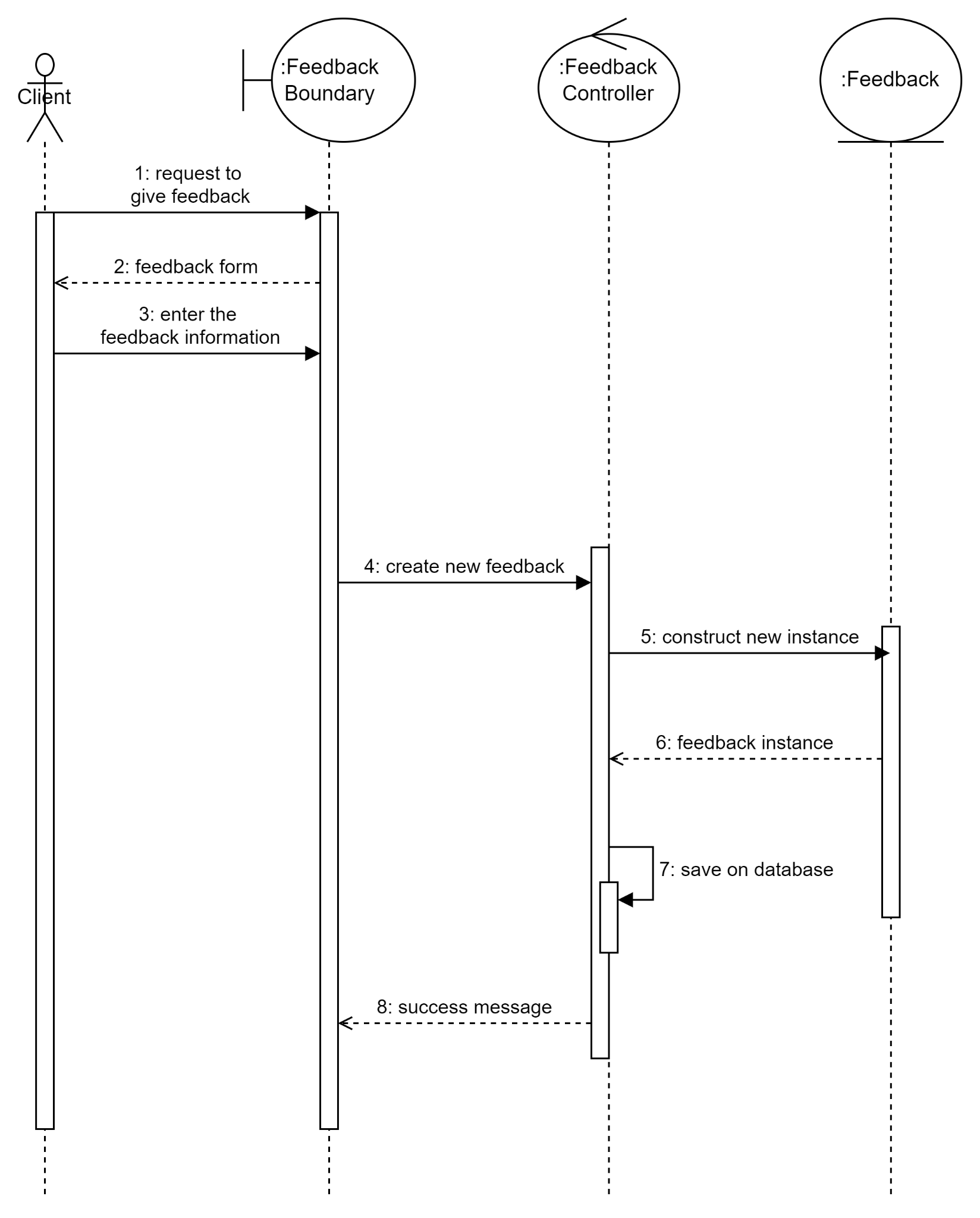


## ***Quản lý cuộc h****ẹn*

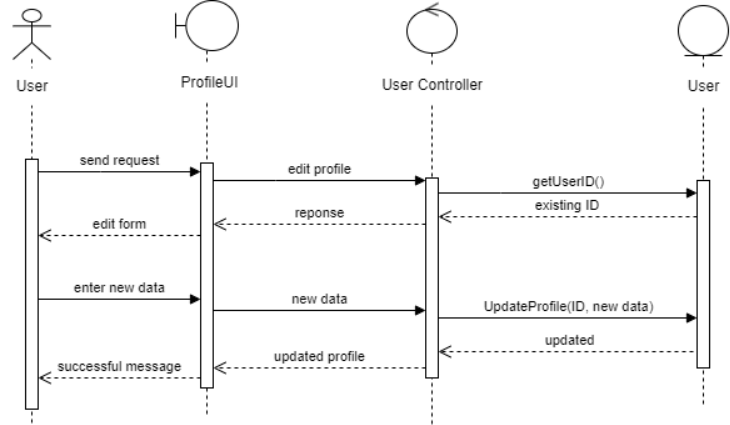
## ***Report***



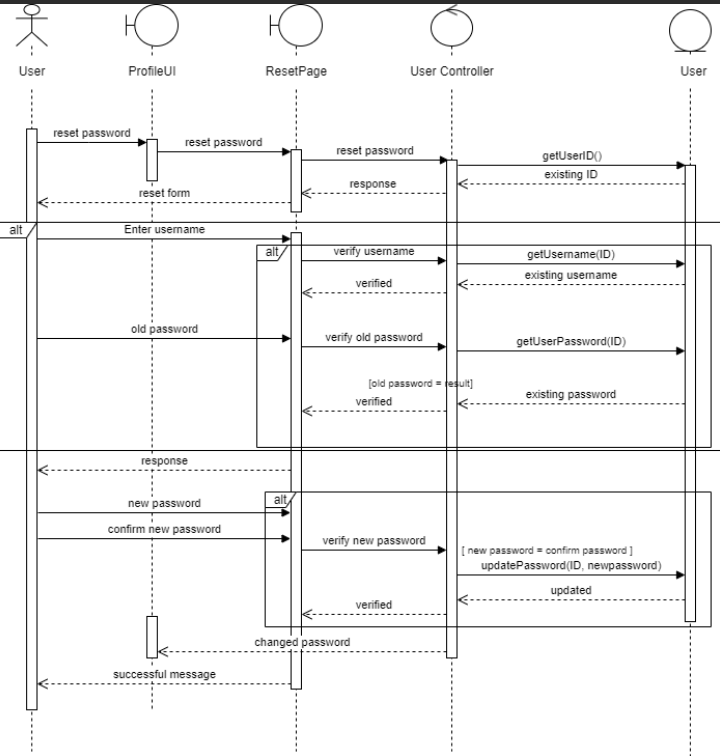
## ***Đánh giá, phản hồi về luật sư***

******

## ***Xem và chỉnh sửa thông tin***

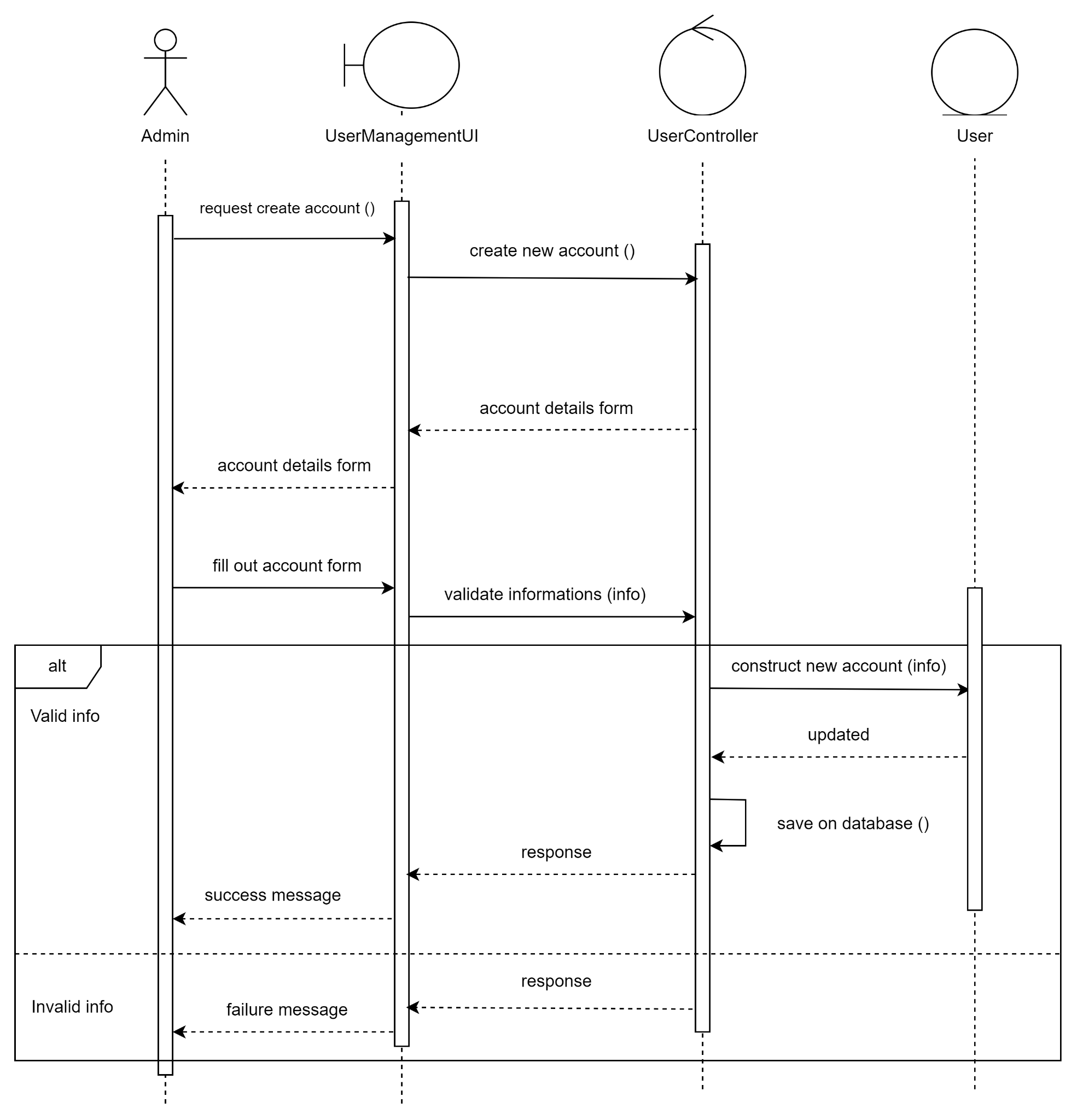


## ***Thay đổi mật khẩu***

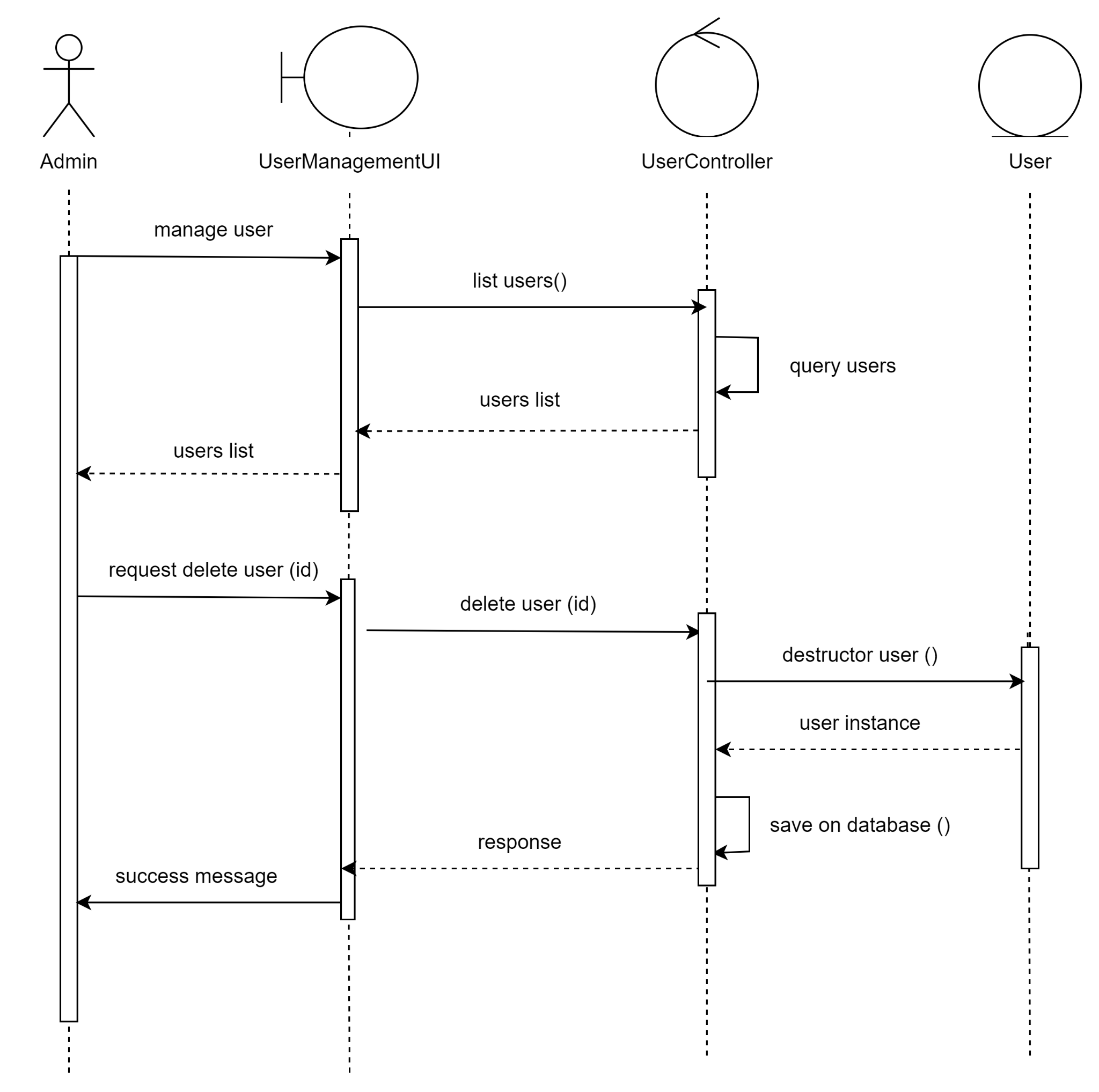


## ***Admin quản lý người dùng và trang web***

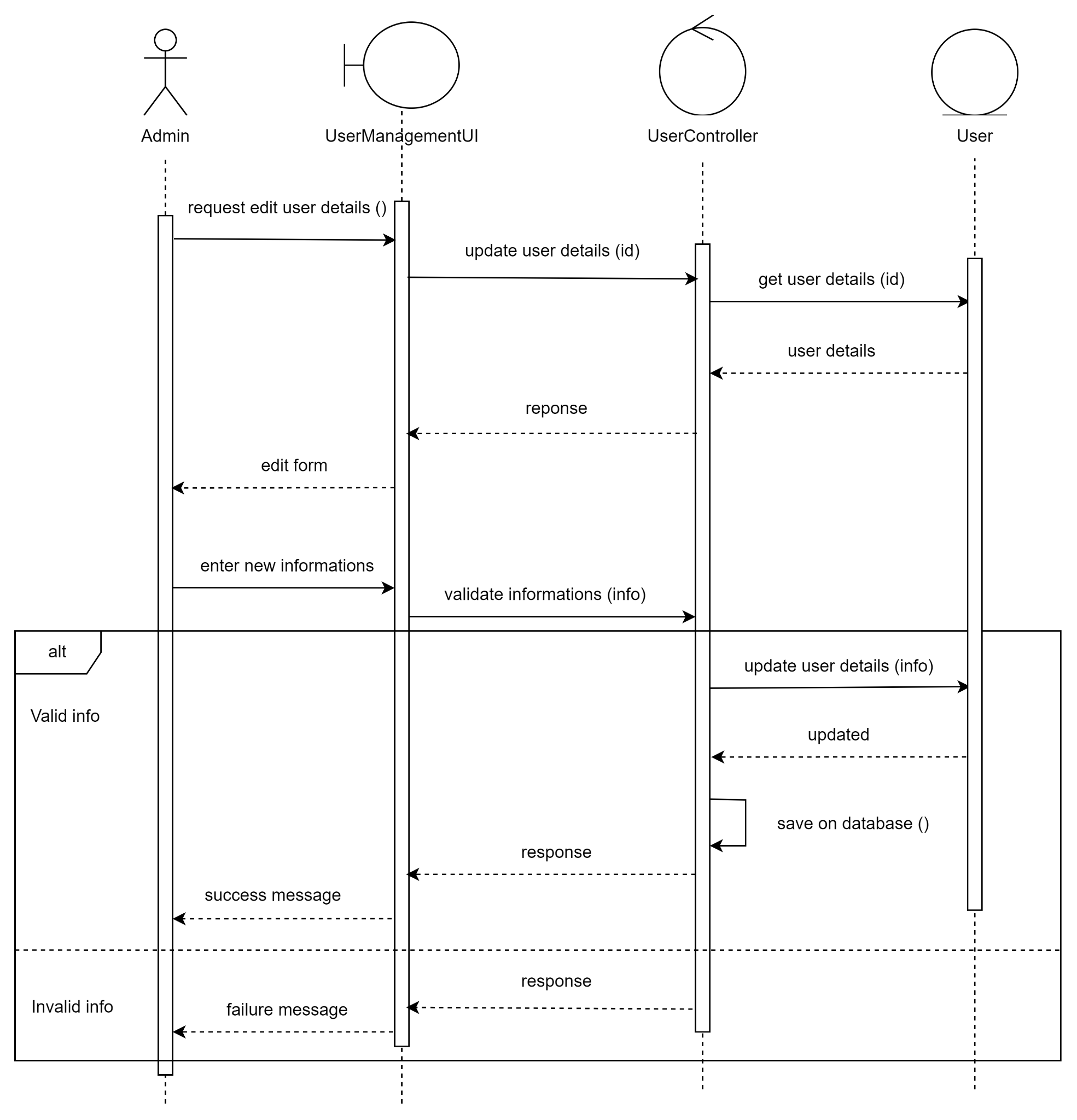
### ***Tạo tài khoản***



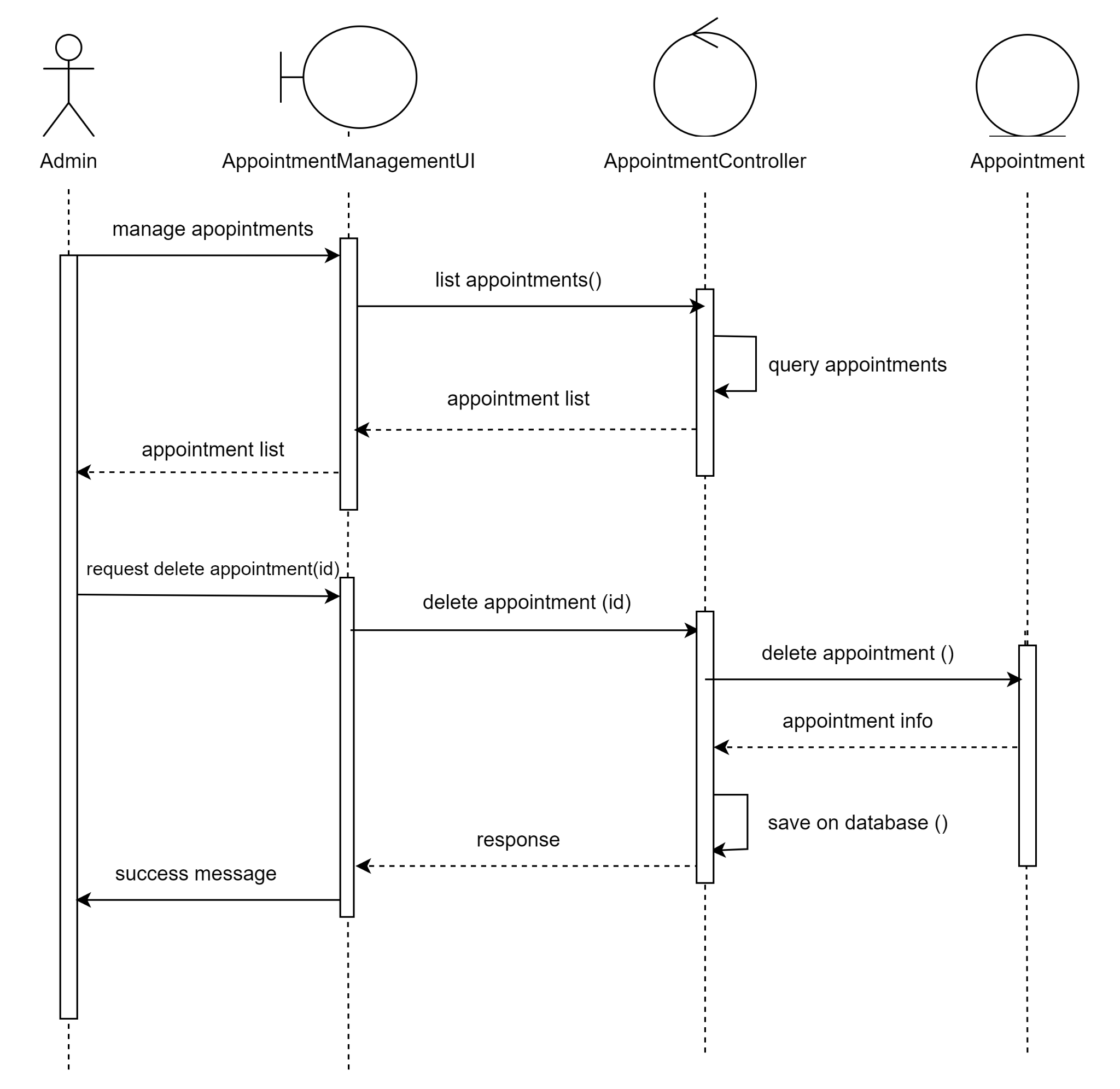
### ***Xóa tài khoản***

******

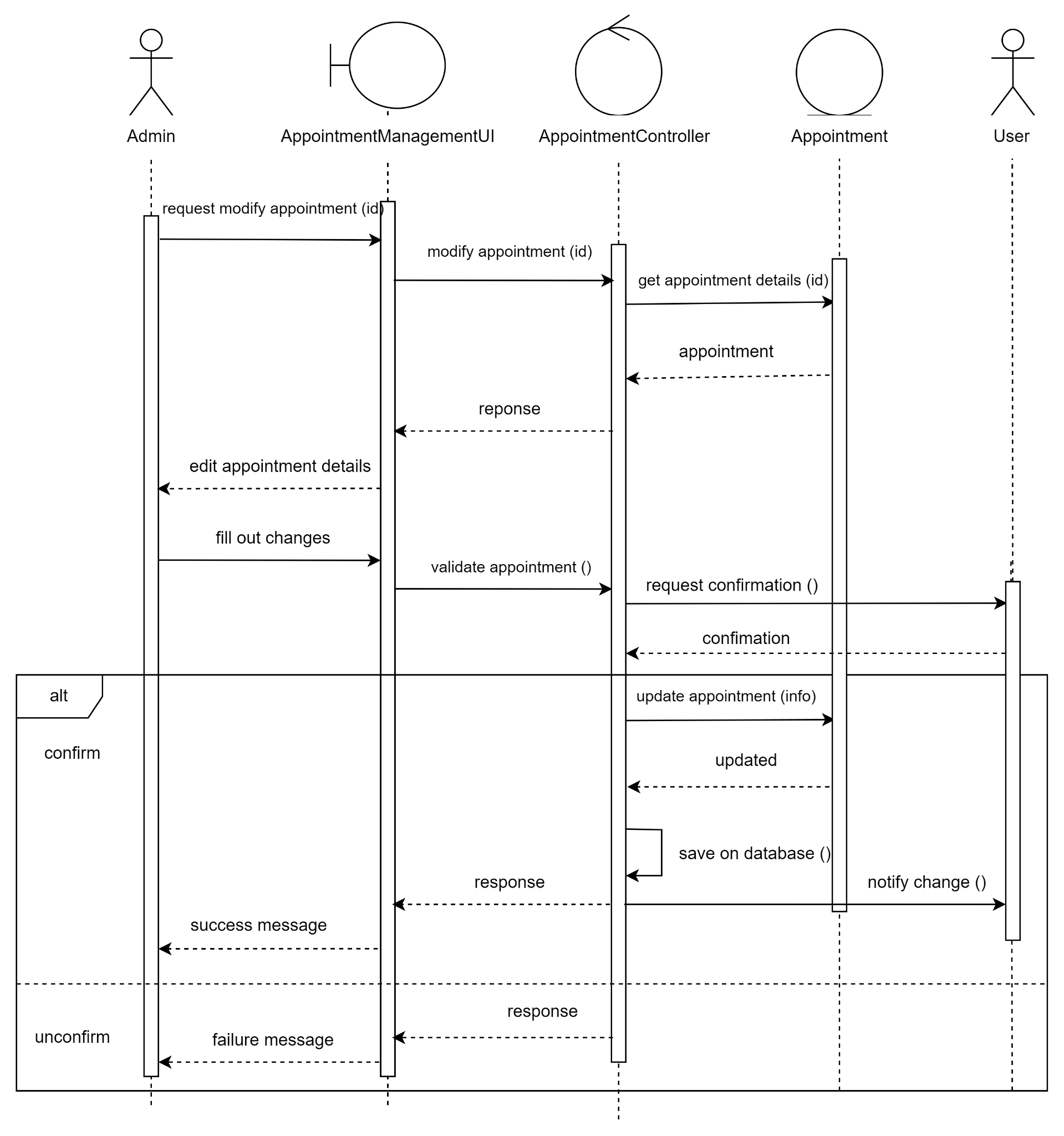
### ***Cập nhật tài khoản***

******

### ***Xóa cuộc hẹn***

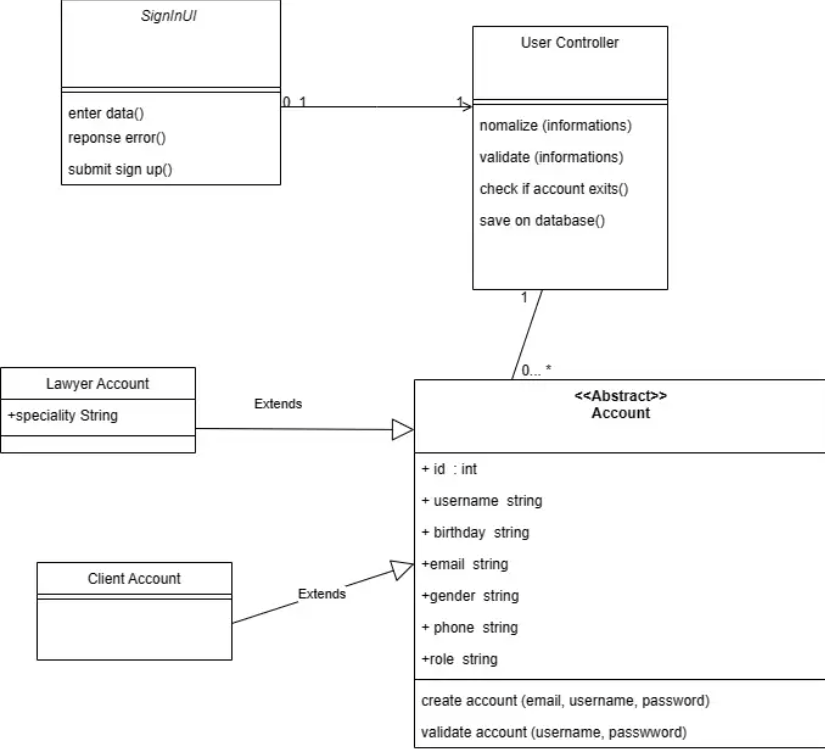
******

### ***Chỉnh sửa cuộc hẹn***

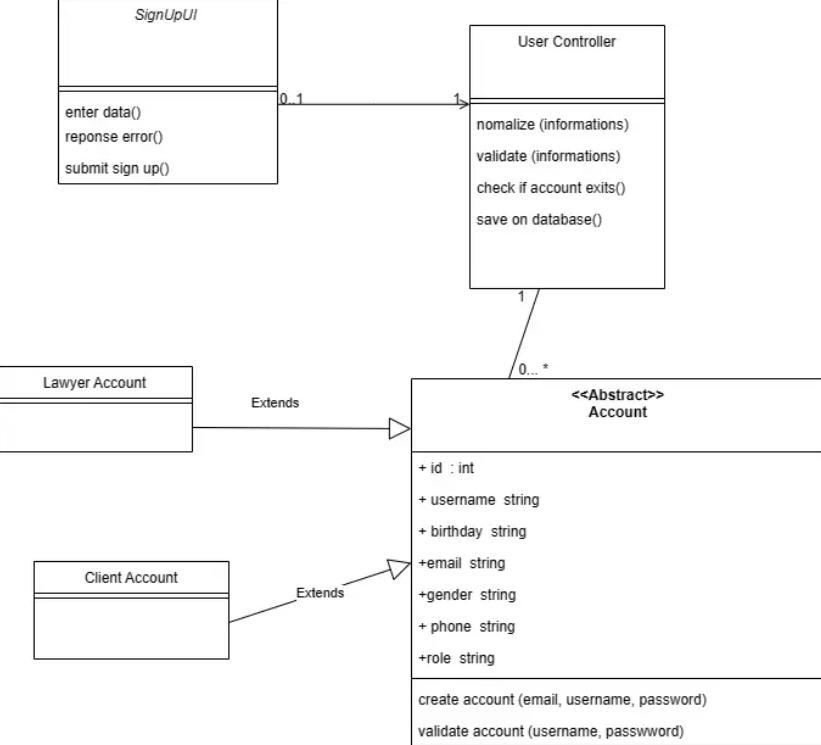
******

# **Biểu đồ lớp các ca sử dụng**

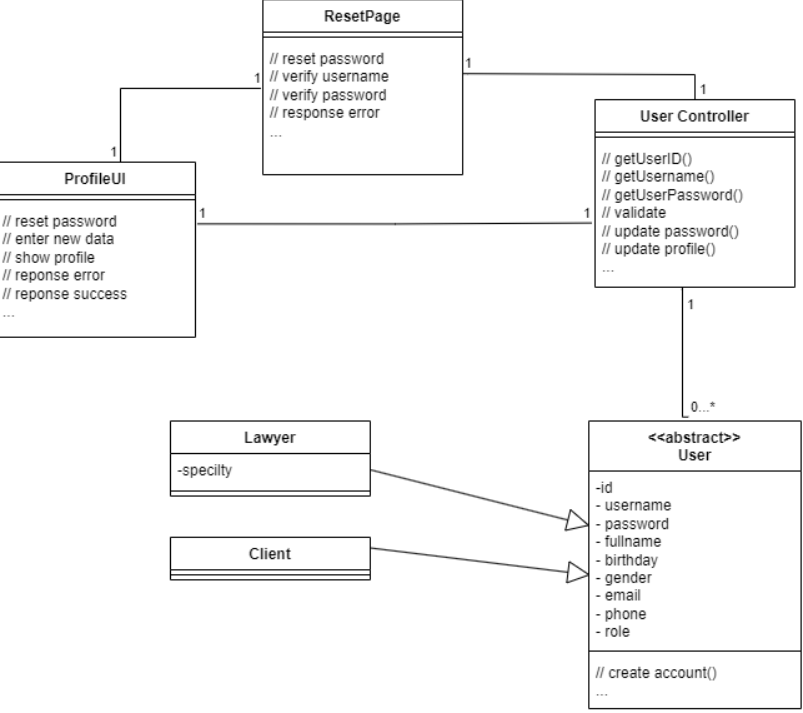
## ***Đăng nhập***

******

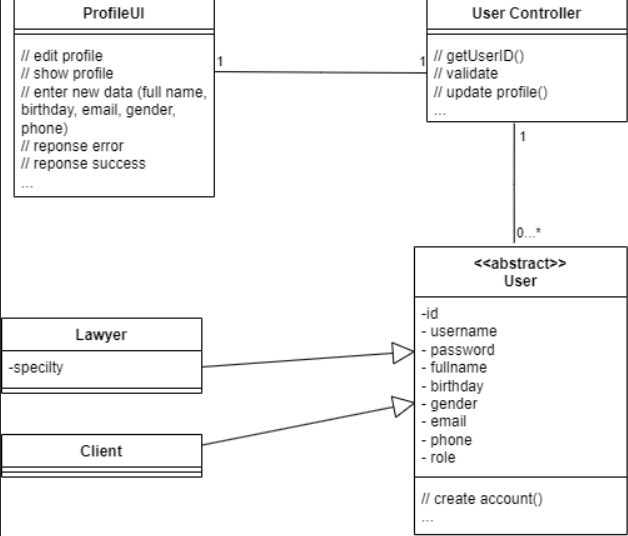
## ***Đăng ký***

******

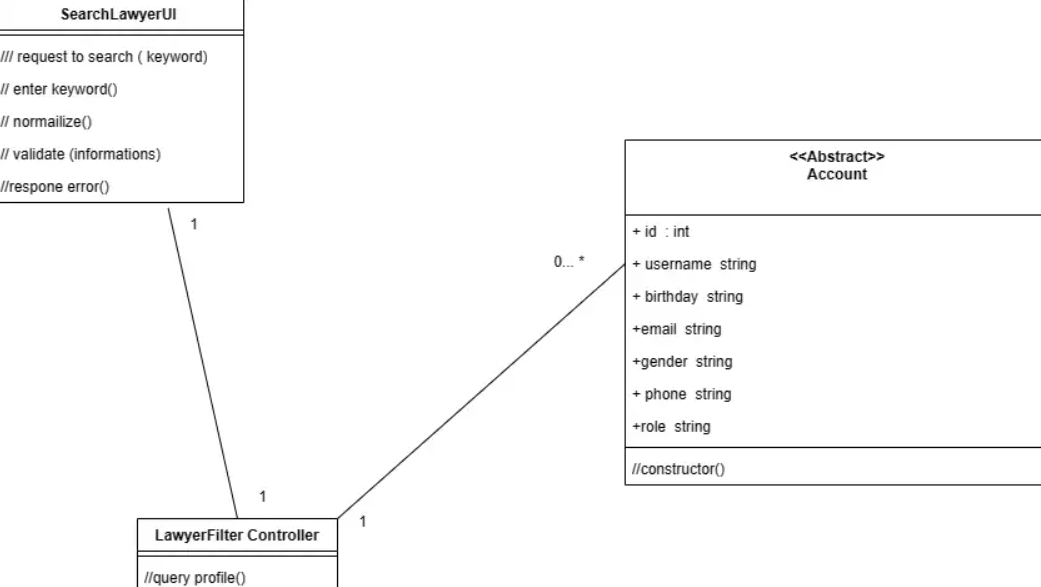
## ***Đổi mật khẩu***

******

## ***Cập nhật thông tin***

******

## ***Tìm kiếm luật sư***

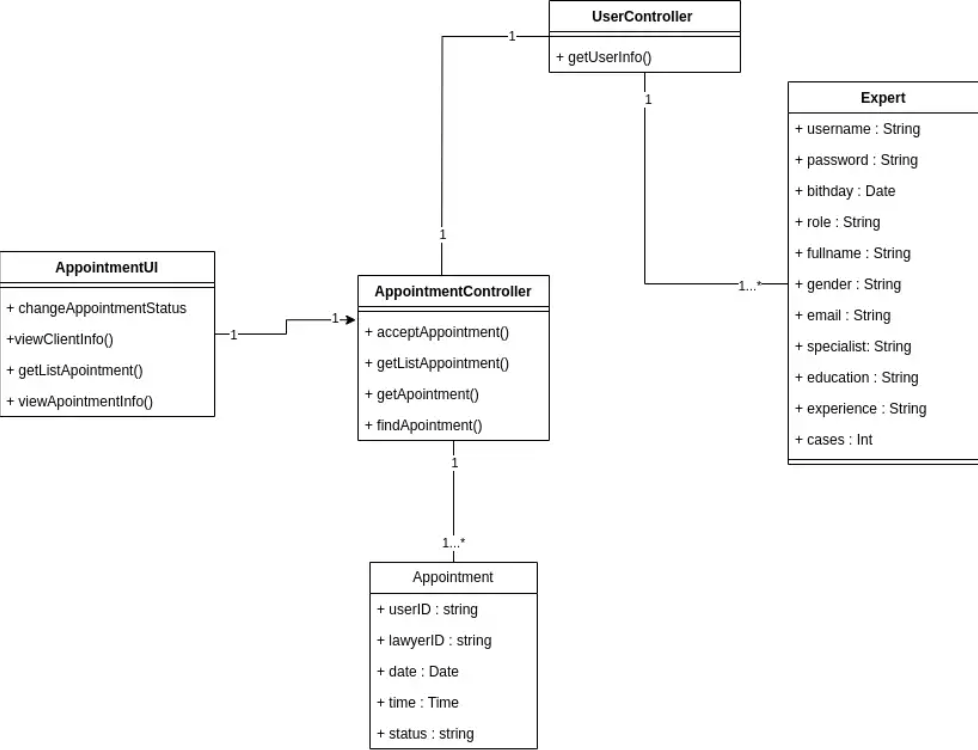
******

## ***Xem thông tin luật sư***

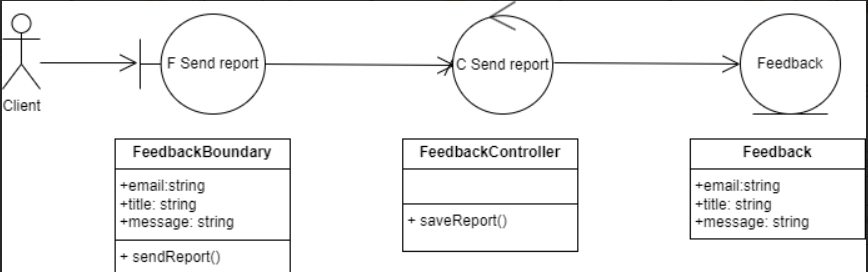
******

## ***Quản lý cuộc hẹn***

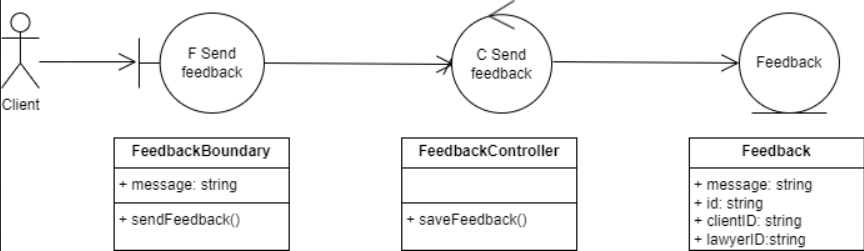
## 



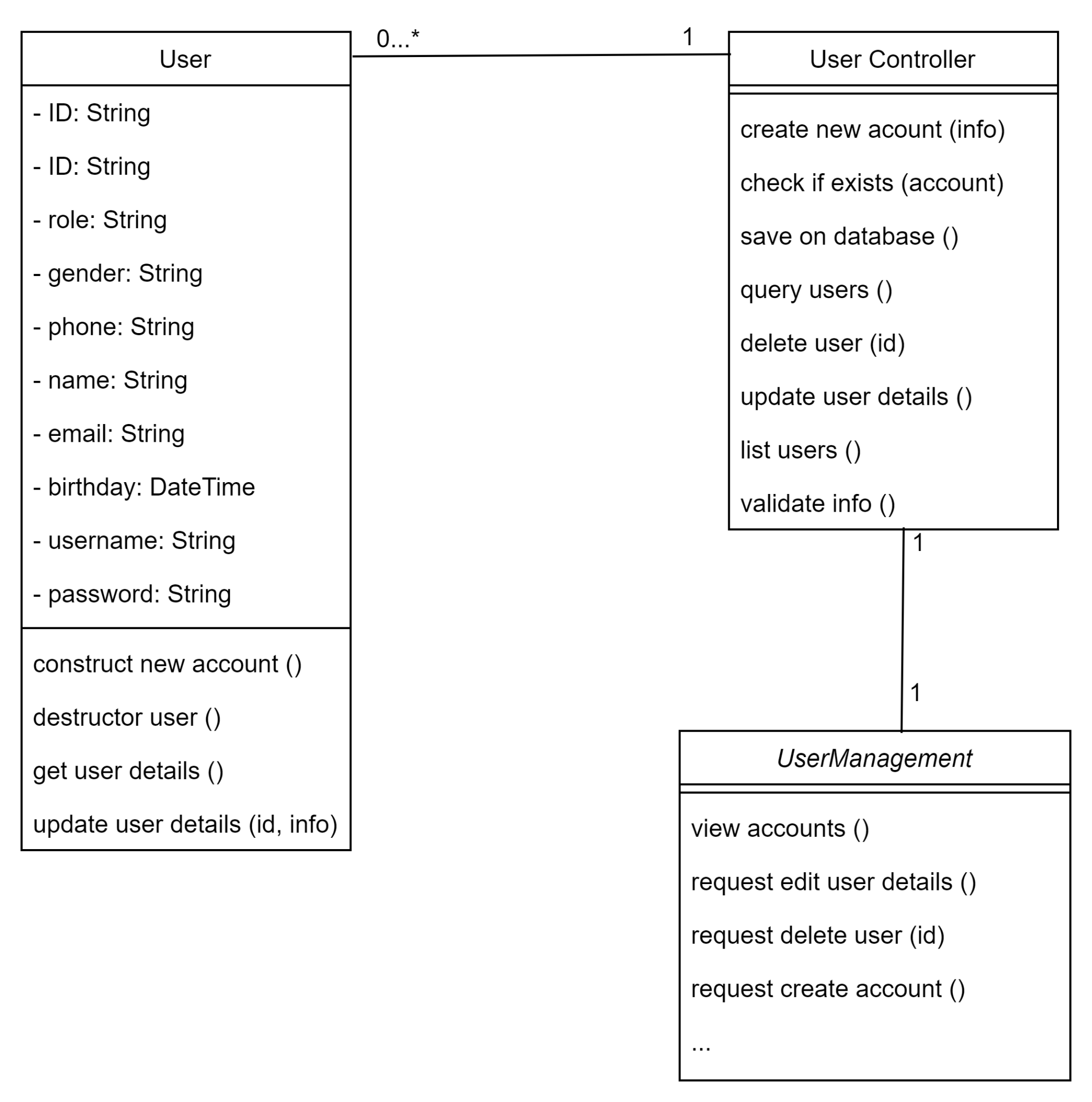
## ***Report***

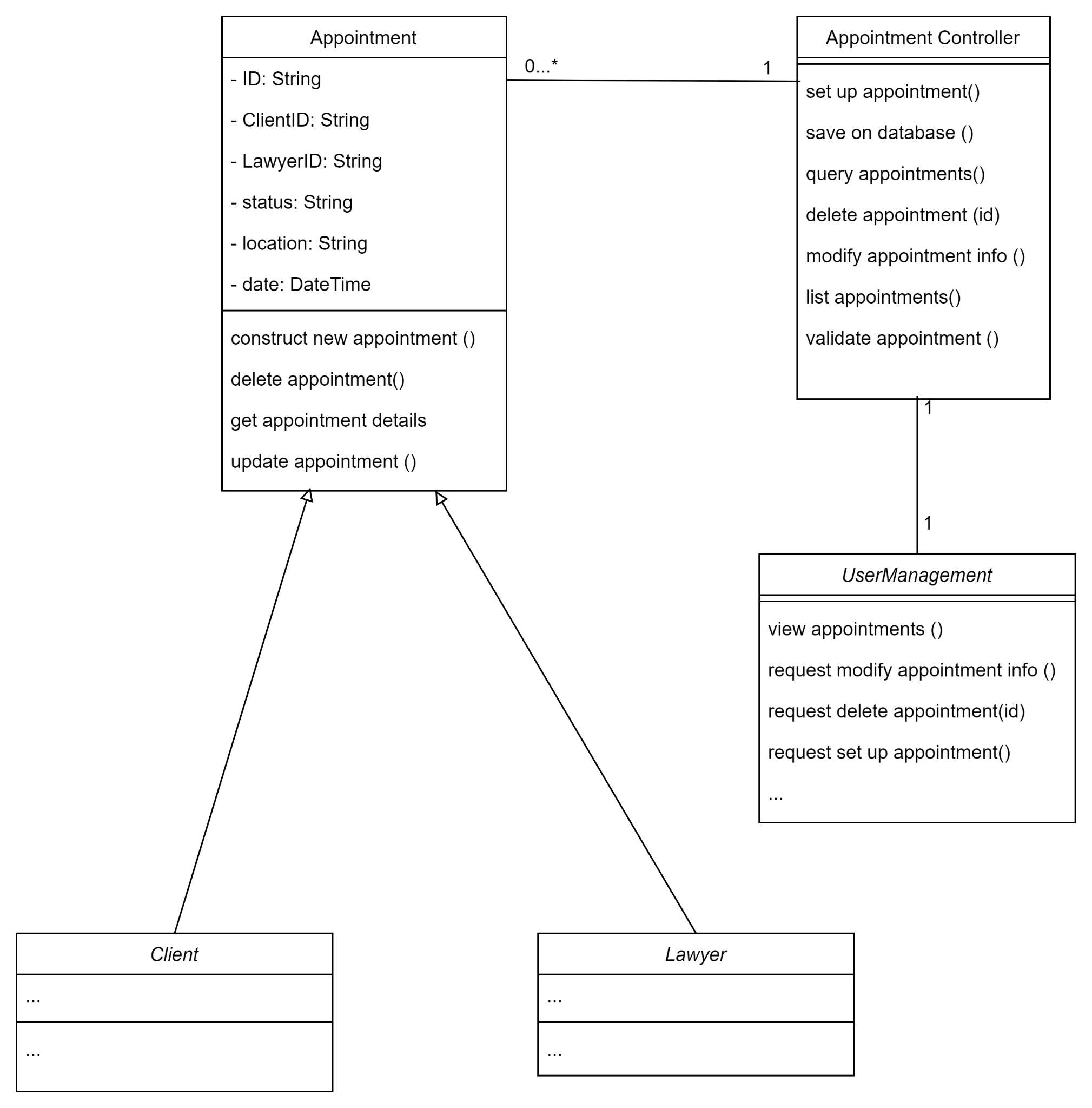
******

## ***Đánh giá, phản hồi về luật sư***

******

## ***Admin quản lý người dùng và trang web***





# **Thiết kế cơ sở dữ liệu**

## ***Giới thiệu***

MongoDB là một hệ thống cơ sở dữ liệu NoSQL (không quan hệ) phổ biến được sử dụng cho các ứng dụng web và dịch vụ đòi hỏi việc lưu trữ dữ liệu linh hoạt và có thể mở rộng.

MongoDB sử dụng mô hình dữ liệu JSON-style document, cho phép bạn lưu trữ dữ liệu dưới dạng document và nested data structures mà không cần phải tuân thủ một schema cố định như trong hệ thống quan hệ.

Cung cấp các tính năng mạnh mẽ cho việc truy vấn dữ liệu bằng cách sử dụng các câu lệnh query linh hoạt và hiệu quả.

## ***Thiết kế***

Thực thể User là nơi lưu trữ thông tin của khách hàng

| Field | Ý nghĩa |
| --- | --- |
| \_id | khóa chính |
| username | tài khoản đăng nhập |
| password | mật khẩu đăng nhập |
| createdAt | ngày tạo tài khoản |
| birthday | ngày sinh |
| role | vai trò |
| fullname | họ và tên |
| email | email |
| gender | giới tính |
| phone | số điện thoại |

Thực thể Expert là nơi lưu trữ thông tin của chuyên gia (luật sư)

| Field | Ý nghĩa |
| --- | --- |
| \_id | khóa chính |
| username | tài khoản đăng nhập |
| password | mật khẩu đăng nhập |
| createdAt | ngày tạo tài khoản |
| birthday | ngày sinh |
| role | vai trò |
| fullname | họ và tên |
| email | email |
| gender | giới tính |
| phone | số điện thoại |
| experience | kinh nghiệm trong nghề |
| education | trình độ học vấn |
| specialist | chuyên môn |
| rate | đánh giá |

Thực thể Appointment là nơi lưu trữ thông tin cuộc hẹn giữa chuyên gia và khách hàng

| Field | Ý nghĩa |
| --- | --- |
| \_id | khóa chính |
| user | id của khách hàng |
| lawyer | id của chuyên gia |
| date | ngày hẹn |
| time | thời gian cuộc hẹn |
| status | trạng thái cuộc hẹn(chờ xác nhận/ đã xác nhận/ đã xong/ đã hủy) |

Thực thể Report là nơi lưu trữ báo cáo của khách hàng về hệ thống, trang web, chất lượng dịch vụ hoặc yêu cầu hỗ trợ

| Field | Ý nghĩa |
| --- | --- |
| \_id | khóa chính |
| user | id của khách hàng |
| content | nội dung của báo cáo |

Thực thể Feedback là nơi lưu trữ đánh giá, phản hồi của khách hàng về luật sư

| Field | Ý nghĩa |
| --- | --- |
| \_id | khóa chính |
| user | id của khách hàng |
| lawyer | id của chuyên gia |
| content | nội dung của bài đánh giá |
| rate | đánh giá chuyên gia theo thang điểm |

## Tổng kết

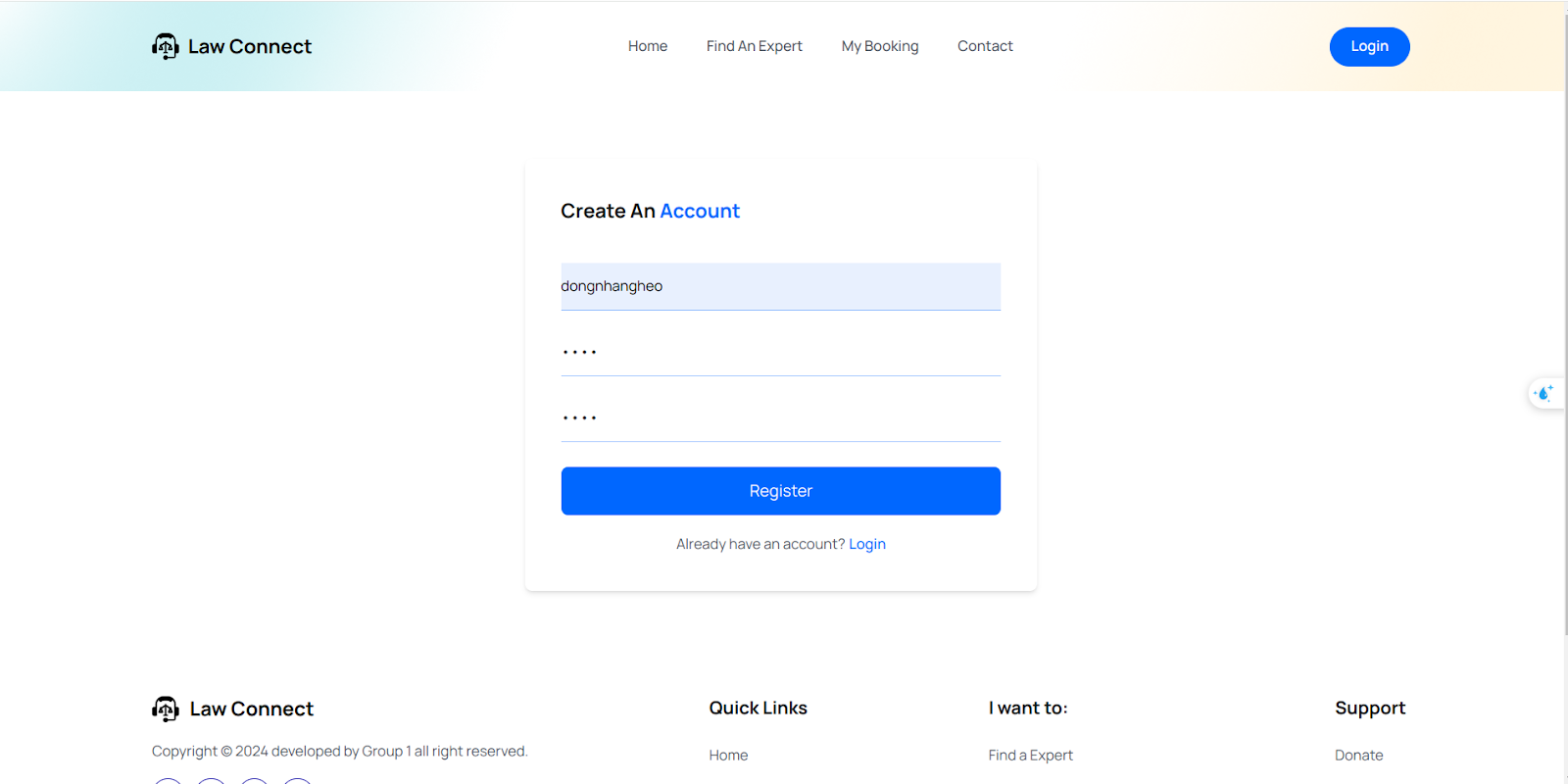


# **Mô phỏng giao diện các chức năng**

## Đăng ký/Đăng nhập

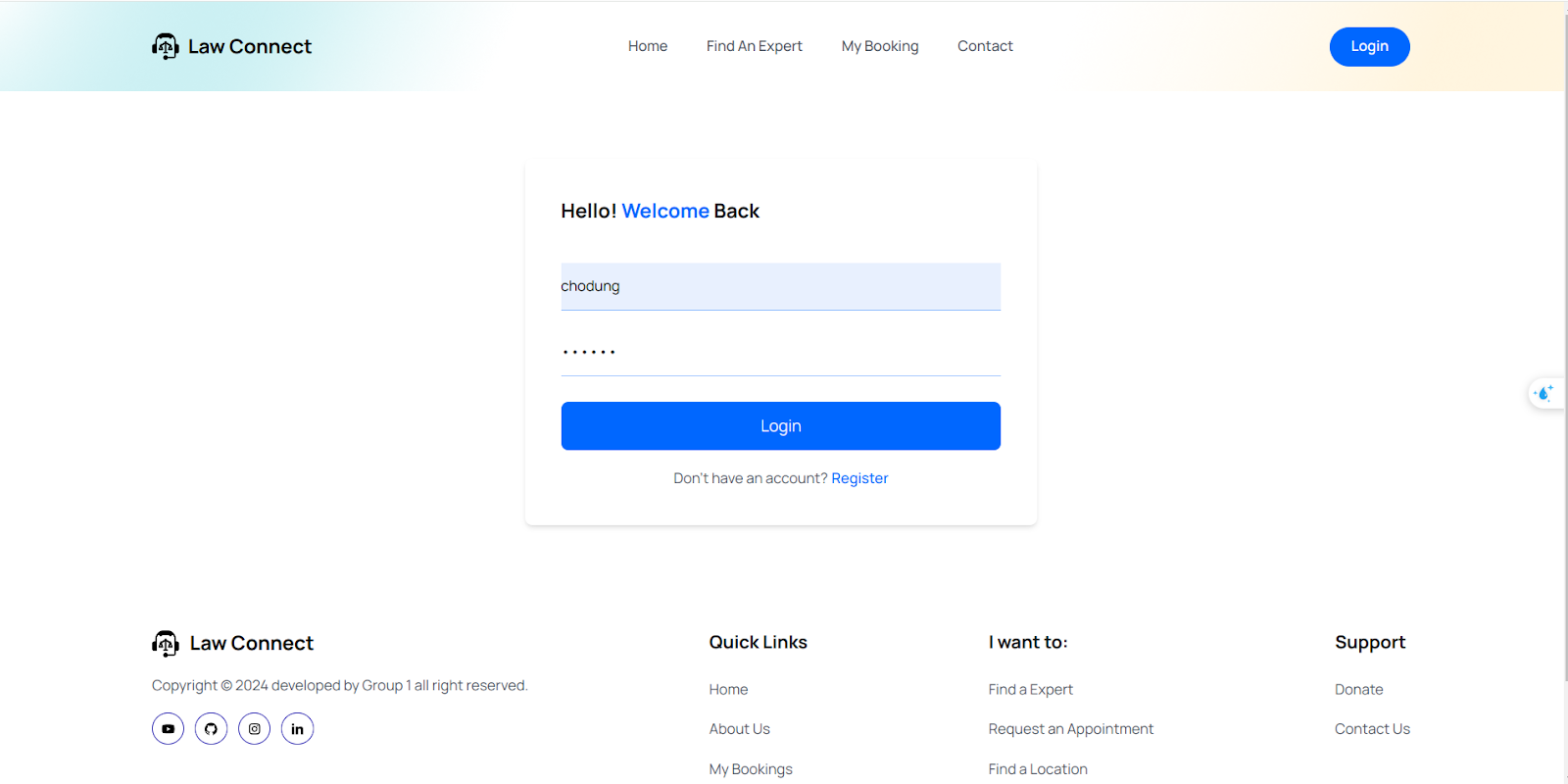
Để có thể sử dụng toàn bộ các chức năng chính của Web, người dùng cần phải có tài khoản để đăng nhập, hoặc đăng ký (nếu chưa có tài khoản)

* **Đối với đăng ký**: Người dùng sẽ cần điền đầy đủ thông tin: Username (tên để hiển thị trong suốt quá trình dùng), Password.



*Giao diện đăng ký tài khoản*

* **Đối với đăng nhập**: Sau khi đã tạo tài khoản thành công hoặc đã có tài khoản trước đó, người dùng cần đăng nhập để sử dụng mọi tính năng của Web. Người dùng cần điền username và mật khẩu của tài khoản đó.

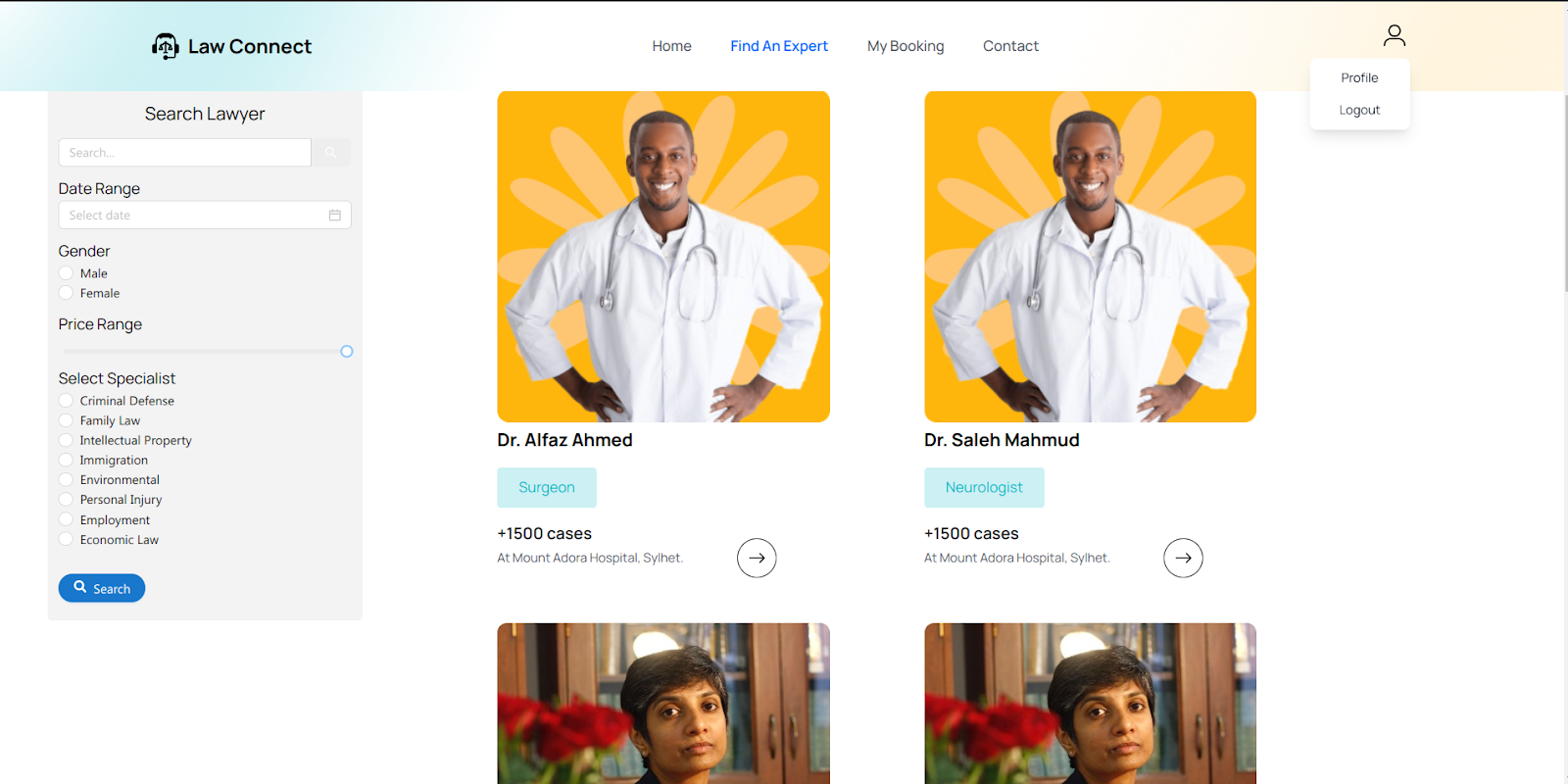


*Giao diện đăng nhập*

## Tìm kiếm luật sư

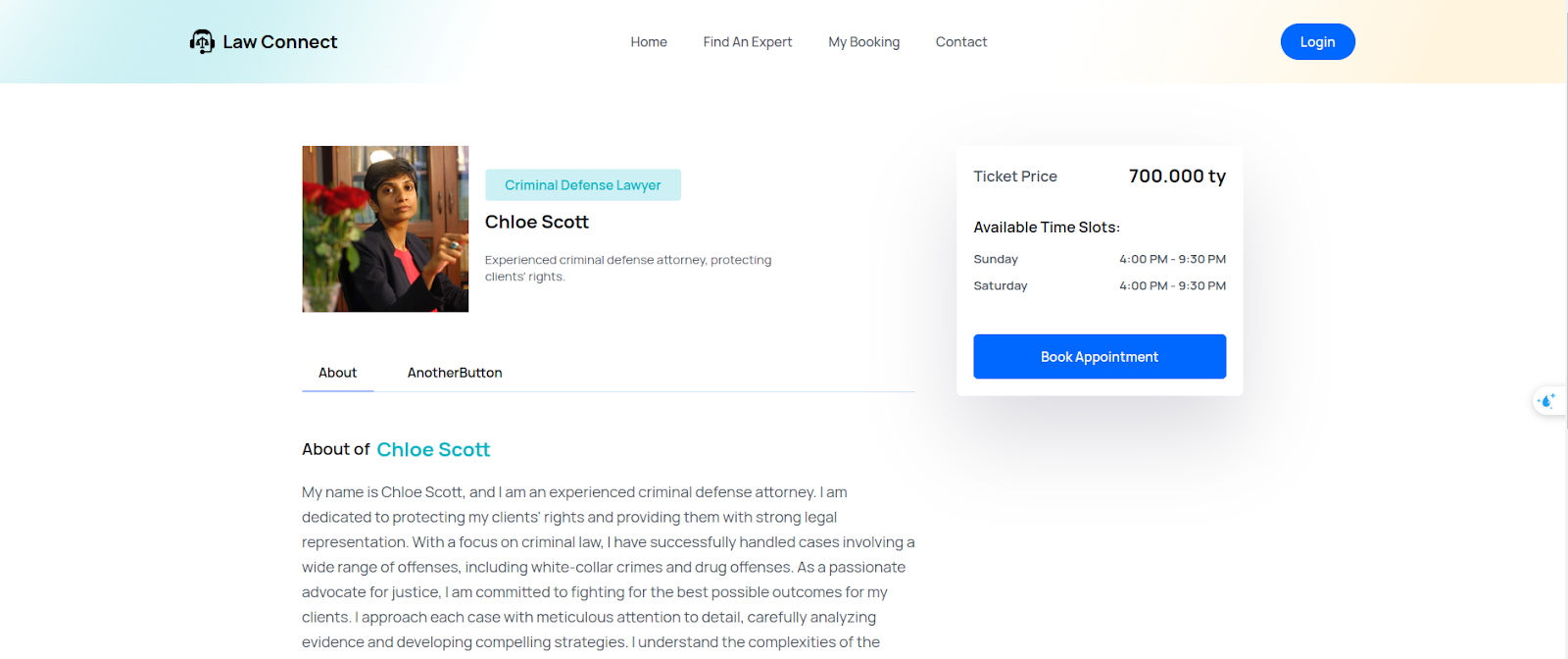
Người dùng tìm kiếm, lựa chọn và xem thông tin luật sư.

Bấm vào nút ‘Find An Expert' để chuyển tới giao diện tìm kiếm. Chức năng này có thể thực hiện kể cả khi chưa đăng nhập.



*Giao diện tìm kiếm*

Nhấp chuột vào nút mũi tên để thực hiện xem thông tin chi tiết về một luật sư. Thông tin về luật sư sẽ hiện ra trên trang.

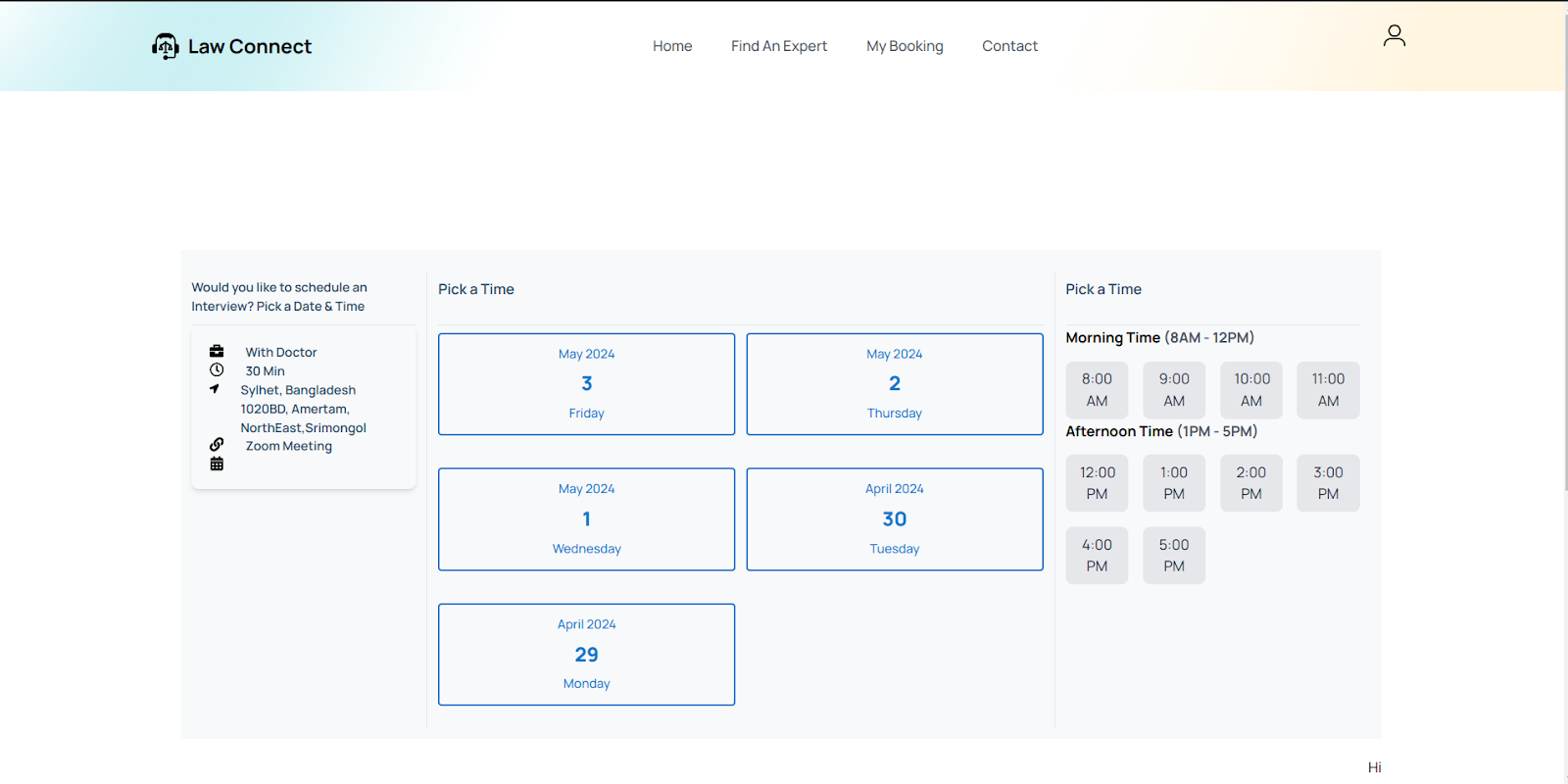
**

*Giao diện xem thông tin luật sư*

## Lịch hẹn

* + 1. ***Đặt lịch hẹn***

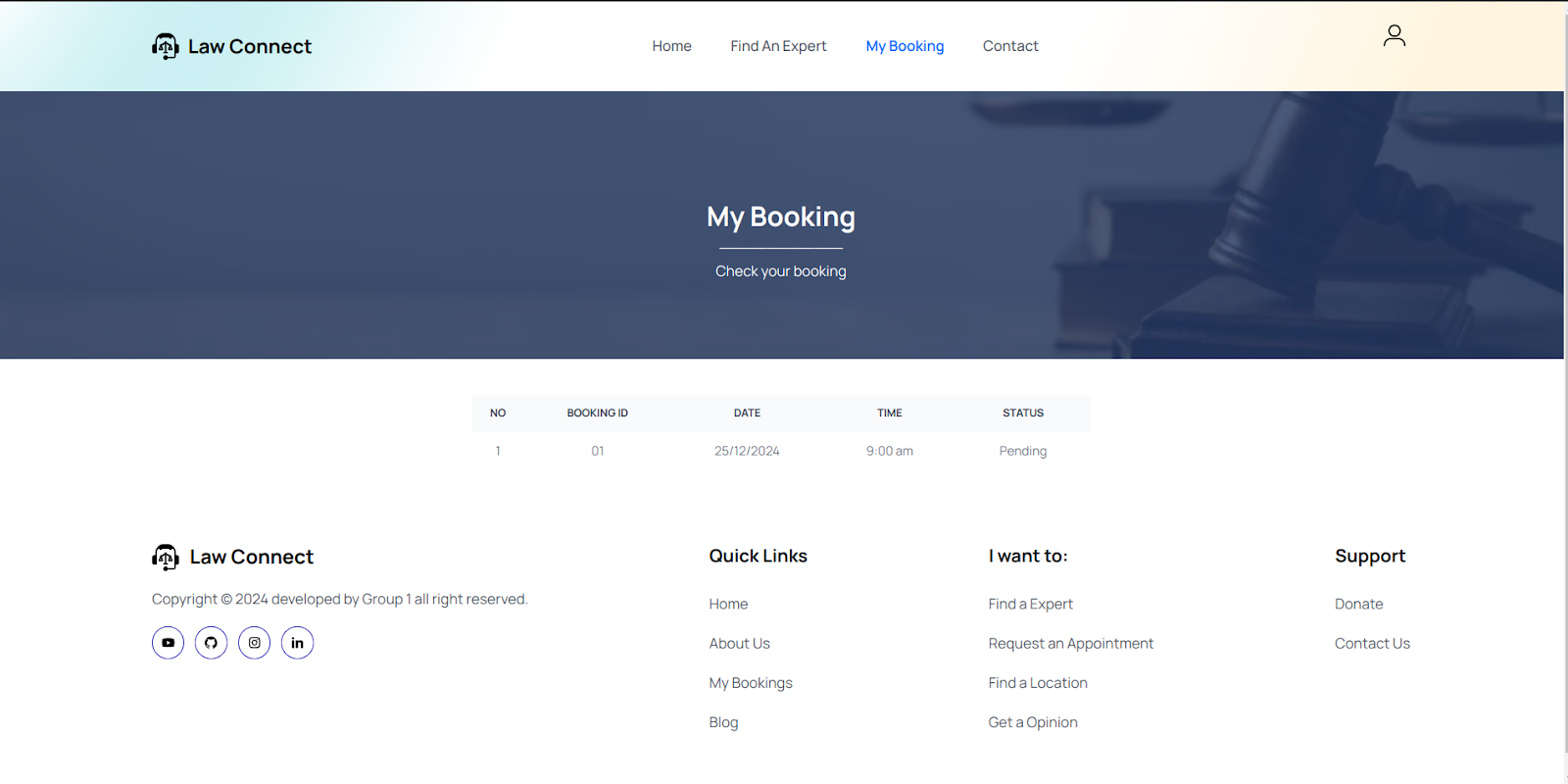
Sau khi bấm vào giao diện xem thông tin luật sư, bấm nút “Book Appointment” ở bên phải để tiếp tục đến phần đặt lịch hẹn. Trong phần đặt lịch hẹn, khách hàng sẽ được lựa chọn ngày và giờ mong muốn.



*Giao diện đặt lịch hẹn*

* + 1. ***Xem danh sách cuộc hẹn***

Khi đã đặt xong cuộc hẹn, ta có thể xem lại một danh sách cuộc hẹn của mình bằng cách bấm vào “My Booking” hoặc “My Appointment" (của luật sư)



*Giao diện xem thông tin cuộc hẹn*

* + 1. ***Xem thông tin cuộc hẹn***

Bấm vào một cuộc hẹn trong danh sách để xem thông tin chi tiết về cuộc hẹn.

* + 1. ***Thay đổi trạng thái cuộc hẹn***

Tại trang xem danh sách các cuộc hẹn, luật sư có thể thay đổi trạng thái các cuộc hẹn của khách hàng: pending (chờ), accept (đồng ý), reject (từ chối).

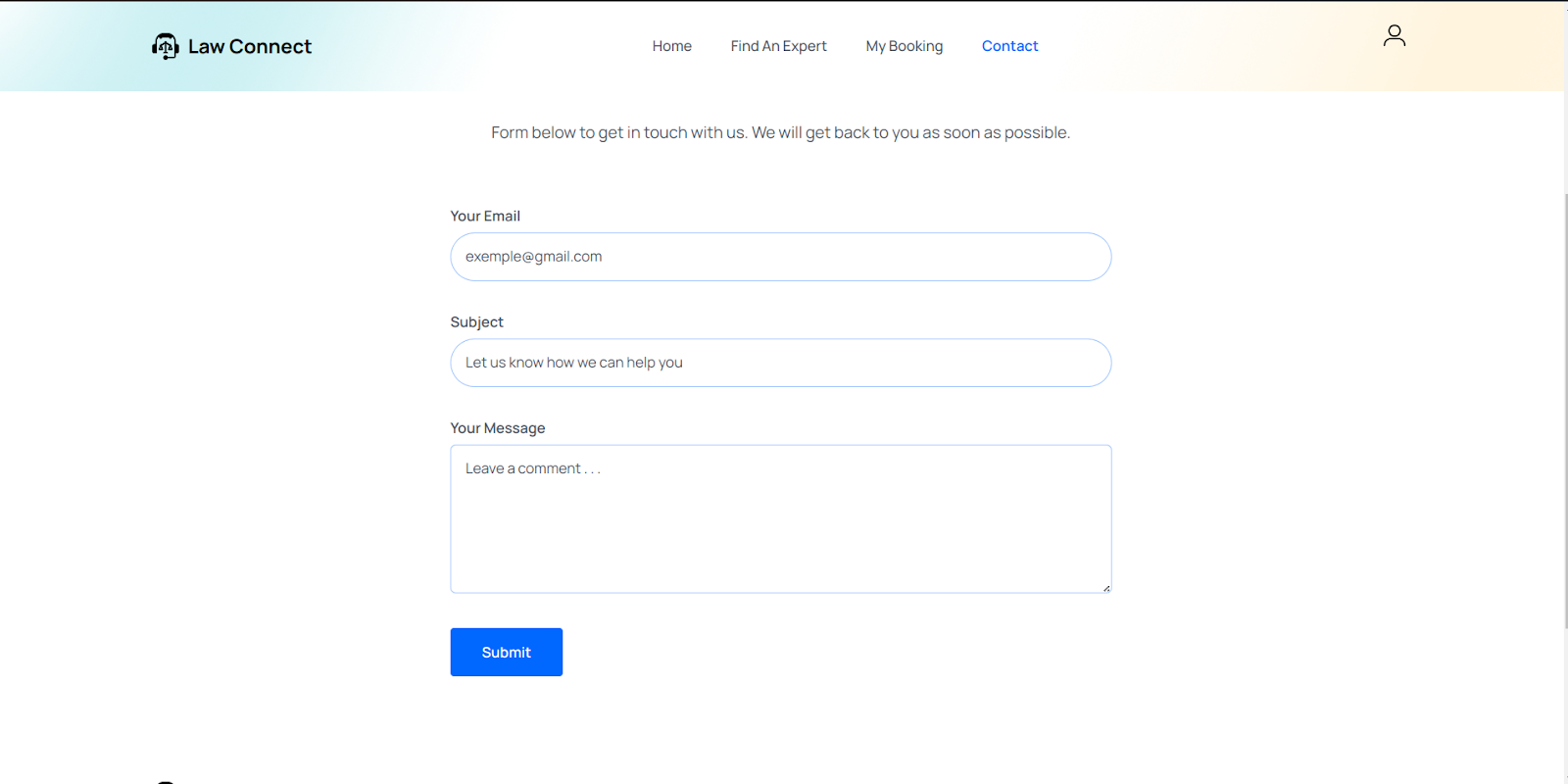
* + 1. ***Xóa cuộc hẹn***

Tại trang xem danh sách các cuộc hẹn, khách hàng có thể hủy các cuộc hẹn của mình bằng cách xóa cuộc hẹn.

## Gửi phản hồi

## Gửi khiếu nại

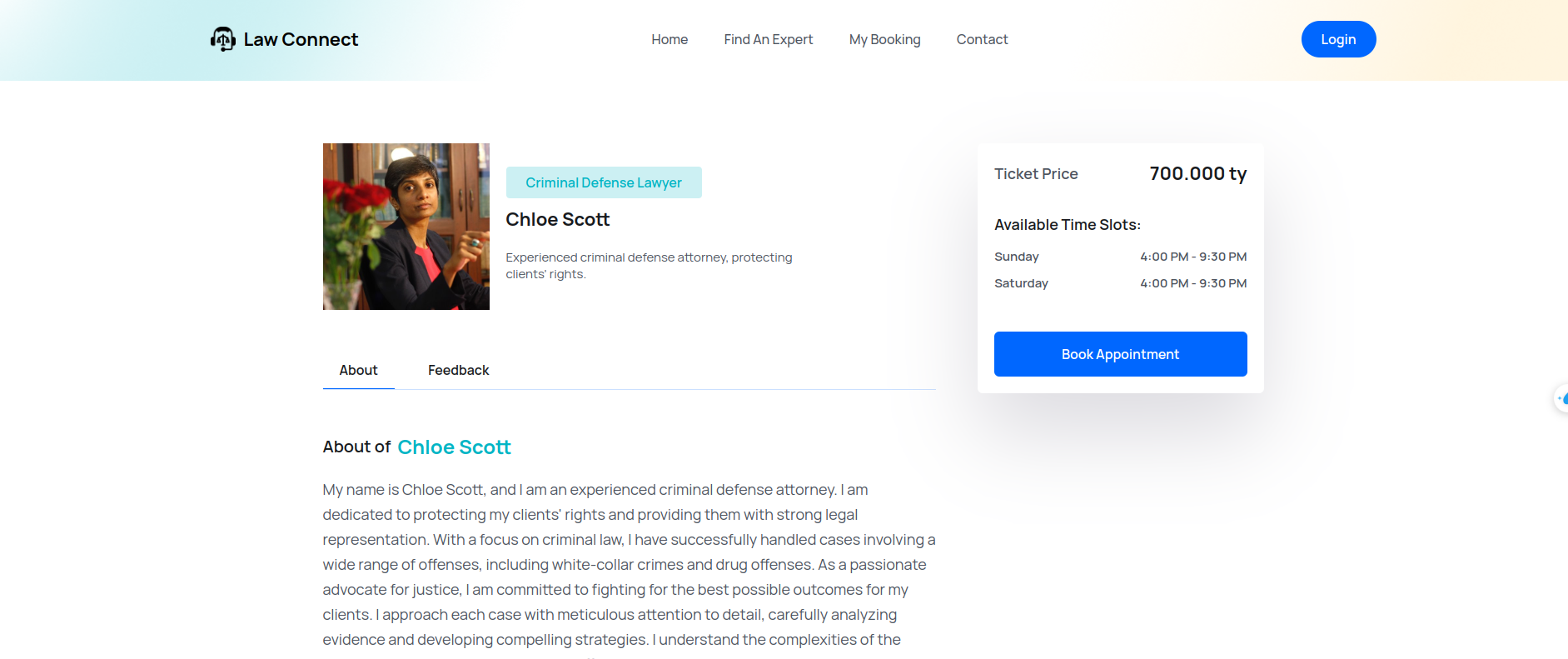
Khách hàng gửi report cho admin và lưu lại thông tin của mình bằng cách điền form html. Report sẽ được lưu vào database.



*Giao diện gửi khiếu nại*

## Feedback

Tại trang xem thông tin chi tiết về luật sư, khách hàng có thể chọn nút “Feedback” để đưa ra bình luận và đánh giá về luật sư. Sau khi bấm chọn vào nút, thực hiện bình luận và gửi.



*Giao diện xem thông tin chi tiết về luật sư*

## Quản lý tài khoản

Sau khi đăng nhập, người dùng có thể vào trang profile để thay đổi thông tin cá nhân của mình, bao gồm cả mật khẩu. Nhập thông tin vào form và bấm update, tự động Server sẽ lưu lại và cập nhật thông tin mới.



*Giao diện thông tin tài khoản cá nhân*

## Admin

Sau khi đăng nhập với tư cách admin, admin có một giao diện khác để quản lý thông tin người dùng và các cuộc hẹn. Thông tin ở các cột đằng trước, những hoạt động của admin sẽ thực hiện tại cột Action.

******

# **Giao diện ngoài**

* Lawyer Connect sử dụng Cloudinary để lưu ảnh phục vụ cho các mục đích: Avatar, ảnh luật sư, …
* Dự kiến sẽ sử dụng một số API ngoài liên quan đến Chat và Payment.