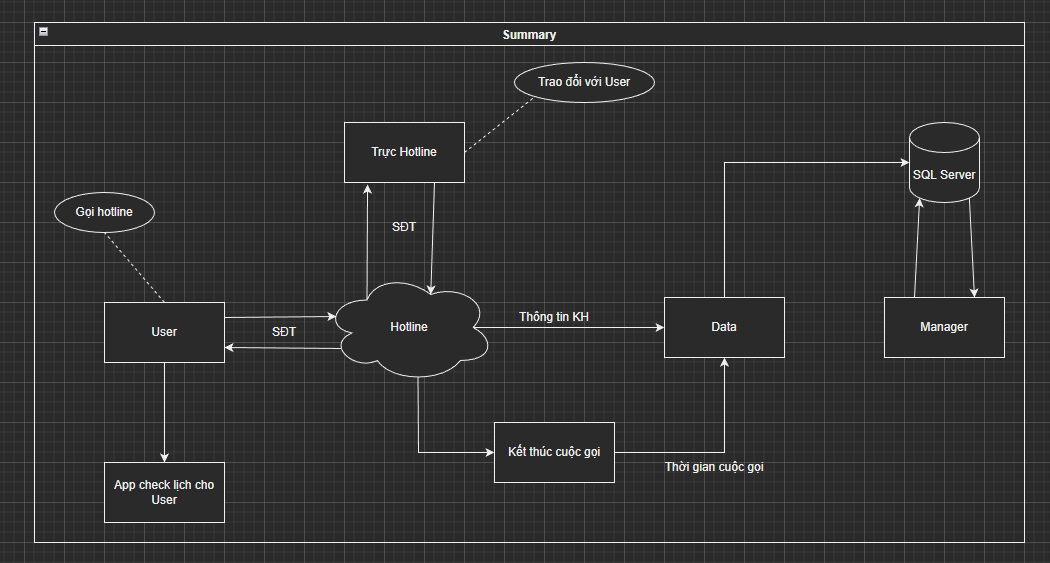
TÀI LIỆU

1. Tổng quan:

- Mô hình tổng quan:



- Mô tả:

+ User gọi điện thoại cho hotline

+ Hotline ghi nhận số điện thoại gọi đến:

. Nếu người dùng tắt máy khi bên hotline chưa trả lời, cuộc gọi sẽ lưu thời gian gọi đến ngay khi người dùng tắt máy và xác định là cuộc gọi nhỡ.

. Nếu hotline nhấc máy và trao đổi với người dùng, thời gian hotline bắt đầu nhấc máy được tính bắt đầu cuộc gọi, và khi 1 trong 2 bên tắt máy thì sẽ xác định thời gian kết thúc cuộc gọi. Trạng thái sẽ được lưu là hoàn thành.

+ Sau khi kết thúc mỗi cuộc gọi, hotline sẽ tự động tạo và lưu lại lịch sử cuộc gọi vào database. Nếu số điện thoại đó đã có người dùng đăng kí, tên của người dùng sẽ được hiển thị, ngược lại sẽ hiển thị “Unknown”.

+ Manager có thể xem, sửa, xoá danh sách người dùng, lịch đặt khám, danh sách nha sĩ, lịch sử cuộc gọi thông qua ứng dụng trên Desktop.

+ User có thể vào ứng dụng trên Android để có thể kiểm tra lịch sử đặt lịch của mình.

1. Chi tiết mô tả:

- User (Sử dụng ứng dụng Android):

+ User truy cập vào ứng dụng và đăng nhập bằng tài khoản và mật khẩu được cấp.

+ User sẽ xem được danh sách đặt lịch khám với các thông tin (mã đặt lịch, tiêu đề, mô tả, ngày khám, trạng thái(xử lí, thành công, huỷ), tên nha sĩ).

+ User sử dụng ứng dụng để xem danh sách nha sĩ, chuyển hướng qua cuộc gọi để gọi cho hotline, xem lịch sử đặt lịch, lịch sử gọi điện.

- Manager (Sử dụng ứng dụng winform):

+ Manager truy cập vào ứng dụng bằng tài khoản và mật khẩu được hệ thống tạo sẵn.

+ Có quyền xem các thông tin như nha sĩ, user, danh sách đặt lịch, danh sách cuộc gọi.

+ Có quyền cập nhật, xoá các thông tin như nha sĩ, user, danh sách đặt lịch.

- Hotline (Ứng dụng được cài vào sđt được cài trên ứng dụng Android):

+ Ứng dụng sẽ tự động bắt số điện thoại gọi tới và lưu lịch sử, trạng thái của cuộc gọi sau mỗi lần kết thúc cuộc gọi.

1. Chi tiết quy trình:

- API (springboot-api):

+ Thông tin file:

. Common: Chứa các hàm được sử dụng chung, trong đó có 1 file Constants.java dùng để chứa các biến được khai báo sẵn để có thể sử dụng bất cứ lúc nào.

. Config: Chứa các file cấu hình cho chương trình (trong đó có file SeedDataRunner sử dụng để khởi tạo dữ liệu).

. Controller: Sử dụng để xử lý các yêu cầu từ người dùng.

. Service: Xử lí nghiệp vụ và xử lý các yêu cầu từ tầng Controller.

. Repository: Giao tiếp với csdl để truy vấn, cung cấp dữ liệu cho Service.

. Domain: Chứa các thực thể đại diện cho các bảng trong db.

. DTO: Class trung gian chứa dữ liệu giao tiếp giữa client và server để tránh gửi quá nhiều thông tin không cần thiết.

. application.properties: Cấu hình kết nối db, cấu hình hibernate.

. pom.xml: Chứa thông tin project, các dependency (thư viện) được sử dụng trong project.

+ Thông tin chức năng:

. Đăng nhập, đăng kí tài khoản cho User.

. Tìm thông tin User bằng sđt.

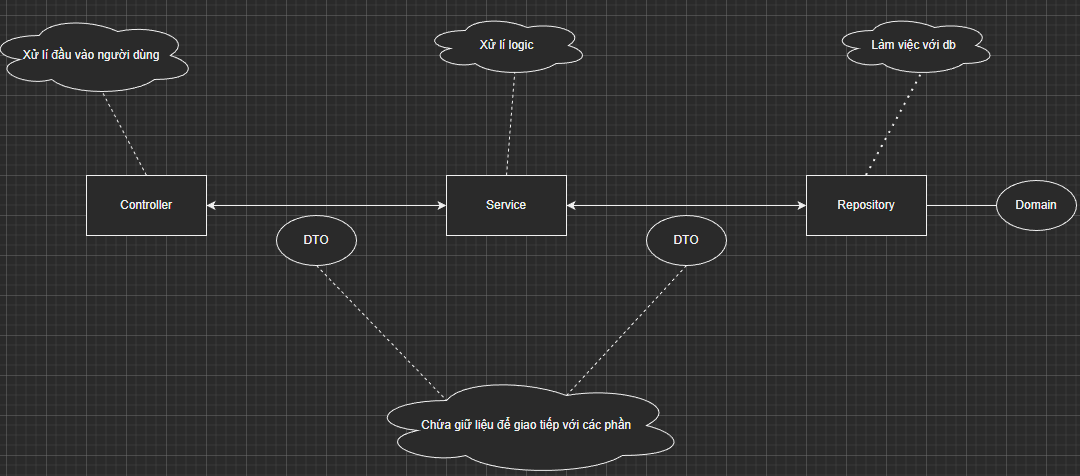
. Xem và tạo thông tin cuộc gọi.

. Xem và cập nhật thông tin nha sĩ.

. Xem, tìm, cập nhật thông tin lịch hẹn.

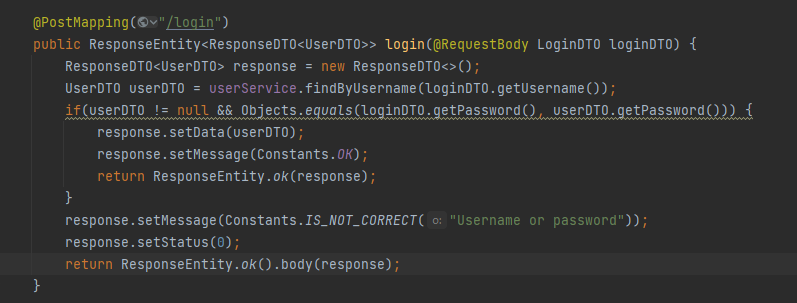
+ Quy trình:

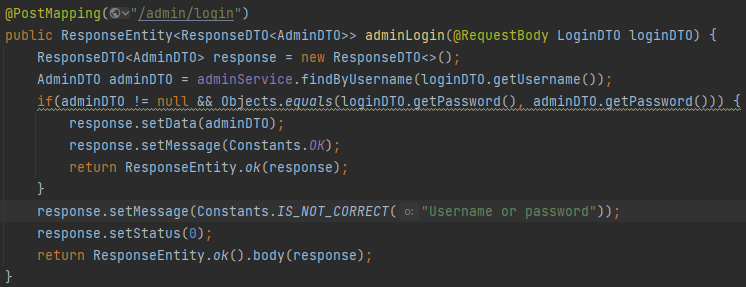
Sơ đồ quy trình chung (springboot):



. Đăng kí, đăng nhập (AuthController):

Đăng nhập User: Kiểm tra trong bảng user xem username đã tồn tại hay chưa, nếu đã có thì sẽ so password đúng với password được nhập vào hay không.

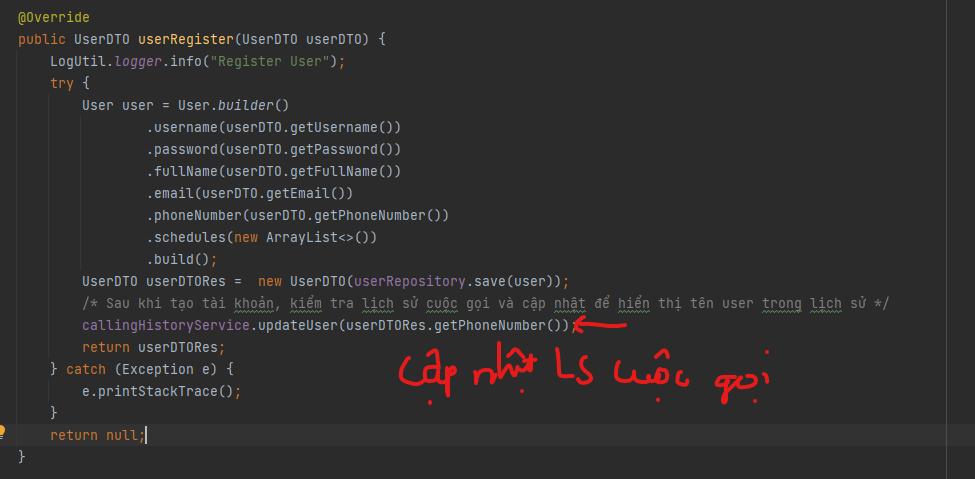
 Đăng nhập Admin: Tương tự nhưng sẽ sử dụng bảng admin



Đăng kí User: Kiểm tra các đầu vào tránh bị null hoặc rỗng. Sau đó sẽ đăng kí User mới. Khi đăng kí thành công, hệ thống sẽ cập nhật danh sách lịch sử cuộc gọi để lưu thông tin theo User mới khi danh sách có số điện thoại trùng với thông tin đăng kí của User mới.



AuthController



AuthService --> AuthServiceImpl

. Lịch sử cuộc gọi (CallingHistoryController):

Xem tất cả lịch sử cuộc gọi (1)

Xem lịch sử cuộc gọi của user (2)

Tạo lịch sử cuộc gọi (3)



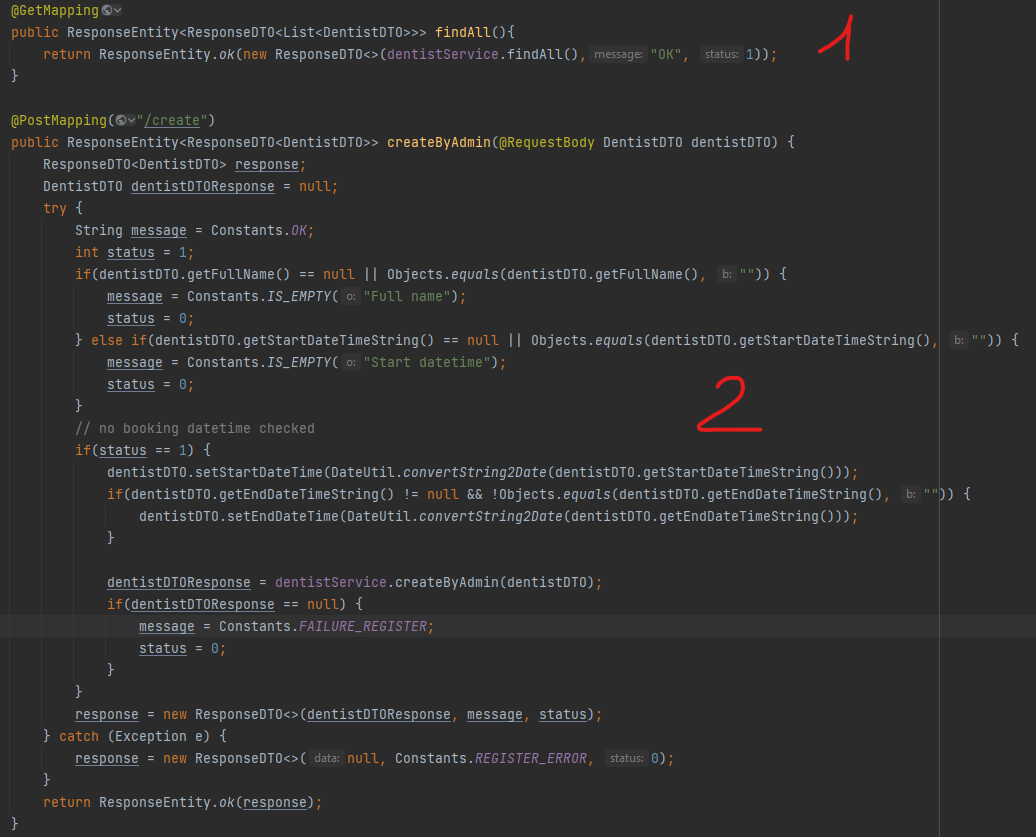
CallingHistoryController

. Nha sĩ (DentistController):

Xem tất cả nha sĩ (1)

Tạo mới nha sĩ (2): Kiểm tra đầu vào fillname và startdatetime không bị rỗng hoặc null. Sau đó thực hiện đăng kí.

Cập nhật thông tin nha sĩ, bao gồm cả xoá (3): Kiểm tra nha sĩ xem có tồn tại hay không. Sau đó mới thực hiện xoá. (Lưu ý: xoá chỉ là cập nhật trạng thái, không xoá hẳn, tránh lỗi).





DentistController

. Lịch hẹn (ScheduleController):

Xem tất cả (1)

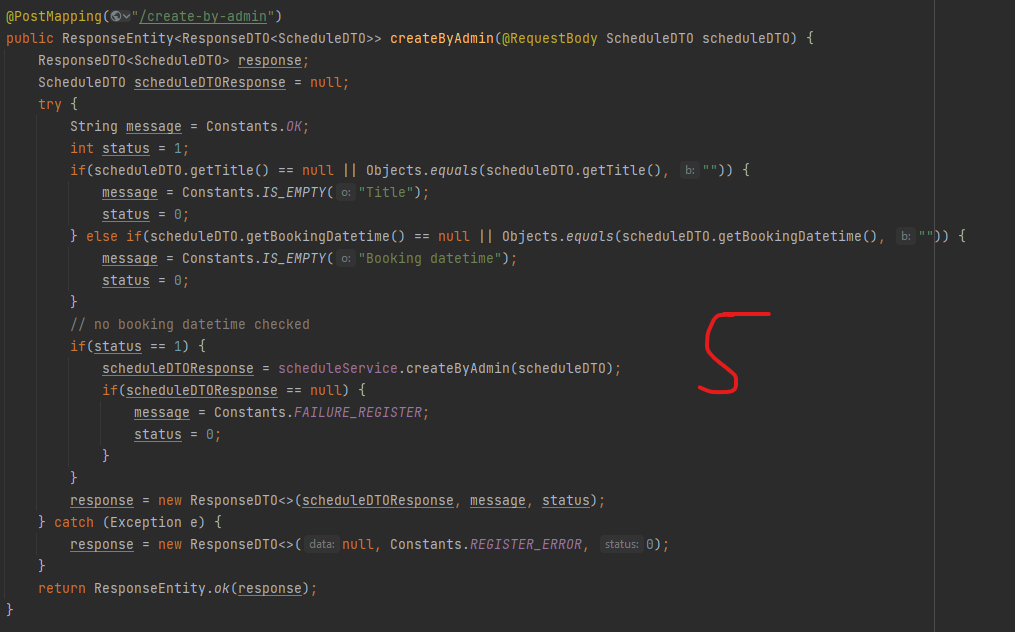
Tìm kiếm lịch theo code (2)

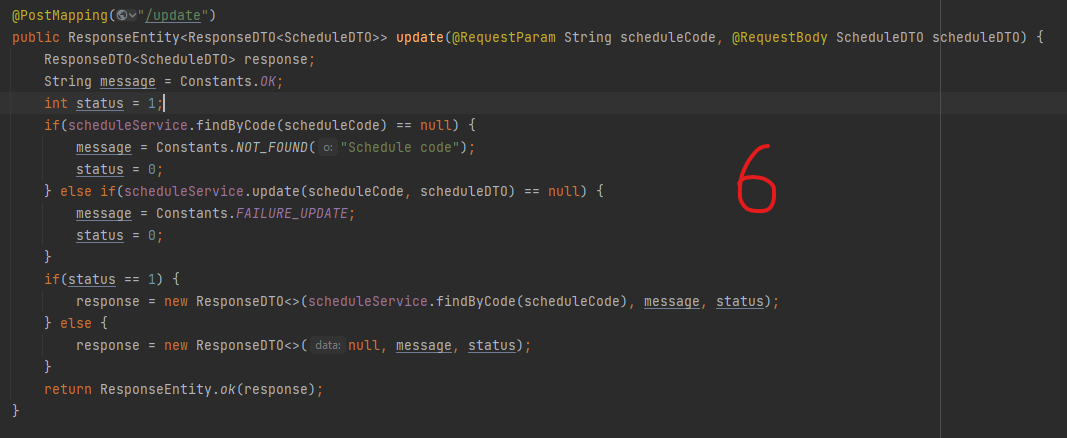
Xem Schedule của User (3) và Dentist (4): Dữ liệu trả theo người dùng (2 chức năng này không cần dùng cho lắm)

Tạo mới từ Admin (5): Kiểm tra đầu vào 2 trường title và booking\_datetime

Cập nhật thông tin của Schedule (6): Cập nhật trạng thái (lưu ý cập nhật chỉ cập nhật trạng thái, khi muốn đổi lịch thì huỷ lịch cũ và tạo lịch mới, không chỉnh sửa thông tin lịch đã tạo).





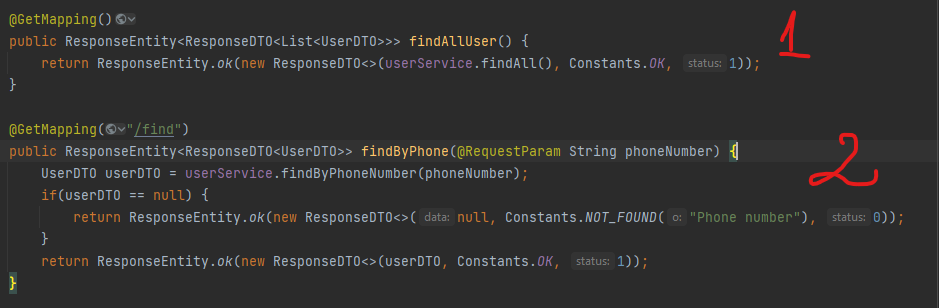


ScheduleController

. User (UserController):

Xem tất cả (1)

Tìm theo số điện thoại (2)

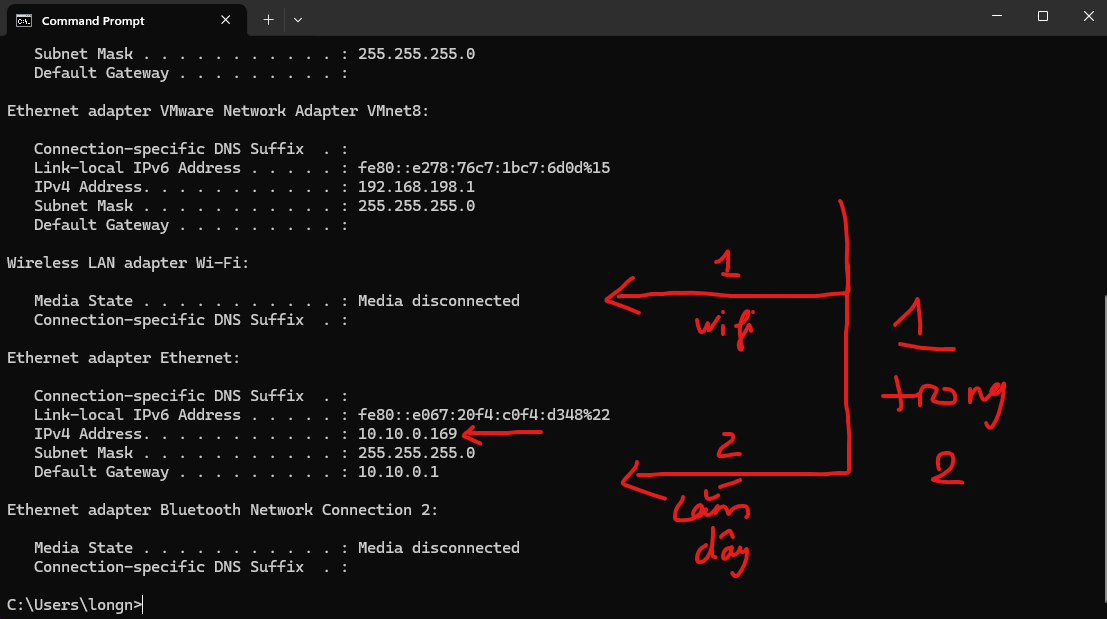


UserController

- Incoming calling process (Hotline cài đặt trên android): (Phần này mình đã note kĩ trong code rồi á. Các bạn mở code ra xem nhé.)

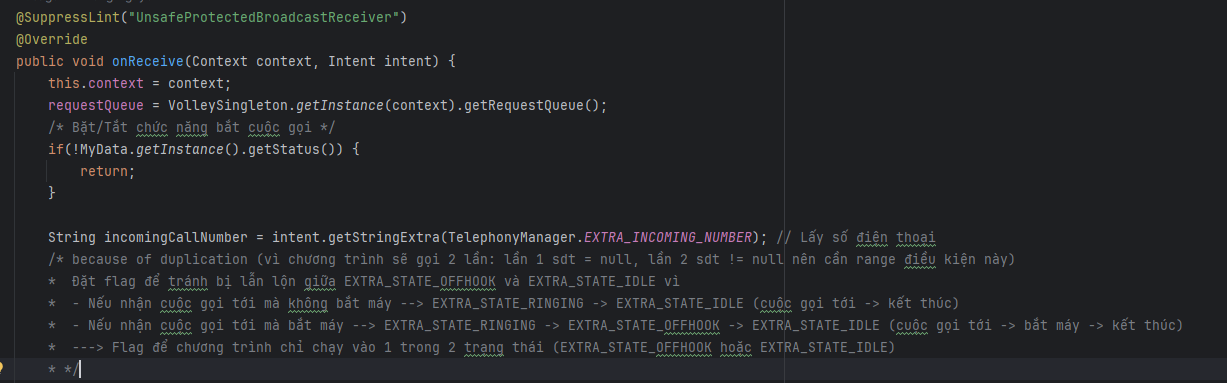
+ Thông tin file:

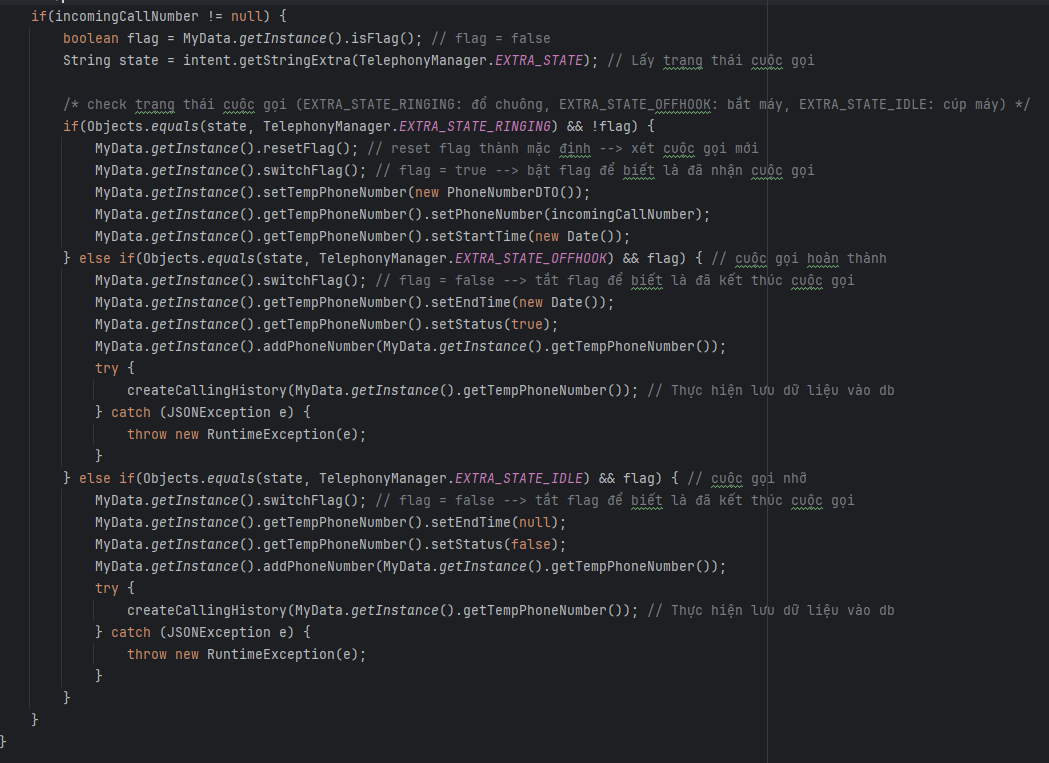
. Constants: Chứa các biến được khai báo sẵn để sử dụng (lưu ý: biến IP là IP của springboot-api, check ip trên máy khi chạy springboot-api).

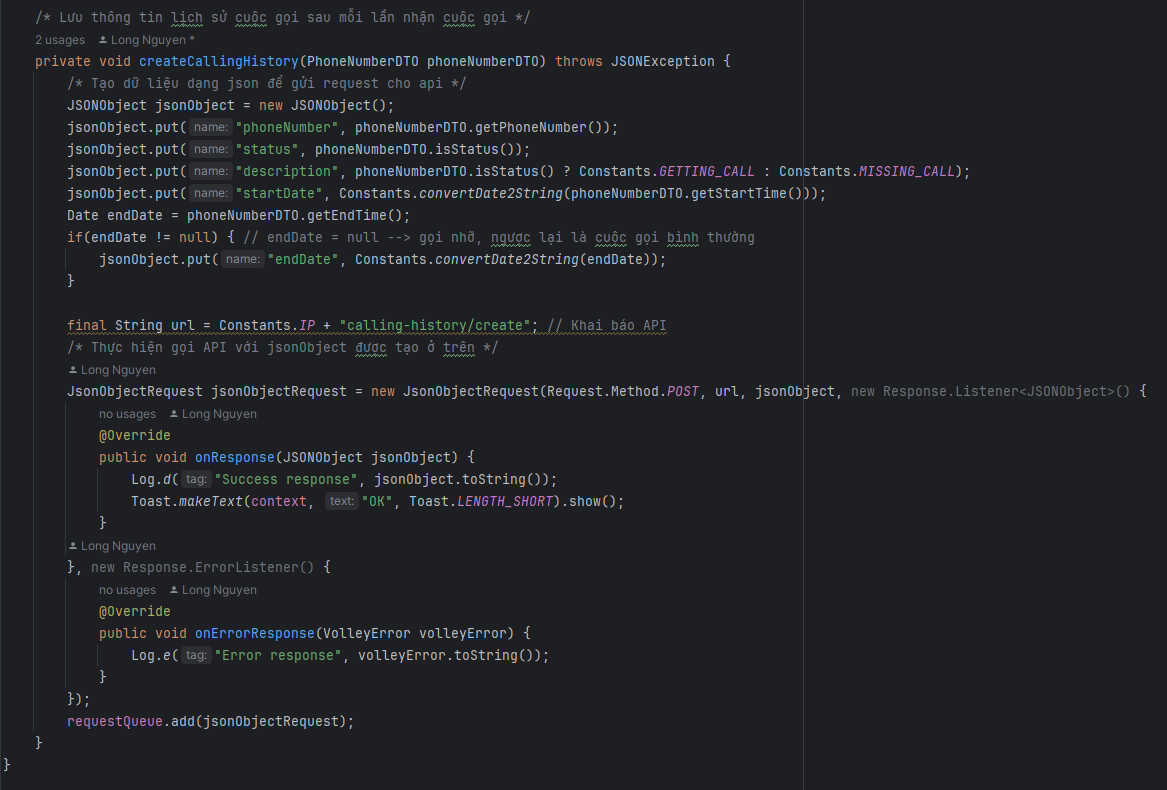


IP sẽ là 10.10.x.x hoặc 192.168.x.x

. IncomingCallingProcess: cấu hình để bắt cuộc gọi tới (đã note kĩ trong code).





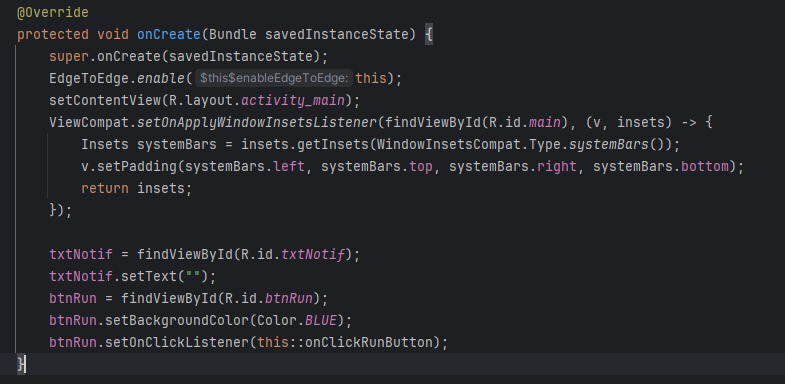


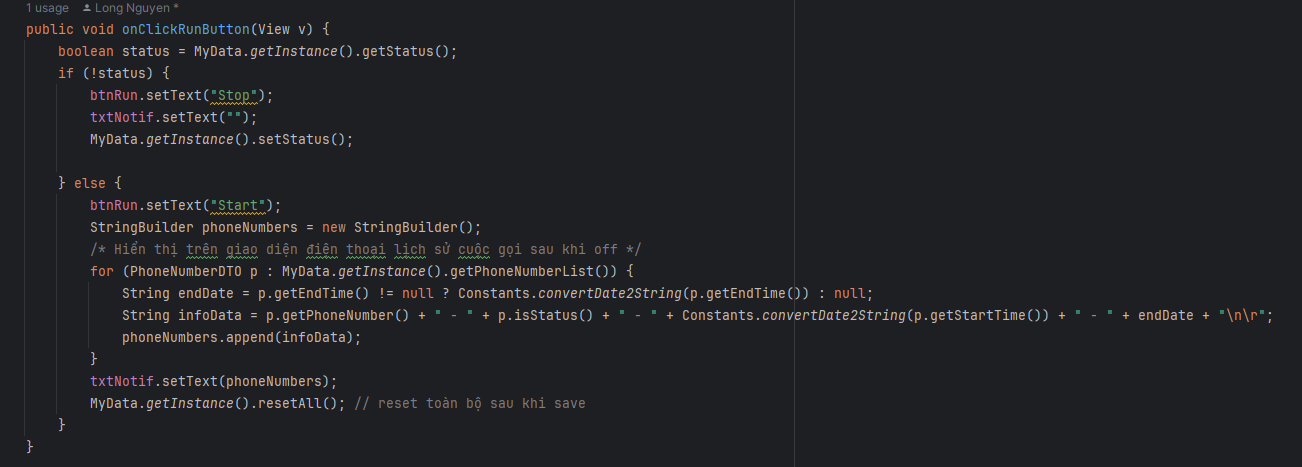
. MyData: File chứa class được tạo ngay từ lúc chạy chương trình dùng để lưu tạm thời thông tin trạng thái để bắt cuộc gọi tới.

. PhoneNumberDTO: Chứa thông tin chi tiết cuộc gọi tới và lưu vào db.

. VolleySingleton: File cấu hình để gọi API.

. MainActivity: File chính chạy chương trình (onCreate: Vẽ giao diện, cấu hình. onClickRunButton: Cấu hình nút bấm trên giao diện.)





- Manager application (Hotline cài đặt trên android):

+ Thông tin file:

. Models: Tương tự như DTO

. NavigationControllers: Hiển thị các navigation bar (Dentist, Customer, Schedule, HistoryCall, Setting).

. Services: Gọi API để trả về data (APIService.cs: cài đặt API, các Service còn lại thực hiện gọi các API liên quan).

. UserControllers + các file bên ngoài: View hiển thị danh sách các thông tin liên quan. Sẽ bao gồm 3 file: .cs, .Designer.cs, .resx



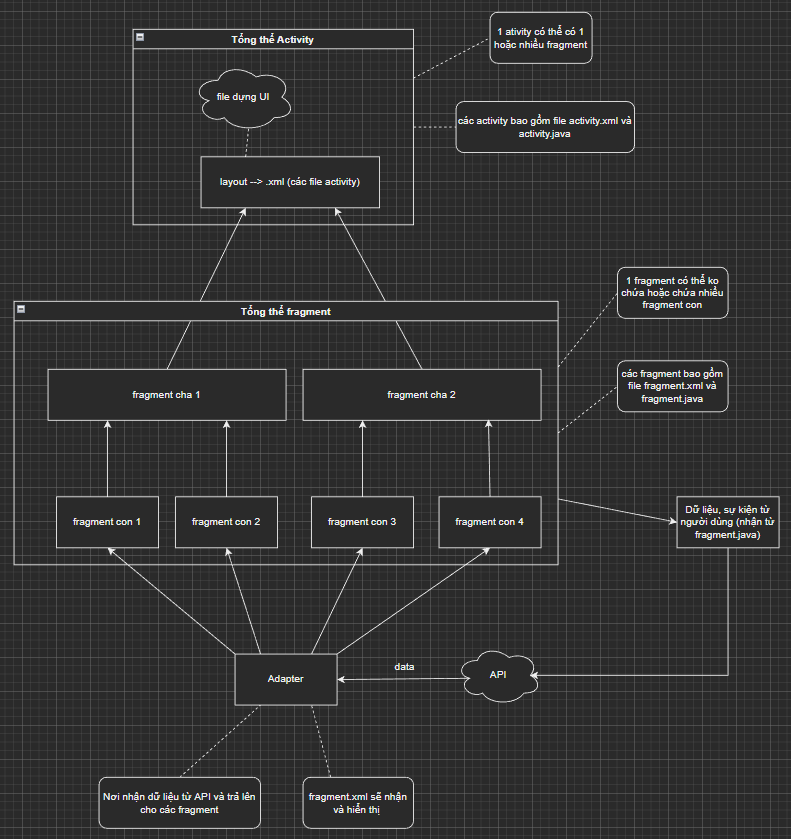
- User application (Hotline cài đặt trên android):

+ Thông tin file:

. Application.java: Khai báo các biến sử dụng lại nhiều lần (lưu ý có biến IP sẽ cấu hình tương tự như cấu hình IP trong Incoming-calling- process).

. ApiService.java: Cấu hình API

. Mô hình chung của Android UI:



. Ví dụ chức năng đăng nhập:

