**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**ÍÎ&ÓÔ**



**BÁO CÁO CHUYÊN ĐỀ TỐT NGHIỆP**

NGÀNH : CÔNG NGHỆ THÔNG TIN

**ĐỀ TÀI : HỆ THỐNG CẢNH BÁO SỚM ĐỘT QUỴ - SEWS**

Giảng viên hướng dẫn : **Hoàng Công Trình**

Sinh viên thực hiện : **Lê Quang Long 23810067**

**Trịnh Thành Luân 23810067**

**Trần Hoàng Quân 238100**

Lớp : **23LC10DN2**

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 12 NĂM 2025**

**LỜI MỞ ĐẦU**

Trong bối cảnh xã hội hiện đại, đột quỵ não đã trở thành một trong những nguyên nhân gây tử vong và tàn phế hàng đầu trên thế giới. Theo thống kê của Tổ chức Y tế Thế giới (WHO), mỗi năm có khoảng 15 triệu người mắc đột quỵ, trong đó 5 triệu người tử vong và 5 triệu người phải sống với di chứng tàn phế suốt đời. Tại Việt Nam, tỷ lệ mắc đột quỵ ngày càng gia tăng và có xu hướng trẻ hóa, đặt ra thách thức lớn cho hệ thống y tế và cộng đồng.

Điều đáng chú ý là phần lớn các ca đột quỵ có thể được phòng ngừa nếu phát hiện sớm các yếu tố nguy cơ và có biện pháp can thiệp kịp thời. Tuy nhiên, việc tiếp cận các dịch vụ y tế chuyên sâu còn nhiều hạn chế, đặc biệt ở các vùng nông thôn và với những người có điều kiện kinh tế khó khăn. Đây chính là động lực để em thực hiện đề tài "Hệ thống cảnh báo sớm đột quỵ - SEWS".

Đồ án này hướng đến việc xây dựng một ứng dụng di động thông minh, ứng dụng công nghệ trí tuệ nhân tạo (AI) để dự đoán nguy cơ đột quỵ dựa trên các chỉ số sức khỏe của người dùng. Bên cạnh đó, hệ thống còn cung cấp các tính năng theo dõi sức khỏe, kết nối với bác sĩ, và hỗ trợ khẩn cấp SOS, nhằm tạo ra một giải pháp toàn diện cho việc phòng ngừa và ứng phó với đột quỵ.

Với quy mô lớn của dự án bao gồm 91 màn hình giao diện, 34 service classes, và tích hợp nhiều công nghệ hiện đại như Flutter, Firebase, Flask API, em đã sử dụng các công cụ AI hỗ trợ trong quá trình phát triển. Điều này không chỉ giúp đảm bảo tiến độ mà còn là cơ hội để em học hỏi và làm quen với xu hướng phát triển phần mềm hiện đại, nơi AI đóng vai trò như một công cụ hỗ trợ đắc lực cho developer.

Em hy vọng đồ án này sẽ đóng góp một phần nhỏ vào việc nâng cao nhận thức cộng đồng về phòng ngừa đột quỵ, đồng thời là nền tảng để phát triển thành một sản phẩm thực tế, mang lại giá trị cho xã hội.

**LỜI CẢM ƠN**

Để hoàn thành đồ án tốt nghiệp này, em đã nhận được rất nhiều sự giúp đỡ, hỗ trợ và động viên từ thầy cô, gia đình và bạn bè.

Trước hết, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Hoàng Công Trình, người đã trực tiếp hướng dẫn em trong suốt quá trình thực hiện đồ án. Thầy đã tận tình chỉ bảo, định hướng và đưa ra những góp ý quý báu giúp em hoàn thiện đề tài. Sự kiên nhẫn và tâm huyết của thầy là nguồn động lực lớn để em vượt qua những khó khăn trong quá trình nghiên cứu và phát triển hệ thống.

Em cũng xin gửi lời cảm ơn đến gia đình, những người luôn ở bên cạnh, động viên và tạo điều kiện tốt nhất để em tập trung học tập và nghiên cứu.

Cuối cùng, em xin cảm ơn các bạn sinh viên cùng khóa đã chia sẻ kinh nghiệm, hỗ trợ kỹ thuật và đóng góp ý kiến trong quá trình phát triển ứng dụng.

Mặc dù đã cố gắng hết sức, nhưng do kiến thức và kinh nghiệm còn hạn chế, đồ án không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý của quý thầy cô và các bạn để đồ án được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, tháng 12 năm 2025

Sinh viên thực hiện

Lê Quang Long ( Đại diện nhóm )

MỤC LỤC

[PHẦN 1: TỔNG QUAN DỰ ÁN 6](#_Toc216365371)

[1.1. Giới thiệu đề tài 6](#_Toc216365372)

[1.2. Mục tiêu dự án 6](#_Toc216365373)

[1.3. Phạm vi dự án 7](#_Toc216365374)

[1.4. Đối tượng sử dụng 9](#_Toc216365375)

[1.5. Kết luận 10](#_Toc216365376)

[PHẦN 2: PHÂN TÍCH YÊU CẦU 11](#_Toc216365377)

[2.1. Yêu cầu chức năng 11](#_Toc216365378)

[2.2. Yêu cầu phi chức năng 17](#_Toc216365379)

[PHẦN 3: THIẾT KẾ HỆ THỐNG 18](#_Toc216365380)

[3.1. Kiến trúc hệ thống 18](#_Toc216365381)

[3.2. Thiết kế cơ sở dữ liệu 19](#_Toc216365382)

[3.3. Thiết kế giao diện (UI/UX) 20](#_Toc216365383)

[3.4. Biểu đồ lớp (Class Diagram) 31](#_Toc216365385)

[3.5. Biểu đồ tuần tự (Sequence Diagram) 32](#_Toc216365387)

[PHẦN 4: CÔNG NGHỆ SỬ DỤNG 34](#_Toc216365388)

[4.1. Nền tảng phát triển ứng dụng di động (Frontend) 34](#_Toc216365389)

[4.2. Hạ tầng Backend và Lưu trữ dữ liệu 34](#_Toc216365390)

[4.3. Mô hình Trí tuệ nhân tạo (AI) và API Service 35](#_Toc216365391)

[4.4. Các thư viện hỗ trợ và Công cụ phát triển 35](#_Toc216365392)

[4.5. Tổng kết công nghệ 35](#_Toc216365393)

[PHẦN 5: CÀI ĐẶT VÀ TRIỂN KHAI 37](#_Toc216365394)

[5.1. Yêu cầu hệ thống 37](#_Toc216365395)

[5.2. Thiết lập môi trường phát triển 37](#_Toc216365396)

[5.3. Cấu hình Hạ tầng Firebase 37](#_Toc216365397)

[5.4. Quy trình Biên dịch và Triển khai 38](#_Toc216365398)

[5.5. Tổng kết cài đặt 38](#_Toc216365399)

[PHẦN 6: HƯỚNG DẪN SỬ DỤNG 39](#_Toc216365400)

[6.1. Phân hệ dành cho Người dùng cuối (User) 39](#_Toc216365401)

[6.2. Phân hệ dành cho Bác sĩ (Doctor) 40](#_Toc216365402)

[6.3. Phân hệ Quản trị viên (Admin) 40](#_Toc216365403)

[PHẦN 7: KẾT LUẬN 41](#_Toc216365404)

[7.1. Các kết quả đạt được 41](#_Toc216365405)

[7.2. Những hạn chế và Tồn tại 41](#_Toc216365406)

[7.3. Hướng phát triển 42](#_Toc216365407)

[7.4. Kết luận chung 43](#_Toc216365408)

[PHẦN PHỤ LỤC 44](#_Toc216365409)

[Phụ lục A: Đặc tả kỹ thuật API và Cấu trúc dữ liệu 44](#_Toc216365410)

[Phụ lục B: Tổ chức mã nguồn dự án 44](#_Toc216365411)

[Phụ lục C: Tài liệu tham khảo 45](#_Toc216365412)

[Phụ lục D: Báo cáo việc ứng dụng Trí tuệ nhân tạo (AI) trong dự án 45](#_Toc216365413)

# PHẦN 1: TỔNG QUAN DỰ ÁN

## Giới thiệu đề tài

*1.1.1. Tên đề tài*

SEWS - Stroke Early Warning System (Hệ thống Cảnh báo Sớm Đột quỵ)

*1.1.2. Bối cảnh và lý do chọn đề tài*

Đột quỵ não là một trong những nguyên nhân gây tử vong và tàn phế hàng đầu trên thế giới. Theo thống kê của Tổ chức Y tế Thế giới (WHO):

Mỗi năm có khoảng 15 triệu người trên thế giới bị đột quỵ. Trong đó 5 triệu người tử vong và 5 triệu người bị tàn phế vĩnh viễn. Tại Việt Nam, tỷ lệ đột quỵ đang gia tăng nhanh chóng, đặc biệt ở nhóm người trẻ tuổi.

Thời gian vàng trong điều trị đột quỵ là 3-4.5 giờ đầu tiên. Việc phát hiện sớm các dấu hiệu và yếu tố nguy cơ đột quỵ có thể giảm thiểu tỷ lệ tử vong, hạn chế di chứng sau đột quỵ, và nâng cao chất lượng cuộc sống cho bệnh nhân.

Với sự phát triển của công nghệ AI/Machine Learning và thiết bị di động, việc xây dựng một hệ thống cảnh báo sớm đột quỵ là hoàn toàn khả thi và cần thiết.

*1.1.3. Mô tả tổng quan*

SEWS (Stroke Early Warning System) là ứng dụng di động được phát triển trên nền tảng Flutter, tích hợp trí tuệ nhân tạo để:

* Dự đoán nguy cơ đột quỵ sử dụng mô hình Random Forest với preprocessor.
* Dự đoán nguy cơ tiểu đường type 2 dựa trên các yếu tố nguy cơ.
* Theo dõi sức khỏe liên tục với biểu đồ trực quan (huyết áp, đường huyết, BMI).
* Kết nối người dùng với bác sĩ để tư vấn và đặt lịch khám.
* Hỗ trợ khẩn cấp với tính năng SOS gửi vị trí GPS.

## Mục tiêu dự án

*1.2.1. Mục tiêu tổng quát*

Xây dựng một hệ thống phần mềm hoàn chỉnh hỗ trợ cảnh báo sớm nguy cơ đột quỵ, kết nối người dùng với đội ngũ y tế, và quản lý sức khỏe toàn diện.

*1.2.2. Mục tiêu cụ thể*

(1) Dự đoán nguy cơ đột quỵ: Sử dụng mô hình AI (Random Forest) để phân tích 14 features và dự đoán nguy cơ đột quỵ.

(2) Dự đoán nguy cơ tiểu đường: Thuật toán đánh giá nguy cơ tiểu đường type 2 dựa trên 6 yếu tố nguy cơ.

(3) Theo dõi sức khỏe: Ghi nhận và hiển thị các chỉ số sức khỏe (huyết áp, đường huyết, BMI) với biểu đồ fl\_chart.

(4) Kết nối y tế: Cho phép người dùng đặt lịch hẹn, chat trực tiếp với bác sĩ.

(5) Hỗ trợ khẩn cấp: Cung cấp tính năng SOS gửi yêu cầu cấp cứu kèm vị trí GPS.

(7) Nhắc nhở thuốc: Hệ thống nhắc nhở uống thuốc đúng giờ với local notifications.

(8) Cung cấp kiến thức: Chia sẻ bài viết, kiến thức về sức khỏe và phòng ngừa đột quỵ.

(9) Xây dựng cộng đồng: Diễn đàn để người dùng chia sẻ kinh nghiệm.

(10) Hỗ trợ offline: Cache dữ liệu với Hive để sử dụng khi không có mạng.

## 1.3. Phạm vi dự án

*1.3.1. Phạm vi chức năng*

Hệ thống bao gồm 3 module chính phục vụ 3 đối tượng người dùng:

**A. Module Người dùng (User) - 54 screens, 23 thư mục chức năng**

* Xác thực: Đăng ký, đăng nhập (Email, Google), quên mật khẩu, onboarding.
* Dự đoán AI: Dự đoán nguy cơ đột quỵ, tiểu đường với form nhập liệu và hiển thị kết quả.
* Theo dõi sức khỏe: Biểu đồ huyết áp, đường huyết, BMI, lịch sử sức khỏe.
* Bác sĩ: Danh sách bác sĩ, chi tiết bác sĩ, đánh giá bác sĩ.
* Lịch hẹn: Đặt lịch, xem lịch hẹn, báo cáo lịch hẹn.
* Chat: Danh sách chat, chi tiết chat, chọn bác sĩ chat.
* SOS khẩn cấp: Gửi SOS, theo dõi trạng thái SOS.
* Gia đình: Quản lý gia đình, nhóm gia đình, mời thành viên.
* Nhắc nhở: Danh sách nhắc nhở, thêm/sửa nhắc nhở.
* Kiến thức: Danh sách bài viết, chi tiết bài viết.
* Cộng đồng: Diễn đàn, chi tiết chủ đề.
* Cài đặt: Cài đặt chung, đổi mật khẩu, thông báo.
* Pháp lý: Điều khoản dịch vụ, chính sách bảo mật.

**B. Module Bác sĩ (Doctor) - 17 screens, 11 thư mục chức năng**

* Dashboard: Thống kê ca trực, tổng quan hoạt động.
* Bệnh nhân: Danh sách bệnh nhân, hồ sơ bệnh nhân.
* Lịch hẹn: Quản lý lịch hẹn, chi tiết yêu cầu lịch hẹn.
* SOS: Hàng đợi SOS, chi tiết ca SOS.
* Chat: Danh sách chat, chi tiết chat, chọn bệnh nhân chat.
* Đánh giá: Xem đánh giá từ bệnh nhân.
* Lịch làm việc: Quản lý slot khám.
* Cài đặt: Cài đặt bác sĩ.
* Thông báo: Thông báo cho bác sĩ.

**C. Module Quản trị viên (Admin) - 20 screens, 14 thư mục chức năng**

* Dashboard: Thống kê tổng quan hệ thống (9 mục quản lý).
* Quản lý Users: CRUD người dùng.
* Quản lý Bác sĩ: CRUD bác sĩ.
* Quản lý Bệnh nhân: CRUD bệnh nhân.
* Quản lý SOS: Giám sát ca khẩn cấp.
* Quản lý Dự đoán: Xem lịch sử dự đoán.
* Quản lý Lịch hẹn: Giám sát lịch hẹn.
* Quản lý Kiến thức: CRUD bài viết.
* Quản lý Cộng đồng: Quản lý diễn đàn.

*1.3.2. Phạm vi kỹ thuật*

**A. Cấu trúc dự án**

* Thư mục lib/features/: 91 files - Màn hình UI (User: 54, Doctor: 17, Admin: 20)
* Thư mục lib/services/: 34 files - Business logic và API services
* Thư mục lib/data/models/: 14 files - Data models
* Thư mục lib/widgets/: 7 files - Shared widgets
* Thư mục lib/utils/: 4 files - Utilities
* Thư mục test/services/: 13 files - Unit tests
* Thư mục assets/models/: 10 files - AI model và Flask API

**B. Công nghệ sử dụng**

* Framework: Flutter 3.5.4+
* Ngôn ngữ: Dart 3.5.4+
* State Management: Provider 6.1.1
* Authentication: Firebase Auth 6.1.2
* Database: Firebase Realtime Database 12.0.4
* Storage: Firebase Storage 13.0.4
* Push Notification: Firebase Messaging 16.0.4
* Local Notification: flutter\_local\_notifications 17.0.0
* AI/ML API: Flask + scikit-learn + joblib (Python 3.8+)
* AI Model: Random Forest Classifier (moHinhDotQuy\_final.pkl)
* Offline Cache: Hive 2.2.3
* Charts: fl\_chart 0.69.0
* Location: Geolocator 10.1.0 + Geocoding 2.1.1
* Image: image\_picker 1.0.7, cached\_network\_image 3.3.1
* Network: connectivity\_plus 6.0.5, http 1.1.0

*1.3.3. Phạm vi triển khai*

* Nền tảng: Android (API 23+), Web
* Firebase Project: warning-software-stroke
* Ngôn ngữ giao diện: Tiếng Việt
* Phiên bản: 1.0.0+1
* File APK: release/SEWS\_v1.0.0.apk (67.3 MB)

## 1.4. Đối tượng sử dụng

*1.4.1. Người dùng (User)*

**Đặc điểm:**

* Người có nguy cơ đột quỵ hoặc muốn theo dõi sức khỏe.
* Người thân của bệnh nhân đột quỵ.
* Người quan tâm đến sức khỏe tim mạch.
* Độ tuổi: Từ 18 tuổi trở lên.

**Nhu cầu:**

* Đánh giá nguy cơ đột quỵ, tiểu đường cá nhân.
* Theo dõi các chỉ số sức khỏe theo thời gian.
* Kết nối với bác sĩ khi cần tư vấn.
* Nhận hỗ trợ khẩn cấp khi có dấu hiệu bất thường.

*1.4.2. Bác sĩ (Doctor)*

**Đặc điểm:**

* Bác sĩ chuyên khoa tim mạch, thần kinh.
* Bác sĩ đa khoa.
* Nhân viên y tế có chứng chỉ hành nghề.

**Nhu cầu:**

* Quản lý danh sách bệnh nhân hiệu quả.
* Theo dõi tình trạng sức khỏe bệnh nhân từ xa.
* Xử lý ca khẩn cấp SOS nhanh chóng.
* Tư vấn bệnh nhân trực tuyến qua chat.
* Quản lý lịch làm việc và slot khám.

*1.4.3. Quản trị viên (Admin)*

**Đặc điểm:**

* Quản lý hệ thống y tế.
* Nhân viên IT của cơ sở y tế.

**Nhu cầu:**

* Giám sát toàn bộ hoạt động hệ thống.
* Quản lý người dùng, bác sĩ, bệnh nhân.
* Quản lý nội dung (bài viết kiến thức, diễn đàn).
* Theo dõi và xử lý các ca SOS.
* Xem thống kê và báo cáo.

## 1.5. Kết luận

*Thống kê dự án:*

* Tổng số screens: 91 màn hình
* Tổng số services: 34 services
* Tổng số models: 14 models
* Tổng số test files: 16 files (13 service tests + 3 backend tests)
* Dependencies: 25+ packages

*Tóm tắt:*

* Tên dự án: SEWS - Stroke Early Warning System
* Loại ứng dụng: Cross-platform Mobile App
* Framework: Flutter 3.5.4+
* Nền tảng: Android, Web
* Số module: 3 (User, Doctor, Admin)
* Backend: Firebase (Auth, Realtime DB, Storage, Messaging)
* AI/ML: Flask API + Random Forest Model
* Phiên bản: 1.0.0+1

# PHẦN 2: PHÂN TÍCH YÊU CẦU

## 2.1. Yêu cầu chức năng

*2.1.1 Chức năng người dùng ( User )*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm chức năng** | **Tên chức năng (Use Case)** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **A. Xác thực** | **UC1:** Đăng ký tài khoản | Họ tên, email, mật khẩu, xác nhận mật khẩu | Tài khoản được tạo, chuyển đến đăng nhập |
|  | **UC2:** Đăng nhập Email | Email, mật khẩu | Đăng nhập thành công, vào Dashboard |
|  | **UC3:** Đăng nhập Google | Tài khoản Google | Đăng nhập thành công, tự tạo tài khoản nếu chưa có |
|  | **UC4:** Quên mật khẩu | Email đã đăng ký | Email chứa link đặt lại mật khẩu |
|  | **UC5:** Đổi mật khẩu | Mật khẩu cũ, mới, xác nhận mới | Mật khẩu được cập nhật |
| **B. Dự đoán AI** | **UC6:** Dự đoán nguy cơ đột quỵ | Tuổi, chỉ số sức khỏe, tiền sử bệnh, lối sống | Điểm nguy cơ, phân loại (BMI, huyết áp...), mức độ rủi ro |
|  | **UC7:** Dự đoán nguy cơ tiểu đường | Tuổi, chỉ số đường huyết, gia đình, hoạt động | Điểm nguy cơ, BMI, khuyến nghị |
|  | **UC8:** Xem kết quả dự đoán | Kết quả từ UC6 hoặc UC7 | Hiển thị điểm số, chỉ số chi tiết, khuyến nghị |
|  | **UC9:** Xem lịch sử dự đoán | User ID | Danh sách dự đoán theo thời gian |
| **C. Theo dõi sức khỏe** | **UC10:** Nhập chỉ số sức khỏe | Huyết áp, đường huyết, cân nặng, ghi chú | Dữ liệu lưu vào Firebase |
|  | **UC11:** Xem biểu đồ huyết áp | User ID, khoảng thời gian | Biểu đồ đường (fl\_chart) |
|  | **UC12:** Xem biểu đồ đường huyết | User ID, khoảng thời gian | Biểu đồ đường kèm ngưỡng cảnh báo |
|  | **UC13:** Xem biểu đồ BMI | User ID, khoảng thời gian | Biểu đồ BMI phân loại |
|  | **UC14:** Xem lịch sử sức khỏe | User ID | Danh sách bản ghi theo ngày |
| **D. Kết nối bác sĩ** | **UC15:** Xem danh sách bác sĩ | Bộ lọc (chuyên khoa, đánh giá) | Danh sách bác sĩ cơ bản |
|  | **UC16:** Xem chi tiết bác sĩ | Doctor ID | Thông tin, chuyên khoa, lịch làm việc |
|  | **UC17:** Đặt lịch hẹn | Doctor ID, thời gian, lý do | Yêu cầu được tạo, chờ duyệt |
|  | **UC18:** Chat với bác sĩ | Tin nhắn, hình ảnh | Tin nhắn gửi realtime (Firebase) |
|  | **UC19:** Đánh giá bác sĩ | Số sao, nhận xét | Đánh giá được lưu |
| **E. SOS khẩn cấp** | **UC20:** Gửi yêu cầu SOS | Vị trí GPS, mô tả triệu chứng | Yêu cầu tạo, báo bác sĩ trực |
|  | **UC21:** Theo dõi trạng thái SOS | SOS ID | Trạng thái xử lý, thông tin bác sĩ |
| **F. Quản lý gia đình** | **UC22:** Tạo nhóm gia đình | Tên nhóm | Nhóm được tạo (User là admin) |
|  | **UC23:** Mời thành viên | Email hoặc mã mời | Lời mời được gửi |
|  | **UC24:** Theo dõi sức khỏe người thân | Member ID | Thông tin sức khỏe thành viên |
| **G. Nhắc nhở thuốc** | **UC25:** Tạo nhắc nhở | Tên thuốc, liều, thời gian, tần suất | Local notification được lên lịch |
|  | **UC26:** Sửa nhắc nhở | Reminder ID, thông tin mới | Nhắc nhở cập nhật |
|  | **UC27:** Xóa nhắc nhở | Reminder ID | Nhắc nhở bị xóa |
| **H. Kiến thức & Cộng đồng** | **UC28:** Đọc bài viết | Article ID/Danh mục | Nội dung bài viết |
|  | **UC29:** Tham gia diễn đàn | Thread ID | Nội dung thảo luận |
|  | **UC30:** Đăng bài viết | Tiêu đề, nội dung, hình ảnh | Bài viết được đăng |

*2.1.2 Chức năng bác sĩ ( Doctor )*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm chức năng** | **Tên chức năng (Use Case)** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **A. Xác thực** | **UC1:** Đăng ký tài khoản | Họ tên, email, mật khẩu, xác nhận mật khẩu | Tài khoản được tạo, chuyển đến đăng nhập |
|  | **UC2:** Đăng nhập Email | Email, mật khẩu | Đăng nhập thành công, vào Dashboard |
|  | **UC3:** Đăng nhập Google | Tài khoản Google | Đăng nhập thành công, tự tạo tài khoản nếu chưa có |
|  | **UC4:** Quên mật khẩu | Email đã đăng ký | Email chứa link đặt lại mật khẩu |
|  | **UC5:** Đổi mật khẩu | Mật khẩu cũ, mới, xác nhận mới | Mật khẩu được cập nhật |
| **B. Dự đoán AI** | **UC6:** Dự đoán nguy cơ đột quỵ | Tuổi, chỉ số sức khỏe, tiền sử bệnh, lối sống | Điểm nguy cơ, phân loại (BMI, huyết áp...), mức độ rủi ro |
|  | **UC7:** Dự đoán nguy cơ tiểu đường | Tuổi, chỉ số đường huyết, gia đình, hoạt động | Điểm nguy cơ, BMI, khuyến nghị |
|  | **UC8:** Xem kết quả dự đoán | Kết quả từ UC6 hoặc UC7 | Hiển thị điểm số, chỉ số chi tiết, khuyến nghị |
|  | **UC9:** Xem lịch sử dự đoán | User ID | Danh sách dự đoán theo thời gian |
| **C. Theo dõi sức khỏe** | **UC10:** Nhập chỉ số sức khỏe | Huyết áp, đường huyết, cân nặng, ghi chú | Dữ liệu lưu vào Firebase |
|  | **UC11:** Xem biểu đồ huyết áp | User ID, khoảng thời gian | Biểu đồ đường (fl\_chart) |
|  | **UC12:** Xem biểu đồ đường huyết | User ID, khoảng thời gian | Biểu đồ đường kèm ngưỡng cảnh báo |
|  | **UC13:** Xem biểu đồ BMI | User ID, khoảng thời gian | Biểu đồ BMI phân loại |
|  | **UC14:** Xem lịch sử sức khỏe | User ID | Danh sách bản ghi theo ngày |
| **D. Kết nối bác sĩ** | **UC15:** Xem danh sách bác sĩ | Bộ lọc (chuyên khoa, đánh giá) | Danh sách bác sĩ cơ bản |
|  | **UC16:** Xem chi tiết bác sĩ | Doctor ID | Thông tin, chuyên khoa, lịch làm việc |
|  | **UC17:** Đặt lịch hẹn | Doctor ID, thời gian, lý do | Yêu cầu được tạo, chờ duyệt |
|  | **UC18:** Chat với bác sĩ | Tin nhắn, hình ảnh | Tin nhắn gửi realtime (Firebase) |
|  | **UC19:** Đánh giá bác sĩ | Số sao, nhận xét | Đánh giá được lưu |
| **E. SOS khẩn cấp** | **UC20:** Gửi yêu cầu SOS | Vị trí GPS, mô tả triệu chứng | Yêu cầu tạo, báo bác sĩ trực |
|  | **UC21:** Theo dõi trạng thái SOS | SOS ID | Trạng thái xử lý, thông tin bác sĩ |
| **F. Quản lý gia đình** | **UC22:** Tạo nhóm gia đình | Tên nhóm | Nhóm được tạo (User là admin) |
|  | **UC23:** Mời thành viên | Email hoặc mã mời | Lời mời được gửi |
|  | **UC24:** Theo dõi sức khỏe người thân | Member ID | Thông tin sức khỏe thành viên |
| **G. Nhắc nhở thuốc** | **UC25:** Tạo nhắc nhở | Tên thuốc, liều, thời gian, tần suất | Local notification được lên lịch |
|  | **UC26:** Sửa nhắc nhở | Reminder ID, thông tin mới | Nhắc nhở cập nhật |
|  | **UC27:** Xóa nhắc nhở | Reminder ID | Nhắc nhở bị xóa |
| **H. Kiến thức & Cộng đồng** | **UC28:** Đọc bài viết | Article ID/Danh mục | Nội dung bài viết |
|  | **UC29:** Tham gia diễn đàn | Thread ID | Nội dung thảo luận |
|  | **UC30:** Đăng bài viết | Tiêu đề, nội dung, hình ảnh | Bài viết được đăng |

*2.1.3 Chức năng quản trị viên ( admin )*

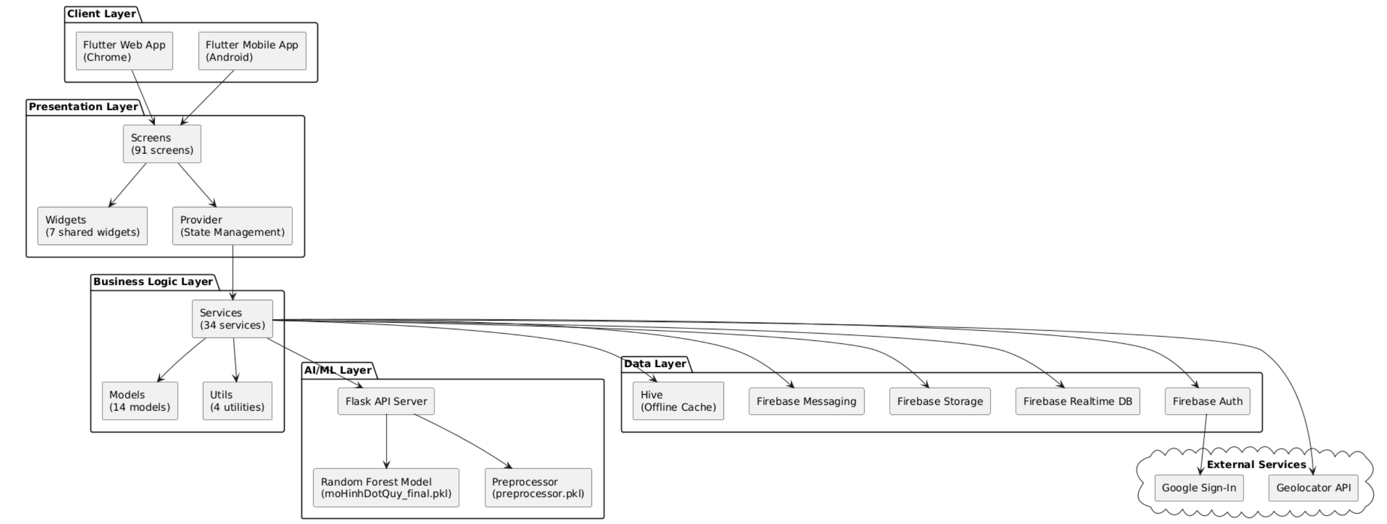
|  |  |  |  |
| --- | --- | --- | --- |
| **Nhóm chức năng** | **Tên chức năng (Use Case)** | **Đầu vào (Input)** | **Đầu ra (Output)** |
| **A. Xác thực** | **UC1:** Đăng ký tài khoản | Họ tên, email, mật khẩu, xác nhận mật khẩu | Tài khoản được tạo, chuyển đến đăng nhập |
|  | **UC2:** Đăng nhập Email | Email, mật khẩu | Đăng nhập thành công, vào Dashboard |
|  | **UC3:** Đăng nhập Google | Tài khoản Google | Đăng nhập thành công, tự tạo tài khoản nếu chưa có |
|  | **UC4:** Quên mật khẩu | Email đã đăng ký | Email chứa link đặt lại mật khẩu |
|  | **UC5:** Đổi mật khẩu | Mật khẩu cũ, mới, xác nhận mới | Mật khẩu được cập nhật |
| **B. Dự đoán AI** | **UC6:** Dự đoán nguy cơ đột quỵ | Tuổi, chỉ số sức khỏe, tiền sử bệnh, lối sống | Điểm nguy cơ, phân loại (BMI, huyết áp...), mức độ rủi ro |
|  | **UC7:** Dự đoán nguy cơ tiểu đường | Tuổi, chỉ số đường huyết, gia đình, hoạt động | Điểm nguy cơ, BMI, khuyến nghị |
|  | **UC8:** Xem kết quả dự đoán | Kết quả từ UC6 hoặc UC7 | Hiển thị điểm số, chỉ số chi tiết, khuyến nghị |
|  | **UC9:** Xem lịch sử dự đoán | User ID | Danh sách dự đoán theo thời gian |
| **C. Theo dõi sức khỏe** | **UC10:** Nhập chỉ số sức khỏe | Huyết áp, đường huyết, cân nặng, ghi chú | Dữ liệu lưu vào Firebase |
|  | **UC11:** Xem biểu đồ huyết áp | User ID, khoảng thời gian | Biểu đồ đường (fl\_chart) |
|  | **UC12:** Xem biểu đồ đường huyết | User ID, khoảng thời gian | Biểu đồ đường kèm ngưỡng cảnh báo |
|  | **UC13:** Xem biểu đồ BMI | User ID, khoảng thời gian | Biểu đồ BMI phân loại |
|  | **UC14:** Xem lịch sử sức khỏe | User ID | Danh sách bản ghi theo ngày |
| **D. Kết nối bác sĩ** | **UC15:** Xem danh sách bác sĩ | Bộ lọc (chuyên khoa, đánh giá) | Danh sách bác sĩ cơ bản |
|  | **UC16:** Xem chi tiết bác sĩ | Doctor ID | Thông tin, chuyên khoa, lịch làm việc |
|  | **UC17:** Đặt lịch hẹn | Doctor ID, thời gian, lý do | Yêu cầu được tạo, chờ duyệt |
|  | **UC18:** Chat với bác sĩ | Tin nhắn, hình ảnh | Tin nhắn gửi realtime (Firebase) |
|  | **UC19:** Đánh giá bác sĩ | Số sao, nhận xét | Đánh giá được lưu |
| **E. SOS khẩn cấp** | **UC20:** Gửi yêu cầu SOS | Vị trí GPS, mô tả triệu chứng | Yêu cầu tạo, báo bác sĩ trực |
|  | **UC21:** Theo dõi trạng thái SOS | SOS ID | Trạng thái xử lý, thông tin bác sĩ |
| **F. Quản lý gia đình** | **UC22:** Tạo nhóm gia đình | Tên nhóm | Nhóm được tạo (User là admin) |
|  | **UC23:** Mời thành viên | Email hoặc mã mời | Lời mời được gửi |
|  | **UC24:** Theo dõi sức khỏe người thân | Member ID | Thông tin sức khỏe thành viên |
| **G. Nhắc nhở thuốc** | **UC25:** Tạo nhắc nhở | Tên thuốc, liều, thời gian, tần suất | Local notification được lên lịch |
|  | **UC26:** Sửa nhắc nhở | Reminder ID, thông tin mới | Nhắc nhở cập nhật |
|  | **UC27:** Xóa nhắc nhở | Reminder ID | Nhắc nhở bị xóa |
| **H. Kiến thức & Cộng đồng** | **UC28:** Đọc bài viết | Article ID/Danh mục | Nội dung bài viết |
|  | **UC29:** Tham gia diễn đàn | Thread ID | Nội dung thảo luận |
|  | **UC30:** Đăng bài viết | Tiêu đề, nội dung, hình ảnh | Bài viết được đăng |

## 2.2. Yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **Danh mục** | **Yêu cầu (Requirement)** | **Mô tả / Tiêu chí chi tiết** |
| **Hiệu năng** | **NFR1:** Thời gian phản hồi | Tải màn hình < 2s; Dự đoán AI < 3s; Chat < 1s 256 |
|  | **NFR2:** Khả năng chịu tải | Hỗ trợ 1000 users đồng thời; Firebase tự động scale 257 |
|  | **NFR3:** Tối ưu hóa | Cache ảnh (cached\_network\_image), cache offline (Hive), Lazy loading 258 |
| **Bảo mật** | **NFR4:** Xác thực | Firebase Auth (JWT), Đăng nhập 2 yếu tố, Token auto refresh 259 |
|  | **NFR5:** Phân quyền | Role (User, Doctor, Admin); Kiểm tra Client & Server 260 |
|  | **NFR6:** Mã hóa | SHA-256 cho mật khẩu; HTTPS; Firebase Security Rules 261 |
| **Khả dụng** | **NFR7:** Uptime | 99.9% (Firebase SLA) 262 |
|  | **NFR8:** Offline mode | Hoạt động offline (cache); Tự đồng bộ khi có mạng 263 |
|  | **NFR9:** Backup | Tự động backup (Firebase); Export JSON 264 |
| **Mở rộng** | **NFR10:** Kiến trúc | Module hóa; Tách biệt UI/Logic/Data; Provider pattern 265 |
|  | **NFR11:** API | Flask API scale độc lập; Hỗ trợ model AI mới 266 |
| **Giao diện** | **NFR12:** Responsive | Hỗ trợ mobile và web; Đa dạng kích thước màn hình 267 |
|  | **NFR13:** Accessibility | Chỉnh font size; Độ tương phản cao; Hỗ trợ tiếng Việt 268 |
|  | **NFR14:** UX | Material Design 3; Màu chủ đạo #135BEC; Animation mượt (300ms) 269 |

# PHẦN 3: THIẾT KẾ HỆ THỐNG

## 3.1. Kiến trúc hệ thống

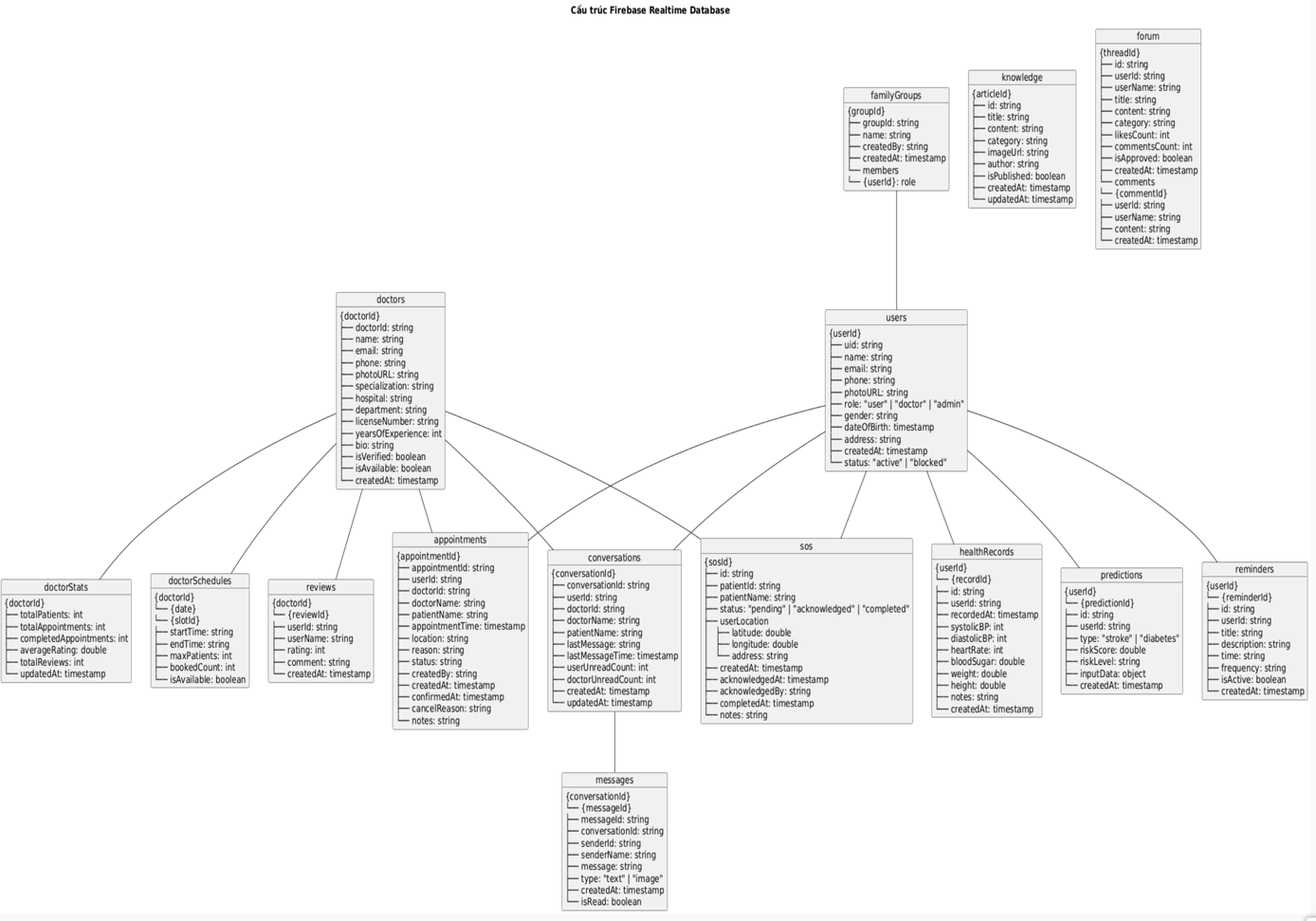
Hệ thống SEWS sử dụng kiến trúc phân lớp (Layered Architecture) kết hợp mô hình Client-Server.

|  |  |
| --- | --- |
| **Lớp (Layer)** | **Thành phần chính & Mô tả** |
| **A. Client Layer** | **Flutter Mobile App** (Android API 23+) và **Flutter Web App** (Chrome)2. |
| **B. Presentation Layer** | Gồm **91 màn hình** (User, Doctor, Admin), các **Widgets** tái sử dụng và **Provider** để quản lý trạng thái 3. |
| **C. Business Logic Layer** | **34 Services** xử lý nghiệp vụ, **14 Models** dữ liệu và các Utils hỗ trợ4. |
| **D. Data Layer** | **Firebase** (Auth, Realtime DB, Storage, Cloud Messaging) và **Hive** (Local database) 5. |
| **E. AI/ML Layer** | **Flask API Server** chạy Python, chứa Model **Random Forest** dự đoán đột quỵ6. |

 **Luồng dữ liệu:** UI -> Provider -> Service -> Firebase/API -> Response -> Provider -> UI.

 **Giao tiếp:** Client dùng Firebase SDK (realtime) và HTTP REST API (JSON) để giao tiếp với Server. Hỗ trợ đồng bộ offline bằng Hive

## 3.2. Thiết kế cơ sở dữ liệu

Hệ thống sử dụng Firebase Realtime Database (NoSQL).

|  |  |
| --- | --- |
| **Node (Nút dữ liệu)** | **Mô tả & Các trường chính** |
| **users** | Thông tin bệnh nhân: tên, email, sđt, địa chỉ, ngày sinh11. |
| **doctors** | Thông tin bác sĩ: chuyên khoa, bệnh viện, chứng chỉ, kinh nghiệm12. |
| **doctorStats** | Thống kê bác sĩ: số bệnh nhân, lịch hẹn, đánh giá13. |
| **appointments** | Lịch hẹn: thời gian, trạng thái (pending, confirmed...), lý do14. |
| **healthRecords** | Chỉ số sức khỏe: huyết áp, đường huyết, cân nặng, nhịp tim15. |
| **predictions** | Lịch sử dự đoán AI: loại (đột quỵ/tiểu đường), điểm nguy cơ16. |
| **conversations** | Thông tin hội thoại chat: người tham gia, tin nhắn cuối17. |
| **messages** | Chi tiết tin nhắn: nội dung (text/image), người gửi, thời gian18. |
| **sos** | Ca khẩn cấp: vị trí GPS, trạng thái, bác sĩ tiếp nhận19. |
| **familyGroups** | Nhóm gia đình và danh sách thành viên20. |
| **reminders** | Nhắc nhở thuốc: tên thuốc, thời gian, tần suất21. |
| **knowledge / forum** | Bài viết kiến thức và bài đăng diễn đàn (kèm bình luận) 22. |
| **doctorSchedules** | Lịch làm việc của bác sĩ theo slot23. |
| **reviews** | Đánh giá và nhận xét về bác sĩ24. |

## 3.3. Thiết kế giao diện (UI/UX)

**Nguyên tắc:** Sử dụng Material Design 3, màu chủ đạo Xanh dương (#135BEC)25. Thiết kế Responsive (Mobile & Web)26. Hỗ trợ Accessibility (chỉnh font, tương phản cao)27.

|  |  |  |
| --- | --- | --- |
| **Module** | **Số lượng** | **Thành phần chính** |
| **User** | 54 màn hình | Splash, Auth, Dashboard, Health Hub, Prediction Hub, Doctors Hub, Profile 28. |
| **Doctor** | 17 màn hình | Dashboard, Patient List, Appointment, SOS Queue, Chat, Schedule 29. |
| **Admin** | 20 màn hình | Dashboard, Quản lý (User, Doctor, Patient, SOS, Knowledge, Community) 30. |

Điều hướng: User và Doctor dùng Bottom Navigation Bar (4 tab). Admin dùng Sidebar Navigation

**Bộ giao diện dành cho người dùng ( User )**

Phân hệ người dùng bao gồm **54 màn hình** được thiết kế tuân thủ **Material Design 3** , sử dụng màu chủ đạo Xanh dương (#135BEC). Điều hướng chính được thực hiện qua **Bottom Navigation Bar**.

Một số mẫu giao diện đã được chèn vào minh hoạ :

1. Xác thực và Hồ sơ (Auth & Profile)

* **Tạo Tài khoản & Đăng nhập:** Hỗ trợ xác thực qua Email/Password và Google Sign-In.
* **Quên Mật khẩu & Đổi mật khẩu:** Cung cấp luồng bảo mật tiêu chuẩn.
* **Hồ sơ:** Quản lý thông tin cá nhân (tuổi, giới tính, tiền sử bệnh).

1. Dự đoán Nguy cơ (Prediction Hub)

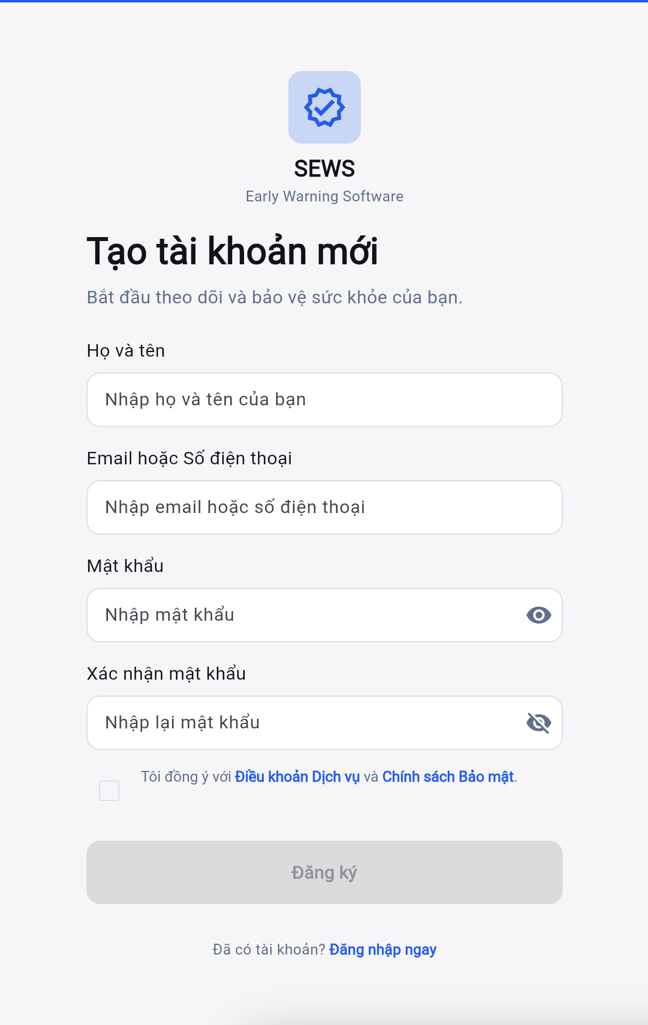
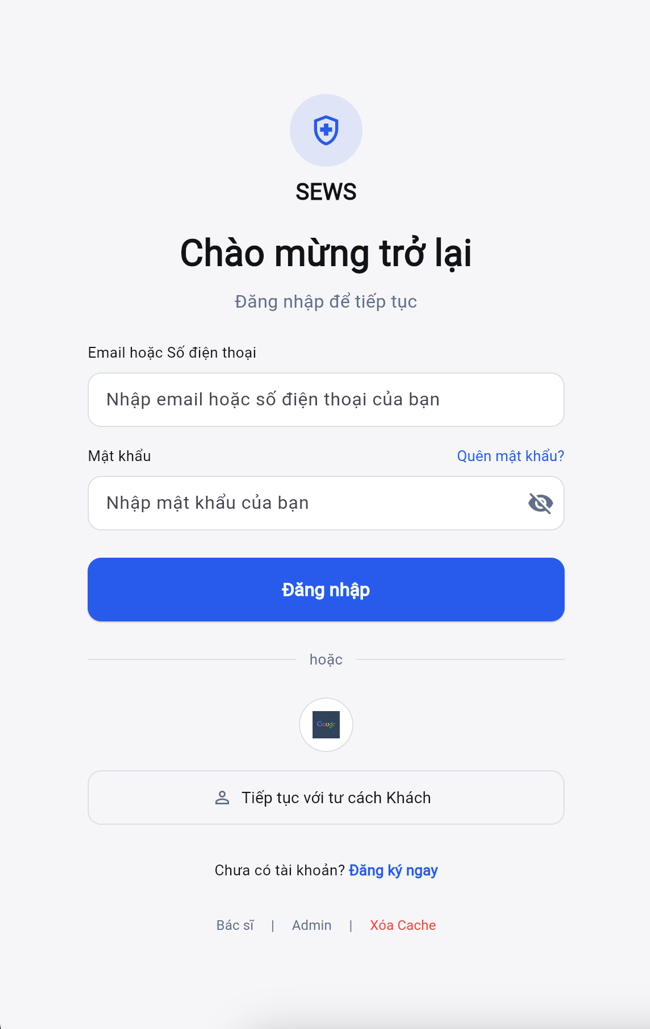
* **Dự đoán Đột quỵ:** Màn hình nhập liệu **14 chỉ số** sức khỏe và lối sống.
* **Dự đoán Tiểu đường:** Màn hình thu thập **6 yếu tố nguy cơ** để đánh giá.
* **Kết quả & Lịch sử:** Hiển thị Điểm nguy cơ, Mức độ cảnh báo (Thấp/Trung/Cao) và khuyến nghị y tế.

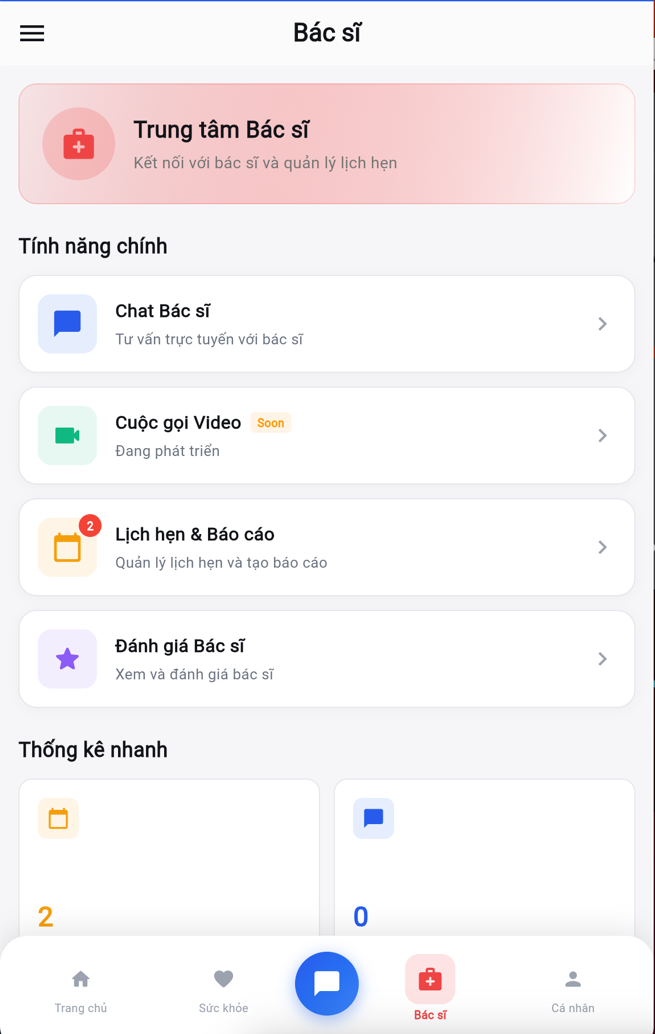
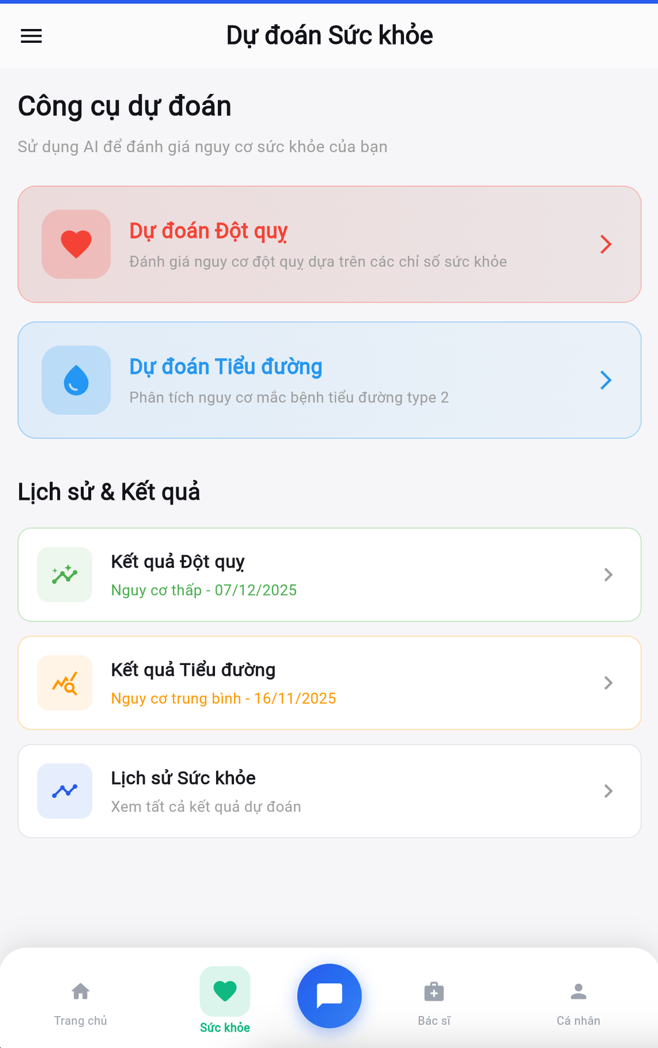
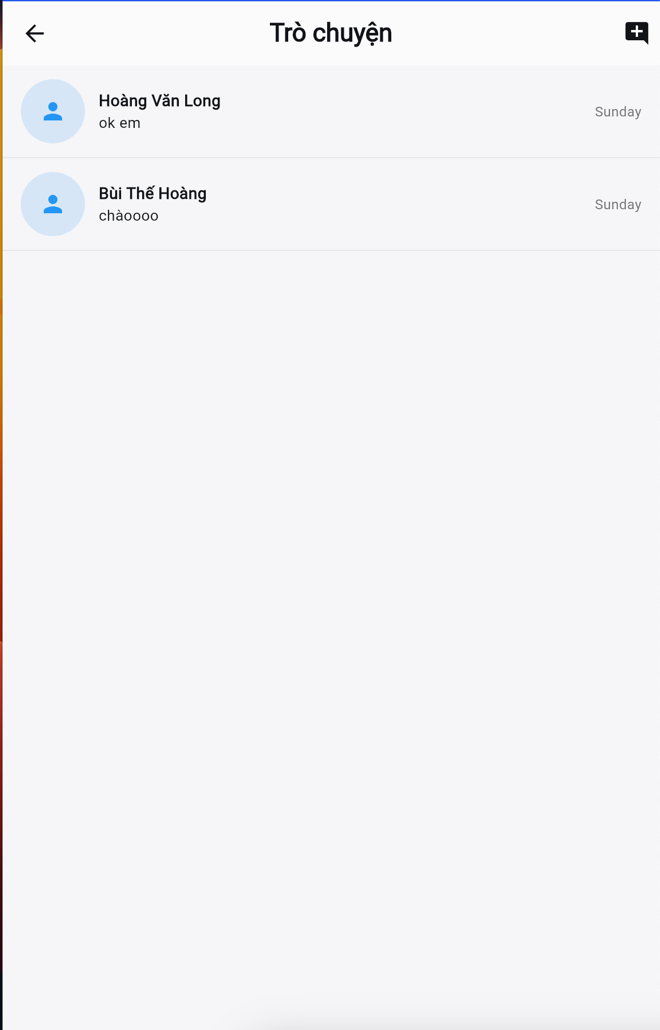
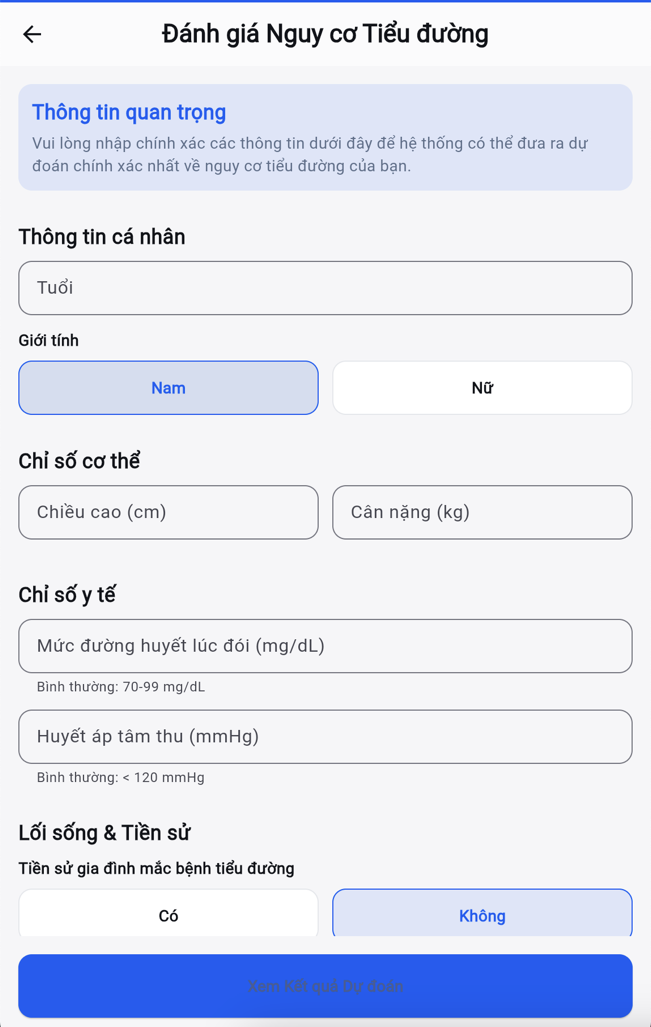
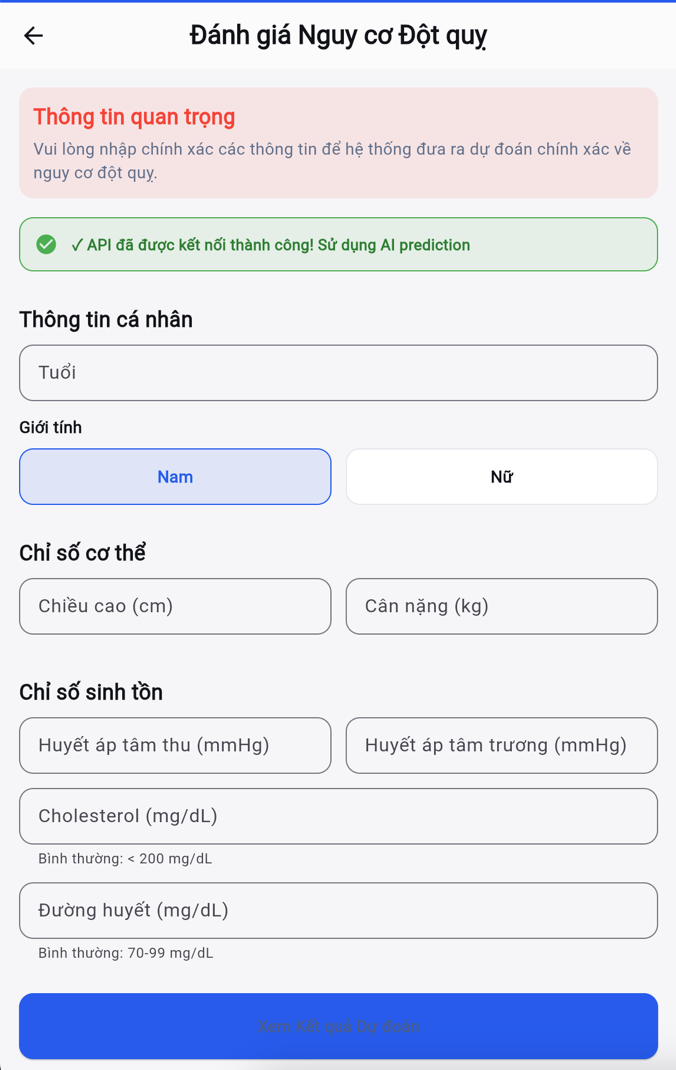
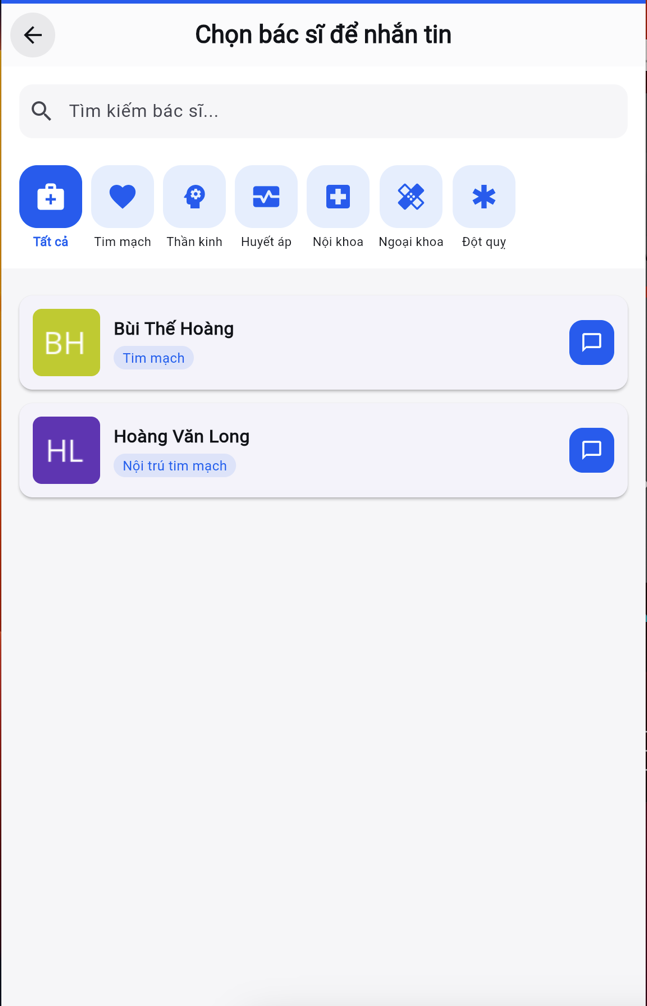
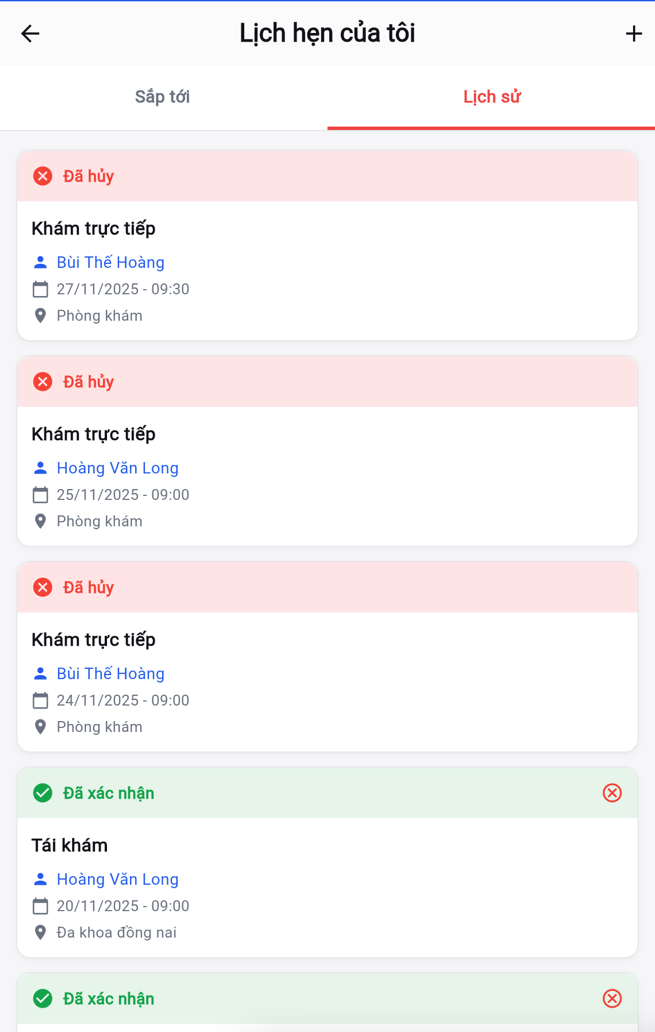
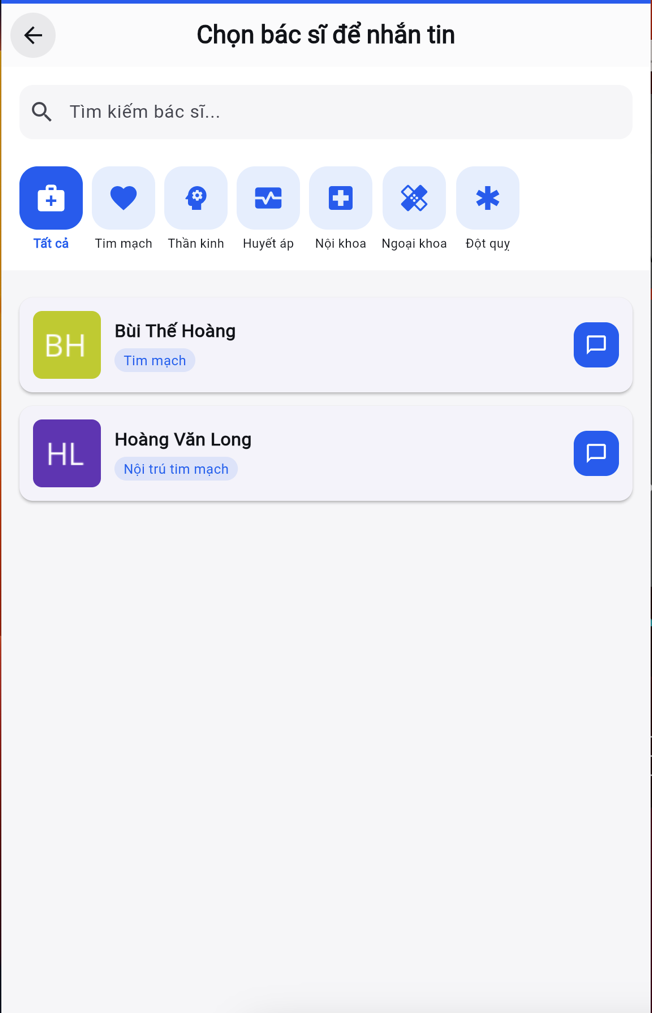
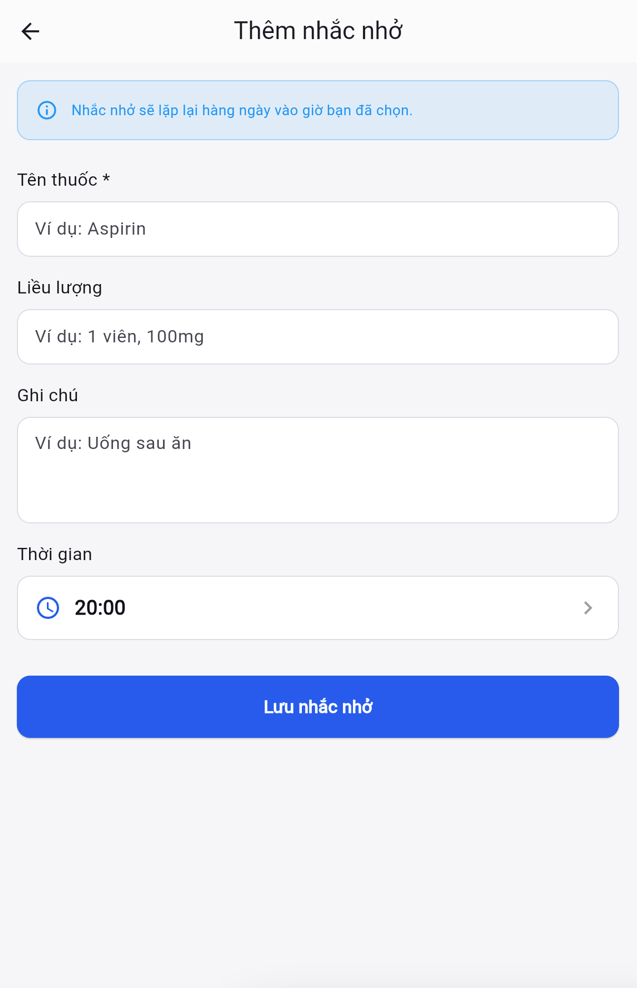
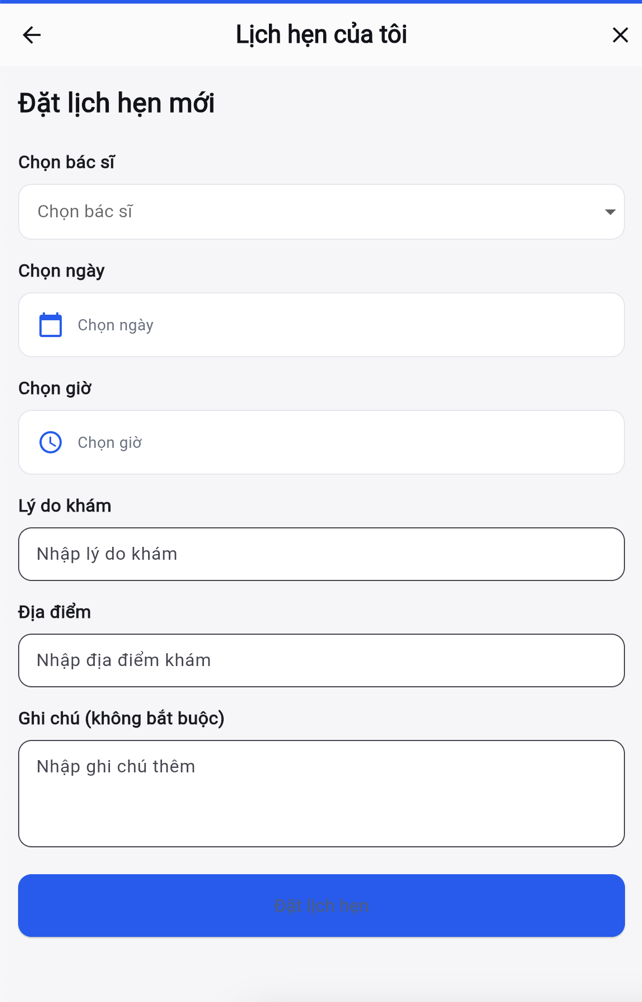
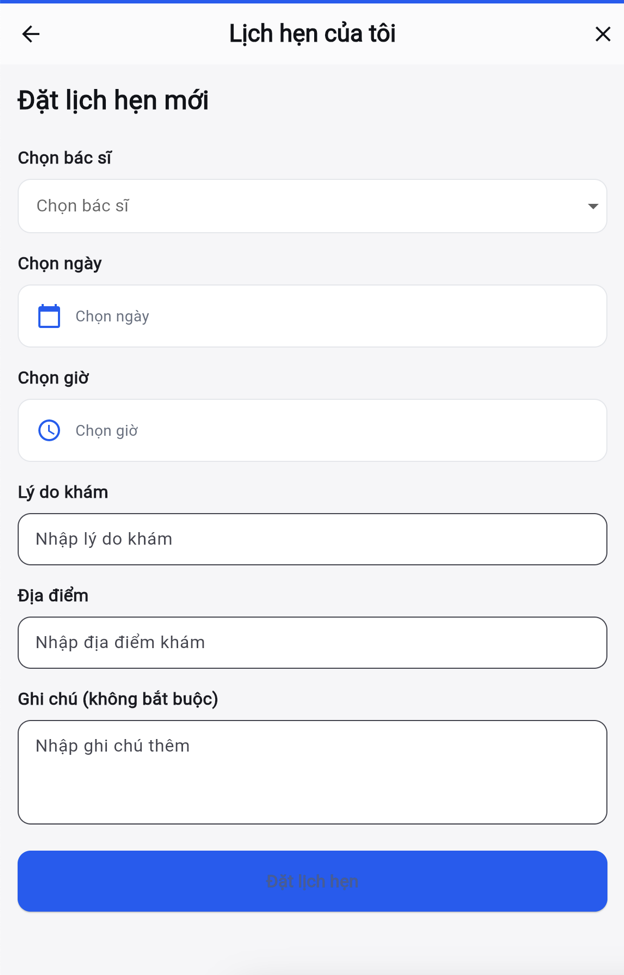
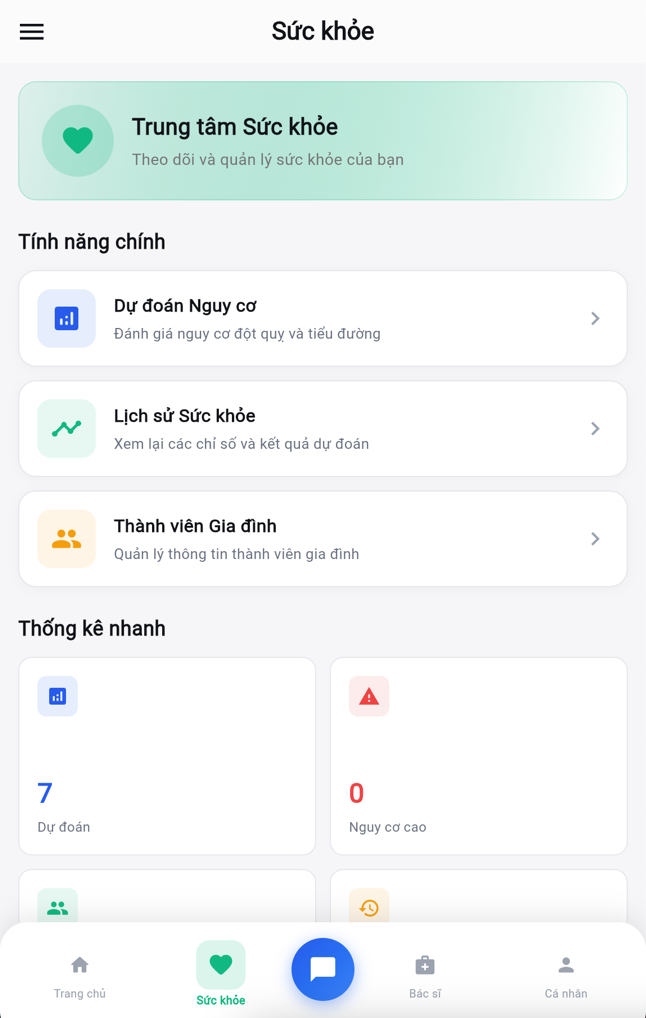
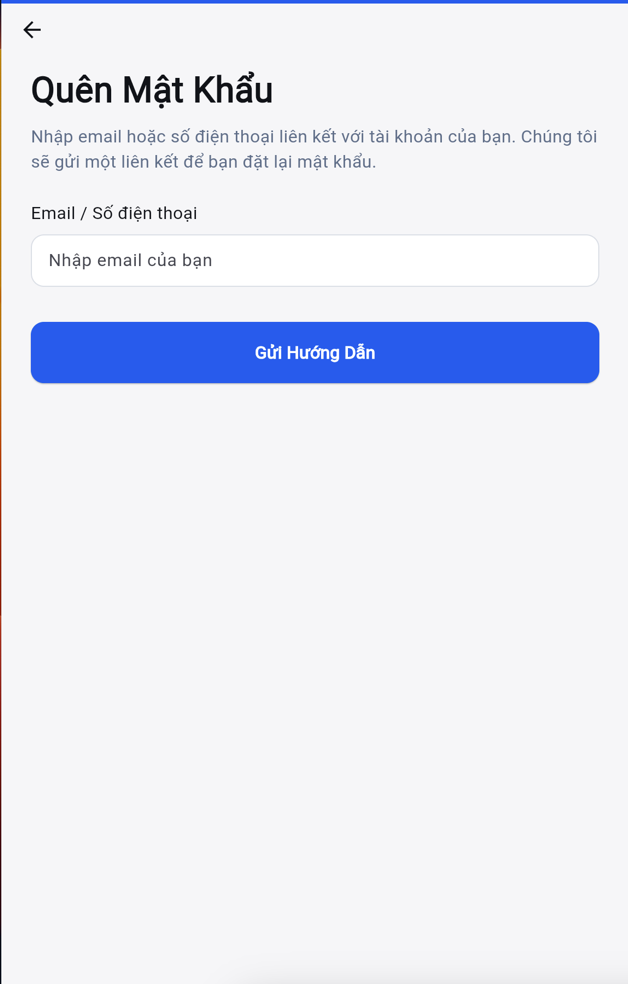
1. Theo dõi Sức khỏe (Health Hub)

* **Nhập liệu Chỉ số:** Giao diện nhập Huyết áp, Đường huyết, Cân nặng.
* **Biểu đồ Trực quan:** Hiển thị biến thiên chỉ số theo thời gian (7/30/90 ngày) sử dụng thư viện **fl\_chart**.
* **Lịch sử Sức khỏe:** Danh sách các bản ghi sức khỏe theo ngày.

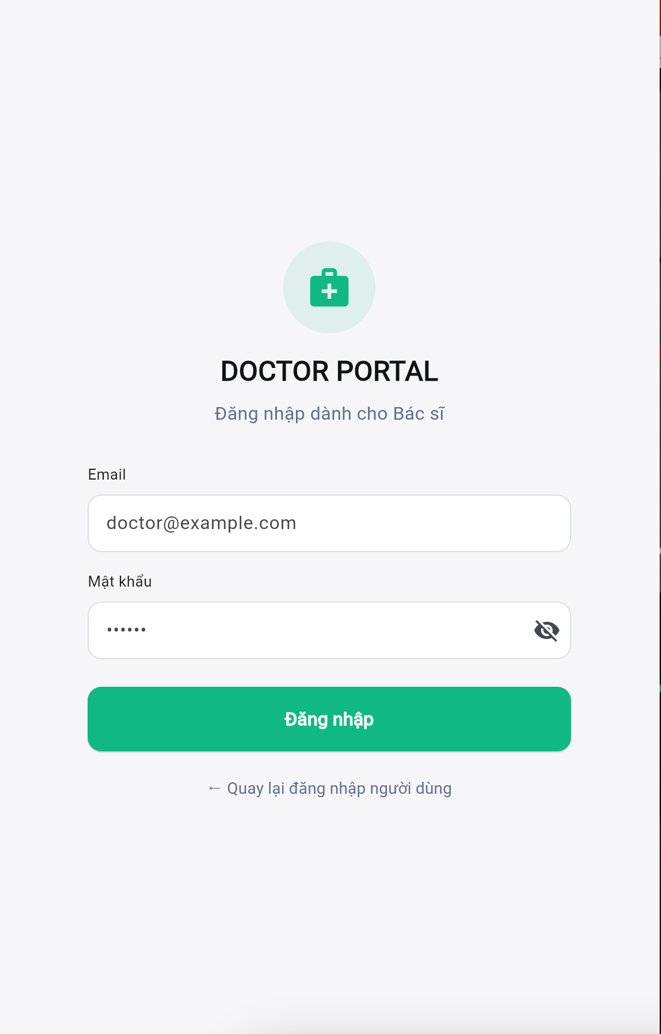
1. Kết nối Y tế và Khẩn cấp (Doctor & SOS)

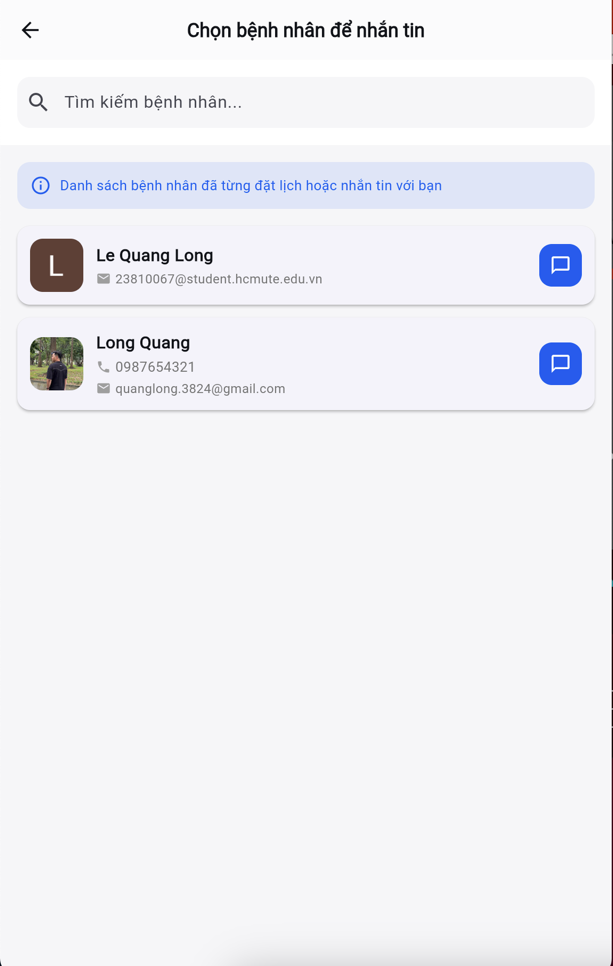
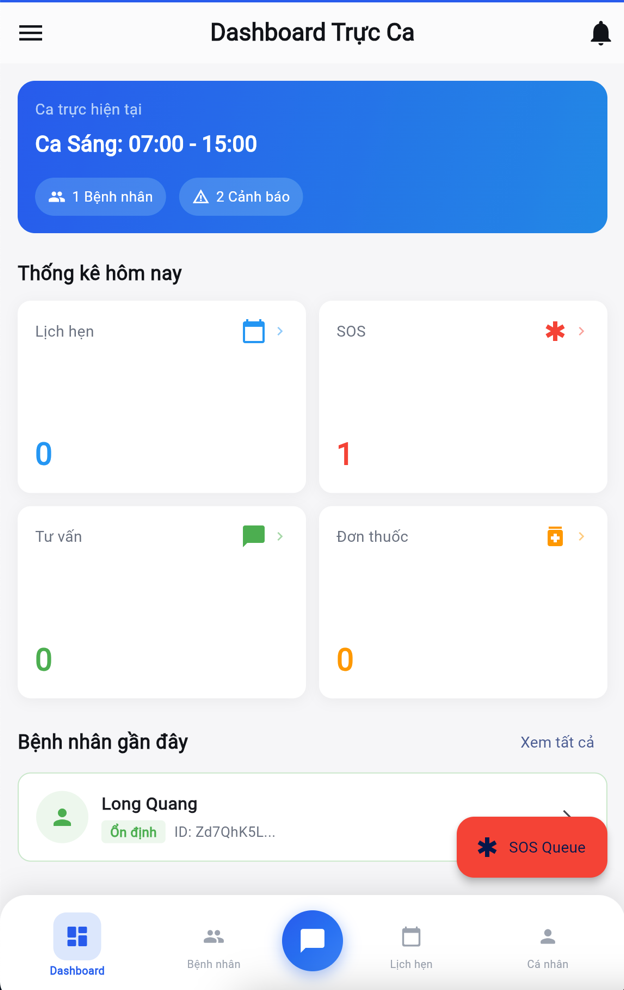
* **Doctors Hub:** Danh sách, chi tiết Bác sĩ và Đánh giá.
* **Đặt Lịch hẹn:** Chọn slot khám, nhập lý do, theo dõi trạng thái lịch hẹn.
* **Chat Trực tuyến:** Giao diện nhắn tin realtime (text/image) với bác sĩ.
* **SOS Khẩn cấp:** Nút kích hoạt nhanh chóng, gửi vị trí **GPS** và mô tả triệu chứng đến bác sĩ trực.

** **

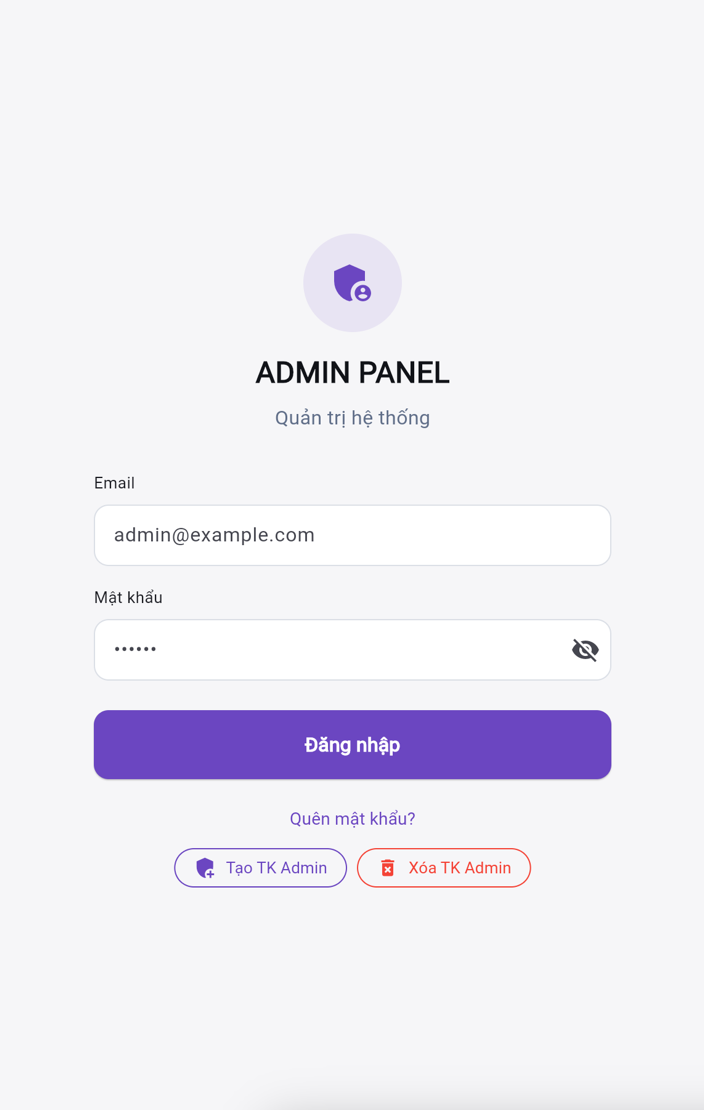
**      **

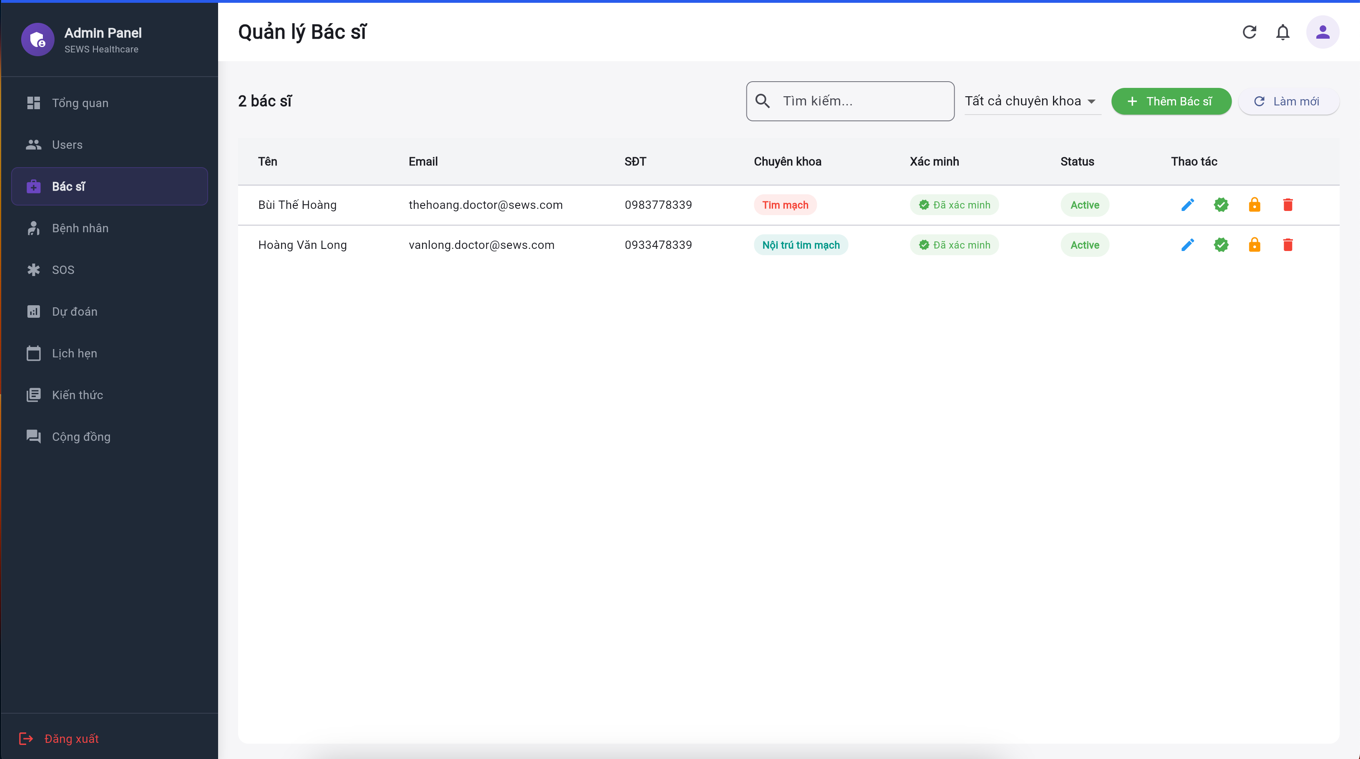
**Bộ giao diện dành cho Bác Sĩ ( Doctor)**

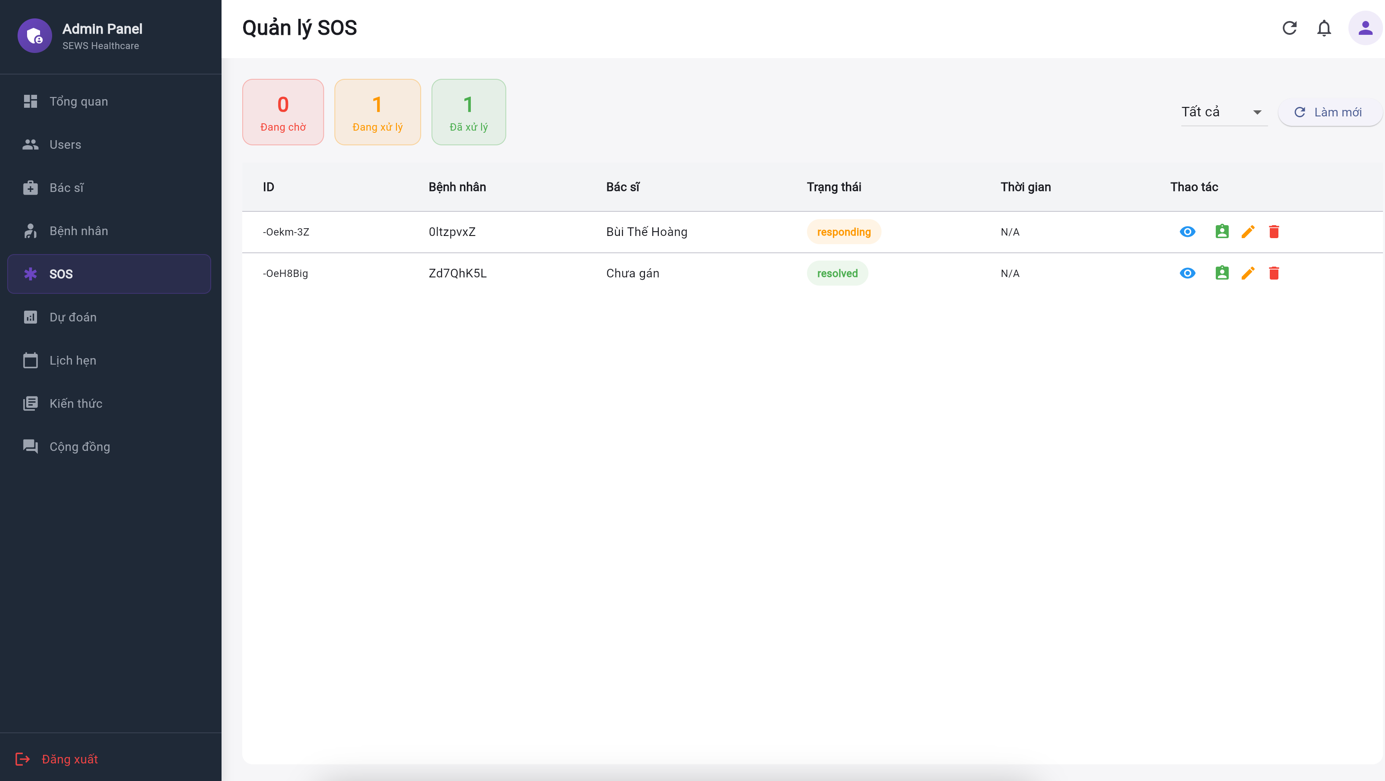
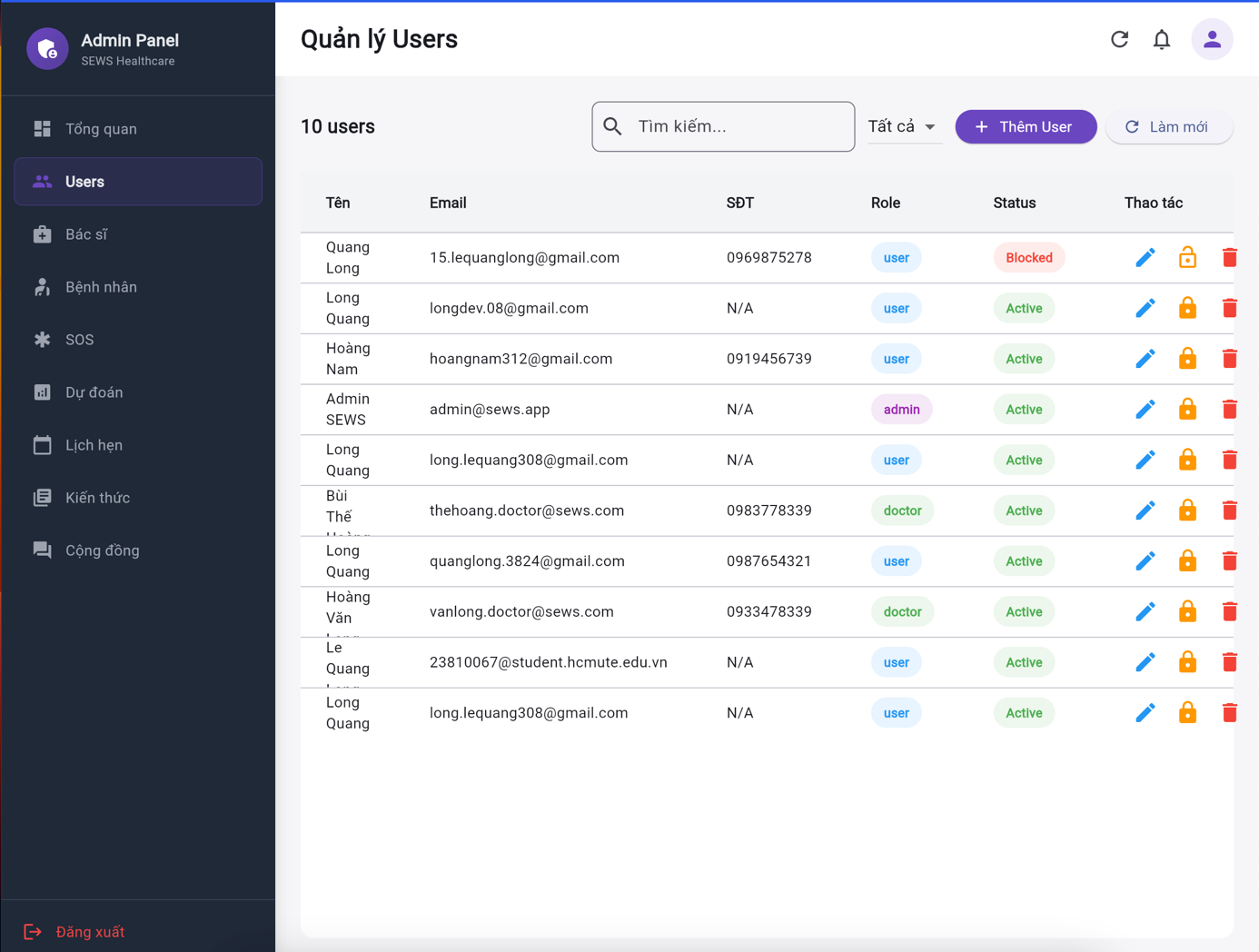
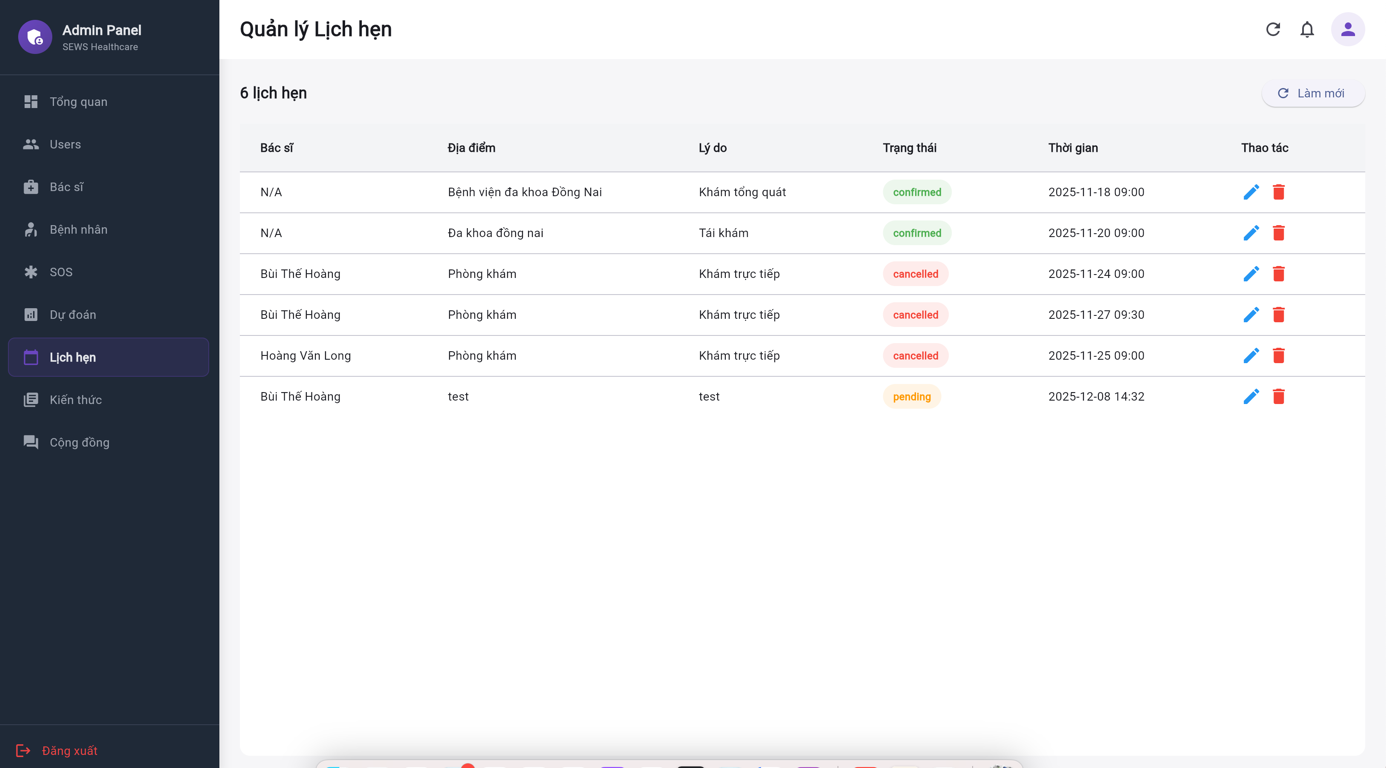
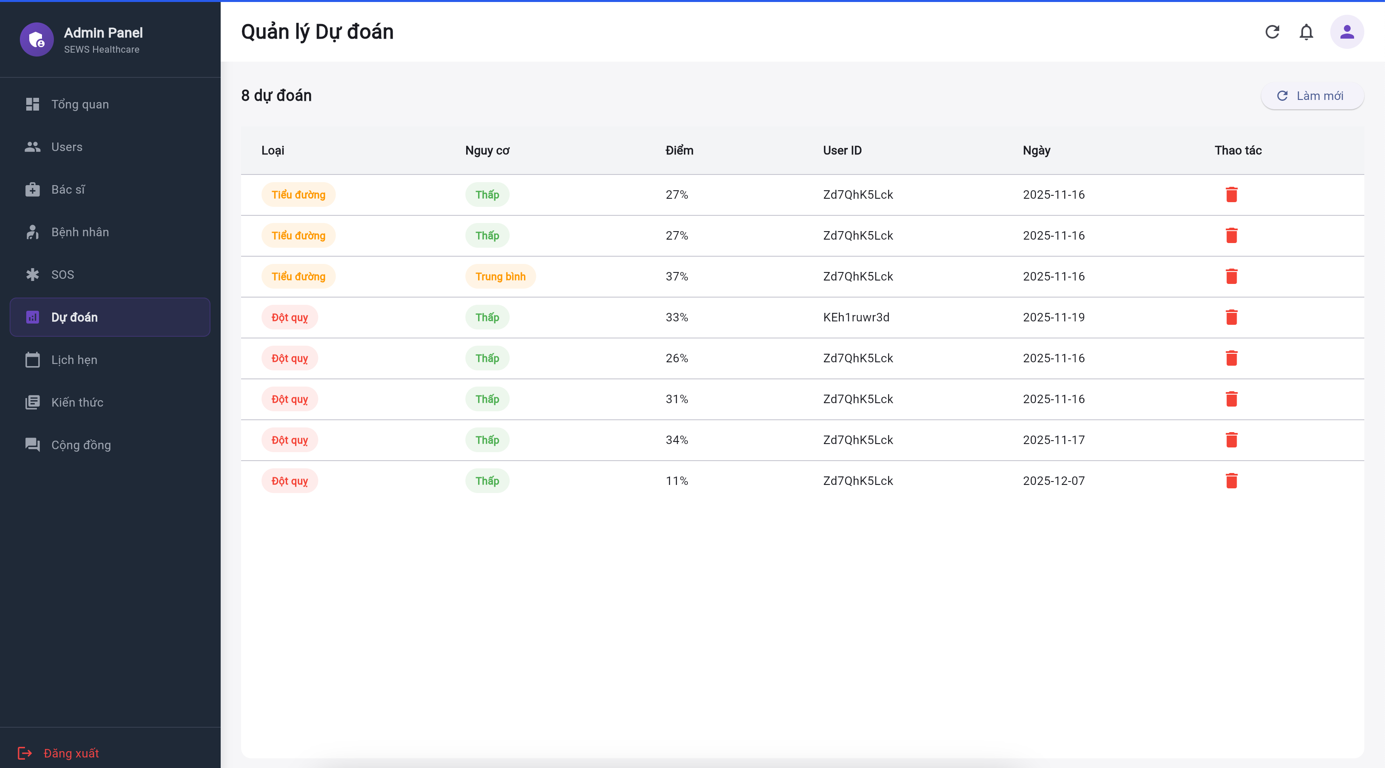
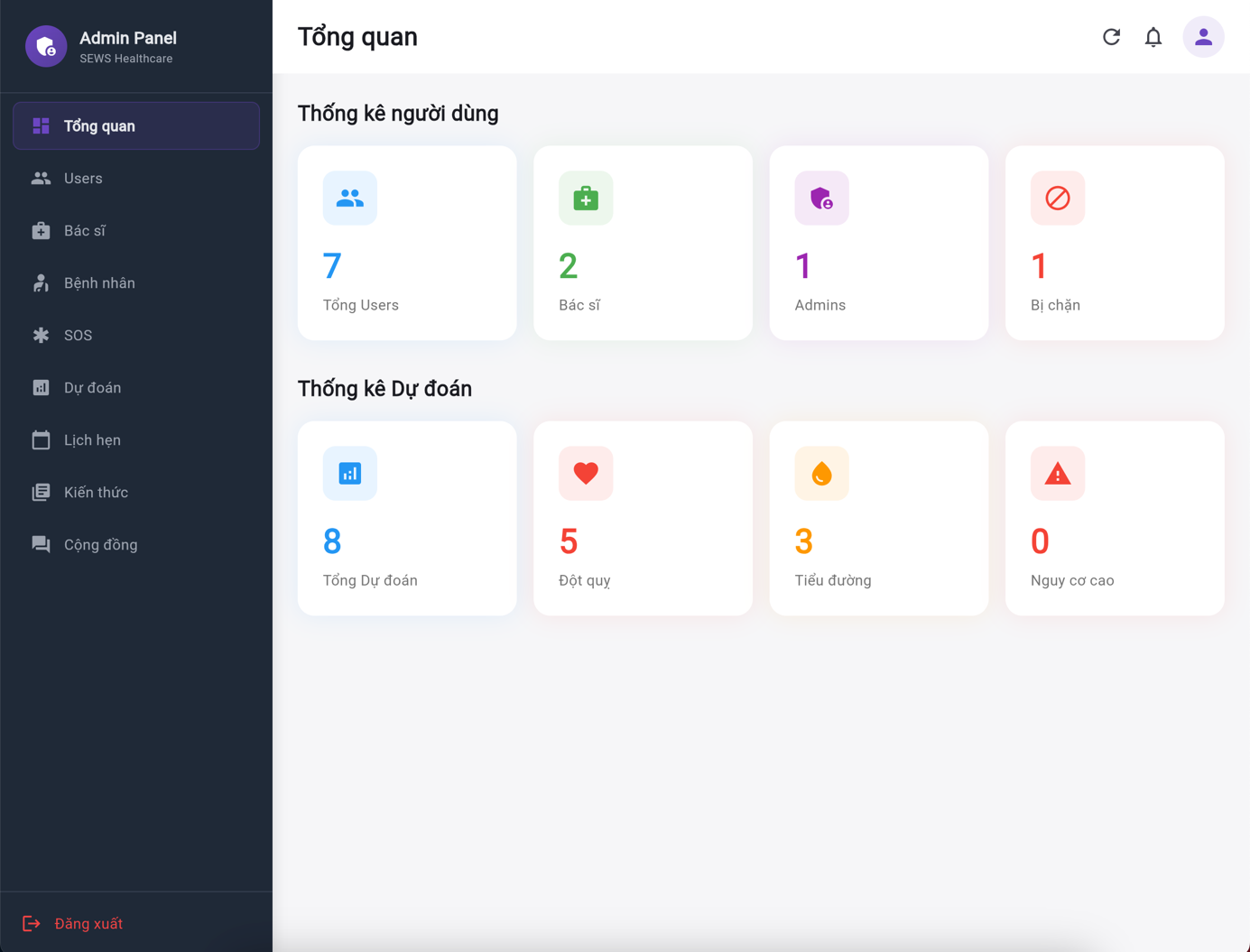
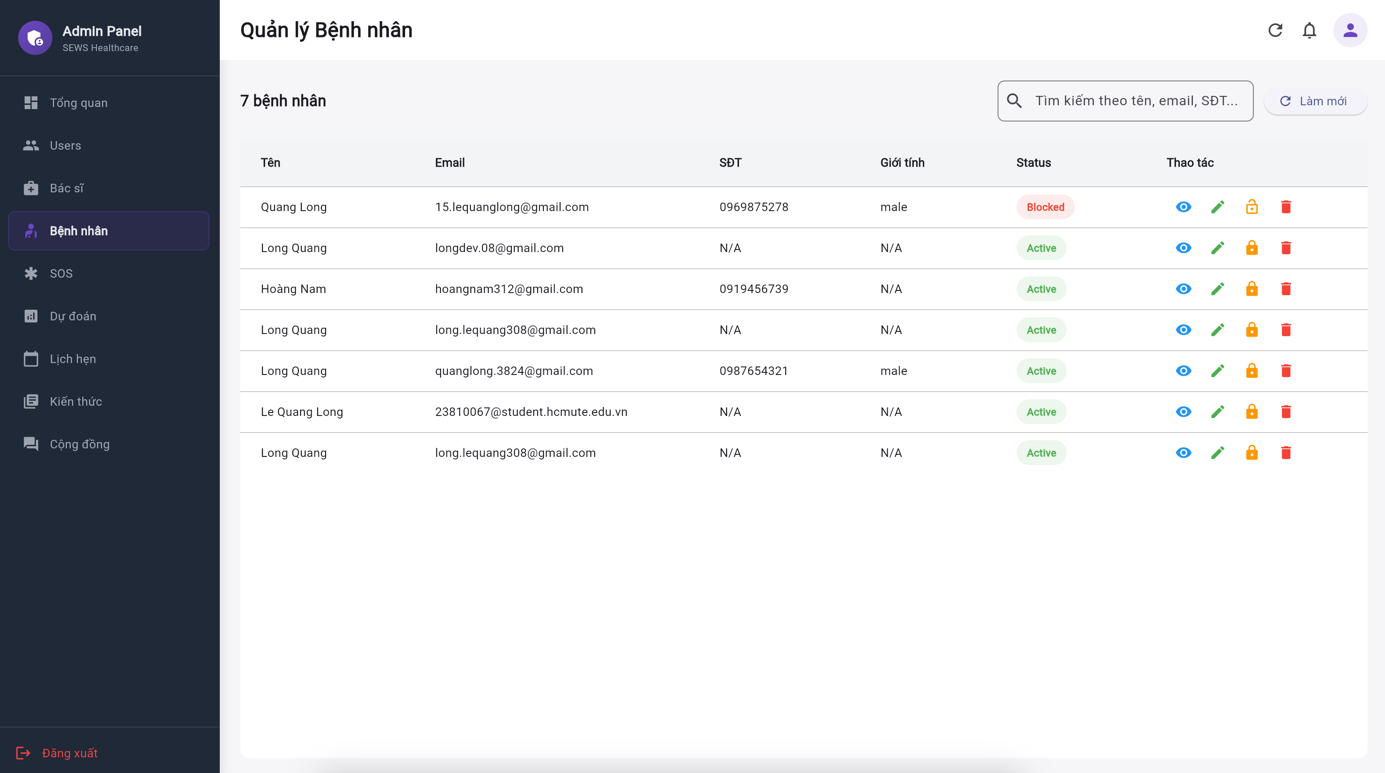
****

** **

**Bộ giao diện dành cho Admin ( Quản trị viên )**

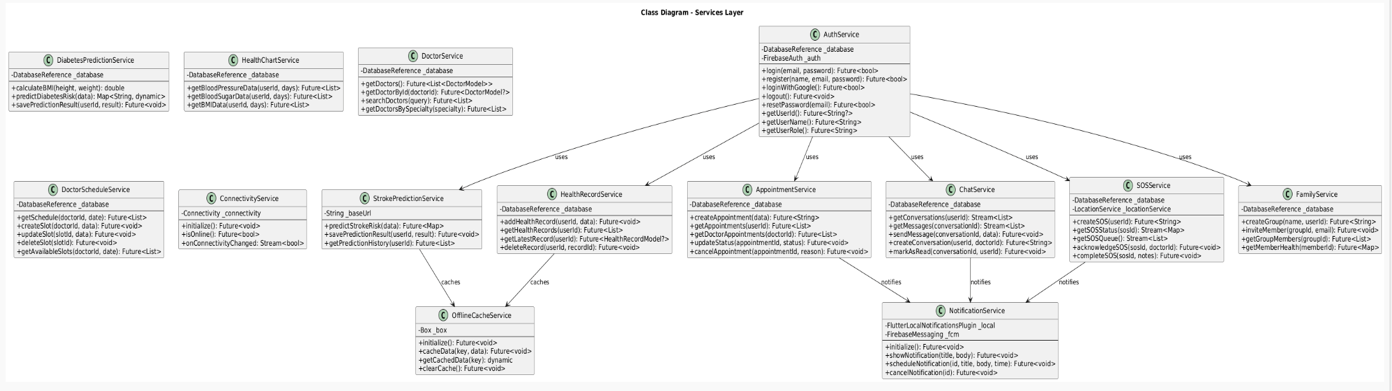
****

****

****

## 3.4. Biểu đồ lớp (Class Diagram)

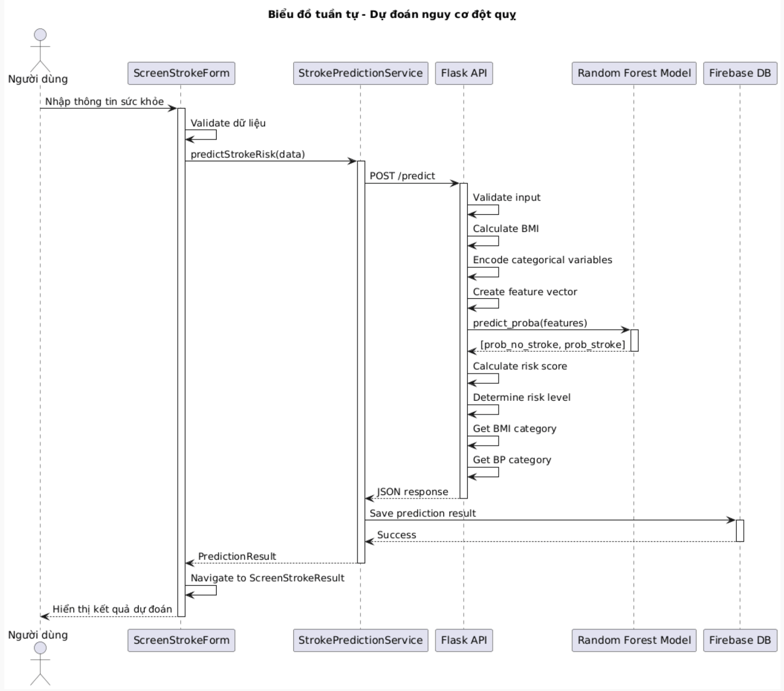
|  |  |
| --- | --- |
| **Model Class** | **Thuộc tính chính** |
| **UserModel** | id, name, email, phone, role, gender, dateOfBirth32. |
| **DoctorModel** | doctorId, specialization, hospital, licenseNumber, bio, isAvailable33. |
| **DoctorStatsModel** | totalPatients, totalAppointments, averageRating34. |
| **AppointmentModel** | appointmentId, doctorId, patientName, time, status, reason35. |
| **HealthRecordModel** | systolicBP, diastolicBP, bloodSugar, weight, notes36. |
| **PredictionModel** | riskScore, riskLevel, inputData, type (stroke/diabetes)37. |
| **SOSCaseModel** | status, latitude, longitude, address, priority38. |
| **Chat Models** | **ConversationModel** (lastMessage), **MessageModel** (content, type)39. |

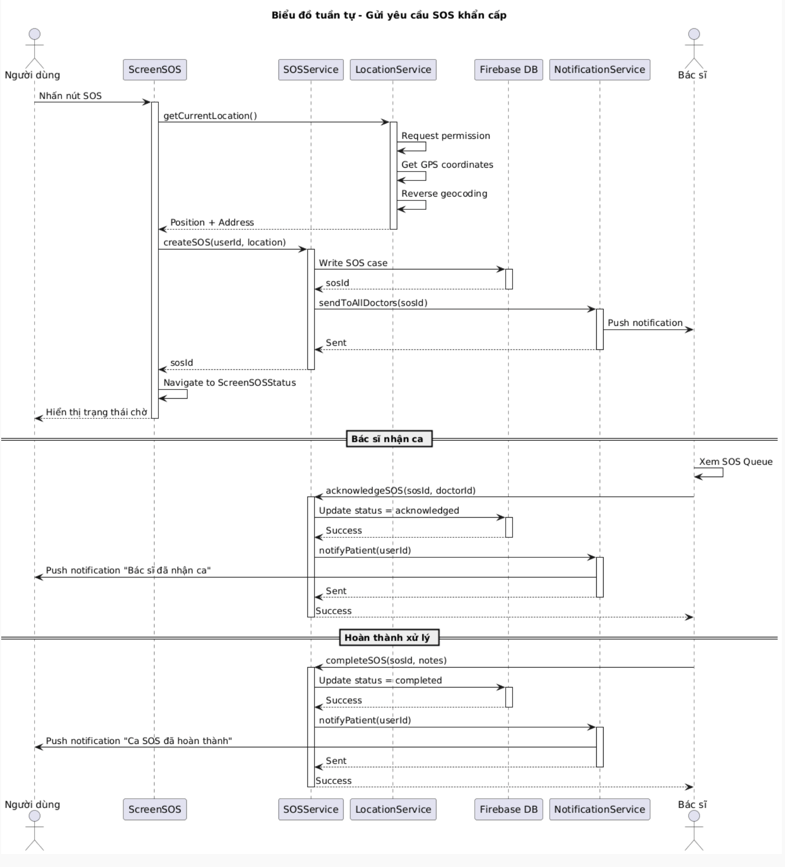


|  |  |
| --- | --- |
| **Service Class** | **Chức năng chính** |
| **AuthService** | Đăng nhập (Email/Google), đăng ký, quên mật khẩu40. |
| **PredictionService** | StrokePredictionService, DiabetesPredictionService (tính toán, gọi API)41. |
| **HealthService** | HealthRecordService (CRUD chỉ số), HealthChartService (dữ liệu biểu đồ)42. |
| **DoctorService** | Quản lý thông tin bác sĩ, tìm kiếm, quản lý lịch làm việc (DoctorScheduleService)43. |
| **AppointmentService** | Tạo, duyệt, hủy và quản lý trạng thái lịch hẹn44. |
| **SOSService** | Tạo yêu cầu SOS, theo dõi trạng thái, tiếp nhận ca45. |
| **ChatService** | Gửi/nhận tin nhắn realtime, quản lý hội thoại46. |
| **Utility Services** | NotificationService (push noti), OfflineCacheService (Hive), ConnectivityService 47. |

## 3.5. Biểu đồ tuần tự (Sequence Diagram)

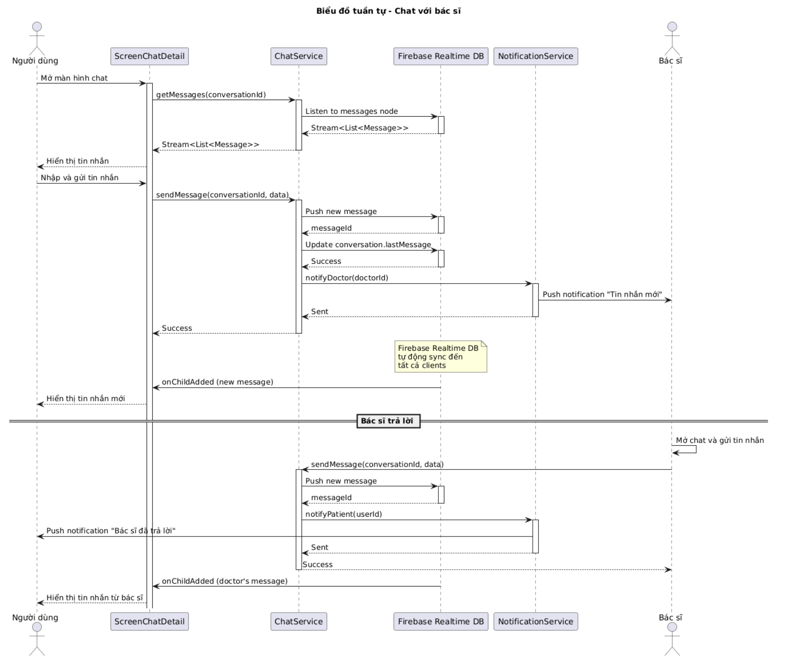
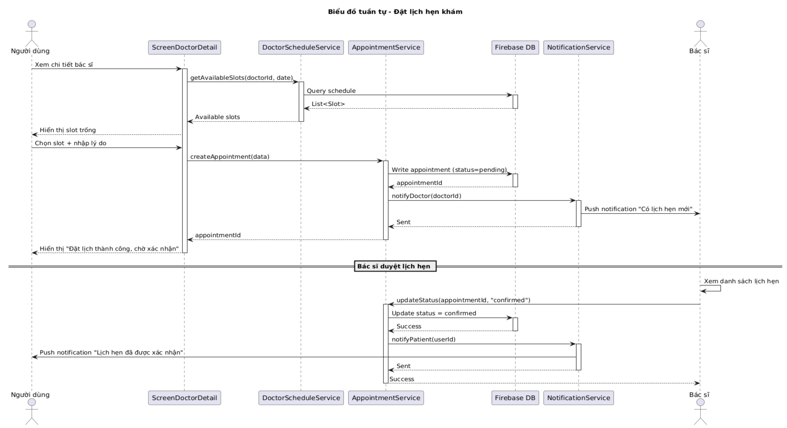
User nhập liệu -> Service gọi Flask API -> Model Random Forest xử lý -> Trả về kết quả & Risk Score -> Lưu Firebase -> Hiển thị .





User nhấn SOS -> Lấy GPS -> Tạo ca trên Firebase -> Thông báo tới Bác sĩ -> Bác sĩ nhận & xử lý -> Hoàn thành

User chọn Slot -> Tạo lịch (Pending) -> Bác sĩ nhận thông báo -> Bác sĩ duyệt (Confirmed) -> Thông báo lại cho User .

****

User gửi tin -> Lưu vào firebase -> Sync Realtime -> Bác sĩ nhận thông báo & Trả lời -> Sync về User

# PHẦN 4: CÔNG NGHỆ SỬ DỤNG

## 4.1. Nền tảng phát triển ứng dụng di động (Frontend)

Để xây dựng ứng dụng phía người dùng (Frontend), nhóm thực hiện đã lựa chọn **Flutter Framework** phiên bản SDK 3.5.4+ kết hợp với ngôn ngữ lập trình Dart 3.5.4+. Flutter là bộ công cụ phát triển đa nền tảng (cross-platform) của Google, cho phép tạo ra ứng dụng native cho cả mobile và web chỉ từ một mã nguồn duy nhất. Việc sử dụng Flutter mang lại lợi thế lớn về mặt hiệu năng nhờ khả năng biên dịch trực tiếp sang mã máy (native code ARM/x86) mà không cần qua cầu nối (bridge) như React Native, đảm bảo ứng dụng hoạt động mượt mà ở mức 60fps. Bên cạnh đó, tính năng Hot Reload giúp nhóm tăng tốc độ phát triển bằng cách cập nhật giao diện tức thì mà vẫn giữ nguyên trạng thái (state) của ứng dụng.

Về kiến trúc quản lý trạng thái (State Management), dự án áp dụng mô hình **Provider**. Đây là mô hình được Google đề xuất nhờ sự đơn giản, hiệu năng tốt thông qua cơ chế ChangeNotifier và khả năng tích hợp chặt chẽ với cây Widget của Flutter. Cấu trúc dự án được tổ chức khoa học trong thư mục lib/, phân chia rõ ràng thành các module như features (chứa 54 màn hình User, 17 màn hình Doctor, 20 màn hình Admin), data (models, providers), và services để đảm bảo tính dễ bảo trì.

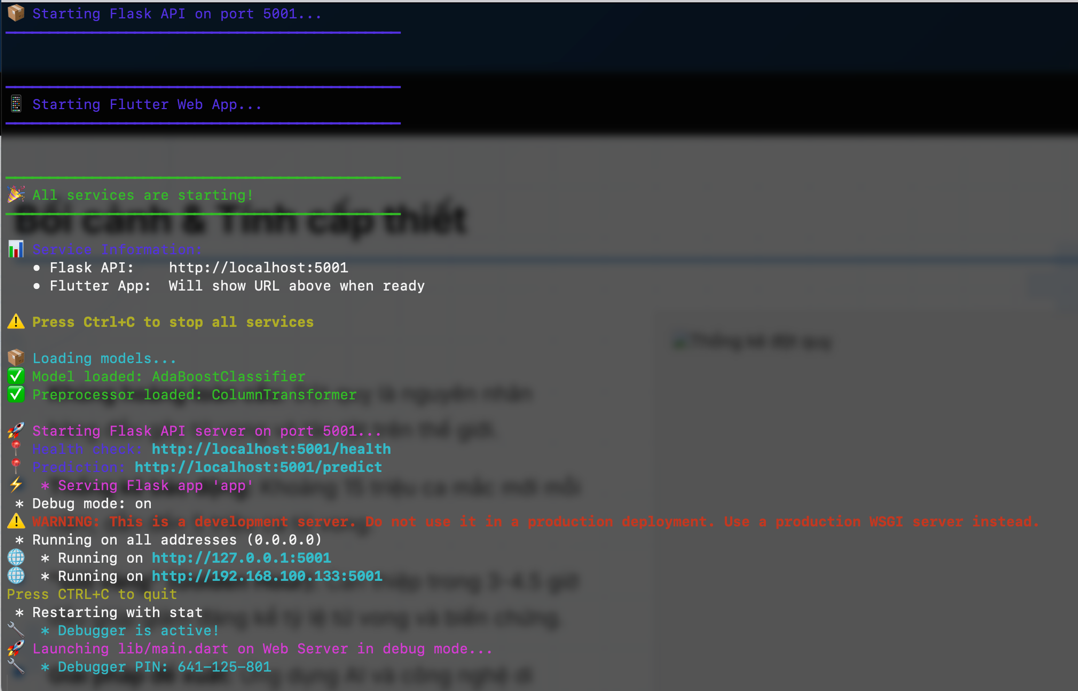
## 4.2. Hạ tầng Backend và Lưu trữ dữ liệu

Thay vì xây dựng máy chủ vật lý truyền thống, hệ thống sử dụng **Firebase** – nền tảng Backend-as-a-Service (BaaS) của Google. Các dịch vụ cốt lõi được triển khai bao gồm:

* **Firebase Authentication:** Quản lý xác thực người dùng thông qua Email/Password và Google Sign-In, hỗ trợ tự động làm mới JWT token và duy trì phiên đăng nhập an toàn.
* **Firebase Realtime Database:** Đóng vai trò là cơ sở dữ liệu chính, lưu trữ thông tin dưới dạng cây JSON với độ trễ thấp (<100ms) và khả năng đồng bộ thời gian thực (realtime synchronization). Dịch vụ này đặc biệt phù hợp cho các tính năng chat, cập nhật chỉ số sức khỏe và xử lý tín hiệu SOS.
* **Firebase Cloud Storage & Cloud Messaging (FCM):** Storage được dùng để lưu trữ media (avatar, ảnh y tế) , trong khi FCM đảm nhiệm việc gửi thông báo đẩy (push notifications) cho lịch hẹn, nhắc uống thuốc và cảnh báo khẩn cấp.

Toàn bộ dữ liệu được bảo vệ bởi **Firebase Security Rules**, đảm bảo người dùng chỉ truy cập được dữ liệu cá nhân, trong khi bác sĩ và admin có các quyền truy cập mở rộng tương ứng.

## 4.3. Mô hình Trí tuệ nhân tạo (AI) và API Service



Hệ thống tích hợp module AI dự đoán nguy cơ đột quỵ được xây dựng trên ngôn ngữ Python với framework **Flask** làm API Server. Mô hình học máy sử dụng thuật toán **Random Forest Classifier**, được huấn luyện dựa trên 14 đặc trưng đầu vào (như tuổi, BMI, đường huyết, tiền sử bệnh tim, v.v.). API cung cấp endpoint /predict để nhận dữ liệu sức khỏe từ ứng dụng Flutter và trả về kết quả phân loại nguy cơ (Thấp, Trung bình, Cao) cùng xác suất đột quỵ.

Để đảm bảo tính sẵn sàng cao, hệ thống trang bị thêm thuật toán **Fallback Rule-based**. Trong trường hợp model AI gặp sự cố không thể tải, hệ thống sẽ tự động chuyển sang tính điểm nguy cơ dựa trên các luật y khoa (ví dụ: tuổi >= 75 cộng 25 điểm, huyết áp >= 180/110 cộng 20 điểm). Ngoài ra, hệ thống còn tích hợp dịch vụ dự đoán tiểu đường Type 2 dựa trên thang điểm rủi ro (tuổi, BMI, huyết áp, tiền sử gia đình).

## 4.4. Các thư viện hỗ trợ và Công cụ phát triển

Để hoàn thiện các tính năng phức tạp, dự án sử dụng hệ sinh thái phong phú các thư viện mã nguồn mở. Đối với Flutter, các package nổi bật bao gồm fl\_chart (vẽ biểu đồ sức khỏe), geolocator (định vị GPS cho SOS), hive (lưu trữ cache offline), và shared\_preferences (lưu cấu hình cục bộ). Phía AI Service sử dụng các thư viện tính toán khoa học mạnh mẽ như scikit-learn, pandas, và numpy.

Quá trình phát triển được hỗ trợ bởi các công cụ tiêu chuẩn công nghiệp như Visual Studio Code, Git/GitHub cho quản lý mã nguồn, và Figma cho thiết kế giao diện UI/UX.

## 4.5. Tổng kết công nghệ

Để đảm bảo tính minh bạch về kỹ thuật và khả năng duy trì, phát triển của hệ thống, nhóm thực hiện tổng hợp lại các nền tảng và thư viện cốt lõi đã được áp dụng trong dự án.

Về phía ứng dụng di động (Frontend), dự án được xây dựng trên nền tảng **Flutter SDK phiên bản 3.5.4+** cùng ngôn ngữ **Dart 3.5.4+**, đảm bảo tận dụng được các tính năng mới nhất về hiệu năng và bảo mật. Kiến trúc ứng dụng được củng cố bởi **Provider 6.1.1** cho việc quản lý trạng thái và **Hive 2.2.3** cho khả năng lưu trữ đệm (cache) ngoại tuyến. Các tính năng trực quan hóa dữ liệu và định vị được hỗ trợ lần lượt bởi thư viện fl\_chart 0.69.0 và Geolocator 10.1.0.

Hạ tầng Backend hoàn toàn dựa trên hệ sinh thái **Firebase**, bao gồm Authentication 6.1.2 cho xác thực, Realtime Database 12.0.4 cho cơ sở dữ liệu thời gian thực, và Cloud Messaging 16.0.4 cho hệ thống thông báo.

Đối với phân hệ Trí tuệ nhân tạo (AI/ML), Server được triển khai bằng **Python Flask** kết hợp với các thư viện xử lý dữ liệu tiêu chuẩn như scikit-learn, pandas, và numpy để vận hành mô hình Random Forest.

Tổng cộng, dự án đã tích hợp và tối ưu hóa hơn **25 thư viện (packages) Flutter** và **6 thư viện Python**, tạo nên một hệ sinh thái công nghệ đồng bộ và chặt chẽ để giải quyết bài toán cảnh báo sớm đột quỵ và tiểu đường.

# PHẦN 5: CÀI ĐẶT VÀ TRIỂN KHAI

## 5.1. Yêu cầu hệ thống

Để đảm bảo quá trình phát triển diễn ra thuận lợi và ứng dụng vận hành ổn định phía người dùng cuối, nhóm thực hiện đã xác định các yêu cầu cấu hình phần cứng và phần mềm cụ thể.

Đối với môi trường phát triển (Development), máy tính cần được trang bị vi xử lý Intel Core i5 hoặc Apple M1/M2 trở lên, cùng với bộ nhớ RAM tối thiểu 8GB (khuyến nghị 16GB) để đáp ứng khả năng chạy máy ảo và biên dịch mã nguồn. Về phía thiết bị người dùng, ứng dụng được tối ưu hóa để hoạt động trên hệ điều hành Android phiên bản 6.0 (API 23) trở lên, yêu cầu phần cứng khiêm tốn với RAM 2GB và bộ nhớ trống 200MB.

Về mặt phần mềm, hệ thống yêu cầu cài đặt Flutter SDK phiên bản 3.5.4+ và Dart SDK 3.5.4+. Môi trường lập trình chính sử dụng Visual Studio Code kết hợp với Android Studio để quản lý máy ảo (Emulator). Ngoài ra, để vận hành module AI, hệ thống cần môi trường Python 3.8 trở lên.

## 5.2. Thiết lập môi trường phát triển

Quy trình cài đặt môi trường bắt đầu với việc thiết lập **Flutter SDK**. Sau khi tải về từ trang chủ và giải nén, đường dẫn bin của Flutter được thêm vào biến môi trường (PATH) của hệ điều hành (Windows/macOS/Linux). Việc cài đặt được xác nhận thành công khi lệnh flutter doctor trả về kết quả tích cực với tất cả các hạng mục đều được đánh dấu xanh.

Tiếp theo, **Android Studio** được cài đặt để cung cấp bộ công cụ Android SDK (Platform 33 - Android 13) và trình giả lập thiết bị (Virtual Device) như Pixel 6. Song song đó, **Visual Studio Code** được thiết lập làm trình soạn thảo code chính với các tiện ích mở rộng (Extensions) hỗ trợ cho Flutter và Dart. Đối với Backend AI, môi trường Python được cấu hình cùng các thư viện cần thiết thông qua pip để đảm bảo API hoạt động chính xác.

Sau khi hoàn tất cài đặt công cụ, mã nguồn dự án được tải về từ GitHub repository. Các thư viện phụ thuộc (dependencies) cho Flutter được cài đặt qua lệnh flutter pub get, và các thư viện Python cho AI được cài đặt qua tệp requirements.txt.

## 5.3. Cấu hình Hạ tầng Firebase

Hệ thống sử dụng Firebase làm nền tảng Backend chính, do đó việc cấu hình kết nối là bước quan trọng. Dự án trên Firebase Console được khởi tạo với tên warning-software-stroke.

Để kết nối ứng dụng Android, tệp cấu hình google-services.json (chứa thông tin định danh và chứng chỉ SHA-1) được đặt vào thư mục android/app/. Đối với phiên bản Web, các thông số như apiKey, authDomain, và projectId được cấu hình trực tiếp trong tệp lib/firebase\_options.dart.

Các dịch vụ thành phần của Firebase cũng được thiết lập chi tiết:

* **Authentication:** Kích hoạt phương thức đăng nhập bằng Email/Password và Google Sign-In.
* **Realtime Database:** Thiết lập cơ sở dữ liệu và áp dụng Security Rules để phân quyền chặt chẽ (ví dụ: người dùng chỉ đọc/ghi dữ liệu của chính mình, Admin có quyền quản trị).
* **Cloud Storage:** Cấu hình Rules để cho phép người dùng đã xác thực được phép tải lên và lưu trữ hình ảnh.
* **Cloud Messaging:** Sử dụng Server Key để tích hợp tính năng gửi thông báo đẩy.

## 5.4. Quy trình Biên dịch và Triển khai

Quá trình chạy thử nghiệm và đóng gói ứng dụng được chia thành hai giai đoạn:

Trong giai đoạn phát triển (Development), ứng dụng được chạy trực tiếp trên máy ảo hoặc thiết bị thật (qua kết nối USB Debugging) bằng lệnh flutter run. Tính năng Hot Reload được tận dụng tối đa để kiểm tra nhanh các thay đổi về giao diện. Module AI Server cũng được khởi chạy song song tại localhost:5000 để phục vụ các yêu cầu dự đoán.

Trong giai đoạn phát hành (Release), ứng dụng Android được đóng gói dưới dạng tệp APK (flutter build apk --release) hoặc App Bundle (flutter build appbundle) để sẵn sàng đưa lên Google Play Store. Quá trình này bao gồm việc tạo Keystore và cấu hình chữ ký số (signing key) trong tệp key.properties để đảm bảo tính xác thực của ứng dụng. Đối với phiên bản Web, lệnh flutter build web --release sẽ tạo ra thư mục tĩnh để có thể triển khai lên các dịch vụ hosting như Firebase Hosting.

## 5.5. Tổng kết cài đặt

Tính đến thời điểm hiện tại, toàn bộ môi trường phát triển bao gồm Flutter SDK, Android Studio, và Python Environment đã được thiết lập hoàn chỉnh. Mã nguồn dự án đã được đồng bộ và các dịch vụ Firebase đã được cấu hình và kết nối thành công. Hệ thống sẵn sàng cho việc tiếp tục phát triển tính năng hoặc đóng gói để chuyển giao cho người dùng cuối.

# PHẦN 6: HƯỚNG DẪN SỬ DỤNG

## 6.1. Phân hệ dành cho Người dùng cuối (User)

Quy trình sử dụng ứng dụng bắt đầu từ bước **Đăng ký và Xác thực**. Người dùng mới có thể tạo tài khoản thông qua Email hoặc sử dụng tính năng đăng nhập nhanh bằng tài khoản Google (Google Sign-In) để tiết kiệm thời gian. Hệ thống cũng hỗ trợ quy trình khôi phục mật khẩu tự động qua email trong trường hợp người dùng quên thông tin đăng nhập.

Sau khi đăng nhập thành công, người dùng sẽ tiếp cận **Màn hình chính (Dashboard)**. Đây là trung tâm điều khiển hiển thị tổng quan các thông báo quan trọng, kết quả dự đoán gần nhất và các phím tắt (Quick Actions) dẫn đến các chức năng cốt lõi như Dự đoán, Đặt lịch hẹn hoặc Kích hoạt SOS.

Chức năng cốt lõi của hệ thống là **Dự đoán nguy cơ bệnh**, được thực hiện tại **Prediction Hub**:

* **Dự đoán Đột quỵ:** Người dùng nhập 12 chỉ số sức khỏe (như tuổi, BMI, huyết áp, glucose, tiền sử bệnh...). Hệ thống sẽ trả về điểm nguy cơ (0-100), mức độ cảnh báo (Thấp/Trung/Cao) và các khuyến nghị y tế tương ứng.
* **Dự đoán Tiểu đường:** Tương tự, người dùng cung cấp các thông tin về đường huyết lúc đói, tiền sử gia đình và mức độ hoạt động để nhận kết quả đánh giá rủi ro.

Bên cạnh việc dự đoán, ứng dụng đóng vai trò là một sổ tay y tế điện tử thông qua **Health Hub**. Tại đây, người dùng có thể nhập liệu hàng ngày các chỉ số sinh tồn (huyết áp, nhịp tim, cân nặng) và theo dõi sự biến thiên của sức khỏe thông qua các biểu đồ trực quan theo chu kỳ 7, 30 hoặc 90 ngày. Hệ thống cũng hỗ trợ **Quản lý thành viên gia đình**, cho phép người dùng tạo nhóm, mời người thân và theo dõi sức khỏe cho cha mẹ hoặc con cái ngay trên cùng một giao diện.

Để kết nối với chuyên gia, người dùng truy cập **Doctors Hub** để xem hồ sơ bác sĩ và thực hiện **Đặt lịch hẹn**. Quy trình đặt lịch được thiết kế khép kín từ việc chọn khung giờ (slot), nhập lý do khám đến khi nhận thông báo xác nhận. Ngoài ra, tính năng **Chat trực tuyến** cho phép gửi tin nhắn và hình ảnh để nhận tư vấn sơ bộ.

Đặc biệt, trong tình huống nguy cấp, nút **SOS khẩn cấp** (luôn hiển thị trên màn hình) cho phép người dùng kích hoạt quy trình báo động đỏ. Hệ thống sẽ tự động định vị GPS và gửi tín hiệu cầu cứu đến các bác sĩ trực. Ngoài ra, tính năng **Nhắc nhở dùng thuốc** giúp bệnh nhân tuân thủ phác đồ điều trị thông qua việc thiết lập lịch thông báo định kỳ.

## 6.2. Phân hệ dành cho Bác sĩ (Doctor)

Giao diện dành cho đội ngũ y tế tập trung vào hiệu suất làm việc và khả năng phản ứng nhanh. Sau khi đăng nhập bằng tài khoản chuyên dụng , Bác sĩ sẽ tiếp cận **Dashboard quản lý** hiển thị số liệu thời gian thực về các ca SOS chờ xử lý, lịch hẹn trong ngày và danh sách bệnh nhân mới.

Quy trình làm việc của bác sĩ bao gồm ba mảng chính:

* **Quản lý Bệnh nhân (Patient Management):** Bác sĩ có thể tra cứu danh sách bệnh nhân, xem chi tiết hồ sơ sức khỏe, lịch sử dự đoán và thực hiện kê đơn thuốc điện tử ngay trên ứng dụng.
* **Quản lý Lịch hẹn (Appointment System):** Hệ thống cho phép bác sĩ duyệt (xác nhận) hoặc từ chối các yêu cầu đặt lịch từ bệnh nhân. Sau khi thăm khám, bác sĩ cập nhật trạng thái lịch hẹn thành "Hoàn thành" kèm theo ghi chú chuyên môn. Để thuận tiện, bác sĩ có thể chủ động thiết lập **Lịch làm việc** bằng cách tạo các slot khám (thời gian bắt đầu/kết thúc, số lượng bệnh nhân tối đa).
* **Xử lý Khẩn cấp (SOS Response):** Tại màn hình **SOS Queue**, các ca cấp cứu được sắp xếp theo độ ưu tiên. Bác sĩ nhận ca sẽ thấy vị trí bệnh nhân trên bản đồ và có thể liên hệ tức thì để hướng dẫn sơ cứu.

## 6.3. Phân hệ Quản trị viên (Admin)

Phân hệ quản trị được xây dựng trên nền tảng Web, cung cấp cái nhìn toàn cảnh về hệ thống. Dashboard của Admin hiển thị các biểu đồ thống kê về lượng người dùng, tần suất sử dụng dịch vụ và tình hình hoạt động của hệ thống.

Vai trò chính của Admin bao gồm:

* **Quản trị Người dùng và Bác sĩ:** Admin có quyền xem danh sách, chỉnh sửa thông tin hoặc khóa tài khoản vi phạm. Đặc biệt đối với tài khoản Bác sĩ, Admin chịu trách nhiệm kiểm duyệt hồ sơ, chứng chỉ hành nghề trước khi cấp quyền hoạt động trên hệ thống.
* **Quản trị Nội dung:** Admin quản lý kho tri thức y khoa (**Knowledge Base**) bằng cách soạn thảo, xuất bản hoặc chỉnh sửa các bài viết chuyên môn. Đồng thời, Admin cũng đóng vai trò kiểm duyệt viên cho **Cộng đồng**, đảm bảo môi trường thảo luận lành mạnh bằng cách xóa các bài đăng không phù hợp.
* **Giám sát SOS:** Admin có thể theo dõi toàn bộ tiến trình xử lý các ca SOS để đảm bảo không có trường hợp khẩn cấp nào bị bỏ sót.

# PHẦN 7: KẾT LUẬN

## 7.1. Các kết quả đạt được

Sau quá trình nghiên cứu và triển khai thực hiện đề tài, nhóm đã hoàn thành việc xây dựng hệ thống SEWS (Stroke Early Warning System) với các kết quả cụ thể như sau:

* **Về mặt chức năng:** Nhóm đã bước đầu tích hợp thành công mô hình trí tuệ nhân tạo (AI) sử dụng thuật toán Random Forest với 14 đặc trưng đầu vào để dự đoán nguy cơ đột quỵ. Hệ thống API được xây dựng bằng Flask để xử lý các yêu cầu dự đoán từ ứng dụng di động , đồng thời trang bị sẵn thuật toán dự phòng (rule-based) hoạt động khi mô hình AI gặp sự cố kết nối. Đối với bệnh tiểu đường, ứng dụng cung cấp các đánh giá rủi ro dựa trên 6 yếu tố nguy cơ cơ bản. Các chức năng nghiệp vụ y tế thiết yếu như theo dõi biểu đồ sức khỏe , đặt lịch hẹn , và tính năng SOS khẩn cấp gửi vị trí GPS tự động đã được hình thành và đưa vào hoạt động.
* **Về mặt kỹ thuật:** Ứng dụng được phát triển trên nền tảng Flutter phiên bản 3.5.4+ cho phép vận hành trên cả Android và Web. Kiến trúc hệ thống được chia thành 5 lớp rõ ràng (UI, Business Logic, Data Layer...), giúp mã nguồn dễ đọc và dễ bảo trì. Việc sử dụng Firebase làm Backend đã hỗ trợ tốt cho các tác vụ thời gian thực như Chat và Thông báo , trong khi thư viện Hive đảm bảo khả năng truy cập dữ liệu cơ bản khi thiết bị mất kết nối mạng.
* **Về mặt sản phẩm:** Đồ án đã đóng gói được tệp cài đặt APK (67.3MB) với giao diện tiếng Việt hoàn chỉnh theo chuẩn Material Design 3. Hệ thống phân quyền đầy đủ cho 3 vai trò: Người dùng, Bác sĩ và Quản trị viên.

## 7.2. Những hạn chế và Tồn tại

Bên cạnh những kết quả bước đầu, nhóm thực hiện xin nghiêm túc nhìn nhận những hạn chế, thiếu sót và các vấn đề kỹ thuật còn tồn tại trong phiên bản hiện nay:

* **Về độ ổn định và lỗi kỹ thuật:** Do giới hạn về thời gian thực hiện và kinh nghiệm lập trình, ứng dụng hiện tại vẫn đang ở giai đoạn kiểm thử (Beta). Nhóm thừa nhận hệ thống vẫn còn tồn tại các lỗi (bugs) trong quá trình vận hành thực tế, đôi khi gây ra hiện tượng giật lag hoặc thoát ứng dụng đột ngột. Một số tính năng xử lý thời gian thực chưa hoạt động mượt mà trong điều kiện mạng không ổn định. Hệ thống cũng chưa có cơ chế giám sát lỗi (monitoring) và quy trình kiểm thử tự động (CI/CD) cho môi trường production.
* **Về tính năng chưa hoàn thiện:**
  + **Kết nối y tế:** Tính năng gọi video (Video Call) để tư vấn từ xa chưa được triển khai. Cổng thanh toán trực tuyến cũng chưa được tích hợp, khiến quy trình đặt lịch khám chưa thể khép kín hoàn toàn.
  + **Dữ liệu đầu vào:** Việc nhập liệu các chỉ số sức khỏe hiện vẫn phải thực hiện thủ công, chưa có cơ chế tự động đồng bộ từ các thiết bị đeo thông minh (smartwatch) hoặc nhập nhanh từ file hồ sơ bệnh án có sẵn.
  + **Mô hình dự đoán:** Dự đoán tiểu đường vẫn đang sử dụng thuật toán dựa trên luật (rule-based) thay vì Machine Learning. Mô hình AI hiện tại cũng chưa có khả năng tự học (continuous learning) từ dữ liệu mới người dùng nhập vào.
* **Về hiệu suất mô hình AI:** Kết quả thực nghiệm cho thấy mô hình Random Forest đạt độ chính xác tổng thể tốt khi đánh giá trên tập dữ liệu kiểm thử lớn (batch testing với số lượng > 100 mẫu). Tuy nhiên, **độ ổn định khi dự đoán cho từng cá nhân riêng lẻ (1:1 inference) trong điều kiện thực tế còn hạn chế**. Kết quả dự đoán đơn lẻ có thể bị dao động hoặc thiếu chính xác khi dữ liệu đầu vào của người dùng rơi vào các vùng biên (edge cases) hoặc chưa được đại diện đầy đủ trong tập dữ liệu huấn luyện ban đầu.
* **Về triển khai và bảo mật:** Ứng dụng chưa hỗ trợ nền tảng iOS do thiếu tài khoản nhà phát triển Apple. Các tính năng bảo mật nâng cao như xác thực hai yếu tố (2FA) và mã hóa đầu cuối cho tin nhắn chưa được áp dụng. API AI hiện tại vẫn chạy cục bộ hoặc trên môi trường test, chưa được triển khai lên hạ tầng Cloud Production chuyên nghiệp.

## 7.3. Hướng phát triển

Dựa trên những hạn chế đã phân tích, nhóm đề xuất lộ trình phát triển và khắc phục như sau:

* **Giai đoạn ngắn hạn (3-6 tháng):** Tập trung sửa lỗi (fix bugs) để ổn định hóa hệ thống hiện tại. Thu thập thêm dữ liệu để huấn luyện lại mô hình AI nhằm nâng cao độ chính xác , đồng thời nghiên cứu áp dụng Machine Learning cho bài toán dự đoán tiểu đường. Nhóm cũng sẽ tiến hành triển khai API AI lên các nền tảng Cloud như AWS hoặc Google Cloud.
* **Giai đoạn trung hạn (6-12 tháng):** Tích hợp kết nối IoT để tự động lấy dữ liệu từ các thiết bị đo huyết áp Bluetooth và đồng bộ với Google Fit/Apple Health. Phát triển thêm tính năng Chatbot AI để hỗ trợ tư vấn sơ bộ cho bệnh nhân và mở rộng hệ thống kết nối với các phần mềm quản lý bệnh viện (HIS).
* **Giai đoạn dài hạn (1-2 năm):** Nghiên cứu các công nghệ AI nâng cao như thị giác máy tính (Computer Vision) để phân tích hình ảnh y tế và xử lý ngôn ngữ tự nhiên (NLP) để phân tích triệu chứng bệnh. Mục tiêu cuối cùng là hợp tác với các đơn vị y tế để đưa sản phẩm vào ứng dụng thực tiễn.

## 7.4. Kết luận chung

Tóm lại, mặc dù vẫn còn những khiếm khuyết về mặt kỹ thuật và tính năng cần được tiếp tục hoàn thiện, dự án **SEWS** đã chứng minh được tính khả thi của việc ứng dụng công nghệ di động và trí tuệ nhân tạo trong hỗ trợ chăm sóc sức khỏe cộng đồng.

Sản phẩm đã tạo ra một công cụ hỗ trợ người dùng chủ động theo dõi sức khỏe, phát hiện sớm các nguy cơ đột quỵ và cung cấp giải pháp hỗ trợ khẩn cấp trong "thời gian vàng". Đây là tiền đề quan trọng để nhóm tiếp tục nghiên cứu và phát triển sản phẩm hoàn thiện hơn trong tương lai.

Cuối cùng, nhóm xin gửi lời cảm ơn chân thành đến thầy/cô hướng dẫn và các bạn đã hỗ trợ nhóm hoàn thành đồ án này.

# PHẦN PHỤ LỤC

## Phụ lục A: Đặc tả kỹ thuật API và Cấu trúc dữ liệu

Để đảm bảo tính nhất quán trong giao tiếp giữa Client (ứng dụng di động/web) và Server, hệ thống tuân thủ các chuẩn giao tiếp dữ liệu cụ thể:

* **Flask AI API:** Đóng vai trò là "bộ não" xử lý dự đoán, API này hoạt động tại cổng 5000 (môi trường phát triển). API cung cấp hai endpoint chính: endpoint kiểm tra trạng thái hệ thống (/health) để đảm bảo các model và bộ xử lý dữ liệu đã được tải thành công; và endpoint dự đoán (/predict) tiếp nhận 12 chỉ số sức khỏe (tuổi, huyết áp, glucose, BMI...) thông qua phương thức POST để trả về kết quả phân tích nguy cơ đột quỵ cùng các khuyến nghị y tế.
* **Firebase Realtime Database:** Đóng vai trò lưu trữ trạng thái ứng dụng thời gian thực. Cấu trúc dữ liệu được tổ chức thành các nhánh (paths) riêng biệt, bao gồm: quản lý người dùng và bác sĩ (/users, /doctors), hệ thống đặt lịch hẹn (/appointments), hồ sơ sức khỏe (/healthRecords), và đặc biệt là hệ thống tin nhắn (/messages) cùng hàng đợi cấp cứu (/sos) để phục vụ tính năng phản hồi tức thì.

## Phụ lục B: Tổ chức mã nguồn dự án

Mã nguồn dự án early\_warning\_software\_for\_stroke được tổ chức theo kiến trúc phân tầng (Layered Architecture) để tối ưu hóa việc bảo trì và mở rộng:

* **Thư mục gốc:** Chứa các cấu hình môi trường, mã nguồn Native Android, và thư mục assets – nơi lưu trữ các mô hình AI (.pkl), dữ liệu huấn luyện và tài nguyên tĩnh.
* **Thư mục lib (Flutter Code):** Đây là nơi chứa mã nguồn chính của ứng dụng, được chia nhỏ thành:
  + data/: Chứa 14 mô hình dữ liệu (models) và các providers quản lý trạng thái.
  + features/: Chứa mã nguồn giao diện (UI) cho 91 màn hình, được phân chia rõ ràng theo từng phân hệ chức năng như: xác thực (auth), bảng điều khiển (dashboard), dự đoán (prediction), và quản lý gia đình.
  + services/: Bao gồm 34 lớp dịch vụ xử lý logic nghiệp vụ tách biệt như StrokePredictionService, SosService, ChatService, đảm bảo nguyên tắc đơn trách nhiệm (Single Responsibility).
* **Tài liệu và Kiểm thử:** Thư mục docs lưu trữ toàn bộ báo cáo và biểu đồ thiết kế hệ thống (UML), trong khi thư mục test chứa các kịch bản kiểm thử đơn vị (unit tests) cho các dịch vụ cốt lõi.

## Phụ lục C: Tài liệu tham khảo

Quá trình thực hiện đồ án dựa trên sự nghiên cứu kỹ lưỡng từ ba nguồn tài liệu chính:

1. **Tài liệu kỹ thuật:** Các hướng dẫn chính thống (Official Documentation) từ Flutter, Firebase, Flask và thư viện scikit-learn để xây dựng nền tảng công nghệ.
2. **Tài liệu y khoa:** Các nghiên cứu và hướng dẫn từ Tổ chức Y tế Thế giới (WHO) và Hiệp hội Đột quỵ Hoa Kỳ, đặc biệt là các nghiên cứu về yếu tố nguy cơ đột quỵ của Goldstein và cộng sự, nhằm đảm bảo tính chính xác của các tham số đầu vào cho mô hình AI.
3. **Công cụ thiết kế:** Hướng dẫn thiết kế giao diện Material Design 3 và các công cụ mô hình hóa hệ thống như PlantUML.

## Phụ lục D: Báo cáo việc ứng dụng Trí tuệ nhân tạo (AI) trong dự án

Dự án SEWS có quy mô lớn với hơn 90 màn hình giao diện và tích hợp đa nền tảng công nghệ, đặt ra thách thức lớn về khối lượng công việc so với thời gian quy định. Do đó, nhóm thực hiện đã áp dụng AI một cách có kiểm soát ở hai cấp độ:

**1. AI trong sản phẩm (Product AI):** Đây là lõi công nghệ của dự án, bao gồm mô hình **Random Forest Classifier** được huấn luyện trên tập dữ liệu y tế để dự đoán xác suất đột quỵ. Quy trình phát triển bao gồm thu thập dữ liệu, tiền xử lý, trích xuất đặc trưng (feature engineering) và đánh giá mô hình bằng kỹ thuật cross-validation. Bên cạnh đó, thuật toán dựa trên luật (Rule-based) được phát triển song song để dự đoán nguy cơ tiểu đường dựa trên các khuyến cáo y khoa.

**2. AI hỗ trợ phát triển (Development AI):** Nhóm sử dụng các công cụ AI (như GitHub Copilot, Claude AI, ChatGPT) để hỗ trợ các tác vụ lặp lại (boilerplate code), tạo dữ liệu mẫu, viết tài liệu và tối ưu hóa mã nguồn. Cụ thể, AI giúp giảm 30-40% thời gian viết các đoạn code cơ bản, cho phép lập trình viên tập trung vào xử lý các logic nghiệp vụ phức tạp và tích hợp hệ thống.

**Cam kết về tính toàn vẹn:** Nhóm thực hiện tuân thủ nguyên tắc "AI là công cụ hỗ trợ, không thay thế con người". Mọi mã nguồn và nội dung do AI đề xuất đều trải qua quy trình kiểm duyệt (review), tinh chỉnh và kiểm thử nghiêm ngặt bởi lập trình viên. Nhóm chịu trách nhiệm hoàn toàn về thiết kế kiến trúc, logic nghiệp vụ, bảo mật và tính chính xác cuối cùng của sản phẩm.

# (HẾT)