**TRƯỜNG ĐẠI HỌC TRÀ VINH**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

****

**ISO 9001:2015**

**TRẦN TRỌNG NHÂN**

**PHÁT TRIỂN ỨNG DỤNG ĐĂNG KÝ**

**LỊCH KHÁM BỆNH CHO PHÒNG KHÁM**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**TRÀ VINH, NĂM 2024**

**TRƯỜNG ĐẠI HỌC TRÀ VINH**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

**PHÁT TRIỂN ỨNG DỤNG ĐĂNG KÝ**

**LỊCH KHÁM BỆNH CHO PHÒNG KHÁM**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

Sinh viên: **Trần Trọng Nhân**

Lớp: **DA20TTA**

MSSV: **110120051**

GVHD: **ThS. Phạm Minh Đương**

**TRÀ VINH, NĂM 2024**

**LỜI MỞ ĐẦU**

Phát triển ứng dụng di động để đặt lịch khám bệnh cho phòng khám là một bước tiến quan trọng trong việc hiện đại hóa dịch vụ y tế, đáp ứng nhu cầu ngày càng cao của bệnh nhân trong việc tiếp cận dịch vụ chăm sóc sức khỏe. Ứng dụng này sẽ góp phần tối ưu hóa quy trình quản lý và sắp xếp lịch hẹn, giảm thiểu thời gian chờ đợi của bệnh nhân . Bằng cách tích hợp công nghệ tiên tiến, ứng dụng sẽ mang lại trải nghiệm thuận tiện và dễ dàng cho người sử dụng, từ đó góp phần nâng cao hiệu quả và chất lượng dịch vụ và sự hài lòng của bệnh nhân.

Hiện nay, công nghệ thông tin ngày càng phát triển, việc triển khai ứng dụng di động này là một minh chứng cho sự đổi mới và bắt kịp xu hướng thời đại của ngành y tế. Ứng dụng cho phép bệnh nhân dễ dàng đặt lịch hẹn và theo dõi quá trình điều trị mọi lúc, mọi nơi, chỉ với vài thao tác đơn giản trên điện thoại thông minh. Không chỉ giúp bệnh nhân tiết kiệm thời gian, ứng dụng còn cung cấp các chức năng như nhắc nhở lịch hẹn, lưu trữ hồ sơ bệnh án điện tử, giúp người bệnh chủ động hơn trong việc chăm sóc sức khỏe của mình.

Đối với các phòng khám, ứng dụng di động này hỗ trợ việc quản lý thông tin bệnh nhân, tối ưu hóa nguồn lực và nâng cao chất lượng dịch vụ chăm sóc y tế. Các phòng khám có thể dễ dàng theo dõi lịch hẹn, quản lý hồ sơ bệnh án và tương tác với bệnh nhân một cách hiệu quả hơn. Hơn nữa, ứng dụng còn cho phép thu thập dữ liệu và phân tích thông tin, giúp các bác sĩ và nhân viên y tế đưa ra các quyết định chính xác và kịp thời.

Với những lợi ích vượt trội mà ứng dụng di động đặt lịch khám bệnh mang lại, việc phát triển và triển khai ứng dụng này không chỉ là một xu hướng mà còn là một yêu cầu cấp thiết để cải thiện hệ thống y tế hiện nay. Chính vì vậy, việc nghiên cứu và phát triển ứng dụng này hứa hẹn sẽ tạo ra một bước ngoặt quan trọng, mang lại nhiều lợi ích cho cả bệnh nhân và các cơ sở y tế, đồng thời nâng cao chất lượng dịch vụ chăm sóc sức khỏe trong tương lai.

**LỜI CẢM ƠN**

Tôi xin chân thành gửi lời cảm ơn đến quý thầy cô Trường Đại học Trà Vinh nói chung và các thầy cô Bộ môn trong khoa Kỹ thuật và Công nghệ nói riêng đã tạo điều kiện cho tôi có cơ hội thực hành, tiếp xúc để tôi có thể tránh được những vướng mắc và bỡ ngỡ trong môi trường công việc thời gian tới.

Tôi xin chân thành cảm ơn thầy Phạm Minh Đương. Nhờ sự giúp đỡ tận tình và những chỉ bảo của Thầy từ lúc bắt đầu cho tới lúc kết thúc đồ án mà tôi đã hoàn thành đúng thời hạn quy định và tích lũy được cho mình một lượng nền tảng kiến thức quý báu.

Mặc dù đã cố gắng hoàn thành đề tài tốt nhất nhưng do thời gian và kiến thức chuyên ngành còn hạn chế nên tôi vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá, và trình bày về đề tài. Rất mong nhận được sự quan tâm và góp ý của các thầy, cô giảng viên bộ môn để đề tài của tôi được hoàn chỉnh và đầy đủ hơn.

Tôi xin chân thành cảm ơn.

*Trà Vinh, ngày 15 tháng 7 năm 2024*

Sinh viên thực hiện

**Trần Trọng Nhân**

**NHẬN XÉT**

**(Của giảng viên hướng dẫn trong đồ án, khoá luận của sinh viên)**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Giảng viên hướng dẫn**

(ký và ghi rõ họ tên)

**Phạm Minh Đương**

UBND TỈNH TRÀ VINH **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TRÀ VINH Độc lập – Tự do – Hạnh Phúc**

**BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP**

*(Của giảng viên hướng dẫn)*

Họ và tên sinh viên: Trần Trọng Nhân MSSV: 110120051

Ngành: Công nghệ thông tin Khóa: 2020

Tên đề tài: Phát triển ứng dụng đăng ký lịch khám bệnh cho phòng khám

Họ và tên Giáo viên hướng dẫn: Phạm Minh Đương

Chức danh: Giảng viên Học vị: Thạc sĩ

**NHẬN XÉT**

1. Nội dung đề tài:

1. Ưu điểm:

1. Khuyết điểm:

1. Điểm mới đề tài:

1. Giá trị thực trên đề tài:

7. Đề nghị sửa chữa bổ sung:

8. Đánh giá:

Trà Vinh*, ngày 01 tháng 8 năm 2024*

Giảng viên hướng dẫn

*(Ký & ghi rõ họ tên)*

**Phạm Minh Đương**

**MỤC LỤC**

[CHƯƠNG 1. ĐẶT VẤN ĐỀ 1](#_Toc11353)

[1.1. Lý do chọn đề tài 1](#_Toc8386)

[1.2. Mục tiêu 1](#_Toc28168)

[1.3. Nội dung 2](#_Toc9763)

[1.4. Đối tượng và phạm vi nghiên cứu 2](#_Toc32700)

[1.5. Phương pháp nghiên cứu 2](#_Toc2084)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 4](#_Toc7814)

[2.1. Công nghệ, ngôn ngữ, công cụ thực hiện 4](#_Toc16131)

*[2.1.1. Flutter Framework](#_Toc6996)* [4](#_Toc6996)

*[2.1.2. Firebase](#_Toc21237)* [8](#_Toc21237)

*[2.1.3. Ngôn ngữ Dart](#_Toc29826)* [9](#_Toc29826)

*[2.1.4. Mô hình MVC](#_Toc27648)* [11](#_Toc27648)

*[2.1.5. Visual Studio Code](#_Toc4296)* [12](#_Toc4296)

[2.2. Nghiệp vụ liên quan đến đề tài 13](#_Toc24983)

*[2.2.1. Thanh toán điện tử](#_Toc20202)* [13](#_Toc20202)

[2.3. Các công trình nghiên cứu liên quan 13](#_Toc20515)

[CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU 15](#_Toc30609)

[3.1. Mô tả bài toán 15](#_Toc11238)

*[3.1.1. Yêu cầu hệ thống](#_Toc19251)* [15](#_Toc19251)

*[3.1.2. Quy trình làm việc](#_Toc17368)* [16](#_Toc17368)

*[3.1.3. Yêu cầu chức năng](#_Toc29687)* [17](#_Toc29687)

[3.2. Thiết kế dữ liệu 18](#_Toc31844)

*[3.2.1. Cấu trúc dữ liệu dưới dạng JSON](#_Toc9202)* [18](#_Toc9202)

*[3.2.2. Dữ liệu mẫu](#_Toc2362)* [20](#_Toc2362)

*[3.2.3. Danh sách các bảng dữ liệu](#_Toc10812)* [22](#_Toc10812)

[3.3. Thiết kế xử lý 24](#_Toc26260)

*[3.3.1. Sơ đồ use case Người dùng](#_Toc23974)* [24](#_Toc23974)

*[3.3.2. Sơ đồ use case Bác sĩ](#_Toc5521)* [25](#_Toc5521)

*[3.3.3. Sơ đồ use case Người quản lý quầy tiếp nhận](#_Toc29207)* [26](#_Toc29207)

[CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU 27](#_Toc31907)

[4.1. Giao diện người dùng và chức năng 27](#_Toc13784)

[4.2. Giao diện bác sĩ và chức năng 52](#_Toc5145)

[4.3. Giao diện quầy tiếp nhận và chức năng 53](#_Toc24164)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 57](#_Toc2125)

[5.1. Kết luận 57](#_Toc2151)

[5.2. Hướng phát triển 57](#_Toc1542)

[DANH MỤC TÀI LIỆU THAM KHẢO 59](#_Toc27929)

**DANH MỤC CÁC BẢNG**

[Bảng 2.1 Bảng giải thích cấu trúc thư mục quan trọng của Flutter Framework 6](#_Toc7965)

[Bảng 3.1 Bảng hospital (Phòng khám) 22](#_Toc29564)

[Bảng 3.2 Bảng medical (Hồ sơ) 22](#_Toc8069)

[Bảng 3.3 Bảng payments (Thanh toán) 23](#_Toc6803)

[Bảng 3.4 Bảng users (Tài khoản) 23](#_Toc2817)

[Bảng 3.5 Bảng appointments (Đơn đặt lịch khám) 23](#_Toc32732)

**DANH MỤC CÁC HÌNH ẢNH**

[Hình 2.1 Cấu trúc thư mục của Flutter 5](#_Toc1303)

[Hình 2.2 Mô hình MVC 12](#_Toc20031)

[Hình 3.1 Sơ đồ Use case Người dùng 24](#_Toc21516)

[Hình 3.2 Sơ đồ Use case Bác sĩ 25](#_Toc25409)

[Hình 3.3 Sơ đồ Use case Người quản lý quầy tiếp nhận 26](#_Toc20404)

[Hình 4.1 Giao diện khi bắt đầu ứng dụng 27](#_Toc21626)

[Hình 4.2 Giao diện đăng ký tài khoản 28](#_Toc12936)

[Hình 4.3 Giao diện đăng nhập 29](#_Toc26953)

[Hình 4.4 Giao diện trang chủ 30](#_Toc249)

[Hình 4.5 Giao diện giới thiệu ứng dụng 31](#_Toc23750)

[Hình 4.6 Giao diện hướng dẫn đăng ký đặt lịch khám 32](#_Toc31492)

[Hình 4.7 Giao diện xem danh sách các phòng khám và chọn phòng khám 33](#_Toc31736)

[Hình 4.8 Giao diện xem hồ sơ đăng ký bệnh nhân 34](#_Toc19463)

[Hình 4.9 Giao diện tạo mới hồ sơ bệnh nhân 35](#_Toc29769)

[Hình 4.10 Giao diện xem chi tiết hồ sơ bệnh nhân 36](#_Toc27859)

[Hình 4.11 Giao diện chỉnh sửa thông tin hồ sơ bệnh nhân 37](#_Toc26166)

[Hình 4.12 Giao diện xóa hồ sơ bệnh nhân 38](#_Toc27650)

[Hình 4.13 Giao diện chọn ngày khám 39](#_Toc4171)

[Hình 4.14 Giao diện chọn khung giờ khám 40](#_Toc4235)

[Hình 4.15 Giao diện chọn hồ sơ khám 41](#_Toc9807)

[Hình 4.16 Giao diện kiểm tra và xác nhận thông tin khi bệnh nhân có BHYT 42](#_Toc2707)

[Hình 4.17 Giao diện kiểm tra và xác nhận thông tin khi bệnh nhân không có BHYT 43](#_Toc10428)

[Hình 4.18 Giao diện chọn phương thức thanh toán VNpay 44](#_Toc9862)

[Hình 4.19 Giao diện thanh toán VNpay bằng thẻ ngân hàng 45](#_Toc20594)

[Hình 4.20 Giao diện xác thực mã OTP 46](#_Toc13927)

[Hình 4.21 Giao diện xác nhận lại thông tin 47](#_Toc31981)

[Hình 4.22 Giao diện xem các cuộc hẹn sắp diễn ra 48](#_Toc16571)

[Hình 4.23 Giao diện xem chi tiết cuộc hẹn 49](#_Toc22092)

[Hình 4.24 Giao diện xem các cuộc hẹn đã hủy 50](#_Toc13543)

[Hình 4.25 Giao diện xem chi tiết cuộc hẹn đã hủy 51](#_Toc19269)

[Hình 4.26 Giao diện xem lịch đặt khám 52](#_Toc17741)

[Hình 4.27 Giao diện xem và xác nhận các cuộc hẹn của quầy tiếp nhận 53](#_Toc20644)

[Hình 4.28 Giao diện chọn phòng khám để tra cứu của quầy tiếp nhận 54](#_Toc24249)

[Hình 4.29 Giao diện xem thống kê phòng khám và trạng thái đơn đăng ký để thống kê của quầy tiếp nhận 55](#_Toc6198)

[Hình 4.30 Giao diện tìm kiếm mã bệnh nhân 56](#_Toc5377)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| API | Application Programming Interface |
| ARM | Advanced RISC Machine |
| BaaS | Backend as a Service |
| BHYT | Bảo hiểm y tế |
| CCCD/CMND | Căn cước công dân/ chứng minh nhân dân |
| Dependencies | Phụ thuộc |
| Edit the system environment variables | Cài đặt biến môi trường |
| iOS | iPhone Operating System |
| JSON | JavaScript Object Notation |
| MVC | Model-View-Controller |
| NoSQL | No Structured Query Language |
| PIN | Personal Identification Number |
| OTP | One Time Password |
| SDK | Software Development Kit |
| SSL | Secure Sockets Layer |
| UI | User Interface |
| XML | eXtensible Markup Language |

thời gian trung bình

# ĐẶT VẤN ĐỀ

## Lý do chọn đề tài

Việc chọn đề tài “Phát triển ứng dụng đăng ký lịch khám bệnh cho phòng khám” là một quyết định có tính thực tiễn. Trong thời đại công nghệ phát triển như hiện nay, việc tận dụng công nghệ thông tin để cải thiện dịch vụ y tế là điều hết sức cần thiết và hữu ích.

Trước hết, ứng dụng này giúp tối ưu hóa quá trình đăng ký lịch khám bệnh, từ việc tiết kiệm thời gian của bệnh nhân đến việc quản lý lịch trực tiếp của phòng khám. Thay vì phải đến trực tiếp phòng khám để đăng ký lịch khám, bệnh nhân có thể dễ dàng thực hiện việc này thông qua ứng dụng trên điện thoại di động của phòng khám. Điều này giúp giảm thiểu sự chật vật và thời gian chờ đợi của bệnh nhân, đồng thời giúp phòng khám quản lý lịch hẹn một cách hiệu quả hơn.

Bên cạnh đó, ứng dụng đăng ký lịch khám bệnh còn cung cấp tính năng nhắc nhở và thông báo cho bệnh nhân về lịch hẹn của mình, giúp họ không bỏ lỡ các cuộc hẹn quan trọng. Điều này đồng nghĩa với việc giảm thiểu số lượng bệnh nhân không đến đúng giờ, giúp phòng khám tối ưu hoá thời gian làm việc và tăng hiệu suất phục vụ.

Tóm lại, việc phát triển ứng dụng đăng ký lịch khám bệnh không chỉ giúp cải thiện trải nghiệm của bệnh nhân mà còn là một bước tiến quan trọng trong việc hiện đại hóa và nâng cao chất lượng dịch vụ y tế.

## Mục tiêu

Xây dựng được tương đối một ứng dụng di động bao gồm các chức năng chính bao gồm: chức năng đăng ký lịch cho bệnh nhân, tạo hồ sơ đăng ký khám, quản lý lịch khám cho bệnh nhân. Chức năng xác nhận, xem thống kê và tìm kiếm đơn đặt lịch cho quầy tiếp nhận. Chức năng xem lịch dành cho bác sĩ.

Tự động hóa quy trình đăng ký lịch khám bệnh thông qua ứng dụng di động.

Giúp bệnh nhân chủ động và tiết kiệm thời gian trong việc đăng ký khám bệnh.

Giúp phòng khám dễ dàng theo dõi lịch hẹn, chủ động trong việc quản lý thời gian và tương tác với bệnh nhân.

## Nội dung

Xây dựng ứng dụng đăng ký lịch khám bệnh cho phòng khám gồm những công việc như:

- Phân tích tình trạng hiện tại của quy trình đăng ký lịch khám bệnh.

- Thiết kế giao diện người dùng và các tính năng cần thiết.

- Xây dựng ứng dụng dựa trên thiết kế đã đề xuất.

## Đối tượng và phạm vi nghiên cứu

**Đối tượng nghiên cứu**

Tìm hiểu quy trình đăng ký đặt lịch khám trực tuyến dành cho khám dịch vụ và khám có bảo hiểm y tế.

Ứng dụng di động đăng ký đặt lịch khám bệnh.

Các đối tượng như Model, View, Controller, Route… trong mô hình MVC.

Các thành phần chính của Flutter Framework.

Các công nghệ, ngôn ngữ lập trình có sử dụng để xây dựng ứng dụng di động.

**Phạm vi nghiên cứu**

Phạm vi nghiên cứu tập trung vào ứng dụng Flutter Framework để xây dựng một ứng dụng di động đăng ký đặt lịch khám bệnh cho phòng khám. Nghiên cứu sẽ tập trung vào hiểu rõ mô hình MVC trong Flutter.

Quy trình đăng ký đặt lịch khám trực tuyến.

## Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết:

Thu thập tài liệu về nghiệp vụ của ứng dụng đặt lịch khám bệnh, ngôn ngữ lập trình và các công cụ, dịch vụ, nền tảng lưu trữ dữ liệu thực hiện.

Tìm hiểu về quy trình đăng ký đặt lịch khám đối với trường hợp khám dịch vụ và khám có bảo hiểm y tế, quản lý thông tin khách hàng, xác nhận hoặc hủy đơn đặt lịch dành cho quầy tiếp nhận.

Phương pháp nghiên cứu thực nghiệm: Xây dựng ứng dụng minh họa cho vấn đề nghiên cứu.

Phương pháp thực nghiệm: Xây dựng ứng dụng minh họa cho vấn đề nghiên cứu.

# CƠ SỞ LÝ THUYẾT

## Công nghệ, ngôn ngữ, công cụ thực hiện

### *Flutter Framework*

**Giới thiệu về Flutter**

Flutter là một Mobile Framework của Google dùng để tạo giao diện chất lượng cao trên các hệ điều hành iOS và Android.

Flutter là một mã nguồn mở và hoàn toàn miễn phí [3].

**Quá trình phát triển của Flutter**

Năm 2015, Flutter Framework được công bố bởi Google, hoạt động với ngôn ngữ Dart, làm nền tảng để phát triển Android.

Năm 2017, phiên bản Alpha đã được phát hành lần đầu tiên. Tại sự kiện I/O cùng năm Google đã quảng bá việc Flutter sẽ phát triển đa nền tảng.

Năm 2018, Flutter 1.0 được phát hành tại sự kiện Flutter Live và có sẵn SDK để các nhà phát triển sử dụng nhằm tạo ứng dụng dễ dàng hơn.

Năm 2019, tại Google I/O, Flutter 1.12 ra đời và Flutter dành cho các nền tảng Windows, macOS, Linux đã được phát hành.

Năm 2021, Flutter cập nhật phiên bản 2.0.6.

Năm 2022, Flutter cập nhật phiên bản 3.0.

Phiên bản mới nhất của Flutter là 3.22.2 tính đến ngày 1/6/2024 [3].

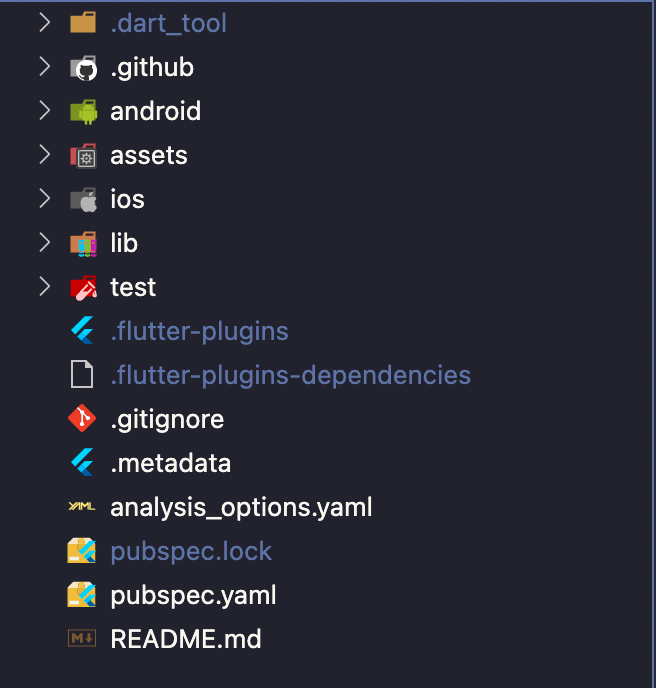
**Các thành phần chính của Flutter Framework**

SDK – Software Development Kit (Bộ công cụ phát triển phần mềm) là các công cụ giúp pháp triển ứng dụng, gồm các công cụ để biên dịch mã nguồn dùng cho hai hệ điều hành là Android và iOS.

Framework – UI Library Based on Widgets (Thư viện giao diện người dùng dựa trên widgets) là tập hợp các phần tử giao diện có thể tái sử dụng như Button, Text Input, Slider giúp người dùng có thể sáng tạo giao diện ứng dụng theo nhu cầu cá nhân [3].

**Cấu trúc thư mục của Flutter**

Cấu trúc thư mục của Flutter được tổ chức một cách rõ ràng cho việc phát triển ứng dụng theo các nền tảng khác nhau như iOS và Android.



Hình 2.1 Cấu trúc thư mục của Flutter

Bảng 2.1 Bảng giải thích cấu trúc thư mục quan trọng của Flutter Framework

|  |  |  |
| --- | --- | --- |
| **STT** | **Đối tượng** | **Mô tả** |
| 1 | android/ | Chứa các tệp cấu hình và mã nguồn gốc của ứng dụng Android |
| 2 | ios/ | Chứa các tệp cấu hình và mã nguồn gốc của ứng dụng iOS. |
| 3 | lib/ | Thư mục này chứa tất cả các mã nguồn Dart của ứng dụng. File main.dart là tệp chính được chạy đầu tiên khi ứng dụng khởi động. |
| 4 | test/ | Chứa các tệp kiểm thử cho ứng dụng. |
| 5 | web/, windows/, macos/, linux/ | Các thư mục này chứa mã nguồn và cấu hình cho các nền tảng tương ứng nếu người lập trình đang phát triển ứng dụng đa nền tảng. |
| 6 | .dart\_tool/ | Thư mục chứa các công cụ Dart cần thiết cho dự án. |
| 7 | pubspec.yaml | Tệp này khai báo các dependencies của dự án, các asset, và các thông tin khác về dự án. |
| 8 | build.gradle | Tệp cấu hình Gradle dùng để build ứng dụng Android. |
| 9 | .vscode/ | Thư mục này chứa các tệp cấu hình cho Visual Studio Code. |
| 10 | build/ | Thư mục này được tạo tự động trong quá trình build ứng dụng và chứa các tệp tạm thời được tạo ra trong quá trình build. Không cần phải chỉnh sửa các tệp trong thư mục này. |

**Cấu hình Flutter Framework**

*Cấu hình biến môi trường*

Tải dự án Flutter từ trang chủ, giải nén file, vào thư mục bin và sao chép đường dẫn, vào thêm biến môi trường (edit the system environment variables), dán đường dẫn vừa sao chép vào.

Tiếp theo vào Visual Studio Code, chọn thư mục sẽ lưu dự án, mở terminal và chạy lệnh “Flutter create Tenduan”.

*Hướng dẫn chạy dự án*

Tải Android Studio, tải về các thành phần như Android SDK, trình giả lập, để kiểm tra các thành phần này, chạy lệnh “flutter doctor”.

Để chạy chương trình đầu tiên, vào thư mục lib, chọn file main.dart và chọn “run”.

*Cấu hình với Firebase*

Tạo một Firestore Database bằng cách ấn vào Create a project tại trung chủ Firebase.

Vào file android/app/src/build.gradle, tại dòng code

android{

namespace “ten package du an”

compileSdkVersion flutter.compileSdkVersion

}

Sao chép “ten package du an” và dán vào Android package name của Firebase. Sau đó đặt tên cho dự án.

Tải xuống file config từ Firebase và đưa vào thư mục app.

Sau đó tiếp tục thêm SDK Firebase và file android/app/src/build.gradle

dependencies {

implementation 'com.google.firebase:firebase-auth:20.0.0'

implementation 'com.google.firebase:firebase-firestore:23.0.0'

}

**Điểm nổi bật của Flutter Framework**

Tính năng Host Reload: Tính năng này hoạt động trong milliseconds để hiện thị giao diện tới lập trình viên. Sử dụng các widget có thể tùy chỉnh giúp lập trình viên tạo các giao diện chỉ trong vài phút. Ngoài ra, tính năng này còn hỗ trợ thêm fix bug mà không cần phải thông qua máy ảo, máy android hoặc iOS giúp tiết kiệm thời gian hơn.

Expressive and Flexible UI - Giao diện người dùng linh hoạt và biểu cảm: hỗ trợ nhiều các thành phần để xây dựng giao diện của Flutter vô cùng tiện lợi theo các phong cách như Material Design và Cupertino, hỗ trợ API chuyển động, smooth scrolling...

Native Performance – Hiệu xuất gốc: Các widget của fluter kết hợp các sự khác biệt của các nền tảng ví dụ như scrolling, navigation, icons, font để cung cấp một hiệu năng tốt nhất tới iOS và Android [3].

**Nhược điểm**

Để sử dụng được Flutter, lập trình viên hoặc các nhà phát triển phải học thêm ngôn ngữ lập trình Dart vì ngôn ngữ lập trình này không phổ biến, do vậy sẽ gặp khó khăn hơn trong việc phát triển các ứng dụng trên thiết bị di động.

Cập nhật lên kho ứng dụng khó khăn hơn vì phải thông qua trung tâm ứng dụng như App Store hoặc Play Store [3].

### *Firebase*

**Giới thiệu về Firebase**

Firebase là một trong những BaaS, hoạt động trên nền tảng đám mây – cloud cung cấp các giải pháp backend cho các ứng dụng web và di động được hỗ trợ bởi Google. Sử dụng các API và SDK do Firebase cung cấp để kết nối với các dịch vụ ấy.

Firebase là một dịch vụ đa năng và bảo mật cực tốt, hỗ trợ cả hai nền tảng Android và iOS.

Firebase sử dụng cơ sở dữ liệu NoSQL. Nó không có các bảng với các hàng và cột cố định như trong cơ sở dữ liệu quan hệ. Thay vào đó, dữ liệu được lưu trữ dưới dạng JSON trong các tài liệu hoặc đối tượng, cho phép linh hoạt hơn trong việc lưu trữ và truy cập dữ liệu phi cấu trúc hoặc bán cấu trúc [8].

**Lịch sử phát triển của Firebase**

Firebase được phát triển từ một công ty khởi nghiệp ban đầu có tên Envolve, được thành lập bởi James Tamplin và Andrew Lee vào năm 2011. Tháng 9 cùng năm, họ thành lập Firebase như một công ty riêng. Đến tháng 4 năm 2012, Firebase chính thức ra mắt.

Sản phẩm đầu tiên của Firebase là Firebase Realtime Database, một API giúp đồng bộ hóa dữ liệu ứng dụng trên các thiết bị iOS, Android và Web, đồng thời lưu trữ trên nền tảng đám mây của Firebase. Sản phẩm này hỗ trợ các nhà phát triển phần mềm xây dựng các ứng dụng có tính năng cộng tác theo thời gian thực.

Vào tháng 10 năm 2014, Google đã mua lại Firebase. Từ đó đến nay, Firebase đã liên tục ra mắt nhiều tính năng mới và trở thành một công cụ phổ biến được nhiều nhà phát triển sử dụng trong các dự án của mình [8].

**Những service nổi bật**

Dịch vụ Authentication hỗ trợ cho ứng dụng một số phương pháp xác thực thông qua email, mật khẩu, số điện thoải, tài khoản Facebook, tài khoản Google… Tính năng này sẽ giúp lập trình viên dễ dàng xây dựng chức năng đăng nhập mà không cần sử dụng dữ liệu đăng ký riêng.

Firestore: Một cơ sở dữ liệu đám mây có thể mở rộng để lưu trữ và truy vấn dữ liệu cho ứng dụng.

Firebase Authentication: Là dịch vụ có chức năng xác thực bằng mật khẩu, email, tài khoản Google, Facebook [8].

**Điểm mạnh của Firebase**

Tất cả dữ liệu được truyền qua một kết nối an toàn SSL. Kết nối này đảm bảo rằng dữ liệu được truyền giữa host và client được hoạt động một cách riêng tư, đáng tin cậy. Các logic bảo mật dữ liệu được tập trung ở một nơi để dễ dàng cho việc sửa đổi, cập nhật và kiểm thử.

Làm việc Offline Ứng dụng sẽ duy trì tương tác dù gặp vấn đề như mạng chậm chờn, mất mạng hay mạng yếu, dữ liệu ấy sẽ được ghi tạm. Sau khi kết nối lại với internet, client sẽ nhận bất kỳ thay đổi và đồng bộ hoá nó với cơ sở dữ liệu tại Firebase. Tuy nhiên, người dùng vẫn cần phải trả phí để có thể nâng cấp Firebase khi quy mô ứng dụng đủ lớn [8].

**Nhược điểm của Firebase**

Các dịch vụ của Firebase đều hoàn toàn miễn phí nhưng lại bị giới hạn, nếu muốn nâng cấp, mở rộng cần phải trả phí. Cần phải cân nhắc nếu muốn xây dựng hệ thống lớn vì chi phí nâng cấp khá đắt đỏ [8].

### *Ngôn ngữ Dart*

Dart là ngôn ngữ lập trình hướng đối tượng đa mục đích mã nguồn mở, được đặt nền móng và phát triển bởi Google. Lập trình hướng đối tượng thể hiện qua việc hỗ trợ giao diện và lớp. Sự đa dạng của Dart thể hiện thông qua khả năng phát triển ứng dụng web, di động, máy chủ và máy tính để bàn [7].

**Lịch sử hình thành và các phiên bản**

Trong lĩnh vực lập trình, Dart là một dự án tâm huyết của Lars Bak và Kasper Lund, được phát triển dưới sự quản lý của Google. Dart lần đầu tiên được giới thiệu tại hội nghị GOTO ở Đan Mạch vào ngày 10 tháng 10 năm 2011, và nhanh chóng cho thấy tiềm năng lớn của mình. Một cột mốc quan trọng là việc phát hành phiên bản Dart 1.0 vào ngày 14 tháng 11 năm 2013. Mặc dù ban đầu Dart nhận được những phản hồi trái chiều, nhưng điều này không thể cản trở sự phát triển vượt bậc của nó.

Giai đoạn này cũng đánh dấu sự chuyển đổi từ kế hoạch tích hợp máy ảo Dart vào Chrome sang việc biên dịch mã Dart thành JavaScript. Điều này rất quan trọng đối với phiên bản Dart 2.0, ra mắt vào tháng 8 năm 2018, mang đến nhiều cải tiến về ngôn ngữ, bao gồm cả hệ thống kiểu dữ liệu mới.

Dart có nhiều phiên bản khác nhau. Mới nhất chính là Dart 3.0, ra đời vào ngày 10 tháng 5 năm 2023.

**Điểm nổi bật của Dart**

Cú pháp Dart rõ ràng và ngắn gọn, công cụ đơn giản nhưng mạnh mẽ. Type-safe giúp xác định sớm các lỗi.

Dart tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.

Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript [7].

**Nhược điểm**

Ngôn ngữ lập trình Dart là một ngôn ngữ lập trình khá mới nên hiện tại cộng đồng người sử dụng có quy mô nhỏ, chưa có nhiều tài liệu phục vụ cho việc học tập.

Dart chủ yếu được biết đến và sử dụng trong việc phát triển ứng dụng di động với Flutter. Việc sử dụng Dart ngoài lĩnh vực này vẫn còn hạn chế, làm giảm tính linh hoạt của nó so với các ngôn ngữ đa năng khác [7].

### *Mô hình MVC*

**Giới thiệu về mô hình MVC**

MVC là mô hình được thiết kế bởi tiến sỹ Trygve Reenskaug, dùng trong kỹ phần mềm. Mô hình được sử dụng để tạo giao diện người dùng. Được chia thành 3 phần riêng lẻ có nhiệm vụ và chức năng khác nhau:

Model: Quản lý và xử lý dữ liệu, có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng, là liên kết giữa View và Controller. Được thể hiển dưới hình thức là một Cơ sở dữ liệu hoặc đơn giản là một file XML, cho phép xem và truy xuất, thao tác với dữ liệu.

View: Hiển thị dữ liệu, là phần giao diện của Website, người dùng có thể lấy dữ liệu thông qua truy vấn. Lưu ý rằng View không tác động trực tiếp với Controller, View chỉ có chức năng hiển thị yêu cầu và chuyển cho Controller chứ không thể tự lấy dữ liệu ấy từ Controller.

Controller: Điều khiển sự tương tác giữa Models và View, có nhiệm vụ xử lý các yêu cầu của người dùng thông qua View, từ đó lấy các thông tin mà người dùng cần [2].

**Ưu điểm của mô hình MVC**

Tạo cho dự án theo một mô hình chuẩn.

Rõ ràng về mặt trình tự, mỗi phần đều có tách biệt và có nhiệm vụ riêng

Dễ dàng kiểm tra, kiểm soát, sửa chữa, bảo trì nếu có lỗi.

Tính chuyên nghiệp cao, dễ kiểm tra lỗi trước khi bàn giao cho khách hàng [2].

**Nhược điểm của mô hình MVC**

Tốn nhiều thời gian và công sức để quản lý tổ chức của file.

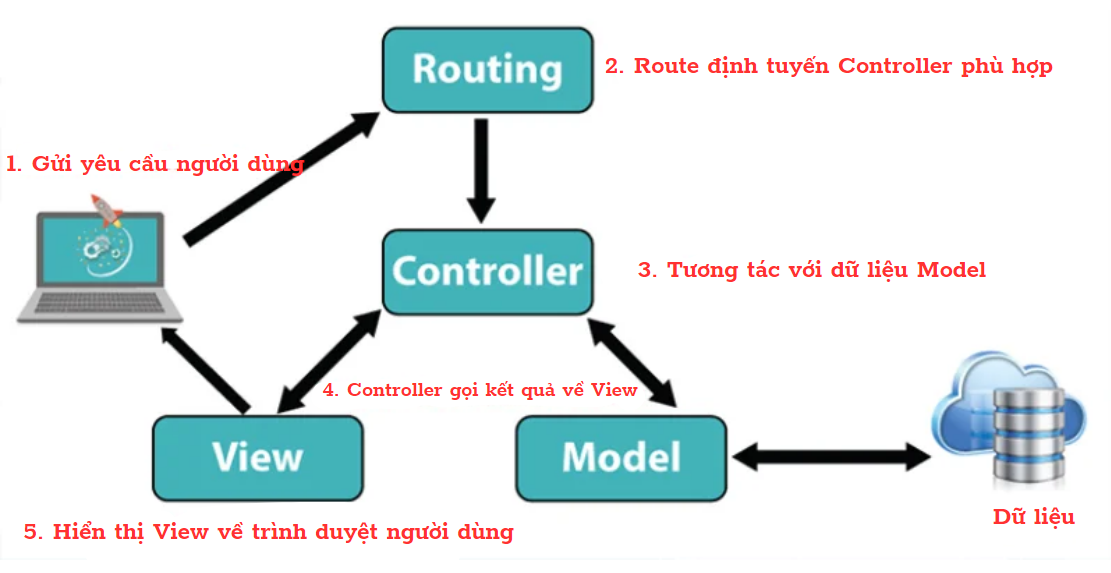
Mô hình tương đối phức tạp, không thích hợp với các dự án quy mô nhỏ.

Yêu cầu bảo trì có kỹ năng cao.

Khó triển khai.

Việc giao tiếp giữa các thành phần phức tạp, khó khăn, đặc biệt là đối với ứng dụng lớn [2].

**Cách MVC hoạt động**



Hình 2.2 Mô hình MVC

Người dùng gửi yêu cầu từ trình duyệt lên Web Server, Routes sẽ Dispatcher (điều hướng) yêu cầu của người dùng đến Controller tương ứng. Controller ấy sẽ lấy dữ liệu (Data) mà người dùng yêu cầu từ Model và trả ngược lại Controller. Tiếp theo, sau khi lấy được dữ liệu từ Model, Controller này sẽ lấy View tương ứng với Model chứa yêu cầu đã lấy được và trả về Web Server và hiển thị dữ liệu yêu cầu mà người dùng đã gửi lên trình duyệt [2].

### *Visual Studio Code*

Visual Studio Code là một trình soạn thảo được các lập trình viên ưa chuộng. Ưu điểm của Visual Studio Code nổi bật là gợi ý các câu lệnh và tự hoàn thành cấu trúc cơ bản của các thẻ, nhẹ và hỗ trợ nhiều ngôn ngữ và vận hành mượt mà trên các nền tảng như Windows, MacOS và Linux. Ngoài ra Visual Studio Code còn hỗ trợ thêm các chức năng Debug, Git, Extensions hỗ trợ rất nhiều cho lập trình viên.

Khi sử dụng Visual Studio Code để xây dựng một ứng dụng di động, chúng ta cần phải cài đặt thêm các Extensions là Dart và Flutter.

## Nghiệp vụ liên quan đến đề tài

### *Thanh toán điện tử*

Thanh toán điện tử là hình thức thanh toán trên Internet, cho phép thực hiện các cuộc giao dịch tài chính thay vì sử dụng tiền mặt.

**Ưu điểm**

Nhanh chóng, tiện lợi, phù hợp với thị trường, giúp khách hàng dễ dàng chuyển và nhận thông qua điện thoại đã kết nối với internet. Chính vì vậy, giao dịch sẽ dễ dàng, nhanh chóng và tiện lợi, không cần mang nhiều tiền mặt trong người và tiết kiệm thời gian.

Chuyên nghiệp hóa.

Dễ dàng kiểm soát, mọi giao dịch sẽ được lưu lại trong lịch sử, giúp quản lý hiệu quả hơn .

Tính bao mật cao, khi thanh toán thông tin sẽ được bảo mật một cách cẩn thận, các giao dịch đều yêu cầu xác nhận mã PIN hoặc OTP để đảm bảo an toàn.

Hạn chế rủi ro khi mang quá nhiều tiền mặt bên người hoặc quên mang ví, chính vì thế thanh toán trực tuyến sẽ giúp hạn chế những rủi ro một cách hiệu quả.

**Các hình thức thanh toán điện tử**

Thanh toán qua ví điện tử.

Thanh toán qua cổng thanh toán điện tử.

Thanh toán bằng thẻ.

Thanh toán bằng cách quét mã.

## Các công trình nghiên cứu liên quan

Đầu tiên, Trần Hữu Cảnh, Nguyễn Hoàng Vũ của trường Đại học Quốc gia TP.Hồ Chí Minh đã hoàn thành Website đặt lịch khám bệnh online. Nhóm tác giả đã hoàn thiện được ứng dụng web đặt lịch khám bệnh online, giao diện dễ thao tác, đăng ký đặt lịch khám và hiển thị thông tin cho người dùng [9].

Tiếp đến là Trần Minh Quân, Hoàng Phước Thành, Nguyễn Phạm Minh Quân, Lê Nguyễn Hữu Quốc thuộc Đại học FPT TP.HCM đã hoàn thành HABS - Hệ thống quản lý đặt lịch và khám bệnh cho bệnh viện . Nhóm tác giả đã hoàn thiện hệ thống đặt lịch khám bệnh bao gồm website và ứng dụng di động, với giao diện dễ dàng sử dụng thân thiện, phù hợp với người dùng sớm được đưa vào hoạt động thực tế [10].

Cuối cùng, nhóm sinh viên gồm Trần Phạm Hồng Trúc, Đỗ Kiến Nhuần, Trần Lê Hồng Phúc, Nguyễn Văn Tùng đã hoàn thành dự án HI! MEDICARE hỗ trợ người sử dụng đặt lịch khám sức khỏe tại địa điểm khám chữa bệnh tại nhà hoặc đăng ký thăm khám trực tuyến, kiểm soát hồ sơ khám bệnh, hồ sơ xét nghiệm hoặc tiêm chủng dễ dàng trong một ứng dụng di động [11].

Qua tìm hiểu các công trình trên, tôi sẽ tiến hành xây dựng một ứng dụng đăng ký đặt lịch khám bệnh cho phòng khám dành cho thiết bị di động bằng Flutter Framework. Ứng dụng bao gồm các chức năng tạo hồ sơ bệnh nhân để đăng ký đặt khám, xem và chọn phòng khám, chọn thời gian để đăng ký khám và thanh toán trực tuyến. Ứng dụng phân quyền cho người dùng, bác sĩ và nhân viên quầy tiếp nhận để đảm bảo đúng quy trình đăng ký đặt khám trực tuyến.

# HIỆN THỰC HÓA NGHIÊN CỨU

## Mô tả bài toán

Bài toán xây dựng ứng dụng đăng ký đặt lịch khám bệnh được thực hiện nhằm tự động hóa quy trình đặt lịch của các bệnh nhân, giúp giảm thiểu thời gian và công sức của cả bệnh nhân và các cơ sở y tế. Ứng dụng này cho phép người dùng dễ dàng chọn thời gian khám phù hợp, và đặt lịch hẹn trực tuyến chỉ với vài thao tác đơn giản. Hệ thống cũng hỗ trợ quản lý thông tin bệnh nhân, cung cấp các thông tin liên quan đến phòng khám. Bằng cách số hóa quy trình đăng ký, ứng dụng không chỉ nâng cao hiệu quả vận hành mà còn cải thiện trải nghiệm của người dùng, đồng thời giảm tải khối lượng công việc cho nhân viên y tế.

### *Yêu cầu hệ thống*

**Hệ thống sẽ có các yêu cầu cơ bản như:**

Chức năng đăng ký, đăng nhập tài khoản: Cho phép bệnh nhân đăng ký, đăng nhập tài khoản vào hệ thống.

Chức năng đăng ký đặt lịch khám: Cho phép bệnh nhân chọn phòng khám, ngày khám, giờ khám.

Chức năng quản lý hồ sơ bệnh nhân: Bao gồm họ và tên bệnh nhân, năm sinh, giới tính, mối quan hệ trong gia đình, số điện thoại, quê quán, căn cước công dân, số bảo hiểm y tế.

Chức năng quản lý lịch khám: Bao gồm thông tin về thời gian đăng ký đặt, phòng khám mà bệnh nhân đã đặt.

Chức năng thanh toán online: Người dùng thanh toán trên ứng dụng mà không cần phải sử dụng tiền mặt để thanh toán.

Bác sĩ có thể xem được lịch đăng ký khám của các bệnh nhân bao gồm tên bệnh nhân, mã bệnh nhân, ngày, giờ khám, bệnh nhân có hay không có bảo hiểm y tế, đã thanh toán hay chưa thanh toán.

Nhân viên quầy tiếp nhận có thể xem được lịch đăng ký khám của bệnh nhân, xác nhận các đơn hợp lệ và hủy các đơn không hợp lệ, xem lịch đăng ký khám của các phòng khám, xem thống kê các đơn đăng ký đặt khám.

### *Quy trình làm việc*

**Người dùng**

Người dùng cài đặt ứng dụng vào thiết bị di động.

Tiếp theo đăng ký tài khoản bằng mail và đăng nhập tài khoản ấy vào ứng dụng.

Người dùng có thể xem giới thiệu ứng dụng để hiểu rõ hơn về chức năng có trong ứng dụng.

Người dùng có thể xem hướng dẫn sử dụng để hiểu rõ hơn về quy định và quy trình làm việc của ứng dụng.

Người dùng vào “Quản lý hồ sơ” để tạo hồ sơ bệnh nhân, thêm các thông tin bao gồm họ và tên, số điện thoại, ngày sinh, giới tính, địa chỉ, số bảo hiểm y tế, số căn cước công dân/ chứng minh nhân dân để tạo hồ sơ mới.

Sau khi tạo xong hồ sơ, người dùng ấn vào “Đặt lịch khám” để đăng ký đặt lịch.

Đầu tiên, chọn phòng khám gồm các phòng có trong danh sách.

Tiếp theo chọn ngày khám.

Tiếp đến, chọn giờ khám trong ngày đã chọn ở bước trên.

Sau đó, chọn hồ sơ bệnh nhân đã tạo, sau đó ấn xác nhận.

Trường hợp bệnh nhân có bảo hiểm y tế, sẽ không thanh toán trực tuyến mà sẽ đến quầy tiếp nhận để người quản lý quầy tiếp nhận kiểm tra bảo hiểm y tế có hợp lệ hay không. Trường hợp bệnh nhân không có bảo hiểm y tế sẽ thanh toán trực tuyến qua ứng dụng, sau khi thanh toán xong, ấn xác nhận để hoàn thành.

Người dùng sau khi đăng ký xong ấn vào “Danh sách lịch khám” để theo dõi đơn đặt lịch của mình.

Lưu ý bệnh nhân phải đến sớm hơn giờ đã đăng ký 15 phút để quầy tiếp nhận kiểm tra đơn đã đăng ký [1].

**Quầy tiếp nhận**

Sau khi người dùng đăng ký, giao diện quầy tiếp nhận sẽ hiển thị các đơn đăng ký bao gồm mã bệnh nhân, ngày đăng ký, giờ đăng ký, khám dịch vụ hay khám có bảo hiểm y tế.

Bệnh nhân sẽ đến quầy tiếp nhận để người quản lý kiểm tra thông tin đã đăng ký, kiểm tra hạn sử dụng của bảo hiểm y tế, nếu bảo hiểm y tế hợp lệ ấn vào nút “Xác nhận” để đơn đăng ký chuyển đến bác sĩ trực ngày tương ứng. Nếu bảo hiểm y tế không hợp lệ ấn vào nút “Hủy” hủy đơn đăng ký ấy [1].

Người quản lý có thể xem thống kê các đơn đăng ký theo từng phòng khám.

Tìm kiếm giúp người quản lý có thể nhanh chóng tìm và xác nhận các đơn giúp tiết kiệm thời gian.

**Bác sĩ**

Sau khi được kiểm tra và xác nhận bởi nhân viên quầy tiếp nhận, đơn đăng ký khám hợp lệ sẽ hiển thị ở giao diện bác sĩ bao gồm: Mã bệnh nhân, ngày đăng ký, giờ đăng ký, khám bảo hiểm y tế hay khám dịch vụ [1].

### *Yêu cầu chức năng*

**Ứng dụng cho phép bệnh nhân**

Đăng ký tài khoản và đăng nhập vào hệ thống.

Cập nhật thông tin cá nhân vào hồ sơ.

Xem giới thiệu về ứng dụng để hiểu rõ về các chức năng của ứng dụng.

Xem hướng dẫn sử dụng ứng dụng để hiểu rõ về quy định và quy trình đăng ký.

Xem danh sách các phòng khám của bệnh viện.

Đăng ký đặt lịch khám bệnh.

Quản lý lịch khám bệnh như xem cuộc hẹn sắp diễn ra và kết thúc.

Người dùng được thanh toán điện tử.

Đăng xuất tài khoản.

**Ứng dụng cho phép bác sĩ**

Xem lịch khám mà các bệnh nhân đã đăng ký khi được người quản lý quầy tiếp nhận xác nhận.

**Ứng dụng cho phép quầy tiếp nhận**

Xác nhận các đơn hợp lệ hoặc hủy các đơn không hợp lệ các đơn đăng ký của bệnh nhân.

Xem lịch đặt khám của các bệnh nhân theo từng phòng khám.

Xem thống kê số lượt đăng ký đặt khám theo từng phòng khám.

Tìm kiếm các đơn đăng ký theo mã bệnh nhân giúp tiết kiệm thời gian.

## Thiết kế dữ liệu

### *Cấu trúc dữ liệu dưới dạng JSON*

**Đối tượng users**

{

"user\_id": "",

"email": "",

"password": "",

"role": "",

"hospital\_id": ""

}

**Đối tượng medical**

{

"medical\_ID": "",

"dob": "",

"gender": "",

"identificationCard": "",

"identificationCardExpired": "",

"insuranceCard": "",

"insuranceCardExpired": "",

"name": "",

"phone": "",

"streetAddress": "",

"userId": ""

}

**Đối tượng hospital**

{

"hospital\_ID": "",

"address": "",

"description": "",

"picture": "",

"title": ""

}

**Đối tượng appointments**

{

"appointment\_ID": "",

"hospitalId": "",

"medicalId": "",

"paymentId": "",

"status": "",

"success": ,

"time\_slot": "",

"userId": "",

"amount": ,

"date": ""

}

**Đối tượng payment**

{

"payment\_id": "",

"status": "",

"timestamp": "",

"transaction\_ref": ""

}

### *Dữ liệu mẫu*

**Đối tượng users**

{

"user\_id": "xTfdszWg2gdaXcsqw3y8HS3gN7V2",

"email": "nhan@gmail.com",

"hospitalId": "null",

"role": "user"

}

**Đối tượng medical**

{

"dob": "2024-06-25",

"gender": "Nam",

"identificationCard": "",

"identificationCardExpired": "",

"insuranceCard": "",

"insuranceCardExpired": "",

"isDeleted": false,

"name": "Nguyễn Thị Đẹp",

"phone": "0342903415",

"streetAddress": "Tra Vinh",

"userId": "xTfdszWg2gdaXcsqw3y8HS3gN7V2"

}

**Đối tượng hospital**

{

"address": "Phòng 2.1 Tầng 2",

"description": "Tai – Mũi – Họng",

"picture": "https://taimuihongsg.com/wp-content/uploads/2021/12/icon-khoa-tai-mui-hong\_taimuihongsg.jpg",

"rating": "5",

"title": "Phòng khám Tai - Mũi - Họng"

}

**Đối tượng appointments**

{

"amount": 0,

"date": "2024-06-26",

"hospitalId": "5",

"isDeleted": false,

"medicalId": "cLwpKMMb4KVcH4warVki",

"paymentId": "424927",

"status": "success",

"time\_slot": "06:45 - 07:00",

"userId": "xTfdszWg2gdaXcsqw3y8HS3gN7V2"

}

**Đối tượng payment**

{

"status": "success",

"timestamp": "June 25, 2024 at 00:15:17 UTC+7",

"transaction\_ref": "424927"

}

### *Danh sách các bảng dữ liệu*

Bảng 3.1 Bảng hospital (Phòng khám)

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** |
| hospital\_ID | Mã phòng khám | String |
| address | Địa chỉ phòng khám | String |
| description | Mô tả | String |
| picture | Hình ảnh | String |
| title | Tên phòng khám | String |

Bảng 3.2 Bảng medical (Hồ sơ)

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** |
| medical\_ID | Mã hồ sơ | String |
| dob | Ngày sinh | Date and time |
| gender | Giới tính | String |
| identificationCard | Căn cước công dân | String |
| identificationCardExpired | Ngày cấp | Date and time |
| insuranceCard | Bảo hiểm y tế | String |
| insuranceCardExpired | Ngày hết hạn | Date and time |
| isDeleted | Xóa | Boolean |
| name | Tên người dùng | String |
| phone | Số điện thoại | String |
| streetAddress | Địa chỉ | String |
| userId | Mã người dùng | String |

Bảng 3.3 Bảng payments (Thanh toán)

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** |
| payment\_id | Mã thanh toán | String |
| status | Trạng thái thanh toán | String |
| timestamp | Mốc thời gian | Date and time |
| transaction\_ref | Mã giao dịch | String |

Bảng 3.4 Bảng users (Tài khoản)

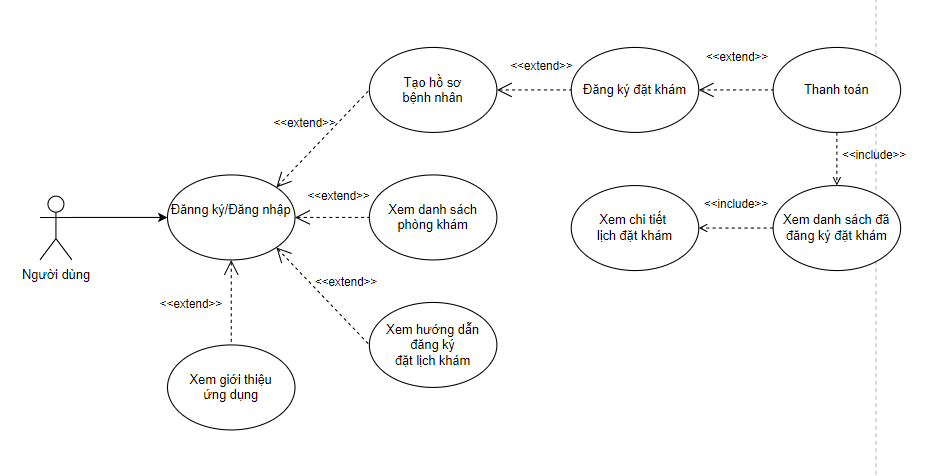
|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** |
| user\_ID | Mã tài khoản | String |
| E-mail | Email | String |
| hospitalId | Mã phòng khám | String |
| password | Mật khẩu | String |
| role | Vai trò | String |

Bảng 3.5 Bảng appointments (Đơn đặt lịch khám)

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** |
| appoinment\_ID | Mã đơn đặt lịch | String |
| hospitalId | Mã phòng khám | String |
| isDeleted | Xóa | Boolean |
| medicalId | Mã hồ sơ | String |
| paymentId | Mã thanh toán | String |
| status | Trạng thái | String |
| success | Xác nhận | String |
| time\_slot | Giờ đặt lịch | String |
| userId | Mã người dùng | String |
| amount | Giá | String |
| date | Ngày đặt lịch | Date and time |

## Thiết kế xử lý

### ***Sơ đồ use case Người dùng***



Hình 3.1 Sơ đồ Use case Người dùng

**Người dùng khi đăng ký tài khoản có thể**

Đăng nhập, đăng xuất.

Xem giới thiệu ứng dụng và hướng dẫn sử dụng.

Xem danh sách các phòng khám.

Tạo hồ sơ bệnh nhân.

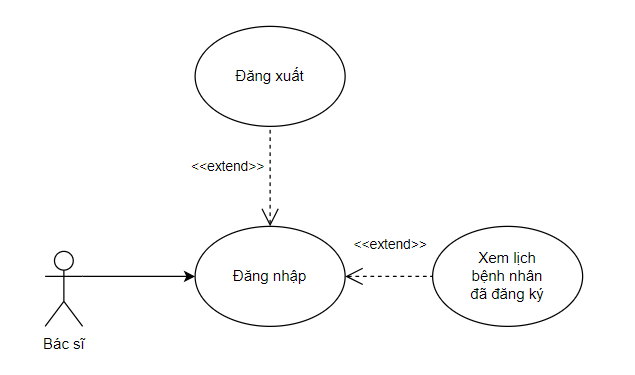
Đăng ký đặt lịch khám.

Thanh toán điện tử.

Xem danh sách đã đăng ký đặt khám.

Xem chi tiết lịch đặt khám.

### ***Sơ đồ use case Bác sĩ***

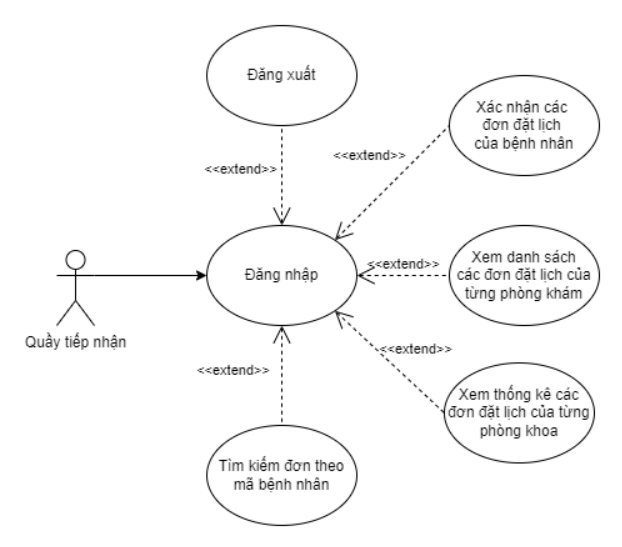


Hình 3.2 Sơ đồ Use case Bác sĩ

**Bác sĩ sẽ đăng nhập sau khi được cấp tài khoản:**

Xem được thông tin đăng ký của bệnh nhân theo từng phòng khoa.

### ***Sơ đồ use case Người quản lý quầy tiếp nhận***



Hình 3.3 Sơ đồ Use case Người quản lý quầy tiếp nhận

**Nhân viên quầy tiếp nhận có thể:**

Đăng nhập, đăng xuất.

Xem danh sách các đơn đặt lịch của bệnh nhân theo từng phòng khoa.

Xem thống kê các đơn đặt lịch của bệnh nhân theo từng phòng khoa.

Xác nhận các đơn đặt lịch hợp kệ hoặc hủy các đơn đặt lịch không hợp lệ.

Tìm kiếm đơn đăng ký theo mã bệnh nhân.

# KẾT QUẢ NGHIÊN CỨU

## Giao diện người dùng và chức năng

**- Giao diện khi bắt đầu ứng dụng**

Khi bắt đầu sử dụng, ứng dụng cho phép người dùng đăng ký tài khoản và đăng nhập.



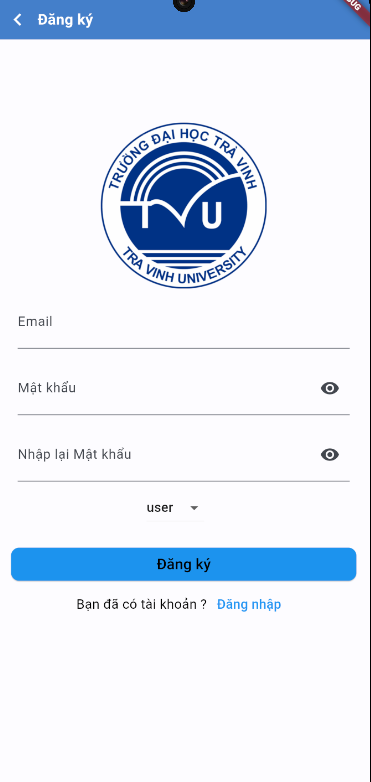
Hình 4.1 Giao diện khi bắt đầu ứng dụng

**- Giao diện đăng ký tài khoản**

Ứng dụng cho phép người dùng đăng ký tài khoản bằng Email.

Biểu tượng hình con mắt cho phép ẩn/ hiện khi người dùng nhập mật khẩu.

Nếu người dùng đã có tài khoản có thể ấn vào chữ Đăng nhập màu xanh ở phía dưới để chuyển sang giao diện đăng nhập tài khoản.

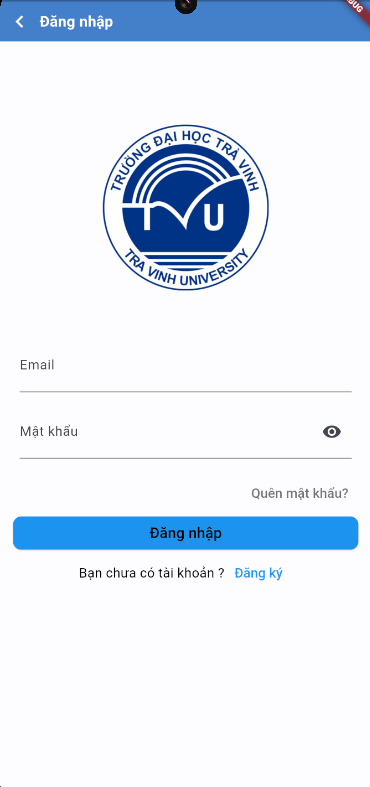


Hình 4.2 Giao diện đăng ký tài khoản

**- Giao diện đăng nhập**

Ứng dụng cho phép người dùng đăng nhập sau khi đã đăng ký tài khoản.

Biểu tượng con mắt cho phép người dùng ẩn/ hiện mật khẩu khi ấn vào.



Hình 4.3 Giao diện đăng nhập

**- Giao diện trang chủ**

Tại đây người dùng có thể thao tác với:

Widget Slider (thanh trượt).

“Danh sách lịch khám” để xem lịch mà người dùng đã đăng ký.

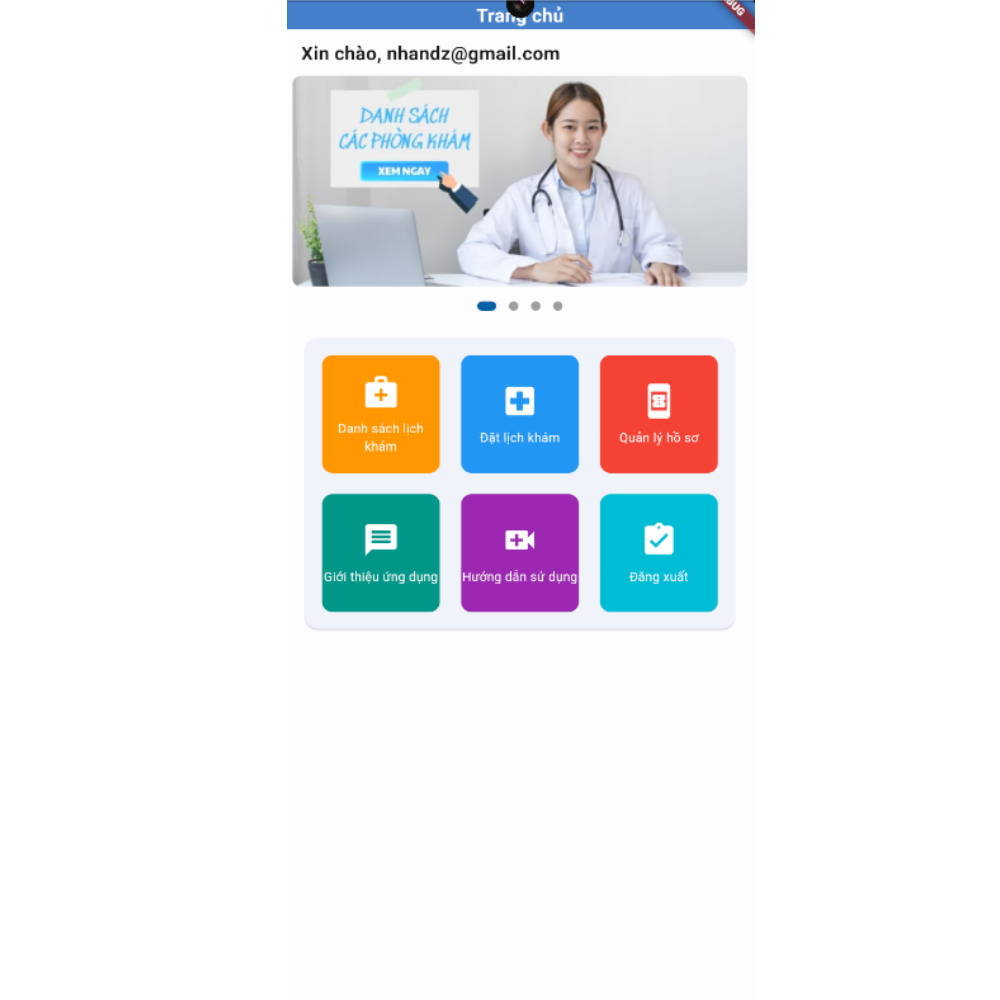
“Đặt lịch khám” để xem các phòng khám và đăng ký đặt lịch khám.

“Quản lý hồ sơ” để tạo hồ sơ bệnh nhân.

Giới thiệu ứng dụng giúp người dùng có thể hiểu rõ về các chức năng của ứng dụng.

Hướng dẫn sử dụng giúp người dùng hiểu rõ về quy định và quy trình đăng ký đặt lịch khám.

Đăng xuất tài khoản.



Hình 4.4 Giao diện trang chủ

**- Giao diện giới thiệu ứng dụng**

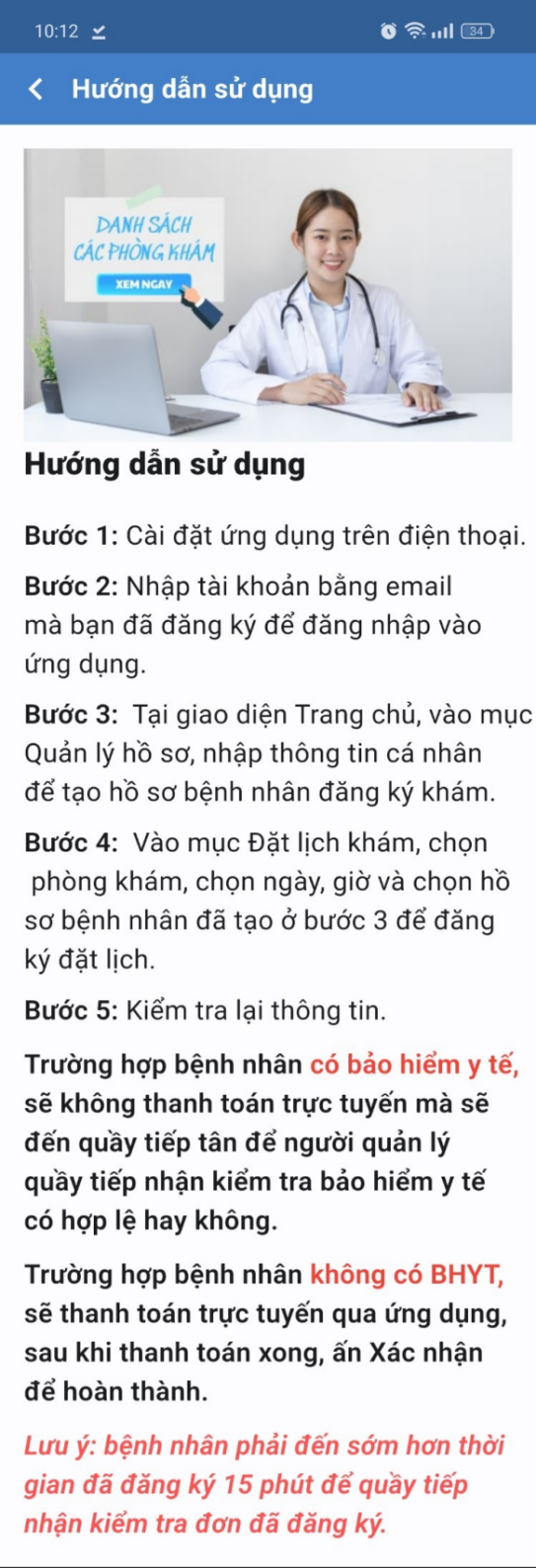
Người dùng sẽ xem được giới thiệu ứng dụng, giới thiệu về các chức năng có trong ứng dụng.



Hình 4.5 Giao diện giới thiệu ứng dụng

**- Giao diện xem hướng dẫn đăng ký đặt lịch khám**

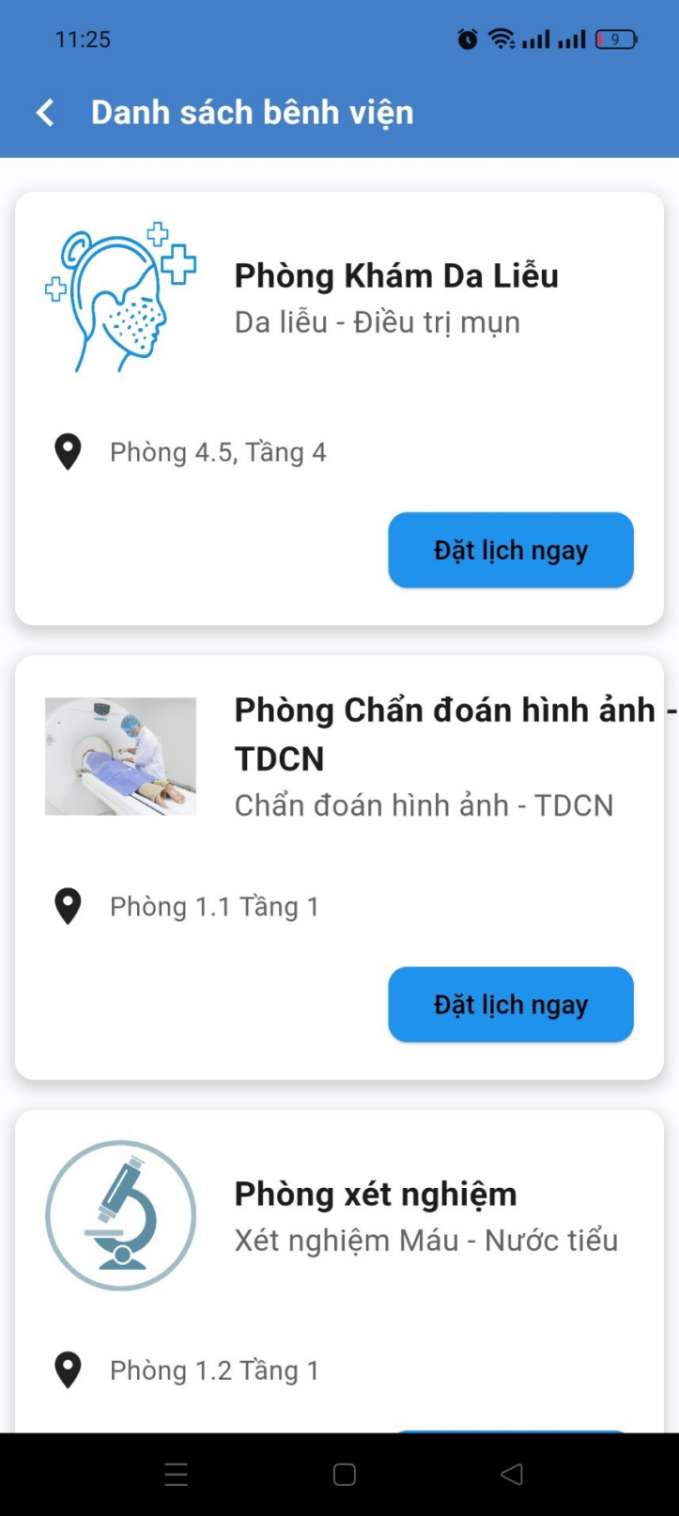
Người dùng xem được hướng dẫn sử dụng để hiểu rõ về quy định và quy trình đăng ký đặt lịch.



Hình 4.6 Giao diện hướng dẫn đăng ký đặt lịch khám

**- Giao diện xem danh sách các phòng khám và chọn phòng khám**

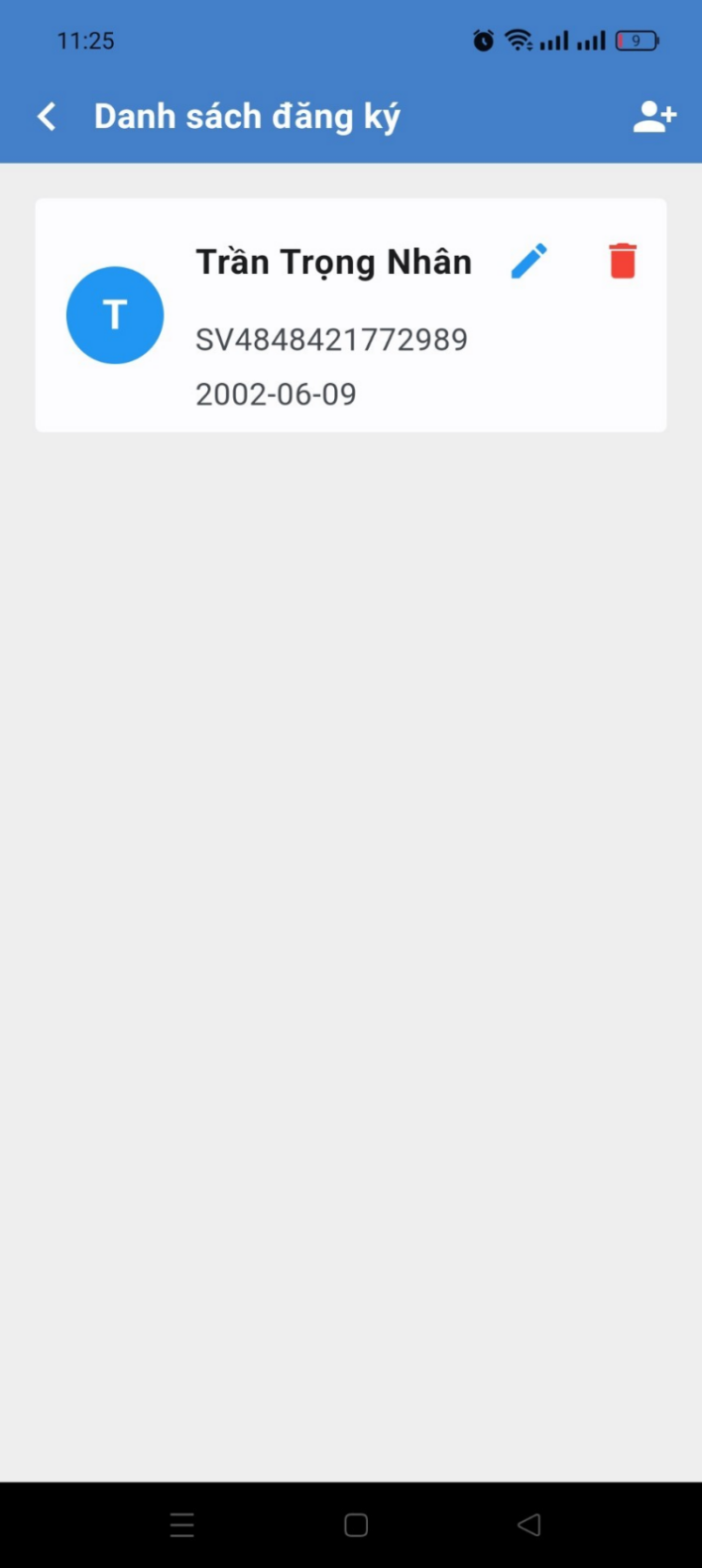
Ứng dụng hiển thị danh sách các phòng khám và các thông tin của phòng khám đó như: tên phòng khám, chuyên khoa, địa chỉ của phòng khám. Người dùng ấn vào nút “Đặt lịch ngay” để đăng ký đặt lịch khám.



Hình 4.7 Giao diện xem danh sách các phòng khám và chọn phòng khám

**- Giao diện xem hồ sơ đăng ký bệnh nhân**

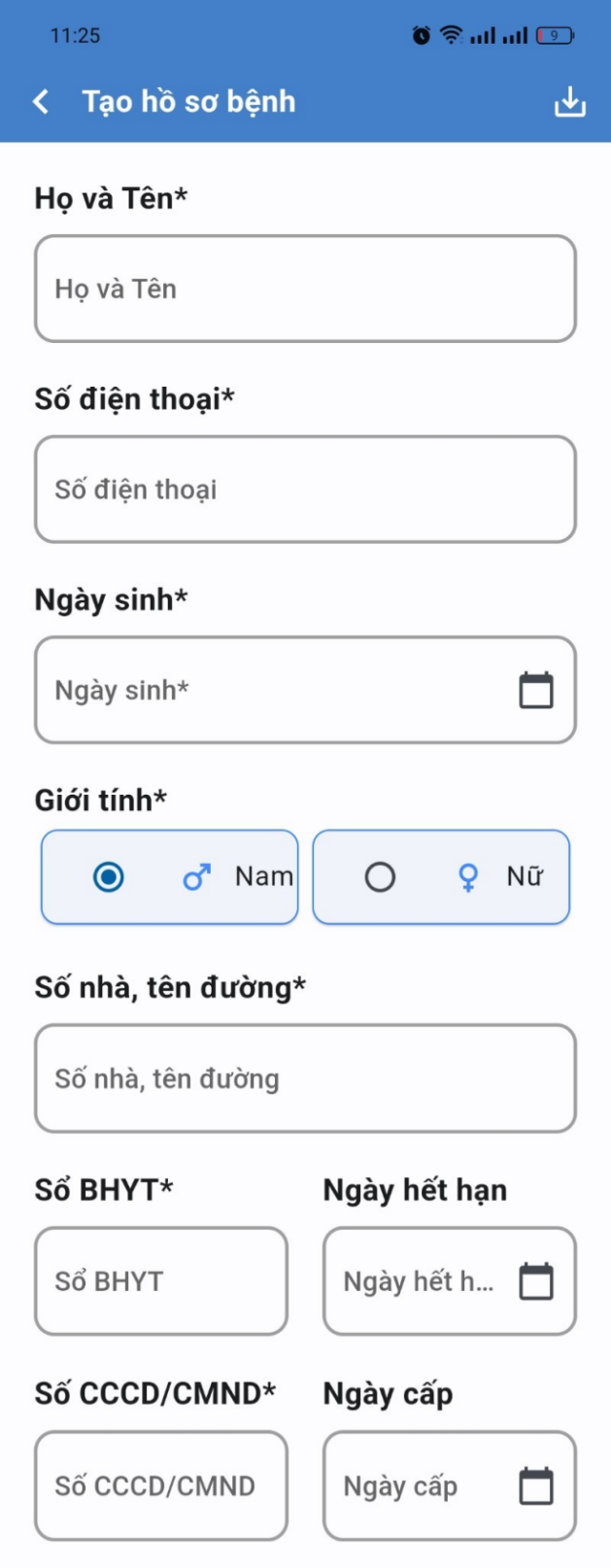
Hiển thị danh sách các hồ sơ bệnh nhân mà người dùng đã đăng ký. Tại giao diện này người dùng có thể xem chi tiết, thêm hồ sơ bệnh nhân, chỉnh sửa và xóa hồ sơ.



Hình 4.8 Giao diện xem hồ sơ đăng ký bệnh nhân

**- Giao diện tạo mới hồ sơ bệnh nhân**

Người dùng nhập các thông tin vào để tạo hồ sơ bệnh nhân bao gồm: họ và tên, số điện thoại, ngày sinh, giới tính, địa chỉ, số bảo hiểm y tế, số CCCD/CMND.

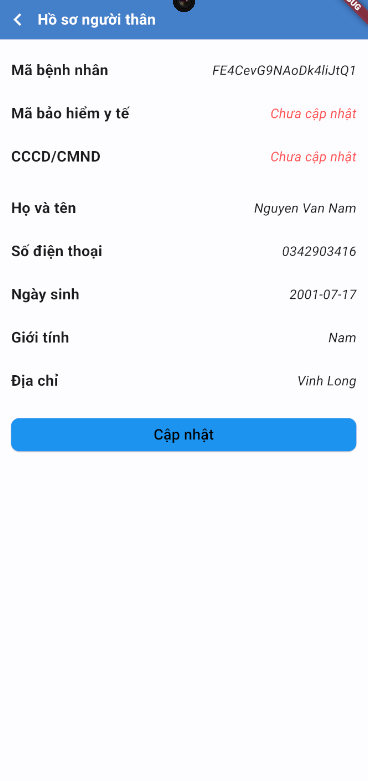


Hình 4.9 Giao diện tạo mới hồ sơ bệnh nhân

**- Giao diện xem chi tiết hồ sơ bệnh nhân**

Ứng dụng sẽ hiển thị các thông tin bao gồm: mã bệnh nhân, tên bệnh nhân, số điện thoại, giới tính, ngày sinh, địa chỉ, số bảo hiểm y tế và căn cước công dân.

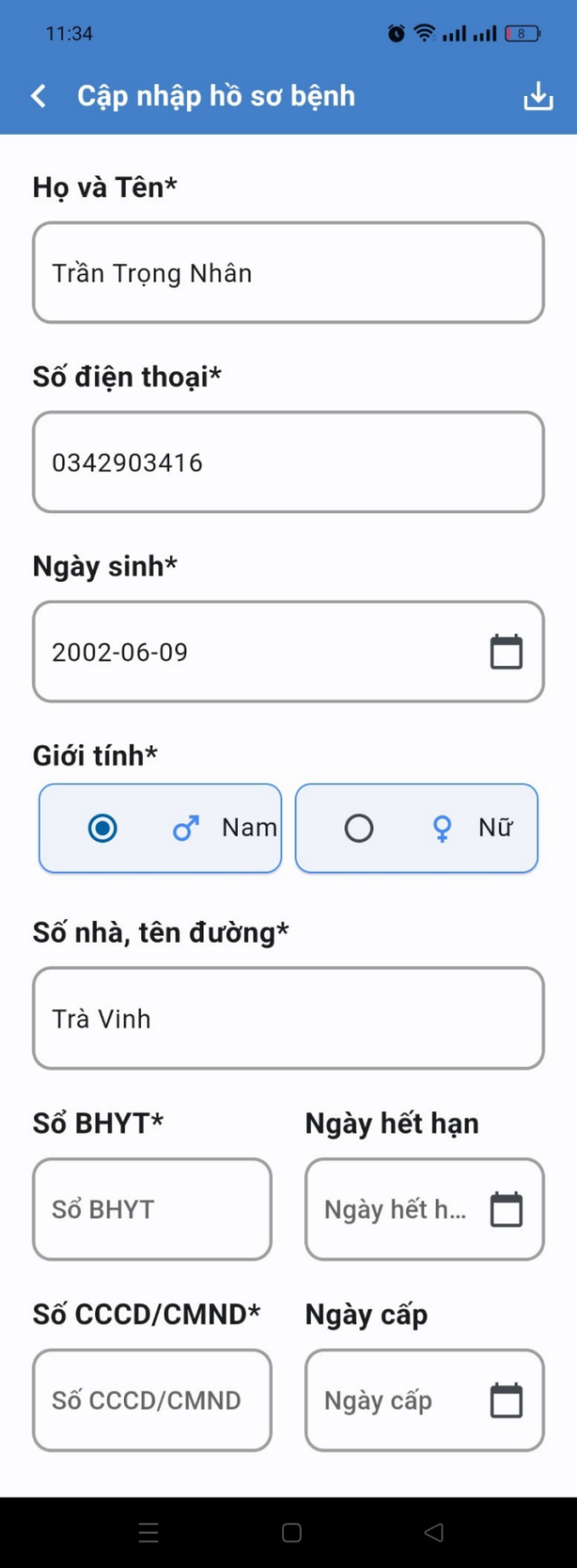
Khi ấn vào nút Cập nhật, người dùng được sẽ được chuyển đến giao diện chỉnh sửa thông tin hồ sơ.



Hình 4.10 Giao diện xem chi tiết hồ sơ bệnh nhân

**- Giao diện chỉnh sửa thông tin hồ sơ bệnh nhân**

Người dùng có thể chỉnh sửa các thông tin nếu nhập sai hoặc có nhu cầu sửa chữa.



Hình 4.11 Giao diện chỉnh sửa thông tin hồ sơ bệnh nhân

**- Giao diện xóa hồ sơ bệnh nhân**

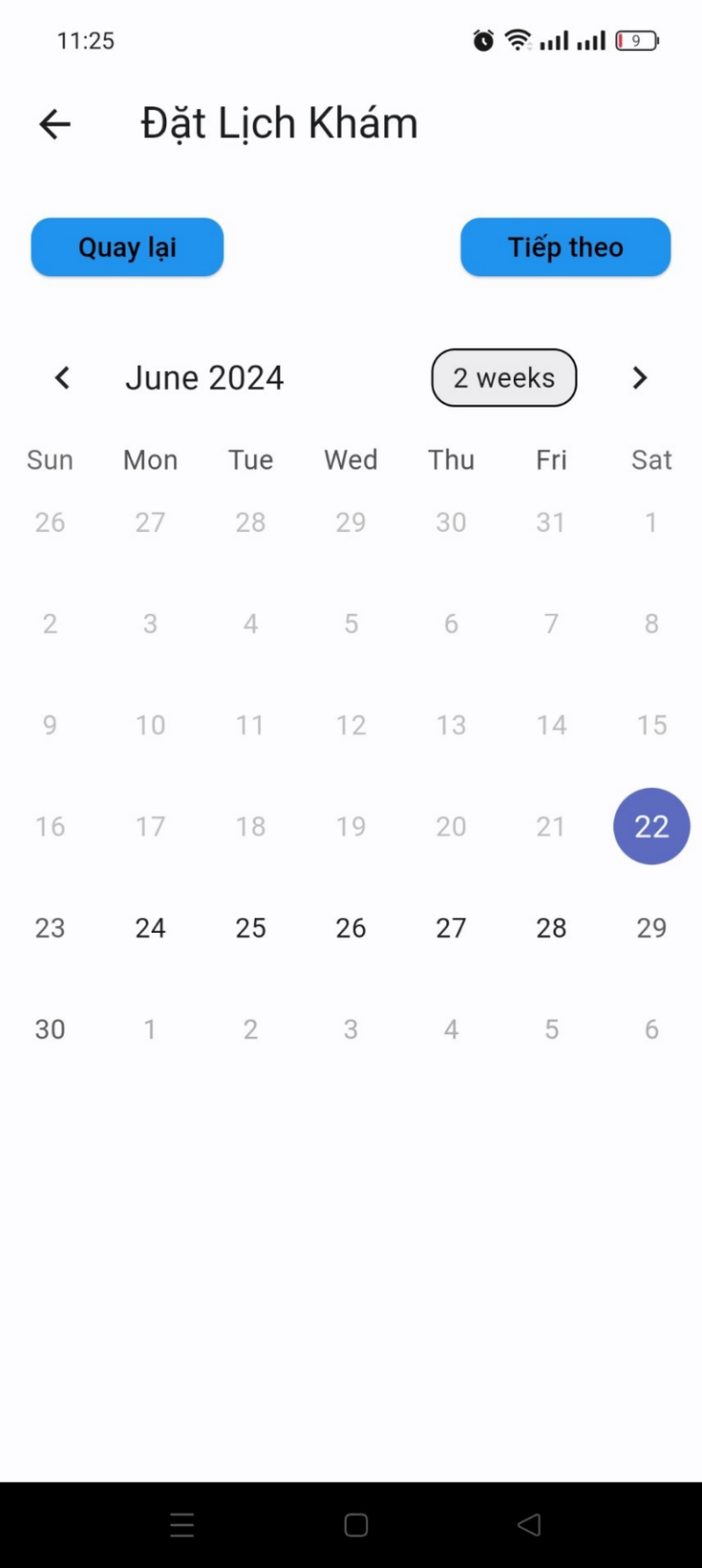
Người dùng có thể xóa hồ sơ bệnh nhân bằng cách ấn vào biểu tượng thùng rác màu đỏ.



Hình 4.12 Giao diện xóa hồ sơ bệnh nhân

**- Giao diện Chọn ngày khám**

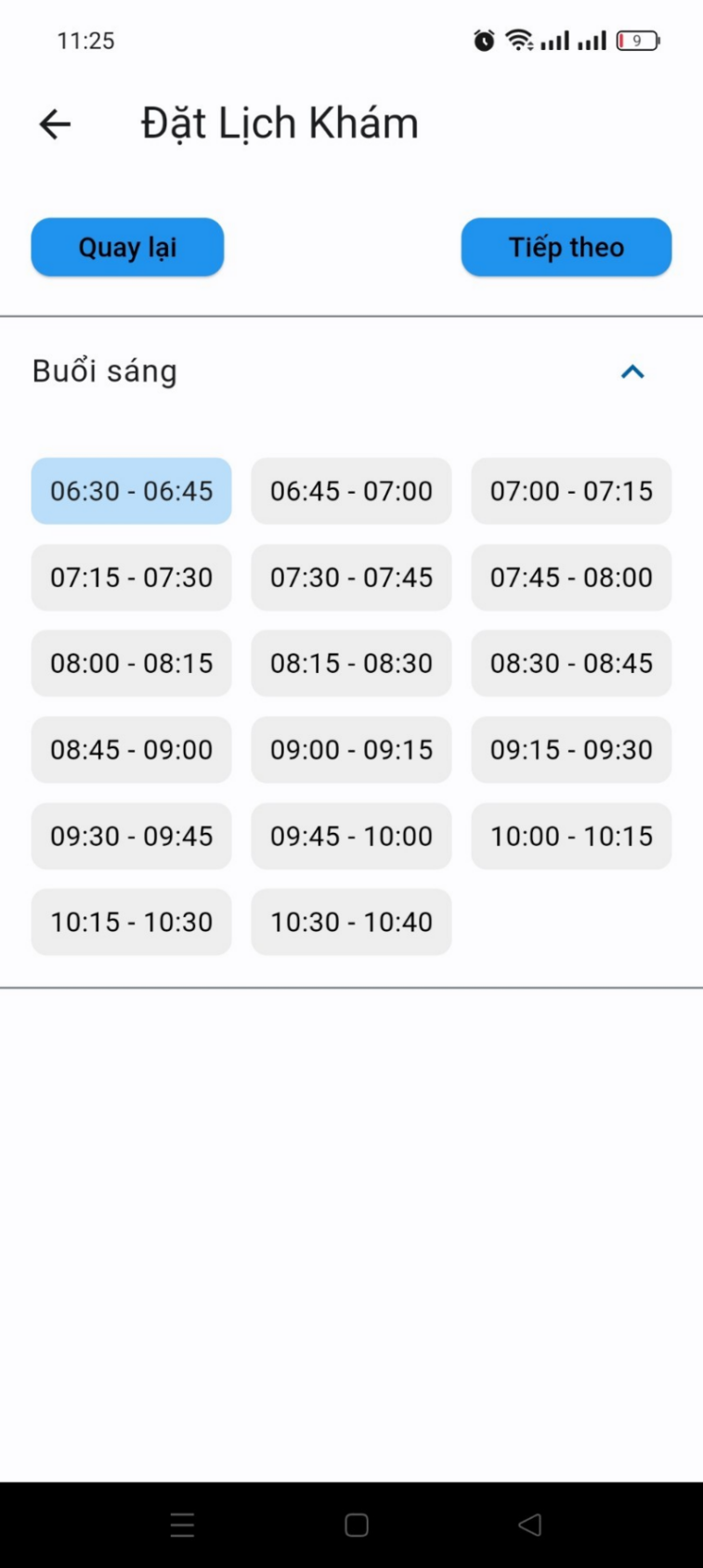
Người dùng có thể chọn ngày khám bằng cách ấn vào ngày mà mình muốn đăng ký đặt khám, ngày dùng chỉ được chọn ngày trong tương lai và cách ngày hiện tại ít nhất 1 ngày.



Hình 4.13 Giao diện chọn ngày khám

**- Giao diện chọn khung giờ khám**

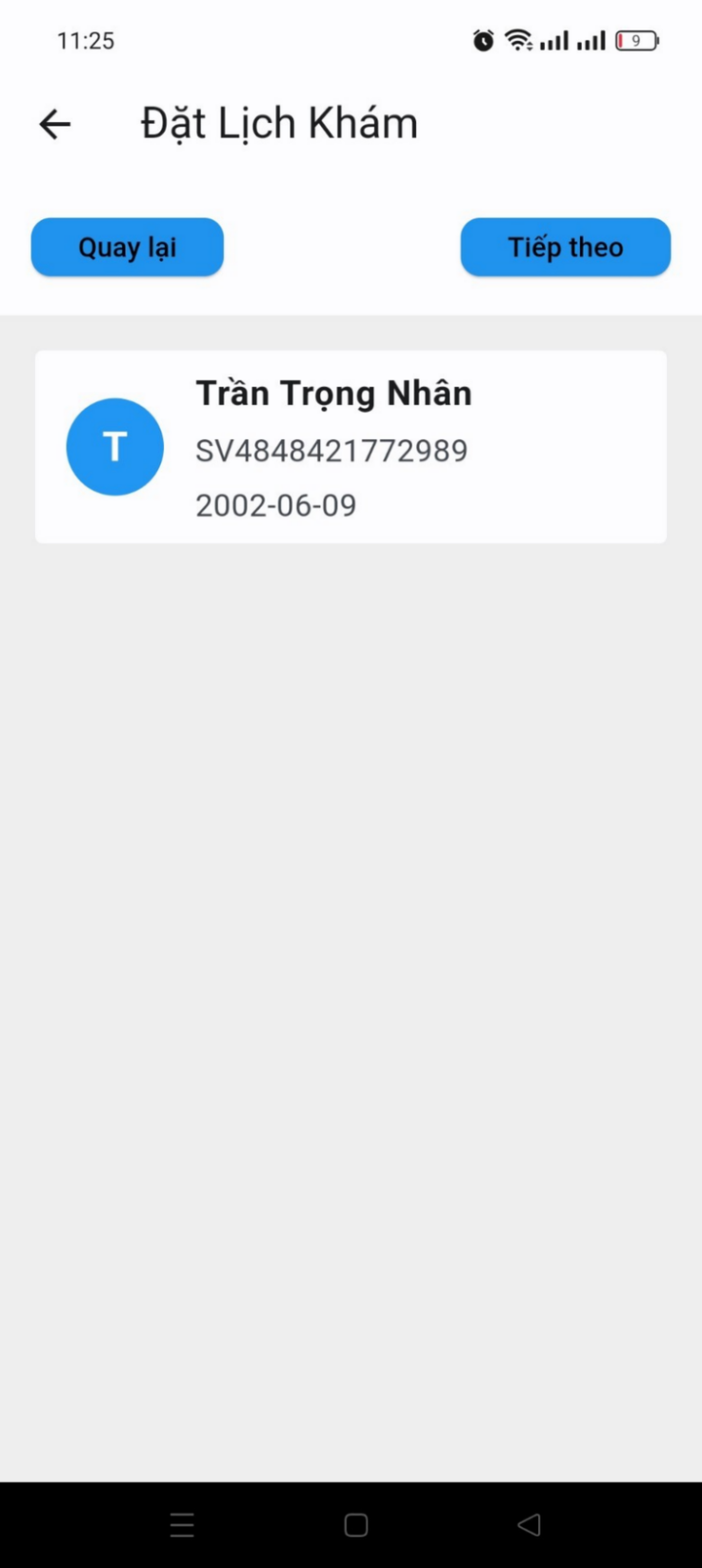
Người dùng có thể chọn khung giờ mà mình muốn đăng ký. Người dùng sẽ không đặt được những khung giờ mà người khác đã đặt.



Hình 4.14 Giao diện chọn khung giờ khám

**- Giao diện chọn hồ sơ khám**

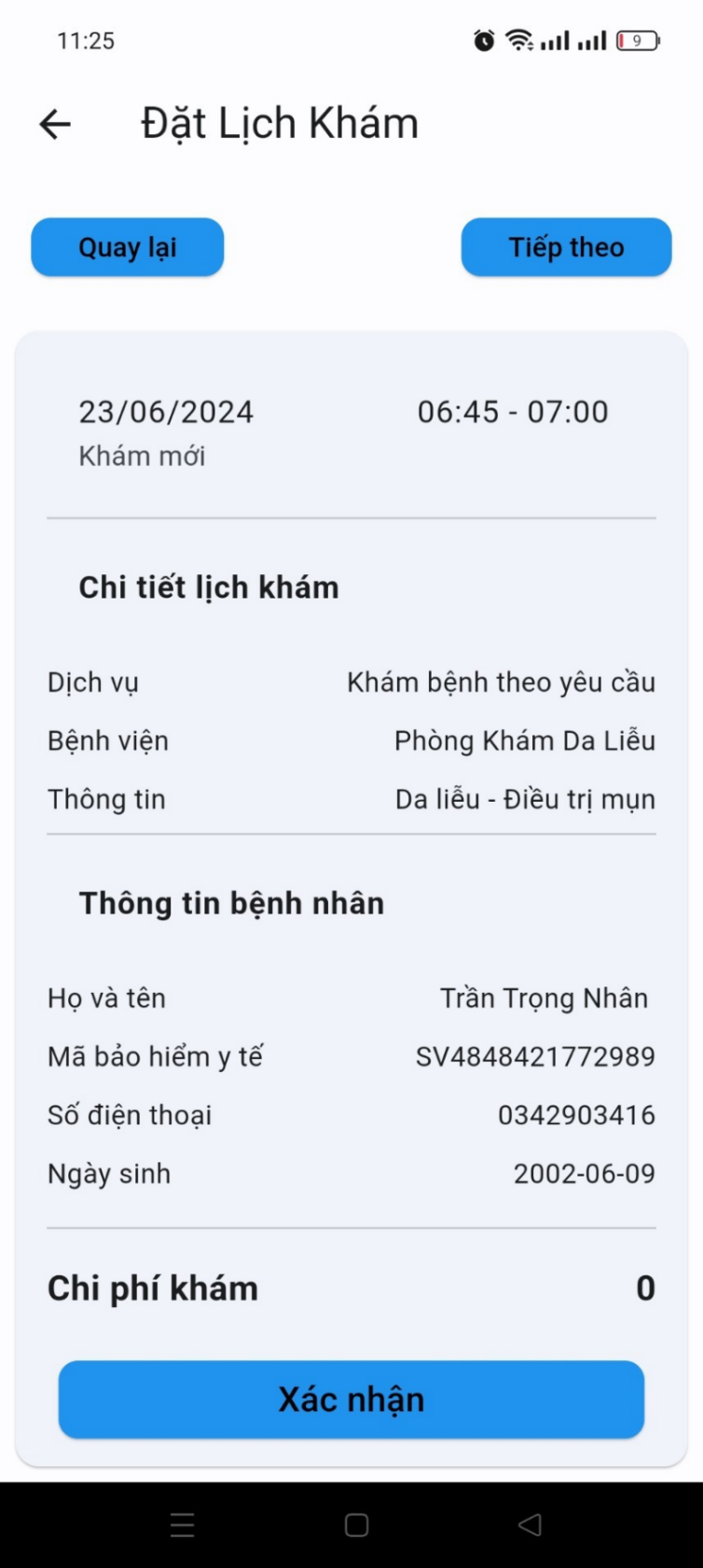
Người dùng sẽ chọn 1 hồ sơ bệnh nhân đã tạo trước đó để đăng ký khám.



Hình 4.15 Giao diện chọn hồ sơ khám

**- Giao diện kiểm tra và xác nhận thông tin khi bệnh nhân có BHYT**

Giao diện sẽ hiển thị các thông tin về ngày, giờ đăng ký, thông tin về bệnh nhân, phòng khám mà bệnh nhân đã đăng ký.



Hình 4.16 Giao diện kiểm tra và xác nhận thông tin khi bệnh nhân có BHYT

**- Giao diện kiểm tra và xác nhận thông tin khi bệnh nhân không có BHYT**

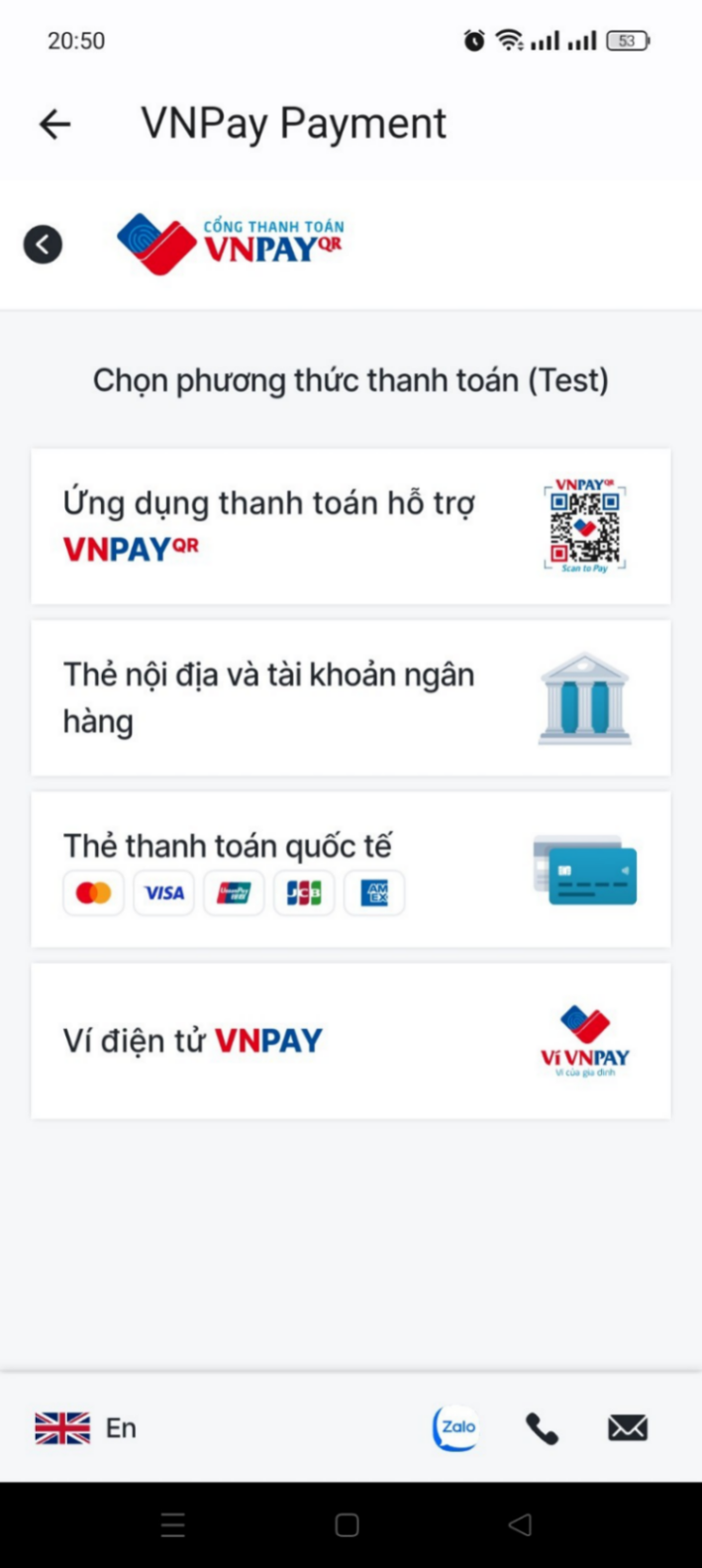
Người dùng kiểm tra thông tin về phòng khám, ngày, giờ khám khi đăng ký đặt lịch và ấn xác nhận để đăng ký đặt lịch, người dùng sẽ phải thanh toán trực tuyến thông qua cổng thanh toán VNpay.



Hình 4.17 Giao diện kiểm tra và xác nhận thông tin khi bệnh nhân không có BHYT

**- Giao diện Chọn phương thức thanh toán VNpay**

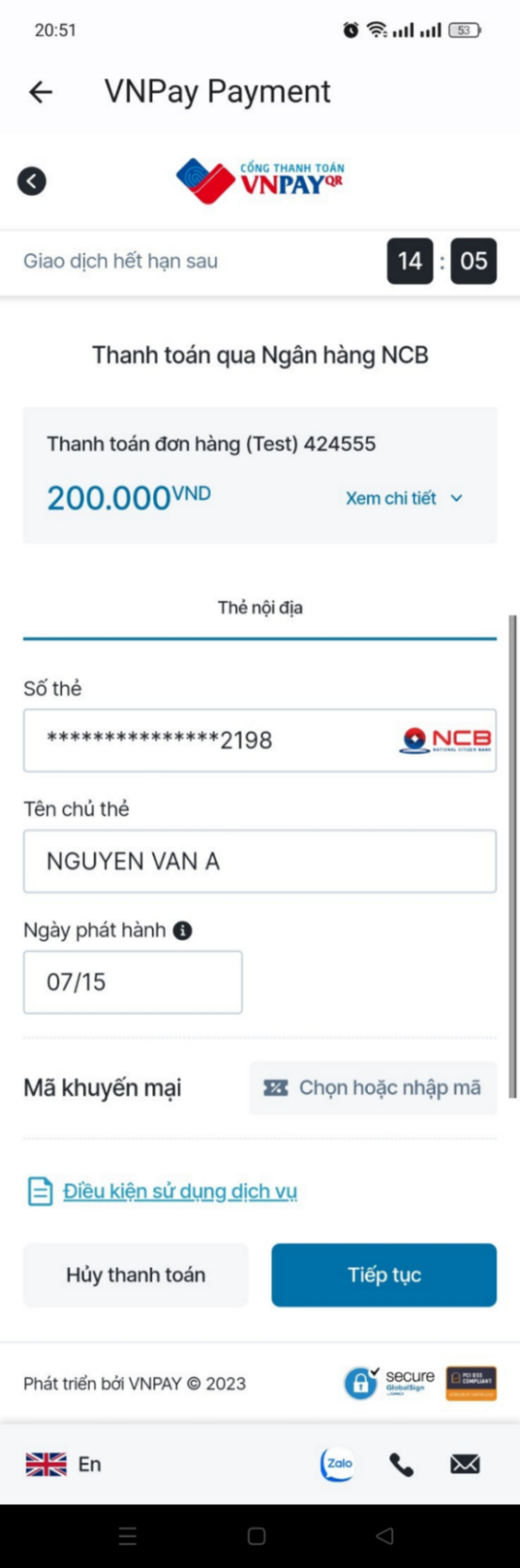
Người dùng được chọn các phương thức thanh toán để thanh toán phí đặt khám bệnh như Quét mã QR, thanh toán bằng ví điện tử, thanh toán bẳng thẻ nội địa, tài khoản ngân hàng và thẻ thanh toán quốc tế.



Hình 4.18 Giao diện chọn phương thức thanh toán VNpay

**- Giao diện thanh toán VNpay bằng thẻ ngân hàng**

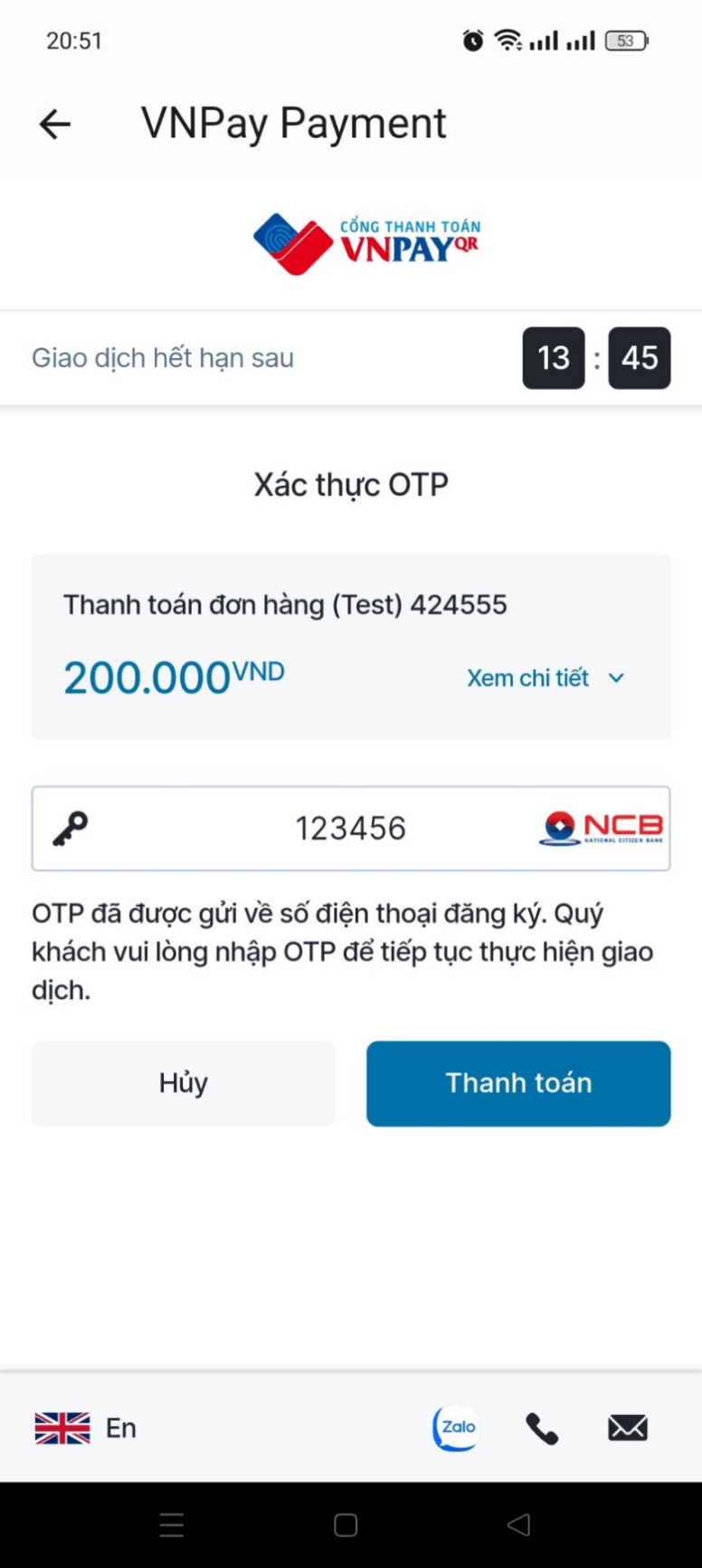
Người dùng nhập thông tin thẻ và tên tài khoản, ngày phát hành để tiến hành thanh toán đơn đặt lịch.



Hình 4.19 Giao diện thanh toán VNpay bằng thẻ ngân hàng

**- Giao diện xác thực mã OTP**

Người dùng nhập mã xác thực để thanh toán.



Hình 4.20 Giao diện xác thực mã OTP

**- Giao diện xác nhận lại thông tin**

Người dùng kiểm tra lại thông tin và ấn xác nhận để đặt lịch.

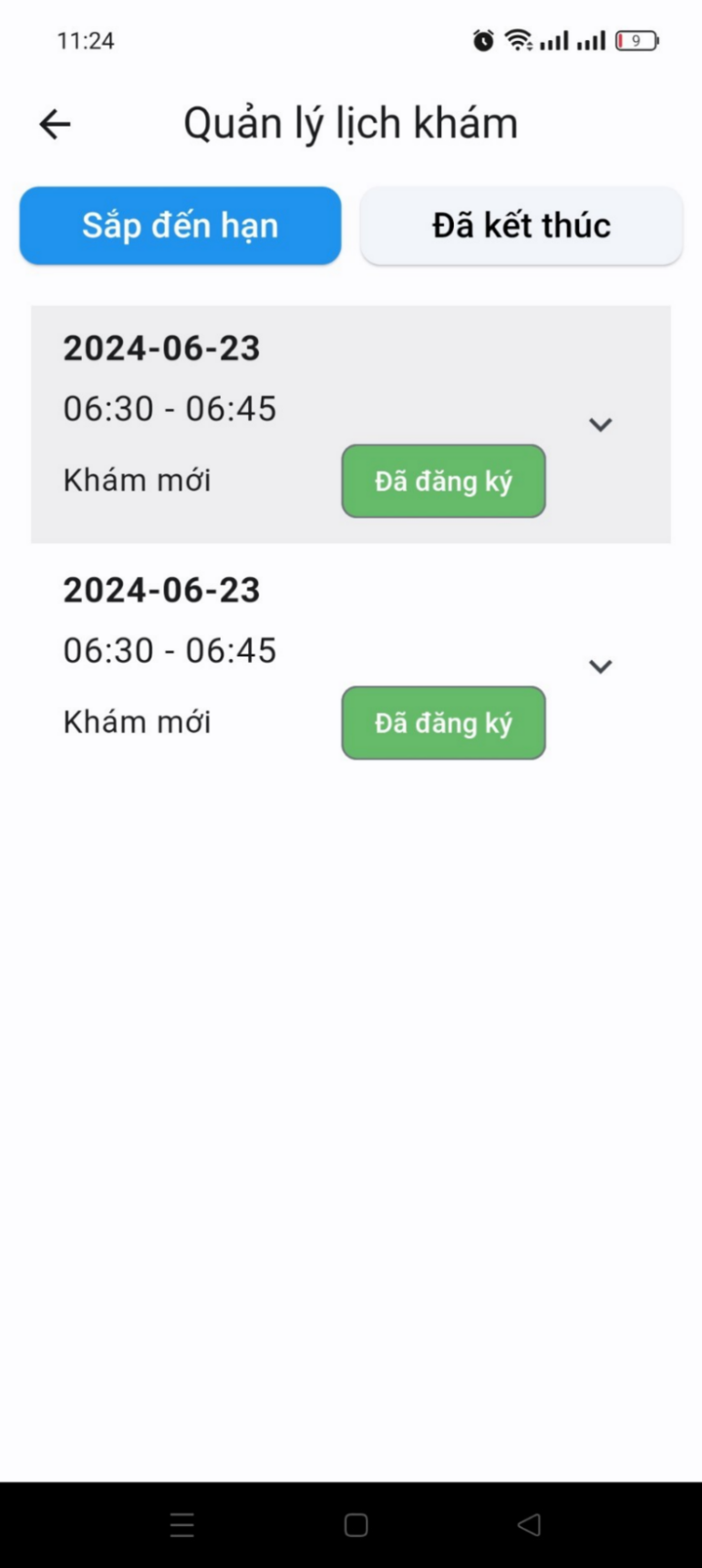


Hình 4.21 Giao diện xác nhận lại thông tin

**- Giao diện xem danh sách đặt lịch**

Người dùng có thể xem danh sách đặt lịch mà mình đã đăng ký đặt.

Để xem chi tiết đơn đặt lịch đã đăng ký, người dùng ấn vào đơn đặt lịch đó.



Hình 4.22 Giao diện xem các cuộc hẹn sắp diễn ra

**- Giao diện xem chi tiết cuộc hẹn**

Người dùng có thể xem được chi tiết cuộc hẹn đã đăng ký bao gồm các thông tin về thời gian đã đặt, thông tin về phòng khám, thông tin về bệnh nhân và những điều cần lưu ý.

Tại giao diện này, người dùng có thể hủy đơn đặt lịch này bằng cách ấn vào nút “Hủy đăng ký” màu đỏ ở phía dưới.

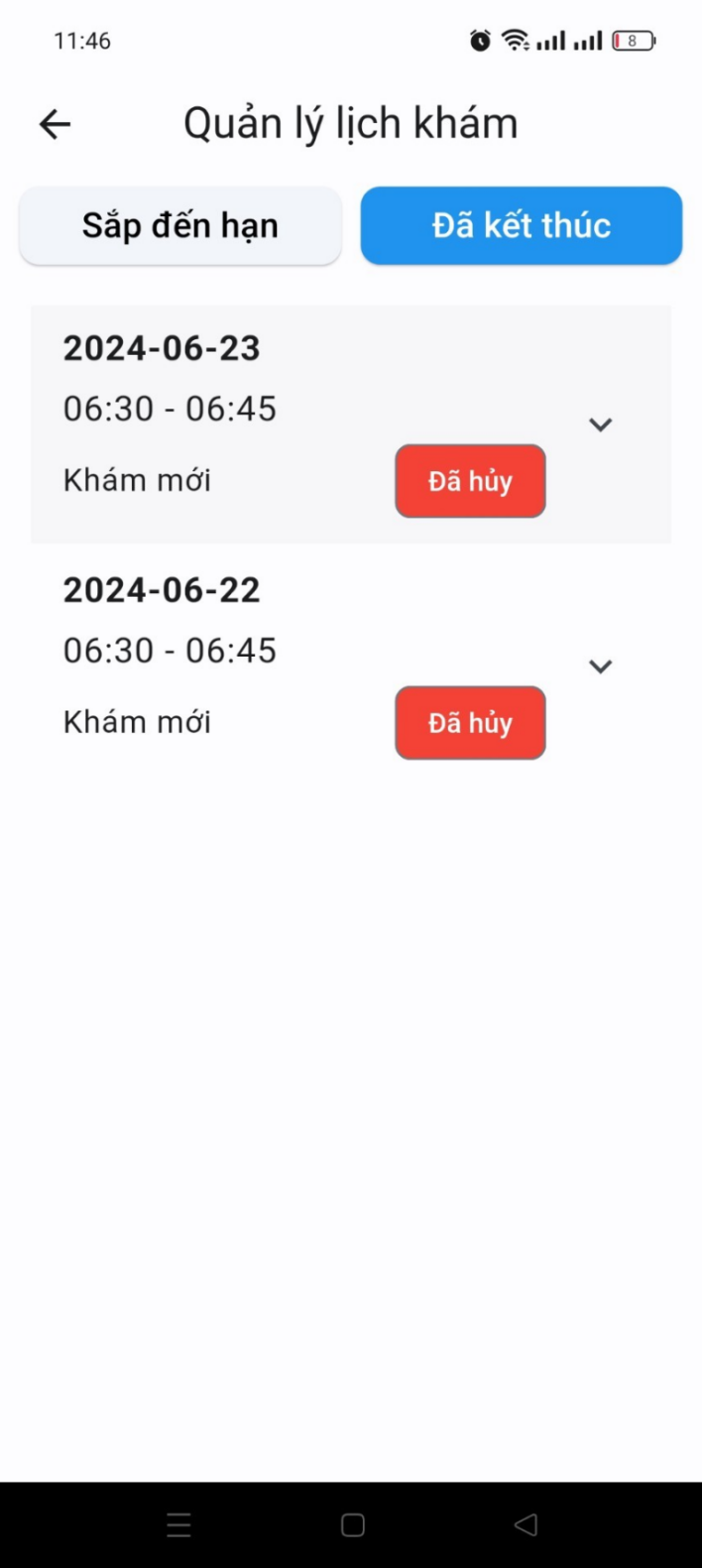


Hình 4.23 Giao diện xem chi tiết cuộc hẹn

**- Giao diện xem các cuộc hẹn đã hủy**

Tại giao diện này, người dùng có thể xem được các cuộc hẹn đã bị hủy hoặc do người dùng hủy.

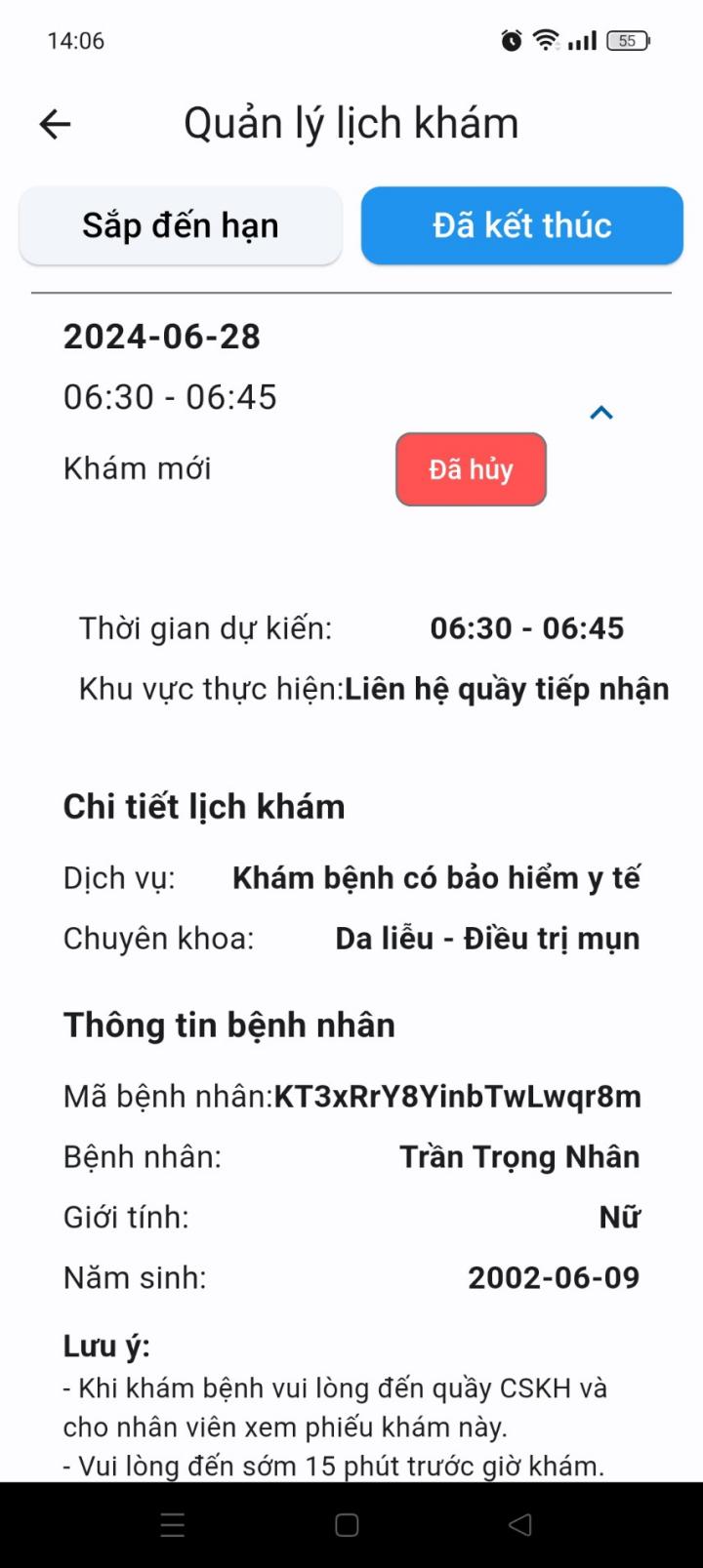
Để xem chi tiết, người dùng ấn vào đơn bị hủy đó, sẽ hiển thị được các thông tin của đơn đăng ký.



Hình 4.24 Giao diện xem các cuộc hẹn đã hủy

**- Giao diện xem chi tiết cuộc hẹn đã hủy**

Tại giao diện này, người dùng có thể xem được các cuộc hẹn đã bị hủy hoặc do người dùng hủy.



Hình 4.25 Giao diện xem chi tiết cuộc hẹn đã hủy

## Giao diện bác sĩ và chức năng

**- Giao diện Xem lịch đăng ký đặt khám của bác sĩ**

Bác sĩ có thể xem được lịch đăng ký hợp lệ của bệnh nhân bao gồm mã bệnh nhân, ngày đăng ký, giờ đăng ký, tên bệnh nhân, trạng thái thanh toán của cuộc hẹn đó.



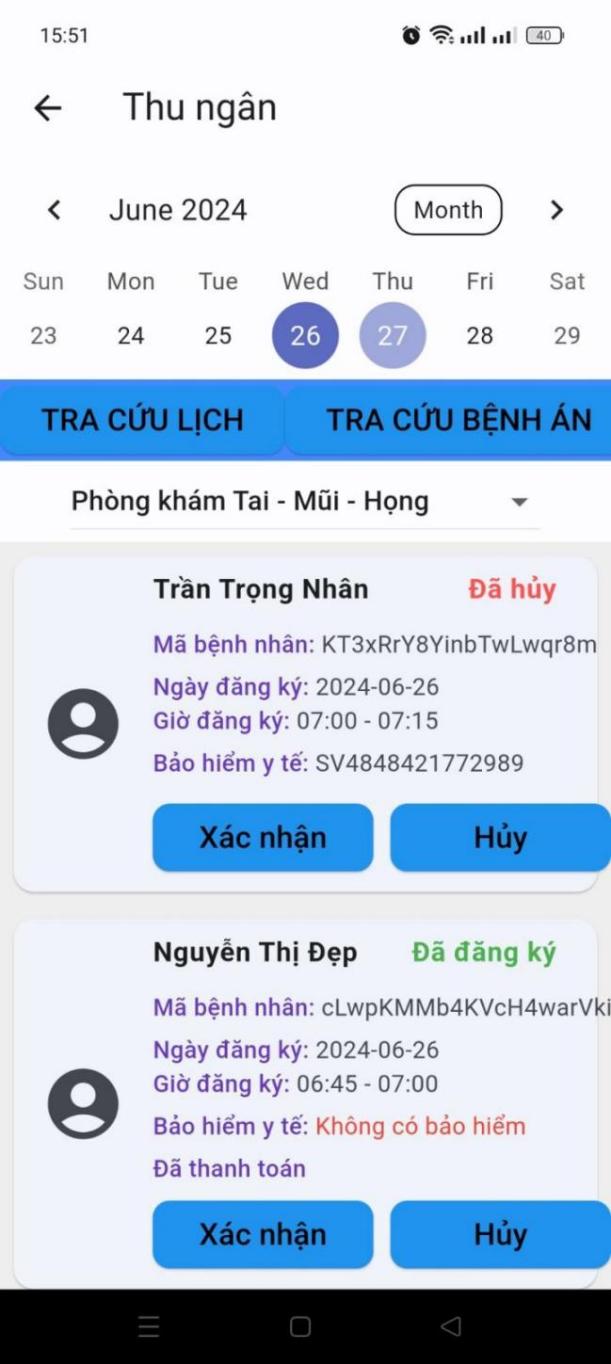
Hình 4.26 Giao diện xem lịch đặt khám

## Giao diện quầy tiếp nhận và chức năng

**- Giao diện xem và xác nhận các cuộc hẹn của quầy tiếp nhận**

Người quản lý quầy tiếp nhận có thể xác nhận các đơn đăng ký của bệnh nhân hoặc hủy các đơn đăng ký không phù hợp.

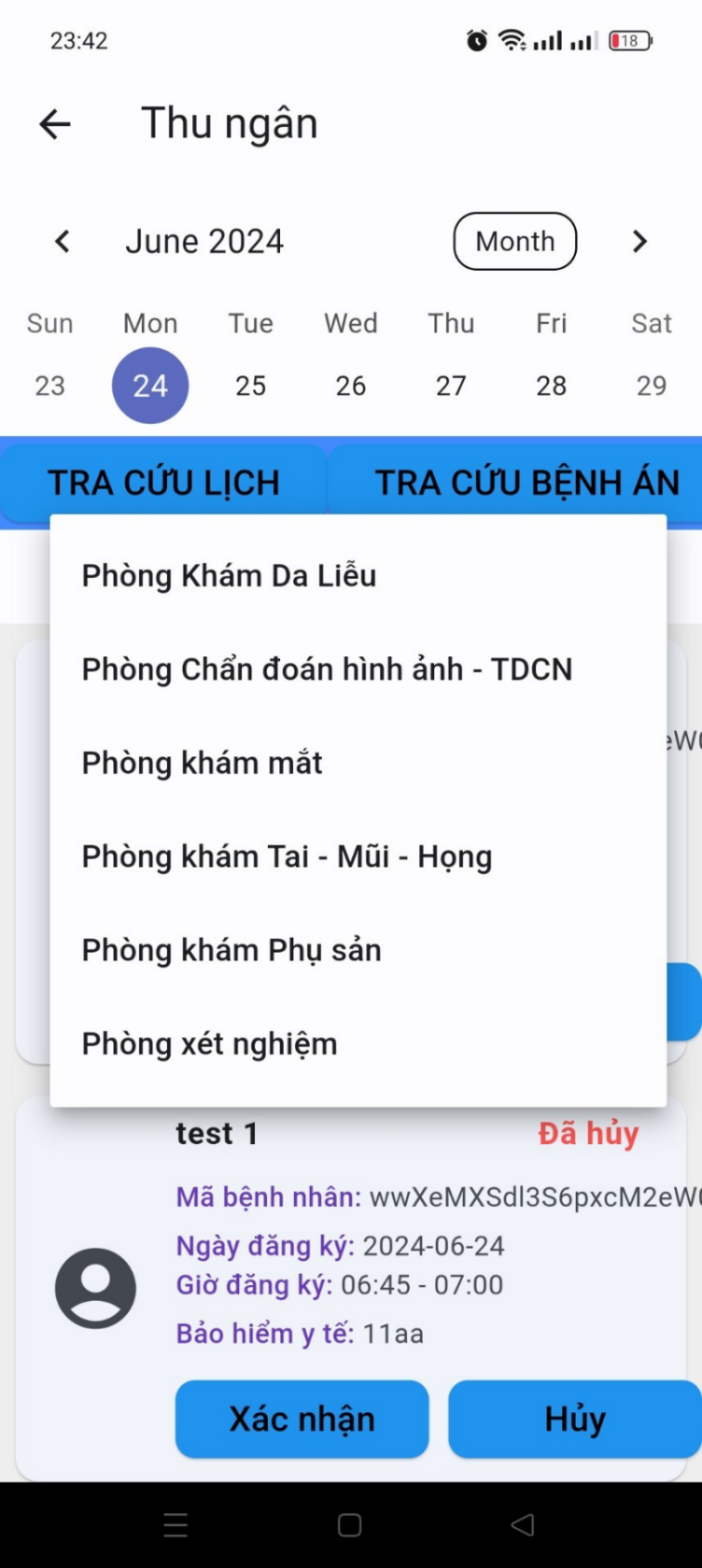
Nếu đơn đăng ký hợp lệ, người quản lý sẽ ấn Xác nhận, thông tin của đơn đăng ký này sẽ được gửi đến giao diện của bác sĩ trực thuộc phòng khám đó. Nếu ấn vào nút Hủy, đơn đăng ký sẽ được chuyển vào giao diện những cuộc hẹn bị hủy của người dùng.



Hình 4.27 Giao diện xem và xác nhận các cuộc hẹn của quầy tiếp nhận

**- Giao diện chọn phòng khám để tra cứu của quầy tiếp nhận**

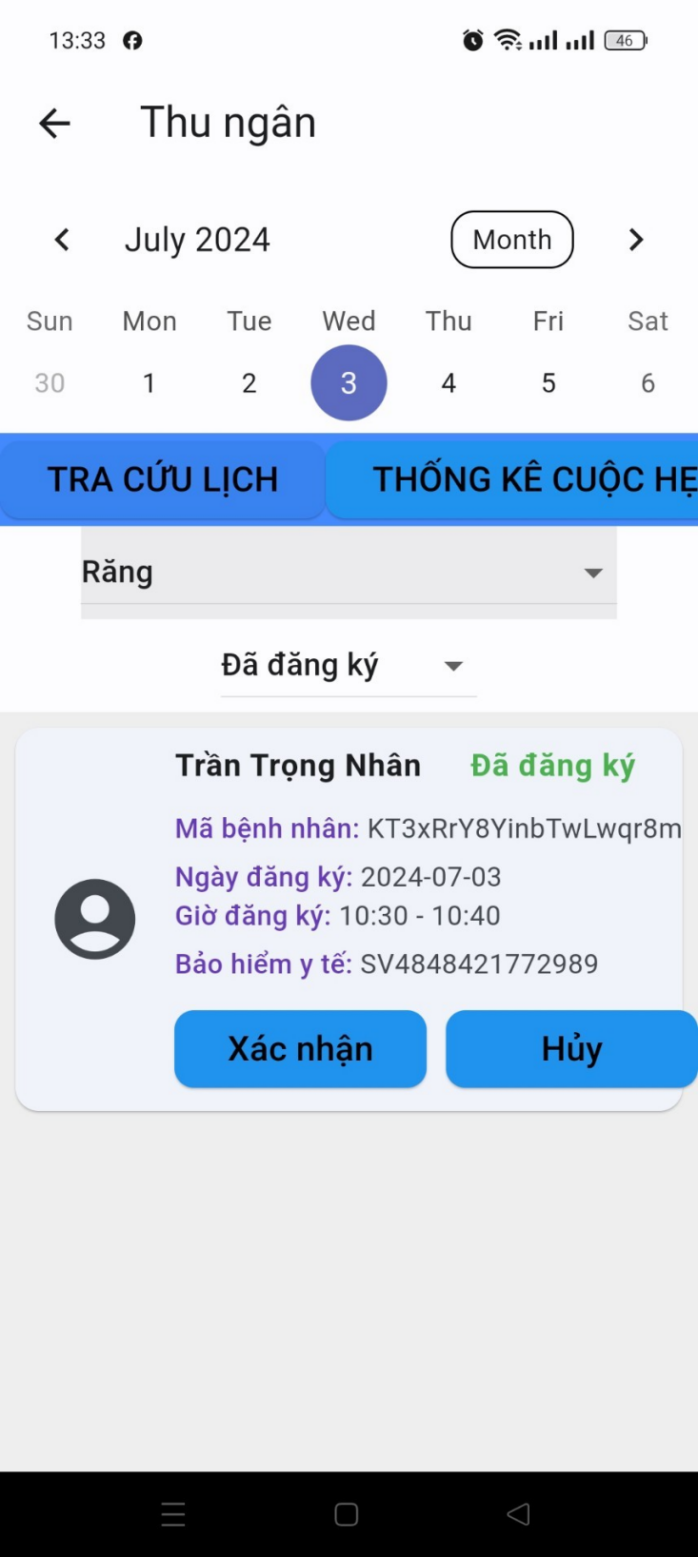
Người quản lý quầy tiếp nhận có thể tra cứu các đơn đăng ký của bệnh nhận theo từng phòng khám.



Hình 4.28 Giao diện chọn phòng khám để tra cứu của quầy tiếp nhận

**- Giao diện xem thống kê phòng khám và trạng thái đơn đăng ký để thống kê của quầy tiếp nhận**

Người quản lý quầy tiếp nhận có thể thống kê các đơn đăng ký của từng phòng khám theo các trạng thái của đơn.

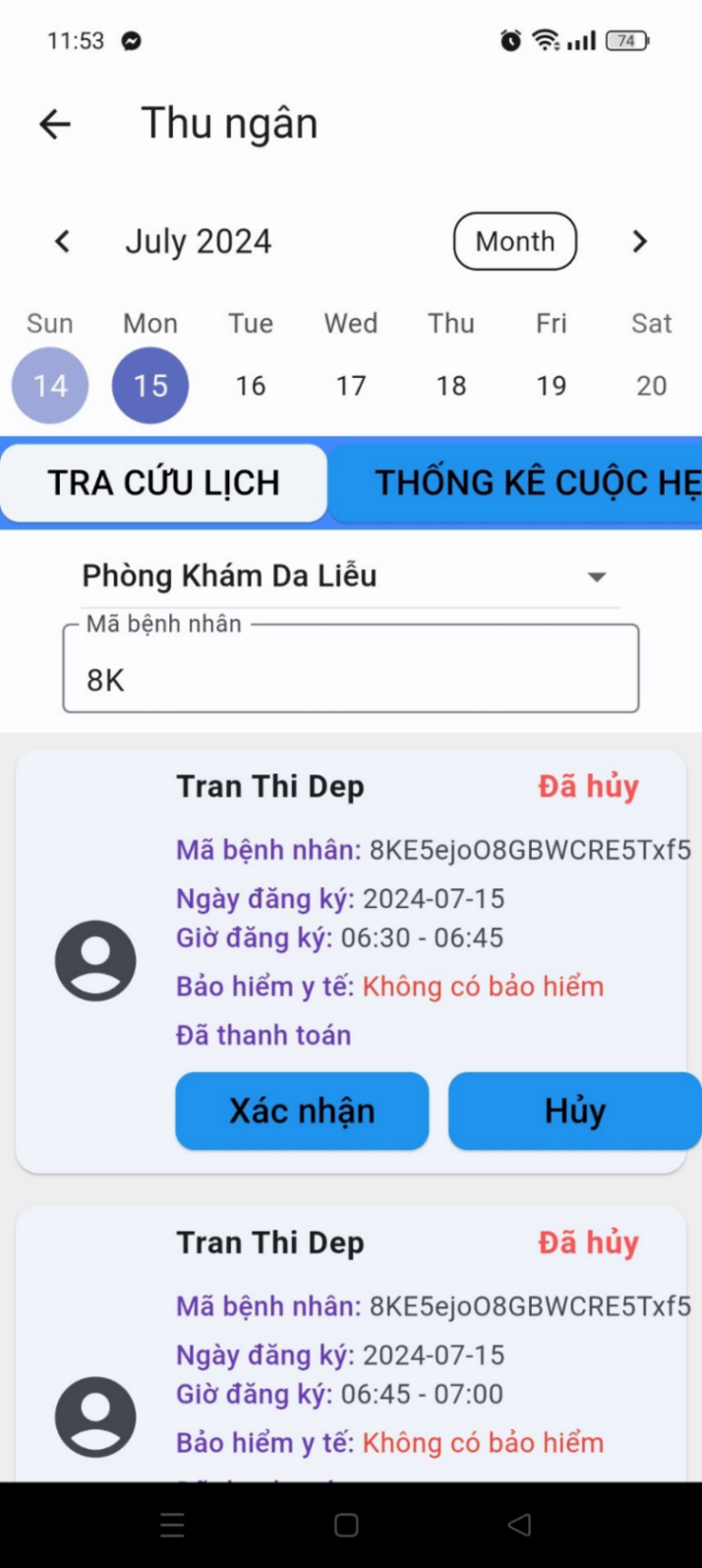


Hình 4.29 Giao diện xem thống kê phòng khám và trạng thái đơn đăng ký để thống kê của quầy tiếp nhận

**- Giao diện tìm kiếm mã bệnh nhân**

Người quản lý quầy tiếp nhận có thể tìm kiếm mã bệnh nhân các đơn đăng ký của từng phòng khám.

Tại đây, người quản lý chỉ cần nhập một vài chữ cái trong mã bệnh nhân, ứng dụng sẽ lập tức hiển thị những đơn đăng ký có mã bệnh nhân giống những chữ cái đó.



Hình 4.30 Giao diện tìm kiếm mã bệnh nhân

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

**Kết quả đạt được**

- Thiết kế được cơ sở dữ liệu tương đối hoàn chỉnh, đáp ứng hiệu quả cho việc thiết kế phần mềm.

- Hoàn thành được chức năng cơ bản như xem, thêm, xóa, sửa thông tin hồ sơ bệnh nhân.

- Có thể xem và chọn phòng khám, đăng ký đặt lịch khám bệnh, quản lý danh sách lịch khám.

- Phân quyền bác sĩ và quản lý quầy tiếp nhận cung cấp khả năng tổ chức linh hoạt, giúp dễ dàng kiểm soát và hoạt động đúng quy trình làm việc.

- Thống kê được đơn đăng ký, đơn đang chờ xác nhận và đơn đã hủy.

- Giao diện ứng dụng thân thiện, dễ sử dụng.

**Ưu điểm**

- Sử dụng đa dạng phương thức thanh toán trực tuyến.

- Ứng dụng hoạt động ổn định, xử lý nhanh chóng các tác vụ và đảm bảo tính bảo mật thông tin cho cả người dùng và doanh nghiệp.

**Hạn chế**

- Ứng dụng chưa có các tính năng nâng cao như thông báo cho người dùng.

- Mã bệnh nhân còn phức tạp.

## Hướng phát triển

Thời gian sắp tới, tôi sẽ giành nhiều thời để tiếp tục nghiên cứu sâu hơn về Flutter Framework cũng như cố gắng nghiên cứu và hiểu những kiến thức nâng cao về Flutter và những thành phần mở rộng của Flutter. Sau đó sẽ vận dụng những gì nghiên cứu để ứng dụng xây dựng hoàn chỉnh ứng dụng.

Nếu điều kiện cho phép, tôi sẽ tiếp tục nghiên cứu cũng như tìm hiểu chi tiết hơn về quy trình đăng ký đặt lịch khám bệnh cũng như cách hình thành ứng dụng, tìm hiểu những kiến thức mới góp phần cải tiến ứng dụng hoàn chỉnh. Mục tiêu là không ngừng cải tiến ứng dụng của mình, đưa vào hoạt động những chức năng mới và tối ưu hóa hiệu suất để mang lại trải nghiệm tốt nhất cho người sử dụng.

Tôi tin rằng với sự nỗ lực và kiên trì, cùng với việc không ngừng học hỏi và áp dụng những kiến thức mới, tôi sẽ đạt được mục tiêu xây dựng một ứng dụng đăng ký đặt lịch khám bệnh chuyên nghiệp, hiệu quả và mang lại giá trị thực sự cho người tiêu dùng.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Bệnh viện Đại học Y dược TP.HCM (2018). Quy Trình Đăng Ký Khám Bệnh Theo Hẹn, trang web: https://umc-v1.medpro.vn/quytrinh, ngày truy cập: 10/5/2024.

[2] FPT CLOUD (2022). Mô hình MVC, trang web: https://fptcloud.com/mvc-la-gi/, ngày truy cập: 10/12/2024.

[3] Flutter (2023). Flutter, trang web: https://flutter.dev/, ngày truy cập: 14/5/2024.

[4] Nguyễn Hà Giang (2018). *Lập trình trên thiết bị di động*. Giáo trình giảng dạy, Trường Đại học Công nghệ TP.HCM HUTECH.

[5] Nguyễn Hoàng Duy Thiện (2013). *Tài liệu giảng dạy môn Lập trình thiết bị di động.* Trường Đại học Trà Vinh.

[6] Ngô Bá Hùng và Đoàn Hòa Minh (2016). *Giáo trình Lập trình cho thiết bị di động*. Giáo trình giảng dạy, Trường Đại học Cần Thơ.

[7] Nguyên (2023). Dart là gì? Giới thiệu cơ bản về ngôn ngữ lập trình Dart, trang web: https://200lab.io/blog/tu-hoc-ngon-dart-nhung-dieu-can-biet-truoc-khi-bat-dau/, ngày truy cập: 1/5/2023.

[8] Phạm Xuân Nam (2021). Firebase là gì?, trang web: https://viblo.asia/p/tim-hieu-so-luoc-ve-firebase-Eb85oeOmZ2G, ngày truy cập: 20/5/2024.

[9] Trần Hữu Cảnh và Nguyễn Hoàng Vũ (2020). *Website đặt lịch khám bệnh online*. Đồ án tốt nghiệp, Trường Đại học Quốc gia TP. Hồ Chí Minh.

[10] Trần Minh Quân, Hoàng Phước Thành, Nguyễn Phạm Minh Quân và Lê Nguyễn Hữu Quốc (2023). Hệ thống quản lý đặt lịch và khám bệnh cho bệnh viện (HABS), trang web: https://fpt.edu.vn/tin-tuc/fpt-edu-tin-tuc-chung/do-an-cua-sinh-vien-dai-hoc-fpt-duoc-doanh-nghiep-ho-tro-phat-trien, ngày truy cập: 15/06/2024.

[11] Trần Phạm Hồng Trúc, Đỗ Kiến Nhuần, Trần Lê Hồng Phúc và Nguyễn Văn Tùng (2020). Dự án HI! MEDICARE, trang web:

https://www.behance.net/gallery/170191041/D-AN-SEM-2App-HI-MEDICARE, ngày truy cập: 16/06/2024.

[12] Trần Phúc Hiếu, Nguyễn Duy Hùng và Nguyễn Quốc Thịnh (2022). *Quản lý hệ thống đặt lịch khám bệnh*. Đồ án tốt nghiệp, Trường Đại học Công nghệ TP. Hồ Chí Minh.