

TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN ĐỀ PYTHON

QUẢN LÝ BỆNH NHÂN

Giảng viên hướng dẫn:

Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện:

Hàng Thái Anh – DTH235604

Lê Thị Ngọc Hân – DTH235640

An Giang, 11/2025

LỜI NÓI ĐẦU

Kính gửi Thầy Nguyễn Ngọc Minh!

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến Thầy vì đã tận tình hướng dẫn và truyền đạt những kiến thức quý báu trong học phần *Chuyên đề Python*. Đây chính là nền tảng quan trọng để nhóm em có thể hoàn thành đồ án “**Quản lý bệnh nhân**” này.

Trong bối cảnh xã hội hiện đại, sức khỏe luôn là mối quan tâm hàng đầu của cộng đồng. Tuy nhiên, áp lực lên các cơ sở y tế ngày càng lớn, kéo theo khối lượng hồ sơ, thông tin bệnh nhân khổng lồ cần được lưu trữ và xử lý mỗi ngày. Việc quản lý thủ công không chỉ tốn kém thời gian mà còn dễ dẫn đến những sai sót không đáng có, ảnh hưởng trực tiếp đến chất lượng khám chữa bệnh.

Với tình hình thực tế trên, đề tài “**Quản lý bệnh nhân**” được nhóm em tạo ra với mong muốn vận dụng ngôn ngữ lập trình Python và tư duy quản lý cơ sở dữ liệu vào thực tế. Hệ thống được xây dựng nhằm mục tiêu đơn giản hóa quy trình tiếp nhận, hỗ trợ tra cứu hồ sơ bệnh án nhanh chóng, quản lý thông tin điều trị và viện phí một cách minh bạch, chính xác. Chúng em hy vọng sản phẩm này sẽ là một bước thử nghiệm ý nghĩa, góp phần nhỏ bé vào việc tin học hóa quy trình quản lý y tế.

Trong quá trình thực hiện, nhóm đã nỗ lực tìm tòi, áp dụng các kỹ thuật lập trình đã học cũng như tự nghiên cứu thêm các công nghệ mới. Nhưng do kinh nghiệm thực tế còn hạn chế, đồ án chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những lời góp ý, chỉ dẫn của Thầy để sản phẩm ngày càng hoàn thiện hơn.

Một lần nữa, nhóm em xin kính chúc Thầy Nguyễn Ngọc Minh thật nhiều sức khỏe, niềm vui và tiếp tục gặt hái nhiều thành công trong sự nghiệp "trồng người".

PHỤ LỤC

I.	ĐẶT VẤN ĐỀ	1
1.	Tính cần thiết của đề tài	1
2.	Mục tiêu nghiên cứu của đề tài	2
a.	Xây dựng thành công ứng dụng Quản lý bệnh nhân với giao diện thân thiện và trực quan	2
b.	Thiết kế cơ sở dữ liệu MySQL để lưu trữ và quản lý toàn bộ thông tin	2
c.	Triển khai đầy đủ các chức năng quản lý dữ liệu (CRUD) và các chức năng nghiệp vụ	3
II.	TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT	4
3.	Đặt vấn đề	4
4.	Lịch sử giải quyết vấn đề	5
a.	Quản lý truyền thống bằng ghi chép	5
b.	Sổ cái, sổ đăng ký khám bệnh	5
c.	Sử dụng Excel hoặc Word để ghi chép và lưu trữ	6
5.	Phạm vi của đề tài	6
a.	Phạm vi chức năng	6
b.	Phạm vi đối tượng	7
6.	Phương pháp nghiên cứu	8
a.	Thu thập tài liệu	8
b.	Phân tích thiết kế hệ thống	8
7.	Cơ sở lý thuyết	8
a.	Ngôn ngữ lập trình python	8

<i>b.</i>	<i>Mô hình CRUD</i>	9
<i>c.</i>	<i>Lý thuyết về phân tích và thiết kế hệ thống thông tin</i>	10
III.	KẾT QUẢ ĐẠT ĐƯỢC VÀ DEMO	11
1.	Thiết kế giao diện	11
<i>a.</i>	<i>Giao diện đăng nhập</i>	11
<i>b.</i>	<i>Giao diện quản lý bệnh nhân</i>	11
<i>c.</i>	<i>Giao diện quản lý bác sĩ</i>	12
<i>d.</i>	<i>Giao diện quản lý đặt lịch khám bệnh</i>	12
<i>e.</i>	<i>Giao diện quản lý khám bệnh</i>	13
2.	Kết quả DEMO	14
<i>a.</i>	<i>Kết quả đăng nhập</i>	14
<i>b.</i>	<i>Kết quả quản lý bệnh nhân</i>	18
<i>c.</i>	<i>Kết quả quản lý bác sĩ</i>	21
<i>d.</i>	<i>Kết quả quản lý đặt lịch khám bệnh</i>	24
<i>e.</i>	<i>Kết quả quản lý khám bệnh</i>	26
3.	CODE thực hiện	28
<i>a.</i>	<i>Code cơ sở dữ liệu MySQL</i>	28
<i>b.</i>	<i>Code kết nối MySQL</i>	29
<i>c.</i>	<i>Thư viện</i>	30
<i>d.</i>	<i>Code đăng nhập</i>	30
<i>e.</i>	<i>Code quản lý bệnh nhân</i>	31
<i>f.</i>	<i>Code quản lý bác sĩ</i>	35
<i>g.</i>	<i>Code quản lý đặt lịch khám bệnh</i>	38
<i>h.</i>	<i>Code quản lý khám bệnh</i>	41
IV.	TÀI LIỆU THAM KHẢO	46

I. ĐẶT VẤN ĐỀ

1. Tính cần thiết của đề tài

Trong những năm gần đây, nhu cầu chăm sóc sức khỏe của người dân ngày càng được nâng cao, kéo theo khối lượng công việc tại các cơ sở y tế (phòng khám, bệnh viện) gia tăng nhanh chóng. Công tác quản lý hồ sơ bệnh nhân đóng vai trò then chốt trong việc theo dõi lịch sử điều trị, nâng cao chất lượng khám chữa bệnh và đảm bảo sự an toàn cho người bệnh. Tuy nhiên, cùng với sự gia tăng số lượng bệnh nhân, nhiều vấn đề bất cập trong khâu quản lý thủ công đã nảy sinh.

Hiện nay, nhiều phòng khám vừa và nhỏ vẫn quản lý theo phương thức truyền thống như ghi chép sổ khám bệnh, lưu trữ hồ sơ giấy hoặc sử dụng các file Excel hay file Word rời rạc. Cách làm này bộc lộ nhiều hạn chế và rủi ro, cụ thể:

Khó kiểm soát hồ sơ lưu trữ: Dữ liệu bệnh án ngày càng nhiều dẫn đến tình trạng quá tải, dễ thất lạc hồ sơ hoặc tồn rất nhiều thời gian để tìm kiếm lại thông tin cũ.

Dữ liệu thiếu đồng bộ: Thông tin bệnh nhân, lịch khám và hóa đơn viện phí không được liên kết chặt chẽ, dẫn đến sai sót trong việc tính toán doanh thu và báo cáo tài chính.

Khó khăn trong tra cứu chuyên môn: Bác sĩ gặp khó khăn khi muốn xem lại lịch sử dị ứng thuốc, các chẩn đoán trước đó của bệnh nhân để đưa ra phác đồ điều trị chính xác.

Quy trình tiếp nhận chậm chạp: Bệnh nhân phải chờ đợi lâu để làm thủ tục hành chính do nhân viên phải viết tay hoặc nhập liệu thủ công nhiều lần.

Bên cạnh đó, chủ trương của ngành Y tế hiện nay là đẩy mạnh chuyển đổi số, hướng tới bệnh án điện tử và minh bạch hóa thông tin viện phí. Điều này đòi hỏi các cơ sở y tế phải trang bị một giải pháp phần mềm quản lý hiệu quả, bảo mật và có khả năng lưu trữ thông tin khoa học.

Vì vậy, việc nghiên cứu và xây dựng chương trình "*Quản lý bệnh nhân*"(sử dụng ngôn ngữ lập trình Python) là hết sức cần thiết. Đề tài nhằm giải quyết triệt để những tồn tại của phương pháp quản lý thủ công, hỗ trợ đội ngũ y bác sĩ làm việc khoa học, chính xác và tiết

kiêm thời gian hơn. Đây là đề tài có ý nghĩa thực tiễn cao, phù hợp với xu hướng tin học hóa trong y tế hiện đại.

2. Mục tiêu nghiên cứu của đề tài

Mục tiêu của đề tài là phát triển một giải pháp quản lý tập trung, tích hợp đầy đủ các module chức năng cần thiết cho một cơ sở y tế: từ *Quản lý danh sách và thông tin chuyên môn của bác sĩ*, *Lưu trữ hồ sơ bệnh nhân*, cho đến việc *Xử lý đặt lịch khám và Ghi nhận kết quả khám bệnh*. Hệ thống nhằm đảm bảo tính toàn vẹn của dữ liệu và hỗ trợ tra cứu thông tin nhanh chóng, chính xác.

a. Xây dựng thành công ứng dụng Quản lý bệnh nhân với giao diện thân thiện và trực quan

Thiết kế giao diện trực quan, dễ tiếp cận, tích hợp các chức năng theo từng tab/module riêng biệt bao gồm: *Quản lý bệnh nhân*; *Quản lý bác sĩ*; *Quản lý đặt lịch khám* và *Quản lý khám bệnh*, giúp nhân viên y tế thao tác thuận tiện và logic.

Xây dựng các khung nhập liệu chi tiết, bao quát đầy đủ các trường thông tin y tế cần thiết (thông tin cá nhân, lịch khám bệnh, chuyên khoa...); hiển thị dữ liệu danh sách dưới dạng bảng rõ ràng, khoa học và dễ theo dõi.

Tối ưu hóa bố cục giao diện để hỗ trợ người dùng **tra cứu hồ sơ** và **cập nhật trạng thái khám** một cách nhanh chóng, giảm thiểu tối đa sai sót trong quá trình nhập liệu thông tin bệnh án.

Giao diện ứng dụng hướng đến sự ổn định, mượt mà trong quá trình vận hành, đảm bảo tính thẩm mỹ và có khả năng mở rộng, tích hợp thêm các tiện ích y tế khác trong tương lai.

b. Thiết kế cơ sở dữ liệu MySQL để lưu trữ và quản lý toàn bộ thông tin

Xây dựng mô hình cơ sở dữ liệu quan hệ chuẩn hóa, bao gồm đầy đủ các bảng nghiệp vụ cốt lõi như: *BenhNhan* (Bệnh nhân), *BacSi* (Bác sĩ), *LichKham* (Lịch hẹn), *PhieuKhamBenh* (Kết quả khám bệnh)...

Xác định và thiết lập hệ thống khóa chính – khóa ngoại (Primary Key - Foreign Key) chặt chẽ giữa các bảng để đảm bảo tính toàn vẹn tham chiếu, ngăn chặn tình trạng dư thừa hoặc mâu thuẫn dữ liệu trong quá trình vận hành.

Cơ sở dữ liệu (Database) được thiết kế chi tiết, đặc biệt là thông tin về hồ sơ y tế, bao gồm:

- Thông tin hành chính: Mã bệnh nhân, họ tên, ngày sinh, giới tính, điện thoại, địa chỉ và bảo hiểm y tế.
- Thông tin khám bệnh: Mã bệnh nhân, Mã bác sĩ, Triệu chứng lâm sàng, kết quả chẩn đoán.
- Trạng thái hồ sơ: Đặt lịch khám, ngày khám và ghi chú đặc biệt.

Thông tin về đội ngũ Bác sĩ (chuyên khoa, trình độ) được lưu trữ minh bạch, liên kết trực tiếp với bảng Lịch khám để hỗ trợ việc điều phối nhân sự và truy xuất trách nhiệm chuyên môn khi cần thiết.

Sử dụng hệ quản trị MySQL nhằm đảm bảo hệ thống hoạt động ổn định, khả năng bảo mật thông tin cao, tốc độ truy vấn dữ liệu nhanh chóng và phù hợp với quy mô dữ liệu ngày càng tăng của phòng khám.

c. *Triển khai đầy đủ các chức năng quản lý dữ liệu (CRUD) và các chức năng nghiệp vụ*

Quản lý hồ sơ Bệnh nhân và Bác sĩ:

- Xây dựng đầy đủ chức năng CRUD (Thêm – Sửa – Xóa – Lưu) cho dữ liệu bệnh nhân và bác sĩ.
- Lưu trữ chi tiết thông tin hành chính, ghi nhận triệu chứng của bệnh nhân và thông tin chuyên khoa, chức vụ của bác sĩ.
- Quản lý Khám chữa bệnh: Xây dựng chức năng tạo Phiếu khám bệnh với đầy đủ thông tin: triệu chứng, chẩn đoán sơ bộ.

Quản lý Đặt lịch khám:

- Cung cấp chức năng đăng ký ngày hẹn, thời gian khám và đặt lịch.
- Hệ thống hỗ trợ kiểm tra tính hợp lệ của thời gian, tránh việc trùng lặp lịch hẹn của cùng một bác sĩ trong cùng khung giờ.

Tính năng mở rộng và Bảo mật:

- Đăng nhập – Phân quyền: Thiết lập cơ chế bảo mật, phân quyền truy cập rõ ràng (ví dụ: Tiếp tân chỉ được xem lịch, Bác sĩ mới được sửa kết quả khám) để bảo vệ dữ liệu y tế nhạy cảm.

II. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

3. Đặt vấn đề

Xu hướng chuyển đổi số trong y tế trong kỷ nguyên công nghệ 4.0, ngành y tế Việt Nam đang chứng kiến những bước chuyển mình mạnh mẽ theo hướng hiện đại hóa và thông minh hóa. Việc ứng dụng công nghệ thông tin vào công tác khám chữa bệnh không còn là một lựa chọn mang tính xu hướng, mà đã trở thành yêu cầu cốt lõi để nâng cao chất lượng dịch vụ. Các cơ sở y tế, từ bệnh viện công đến các phòng khám tư nhân, đang chịu áp lực lớn về việc phải đổi mới quy trình vận hành nhằm đáp ứng nhu cầu chăm sóc sức khỏe ngày càng cao và khắt khe của người dân.

Thực trạng quản lý tại các phòng khám vừa và nhỏ. Tuy nhiên, trái ngược với sự đầu tư về trang thiết bị y tế chuyên môn, khâu quản lý hành chính tại nhiều phòng khám hiện nay vẫn còn mang nặng tính thủ công. Phần lớn các cơ sở này vẫn duy trì thói quen lưu trữ hồ sơ bệnh án qua hệ thống sổ sách giấy tờ công kèn hoặc sử dụng các công cụ văn phòng đơn giản như Microsoft Excel, Word. Quy trình này thường diễn ra rời rạc: nhân viên tiếp tân ghi chép thủ công, bác sĩ kê đơn trên giấy, và việc lưu trữ lịch sử khám bệnh hoàn toàn phụ thuộc vào kho vật lý.

Những bất cập và thách thức về phương thức quản lý truyền thống này đang bộc lộ những "rủi ro" nghiêm trọng, kìm hãm sự phát triển của phòng khám:

Rủi ro về dữ liệu: Hồ sơ giấy dễ bị thất lạc, ẩm mốc hoặc hư hỏng theo thời gian, dẫn đến việc mất mát lịch sử điều trị quan trọng của bệnh nhân.

Sự thiếu đồng bộ và lãng phí thời gian: Việc tra cứu lại thông tin của một bệnh nhân cũ có thể mất hàng giờ đồng hồ lục tìm trong kho lưu trữ. Bên cạnh đó, việc xếp lịch khám thủ công thường xuyên dẫn đến tình trạng chồng chéo lịch hẹn hoặc để trống lịch bác sĩ một cách lãng phí.

Mục tiêu của đề tài xuất phát từ nhu cầu cấp thiết trong thực trạng ngày nay, nhóm nghiên cứu quyết định xây dựng và phát triển hệ thống "**Quản lý bệnh nhân**". Đây là một giải pháp phần mềm toàn diện, được thiết kế chuyên biệt để số hóa quy trình nghiệp vụ phòng khám. Ứng dụng không chỉ tập trung vào việc lưu trữ và bảo mật thông

tin hò sơ bệnh nhân, quản lý chặt chẽ đội ngũ y bác sĩ, mà còn giải quyết bài toán điều phối lịch khám một cách khoa học. Đề tài này được kỳ vọng sẽ thay thế hoàn toàn phương thức làm việc thủ công, giảm thiểu sai sót do con người và tạo ra một môi trường y tế chuyên nghiệp, hiện đại, nơi bác sĩ có thể tập trung tối đa vào công tác chuyên môn thay vì bị phân tâm bởi các thủ tục hành chính.

4. Lịch sử giải quyết vấn đề

a. Quản lý truyền thống bằng ghi chép

Trước khi kỹ nguyên số hóa len lỏi vào các phòng khám, *giấy và bút* là công cụ quản lý duy nhất nhưng cũng chính là rào cản lớn nhất. Sự phụ thuộc vào các ghi chép vật lý rời rạc đã tạo ra những lỗ hổng nghiêm trọng trong vận hành:

- Sự đứt gãy dữ liệu: Thông tin bệnh nhân bị phân tán, sai lệch do thao tác nhập liệu thủ công, dẫn đến những chẩn đoán thiếu cơ sở dữ liệu lịch sử.
- Mù mờ về nguồn lực: Không thể nắm bắt tức thời trạng thái làm việc của bác sĩ hay các khung giờ trống, khiến việc điều phối lịch hẹn trở nên thụ động và kém hiệu quả.
- Lãng phí tài nguyên thời gian: Quy trình truy vết hò sơ bệnh án y tế trở thành gánh nặng hành chính, chiếm dụng thời gian quý báu lẽ ra dành cho công tác chuyên môn.
- Rào cản mở rộng: Khi lượng bệnh nhân tăng đột biến, phương thức thủ công trở nên quá tải, dễ dẫn đến khủng hoảng dịch vụ và mất kiểm soát quy trình.

b. Sổ cái, sổ đăng ký khám bệnh

Trong khi phần mềm cho phép tìm kiếm bệnh nhân chỉ trong tích tắc bằng một từ khóa, thì phương pháp dùng Sổ đăng ký lại buộc nhân viên phải dựa vào trí nhớ hoặc dò tìm thủ công.

- Về lưu trữ: Sổ sách chiếm diện tích vật lý lớn, dễ hư hỏng và không thể sao lưu (backup).
- Về bảo mật: Cuốn sổ cái nằm trên bàn ai cũng có thể lật xem, hoàn toàn không có cơ chế phân quyền hay bảo mật thông tin riêng tư của người bệnh.
- Về kết nối: Dữ liệu trong sổ đăng ký bị cô lập, không thể tự động chuyển vào phiếu khám của bác sĩ, buộc bác sĩ phải hỏi và ghi chép lại từ đầu, gây lãng phí thao tác thừa.

c. Sử dụng Excel hoặc Word để ghi chép và lưu trữ

Mặc dù Excel và Word đã giải quyết được vấn đề lưu trữ văn bản, nhưng chúng hoàn toàn bất lực trong việc quản lý quy trình nghiệp vụ động tại phòng khám:

- Thiếu tính đồng bộ thời gian thực: Khi lễ tân thêm một lịch hẹn mới, thông tin không tự động xuất hiện trên màn hình của bác sĩ, dẫn đến độ trễ trong khâu tiếp nhận.
- Dữ liệu bị cô lập (Data Silos): Thông tin bệnh nhân, lịch sử khám và đơn thuốc nằm ở các file hoặc sheet tách biệt, không có sự liên kết logic, khiến việc tra cứu tổng hợp trở nên khó khăn.
- Rủi ro vận hành: Không có cơ chế sao lưu tự động hay phân quyền, chỉ cần một thao tác 'xóa nhầm' hoặc lỗi file là toàn bộ dữ liệu khám bệnh có thể biến mất.
- Thiếu các tính năng thông minh: Excel không thể tự động nhắc lịch tái khám cho bệnh nhân hay cảnh báo bác sĩ về các loại thuốc bệnh nhân bị dị ứng.
- Giới hạn truy cập: Không hỗ trợ đa người dùng cùng lúc (concurrency), gây tắc nghẽn khi nhiều bộ phận cần thao tác trên cùng một dữ liệu.

5. Phạm vi của đề tài

Đề tài tập trung nghiên cứu và xây dựng hệ thống phần mềm được thiết kế chuyên biệt, phù hợp với đặc thù vận hành của các phòng khám tư nhân và cơ sở khám chữa bệnh quy mô vừa và nhỏ. Hệ thống đảm bảo tính tinh gọn, dễ triển khai mà không đòi hỏi hạ tầng phần cứng quá phức tạp.

a. Phạm vi chức năng

Quản lý hồ sơ Bệnh nhân: Xây dựng chức năng lưu trữ và cập nhật thông tin hành chính (Họ tên, tuổi, địa chỉ) và thông tin y tế (BHYT) của bệnh nhân. Hỗ trợ tra cứu nhanh hồ sơ khi bệnh nhân đến tái khám.

Quản lý đội ngũ Bác sĩ: Quản lý danh sách, thông tin cá nhân, trình độ và chuyên khoa của các bác sĩ. Đây là cơ sở dữ liệu quan trọng để thực hiện việc phân công và xếp lịch khám phù hợp với nhu cầu của người bệnh.

Quản lý Đặt lịch khám: Hỗ trợ nhân viên tiếp nhận lịch hẹn, sắp xếp thời gian và cập nhật trạng thái lịch khám (Đặt lịch, Hủy). Hệ

thống giúp kiểm soát để đảm bảo không xảy ra tình trạng trùng giờ hẹn của cùng một bác sĩ.

Quản lý quá trình Khám bệnh: Số hóa quy trình thăm khám lâm sàng. Tại đây, bác sĩ có thể ghi nhận các triệu chứng, đưa ra chẩn đoán và lưu lại kết quả vào hệ thống, sau đó kê đơn thuốc cho từng bệnh nhân.

b. Phạm vi đối tượng

Người dùng hệ thống: Quản trị viên (Admin), Đội ngũ Bác sĩ và Nhân viên.

Cơ chế phân quyền: Hệ thống thiết lập chế độ bảo mật dựa trên vai trò, đảm bảo mỗi đối tượng chỉ thao tác trên các Form chức năng phù hợp với nhiệm vụ của mình:

Đối với tài khoản Quản trị viên (Admin): Đây là tài khoản có quyền kiểm soát toàn bộ hệ thống. Admin chịu trách nhiệm quản lý tài khoản người dùng, quản lý danh mục Bác sĩ và quản lý danh mục Thuốc (nhập mới, chỉnh sửa thông tin thuốc). Admin đảm bảo dữ liệu nền tảng luôn chính xác để phục vụ cho quá trình vận hành.

Đối với tài khoản Bác sĩ: Tập trung hoàn toàn vào nghiệp vụ khám chữa bệnh. Quyền hạn bao gồm:

- Truy cập form khám bệnh: Để ghi nhận triệu chứng, chẩn đoán bệnh lý cho bệnh nhân hiện tại.
- Truy cập form thuốc: Để tra cứu danh mục thuốc có trong kho và thực hiện thao tác kê đơn trực tiếp trên phần mềm dựa trên kết quả chẩn đoán.

Đối với tài khoản Nhân viên (Tiếp nhận/Lễ tân): Phụ trách khâu đầu vào. Nhân viên có quyền truy cập Form *Quản lý bệnh nhân* để thêm mới hoặc cập nhật thông tin hành chính. Đồng thời, nhân viên sử dụng Form Đặt lịch để sắp xếp thời gian khám và điều phối bệnh nhân về cho các bác sĩ.

Ý nghĩa: Việc phân quyền này giúp quy trình diễn ra một chiều và khép kín: Nhân viên tiếp nhận và đầy dữ liệu vào Lịch khám Bác sĩ nhận

lịch, khám và xuất đơn thuốc. Bác sĩ không cần lo về việc nhập thông tin hành chính, và Nhân viên không thể can thiệp vào đơn thuốc.

Đối tượng quản lý chính: Hồ sơ bệnh nhân, Thông tin bác sĩ, Lịch khám và Khám bệnh.

6. Phương pháp nghiên cứu

a. Thu thập tài liệu

Tài liệu pháp lý và quy định nghiệp vụ:

- Nghiên cứu Luật Khám bệnh, chữa bệnh và các quy định hiện hành của Bộ Y tế về quy trình tiếp nhận, quản lý hồ sơ bệnh án và quy chế kê đơn thuốc trong điều trị ngoại trú.
- Tìm hiểu các biểu mẫu chuẩn trong y tế như: Phiếu khám bệnh, Đơn thuốc để thiết kế giao diện nhập liệu và mẫu in án cho phù hợp với thực tế.

b. Phân tích thiết kế hệ thống

Vận dụng nền tảng kiến thức từ môn học *Phân tích thiết kế hệ thống thông tin*, nhóm đã tiến hành mô hình hóa yêu cầu bài toán một cách chặt chẽ và khoa học thông qua các công cụ phân tích kỹ thuật:

- Tham khảo sơ đồ quy trình nghiệp vụ (Business Process Diagram): Để mô tả trực quan các bước vận hành thực tế tại phòng khám, từ khâu tiếp nhận bệnh nhân, xếp lịch đến quy trình khám chữa bệnh và cấp thuốc.
- Xây dựng sơ đồ luồng dữ liệu (DFD): Để phân tích dòng chảy của thông tin trong hệ thống, xác định rõ nguồn dữ liệu đầu vào, nơi lưu trữ và các thông tin đầu ra cần thiết cho từng chức năng.
- Xây dựng sơ đồ thực thể – liên kết (ERD): Để thiết kế cấu trúc cơ sở dữ liệu, xác định các thực thể quan trọng (Bệnh nhân, Bác sĩ, Lịch khám...) và mối quan hệ ràng buộc giữa chúng.
- Việc áp dụng các mô hình phân tích trên giúp đảm bảo hệ thống được thiết kế logic, đầy đủ chức năng và sát với yêu cầu nghiệp vụ thực tế.

7. Cơ sở lý thuyết

a. Ngôn ngữ lập trình python

Python được biết đến là ngôn ngữ lập trình bậc cao với tư duy hướng đối tượng mạnh mẽ, nổi bật nhờ cú pháp gãy gọn và khả năng tương thích đa nền tảng. Không chỉ dừng lại ở việc dễ đọc – dễ học, sức

mạnh thực sự của Python nằm ở hệ sinh thái thư viện phong phú, hỗ trợ tối đa cho việc phát triển ứng dụng nhanh chóng và hiệu quả.

Trong khuôn khổ đề án *Quản lý bệnh nhân*, Python đóng vai trò là công nghệ nền tảng cốt lõi. Chúng em sử dụng Python để kiến tạo giao diện người dùng trực quan, đảm bảo trải nghiệm thao tác thuận tiện cho đội ngũ y bác sĩ. Đồng thời, ở tầng xử lý Python đảm nhận nhiệm vụ xử lý các logic nghiệp vụ phức tạp như điều phối lịch khám, truy xuất hồ sơ bệnh án và tính toán kê đơn. Đặc biệt, thông qua các thư viện kết nối chuyên dụng, Python hoạt động như một cầu nối linh hoạt với hệ quản trị cơ sở dữ liệu MySQL, đảm bảo dòng chảy dữ liệu trong hệ thống luôn được duy trì liền mạch, an toàn và chính xác.

b. Mô hình CRUD

Để đảm bảo tính toàn vẹn và linh hoạt của dòng chảy thông tin, hệ thống được xây dựng dựa trên nền tảng 4 thao tác chuẩn hóa của mô hình CRUD:

- Create (Khởi tạo): Là thao tác đưa dữ liệu đầu vào vào hệ thống. Trong ngữ cảnh phòng khám, chức năng này cho phép số hóa thông tin hành chính của bệnh nhân và ghi nhận các lượt đăng ký khám mới.
- Read (Tra cứu): Là khả năng truy vấn và hiển thị dữ liệu từ kho lưu trữ MySQL lên giao diện ứng dụng. Đây là công cụ đắc lực giúp y bác sĩ nắm bắt tiền sử bệnh lý của bệnh nhân chỉ trong vài cú click chuột.
- Update (Hiệu chỉnh): Cho phép sửa đổi thông tin để đảm bảo tính thời sự của dữ liệu. Ví dụ: Cập nhật đơn thuốc, thay đổi thông tin liên lạc của bệnh nhân hoặc điều chỉnh lịch trực của bác sĩ.
- Delete (Hủy bỏ): Là thao tác xóa logic hoặc xóa vật lý đối với các dữ liệu không còn giá trị sử dụng (như dữ liệu nhập thử nghiệm, nhập trùng lặp), đảm bảo tài nguyên hệ thống không bị lãng phí.

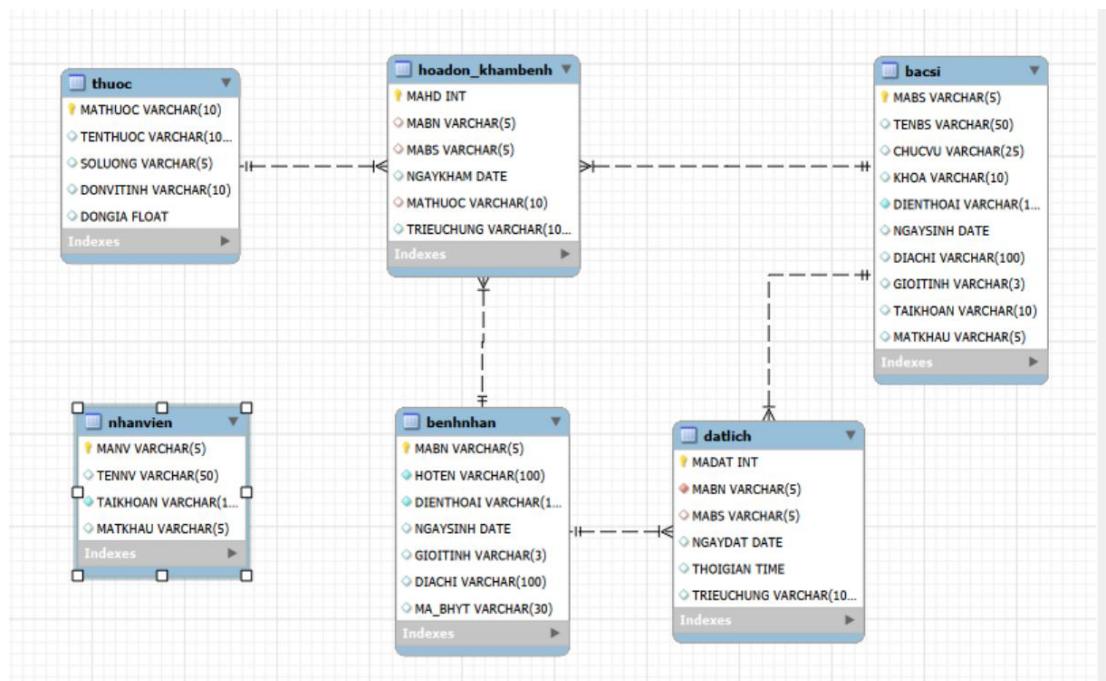
Việc tuân thủ chặt chẽ mô hình CRUD giúp cấu trúc chương trình trở nên mạch lạc, dễ dàng bảo trì và mở rộng tính năng trong tương lai.

c. Lý thuyết về phân tích và thiết kế hệ thống thông tin

Nếu ví việc lập trình (Coding) là quá trình xây dựng những viên gạch, thì **Phân tích và Thiết kế hệ thống** chính là bước kiến tạo nên bản vẽ kiến trúc của ngôi nhà đó.

Quá trình này là cầu nối thiết yếu giữa nhu cầu nghiệp vụ của người dùng và giải pháp công nghệ của lập trình viên:

- **Ở bước Phân tích:** Nhóm đã đóng vai trò là những người lắng nghe, tìm hiểu xem người dùng thực sự cần quản lý những dữ liệu gì (Bệnh nhân, Bác sĩ, Lịch khám) và quy trình làm việc hiện tại của họ ra sao.
- **Ở bước Thiết kế:** Từ những yêu cầu đó, nhóm tiến hành mô hình hóa thành các sơ đồ kỹ thuật (Database, Giao diện...), tạo nên một "bước nền" vững chắc trước khi bắt tay vào viết những dòng code đầu tiên.



Hình 1. Sơ đồ Quan hệ Thực thể (ERD) của Database QuanLyBenhNhan

Vận dụng lý thuyết này vào đề tài *Quản lý bệnh nhân*, nhóm đã tiến hành phân tích quy trình khám chữa bệnh thực tế, từ đó thiết kế nên mô hình cơ sở dữ liệu MySQL và giao diện Python tối ưu nhất cho người sử dụng. Nhờ tuân thủ quy trình này, sản phẩm cuối cùng đảm bảo được tính logic, tránh được các sai sót về cấu trúc và đáp ứng đúng nhu cầu thực tế của phòng khám.

III. KẾT QUẢ ĐẠT ĐƯỢC VÀ DEMO

1. Thiết kế giao diện

a. Giao diện đăng nhập



Hình 2. Giao diện đăng nhập Hệ thống

b. Giao diện quản lý bệnh nhân

The screenshot shows a Windows-style application window titled "Quản lý bệnh nhân - Quyền: ADMIN". The main title of the application is "QUẢN LÝ BỆNH NHÂN". The interface includes a "Thông tin" section with fields for: Mã bệnh nhân, Họ và tên, Số BHYT, Địa chỉ, Số điện thoại, Ngày sinh, and Giới tính. Below this is a table listing five patients with columns: Mã bệnh nhân, Họ và tên, Giới tính, Số BHYT, Địa chỉ, Số điện thoại, and Ngày sinh. At the bottom are five buttons: "Thêm" (highlighted in blue), "Xóa", "Sửa", "Lưu" (highlighted in pink), and "Làm mới".

Mã bệnh nhân	Họ và tên	Giới tính	Số BHYT	Địa chỉ	Số điện thoại	Ngày sinh
BN001	Trần Hoàng Quân	Nam	1014523125	Long Xuyên	0987654921	20/05/1998
BN002	Phạm Kiều Thu	Nữ	1097583501	Châu Đốc	0901234767	08/03/1999
BN003	Nguyễn Văn Thế	Nam	1062378926	Bình Mỹ- An Giang	0912344678	02/02/2002
BN004	Nguyễn Ngọc Châu	Nữ	1035773290	Chợ Mới- An Giang	0968089999	30/07/1997
BN005	Lê Quỳnh Thư	Nữ	1025644893	Châu Đốc- An Giang	0977665524	19/01/2002

Hình 3. Giao diện Quản lý Bệnh nhân

c. Giao diện quản lý bác sĩ

The screenshot shows a Windows application window titled 'QUẢN LÝ BÁC SĨ'. At the top, there's a header bar with the title and some icons. Below it is a form section labeled 'Thông tin' (Information) containing fields for: Mã bác sĩ (Doctor ID), Họ và tên (Name), Chức vụ (Position), Chuyên khoa (Specialty), Điện thoại (Phone), Ngày sinh (Date of Birth), Địa chỉ (Address), and Giới tính (Gender). Below the form is a table listing five doctors with columns for ID, Name, Position, Specialty, Phone, Date of Birth, Address, and Gender. At the bottom are five buttons: 'Thêm' (Add), 'Xóa' (Delete), 'Sửa' (Edit), 'Lưu' (Save), and 'Làm mới' (New).

Mã bác sĩ	Họ và tên	Chức vụ	Chuyên khoa	Điện thoại	Ngày sinh	Địa chỉ	Giới tính
BS001	Nguyễn Văn Hưng	Bác sĩ chính thức	Nội	0987654321	15/05/1990	Long Xuyên	Nam
BS002	Trần Thị Diễm	Bác sĩ thực tập	Ngoại	0901234567	20/11/1980	Chợ Mới	Nữ
BS003	Hoàng Xuân Vinh	Bác sĩ thực tập	Ngoại	0912345678	10/02/1995	Bình Mỹ- An Giang	Nam
BS004	Phạm Huỳnh Thúy An	Bác sĩ chính thức	Ngoại	0968889999	25/07/1997	Đồng Xuyên- Long Xuyên-	Nữ
BS005	Trương Đinh Hoàng	Bác sĩ thực tập	Nội	0977665544	01/01/2000	Bình Khánh- An Giang	Nam

Hình 4. Giao diện Quản lý Bác sĩ

d. Giao diện quản lý đặt lịch khám bệnh

The screenshot shows a Windows application window titled 'ĐẶT LỊCH KHÁM BỆNH'. At the top, there's a header bar with the title and some icons. Below it is a form section labeled 'Thông tin' (Information) containing fields for: Mã đặt (Appointment ID), Ngày đặt (Booking Date), Thời gian (Time), Mã bệnh nhân (Patient ID), and Mã bác sĩ (Doctor ID). Below the form is a table listing bookings with columns for ID, Date, Time, Patient ID, Doctor ID, and Symptoms. At the bottom are two buttons: 'Đặt lịch' (Book) and 'Hủy' (Cancel).

Mã đặt	Ngày đặt	Thời gian	Mã bệnh nhân	Mã bác sĩ	Triệu chứng
	29/11/2025	07:00			

Hình 5. Giao diện Quản lý Đặt lịch khám

e. Giao diện quản lý khám bệnh

The screenshot shows a Windows application window titled 'KHÁM BỆNH'. The main area is divided into two main sections: 'Đặt lịch (Chờ khám)' (Appointment Booking) on the left and 'Thuốc' (Medicine) on the right.

Đặt lịch (Chờ khám)

Mã đặt	Mã BN	Mã BS	Triệu chứng

Thuốc

Mã thuốc	<input type="text"/>
Tên thuốc	<input type="text"/>
Số lượng	<input type="text"/>
Đơn giá	<input type="text"/> VND

Khám bệnh

Mã bác sĩ	<input type="text"/>
Mã bệnh nhân	<input type="text"/>
Triệu Chứng	<input type="text"/>
Chẩn đoán	<input type="text"/>

Buttons:

- Xuất hóa đơn (Print Prescription) - Red button
- Hủy (Cancel) - Grey button

Hình 6. Giao diện Khám bệnh

2. Kết quả DEMO

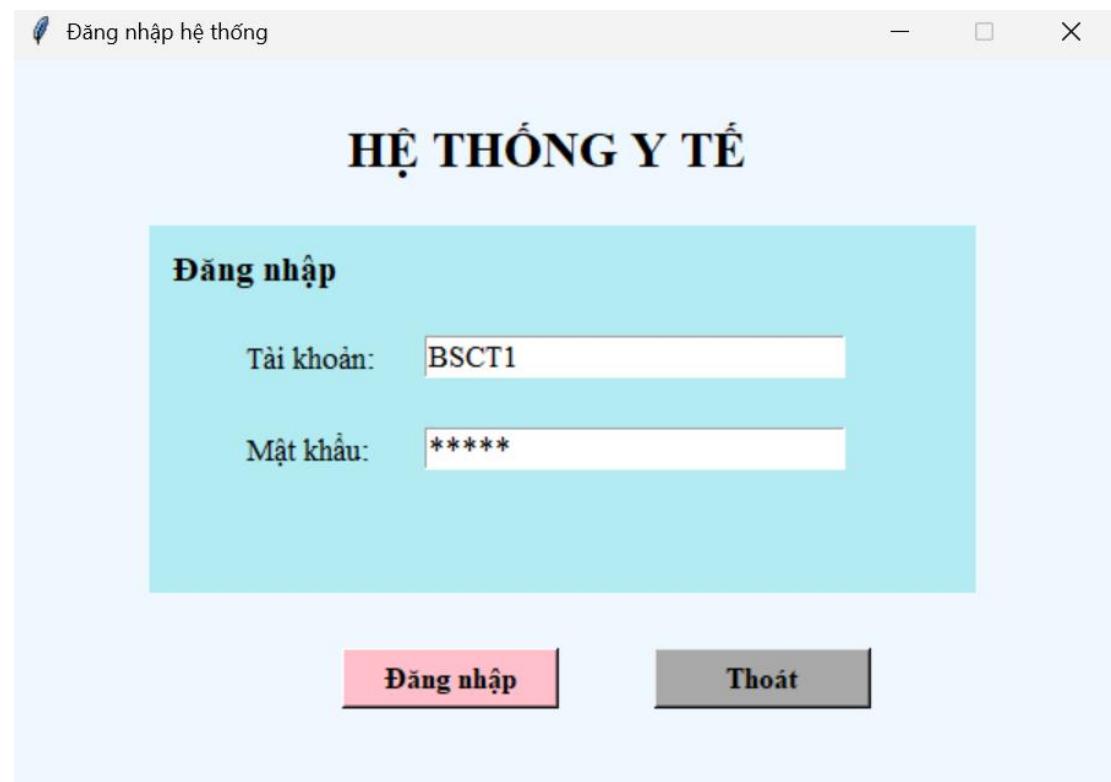
a. Kết quả đăng nhập



Hình 7. Demo đăng nhập bằng quyền ADMIN



Hình 8. Demo đăng nhập bằng quyền ADMIN thành công



Hình 9. Demo đăng nhập bằng quyền Bác sĩ



Hình 10. Demo đăng nhập bằng quyền Bác sĩ thành công

Quản lý bệnh nhân - Quyền: BACSI

Hệ thống (bacsi) Quản lý

- Quản lý Bệnh Nhân
- Quản lý Bác Sĩ (Admin only)
- Đặt lịch khám (No Access)
- Khám bệnh

QUẢN LÝ BỆNH NHÂN

Thông tin

Mã bệnh nhân:	Địa chỉ:
Họ và tên:	Số điện thoại:
Mã BHYT:	Ngày sinh:
Giới tính:	Nam

Thao tác

Thêm Xóa Sửa Lưu Làm mới

Mã bệnh nhân	Họ và tên	Giới tính	Mã BHYT	Địa chỉ	Số điện thoại	Ngày sinh
BN001	Trần Hoàng Quân	Nam	1014523125	Long Xuyên	0987654921	20/05/1998
BN002	Phạm Kiều Thu	Nữ	1097583501	Châu Đốc	0901234767	08/03/1999
BN003	Nguyễn Văn Thể	Nam	1062378926	Bình Mỹ- An Giang	0912344678	02/02/2002
BN004	Nguyễn Ngọc Châu	Nữ	1035773290	Chợ Mới- An Giang	0968089999	30/07/1997
BN005	Lê Quýnh Thư	Nữ	1025644893	Châu Đốc- An Giang	0977665524	19/01/2002

Hình 11. Demo các chức năng khi đăng nhập bằng quyền Bác sĩ.



Hình 12. Demo đăng nhập bằng quyền Nhân viên



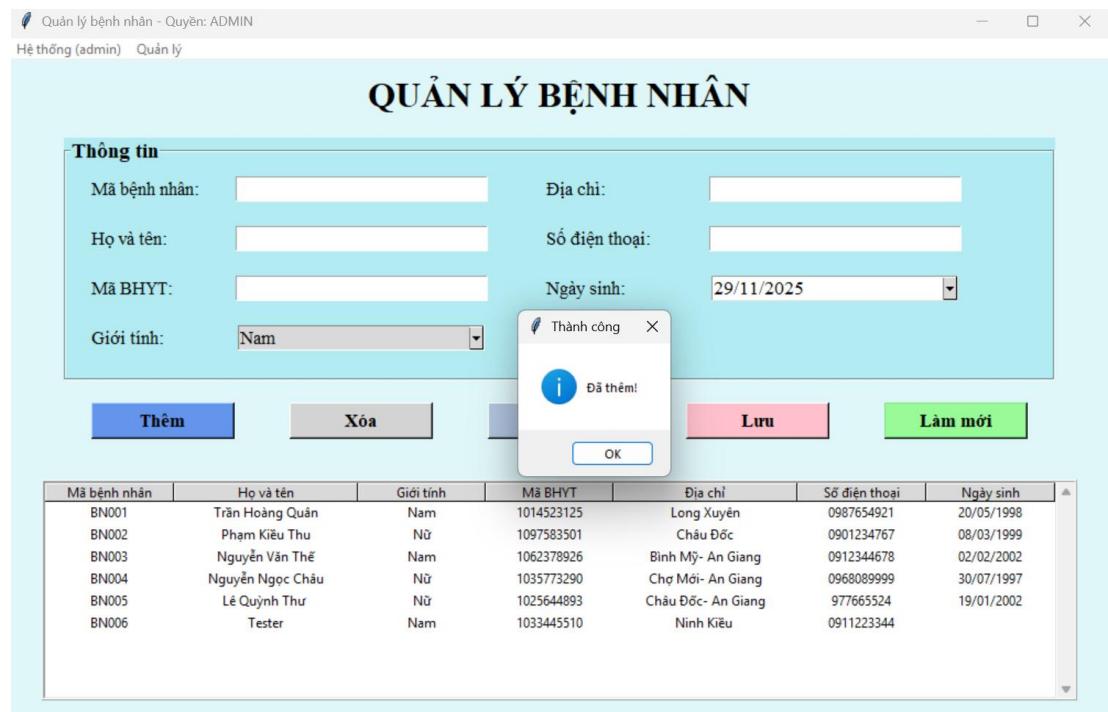
Hình 13. Demo đăng nhập bằng quyền Nhân viên thành công



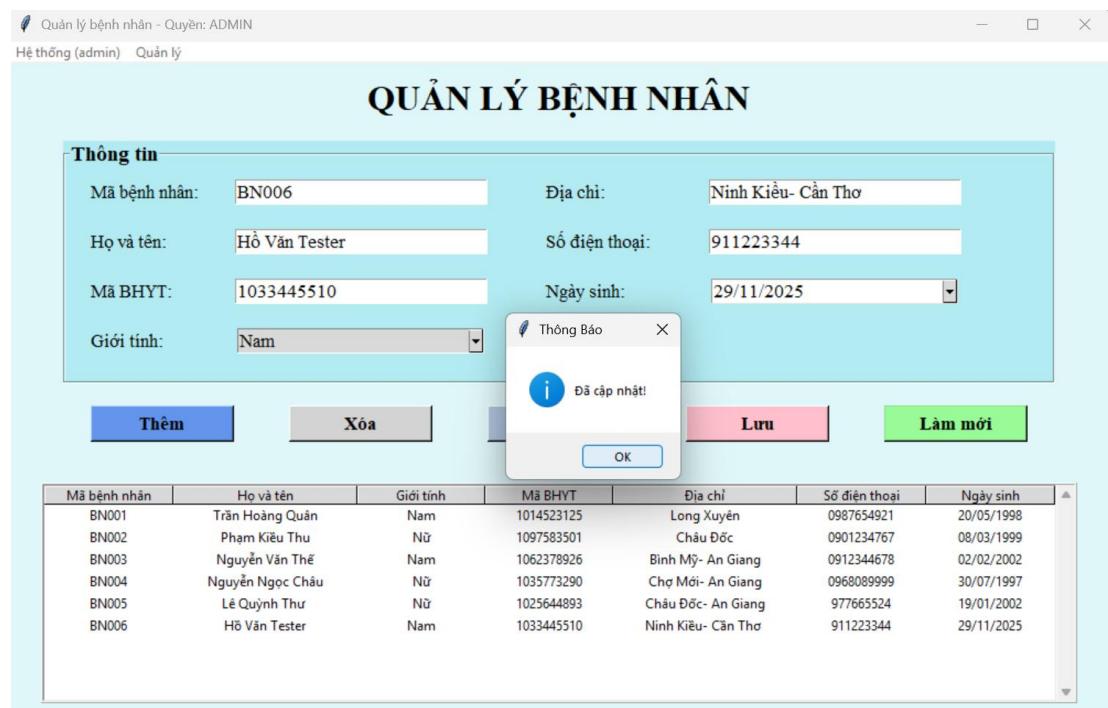
Hình 14. Demo các chức năng khi đăng nhập bằng quyền Nhân viên.



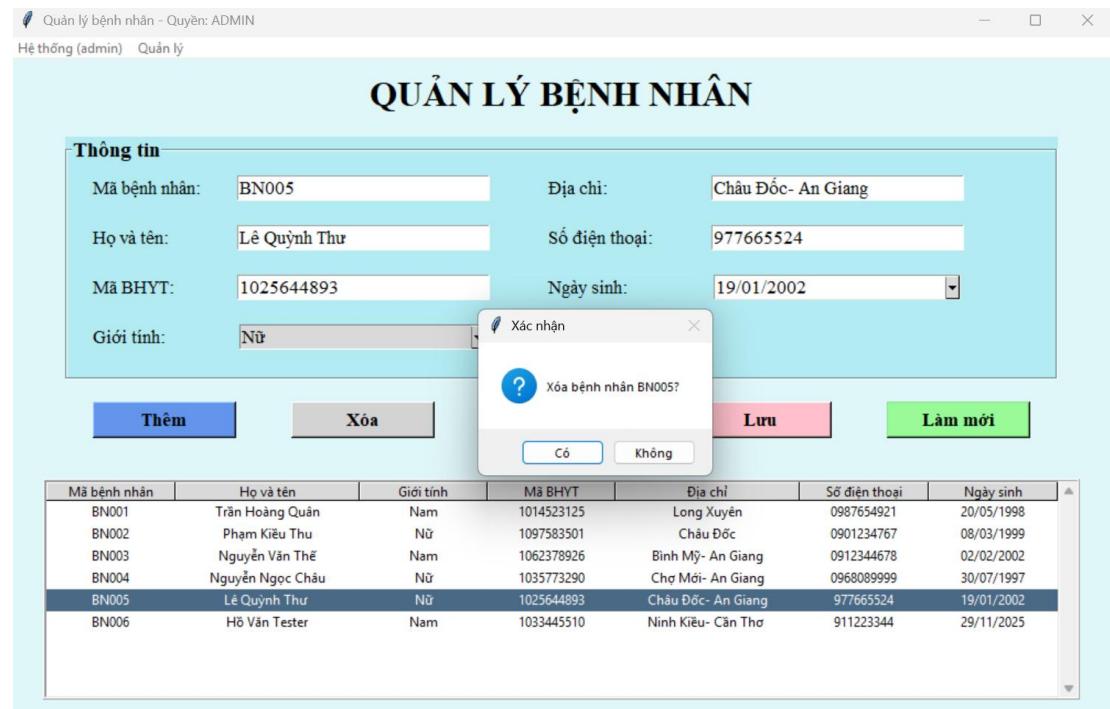
Hình 15. Demo chức năng “Thêm”



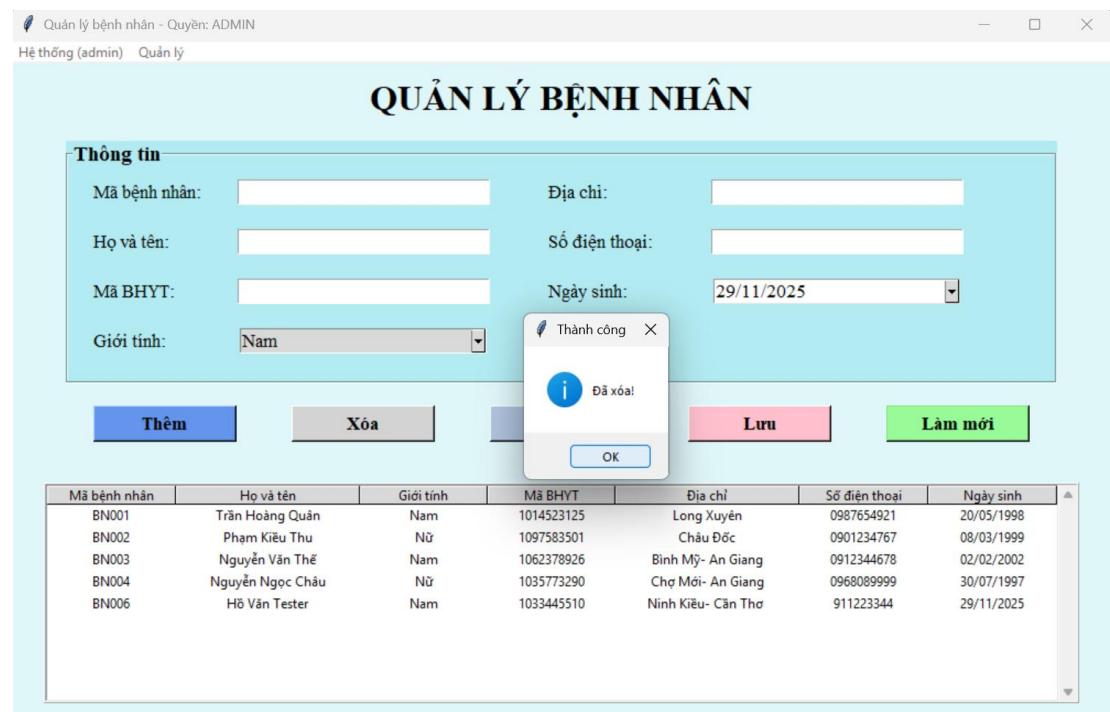
Hình 16. Kết quả demo chức năng “Thêm”



Hình 17. Kết quả demo chức năng “Sửa”



Hình 18. Demo chức năng “Xóa”



Hình 19. Kết quả demo chức năng “Xóa”

Quản lý bệnh nhân - Quyền: ADMIN

Hệ thống (admin) Quản lý

QUẢN LÝ BỆNH NHÂN

Thông tin

Mã bệnh nhân:	<input type="text"/>	Địa chỉ:	<input type="text"/>
Họ và tên:	<input type="text"/>	Số điện thoại:	<input type="text"/>
Mã BHYT:	<input type="text"/>	Ngày sinh:	29/11/2025 <input type="button" value="▼"/>
Giới tính:	Nam <input type="radio"/>	Thông báo ! Dữ liệu đã được lưu và cập nhật từ hệ thống! <input type="button" value="OK"/>	

Thêm **Xóa** **Lưu** **Làm mới**

Mã bệnh nhân	Họ và tên	Giới tính	Mã BHYT	Địa chỉ	Số điện thoại	Ngày sinh
BN001	Trần Hoàng Quân	Nam	1014523125	Long Xuyên	0987654921	20/05/1998
BN002	Phạm Kiều Thu	Nữ	1097583501	Châu Đức	0901234767	08/03/1999
BN003	Nguyễn Văn Thể	Nam	1062378926	Bình Mỹ- An Giang	0912344678	02/02/2002
BN004	Nguyễn Ngọc Châu	Nữ	1035773290	Chợ Mới- An Giang	0968089999	30/07/1997
BN006	Hồ Văn Tester	Nam	1033445510	Ninh Kiều- Cần Thơ	911223344	29/11/2025

Hình 20. Kết quả demo chức năng “Lưu”

Quản lý Bác sĩ - Quyền: ADMIN

Hệ thống (admin) Quản lý

QUẢN LÝ BÁC SĨ

Thông tin

Mã bác sĩ:	<input type="text" value="BS006"/>	Điện thoại:	<input type="text" value="0246813579"/>
Họ và tên:	<input type="text" value="Tester"/>	Ngày sinh:	<input style="width: 100px;" type="text" value="15/05/1990"/>
Chức vụ:	<input style="width: 150px;" type="text" value="Bác sĩ thực tập"/>	Địa chỉ:	<input type="text" value="Đồng Nai"/>
Chuyên khoa:	<input style="width: 150px;" type="text" value="Ngoại"/>	Giới tính:	<input style="width: 100px;" type="text" value="Nam"/>

Thêm **Xóa** **Sửa** **Lưu** **Làm mới**

Mã bác sĩ	Họ và tên	Chức vụ	Chuyên khoa	Điện thoại	Ngày sinh	Địa chỉ	Giới tính
BS001	Nguyễn Văn Hưng	Bác sĩ thực tập	Nội	987654321	15/05/1990	Long Xuyên	Nam
BS002	Trần Thị Diễm	Bác sĩ thực tập	Ngoại	0901234567	20/11/1980	Chợ Mới	Nữ
BS003	Hoàng Xuân Vinh	Bác sĩ thực tập	Ngoại	0912345678	10/02/1995	Bình Mỹ- An Giang	Nam
BS004	Phạm Huỳnh Thúy An	Bác sĩ chính thức	Ngoại	0968889999	25/07/1997	Đồng Xuyên- Long Xuyên-	Nữ
BS005	Trương Đinh Hoàng	Bác sĩ thực tập	Nội	0977665544	01/01/2000	Bình Khánh- An Giang	Nam

Hình 21. Demo chức năng “Thêm”

Quản lý Bác sĩ - Quyền: ADMIN

Hệ thống (admin) Quản lý

QUẢN LÝ BÁC SĨ

Thông tin							
Mã bác sĩ:		Điện thoại:		Ngày sinh:	29/11/2025	Địa chỉ:	
Họ và tên:		Chức vụ:	Bác sĩ chính thức	Địa chỉ:		Giới tính:	
Chuyên khoa:	Nội						
Thêm		Xóa	Sửa	Lưu	Làm mới		

Mã bác sĩ	Họ và tên	Chức vụ	Chuyên khoa	Điện thoại	Ngày sinh	Địa chỉ	Giới tính
BS001	Nguyễn Văn Hưng	Bác sĩ thực tập	Nội	987654321	15/05/1990	Long Xuyên	Nam
BS002	Trần Thị Diễm	Bác sĩ thực tập	Ngoại	0901234567	20/11/1980	Chợ Mới	Nữ
BS003	Hoàng Xuân Vinh	Bác sĩ thực tập	Ngoại	0912345678	10/02/1995	Bình Mỹ- An Giang	Nam
BS004	Phạm Huỳnh Thúy An	Bác sĩ chính thức	Ngoại	0968889999	25/07/1997	Đồng Xuyên- Long Xuyên-	Nữ
BS005	Trương Đình Hoàng	Bác sĩ thực tập	Nội	0977665544	01/01/2000	Bình Khánh- An Giang	Nam
BS006	Tester	Bác sĩ thực tập	Ngoại	0246813579	15/05/1990	Đồng Nai	Nam

Hình 22. Kết quả demo chức năng “Thêm”

Quản lý Bác sĩ - Quyền: ADMIN

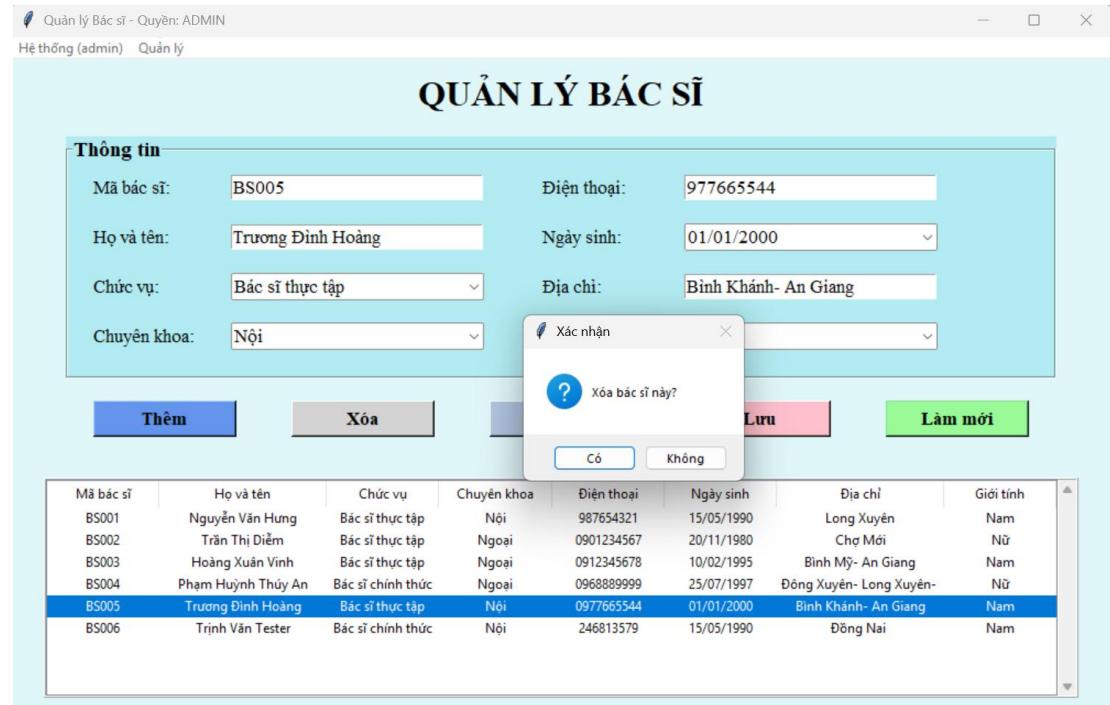
Hệ thống (admin) Quản lý

QUẢN LÝ BÁC SĨ

Thông tin							
Mã bác sĩ:	BS006	Điện thoại:	246813579	Ngày sinh:	15/05/1990	Địa chỉ:	
Họ và tên:	Trịnh Văn Tester	Chức vụ:	Bác sĩ chính thức	Địa chỉ:	Đồng Nai	Giới tính:	
Chuyên khoa:	Nội						
Thêm		Xóa	Sửa	Lưu	Làm mới		

Mã bác sĩ	Họ và tên	Chức vụ	Chuyên khoa	Điện thoại	Ngày sinh	Địa chỉ	Giới tính
BS001	Nguyễn Văn Hưng	Bác sĩ thực tập	Nội	987654321	15/05/1990	Long Xuyên	Nam
BS002	Trần Thị Diễm	Bác sĩ thực tập	Ngoại	0901234567	20/11/1980	Chợ Mới	Nữ
BS003	Hoàng Xuân Vinh	Bác sĩ thực tập	Ngoại	0912345678	10/02/1995	Bình Mỹ- An Giang	Nam
BS004	Phạm Huỳnh Thúy An	Bác sĩ chính thức	Ngoại	0968889999	25/07/1997	Đồng Xuyên- Long Xuyên-	Nữ
BS005	Trương Đình Hoàng	Bác sĩ thực tập	Nội	0977665544	01/01/2000	Bình Khánh- An Giang	Nam
BS006	Trịnh Văn Tester	Bác sĩ chính thức	Nội	246813579	15/05/1990	Đồng Nai	Nam

Hình 23. Kết quả demo chức năng “Sửa”



Hình 24. Demo chức năng “Xóa”



Hình 25. Kết quả demo chức năng “Xóa”

Quản lý Bác sĩ - Quyền: ADMIN

Hệ thống (admin) Quản lý

QUẢN LÝ BÁC SĨ

Thông tin								
Mã bác sĩ:		Điện thoại:						
Họ và tên:		Ngày sinh:	29/11/2025					
Chức vụ:	Bác sĩ chính thức	Địa chỉ:						
Chuyên khoa:	Nội							
<input type="button" value="Thêm"/> <input type="button" value="Xóa"/> <input type="button" value="Làm mới"/>								
<div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: auto;"> Dữ liệu đã được lưu và cập nhật từ hệ thống! </div>								
Mã bác sĩ	Họ và tên	Chức vụ	Chuyên khoa	Điện thoại	Ngày sinh	Địa chỉ	Giới tính	
BS001	Nguyễn Văn Hưng	Bác sĩ thực tập	Nội	987654321	15/05/1990	Long Xuyên	Nam	
BS002	Trần Thị Diễm	Bác sĩ thực tập	Ngoại	0901234567	20/11/1980	Chợ Mới	Nữ	
BS003	Hoàng Xuân Vinh	Bác sĩ thực tập	Ngoại	0912345678	10/02/1995	Bình Mỹ- An Giang	Nam	
BS004	Phạm Huỳnh Thúy An	Bác sĩ chính thức	Ngoại	0968889999	25/07/1997	Đồng Xuyên- Long Xuyên-	Nữ	
BS006	Trịnh Văn Tester	Bác sĩ chính thức	Nội	246813579	15/05/1990	Đồng Nai	Nam	

Hình 26. Kết quả demo chức năng “Lưu”

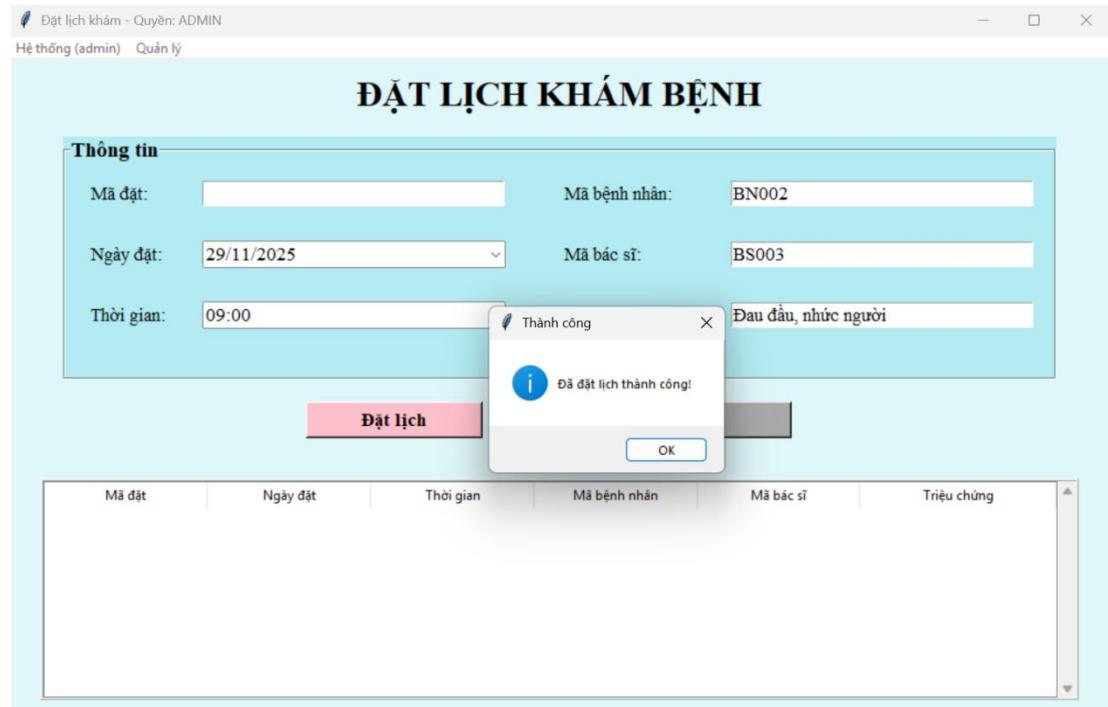
Đặt lịch khám - Quyền: ADMIN

Hệ thống (admin) Quản lý

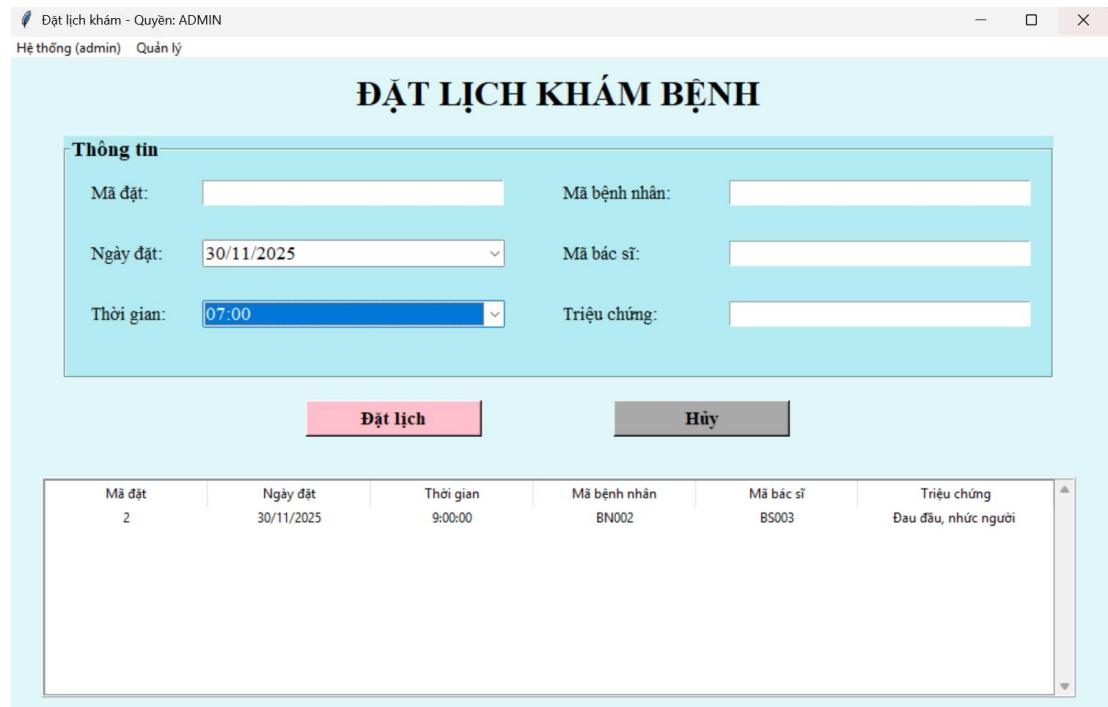
ĐẶT LỊCH KHÁM BỆNH

Thông tin					
Mã đặt:		Mã bệnh nhân:	BN002		
Ngày đặt:	29/11/2025	Mã bác sĩ:	BS003		
Thời gian:	09:00	Triệu chứng:	Đau đầu, nhức người		
<input type="button" value="Đặt lịch"/> <input type="button" value="Hủy"/>					
Mã đặt	Ngày đặt	Thời gian	Mã bệnh nhân	Mã bác sĩ	Triệu chứng

Hình 27. Demo chức năng “Đặt lịch”



Hình 28. Kết quả demo chức năng “Đặt lịch”



Hình 29. Kết quả demo chức năng “Đặt lịch” sau khi thêm bệnh nhân

e. Kết quả quản lý khám bệnh

The screenshot shows a software application window titled "KHÁM BỆNH". The interface is divided into several sections:

- Đặt lịch (Chờ khám)**: A table showing an appointment record:

Mã đặt	Mã BN	Mã BS	Triệu chứng
1	BN002	BS003	Đau đầu, nhức người
- Thuốc**: Prescription details:

Mã thuốc	T001	
Tên thuốc	Paracetamol 500mg	
Số lượng	3	
Đơn giá	10000	VND
- Khám bệnh**: Patient information:

Mã bác sĩ	BS003
Mã bệnh nhân	BN002
Triệu Chứng	Đau đầu, nhức người
Chuẩn đoán	Sốt nhẹ
- Buttons**:
 - Xuất phiếu khám (pink button)
 - Hủy (grey button)

Hình 30. Demo nhập chẩn đoán và kê đơn thuốc

PHIẾU KHÁM & ĐƠN THUỐC

Ngày: 30/11/2025 00:54:24

THÔNG TIN BỆNH NHÂN

Mã bệnh nhân : BN002
Mã bác sĩ : BS003

CHẨN ĐOÁN

Triệu chứng : Đau đầu, nhức người
Chẩn đoán : Sốt nhẹ

ĐƠN THUỐC & THANH TOÁN

Tên thuốc : Paracetamol 500mg
Số lượng : 3
Đơn giá : 10,000 VNĐ

TỔNG CỘNG: 30,000 VNĐ

In Phiếu / Đóng

Hình 31. Demo Phiếu khám khi chọn vào nút Xuất phiếu khám

3. CODE thực hiện

a. Code cơ sở dữ liệu MySQL

```
1 • DROP DATABASE IF EXISTS QuanLyBenhNhan;
2 • CREATE DATABASE QuanLyBenhNhan
3   DEFAULT CHARACTER SET utf8mb4
4   COLLATE utf8mb4_unicode_ci;
5 • USE QuanLyBenhNhan;
6
7 • ◇ CREATE TABLE BENHNHAN(
8   MABN VARCHAR(5) PRIMARY KEY,
9   HOTEN VARCHAR(100) NOT NULL,
10  DIENTHOAI VARCHAR(10) UNIQUE NOT NULL,
11  NGAYSINH DATE,
12  GIOITINH VARCHAR(3) CHECK(GIOITINH IN ('Nam','Nữ')),
13  DIACHI VARCHAR(100),
14  MA_BHYT VARCHAR(30)
15 ) ENGINE=InnoDB;
16
17
18 • ◇ CREATE TABLE BACSI(
19   MABS varchar(5) PRIMARY KEY,
20   TENBS varchar(50),
21   CHUCVU VARCHAR(25) CHECK (CHUCVU IN ('Bác sĩ chính thức','Bác sĩ thực tập')),
22   KHOA VARCHAR(10) CHECK (KHOA IN ('Nội','Ngôại')),
```



```
23   DIENTHOAI VARCHAR(10) UNIQUE NOT NULL,
24   NGAYSINH DATE,
25   DIACHI VARCHAR(100),
26   GIOITINH VARCHAR(3) CHECK(GIOITINH IN ('Nam','Nữ')),
27   TAIKHOAN VARCHAR(10),
28   MATKHAU VARCHAR(5)
29 ) ENGINE=InnoDB;
30
31
32 • ◇ CREATE TABLE NHANVIEN(
33   MANV VARCHAR(5) PRIMARY KEY,
34   TENVN VARCHAR(50),
35   TAIKHOAN VARCHAR(10) UNIQUE NOT NULL,
36   MATKHAU VARCHAR(5)
37 ) ENGINE=InnoDB;
```



```
38
39
40 • ◇ CREATE TABLE THUOC(
41   MATHUOC VARCHAR(10) PRIMARY KEY,
42   TENTHUOC VARCHAR(100),
43   SOLUONG VARCHAR(5),
44   DONVITINH VARCHAR(10) CHECK (DONVITINH IN('Hộp','Vi','Viên','Ông','Gói','Tuýp','Lọ','Chai')),
```



```
45   DONGIA FLOAT
46 ) ENGINE=InnoDB;
47
48
49 • ◇ CREATE TABLE DATLICH(
50   MADAT INT AUTO_INCREMENT PRIMARY KEY,
51   MABN VARCHAR(5) NOT NULL,
52   MABS VARCHAR(5),
53   NGAYDAT DATE,
54   THOIGIAN time,
55   TRIEUCHUNG VARCHAR(100),
56   FOREIGN KEY (MABN) REFERENCES BENHNHAN(MABN) ON DELETE CASCADE,
57   FOREIGN KEY (MABS) REFERENCES BACSI(MABS) ON DELETE CASCADE
58 ) ENGINE=InnoDB;
```



```
59
60
61 • ◇ CREATE TABLE HOADON_KHAMBNH(
62   MAHD INT AUTO_INCREMENT PRIMARY KEY,
63   MABN VARCHAR(5),
64   MABS VARCHAR(5),
65   NGAYKHAM DATE,
66   MATHUOC VARCHAR(10),
```

```

67     TRIEUCHUNG VARCHAR(100),
68     FOREIGN KEY (MABN) REFERENCES BENHNHAN(MABN) ON DELETE CASCADE,
69     FOREIGN KEY (MABS) REFERENCES BACSI(MABS) ON DELETE CASCADE,
70     FOREIGN KEY (MATHUOC) REFERENCES THUOC(MATHUOC) ON DELETE CASCADE
71 ) ENGINE=InnoDB;
72
73
74 •   INSERT INTO NHANVIEN (MANV, TENNV, TAIKHOAN, MATKHAU) VALUES
75     ('NV001', 'Trần Thúy Thành Thúy', 'NV01', '00000');
76
77
78 •   INSERT INTO BACSI (MABS, TENBS, CHUCVU, KHOA , DIENTHOAI, GIOITINH, DIACHI, NGAYSINH, TAIKHOAN, MATKHAU) VALUES
79     ('BS001', 'Nguyễn Văn Hưng', 'Bác sĩ chính thức','Nội', '0987654321', 'Nam', 'Long Xuyên', '1990-05-15', 'BSCT1', 'CT001'),
80     ('BS002', 'Trần Thị Diễm', 'Bác sĩ thực tập','Ngoại', '0901234567', 'Nữ', 'Chợ Mới', '1980-11-20', 'BSTT0', 'TT001'),
81     ('BS003', 'Hoàng Xuân Vinh', 'Bác sĩ thực tập','Ngoại', '0912345678', 'Nam', 'Bình Mỹ- An Giang', '1995-02-10', 'BSTT1', 'TT002'),
82     ('BS004', 'Phạm Huỳnh Thúy An', 'Bác sĩ chính thức','Ngoại', '0968889999', 'Nữ', 'Đông Xuyên- Long Xuyên- An Giang', '1997-07-15'),
83     ('BS005', 'Trương Đình Hoàng', 'Bác sĩ thực tập','Nội', '0977665544', 'Nam', 'Bình Khánh- An Giang', '2000-01-01', 'BSTT2', 'TT003');
84
85
86 •   INSERT INTO BENHNHAN (MABN, HOTEN, DIENTHOAI, GIOITINH, NGAYSINH, DIACHI, MA_BHYT) VALUES
87     ('BN001', 'Trần Hoàng Quân', '0987654921', 'Nam', '1998-05-20', 'Long Xuyên', '1014523125'),
88     ('BN002', 'Phạm Kiều Thu', '0901234767', 'Nữ', '1999-03-08', 'Châu Đốc', '1097583501'),
89
90     ('BN003', 'Nguyễn Văn Thể', '0912344678', 'Nam', '2002-02-02', 'Bình Mỹ- An Giang', '1062378926'),
91     ('BN004', 'Nguyễn Ngọc Châu', '0968889999', 'Nữ', '1997-07-30', 'Chợ Mới- An Giang', '1035773290'),
92     ('BN005', 'Lê Quỳnh Thư', '0977665524', 'Nữ', '2002-01-19', 'Châu Đốc- An Giang', '1025644893');
93
94 •   INSERT INTO THUOC (MATHUOC, TENTHUOC, SOLUONG, DONVITINH, DONGIA) VALUES
95     ('T001', 'Paracetamol 500mg', '100', 'Viên', 1000),
96     ('T002', 'Panadol Extra', '50', 'Vi', 1500),
97     ('T003', 'Amoxicillin 500mg', '200', 'Viên', 2500),
98     ('T004', 'Vitamin C', '100', 'Hộp', 2000),
99     ('T005', 'Berberin', '50', 'Lọ', 5000),
100    ('T006', 'Efferalgan 500mg', '200', 'Viên', 3000),
101    ('T007', 'Ibuprofen 400mg', '100', 'Viên', 2000),
102    ('T008', 'Cephalexin 500mg', '150', 'Viên', 1500),
103    ('T009', 'Thuốc ho Prospan', '20', 'Chai', 85000),
104    ('T010', 'Smecta (Thuốc tiêu hóa)', '50', 'Gói', 4000),
105    ('T011', 'Omeprazol 20mg (Đà dày)', '100', 'Viên', 2500),
106    ('T012', 'C sủi Plussz', '30', 'Tuýp', 35000),
107    ('T013', 'Nuốc muối sinh lý', '50', 'Chai', 5000),
108    ('T014', 'Salonpas Gel', '20', 'Tuýp', 45000),
109    ('T015', 'Hoạt huyết dưỡng não', '40', 'Hộp', 90000);
110
111 •   SELECT * FROM BACSI;
112 •   SELECT * FROM BENHNHAN;
113 •   SELECT * FROM THUOC;
114
115 •   ALTER TABLE HOADON_KHAMBENH ADD COLUMN CHUANDOAN VARCHAR(255);

```

b. Code kết nối MySQL

```

8  # Cấu hình kết nối Database
9  db_connect = {
10    'host': 'localhost',
11    'user': 'root',
12    'password': 'han1234',
13    'database': 'QuanLyBenhNhan'
14 }
15 def get_connection():
16     try:
17         conn = mysql.connector.connect(**db_connect)
18         return conn
19     except mysql.connector.Error as err:
20         messagebox.showerror("Lỗi kết nối", f"Không thể kết nối Database: {err}")
21         return None
22

```

c. Thư viện

```
1  from tkinter import *
2  from tkinter import ttk
3  from tkinter import messagebox
4  from datetime import datetime
5  from tkcalendar import DateEntry
6  import mysql.connector
7
```

d. Code đăng nhập

```
35
36
37 class LoginSystem:
38
39     # Giao diện chính
40     def __init__(self, root):
41         self.root = root
42         self.root.title("Đăng nhập hệ thống")
43         w, h = 600, 400
44         sw = root.winfo_screenwidth();
45         sh = root.winfo_screenheight()
46         self.root.geometry(f"{w}x{h}+{((sw//2)-(w//2))+((sh//2)-(h//2))}")
47         self.root.resizable(False, False)
48         self.root.config(bg="#F0F8FF")
49
50     # Biến lưu thông tin đăng nhập
51     self.username = StringVar();
52     self.password = StringVar()
53
54     # Tạo khung đăng nhập
55     Label(self.root, text="HỆ THỐNG Y TẾ", font=("Times New Roman", 20, "bold"), bg="#F0F8FF").place(x=180, y=30)
56     frame = Frame(self.root, bg="#B2EBF2", width=450, height=200); frame.place(x=75, y=90)
57     Label(frame, text="Đăng nhập", font=("Times New Roman", 14, "bold"), bg="#B2EBF2").place(x=10, y=10)
58
59     # Các thành phần nhập liệu
60     Label(frame, text="Tài khoản:", font=("Times New Roman", 13), bg="#B2EBF2").place(x=50, y=60)
61     Entry(frame, textvariable=self.username, font=("Times New Roman", 13), width=25).place(x=150, y=60)
62
63
64     Label(frame, text="Mật khẩu:", font=("Times New Roman", 13), bg="#B2EBF2").place(x=50, y=110)
65     e_pass = Entry(frame, textvariable=self.password, font=("Times New Roman", 13), width=25, show="*");
66     e_pass.place(x=150, y=110)
67     e_pass.bind('<Return>', lambda e: self.login())
68
69     # Nút đăng nhập và thoát
70     Button(self.root, text="Đăng nhập", command=self.login, font=("Times New Roman", 12, "bold"),
71             bg="#FFC0CB", width=12).place(x=180, y=320)
72     Button(self.root, text="Thoát", command=root.quit, font=("Times New Roman", 12, "bold"),
73             bg="#A9A9A9", width=12).place(x=350, y=320)
74
75     # Hàm xử lý đăng nhập
76     def login(self):
77         user = self.username.get()
78         pwd = self.password.get()
79
80         if user == "admin" and pwd == "123":
81             messagebox.showinfo("Thành công", "Đăng nhập với quyền Quản trị viên")
82             self.open_main("admin")
83             return
84
85         conn = get_connection()
86         if conn:
87             cursor = conn.cursor()
88             sql_nv = "SELECT * FROM NHANVIEN WHERE TAIKHOAN = %s AND MATKHAU = %s"
89             cursor.execute(sql_nv, (user, pwd))
90             if cursor.fetchone():
```

```

89         conn.close();
90         messagebox.showinfo("Thành công", f"Đăng nhập với quyền Nhân viên!")
91         self.open_main("nhanvien");
92         return
93
94     sql_bs = "SELECT * FROM BACSI WHERE TAIKHOAN = %s AND MATKHAU = %s"
95     cursor.execute(sql_bs, (user, pwd))
96     if cursor.fetchone():
97         conn.close();
98         messagebox.showinfo("Thành công", f"Đăng nhập với quyền Bác sĩ!")
99         self.open_main("bacsi");
100        return
101
102    conn.close()
103    messagebox.showerror("Lỗi", "Sai thông tin đăng nhập!")
104 # Mở giao diện sau khi đăng nhập thành công
105 def open_main(self, role):
106     self.root.destroy()
107     new_root = Tk()
108     QuanLyBenhNhan(new_root, role)
109     new_root.mainloop()
110

```

e. Code quản lý bệnh nhân

```

111 class QuanLyBenhNhan:
112     # FROM QUẢN LÝ BỆNH NHÂN
113
114     # Tạo giao diện chính
115     def __init__(self, root, role):
116         self.root = root
117         self.role = role
118         self.root.title(f"Quản lý bệnh nhân - Quyền: {self.role.upper()}")
119         self.root.geometry("1000x600")
120         self.center_window(1000, 600)
121         self.root.config(bg="#E0F7FA")
122         self.create_menu()
123
124     # Giao diện quản lý bệnh nhân
125     style = ttk.Style(); style.theme_use("default")
126     Label(self.root, text="QUẢN LÝ BỆNH NHÂN", font=("Times New Roman", 24, "bold"), bg="#E0F7FA").pack(pady=10)
127     frame_info = LabelFrame(self.root, text="Thông tin", font=("Times New Roman", 14, "bold"), bg="#B2EBF2", bd=2, relief=GROOVE)
128     frame_info.place(x=50, y=70, width=900, height=220)
129
130     # Các thành phần nhập liệu
131     self.entries = {}
132     labels_left = ["Mã bệnh nhân:", "Họ và tên:", "Mã BHYT:", "Giới tính:"]
133     labels_right = ["Địa chỉ:", "Số điện thoại:", "Ngày sinh:"]
134
135     # Diền các nhãn và ô nhập liệu- bên trái
136     for i, text in enumerate(labels_left):
137         Label(frame_info, text=text, font=("Times New Roman", 13), bg="#B2EBF2").grid(row=i, column=0, padx=20, pady=10, sticky="w")
138         if text == "Giới tính:":
139             cbo = ttk.Combobox(frame_info, values=["Nam", "Nữ"], font=("Times New Roman", 13), state="readonly", width=23)
140             cbo.grid(row=i, column=1, padx=10, pady=10);
141             cbo.current(0);
142             self.entries["Giới tính"] = cbo
143         else:
144             entry = Entry(frame_info, font=("Times New Roman", 13), width=25)
145             entry.grid(row=i, column=1, padx=10, pady=10);
146             self.entries[text.replace(":", "")] = entry
147
148     # Diền các nhãn và ô nhập liệu- bên phải
149     for i, text in enumerate(labels_right):
150         Label(frame_info, text=text, font=("Times New Roman", 13), bg="#B2EBF2").grid(row=i, column=2, padx=40, pady=10, sticky="w")
151         if text == "Ngày sinh:":
152             date_entry = DateEntry(frame_info, width=23, background='darkblue',
153                                     foreground='white', borderwidth=2, date_pattern='dd/mm/yyyy', font=("Times New Roman", 13))
154             date_entry.grid(row=i, column=3, padx=10, pady=10);
155             self.entries["Ngày sinh"] = date_entry
156         else:
157             entry = Entry(frame_info, font=("Times New Roman", 13), width=25)
158             entry.grid(row=i, column=3, padx=10, pady=10);
159             self.entries[text.replace(":", "")] = entry
160
161
162
163

```

```

164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189

```

Khung nút chức năng
frame_buttons = Frame(self.root, bg="#E0F7FA")
frame_buttons.place(x=50, y=300, width=900, height=60)

Nút chức năng
for i in range(5):
 frame_buttons.grid_columnconfigure(i, weight=1)

btn_configs = [
 ("Thêm", "#6495ED", self.them),
 ("Xóa", "#D3D3D3", self.xoa),
 ("Sửa", "#B0C4DE", self.sua),
 ("Lưu", "#FFCCBC", self.luu),
 ("Làm mới", "#98FB98", self.clear_form)
]

for i, (text, color, cmd) in enumerate(btn_configs):
 Button(frame_buttons, text=text, command=cmd, font=("Times New Roman", 13, "bold"),
 bg=color, width=12, bd=2, relief=RIDGE).grid(row=0, column=i, padx=5, pady=10)

Khung bảng dữ liệu
frame_table = Frame(self.root, bd=2, relief=RIDGE)
frame_table.place(x=30, y=380, width=940, height=200)
scroll_y = Scrollbar(frame_table, orient=VERTICAL);
scroll_y.pack(side=RIGHT, fill=Y)

```

190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

```

columns = ("Mã bệnh nhân", "Họ và tên", "Giới tính", "Mã BHYT", "Địa chỉ", "Số điện thoại", "Ngày sinh",)
self.tree = ttk.Treeview(frame_table, columns=columns, show="headings", yscrollcommand=scroll_y.set)
scroll_y.config(command=self.tree.yview)

Định dạng cột
for col in columns:
 self.tree.heading(col, text=col);
 self.tree.column(col, width=150 if col in ["Họ và tên", "Địa chỉ"] else 100, anchor="center")
 self.tree.pack(fill=BOTH, expand=True)
 self.tree.bind("<>TreeviewSelect>", self.on_tree_select)

self.load_data()

Hàm căn giữa cửa sổ
def center_window(self, width, height):
 screen_width = self.root.winfo_screenwidth();
 screen_height = self.root.winfo_screenheight();
 x = (screen_width // 2) - (width // 2);
 y = (screen_height // 2) - (height // 2);
 self.root.geometry(f"{width}x{height}+{x}+{y}")

Hàm tạo menu
def create_menu(self):
 menubar = Menu(self.root);
 self.root.config(menu=menubar)
 sys_menu = Menu(menubar, tearoff=0)

```

216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242

```

sys_menu.add_command(label="Đang xuất", command=self.dang_xuat);
sys_menu.add_command(label="Thoát", command=self.root.quit)
menubar.add_cascade(label=f"Hệ thống ({self.role})", menu=sys_menu)

man_menu = Menu(menubar, tearoff=0)
man_menu.add_command(label="Quản lý Bệnh Nhân", state="disabled")

if self.role == "admin":
 man_menu.add_command(label="Quản lý Bác Sĩ", command=self.open_quan_ly_bac_si)
else:
 man_menu.add_command(label="Quản lý Bác Sĩ (Admin only)", state="disabled")

if self.role in ["admin", "nhanvien"]:
 man_menu.add_command(label="Đặt lịch khám", command=self.open_quan_ly_dat_lich)
else:
 man_menu.add_command(label="Đặt lịch khám (No Access)", state="disabled")

if self.role in ["admin", "bacsi"]:
 man_menu.add_command(label="Khám bệnh", command=self.open_kham_benh)
else:
 man_menu.add_command(label="Khám bệnh (No Access)", state="disabled")

menubar.add_cascade(label="Quản lý", menu=man_menu)

Hàm tái dữ liệu từ database vào bảng
def load_data(self):
 for row in self.tree.get_children():

```

243         self.tree.delete(row)
244     conn = get_connection()
245     if conn:
246         cursor = conn.cursor()
247         cursor.execute("SELECT * FROM BENHNHAN")
248         for row in cursor.fetchall():
249             self.tree.insert("", END, values=(
250                 row[0],
251                 row[1],
252                 row[4],
253                 row[6],
254                 row[5],
255                 row[2],
256                 date_db_to_ui(row[3])
257             ))
258         conn.close()
259
260     # Hàm thêm bệnh nhân
261     def them(self):
262         vals = {
263             "MABN": self.entries["Mã bệnh nhân"].get(),
264             "HOTEN": self.entries["Họ và tên"].get(),
265             "GIOITINH": self.entries["Giới tính"].get(),
266             "MA_BHYT": self.entries["Mã BHYT"].get(),
267             "DIACHI": self.entries["Địa chỉ"].get(),
268             "DIENTHOAI": self.entries["Số điện thoại"].get(),
269             "NGAYSINH": date_ui_to_db(self.entries["Ngày sinh"].get())
270         }
271         if vals["MABN"] == "":
272             messagebox.showerror("Lỗi", "Vui lòng nhập Mã bệnh nhân");
273             return
274         conn = get_connection()
275         if conn:
276             try:
277                 cursor = conn.cursor()
278                 sql = "INSERT INTO BENHNHAN (MABN, HOTEN, GIOITINH, MA_BHYT, DIACHI, DIENTHOAI, NGAYSINH) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
279                 cursor.execute(sql, tuple(vals.values()))
280                 conn.commit()
281                 self.load_data()
282                 self.clear_form()
283                 messagebox.showinfo("Thành công", "Đã thêm!")
284             except mysql.connector.Error as err:
285                 messagebox.showerror("Lỗi Database", f"Chi tiết: {err}")
286             finally:
287                 conn.close()
288
289     # Hàm xóa bệnh nhân
290     def xoa(self):
291         sel = self.tree.selection()
292         if sel:
293             mabn = self.tree.item(sel[0])['values'][0]
294             if messagebox.askyesno("Xác nhận", f"Xóa bệnh nhân {mabn}?"):

```

```

295     conn = get_connection()
296     if conn:
297         try:
298             conn.cursor().execute("DELETE FROM BENHNHAN WHERE MABN = %s", (mabn,))
299             conn.commit();
300             self.load_data();
301             self.clear_form()
302             messagebox.showinfo("Thành công", "Đã xóa!")
303         except Exception as e:
304             messagebox.showerror("Lỗi", str(e))
305         conn.close()
306
307     # Hàm sửa thông tin bệnh nhân
308     def sua(self):
309         sel = self.tree.selection()
310         if not sel:
311             return
312         vals = [self.entries["Họ và tên"].get(),
313                 self.entries["Giới tính"].get(),
314                 self.entries["Mã BHYT"].get(),
315                 self.entries["Địa chỉ"].get(),
316                 self.entries["Số điện thoại"].get(),
317                 date_ui_to_db(self.entries["Ngày sinh"].get()),
318                 self.entries["Mã bệnh nhân"].get()]
319         conn = get_connection()
320         if conn:
321             try:

```

```

322                 conn.cursor().execute("UPDATE BENHNHAN SET HOTEN=%s, GIOITINH=%s, MA_BHYT=%s, DIACHI=%s, DIENTHOAI=%s, NGAYSINH=%s"
323                                         | | | | | "WHERE MABN=%s", vals)
324                 conn.commit();
325                 self.load_data();
326                 messagebox.showinfo("Thông Báo", "Đã cập nhật!")
327             except Exception as e:
328                 messagebox.showerror("Lỗi", str(e))
329             conn.close()
330
331     # Hàm lưu dữ liệu
332     def luu(self):
333         self.load_data()
334         messagebox.showinfo("Thông báo", "Dữ liệu đã được lưu và cập nhật từ hệ thống!")
335
336     # Hàm làm mới form nhập liệu
337     def clear_form(self):
338         for k, v in self.entries.items():
339             if isinstance(v, ttk.Combobox):
340                 v.current(0)
341             elif isinstance(v, DateEntry):
342                 v.set_date(datetime.now())
343             else:
344                 v.delete(0, END)
345
346     # Hàm xử lý khi chọn một dòng trong bảng
347     def on_tree_select(self, event):

```

```

348     try:
349         sel = self.tree.selection()[0];
350         val = self.tree.item(sel)['values']
351         self.entries["Mã bệnh nhân"].delete(0, END);
352         self.entries["Mã bệnh nhân"].insert(0, val[0])
353         self.entries["Họ và tên"].delete(0, END);
354         self.entries["Họ và tên"].insert(0, val[1])
355         self.entries["Giới tính"].set(val[2])
356         self.entries["Mã BHYT"].delete(0, END);
357         self.entries["Mã BHYT"].insert(0, val[3])
358         self.entries["Địa chỉ"].insert(0, val[4])
359         self.entries["Số điện thoại"].delete(0, END);
360         self.entries["Số điện thoại"].insert(0, str(val[5]))
361         try:
362             self.entries["Ngày sinh"].set_date(datetime.strptime(val[6], '%d/%m/%Y'))
363         except:
364             pass
365         except:
366             pass
367         pass
368
369     # Hàm điều hướng menu
370     def dang_xuat(self):
371         self.root.destroy();
372         root = Tk();
373         LoginSystem(root);

```

```

374     |     root.mainloop()
375     | def thoat_ung_dung(self):
376     |     self.root.quit()
377     | def open_quan_ly_dat_lich(self):
378     |     self.root.destroy();
379     |     new_root = Tk();
380     |     QuanLyDatLich(new_root, self.role);
381     |     new_root.mainloop()
382     | def open_quan_ly_bac_si(self):
383     |     self.root.destroy();
384     |     new_root = Tk();
385     |     QuanLyBacSi(new_root, self.role);
386     |     new_root.mainloop()
387     | def open_kham_benh(self):
388     |     self.root.destroy();
389     |     new_root = Tk();
390     |     QuanLyKhamBinh(new_root, self.role);
391     |     new_root.mainloop()
392

```

f. Code quản lý bác sĩ

```

392
393
394 class QuanLyBacSi:
395
396     # Tạo giao diện chính
397     def __init__(self, root, role):
398         self.root = root;
399         self.role = role
400         self.root.title("Quản lý Bác sĩ - Quyền: {" + self.role.upper() + "}")
401         self.root.geometry("1000x600")
402         self.center_window(1000, 600)
403         self.root.config(bg="#E0F7FA")
404         self.create_menu()
405
406     # Giao diện quản lý bác sĩ
407     Label(self.root, text="QUẢN LÝ BÁC SĨ", font=("Times New Roman", 24, "bold"), bg="#E0F7FA").pack(pady=10)
408     frame_info = LabelFrame(self.root, text="Thông tin", font=("Times New Roman", 14, "bold"), bg="#B2EBF2", bd=2, relief=GROOVE)
409     frame_info.place(x=50, y=70, width=900, height=220)
410
411     # Các thành phần nhập liệu
412     self.entries = {}
413     labels_left = ["Mã bác sĩ:", "Họ và tên:", "Chức vụ:", "Chuyên khoa:"]
414     labels_right = ["Điện thoại:", "Ngày sinh:", "Địa chỉ:", "Giới tính:"]
415
416     # Diển các nhãn và ô nhập liệu- bên trái
417     for i, text in enumerate(labels_left):
418         Label(frame_info, text=text, font=("Times New Roman", 13), bg="#B2EBF2").grid(row=i, column=0, padx=20, pady=10, sticky="w")
419         if text == "Chức vụ:":
420             cbo = ttk.Combobox(frame_info, values=["Bác sĩ chính thức", "Bác sĩ thực tập"], font=("Times New Roman", 13), state="readonly", width=23)
421             cbo.grid(row=i, column=1, padx=10, pady=10);
422             cbo.current(0);
423             self.entries["Chức vụ"] = cbo
424         elif text == "Chuyên khoa:":
425             cbo = ttk.Combobox(frame_info, values=["Nội", "Ngoại"], font=("Times New Roman", 13), state="readonly", width=23)
426             cbo.grid(row=i, column=1, padx=10, pady=10);
427             cbo.current(0);
428             self.entries["Chuyên khoa"] = cbo
429         else:
430             entry = Entry(frame_info, font=("Times New Roman", 13), width=25);
431             entry.grid(row=i, column=1, padx=10, pady=10);
432             self.entries[text.replace(":", "")] = entry
433
434     # Diển các nhãn và ô nhập liệu- bên phải
435     for i, text in enumerate(labels_right):
436         Label(frame_info, text=text, font=("Times New Roman", 13),
437               bg="#B2EBF2").grid(row=i, column=2, padx=40, pady=10, sticky="w")
438
439
440
441
442
443
444

```

```

445     if text == "Ngày sinh:":
446         date_entry = DateEntry(frame_info, width=23, background='darkblue', foreground='white', borderwidth=2,
447                               date_pattern='dd/mm/yyyy', font=("Times New Roman", 13))
448         date_entry.grid(row=i, column=3, padx=10, pady=10);
449         self.entries["Ngày sinh"] = date_entry
450     elif text == "Giới tính:":
451         cbo = ttk.Combobox(frame_info, values=["Nam", "Nữ"], font=("Times New Roman", 13), state="readonly", width=23)
452         cbo.grid(row=i, column=3, padx=10, pady=10);
453         cbo.current(0);
454         self.entries["Giới tính"] = cbo
455     else:
456         entry = Entry(frame_info, font=("Times New Roman", 13), width=25);
457         entry.grid(row=i, column=3, padx=10, pady=10);
458         self.entries[text.replace(":", "")] = entry
459
460     # Khung nút chức năng
461     frame_buttons = Frame(self.root, bg="#E0F7FA")
462     frame_buttons.place(x=50, y=300, width=900, height=60)
463
464     # Nút chức năng
465     for i in range(5):
466         frame_buttons.grid_columnconfigure(i, weight=1)
467
468     btn_configs = [
469         ("Thêm", "#6495ED", self.them),
470         ("Xóa", "#03D3D3", self.xoa),
471         ("Sửa", "#BEC4DE", self.sua),
472
473         ("Lưu", "#FFC0CB", self.luu),
474         ("Làm mới", "#98FB98", self.clear_form)
475     ]
476
477     for i, (text, color, cmd) in enumerate(btn_configs):
478         Button(frame_buttons, text=text, command=cmd, font=("Times New Roman", 13, "bold"),
479                bg=color, width=12, bd=2, relief=RIDGE).grid(row=0, column=i, padx=5, pady=10)
480
481     # Khung bảng dữ liệu
482     frame_table = Frame(self.root, bd=2, relief=RIDGE)
483     frame_table.place(x=30, y=380, width=940, height=200)
484     scroll_y = Scrollbar(frame_table, orient=VERTICAL);
485     scroll_y.pack(side=RIGHT, fill=Y)
486
487     columns = ("Mã bác sĩ",
488                "Họ và tên",
489                "Chức vụ",
490                "Chuyên khoa",
491                "Điện thoại",
492                "Ngày sinh",
493                "Địa chỉ",
494                "Giới tính")
495
496     self.tree = ttk.Treeview(frame_table, columns=columns, show="headings", yscrollcommand=scroll_y.set)
497     scroll_y.config(command=self.tree.yview)
498     for col in columns:
499         self.tree.heading(col, text=col);
500         self.tree.column(col, width=150 if col in ["Họ và tên", "Địa chỉ"] else 100, anchor="center")
501     self.tree.pack(fill=BOTH, expand=True)
502
503     self.tree.bind("<<TreeviewSelect>>", self.on_tree_select)
504
505     self.load_data()
506
507     # Hàm căn giữa cửa sổ
508     def center_window(self, width, height):
509         x = (self.root.winfo_screenwidth() // 2) - (width // 2);
510         y = (self.root.winfo_screenheight() // 2) - (height // 2)
511         self.root.geometry(f"{width}x{height}+{x}+{y}")
512
513     # Hàm tạo menu
514     def create_menu(self):
515         menubar = Menu(self.root);
516         self.root.config(menu=menubar)
517         sys_menu = Menu(menubar, tearoff=0)
518         sys_menu.add_command(label="Đăng xuất", command=lambda: self.logout())
519         sys_menu.add_command(label="Thoát", command=(parameter) self: Self@QuanLyBacSi)
520         menubar.add_cascade(label="Hệ thống ({self.role})", menu=sys_menu)
521         man_menu = Menu(menubar, tearoff=0)
522         man_menu.add_command(label="Quản lý Bệnh Nhân", command=self.open_quan_ly_benh_nhan)
523         man_menu.add_command(label="Quản lý Bác Sĩ", state="disabled")
524
525         if self.role in ["admin", "nhanvien"]:
526             man_menu.add_command(label="Đặt lịch khám", command=self.open_quan_ly_dat_lich)
527         else:
528             man_menu.add_command(label="Đặt lịch khám (No Access)", state="disabled")

```

```

524
525     if self.role in ["admin", "bacsi"]:
526         man_menu.add_command(label="Khám bệnh", command=self.open_kham_benh)
527     else:
528         man_menu.add_command(label="Khám bệnh (No Access)", state="disabled")
529
530     menubar.add_cascade(label="Quản lý", menu=man_menu)
531
532     # Hàm tải dữ liệu từ database vào bảng
533     def load_data(self):
534         for row in self.tree.get_children():
535             self.tree.delete(row)
536         conn = get_connection()
537         if conn:
538             cursor = conn.cursor()
539             cursor.execute("SELECT * FROM BACSI")
540             for row in cursor.fetchall():
541                 self.tree.insert("", END,
542                                 values=(row[0], row[1], row[2], row[3], row[4], date_db_to_ui(row[5]), row[6], row[7]))
543             conn.close()
544
545     # Hàm thêm bác sĩ
546     def them(self):
547         vals = { "MABS": self.entries["Mã bác sĩ"].get(),
548                  "TENBS": self.entries["Họ và tên"].get(),
549                  "CHUCVU": self.entries["Chức vụ"].get(),
550                  "KHOA": self.entries["Chuyên khoa"].get(),
551                  "DIENTHOAI": self.entries["Điện thoại"].get(),
552                  "NGAYSINH": date_ui_to_db(self.entries["Ngày sinh"].get()),
553                  "DIACHI": self.entries["Địa chỉ"].get(),
554                  "GIOITINH": self.entries["Giới tính"].get() }
555         if vals["MABS"] == "": messagebox.showerror("Lỗi", "Nhập Mã BS"); return
556         conn = get_connection()
557         if conn:
558             try:
559                 conn.cursor().execute("INSERT INTO BACSI (MABS, TENBS, CHUCVU, KHOA, DIENTHOAI, NGAYSINH, DIACHI, GIOITINH) \
560                                         VALUES (%s,%s,%s,%s,%s,%s,%s)", tuple(vals.values()))
561             conn.commit();
562             self.load_data();
563             self.clear_form()
564             messagebox.showinfo("Thành công", "Đã thêm!")
565         except Exception as e:
566             messagebox.showerror("Lỗi", str(e))
567             conn.close()
568
569     # Hàm xóa bác sĩ
570     def xoa(self):
571         sel = self.tree.selection()
572         if sel and messagebox.askyesno("Xác nhận", "Xóa bác sĩ này?"):
573             conn = get_connection()
574             if conn: conn.cursor().execute("DELETE FROM BACSI WHERE MABS=%s", (self.tree.item(sel[0])['values'][0],));
575             conn.commit();
576             conn.close()
577
578         self.load_data();
579         messagebox.showinfo("Thành công", "Đã xóa!")
580         self.clear_form()
581
582     # Hàm sửa thông tin bác sĩ
583     def sua(self):
584         sel = self.tree.selection()
585         if sel:
586             vals = [self.entries["Họ và tên"].get(),
587                     self.entries["Chức vụ"].get(),
588                     self.entries["Chuyên khoa"].get(),
589                     self.entries["Điện thoại"].get(),
590                     date_ui_to_db(self.entries["Ngày sinh"].get()),
591                     self.entries["Địa chỉ"].get(),
592                     self.entries["Giới tính"].get(),
593                     self.entries["Mã bác sĩ"].get()]
594         conn = get_connection()
595         if conn:
596             try:
597                 conn.cursor().execute("UPDATE BACSI SET TENBS=%s, CHUCVU=%s, KHOA=%s, DIENTHOAI=%s, NGAYSINH=%s, \
598                                         DIACHI=%s, GIOITINH=%s WHERE MABS=%s", vals);
599             conn.commit();
600             self.load_data()
601             messagebox.showinfo("Thông báo", "Đã cập nhật! ")
602         except Exception as e:
603             messagebox.showerror("Lỗi", str(e))

```

```

603         conn.close()
604
605     # Hàm lưu dữ liệu
606     def luu(self):
607         self.load_data()
608         messagebox.showinfo("Thông báo", "Dữ liệu đã được lưu và cập nhật từ hệ thống!")
609
610     # Hàm làm mới form nhập liệu
611     def clear_form(self):
612         for k, v in self.entries.items():
613             if isinstance(v, ttk.Combobox):
614                 v.current(0)
615             elif isinstance(v, DateEntry):
616                 v.set_date(datetime.now())
617             else:
618                 v.delete(0, END)
619
620     # Hàm xử lý khi chọn một dòng trong bảng
621     def on_tree_select(self, event):
622         try:
623             sel = self.tree.selection()[0];
624             val = self.tree.item(sel)['values']
625             self.entries["Mã bác sĩ"].delete(0, END);
626             self.entries["Mã bác sĩ"].insert(0, val[0])
627             self.entries["Họ và tên"].delete(0, END);
628             self.entries["Họ và tên"].insert(0, val[1])

```

```

629             self.entries["Chức vụ"].set(val[2]);
630             self.entries["Chuyên khoa"].set(val[3])
631             self.entries["Điện thoại"].delete(0, END);
632             self.entries["Điện thoại"].insert(0, str(val[4]))
633             try:
634                 self.entries["Ngày sinh"].set_date(datetime.strptime(val[5], '%d/%m/%Y'))
635             except:
636                 pass
637             self.entries["Địa chỉ"].delete(0, END);
638             self.entries["Địa chỉ"].insert(0, val[6])
639             self.entries["Giới tính"].set(val[7])
640             except:
641                 pass
642
643     # Hàm điều hướng menu
644     def dang_xuat(self):
645         self.root.destroy();
646         root = Tk();
647         LoginSystem(root);
648         root.mainloop()
649     def thoat_ung_dung(self):
650         self.root.quit()

```

```

651     def open_quan_ly_dat_lich(self):
652         self.root.destroy();
653         new_root = Tk();
654         QuanLyDatLich(new_root, self.role);
655         new_root.mainloop()
656     def open_quan_ly_benh_nhan(self):
657         self.root.destroy();
658         new_root = Tk();
659         QuanLyBenhNhan(new_root, self.role);
660         new_root.mainloop()
661     def open_kham_benh(self):
662         self.root.destroy();
663         new_root = Tk();
664         QuanLyKhamBenh(new_root, self.role);
665         new_root.mainloop()

```

g. Code quản lý đặt lịch khám bệnh

```

667
668
669 class QuanLyDatLich:
670
671     # Tạo giao diện chính
672     def __init__(self, root, role):
673         self.root = root;
674         self.role = role
675         self.root.title("Đặt lịch khám - Quyền: " + self.role.upper())
676         self.root.geometry("1000x600")
677         self.center_window(1000, 600)
678         self.root.config(bg="#E0F7FA")
679         self.create_menu()
680
681     # Giao diện quản lý đặt lịch khám
682     Label(self.root, text="ĐẶT LỊCH KHÁM BỆNH", font=("Times New Roman", 24, "bold"), bg="#E0F7FA").pack(pady=10)
683     frame_info = LabelFrame(self.root, text="Thông tin", font=("Times New Roman", 14, "bold"), bg="#B2EBF2", bd=2, relief=GROOVE)
684     frame_info.place(x=50, y=70, width=900, height=220)
685
686     # Các thành phần nhập liệu
687     self.entries = {}
688     labels_left = ["Mã đặt:", "Ngày đặt:", "Thời gian:"]
689     labels_right = ["Mã bệnh nhân:", ""]
690
691

```

```

692         "Mã bác sĩ:",  
693         "Triệu chứng:")  
694  
695     # Điền các nhãn và ô nhập liệu- bên trái  
696     for i, text in enumerate(labels_left):  
697         Label(frame_info, text=text, font=("Times New Roman", 13),  
698             bg="#E0F7FA").grid(row=i, column=0, padx=20, pady=15, sticky="w")  
699         if text == "Ngày đặt":  
700             date_entry = DateEntry(frame_info, width=28, background='darkblue', foreground='white',  
701             borderwidth=2, date_pattern='dd/mm/yyyy', font=("Times New Roman", 13))  
702             date_entry.grid(row=i, column=1, padx=10, pady=15);  
703             self.entries["Ngày đặt"] = date_entry  
704         elif text == "Thời gian":  
705             times = [f"{h:02d}:{m}" for h in range(7, 17) for m in ["00", "30"]] + ["17:00"]  
706             cbo = ttk.Combobox(frame_info, values=times, font=("Times New Roman", 13), state="readonly", width=28)  
707             cbo.grid(row=i, column=1, padx=10, pady=15); cbo.current(0);  
708             self.entries["Thời gian"] = cbo  
709         else:  
710             entry = Entry(frame_info, font=("Times New Roman", 13), width=30);  
711             entry.grid(row=i, column=1, padx=10, pady=15);  
712             self.entries[text.replace(":", "")] = entry  
713  
714     # Điền các nhãn và ô nhập liệu- bên phải  
715     for i, text in enumerate(labels_right):  
716         Label(frame_info, text=text, font=("Times New Roman", 13),  
717             bg="#E0F7FA").grid(row=i, column=2, padx=40, pady=15, sticky="w")  
718         entry = Entry(frame_info, font=("Times New Roman", 13), width=30);  
719         entry.grid(row=i, column=3, padx=10, pady=15);  
720         self.entries[text.replace(":", "")] = entry  
721  
722     # Khung nút chức năng  
723     frame_buttons = Frame(self.root, bg="#E0F7FA")  
724     frame_buttons.place(x=250, y=300, width=500, height=60)  
725     Button(frame_buttons, text="Đặt lịch", command=self.dat_lich, font=("Times New Roman", 13, "bold"),  
726             bg="#FFC0CB", width=15, bd=2, relief="RAISED").place(x=20, y=10)  
727     Button(frame_buttons, text="Hủy", command=self.huy_lich, font=("Times New Roman", 13, "bold"),  
728             bg="#A9A9A9", width=15, bd=2, relief="RAISED").place(x=300, y=10)  
729  
730     # Khung bảng dữ liệu  
731     frame_table = Frame(self.root, bd=2, relief=RIDGE)  
732     frame_table.place(x=30, y=380, width=940, height=200)  
733     scroll_y = Scrollbar(frame_table, orient=VERTICAL);  
734     scroll_y.pack(side=RIGHT, fill=Y)  
735     columns = ("Mã đặt", "Ngày đặt", "Thời gian", "Mã bệnh nhân", "Mã bác sĩ", "Triệu chứng")  
736     self.tree = ttk.Treeview(frame_table, columns=columns, show="headings", yscrollcommand=scroll_y.set)  
737     scroll_y.config(command=self.tree.yview)  
738     for col in columns:  
739         self.tree.heading(col, text=col);  
740         self.tree.column(col, width=150 if col == "Triệu chứng" else 120, anchor="center")  
741         self.tree.pack(fill=BOTH, expand=True)  
742         self.tree.bind("<<TreeviewSelect>>", self.on_tree_select)

```

```

743     self.load_data()  
744  
745     # Hàm căn giữa cửa sổ  
746     def center_window(self, width, height):  
747         x = (self.root.winfo_screenwidth() // 2) - (width // 2);  
748         y = (self.root.winfo_screenheight() // 2) - (height // 2);  
749         self.root.geometry(f"{width}x{height}+{x}+{y}")  
750  
751     # Hàm tạo menu  
752     def create_menu(self):  
753         menubar = Menu(self.root);  
754         self.root.config(menu=menubar)  
755         sys_menu = Menu(menubar, tearoff=0)  
756         sys_menu.add_command(label="Đăng xuất", command=self.dang_xuat);  
757         sys_menu.add_command(label="Thoát", command=self.root.quit)  
758         menubar.add_cascade(label=f"Hệ thống ({self.role})", menu=sys_menu)  
759         man_menu = Menu(menubar, tearoff=0)  
760         man_menu.add_command(label="Quản lý Bệnh Nhân", command=self.open_quan_ly_benh_nhan)  
761  
762         if self.role == "admin":  
763             man_menu.add_command(label="Quản lý Bác Sĩ", command=self.open_quan_ly_bac_si)  
764         else:  
765             man_menu.add_command(label="Quản lý Bác Sĩ (Admin only)", state="disabled")  
766             man_menu.add_command(label="Đặt lịch khám", state="disabled")  
767  
768         if self.role in ["admin", "bacsi"]:

```

```

770         man_menu.add_command(label="Khám bệnh", command=self.open_kham_benh)
771     else:
772         man_menu.add_command(label="Khám bệnh (No Access)", state="disabled")
773
774     menubar.add_cascade(label="Quản lý", menu=man_menu)
775
776     # Hàm tải dữ liệu từ database vào bảng
777     def load_data(self):
778         for row in self.tree.get_children():
779             self.tree.delete(row)
780         conn = get_connection()
781         if conn:
782             cursor = conn.cursor()
783             cursor.execute("SELECT * FROM DATLICH")
784             for row in cursor.fetchall():
785                 self.tree.insert("", END, values=(row[0], date_db_to_ui(row[3]), str(row[4]), row[1], row[2], row[5]))
786             conn.close()
787
788     # Hàm đặt lịch
789     def dat_lich(self):
790         ngay_dat = date_ui_to_db(self.entries["Ngày đặt"].get())
791         thoi_gian = self.entries["Thời gian"].get()
792         ma_bn = self.entries["Mã bệnh nhân"].get()
793         ma_bs = self.entries["Mã bác sĩ"].get()
794         trieu_chung = self.entries["Triệu chứng"].get()
795
796         # Kiểm tra dữ liệu trống
797         if not ma_bn or not ma_bs:
798             messagebox.showerror("Lỗi", "Vui lòng nhập đầy đủ Mã BN và Mã BS");
799             return
800         conn = get_connection()
801         if conn:
802             try:
803                 cursor = conn.cursor()
804                 sql = "INSERT INTO DATLICH (MABN, MABS, NGAYDAT, THOIGIAN, TRIECHUNG) VALUES (%s, %s, %s, %s, %s)"
805                 val = (ma_bn, ma_bs, ngay_dat, thoi_gian, trieu_chung)
806                 cursor.execute(sql, val)
807                 conn.commit()
808                 messagebox.showinfo("Thành công", "Đã đặt lịch thành công!")
809                 self.load_data()
810                 self.clear_form()
811             except mysql.connector.Error as err:
812                 messagebox.showerror("Lỗi Database", f"Chi tiết lỗi: {err}")
813             finally:
814                 conn.close()
815
816         # Hàm hủy lịch
817         def huy_lich(self):
818             sel = self.tree.selection()
819             if sel and messagebox.askyesno("Xác nhận", "Hủy lịch này?"):
820                 conn = get_connection()
821
822                 if conn: conn.cursor().execute("DELETE FROM DATLICH WHERE MADAT=%s",
823                 | (self.tree.item(sel[0])['values'][0]),); conn.commit();
824                 conn.close();
825                 self.load_data()
826                 self.clear_form()
827
828         # Hàm làm mới form nhập liệu
829         def clear_form(self):
830             for k, v in self.entries.items():
831                 if isinstance(v, ttk.Combobox):
832                     v.current(0)
833                 elif isinstance(v, DateEntry):
834                     v.set_date(datetime.now())
835                 else:
836                     v.delete(0, END)
837
838         # Hàm xử lý khi chọn một dòng trong bảng
839         def on_tree_select(self, event):
840             try:
841                 sel = self.tree.selection()[0]; val = self.tree.item(sel)['values']
842                 self.entries["Mã đặt"].delete(0, END);
843                 self.entries["Mã đặt"].insert(0, val[0])
844                 try:
845                     self.entries["Ngày đặt"].set_date(datetime.strptime(val[1], '%d/%m/%Y'))
846                 except:
847                     pass
848                 self.entries["Thời gian"].set(val[2])

```

```

848     self.entries["Mã bệnh nhân"].delete(0, END);
849     self.entries["Mã bệnh nhân"].insert(0, str(val[3]))
850     self.entries["Mã bác sĩ"].delete(0, END);
851     self.entries["Mã bác sĩ"].insert(0, str(val[4]))
852     self.entries["Triệu chứng"].delete(0, END);
853     self.entries["Triệu chứng"].insert(0, val[5])
854 except:
855     pass
856
857 # Hàm điều hướng menu
858 def dang_xuat(self):
859     self.root.destroy();
860     root = Tk();
861     LoginSystem(root);
862     root.mainloop()
863 def thoat_ung_dung(self):
864     self.root.quit()

```

```

● 865 ▼ def open_quan_ly_benh_nhan(self):
866     self.root.destroy();
867     new_root = Tk();
868     QuanLyBenhNhan(new_root, self.role);
869     new_root.mainloop()
870 ▼ def open_quan_ly_bac_si(self):
871     self.root.destroy();
872     new_root = Tk();
873     QuanLyBacSi(new_root, self.role);
874     new_root.mainloop()
875 ▼ def open_kham_benh(self):
876     self.root.destroy();
877     new_root = Tk();
878     QuanLyKhamBenh(new_root, self.role);
879     new_root.mainloop()
880

```

h. Code quản lý khám bệnh

```

880
881
882
883 class QuanLyKhamBenh:
884
885     # Tạo giao diện chính
886     def __init__(self, root, role):
887         self.root = root;
888         self.role = role
889         self.root.title("Khám Bệnh - Quyền: " + self.role.upper())
890         self.root.geometry("1000x600")
891         self.root.center_window(1000, 600)
892         self.root.config(bg="#E0F7FA")
893         self.create_menu()
894         self.medicine_data = {}
895
896     # Giao diện quản lý khám bệnh
897     Label(self.root, text="KHÁM BỆNH", font=("Times New Roman", 24, "bold"), bg="#E0F7FA").pack(pady=10)
898
899     # Khung danh sách đặt lịch
900     frame_list = LabelFrame(self.root, text="Đặt lịch (Chờ khám)", font=("Times New Roman", 12, "bold"), bg="#E0F7FA")
901     frame_list.place(x=20, y=60, width=530, height=280)
902
903     # Bảng danh sách đặt lịch
904     cols = ("Mã đặt", "Mã BN", "Mã BS", "Triệu chứng")
905     self.tree_datlich = ttk.Treeview(frame_list, columns=cols, show="headings")
906     for col in cols:
907         self.tree_datlich.heading(col, text=col);

```



```

908         self.tree_datlich.column(col, width=150 if col == "Triệu chứng" else 80)
909         self.tree_datlich.pack(fill=BOTH, expand=True)
910         self.tree_datlich.bind("<<TreeviewSelect>>", self.on_schedule_select)
911
912     self.load_schedule_data()
913
914     # Khung khám bệnh
915     frame_kham = LabelFrame(self.root, text="Khám bệnh", font=("Times New Roman", 12, "bold"), bg="#B2EBF2")
916     frame_kham.place(x=20, y=350, width=530, height=250)
917     self.entries_kham = {}
918
919     # Các thành phần nhập liệu khám bệnh
920     labels_kham = ["Mã bác sĩ", "Mã bệnh nhân", "Triệu chứng", "Chuẩn đoán"]
921
922     # Các nhãn và ô nhập liệu
923     for i, text in enumerate(labels_kham):
924         Label(frame_kham, text=text, font=("Times New Roman", 12),
925               bg="#B2EBF2").grid(row=i, column=0, padx=20, pady=15, sticky="w")
926         entry = Entry(frame_kham, font=("Times New Roman", 12), width=35);
927         entry.grid(row=i, column=1, padx=10, pady=15); self.entries_kham[text] = entry
928
929     # Khung thuốc
930     frame_thuoc = LabelFrame(self.root, text="Thuốc", font=("Times New Roman", 12, "bold"), bg="#B2EBF2")
931     frame_thuoc.place(x=570, y=60, width=410, height=280)
932     self.entries_thuoc = {}

```

```

934
935
936     # Các nhân và ô nhập liệu thuốc
937     Label(frame_thuoc, text="Mã thuốc", font=("Times New Roman", 12), bg="#B2EBF2").place(x=20, y=30)
938     self.ent_ma_thuoc = Entry(frame_thuoc, font=("Times New Roman", 12), width=30);
939     self.ent_ma_thuoc.place(x=100, y=30)
940
941     Label(frame_thuoc, text="Tên thuốc", font=("Times New Roman", 12), bg="#B2EBF2").place(x=20, y=80)
942     self.cbo_ten_thuoc = ttk.Combobox(frame_thuoc, font=("Times New Roman", 12), width=28, state="readonly");
943     self.cbo_ten_thuoc.place(x=100, y=80)
944     self.cbo_ten_thuoc.bind("<>", self.on_select_medicine)
945
946     self.load_medicine_data()
947
948     Label(frame_thuoc, text="Số lượng", font=("Times New Roman", 12), bg="#B2EBF2").place(x=20, y=130)
949     self.ent_so_luong = Entry(frame_thuoc, font=("Times New Roman", 12), width=10);
950     self.ent_so_luong.place(x=100, y=130)
951
952     Label(frame_thuoc, text="Đơn giá", font=("Times New Roman", 12), bg="#B2EBF2").place(x=20, y=180)
953     self.ent_don_gia = Entry(frame_thuoc, font=("Times New Roman", 12), width=15);
954     self.ent_don_gia.place(x=100, y=180)
955
956     Label(frame_thuoc, text="VND", font=("Times New Roman", 12, "bold"), bg="#B2EBF2").place(x=300, y=180)
957
958     # Khung nút chức năng
959     frame_btn_action = Frame(self.root, bg="#E0F7FA")
960     frame_btn_action.place(x=570, y=350, width=410, height=60)

961
962     # Chia khung
963     frame_btn_action.grid_columnconfigure(0, weight=1)
964     frame_btn_action.grid_columnconfigure(1, weight=1)
965
966     # Nút Xuất hóa đơn
967     Button(frame_btn_action, text="Xuất phiếu khám", command=self.xuat_hoa_don,
968            font=("Times New Roman", 12, "bold"), bg="#FFC0CB",
969            width=15, height=2).grid(row=0, column=0, padx=5, pady=5)
970
971     # Nút Hủy
972     Button(frame_btn_action, text="Hủy", command=self.huy_kham,
973            font=("Times New Roman", 12, "bold"), bg="#D3D3D3",
974            width=15, height=2).grid(row=0, column=1, padx=5, pady=5)
975
976
977     # Hàm căn giữa cửa sổ
978     def center_window(self, width, height):
979         x = (self.root.winfo_screenwidth() // 2) - (width // 2);
980         y = (self.root.winfo_screenheight() // 2) - (height // 2)
981         self.root.geometry(f"{width}x{height}+{x}+{y}")
982
983
984     # Hàm tạo menu
985     def create_menu(self):
986         menubar = Menu(self.root)

987         self.root.config(menu=menubar)

988
989         sys_menu = Menu(menubar, tearoff=0)
990         sys_menu.add_command(label="Đang xuất", command=self.dang_xuat)
991         sys_menu.add_command(label="Thoát", command=self.root.quit)
992         menubar.add_cascade(label=f'Hệ thống ({self.role})', menu=sys_menu)

993
994         man_menu = Menu(menubar, tearoff=0)
995         man_menu.add_command(label="Quản lý Bệnh Nhân", command=self.open_quan_ly_benh_nhan)

996
997         if self.role == "admin":
998             man_menu.add_command(label="Quản lý Bác Sĩ", command=self.open_quan_ly_bac_si)
999         else:
1000             man_menu.add_command(label="Quản lý Bác Sĩ (Admin only)", state="disabled")

1001
1002         if self.role in ["admin", "nhanvien"]:
1003             man_menu.add_command(label="Đặt lịch khám", command=self.open_quan_ly_dat_lich)
1004         else:
1005             man_menu.add_command(label="Đặt lịch khám (No Access)", state="disabled")

1006
1007         if self.role in ["admin", "bacsi"]:
1008             man_menu.add_command(label="Khám bệnh (Đang mở)", state="disabled")
1009         else:
1010             man_menu.add_command(label="Khám bệnh (No Access)", state="disabled")
1011
1012         menubar.add_cascade(label="Quản lý", menu=man_menu)

```

```

1013     # Hàm tải dữ liệu lịch khám
1014     def load_schedule_data(self):
1015         for row in self.tree_datlich.get_children():
1016             self.tree_datlich.delete(row)
1017         conn = get_connection()
1018         if conn:
1019             cursor = conn.cursor()
1020             cursor.execute("SELECT MADAT, MABN, MABS, TRIECHUNG FROM DATLICH")
1021             for row in cursor.fetchall():
1022                 self.tree_datlich.insert("", END, values=row)
1023             conn.close()
1024
1025     # Hàm tải dữ liệu thuốc
1026     def load_medicine_data(self):
1027         conn = get_connection()
1028         if conn:
1029             cursor = conn.cursor()
1030             cursor.execute("SELECT MATHUOC, TENTHUOC, DONGIA FROM THUOC")
1031             for row in cursor.fetchall():
1032                 self.medicine_data[row[1]] = {"id": row[0], "price": row[2]}
1033                 self.cbo_ten_thuoc['values'] = list(self.medicine_data.keys())
1034             conn.close()
1035
1036     # Hàm xử lý khi chọn một lịch khám
1037     def on_schedule_select(self, event):

```

```

1019         if conn:
1020             cursor = conn.cursor()
1021             cursor.execute("SELECT MADAT, MABN, MABS, TRIECHUNG FROM DATLICH")
1022             for row in cursor.fetchall():
1023                 self.tree_datlich.insert("", END, values=row)
1024             conn.close()
1025
1026     # Hàm tải dữ liệu thuốc
1027     def load_medicine_data(self):
1028         conn = get_connection()
1029         if conn:
1030             cursor = conn.cursor()
1031             cursor.execute("SELECT MATHUOC, TENTHUOC, DONGIA FROM THUOC")
1032             for row in cursor.fetchall():
1033                 self.medicine_data[row[1]] = {"id": row[0], "price": row[2]}
1034                 self.cbo_ten_thuoc['values'] = list(self.medicine_data.keys())
1035             conn.close()
1036
1037     # Hàm xử lý khi chọn một lịch khám
1038     def on_schedule_select(self, event):
1039         try:
1040             sel = self.tree_datlich.selection()[0];
1041             val = self.tree_datlich.item(sel)['values']
1042             self.entries_kham["Mã bệnh nhân"].delete(0, END);
1043             self.entries_kham["Mã bệnh nhân"].insert(0, str(val[1]))
1044             self.entries_kham["Mã bác sĩ"].delete(0, END);

```

```

1045             self.entries_kham["Mã bác sĩ"].insert(0, str(val[2]))
1046             self.entries_kham["Triệu Chứng"].delete(0, END);
1047             self.entries_kham["Triệu Chứng"].insert(0, str(val[3]))
1048         except:
1049             pass
1050
1051     # Hàm xử lý khi chọn một thuốc
1052     def on_select_medicine(self, event):
1053         name = self.cbo_ten_thuoc.get()
1054         if name in self.medicine_data:
1055             data = self.medicine_data[name]
1056             self.ent_ma_thuoc.delete(0, END);
1057             self.ent_ma_thuoc.insert(0, data["id"])
1058             self.ent_don_gia.delete(0, END);
1059             self.ent_don_gia.insert(0, str(int(data["price"])))
1060
1061
1062     # Hàm xuất hóa đơn
1063     def xuat_hoa_don(self):
1064         try:
1065             sl_str = self.ent_so_luong.get()
1066             if not sl_str.strip():
1067                 messagebox.showerror("Lỗi", "Vui lòng nhập số lượng thuốc!")
1068             return
1069

```

```

1070     sl = int(sl_str)
1071     if sl <= 0: raise ValueError
1072
1073     gia = float(self.ent_don_gia.get())
1074     tong_tien = sl * gia
1075
1076     # Lấy thông tin chẩn đoán
1077     trieu_chung = self.entries_kham["Triệu Chứng"].get()
1078     chuan_doen = self.entries_kham["Chẩn đoán"].get()
1079     ten_thuoc = self.cbo_ten_thuoc.get()
1080     ma_bn = self.entries_kham["Mã bệnh nhân"].get()
1081     ma_bs = self.entries_kham["Mã bác sĩ"].get()
1082     ngay_gio = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
1083
1084     # 2. Lưu vào Database
1085     conn = get_connection()
1086     if conn:
1087         try:
1088             cursor = conn.cursor()
1089             sql = "INSERT INTO HOADON_KHAMBENH (MABN, MABS, NGAYKHAM, MATHUOC, TRIECHUNG, CHUANDOAN) VALUES (%s, %s, NOW(), %s, %s, %s)"
1090             val = (ma_bn, ma_bs, ma_thuoc, trieu_chung, chuan_doen)
1091
1092             cursor.execute(sql, val)
1093             conn.commit()
1094

```

```

1095     except Exception as err:
1096         messagebox.showerror("Lỗi Database", f"Không lưu được vào DB (kiểm tra lại cột CHUANDOAN): {err}")
1097         return
1098     finally:
1099         conn.close()
1100
1101     # TẠO GIAO DIỆN HÓA ĐƠN
1102     bill_window = Toplevel(self.root)
1103     bill_window.title("Phiếu Khám")
1104     bill_window.geometry("500x600")
1105     bill_window.config(bg="white")
1106
1107     # Tiêu đề
1108     Label(bill_window, text="PHIẾU KHÁM & ĐƠN THUỐC", font=("Times New Roman", 16, "bold"), bg="white").pack(pady=5)
1109     Label(bill_window, text=f"Ngày: {ngay_gio}", font=("Times New Roman", 10), bg="white").pack()
1110
1111     # Đường kẻ
1112     Frame(bill_window, height=2, bd=1, relief=SUNKEN).pack(fill=X, padx=20, pady=10)
1113
1114     # Nội dung hóa đơn
1115     content_frame = Frame(bill_window, bg="white")
1116     content_frame.pack(fill=BOTH, expand=True, padx=30)
1117
1118     info_text = f"""
1119     THÔNG TIN BỆNH NHÂN
1120

```

```

1121     Mã bệnh nhân : {ma_bn}
1122     Mã bác sĩ   : {ma_bs}
1123
1124     CHẨN ĐOÁN
1125
1126     Triệu chứng  : {trieu_chung}
1127     Chẩn đoán   : {chuan_doen}
1128
1129     ĐƠN THUỐC & THANH TOÁN
1130
1131     Tên thuốc   : {ten_thuoc}
1132     Số lượng    : {sl}
1133     Đơn giá     : {gia:.0f} VNĐ
1134
1135     """
1136
1137     lbl_info = Label(content_frame, text=info_text, font=("Courier New", 12), bg="white", justify=LEFT, anchor="w")
1138     lbl_info.pack(fill=X)
1139
1140     # Tổng tiền
1141     total_frame = Frame(bill_window, bg="#F0F8FF", bd=1, relief=SOLID)
1142     total_frame.pack(fill=X, padx=20, pady=20)
1143     Label(total_frame, text=f"TỔNG CỘNG: {tong_tien:.0f} VNĐ", font=("Times New Roman", 18, "bold"),
1144           bg="#F0F8FF", fg="red").pack(pady=10)

```

```

1145         # Nút đóng
1146         Button(bill_window, text="In Phiếu / Đóng", command=bill_window.destroy,
1147             | | | bg="#FFC0CB", font=("Times New Roman", 12)).pack(pady=10)
1148
1149     except ValueError:
1150         messagebox.showerror("Lỗi", "Số lượng phải là số nguyên dương!")
1151
1152     # Hủy khám
1153     def huy_kham(self):
1154         if messagebox.askyesno("Hủy", "Bạn có chắc muốn hủy đơn thuốc này?"):
1155             self.cbo_ten_thuoc.set('');
1156             self.ent_ma_thuoc.delete(0, END);
1157             self.ent_so_luong.delete(0, END);
1158             self.ent_don_gia.delete(0, END)
1159
1160     # Hàm điều hướng menu
1161     def dang_xuat(self):
1162         self.root.destroy();
1163         root = Tk();
1164         LoginSystem(root);
1165         root.mainloop()
1166
1167     def thoat_ung_dung(self):
1168         self.root.quit()
1169
1170     def open_quan_ly_bac_si(self):
1171         self.root.destroy();
1172         new_root = Tk();
1173         QuanLyBacSi(new_root, self.role);
1174         new_root.mainloop()
1175
1176     def open_quan_ly_dat_lich(self):
1177         self.root.destroy();
1178         new_root = Tk();
1179         QuanLyDatLich(new_root, self.role);
1180         new_root.mainloop()
1181
1182     def open_quan_ly_benh_nhan(self):
1183         self.root.destroy();
1184         new_root = Tk();
1185         QuanLyBenhNhan(new_root, self.role);
1186         new_root.mainloop()
1187
1188     # Main Program
1189     if __name__ == "__main__":
1190         root = Tk()
1191         app = LoginSystem(root)
1192         root.mainloop()
1193

```

IV. TÀI LIỆU THAM KHẢO

[1] TS. Huỳnh Phước Hải, TS. Nguyễn Văn Hòa, *Giáo trình lập trình Python*,
Trường Đại học An Giang – ĐHQG TP. Hồ Chí Minh.