

**ĐẠI HỌC ĐÀ NẴNG
ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH
QUẢN LÝ HỒ SƠ BỆNH ÁN**

SINH VIÊN THỰC HIỆN:

Nguyễn Song Hào LỚP: 23T_KHDL1 NHÓM: 23NH15

Đoàn Ngọc Minh LỚP: 23T_KHDL2 NHÓM: 23NH15

GIẢNG VIÊN HƯỚNG DẪN: TS Trương Ngọc Châu

Đà Nẵng – 11/11/2024

LỜI MỞ ĐẦU

Trong thế kỷ 21 – thời đại của cách mạng công nghệ, việc ứng dụng công nghệ thông tin vào mọi lĩnh vực đời sống, đặc biệt là y tế, đã không còn là lựa chọn mà trở thành điều tất yếu. Ngành y tế, với sứ mệnh cao cả chăm sóc sức khỏe cộng đồng, đang đứng trước yêu cầu cấp bách phải hiện đại hóa để đáp ứng sự gia tăng không ngừng của khối lượng dữ liệu và những thách thức ngày một phức tạp. Thế nhưng, tại nhiều cơ sở y tế, hệ thống quản lý bệnh án thủ công vẫn được duy trì, tạo nên một nghịch lý đáng suy ngẫm.

Phương pháp quản lý truyền thống – dựa trên giấy tờ ghi chép – dường như đang bị thời đại bỏ lại phía sau. Những chồng hồ sơ dày cộp không chỉ chiếm diện tích, đòi hỏi chi phí lưu trữ mà còn đối mặt với nguy cơ mất mát thông tin, sai sót dữ liệu và những ảnh hưởng khó lường từ môi trường như ẩm mốc, cháy nổ. Thử tưởng tượng, trong một trường hợp khẩn cấp, việc tìm kiếm thông tin bệnh án giữa hàng trăm, hàng nghìn hồ sơ giấy tờ có thể làm chậm trễ quá trình cứu chữa – đôi khi chỉ vài phút ngắn ngủi cũng quyết định sự sống còn của một con người.

Chính những bất cập ấy đã thôi thúc chúng em – những người trẻ mang khát khao đóng góp – phát triển “Hệ thống quản lý bệnh nhân”. Đây không chỉ là một dự án công nghệ, mà còn là lời giải cho bài toán quản lý thông tin y tế trong thời đại số hóa. Hệ thống này được xây dựng với triết lý cốt lõi: khoa học, chính xác và an toàn. Thông qua ứng dụng công nghệ lập trình hướng đối tượng, chúng em kỳ vọng mang đến một giải pháp ưu việt giúp lưu trữ, truy xuất thông tin nhanh chóng và chính xác, đồng thời giảm thiểu tối đa sai sót và rủi ro bảo mật.

Chúng em hy vọng rằng hệ thống sẽ không chỉ giảm tải công việc hành chính cho đội ngũ y bác sĩ mà còn mở ra một chương mới trong quản lý y tế, tạo nền tảng cho sự phát triển bền vững. Việc quản lý thông tin bệnh nhân chặt chẽ và hiệu quả sẽ giúp người bệnh yên tâm hơn, đồng thời cho phép bác sĩ tập trung tối đa vào chuyên môn. Đó chính là mục tiêu mà mọi nỗ lực của chúng em đều hướng đến.

Dự án không chỉ là một phần mềm, mà còn là cơ hội để chúng em học hỏi, ứng dụng công nghệ vào thực tiễn và đối mặt với các thách thức xã hội. Với nhiệt huyết và sáng tạo, chúng em tin rằng “Hệ thống quản lý bệnh nhân” sẽ góp phần hiện đại hóa ngành y tế, hướng tới một tương lai thông minh và nhân văn hơn.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

MỤC LỤC

LỜI MỞ ĐẦU	1
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	2
MỤC LỤC	3
DANH MỤC HÌNH VẼ.....	5
1.GIỚI THIỆU ĐỀ TÀI.....	7
2.PHÂN TÍCH CHỨC NĂNG HỆ THỐNG	8
2.1. Chức năng quản lý bệnh nhân và bác sĩ	8
2.2. Các phương thức chính	8
2.2.1 Thêm bệnh nhân	9
2.2.2 Tìm kiếm bệnh nhân	10
2.2.3 Xóa bệnh nhân	11
2.2.4 Hiện thị danh sách bệnh nhân	11
2.2.5 Chỉnh sửa bệnh nhân.....	12
2.2.6 Thống kê loại bệnh	12
2.2.7 In bệnh nhân theo ngày khám bệnh (ngày nhập viện)	13
2.2.8 Thêm một bác sĩ.....	14
2.2.9 Xóa một bác sĩ	14
2.2.10 Hiện thị danh sách các bác sĩ.....	15
2.2.11 Chỉnh sửa một bác sĩ.....	15
2.2.12 Hiện thị danh sách các bác sĩ đang trong ca làm việc	16
2.2.13 Hiện thị danh sách các loại thuốc	16
2.2.14 Hiện thị danh sách các loại thuốc còn có thể sử dụng.....	17
2.2.15 Phương thức lưu trữ.....	17
3. THIẾT KẾ CẤU TRÚC DỮ LIỆU	19
3.1. Phát biểu bài toán	19

3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống	19
4. ỨNG DỤNG HƯỚNG ĐỐI TƯỢNG VÀO BÀI TOÁN.....	23
4.1. Xác định các thực thể chính trong bài toán	23
4.2. Phân tích các lớp của hướng đối tượng.....	24
4.3. Lợi ích của việc ứng dụng hướng đối tượng	33
4.4. Kết quả	34
4.4.1. Giao diện chính của chương trình	35
4.4.2. Kết quả thực thi của chương trình.....	36
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	50
5.1. Kết luận	50
5.1.1. Đạt được	50
5.1.2. Chưa đạt được	51
5.2. Hướng phát triển	51
TÀI LIỆU THAM KHẢO.....	53

DANH MỤC HÌNH VẼ

Hình 2.2.1 Mô tả chức năng thêm bệnh nhân.....	10
Hình 2.2.2 Mô tả chức năng tìm kiếm bệnh nhân	10
Hình 2.2.3 Mô tả chức năng xóa bệnh nhân	11
Hình 2.2.4 Mô tả chức năng hiển thị bệnh nhân.....	11
Hình 2.2.5 Mô tả chức năng chỉnh sửa bệnh nhân	12
Hình 2.2.6 Mô tả chức năng thống kê loại bệnh.....	13
Hình 2.2.7 Mô tả chức năng in bệnh nhân theo ngày khám	13
Hình 2.2.8 Mô tả chức năng thêm một bác sĩ.....	14
Hình 2.2.9 Mô tả chức năng xóa một bác sĩ	15
Hình 2.2.10 Mô tả chức năng hiển thị danh sách bác sĩ	15
Hình 2.2.11 Mô tả chức năng chỉnh sửa một bác sĩ.....	16
Hình 2.2.12 Mô tả chức năng hiển thị các bác sĩ đang trong ca làm.....	16
Hình 2.2.13 Mô tả chức năng hiển thị danh sách các loại thuốc	17
Hình 2.2.14 Mô tả chức năng hiển thị các loại thuốc có thể sử dụng.....	17
Hình 3.2.1 Mô tả thuật toán tìm kiếm tuần tự.....	25
Hình 4.2.1 Mô tả lớp Person.....	24
Hình 4.2.2 Mô tả lớp disease	25
Hình 4.2.3 Mô tả lớp patient.....	26
Hình 4.2.4 Mô tả lớp Inpatient.....	27
Hình 4.2.5 Mô tả lớp Outpatient.....	28
Hình 4.2.6 Mô tả lớp medicine	29
Hình 4.2.7 Mô tả lớp available	30
Hình 4.2.8 Mô tả lớp doctor.....	31
Hình 4.3 Mô tả mối quan hệ giữa các lớp.....	33

Hình 4.5.1.1 Giao diện đăng nhập chương trình	35
Hình 4.5.1.2 Giao diện menu lựa chọn	35
Hình 4.5.2.1 File quản lý các tài khoản của Admin	36
Hình 4.5.2.2 File quản lý thông tin tài khoản của Bác sĩ.....	36
Hình 4.5.2.3 File quản lý thông tin bệnh nhân nội trú.....	37
Hình 4.5.2.4 File quản lý thông tin bệnh nhân ngoại trú	37
Hình 4.5.2.5 File quản lý thông tin các loại thuốc.....	37
Hình 4.5.2.6 Giao diện đăng kí tài khoản admin	38
Hình 4.5.2.7 Giao diện lấy lại mật khẩu đăng nhập	38
Hình 4.5.2.8 Giao diện hiển thị danh sách bác sĩ, với các chức năng	39
Hình 4.5.2.9 Giao diện hiển thị các bác sĩ hiện tại đang trong ca làm việc.....	39
Hình 4.5.2.10 Giao diện xem thông tin của Bác sĩ	40
Hình 4.5.2.11 Giao diện thực hiện chức năng chỉnh sửa thông tin của bác sĩ.....	40
Hình 4.5.2.12 Thực hiện chức năng thêm 1 bác sĩ	41
Hình 4.5.2.13 Giao diện thực hiện chức năng tìm kiếm thông tin của Bác sĩ.....	41
Hình 4.5.2.14 Giao diện hiển thị thông tin bệnh nhân nội trú và các chức năng	42
Hình 4.5.2.15 Giao diện hiển thị thông tin của bệnh nhân ngoại trú.....	42
Hình 4.5.2.16 Giao diện xem thông tin bệnh nhân nội trú	43
Hình 4.5.2.17 Thực hiện chức năng chỉnh sửa bệnh nhân nội trú	43
Hình 4.5.2.18 Thực hiện chức năng xóa.....	44
Hình 4.5.2.19 Thực hiện chức năng tìm kiếm bệnh nhân theo tên.....	44
Hình 4.5.2.20 Thực hiện chức năng tìm kiếm bệnh nhân theo ngày	45
Hình 4.5.2.21 Giao diện medicine hiển thị thông tin của các loại thuốc.....	45
Hình 4.5.2.22 Giao diện hiển thị các loại thuốc đã hết hạn sử dụng	46
Hình 4.5.2.23 Thực hiện chức năng chỉnh sửa thuốc	46
Hình 4.5.2.24 Thực hiện chức năng xóa loại thuốc	47

Hình 4.5.2.25 Thực hiện chức năng thêm 1 loại thuốc mới	47
Hình 4.5.2.26 Thực hiện chức năng tìm kiếm loại thuốc	48
Hình 4.5.2.27 Thực hiện chức năng thay đổi mật khẩu.....	48
Hình 4.5.2.28 Thực hiện chức năng đăng xuất.....	49
Hình 5 Mã qr code PBL.....	54

1. GIỚI THIỆU ĐỀ TÀI

1.1. Đặt vấn đề

Bệnh viện đóng một vai trò thiết yếu trong việc chăm sóc sức khỏe cộng đồng bằng cách cung cấp các dịch vụ y tế chuyên sâu cho bệnh nhân mắc phải nhiều loại bệnh khác nhau, từ những vấn đề liên quan đến lối sống, căng thẳng tâm lý đến các bệnh lý do yếu tố môi trường hoặc di truyền. Đội ngũ nhân viên trong bệnh viện, bao gồm bác sĩ, y tá, và các nhân viên hỗ trợ, là nhân tố chính giúp vận hành hệ thống một cách hiệu quả, đồng thời chịu trách nhiệm lưu giữ và quản lý hồ sơ liên quan đến mọi hoạt động và tương tác trong cơ sở y tế.

Tuy nhiên, việc ghi chép thủ công các hoạt động hàng ngày, từ quản lý hồ sơ bệnh nhân đến ghi nhận tương tác giữa các nhân viên và bệnh nhân, không chỉ mất nhiều thời gian mà còn dễ xảy ra sai sót. Với áp lực từ sự gia tăng dân số và số lượt khám chữa bệnh, phương pháp quản lý thủ công trở nên ngày càng lỗi thời, thiếu hiệu quả, và khó đáp ứng nhu cầu thực tế. Điều này không chỉ làm giảm năng suất làm việc mà còn có thể ảnh hưởng đến chất lượng dịch vụ y tế.

1.2. Giới thiệu đề tài

Các quy trình thủ công và không đồng bộ mà nhiều bệnh viện hiện nay áp dụng đang gây ra sự kém hiệu quả và cản trở khả năng cung cấp dịch vụ chăm sóc bệnh nhân chất lượng cao. Hệ thống thông tin không tổ chức, công nghệ lạc hậu và việc lưu trữ hồ sơ trên giấy là nguyên nhân dẫn đến những sai sót, thiếu sót và sự chậm trễ trong việc quản lý dữ liệu quan trọng của bệnh nhân. Hơn nữa, sự thiếu hụt các giải pháp tích hợp khiến việc phân tích dữ liệu để tối ưu hóa nguồn lực và đưa ra quyết định chiến lược trở nên phức tạp và không hiệu quả.

Một Hệ thống Quản lý Bệnh viện hiện đại và toàn diện là giải pháp thiết yếu để khắc phục các hạn chế của phương pháp quản lý truyền thống, đồng thời nâng cao hiệu quả hoạt động y tế. Hệ thống này không chỉ tự động hóa và tối ưu hóa các chức năng cốt lõi mà còn đảm bảo quản lý bệnh nhân hiệu quả, cải thiện sự phối hợp giữa các bên liên quan. Với các công cụ phân tích mạnh mẽ, hệ thống hỗ trợ đánh giá hiệu suất, thúc đẩy cải tiến liên tục, và ra quyết định dựa trên dữ liệu, từ đó nâng cao chất lượng dịch vụ y tế một cách toàn diện.

So với phương pháp ghi chép thủ công, hệ thống quản lý bệnh viện bằng công nghệ số mang lại nhiều lợi ích vượt trội. Nó cho phép bác sĩ và nhân viên y tế quản lý thông tin bệnh nhân dễ dàng và chính xác thông qua các chức năng như kiểm tra thông tin bệnh nhân, quản lý bệnh án, và cập nhật tình trạng sức khỏe. Hệ thống còn hỗ trợ tra cứu thông tin nhanh chóng, chỉnh sửa hoặc xóa dữ liệu khi cần, đồng thời thống kê các bệnh lý đang được điều trị. Nhờ vậy, quy trình quản lý bệnh nhân trở nên thuận tiện, chính xác, giảm thiểu sai sót, và góp phần nâng cao chất lượng dịch vụ y tế một cách vượt bậc.

2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

Phần này tập trung phân tích chi tiết các tính năng cốt lõi của hệ thống quản lý bệnh nhân, hướng tới việc hỗ trợ người dùng thực hiện các thao tác một cách nhanh chóng, chính xác và thuận tiện nhất. Những tính năng này không chỉ đáp ứng nhu cầu quản lý cơ bản mà còn nâng cao trải nghiệm sử dụng nhờ thiết kế thông minh và tối ưu. Cụ thể, hệ thống cung cấp các chức năng chính sau:

2.1. Chức năng quản lý bệnh nhân và bác sĩ

- Chức năng quản lý bệnh nhân cung cấp đầy đủ các công cụ để quản lý thông tin, bao gồm thêm, tìm kiếm, xóa, chỉnh sửa, hiển thị và quản lý dữ liệu bệnh nhân. Nó cũng bao gồm chức năng thống kê bệnh và liệt kê các bệnh nhân nhập viện tại một thời điểm.
- Chức năng quản lý bác sĩ được thiết kế nhằm hỗ trợ người dùng trong việc thêm, xóa, chỉnh sửa và hiển thị thông tin bác sĩ một cách dễ dàng. Ngoài ra, chức năng này còn giúp xác định các bác sĩ đang làm việc, kiểm tra danh mục thuốc hiện có trong kho, và phân loại các loại thuốc phù hợp để sử dụng, đảm bảo việc điều trị được tối ưu hóa và hiệu quả.

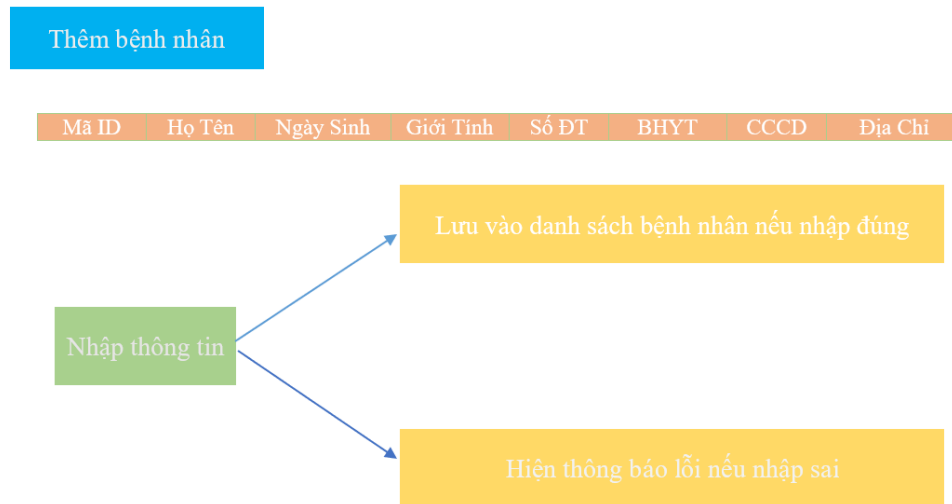
2.2. Các phương thức chính

2.2.1. Thêm bệnh nhân

Chức năng nhập thông tin bệnh nhân mới vào hệ thống là một phần quan trọng trong việc quản lý hồ sơ y tế, đảm bảo việc theo dõi và điều trị được thực hiện chính xác và hiệu quả. Thông qua chức năng này, admin có thể dễ dàng thêm thông tin chi tiết của bệnh nhân vào cơ sở dữ liệu, phục vụ cho việc truy xuất và xử lý dữ liệu khi cần thiết. Người dùng sẽ cần cung cấp các thông tin như họ tên, ngày sinh, giới tính, cân nặng, số điện thoại, mã BHYT, địa chỉ, CCCD, tên bệnh, triệu chứng, phương thức điều trị, ngày nhập viện.

Chức năng nhập thông tin bệnh nhân mới được thiết kế để hỗ trợ tối ưu hóa quản lý hồ sơ y tế, bao gồm việc phân loại và ghi nhận đầy đủ các chi tiết liên quan đến từng loại bệnh nhân. Với sự khác biệt trong nhu cầu và quy trình điều trị, hệ thống cho phép nhập thông tin riêng biệt và chi tiết cho hai nhóm chính: **bệnh nhân nội trú** và **bệnh nhân ngoại trú**.

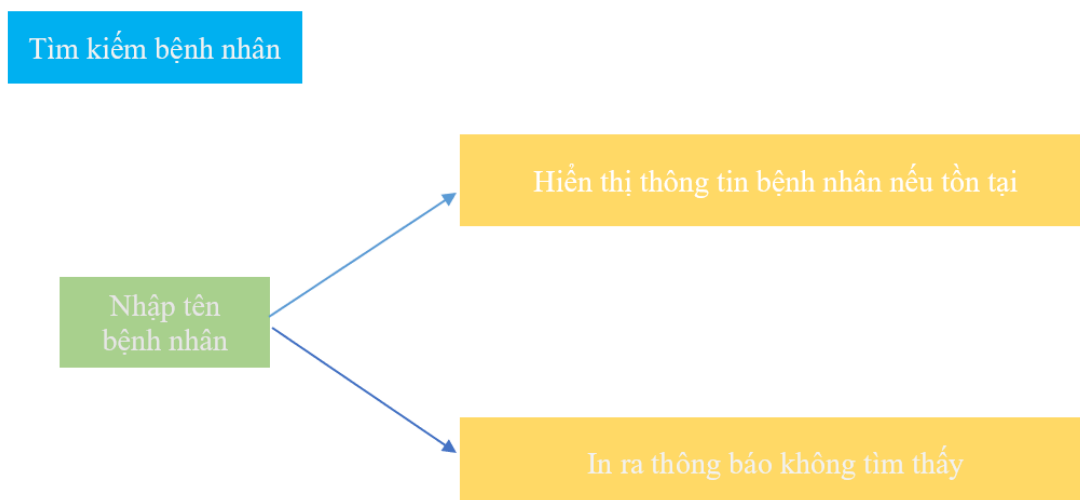
- Đối với bệnh nhân nội trú thì nhập các thông tin như số phòng điều trị và ngày xuất viện.
- Đối với bệnh nhân ngoại trú thì nhập các thông tin như số lần hẹn tái khám và số ngày giữa mỗi lần khám.



Hình 2.2.1 Mô tả chức năng thêm bệnh nhân

2.2.2. Tìm kiếm bệnh nhân

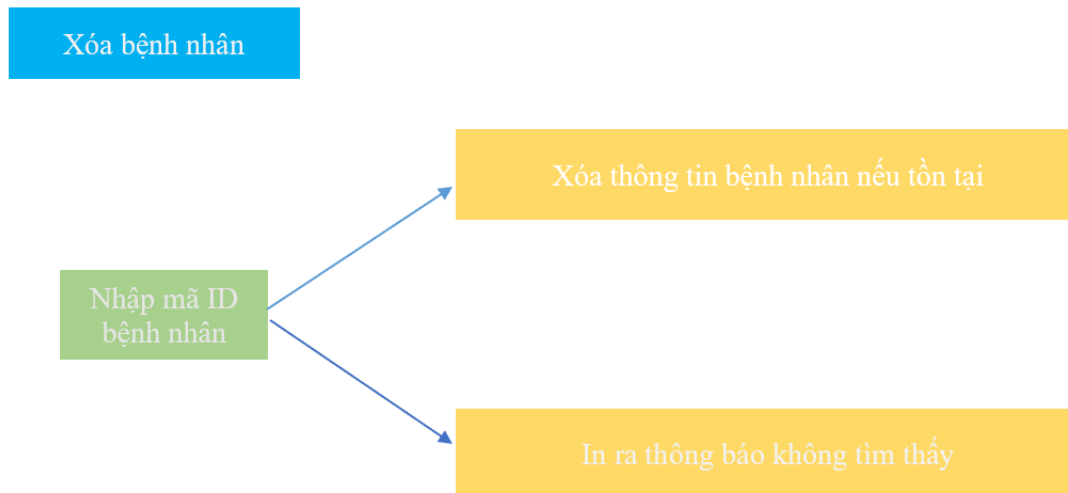
Chức năng tìm kiếm thông tin bệnh nhân theo tên cho phép admin nhanh chóng tra cứu chỉ với một phần tên, nhờ ứng dụng thuật toán tìm kiếm chuỗi. Tính năng này hỗ trợ admin dễ dàng tìm kiếm, tiết kiệm thời gian và tăng hiệu quả quản lý. Các kết quả trả về sẽ hiển thị đầy đủ thông tin chi tiết của bệnh nhân, giúp nhân viên y tế kiểm tra và cập nhật khi cần thiết. Nhờ đó, thời gian tìm kiếm được tối ưu hóa, giảm nguy cơ sai sót.



Hình 2.2.2 Mô tả chức năng tìm kiếm bệnh nhân

2.2.3. Xóa bệnh nhân

Chức năng này cho phép xóa thông tin của bệnh nhân khỏi hệ thống dựa trên mã bệnh nhân. Nếu mã bệnh nhân có tồn tại thì hệ thống sẽ tiến hành xóa bệnh nhân có mã vừa nhập và thông báo đã xóa thành công, còn nếu không tồn tại mã đó thì hệ thống in ra không tìm thấy hồ sơ bệnh nhân có mã vừa nhập.



Hình 2.2.3 Mô tả chức năng xóa bệnh nhân

2.2.4. Hiển thị danh sách bệnh nhân

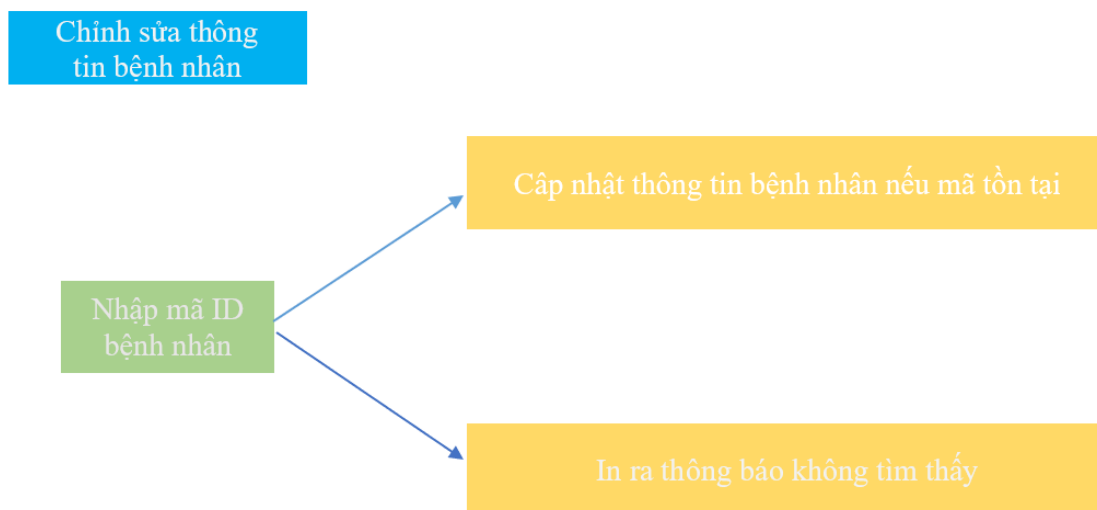
Chức năng này cho phép admin xem danh sách tất cả các bệnh nhân hiện đang được lưu trong hệ thống. Danh sách sẽ hiển thị các thông tin cơ bản của bệnh nhân, bao gồm mã số bệnh nhân, họ tên, khoa, chẩn đoán, và loại bệnh nhân. Nếu bệnh nhân là nội trú, danh sách cũng sẽ hiển thị phòng và giường bệnh. Còn nếu là bệnh nhân nội trú, danh sách sẽ hiển thị số lần hẹn tái khám và số ngày giữa mỗi lần khám.



Hình 2.2.4 Mô tả chức năng hiển thị bệnh nhân

2.2.5. Chỉnh sửa bệnh nhân

Chức năng này cho admin cập nhật lại thông tin của bệnh nhân. Hệ thống yêu cầu nhập mã của bệnh nhân cần cập nhật, nếu mã bệnh nhân có tồn tại thì hệ thống sẽ cho chúng ta nhập lại các thông tin của bệnh nhân như tên, số điện thoại, CCCD.... Và khi cập nhật xong hệ thống sẽ thông báo đã cập nhật thành công, còn nếu không tồn tại mã đó thì hệ thống in ra không tìm thấy hồ sơ.



Hình 2.2.5 Mô tả chức năng chỉnh sửa bệnh nhân

2.2.6. Thống kê loại bệnh

Chức năng này dùng để thống kê số lượng bệnh nhân theo từng loại bệnh, giúp bệnh viện biết rõ các bệnh nào đang được điều trị. Từ đó, bệnh viện có thể quản lý tốt hơn các nguồn lực như nhân lực, thiết bị, và thuốc men, đồng thời lập kế hoạch điều trị và phòng ngừa phù hợp. Thống kê này cũng hỗ trợ bác sĩ và nhân viên y tế theo dõi tình hình sức khỏe của bệnh nhân, đặc biệt hữu ích trong việc kiểm soát các bệnh dễ lây lan hoặc bùng phát dịch bệnh.

Thống kê loại bệnh

Hiển thị các loại bệnh
có trong danh sách

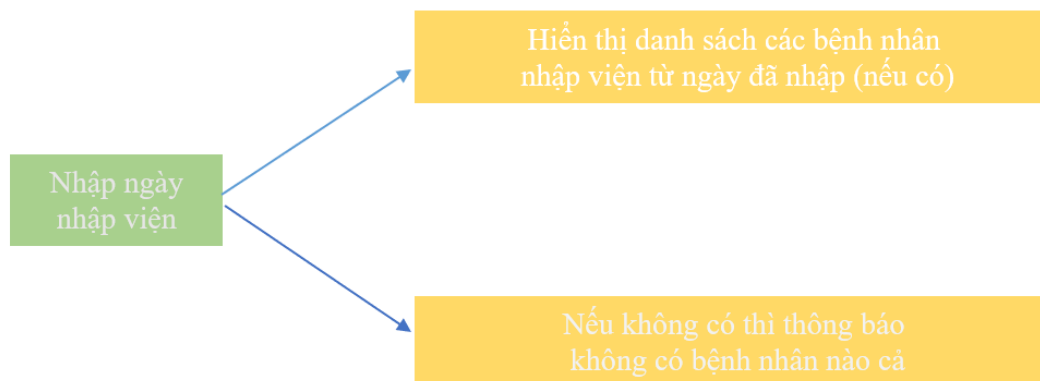
Thống kê số lượng bệnh theo
từng loại bệnh

Hình 2.2.6 Mô tả chức năng thống kê loại bệnh

2.2.7. In bệnh nhân theo ngày khám (ngày nhập viện)

Chức năng này cho phép xuất dữ liệu bệnh nhân dựa trên ngày nhập viện, giúp nhân viên y tế dễ dàng tra cứu thông tin bệnh nhân nhập viện vào một ngày cụ thể. Điều này hỗ trợ việc theo dõi và quản lý số lượng bệnh nhân theo từng thời điểm, phục vụ cho việc thống kê, lập báo cáo và đánh giá tình hình tiếp nhận bệnh nhân trong các khoảng thời gian nhất định. Chức năng này cũng giúp bệnh viện lên kế hoạch phân bổ giường bệnh, trang thiết bị và nhân sự phù hợp để đảm bảo đáp ứng nhu cầu điều trị.

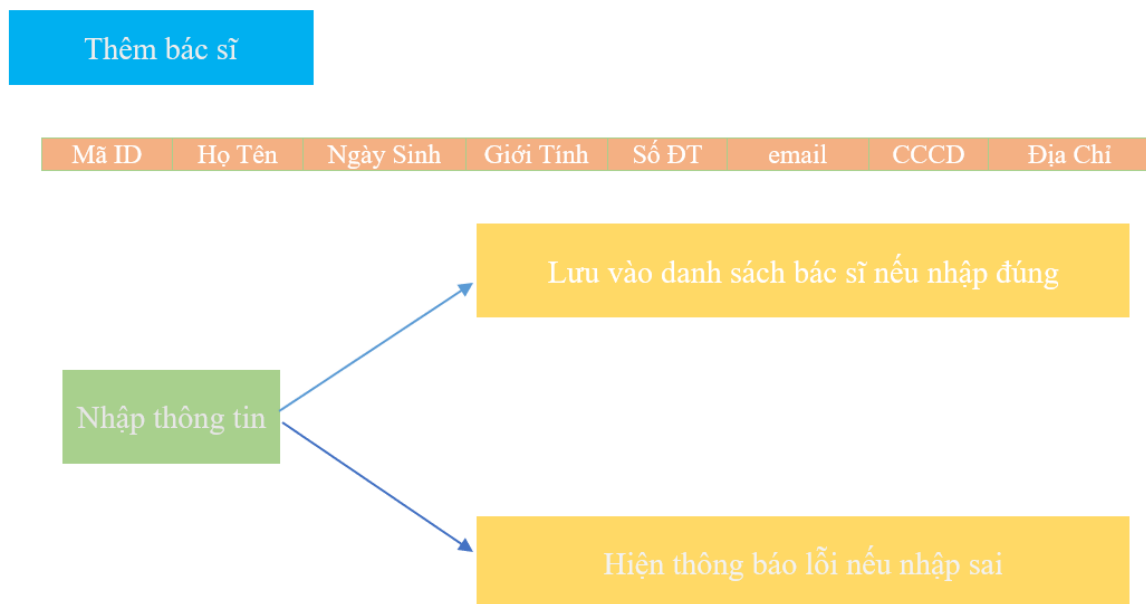
Hiển thị các bệnh
nhân



Hình 2.2.7 Mô tả chức năng in bệnh nhân theo ngày khám

2.2.8. Thêm một bác sĩ

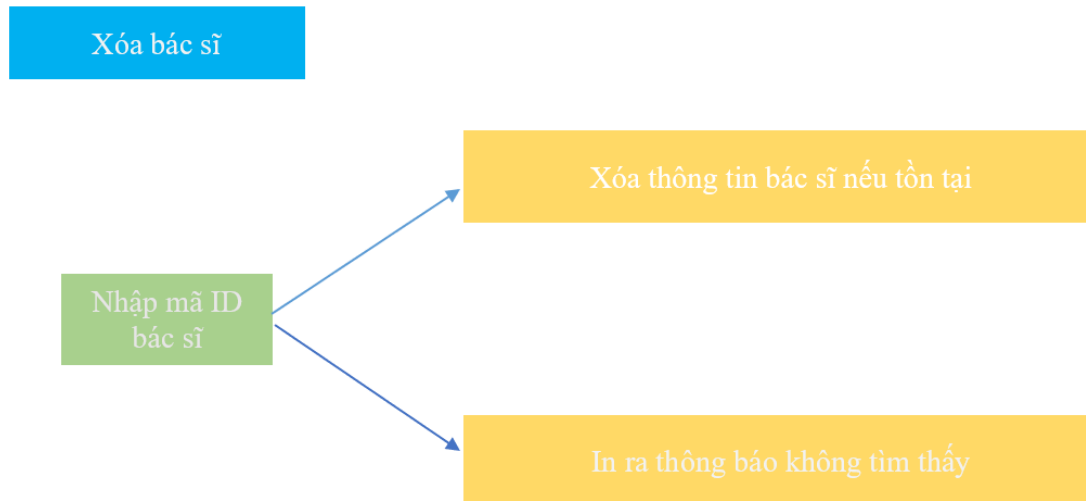
Chức năng thêm mới bác sĩ được thiết kế để hỗ trợ quản trị viên (admin) nhập và quản lý thông tin của đội ngũ bác sĩ một cách chi tiết và hiệu quả trong hệ thống. Khi sử dụng chức năng này, người dùng cần cung cấp các thông tin cần thiết để xác định và phân loại bác sĩ. Cụ thể, các thông tin bao gồm: họ và tên, ngày sinh, giới tính, số điện thoại, địa chỉ, số CCCD, email, khoa chuyên môn, số ngày nghỉ trong tuần, và lịch làm việc theo các ca cụ thể.



Hình 2.2.8 Mô tả chức năng thêm một bác sĩ

2.2.9. Xóa một bác sĩ

Chức năng này cho phép xóa thông tin của bác sĩ khỏi hệ thống dựa trên mã bệnh nhân. Nếu mã bác sĩ có tồn tại thì hệ thống sẽ tiến hành xóa bác sĩ có mã vừa nhập và thông báo đã xóa thành công, còn nếu không tồn tại mã đó thì hệ thống in ra không tìm thấy hồ sơ bác sĩ có mã vừa nhập.



Hình 2.2.9 Mô tả chức năng xóa một bác sĩ

2.2.10. Hiển thị danh sách các bác sĩ

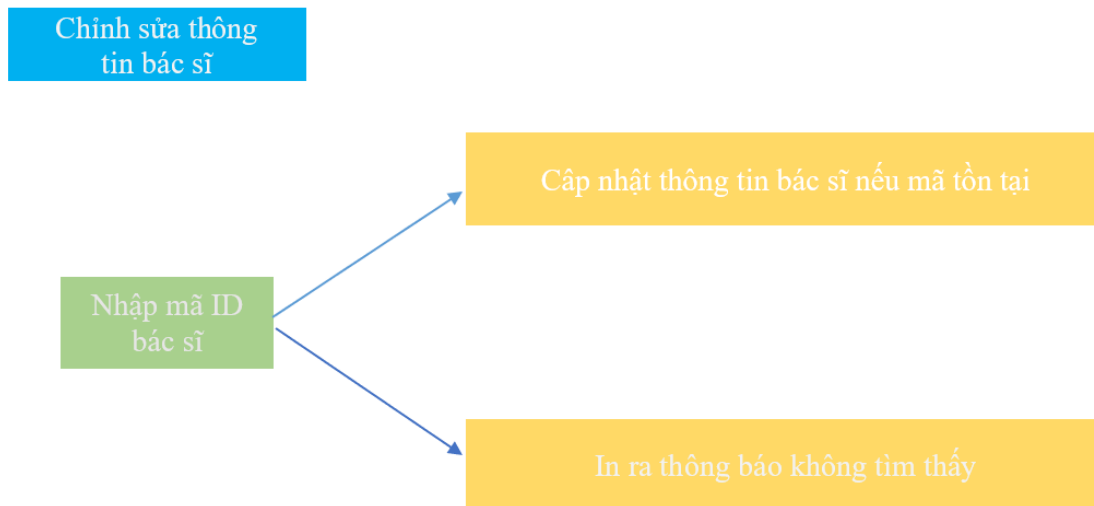
Chức năng này cho phép admin xem danh sách tất cả các bác sĩ hiện đang được lưu trong hệ thống. Danh sách sẽ hiển thị các thông tin cơ bản của bác sĩ, bao gồm mã số bác sĩ, họ tên, số điện thoại, khoa, email, CCCD, giới tính, địa chỉ, số ngày nghỉ, ca làm.



Hình 2.2.10 Mô tả chức năng hiển thị danh sách bác sĩ

2.2.11. Chỉnh sửa một bác sĩ

Chức năng này cho admin cập nhật lại thông tin của bác sĩ. Hệ thống yêu cầu nhập mã của bác sĩ cần cập nhật, nếu mã bác sĩ có tồn tại thì hệ thống sẽ cho chúng ta nhập lại các thông tin của bác sĩ như tên, số điện thoại, CCCD.... Và khi cập nhật xong hệ thống sẽ thông báo đã cập nhật thành công, còn nếu không tồn tại mã đó thì hệ thống in ra không tìm thấy hồ sơ.



Hình 2.2.11 Mô tả chức năng chỉnh sửa một bác sĩ

2.2.12. Hiện thị danh sách các bác sĩ đang trong ca làm việc

Chức năng này cho phép admin hiển thị danh sách các bác sĩ đang trong ca làm việc. Điều này hỗ trợ việc quản lý tài nguyên hiệu quả và lập kế hoạch phòng ngừa, điều trị phù hợp. Bệnh nhân ngoại trú sẽ biết được ca làm của bác sĩ và sẽ có kế hoạch tái khám hợp lý.



Hình 2.2.12 Mô tả chức năng hiển thị các bác sĩ đang trong ca làm

2.2.13. Hiện thị danh sách các loại thuốc

Chức năng này cung cấp danh sách chi tiết về các loại thuốc mà bệnh nhân đã sử dụng, bao gồm mã thuốc, tên thuốc, số lượng, liều lượng, và hạn sử dụng. Nhờ đó, bệnh viện có thể theo dõi hiệu quả việc sử dụng thuốc, đồng thời hỗ trợ bác sĩ quản lý tình trạng kho thuốc. Việc tổng hợp đầy đủ thông tin về các loại thuốc đã được kê đơn giúp bệnh viện lập kế hoạch cung ứng chính xác, đảm bảo sẵn sàng đáp ứng nhu cầu điều trị mà không gây lãng phí hoặc thiếu hụt thuốc.

Thông kê thuốc

Hiển thị danh sách các loại thuốc

Hình 2.2.13 Mô tả chức năng hiển thị danh sách các loại thuốc

2.2.14. Hiển thị danh sách các loại thuốc còn có thể sử dụng

Chức năng này cung cấp danh sách chi tiết các loại thuốc còn hạn sử dụng. Điều này giúp bác sĩ và nhân viên y tế dễ dàng xác định các loại thuốc đã hết hạn, từ đó nhanh chóng loại bỏ khỏi kho để đảm bảo không có thuốc quá hạn được cung cấp cho bệnh nhân.

Ngoài ra, chức năng này còn hỗ trợ bệnh viện tối ưu hóa quản lý thuốc, giảm thiểu tình trạng lãng phí do thuốc hết hạn và hạn chế rủi ro khi sử dụng thuốc không đảm bảo chất lượng. Việc cập nhật thường xuyên danh sách này giúp bác sĩ đưa ra các quyết định kê đơn chính xác, phù hợp với nguồn thuốc còn sử dụng được.

Thông kê thuốc
còn dùng được

Hiển thị danh sách các loại thuốc còn sử dụng được

Hình 2.2.14 Mô tả chức năng hiển thị các loại thuốc có thể sử dụng

2.2.15. Phương thức lưu trữ

Trong hệ thống quản lý thông tin bệnh nhân và bác sĩ, dữ liệu được lưu trữ thông qua phương thức đọc và ghi file **.txt**, một giải pháp đơn giản và hiệu quả cho các ứng dụng nhỏ và vừa. Phương thức này cho phép lưu trữ dữ liệu dưới dạng văn bản, giúp dễ dàng kiểm tra, chỉnh sửa, và sao lưu mà không yêu cầu cơ sở dữ liệu phức tạp. Mỗi khi thực hiện các thao tác như thêm, sửa, xóa hoặc tìm kiếm thông tin, hệ thống sẽ truy cập các file **.txt** để đọc dữ liệu hiện có hoặc cập nhật nội dung mới.

Dữ liệu được tổ chức theo cấu trúc định sẵn trong file, với các trường thông tin được phân tách bằng ký tự đặc biệt (chẳng hạn như dấu phẩy, tab, hoặc xuống dòng) để đảm bảo tính rõ ràng và dễ dàng xử lý. Ví dụ, thông tin của một bệnh nhân hoặc bác sĩ sẽ được lưu trên một dòng, với các trường như họ tên, mã định danh, ngày sinh, hoặc chuyên môn được ghi tuần tự. Khi cần truy xuất, hệ thống sẽ đọc file, phân tích từng dòng, và ánh xạ thông tin vào các đối tượng tương ứng trong chương trình.

Ưu điểm của phương thức này là dễ triển khai, không đòi hỏi các thư viện hoặc công cụ phức tạp, và có thể hoạt động trên nhiều nền tảng. Tuy nhiên, để đảm bảo dữ liệu không bị mất mát hoặc lỗi trong trường hợp file bị chỉnh sửa sai định dạng, hệ thống cũng cần tích hợp các cơ chế kiểm tra và xử lý lỗi khi đọc ghi dữ liệu. Phương thức này phù hợp với mô hình hướng đối tượng, khi mỗi lớp có thể quản lý dữ liệu của chính mình và thao tác trực tiếp với file tương ứng, từ đó duy trì tính tách biệt và dễ bảo trì của mã nguồn.

3. THIẾT KẾ CẤU TRÚC DỮ LIỆU

3.1. Phát biểu bài toán

Hệ thống cần quản lý và xử lý dữ liệu về bệnh nhân một cách nhanh chóng và hiệu quả. Do đó, cấu trúc dữ liệu phải đảm bảo lưu trữ thông tin một cách logic và có tổ chức, để các thao tác như tìm kiếm, cập nhật và xóa thông tin có thể được thực hiện nhanh chóng.

- **Đầu vào:** Hệ thống yêu cầu quản lý thông tin bệnh nhân hoặc bác sĩ. Các dữ liệu này sẽ được nhập từ người dùng hoặc từ các file dữ liệu có sẵn.
- **Đầu ra:** Hệ thống sẽ hiển thị các thông tin bệnh nhân hoặc bác sĩ.

Sử dụng class để định nghĩa các đối tượng như Medicine, Disease, Availabe, Person, Patient, và Doctor nhằm nhóm các thuộc tính và chức năng liên quan vào cùng một đối tượng, giúp việc quản lý dễ dàng hơn và tránh lặp lại mã.

3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống

3.2.1. Phân tích cấu trúc dữ liệu và thuật toán

+ **Vector:** là một cấu trúc dữ liệu quan trọng trong lập trình, hoạt động giống như một mảng động với khả năng thay đổi kích thước tự động. Nó thường được sử dụng để lưu trữ và quản lý dữ liệu tuyến tính trong các chương trình yêu cầu sự linh hoạt cao về kích thước và thao tác trên các phần tử.

Đặc điểm nổi bật của Vector:

- **Kích thước động:** Vector có thể tự động thay đổi kích thước để phù hợp với số lượng phần tử hiện tại. Khi thêm hoặc xóa phần tử, vector sẽ tự động mở rộng hoặc thu nhỏ bộ nhớ mà không yêu cầu sự can thiệp thủ công từ lập trình viên.
- **Truy cập ngẫu nhiên:** Vector cho phép truy cập trực tiếp đến từng phần tử thông qua chỉ số, tương tự như mảng, giúp thao tác với dữ liệu trở nên nhanh chóng và thuận tiện.
- **Quản lý bộ nhớ tự động:** Khi kích thước của vector thay đổi, nó sẽ tự động quản lý việc cấp phát và giải phóng bộ nhớ, giảm bớt gánh nặng xử lý bộ nhớ thủ công và hạn chế lỗi liên quan đến quản lý bộ nhớ.
- **Ho Tăng kích thước linh hoạt:** Khi dung lượng hiện tại của vector không đủ, nó sẽ tự động cấp phát lại bộ nhớ với dung lượng lớn hơn (thường gấp đôi) và sao chép dữ liệu cũ sang vùng nhớ mới, đảm bảo hiệu suất khi thêm dữ liệu.

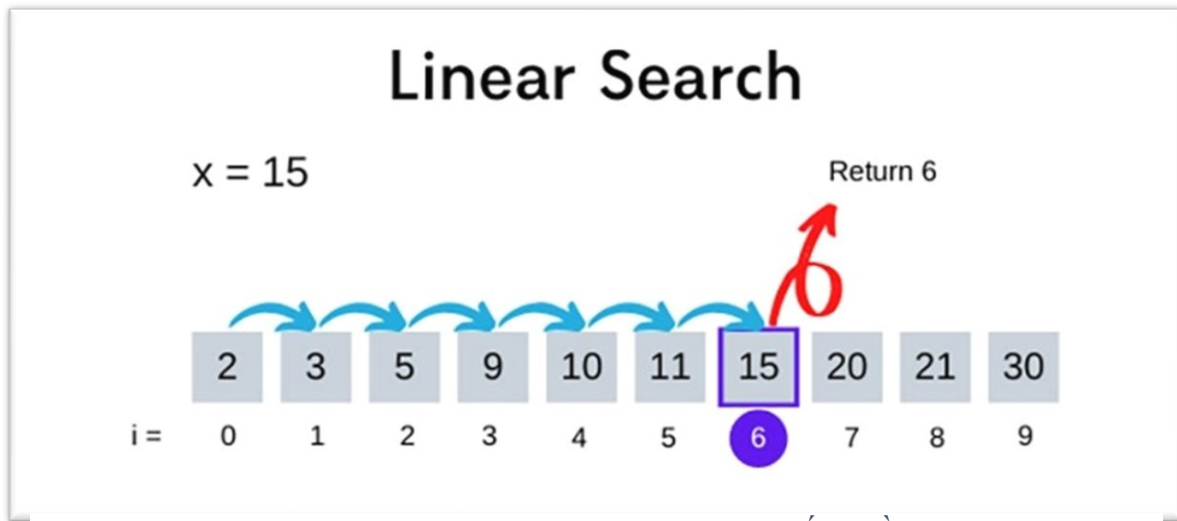
- **Tốc độ truy cập cao:** Với thời gian truy cập trung bình $O(1)$, vector là một lựa chọn tối ưu cho các bài toán yêu cầu tốc độ truy xuất phần tử nhanh.

Các thao tác phổ biến trên Vector:

- **Thêm phần tử** `push_back(element)`: Thêm một phần tử vào cuối vector. Nếu cần, vector sẽ tự động mở rộng bộ nhớ.
- **Xóa phần tử:**
 - **`pop_back()`** Loại bỏ phần tử cuối cùng, giải phóng bộ nhớ tương ứng.
 - **`erase(position)`**: xóa một hoặc nhiều phần tử tại vị trí xác định.
- **Chèn phần tử** `insert(position, value)` : chèn một hoặc nhiều phần tử vào vị trí chỉ định trong vector.
- **Truy cập phần tử** `operator[]`: truy cập vào phần tử dựa trên chỉ số.
- **Kích thước** `size()`: trả về số lượng phần tử hiện có trên vector

+ Thuật toán tìm kiếm tuần tự (Linear search)

- **Linear Search**, hay còn gọi là tìm kiếm tuyến tính, là một thuật toán đơn giản được sử dụng để tìm kiếm một phần tử cụ thể trong danh sách. Thuật toán này duyệt tuần tự từng phần tử trong danh sách từ đầu đến cuối, so sánh từng phần tử với giá trị cần tìm.
- **Ý tưởng:**
 1. Bắt đầu từ phần tử đầu tiên của danh sách và lặp qua từng phần tử.
 2. Ở mỗi bước lặp, so sánh giá trị phần tử hiện tại với giá trị cần tìm.
 - Nếu khớp, trả về vị trí hoặc con trỏ của phần tử
 - Nếu không khớp, tiếp tục lặp.
 3. Nếu duyệt hết danh sách mà không tìm thấy phần tử, trả về NULL hoặc giá trị báo hiệu không tìm thấy.
- **Độ phức tạp về thời gian:**
 - **Trường hợp xấu nhất:** $O(n)$ (khi phần tử nằm ở cuối danh sách hoặc không tồn tại).
 - **Trường hợp tốt nhất:** $O(1)$ (khi phần tử nằm ở đầu danh sách).



Hình 3.2.1 Mô tả thuật toán tìm kiếm tuần tự

3.2.2. Ứng dụng trong hệ thống

- **Danh sách bệnh nhân và bác sĩ:**

Hệ thống sử dụng **mảng động** (dynamic array) thông qua cấu trúc như vector để lưu trữ danh sách bệnh nhân và bác sĩ. Điều này cho phép dễ dàng thêm, xóa và quản lý thông tin mới mà không bị giới hạn bởi kích thước ban đầu. Cấu trúc này giúp đảm bảo việc mở rộng danh sách khi hệ thống phát triển.

- **Thông tin bệnh nhân:**

Mỗi bệnh nhân được biểu diễn thông qua lớp **Patient**, chứa các thuộc tính quan trọng như mã bệnh nhân, họ tên, ngày sinh, giới tính, mã BHYT, địa chỉ, và chẩn đoán. Lớp này kế thừa từ lớp **Person**, kế thừa các thông tin chung của một cá nhân như họ tên, số điện thoại, CCCD và giới tính, nhằm đảm bảo tính nhất quán và tái sử dụng.

Ngoài ra, với bệnh nhân nội trú và ngoại trú, hệ thống mở rộng thêm lớp **Inpatient** và **Outpatient** để lưu trữ thông tin đặc thù như số ngày nằm viện, phòng bệnh hoặc lịch hẹn tái khám. Điều này giúp quản lý chi tiết từng nhóm bệnh nhân mà vẫn duy trì cấu trúc hệ thống rõ ràng.

- **Quản lý bác sĩ:**

Thông tin của bác sĩ được tổ chức trong lớp **Doctor**, bao gồm các thuộc tính như mã bác sĩ, chuyên khoa, phòng làm việc và lịch làm việc. Lớp này kế thừa từ **Person** để tận

dụng các thông tin chung và bổ sung các tính năng như lưu trữ ca trực (shift) hoặc ngày nghỉ (NoD) thông qua lớp **Available**.

- **Đơn thuốc và thông tin điều trị:**

Hệ thống quản lý đơn thuốc của bệnh nhân thông qua lớp **Medicine**, bao gồm các thuộc tính như mã thuốc, tên thuốc, số lượng, hướng dẫn sử dụng, và hạn dùng. Điều này cho phép lưu trữ thông tin điều trị cụ thể, đồng thời hỗ trợ bác sĩ quản lý toa thuốc theo từng bệnh nhân.

- **Phân loại và thống kê:**

Hệ thống sử dụng cấu trúc **map** để phân loại bệnh nhân dựa trên các tiêu chí như khoa điều trị, loại bệnh, hoặc thời gian nhập viện. Các dữ liệu này được liên kết với lớp **Disease**, giúp hỗ trợ truy vấn nhanh và cung cấp báo cáo tổng quan, chẳng hạn như số lượng bệnh nhân trong từng khoa hoặc tình trạng bệnh phổ biến.

4. ỨNG DỤNG HƯỚNG ĐỐI TƯỢNG VÀO BÀI TOÁN

Hướng đối tượng là một phương pháp tiếp cận hiệu quả trong việc thiết kế và phát triển các hệ thống quản lý phức tạp, đặc biệt đối với bài toán quản lý bệnh nhân và bác sĩ. Thay vì xử lý dữ liệu và chức năng theo cách riêng lẻ, hệ thống được tổ chức thành các lớp đại diện cho từng thực thể chính. Mỗi lớp bao gồm các thuộc tính mô tả dữ liệu đặc trưng và các phương thức thực hiện các chức năng liên quan, tạo nên một cấu trúc hệ thống mạch lạc và dễ hiểu.

Cách tiếp cận này không chỉ giúp hệ thống trở nên rõ ràng và dễ quản lý mà còn hỗ trợ việc mở rộng trong tương lai. Khi cần bổ sung thêm chức năng hoặc thông tin mới, việc sửa đổi chỉ cần tập trung vào lớp liên quan mà không ảnh hưởng đến toàn bộ hệ thống. Ngoài ra, việc kế thừa các thuộc tính và phương thức từ lớp cha sang lớp con giúp giảm thiểu sự dư thừa mã nguồn, đồng thời duy trì sự nhất quán trong các thực thể.

Áp dụng hướng đối tượng trong hệ thống quản lý bệnh viện không chỉ làm tăng tính logic và tổ chức mà còn đảm bảo rằng hệ thống có khả năng phát triển lâu dài, đáp ứng tốt các yêu cầu mới mà không làm gián đoạn hoạt động hiện tại.

4.1. Xác Định Các Thực Thể Chính Trong Bài Toán

Trong hệ thống quản lý bệnh viện, các thực thể cốt lõi bao gồm bệnh nhân và bác sĩ, với mỗi thực thể được tổ chức nhằm đáp ứng nhu cầu quản lý và vận hành hiệu quả.

Đối với bệnh nhân, hệ thống tập trung vào việc lưu trữ và quản lý toàn diện các thông tin quan trọng. Các thông tin này bao gồm: họ tên, ngày tháng năm sinh, địa chỉ thường trú, số điện thoại liên lạc, mã bệnh nhân (một định danh duy nhất), thông tin bảo hiểm y tế (nếu có), chẩn đoán y khoa, quá trình điều trị (đối với bệnh nhân nội trú), đơn thuốc đã kê, và lịch hẹn tái khám (đối với bệnh nhân ngoại trú). Những dữ liệu này không chỉ giúp bệnh viện theo dõi sát sao từng trường hợp mà còn tạo cơ sở dữ liệu cần thiết để hỗ trợ quá trình phân tích và tối ưu hóa các dịch vụ y tế.

Với bác sĩ, hệ thống tập trung vào việc quản lý các thông tin chuyên môn và lịch trình làm việc. Các dữ liệu chính bao gồm: họ tên đầy đủ, mã định danh bác sĩ (để đảm bảo tính duy nhất), chuyên môn hoặc khoa phụ trách, lịch làm việc chi tiết theo từng ngày và từng ca, cũng như thông tin về thời gian nghỉ phép. Những thông tin này hỗ trợ bệnh viện trong việc phân bổ nguồn lực hiệu quả, sắp xếp lịch khám hợp lý, và duy trì sự cân đối trong công việc của đội ngũ y tế.

4.2. Phân tích các lớp của hướng đối tượng

Lớp Person : là lớp cơ sở chung được thiết kế để lưu trữ thông tin cơ bản của một cá nhân. Các thuộc tính chính bao gồm: họ tên, giới tính, ngày sinh, địa chỉ, số điện thoại và mã định danh cá nhân (CCCD). Đây là những thuộc tính phổ biến và cần thiết để xác định danh tính và thông tin liên hệ. Việc sử dụng lớp Person làm cơ sở giúp giảm thiểu sự trùng lặp mã khi các lớp khác như patient hay doctor kế thừa, đảm bảo rằng các thông tin chung không cần được định nghĩa lại. Điều này cũng hỗ trợ dễ dàng mở rộng hoặc chỉnh sửa dữ liệu cơ bản mà không ảnh hưởng đến cấu trúc của các lớp con. Lớp này đặt nền móng cho việc tổ chức và quản lý dữ liệu một cách có hệ thống và rõ ràng.

class person
string age string name string address string phone string date string CCCD string sex
void getPersonData()

Hình 4.2.1 Mô tả lớp Person

- Các thuộc tính với quyền truy cập protected:
 - age : tuổi.
 - name : họ và tên.
 - address : địa chỉ .
 - phone : số điện thoại.
 - date : ngày-tháng-năm sinh.
 - CCCD : căn cước công dân.
 - Sex : giới tính.
- Các phương thức với truyền truy cập public:

- `getPersondata()`: hàm nhập thông tin.

Lớp disease : được thiết kế để quản lý thông tin chi tiết về bệnh lý. Bao gồm các thuộc tính: tên bệnh (diseasename), triệu chứng (symptoms), phương pháp điều trị (treatment), và khoa điều trị (department). Lớp này hỗ trợ việc phân loại bệnh, định hướng khoa điều trị phù hợp, và cung cấp thông tin cần thiết để chẩn đoán cũng như lên kế hoạch điều trị. Việc bổ sung lớp Disease giúp chuẩn hóa dữ liệu bệnh lý, dễ dàng tích hợp và mở rộng trong hệ thống quản lý y tế.

class disease
string diseasename string symptoms string treatment string department
void getDiseaseDetail()

Hình 4.2.2 Mô tả lớp disease

- Các thuộc tính với quyền truy cập protected:
 - `diseasename` : tên bệnh.
 - `symptoms` : triệu chứng.
 - `treatment` : cách điều trị.
- Các phương thức với truyền truy cập public:
 - `getDiseaseDetail()`: nhập thông tin về loại bệnh.

Lớp Patient : kế thừa từ lớp Person và mở rộng thêm các thuộc tính chuyên biệt cho bệnh nhân, bao gồm mã bệnh nhân (patientId), mã bảo hiểm y tế (BHYT), cân nặng (weight), và ngày hẹn khám (appointDate). Những thông tin này là cần thiết để quản lý hồ sơ y tế, phục vụ công tác khám chữa bệnh và theo dõi lịch trình điều trị. Việc kế thừa từ

lớp Person đảm bảo rằng mỗi bệnh nhân đều có các thông tin cơ bản được quản lý đồng nhất, đồng thời cung cấp nền tảng cho các lớp con như Inpatient và Outpatient đại diện cho hai loại bệnh nhân. Lớp này đảm bảo tính linh hoạt trong việc xử lý các trường hợp khác nhau và hỗ trợ lưu trữ thông tin chi tiết liên quan đến bệnh án.

class patient
string patientId string BHYT string weight string appointDate
virtual void getPatientInformation() = 0 virtual void addPatient() = 0 virtual void showPatientDataFromFile() = 0 virtual void dischargePatient() = 0 virtual void editPatient() = 0 virtual void xuatinfor() = 0 virtual void disease_statistics() = 0 virtual void searchpatient() = 0 virtual void patienchoice() = 0

Hình 4.2.3 Mô tả lớp patient

- Các thuộc tính với quyền truy cập protected:
 - patientId : mã ID của bệnh nhân.
 - BHYT : mã bảo hiểm y tế của bệnh nhân.
 - weight: cân nặng của bệnh nhân.
 - appointDate: ngày nhập viện.
- Các phương thức với truyền truy cập public:
 - getPatientInformation() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.
 - addPatient() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.
 - dischargePatient() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.
 - editPatient() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.
 - xuatinfor() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.
 - disease_statistics() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.

- searchpatient() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.
- patientchoice() hàm thuần ảo để các lớp dẫn xuất có thể định nghĩa lại.

Lớp Inpatient : mở rộng từ lớp patient, đại diện cho các bệnh nhân nội trú. Lớp này bổ sung các thuộc tính như số ngày nằm viện (xuativien) và số phòng điều trị (roomNumber). Những thông tin này phục vụ việc quản lý chặt chẽ quá trình điều trị tại bệnh viện, bao gồm phân bổ phòng và giường bệnh, theo dõi thời gian lưu trú, và các chi phí liên quan. Lớp này cung cấp khả năng quản lý chi tiết đối với bệnh nhân nội trú, đảm bảo hệ thống hoạt động hiệu quả khi xử lý số lượng lớn bệnh nhân cần theo dõi dài hạn.

class Inpatient
string xuativien string roomNumber
bool checkinpatient(string s) void getPatientInformation() void addPatient() void showPatientDataFromFile() void dischargePatient() void editPatient() void xuatinfor() void disease_statistics() void searchpatient() void patienchoice()

Hình 4.2.4 Mô tả lớp Inpatient

- Các thuộc tính với quyền truy cập private:
 - xuativien: ngày bệnh nhân xuất viện.
 - roomNumber: phòng bệnh nhân điều trị.
- Các phương thức với truyền truy cập public:
 - Checkinpatient() kiểm tra mã ID của các bệnh nhân không được trùng lặp.
 - getPatientInformation() nhập thông tin cho các bệnh nhân nội trú.
 - addPatient() thêm một bệnh nhân nội trú.

- dischargePatient() xóa một bệnh nhân nội trú.
- editPatient() chỉnh sửa bệnh nhân nội trú.
- xuatinfor() xuất ra các bệnh nhân nội trú nhập viện trong thời điểm nhất định.
- disease_statistics() thống kê các loại bệnh của những bệnh nhân nội trú.
- searchpatient() tìm kiếm bệnh nhân nội trú.
- patientchoice() dùng để chọn lựa các chứng năng để quản lý bệnh nhân.

Lớp Outpatient : cũng kế thừa từ lớp patient, được thiết kế dành riêng cho các bệnh nhân ngoại trú. Bệnh nhân ngoại trú thường không cần lưu trú dài ngày tại bệnh viện mà chỉ đến khám và điều trị theo lịch hẹn. Lớp này sử dụng các thuộc tính từ lớp cha như mã bệnh nhân, mã bảo hiểm y tế và ngày hẹn khám để tổ chức và theo dõi thông tin. Lớp này đơn giản hóa quy trình quản lý bệnh nhân không cần điều trị nội trú, đồng thời tích hợp tốt với các hệ thống khác như đặt lịch hẹn hoặc kê đơn thuốc.

class Outpatient
string numberOfFollowUps string daysBetweenVisits
bool checkoutpatient(string s) void getPatientInformation() void addPatient() void showPatientDataFromFile() void dischargePatient() void editPatient() void xuatinfor() void disease_statistics() void searchpatient() void patienchoice()

Hình 4.2.5 Mô tả lớp Outpatient

- Các thuộc tính với quyền truy cập private:
- numberOfFollowUps: số lần hẹn tái khám.

- daysBetweenVisits : khoảng cách số ngày giữa mỗi lần khám.
- Các phương thức với truyền truy cập public:
 - Checkinpatient() kiểm tra mã ID của các bệnh nhân không được trùng lặp.
 - getPatientInformation() nhập thông tin cho các bệnh nhân ngoại trú.
 - addPatient() thêm một bệnh nhân ngoại trú.
 - dischargePatient() xóa một bệnh nhân ngoại trú.
 - editPatient() chỉnh sửa bệnh nhân ngoại trú.
 - xuatinfor() xuất ra các bệnh nhân ngoại trú nhập viện trong thời điểm.
 - disease_statistics() thống kê các loại bệnh của những bệnh nhân ngoại trú.
 - searchpatient() tìm kiếm bệnh nhân ngoại trú.
 - patientchoice() dùng để chọn lựa các chứng năng để quản lý bệnh nhân.

Lớp Medicine : được thiết kế để lưu trữ thông tin chi tiết về thuốc, hỗ trợ quản lý việc kê đơn và theo dõi tồn kho thuốc trong hệ thống y tế. Lớp này hỗ trợ tích hợp với các chức năng khác của hệ thống, như quản lý toa thuốc hoặc kiểm tra kho thuốc. Bằng cách lưu trữ thông tin chi tiết về từng loại thuốc, lớp Medicine đảm bảo tính chính xác và giảm nguy cơ sai sót trong việc kê đơn hay sử dụng thuốc đã hết hạn.

class medicine
string medicine_code string medicine_name string quantity string usage_instructions string expiry_date
void getmedicine() void addMedicineData() void showmedicinefromfile() void condungduoc()

Hình 4.2.6 Mô tả lớp medicine

- Các thuộc tính với quyền truy cập protected:
 - medicine_code : mã ID của thuốc.
 - medicine_name : tên thuốc.
 - quantity : số lượng còn lại.
 - usage_instruction : liều lượng.
 - expiry_date : hạn sử dụng.
- Các phương thức với quyền truy cập public:
 - getmedicine() : nhập thông tin thuốc.
 - addMedicineData() : thêm một loại thuốc.
 - showmedicinefromfile() : hiển thị thông tin về thuốc.
 - condungduoc() : hiển thị các loại thuốc còn hạn sử dụng.

Lớp Available : được thiết kế để quản lý lịch làm việc của bác sĩ, cung cấp thông tin về các ngày làm việc và ca làm cụ thể. Lớp này đóng vai trò quan trọng trong việc quản lý lịch trình của bác sĩ, đảm bảo khả năng phối hợp giữa bệnh viện, bác sĩ, và bệnh nhân. Thông tin từ lớp Available có thể được sử dụng để tự động hóa quá trình đặt lịch khám hoặc điều chỉnh lịch làm việc phù hợp với nhu cầu của bệnh viện.

class available
string DayofWeek[7] string shift1 string shift2 string NoD
void whenAvailable() string onlAvailable()

Hình 4.2.7 Mô tả lớp available

- Các thuộc tính với quyền truy cập protected:
 - DayofWeek[7] : xâu ký tự lưu trữ các ngày nghỉ của bác sĩ trong 1 tuần.
 - Shift1 : bắt đầu ca làm việc của bác sĩ.
 - Shift2 : kết thúc ca làm việc của bác sĩ.
 - NoD : số ngày nghỉ của bác sĩ trong tuần.
- Các phương thức với quyền truy cập public:
 - whenAvailable() : nhập thông tin về lịch trình của bác sĩ trong tuần.
 - onlAvailable() : trả về chuỗi string về ca làm của bác sĩ.

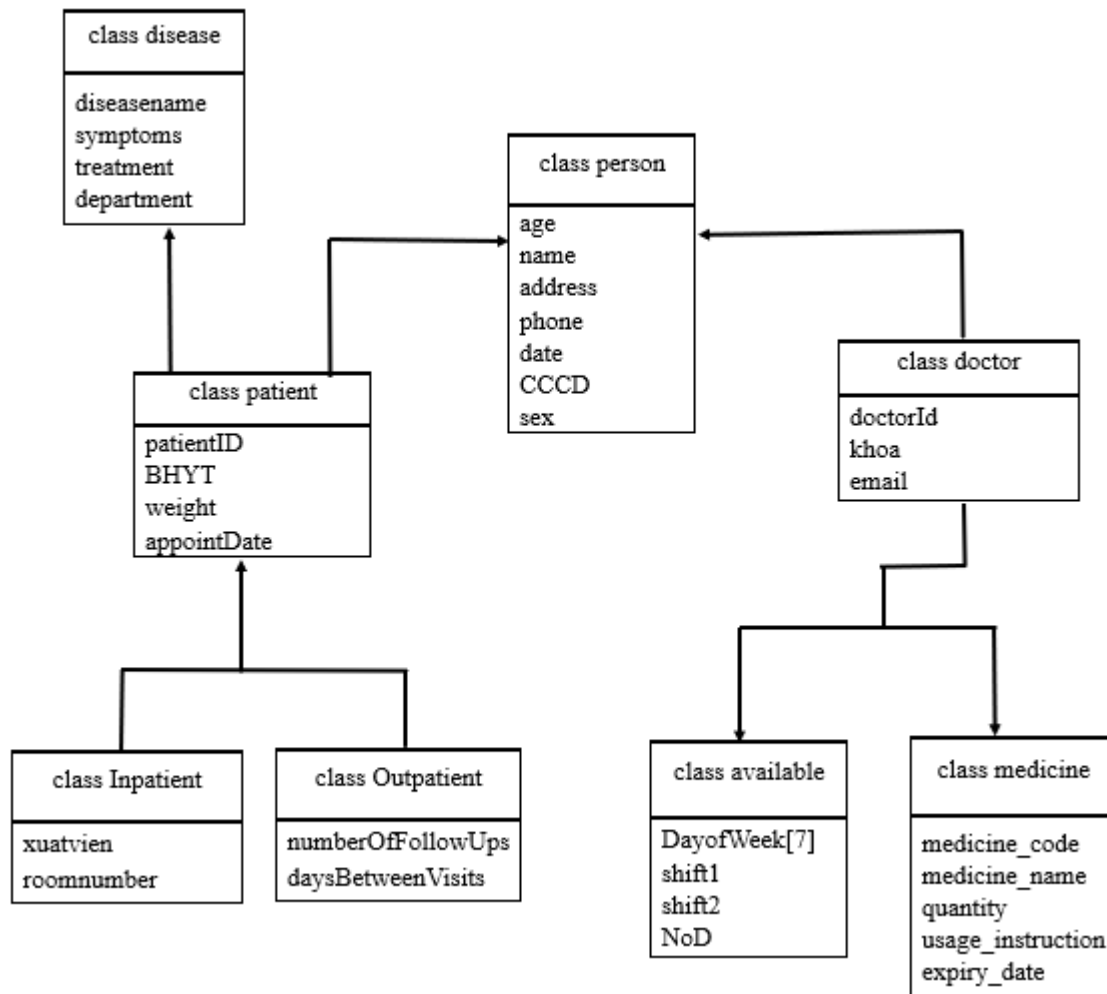
Lớp Doctor : kế thừa từ Lớp Person, Available và Medicine, giúp quản lý thông tin và lịch làm việc của bác sĩ. Các thuộc tính chính của lớp này bao gồm mã bác sĩ (doctorId), khoa chuyên môn (khoa), phòng làm việc (room), và địa chỉ email. Việc sử dụng kế thừa từ lớp Person giúp quản lý các thông tin cơ bản của bác sĩ, trong khi tích hợp từ lớp Available cho phép xử lý lịch làm việc, ca trực, và các khoảng thời gian sẵn sàng. Lớp này đóng vai trò quan trọng trong việc kết nối giữa bệnh nhân và hệ thống điều trị, đảm bảo bác sĩ được phân bổ đúng chuyên môn và lịch trình làm việc phù hợp.

class doctor
string doctorId string khoa string email
bool checkdoctor() void getdoctordata() void addDoctorData() void showDoctorDataFromFile() void deletedoctor() void editDoctor() void online() void showmedici() void showmedicibihu() void doDoctorRelateWork()

Hình 4.2.8 Mô tả lớp doctor

- Các thuộc tính với quyền truy cập private:
 - doctorId : mã ID của bác sĩ.
 - khoa : khoa của bác sĩ.
 - email : email của bác sĩ để bệnh nhân có thể liên lạc.
- Các phương thức với quyền truy cập public:
 - checkdoctor() : kiểm tra mã ID của các bác sĩ không được trùng lặp.
 - getdoctordata() : nhập thông tin cho bác sĩ.
 - addDoctordata() : thêm một bác sĩ.
 - showDoctorData() : hiển thị thông tin các bác sĩ.
 - deletedoctor() : xóa một bác sĩ.
 - editdoctor() : chỉnh sửa thông tin bác sĩ.
 - online() : hiển thị các bác sĩ đang trong ca làm việc.
 - showmedici() : hiển thị danh sách các loại thuốc.
 - showmedicibihu() : hiển thị số lượng thuốc còn sử dụng được.
 - doDoctorRelateWork() : dùng để chọn lựa các chứng năng để quản lý bác sĩ.

4.3. Mối quan hệ



Hình 4.3 Mô tả mối quan hệ giữa các lớp

4.4. Lợi ích của việc ứng dụng hướng đối tượng

Tăng cường tính tổ chức: Với hướng đối tượng, hệ thống được tổ chức thành các thực thể như Patient, Doctor, Medicine, và Available, mỗi thực thể chứa đầy đủ các thuộc tính và phương thức đặc trưng. Điều này giúp hệ thống dễ hiểu hơn, giảm thiểu sự phức tạp trong quản lý và duy trì.

Tái sử dụng mã: Các lớp cơ bản như Person hoặc Disease được thiết kế với mục đích chung, cho phép kế thừa và mở rộng dễ dàng mà không cần viết lại từ đầu. Điều này không chỉ tiết kiệm thời gian mà còn giúp tăng tính đồng nhất trong toàn bộ hệ thống.

Dễ dàng mở rộng: Khi yêu cầu hệ thống thay đổi, như thêm chức năng phân công ca làm việc tự động cho bác sĩ hoặc theo dõi lịch sử điều trị của bệnh nhân, việc bổ sung chỉ cần tạo các lớp mới hoặc mở rộng lớp hiện tại, đảm bảo không làm ảnh hưởng đến phần cốt lõi của hệ thống.

Quản lý dữ liệu hiệu quả hơn: Việc sử dụng các lớp như Patient hay Doctor giúp hệ thống quản lý từng cá nhân một cách trực quan. Dữ liệu được đóng gói trong các đối tượng, giảm nguy cơ nhầm lẫn hoặc sai sót khi thao tác. Hệ thống có thể nhanh chóng truy xuất thông tin cần thiết và đảm bảo tính chính xác cao.

Tăng tính bảo trì: Nhờ tính đóng gói và phân chia rõ ràng các thành phần, việc sửa lỗi hoặc nâng cấp hệ thống trở nên đơn giản hơn. Các thay đổi chỉ cần thực hiện trong một lớp cụ thể mà không gây ảnh hưởng dây chuyền đến toàn bộ hệ thống.

Hỗ trợ đa nhiệm: Với cách tổ chức bằng hướng đối tượng, hệ thống dễ dàng hỗ trợ các tính năng đồng thời như quản lý thông tin bệnh nhân, sắp xếp lịch bác sĩ, và theo dõi tồn kho thuốc một cách độc lập mà không gây xung đột.

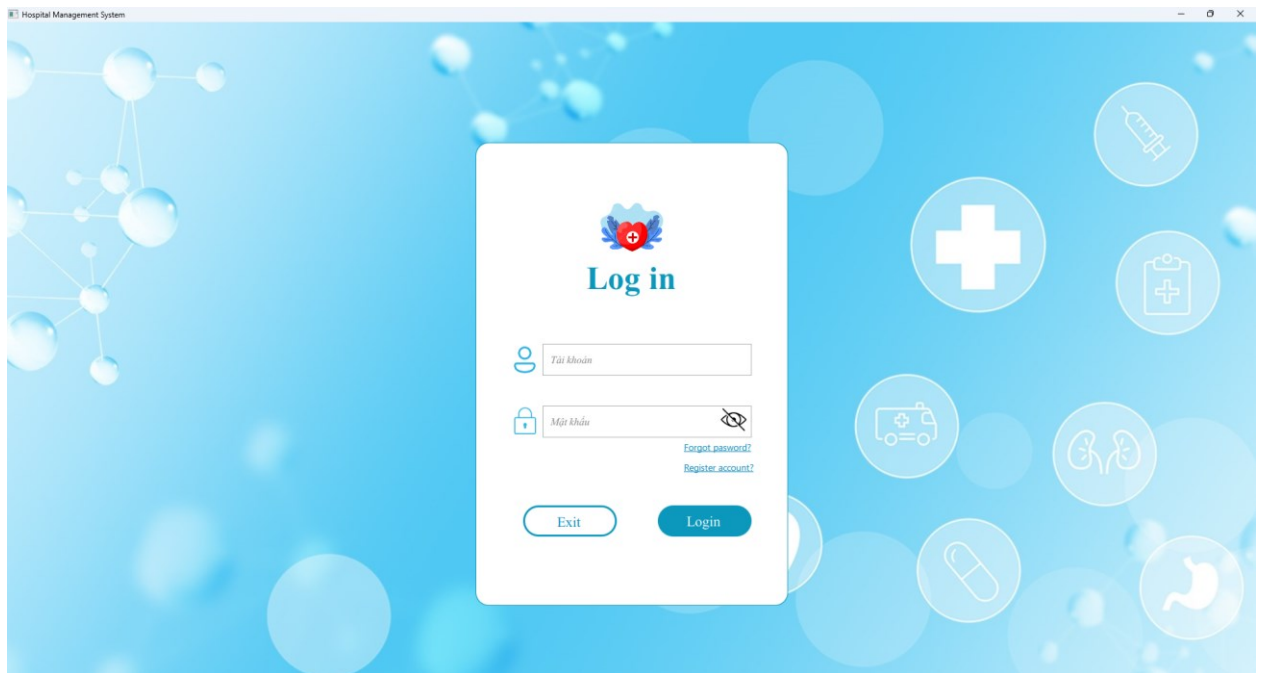
Tăng tính bảo mật: Các lớp có thể được thiết kế với mức độ truy cập cụ thể, như dữ liệu cá nhân của bệnh nhân hoặc lịch sử y tế chỉ có thể được truy cập bởi các thành phần được ủy quyền, giúp tăng cường tính bảo mật cho hệ thống.

Hỗ trợ tính kế thừa trong tương lai: Khi cần tích hợp các tính năng mới hoặc kết nối với hệ thống khác, cấu trúc hướng đối tượng cho phép thực hiện các thay đổi này một cách mượt mà và có hệ thống.

4.5. Kết quả

Khi hệ thống quản lý bệnh nhân được triển khai, giao diện sẽ hướng dẫn người dùng thực hiện các chức năng cơ bản như thêm mới, cập nhật, tìm kiếm, xóa và hiển thị danh sách bệnh nhân. Mỗi thao tác được thiết kế đơn giản, dễ sử dụng, đảm bảo người dùng có thể thao tác một cách trực quan và nhanh chóng.

4.5.1. Giao diện chính của chương trình

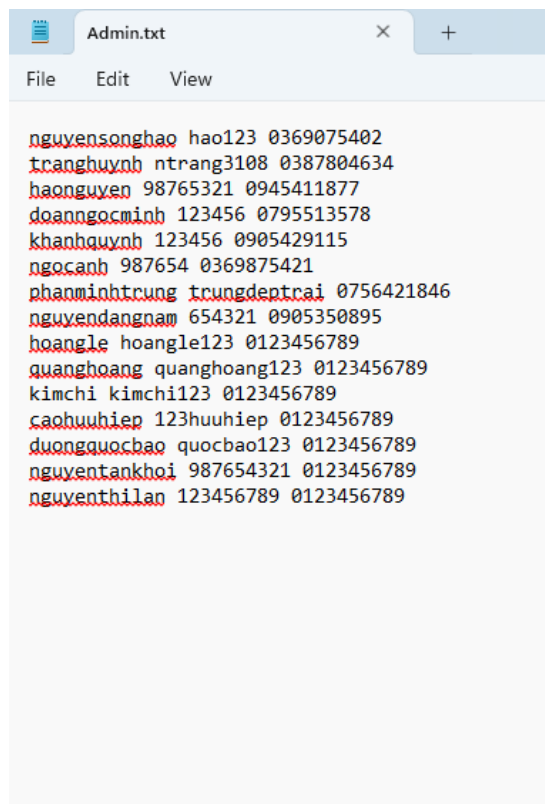


Hình 4.5.1.1 Giao diện đăng nhập chương trình



Hình 4.5.1.2 Giao diện menu lựa chọn

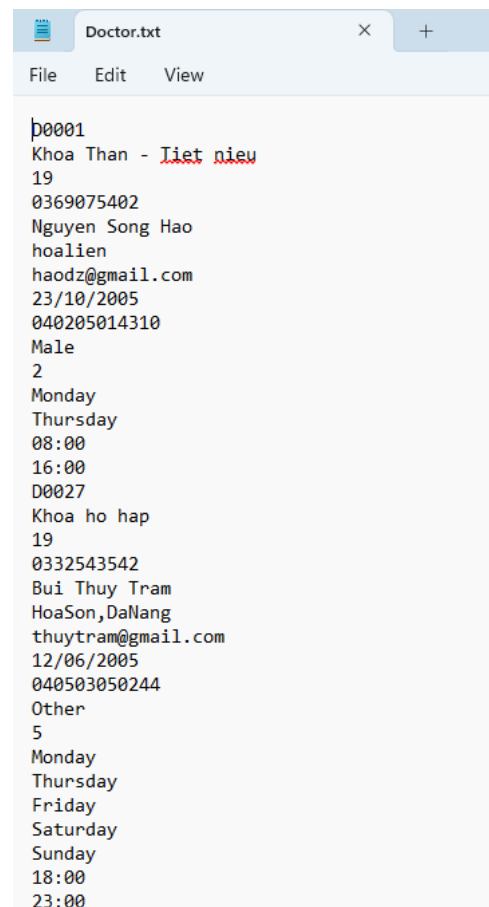
4.5.2. Kết quả thực thi của chương trình



```
Admin.txt
File Edit View
nguyensonghao hao123 0369075402
tranghuy nh ntrang3108 0387804634
haonguyen 98765321 0945411877
doanngocminh 123456 0795513578
khanhquynh 123456 0905429115
ngocanh 987654 0369875421
phanminhtrung trungdeptrai 0756421846
nguyendangnam 654321 0905350895
hoangle hoangle123 0123456789
quanghoang quanghoang123 0123456789
kimchi kimchi123 0123456789
cachuuhiiep 123huuhiiep 0123456789
duongquochao quocbao123 0123456789
nguyentankhoi 987654321 0123456789
nguyenthilan 123456789 0123456789
```

Hình 4.5.2.1 File quản lý các tài khoản của Admin

Hình 4.5.2.2 File quản lý thông tin tài khoản của Bác sĩ



```
Doctor.txt
File Edit View
p0001
Khoa Than - Tiet nien
19
0369075402
Nguyen Song Hao
hoalien
haodz@gmail.com
23/10/2005
040205014310
Male
2
Monday
Thursday
08:00
16:00
D0027
Khoa ho hap
19
0332543542
Bui Thuy Tram
HoaSon,DaNang
thuytram@gmail.com
12/06/2005
040503050244
Other
5
Monday
Thursday
Friday
Saturday
Sunday
18:00
23:00
```

```
In_patient.txt
File Edit View

I0001
123456789
153759555
Nguyen Van A
Thanh Khe Da Nang
20/11/2005
048205987654
Male
Thai hoa khop
Dau cot song
chup x quang
Khoa Xuong Khop
90
170
16/11/2024
14
B101
I0002
123456789
153759555
Tran Thi B
Thanh Khe Da Nang
30/05/2002
048205987654
Female
Thai hoa khop
Dau cot song
chup x quang
Khoa Xuong Khop
90
170
16/11/2024
14
B101
```

Hình 4.5.2.3 File quản lý thông tin bệnh nhân nội trú

```
Out_patient.txt
File Edit View

00001
BHYT098712345
0905452116
Tran Van Ca
Thanh Khe Da Nang
20/11/2005
048205987654
Male
Thai hoa khop
Dau cot song
chup x quang
Khoa Xuong Khop
90
170
16/11/2024
2
7
00002
BHYT456245132
0987452169
Nguyen Ba Canh
Thanh Khe Da Nang
30/05/2002
048205987654
Male
Thai hoa khop
Dau cot song
chup x quang
Khoa Xuong Khop
90
170
16/10/2024
5
3
```

Hình 4.5.2.4 File quản lý thông tin bệnh nhân ngoại trú

```
Medicine.txt
File Edit View

M4735
Paracetamol
50
1 vien/lan, 2-3 lan/ngay
12/12/2023
M4614
Ibuprofen
104
1 vien/lan, 3 lan/ngay
12/12/2024
M1846
Amoxicillin
0
2 vien/lan, 2 lan/ngay
12/02/2024
M1756
Loratadine
70
1 vien/ngay
22/01/2025
M5739
Cetirizine
0
1 vien/lan, 2 lan/ngay
12/02/2024
M3754
Metformin
40
3 vien/lan, 1 lan/ngay
12/12/2023
```

Hình 4.5.2.5 File quản lý thông tin các loại thuốc

Register account

Enter username
doanngocminh1511

Enter password
123456789

Enter your phone
0795513XXX

Enter Passcode
XXXXXX

Cancel Register

Hình 4.5.2.1 Giao diện đăng kí tài khoản admin

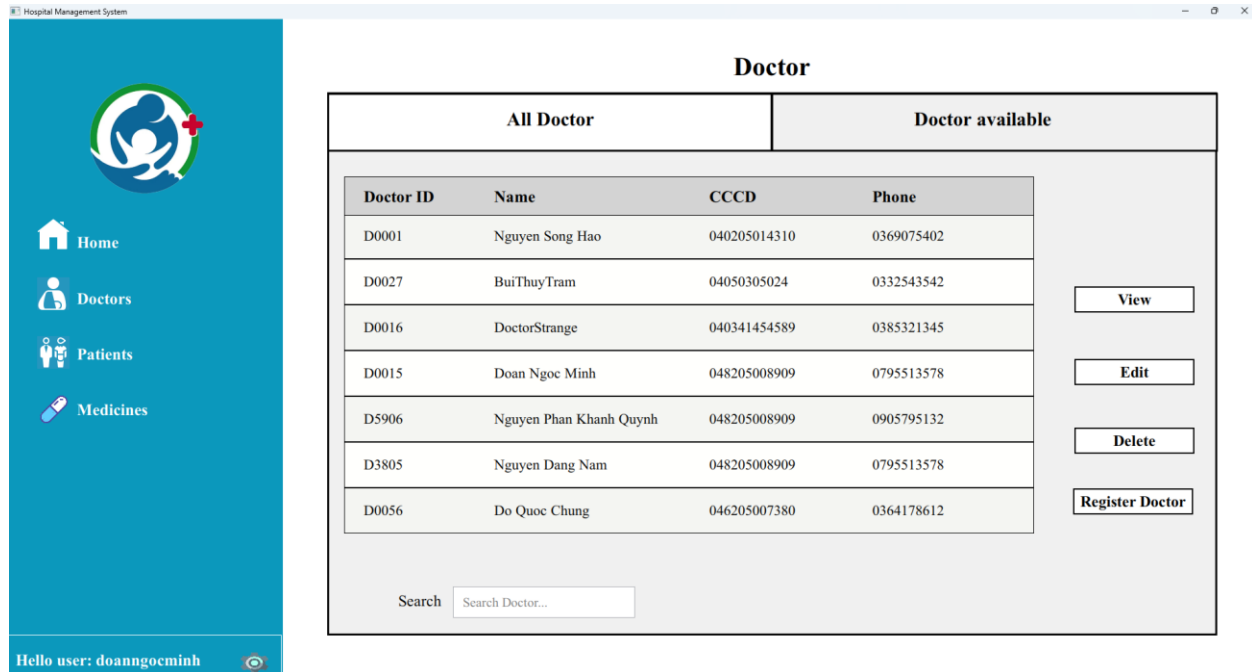
Get pasword

Enter your username
doanngocminh1511

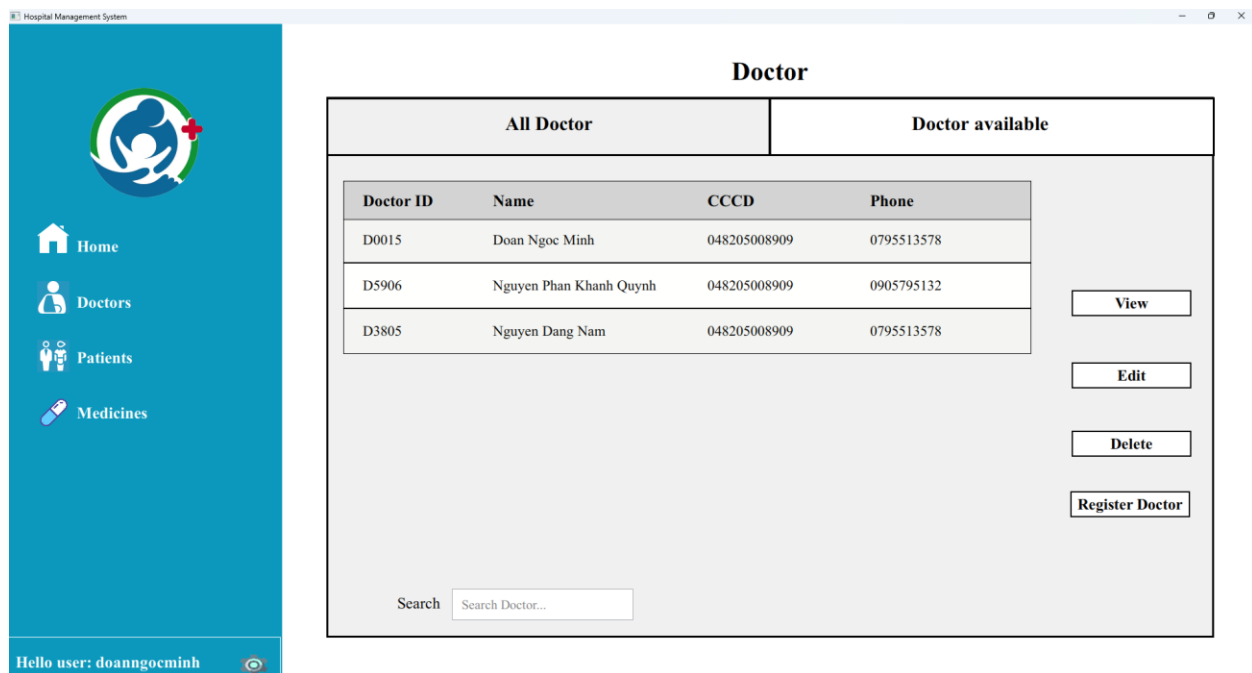
Enter your phone
0795513XXX

Cancel Get pass

Hình 4.5.2.2 Giao diện lấy lại mật khẩu đăng nhập



Hình 4.5.2.3 Giao diện hiển thị danh sách bác sĩ, với các chức năng



Hình 4.5.2.4 Giao diện hiển thị các bác sĩ hiện tại đang trong ca làm việc

View Doctor Information

Name Nguyen Song Hao	ID Number D0001	CNIC 040205014310
Phone No. 0369075402	Mail haodz@gmail.com	Date of Birth 23/10/2005
Gender Male	Address hoalien	
Department Khoa Than - Tiet nieu	Working hours From 08:00 to 16:00	Days <input checked="" type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input type="checkbox"/> Wednesday <input type="checkbox"/> Thursday <input type="checkbox"/> Friday <input type="checkbox"/> Saturday <input type="checkbox"/> Sunday

Cancel

Hình 4.5.2.5 Giao diện xem thông tin của Bác sĩ

Edit Doctor Information

Name Buihuytram	ID Number D0027	CNIC 04050305024
Phone No. 0332543542	Mail thuytram@gmail.com	Date of Birth 12/06/2005
Gender Female	Address Notificants	
Department Khoa Than - Tiet nieu	Working hours From 12:00 to 21:00	Days <input checked="" type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input type="checkbox"/> Wednesday <input checked="" type="checkbox"/> Thursday <input type="checkbox"/> Friday <input type="checkbox"/> Saturday <input checked="" type="checkbox"/> Sunday

Are you sure you want to edit this doctor?

Yes No

Cancel Update

Hình 4.5.2.6 Giao diện thực hiện chức năng chỉnh sửa thông tin của bác sĩ

Add Doctor Details

Name
[Text input: Nguyen Van A]

ID Number
[Text input: D00001]

CNIC
[Text input: 048205XXXXXX]

Phone No.
[Text input: 0795513XXX]

Mail
[Text input: XXXXXXXXXX@gmail.com]

Date of Birth
[Text input: 01/01/1990]

sex
[Dropdown: Select sex]

Address
[Text input: Xã xxxxxx, huyện xxxxxx, thành phố xxxxxx...]

Department
[Dropdown: Select Department]

Working hours
From [Dropdown: Select shift 1] to [Dropdown: Select shift 2]

Days
☐ Monday
☐ Tuesday
☐ Wednesday
☐ Thursday
☐ Friday
☐ Saturday
☐ Sunday

Buttons: [Cancel] [Update]

Footer: Hello user: doannhocminh

Hình 4.5.2.7 Thực hiện chức năng thêm 1 bác sĩ

Doctor

All Doctor		Doctor available	
Doctor ID	Name	CCCD	Phone
D0027	BuiThuyTram	04050305024	0332543542
D0015	Doan Ngoc Minh	048205008909	0795513578
D3805	Nguyen Dang Nam	048205008909	0795513578

Buttons: [View] [Edit] [Delete] [Register Doctor]

Search: [Text input: M]

Footer: Hello user: doannhocminh

Hình 4.5.2.8 Giao diện thực hiện chức năng tìm kiếm thông tin của Bác sĩ

Patient

In Patient		Out Patient	
Patient ID	Name	CCCD	Phone
10001	Nguyen Van A	048205987654	153759555
10002	Tran Thi B	048205987654	153759555
10003	Nguyen Tam Nhu	048205987654	090505059508
10096	Huynh Thi Lan	048205996222	0905350892
11005	Tran Quyet Long	048205123456	0913487652

Search Select Day

View Edit Delete Register Patient

Hình 4.5.2.9 Giao diện hiển thị thông tin bệnh nhân nội trú và các chức năng

Patient

In Patient		Out Patient	
Patient ID	Name	CCCD	Phone
102230170	Tran Van Ca	048205987654	153759555
102230159	Nguyen Ba Canh	048205987654	153759555
102245678	Tran Ba Duong	987451234849	07955135436

Search Select Day

View Edit Delete Register Patient

Hình 4.5.2.10 Giao diện hiển thị thông tin của bệnh nhân ngoại trú

Hospital Management System

View In Patient Details

Name
Nguyen Van A

ID Number
I0001

CNIC
048205987654

Health insurance
123456789

Phone No.
153759555

Date of Birth
20/11/2005

Gender
Male

Weight
90

Height
170

Address
Thanh Khe Da Nang

Department
Khoa Xuong Khop

Disease Name
Thai hoa khop

Symptoms
Dau cot song

Treatment
chup x quang

Appoint Date
16/11/2024

Room Number
B101

Discharge Date
14

Cancel

Hello user: doannhocminh

Hình 4.5.2.11 Giao diện xem thông tin bệnh nhân nội trú

Hospital Management System

Edit In Patient Details

Name
Nguyen Van A

ID Number
I0001

CNIC
048205987654

Health insurance
123456789

Phone No.
153759555

Date of Birth
20/11/2005

Gender
Male

Weight
90

Height
170

Address
Thanh Khe Da Nang

Department
Khoa Xuong Khop

Disease Name
Thai hoa khop

Symptoms
Dau cot song

Treatment
chup x quang

Appoint Date
16/11/2024

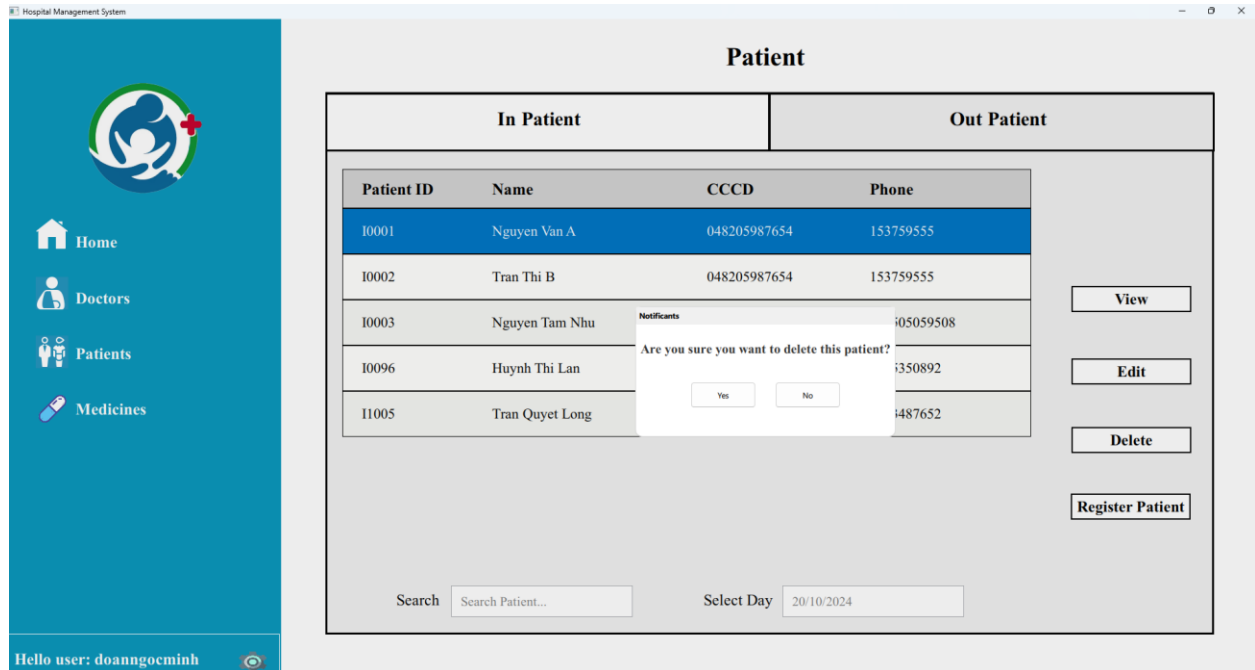
Room Number
B101

Discharge Date
14

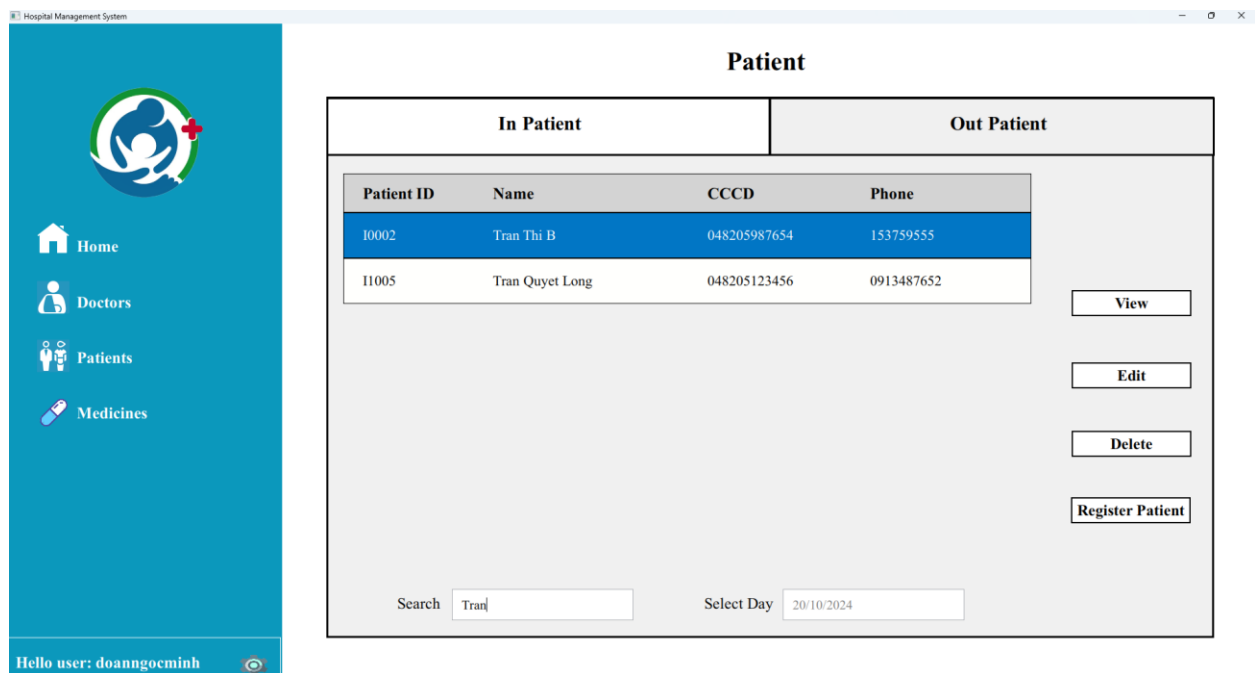
Cancel **Update**

Hello user: doannhocminh

Hình 4.5.2.12 Thực hiện chức năng chỉnh sửa bệnh nhân nội trú



Hình 4.5.2.13 Thực hiện chức năng xóa



Hình 4.5.2.14 Thực hiện chức năng tìm kiếm bệnh nhân theo tên

Patient

In Patient		Out Patient	
Patient ID	Name	CCCD	Phone
I0001	Nguyen Van A	048205987654	153759555
I0002	Tran Thi B	048205987654	153759555
I0096	Huynh Thi Lan	048205996222	0905350892
I1005	Tran Quyet Long	048205123456	0913487652

Search: Select Day:

Buttons: View, Edit, Delete, Register Patient

Hello user: doannhocminh

Hình 4.5.2.15 Thực hiện chức năng tìm kiếm bệnh nhân theo ngày

⇒ Tương tự vậy đối với bệnh nhân ngoại trú

Medicine

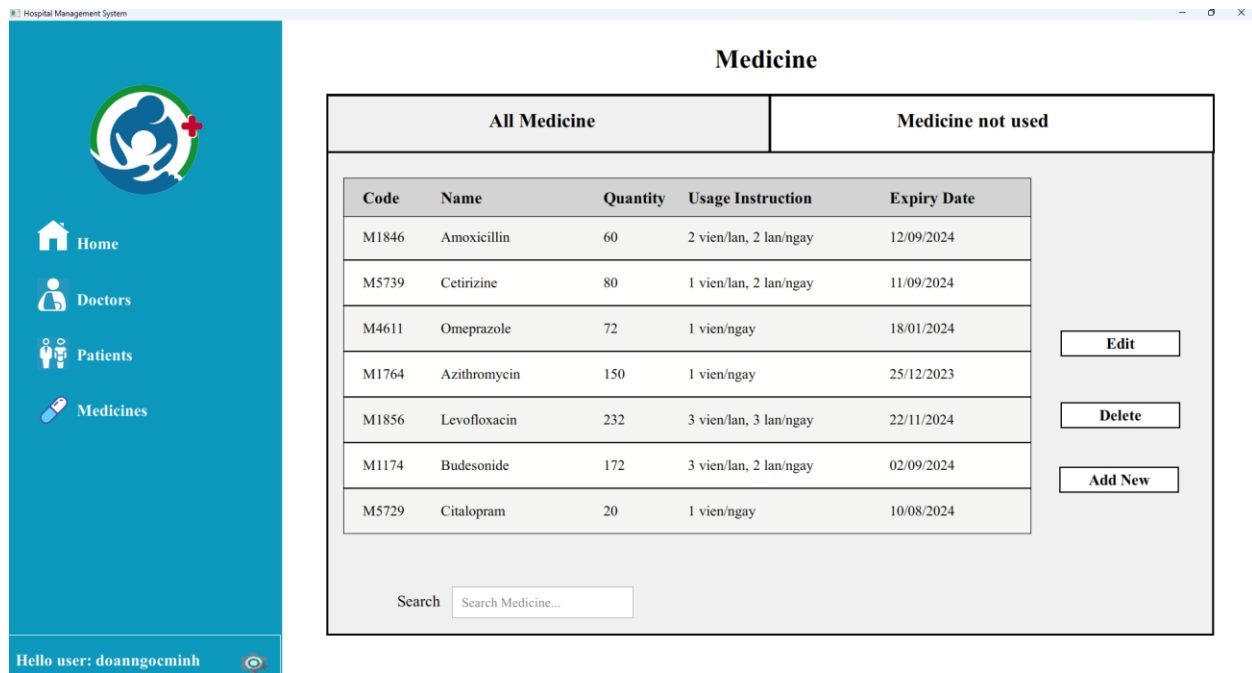
All Medicine		Medicine not used		
Code	Name	Quantity	Usage Instruction	Expiry Date
M4735	Paracetamol	57	1 viên/lần, 2-3 lần/ngày	30/06/2025
M4614	Ibuprofen	104	1 viên/lần, 3 lần/ngày	12/12/2024
M1846	Amoxicillin	60	2 viên/lần, 2 lần/ngày	12/09/2024
M1756	Loratadine	70	1 viên/ngày	22/01/2025
M5739	Cetirizine	80	1 viên/lần, 2 lần/ngày	11/09/2024
M3754	Metformin	40	3 viên/lần, 1 lần/ngày	15/01/2025
M5761	Amlodipine	120	1 viên/lần, 3 lần/ngày	22/09/2026
M3614	Simvastatin	55	2 viên/lần, 1 lần/ngày	15/08/2025

Search:

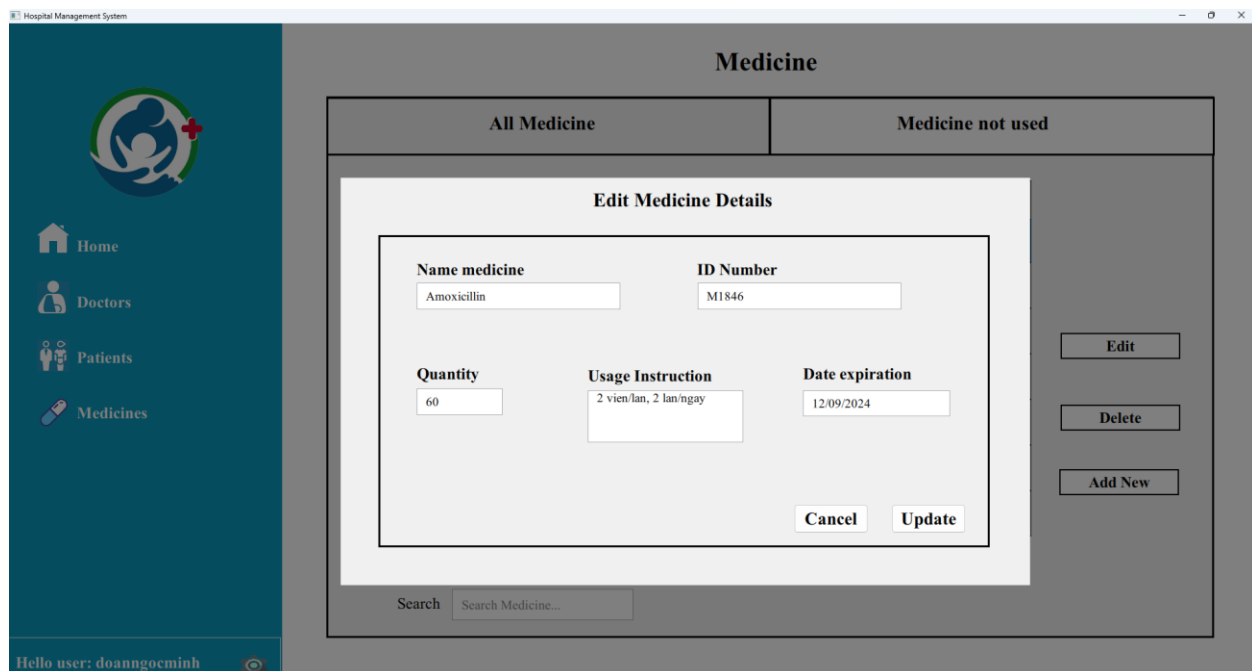
Buttons: Edit, Delete, Add New

Hello user: doannhocminh

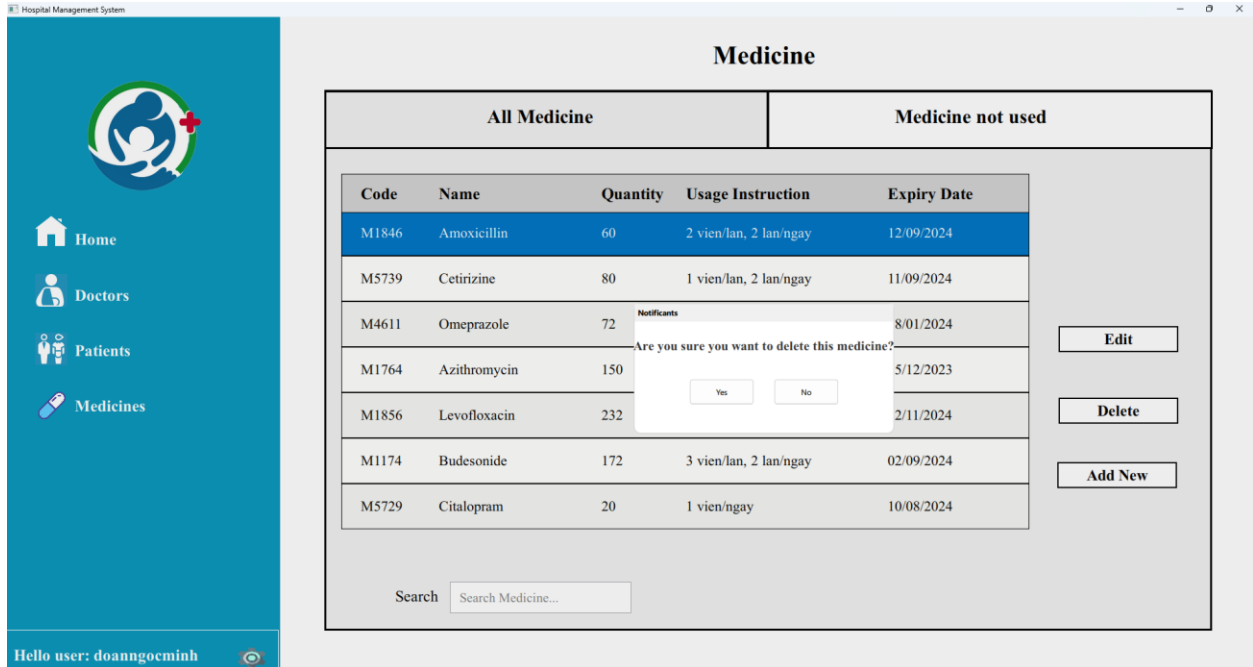
Hình 4.5.2.16 Giao diện medicine hiển thị thông tin của các loại thuốc



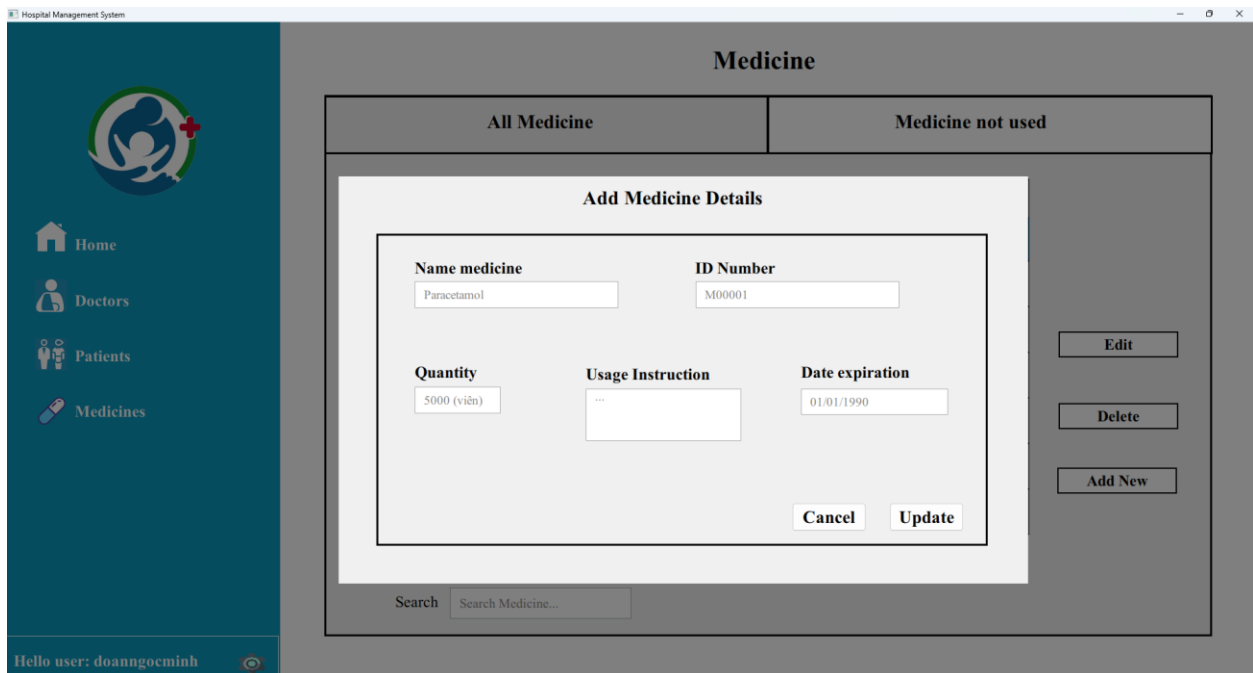
Hình 4.5.2.17 Giao diện hiển thị các loại thuốc đã hết hạn sử dụng



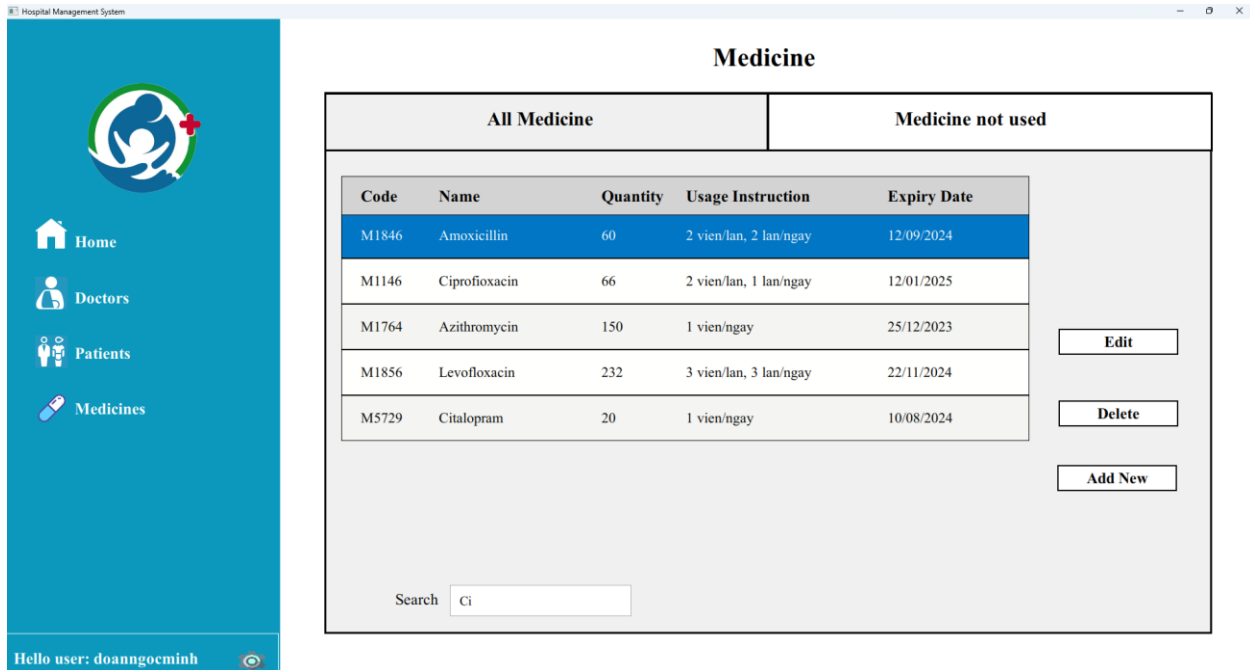
Hình 4.5.2.18 Thực hiện chức năng chỉnh sửa thuốc



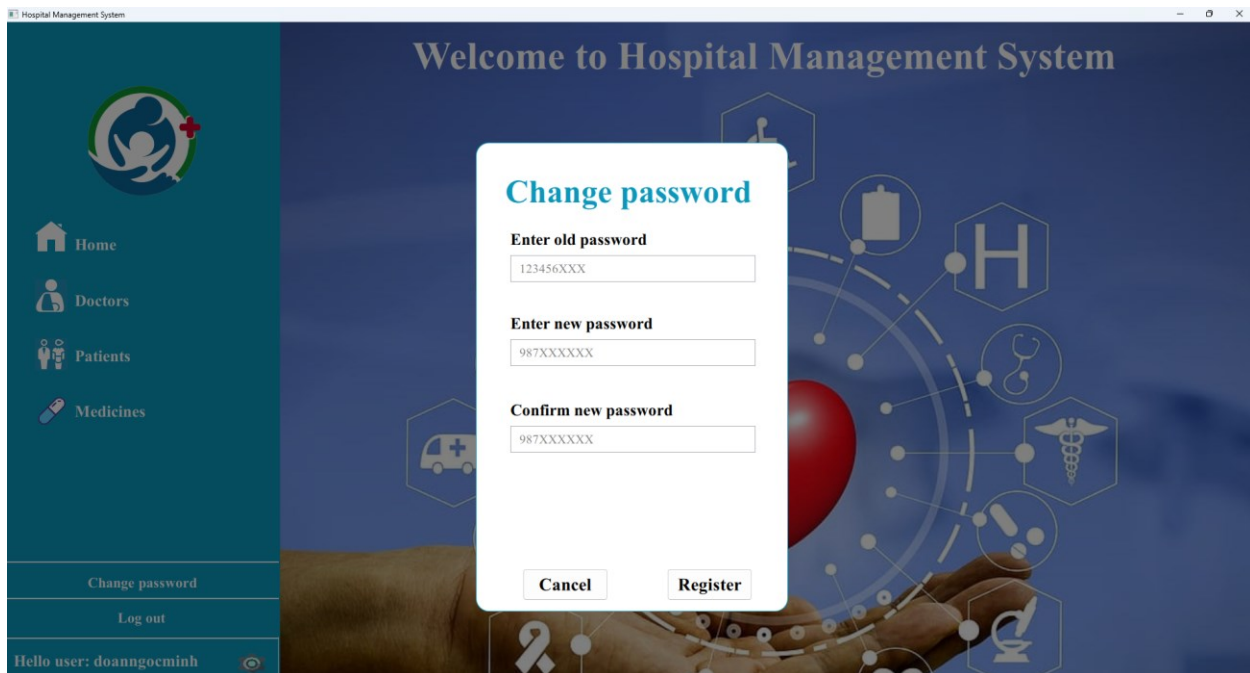
Hình 4.5.2.19 Thực hiện chức năng xóa loại thuốc



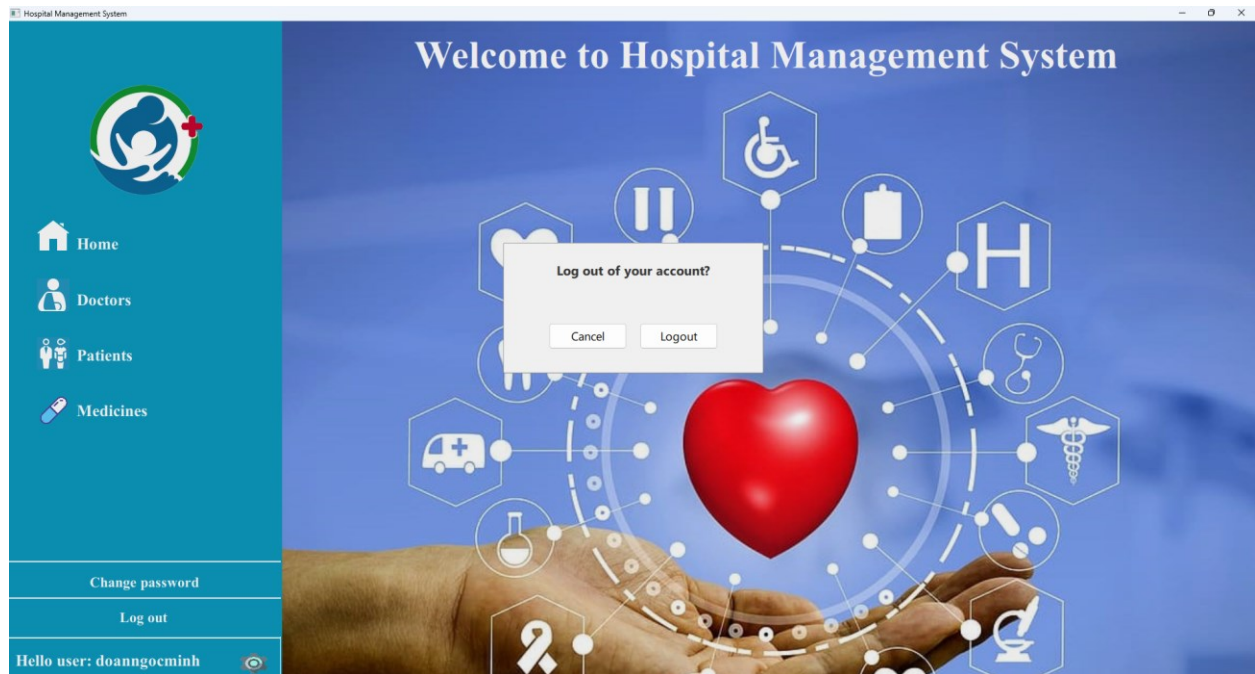
Hình 4.5.2.20 Thực hiện chức năng thêm 1 loại thuốc mới



Hình 4.5.2.21 Thực hiện chức năng tìm kiếm loại thuốc



Hình 4.5.2. 22 Thực hiện chức năng thay đổi mật khẩu



Hình 4.5.2.23 Thực hiện chức năng đăng xuất

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1. Đạt được

Hệ thống quản lý bệnh nhân và bác sĩ đã được thiết kế và triển khai với đầy đủ các chức năng cần thiết, bao gồm thêm mới, tìm kiếm, cập nhật, xóa và hiển thị thông tin. Những chức năng này giúp tối ưu hóa quy trình quản lý thông tin, giảm thiểu thời gian xử lý thủ công và tăng độ chính xác trong việc lưu trữ cũng như truy xuất dữ liệu y tế. Hệ thống không chỉ đáp ứng nhu cầu quản lý thông thường mà còn hỗ trợ tối đa cho các hoạt động vận hành của bệnh viện, từ việc theo dõi hồ sơ bệnh nhân đến phân công lịch làm việc cho bác sĩ.

Một điểm nổi bật của hệ thống là việc áp dụng mô hình lập trình hướng đối tượng. Các lớp như Person, Patient, Doctor, Inpatient, và Outpatient được tổ chức rõ ràng, tận dụng tốt nguyên tắc kế thừa và tính đa hình. Điều này giúp hệ thống dễ dàng mở rộng khi cần bổ sung các chức năng mới hoặc cải tiến các tính năng hiện tại, đồng thời giảm thiểu sự phụ thuộc giữa các thành phần trong hệ thống. Cấu trúc đối tượng không chỉ tăng tính tái sử dụng mà còn đơn giản hóa việc bảo trì, đảm bảo hệ thống hoạt động ổn định trong dài hạn.

Ngoài ra, hệ thống được trang bị các công cụ tìm kiếm nâng cao, cho phép người dùng tra cứu thông tin không chỉ về bệnh nhân mà còn cả bác sĩ, dựa trên nhiều tiêu chí như họ tên, mã bệnh nhân, ngày nhập viện, chẩn đoán, hoặc chuyên môn bác sĩ. Chức năng tìm kiếm này hỗ trợ nhân viên y tế và các bộ phận liên quan nhanh chóng tiếp cận dữ liệu cần thiết để ra quyết định chính xác, từ đó đảm bảo sự phối hợp nhịp nhàng giữa các bác sĩ và bệnh nhân trong quá trình điều trị.

Các tính năng thống kê trong hệ thống cũng không chỉ giúp bệnh viện quản lý số lượng bệnh nhân theo từng loại bệnh hay phòng điều trị, mà còn cung cấp thông tin quan trọng về bác sĩ, như số lượng ca khám, chuyên môn đang tham gia, và lịch làm việc. Điều này giúp bệnh viện có cái nhìn tổng quan về tình hình hoạt động của bác sĩ, từ đó tối ưu hóa việc phân công công việc và điều phối nguồn lực một cách hiệu quả hơn.

Hệ thống cũng chú trọng đến việc đảm bảo tính bảo mật và chuẩn hóa dữ liệu. Các thông tin nhập vào như số điện thoại, mã số BHYT, ngày tháng đều được kiểm tra chặt chẽ để giảm thiểu sai sót. Chức năng đăng nhập với phân quyền rõ ràng, kết hợp với cơ chế mã hóa mật khẩu, giúp bảo vệ thông tin nhạy cảm của bệnh nhân và bác sĩ.

Với cấu trúc vững chắc và khả năng mở rộng cao, hệ thống không chỉ là một công cụ hỗ trợ hiệu quả trong quản lý bệnh viện mà còn là nền tảng để phát triển các giải pháp công nghệ y tế hiện đại trong tương lai.

5.1.2. Chưa đạt được

Hiện tại, hệ thống chỉ sử dụng phương pháp lưu trữ đơn giản bằng tệp văn bản (.txt), chưa có các cơ chế bảo mật mạnh mẽ như mã hóa dữ liệu hoặc xác thực đăng nhập hai lớp. Điều này làm tăng nguy cơ rò rỉ thông tin nhạy cảm của bệnh nhân và bác sĩ nếu tệp bị xâm nhập hoặc sao chép.

Với việc lưu trữ dữ liệu trong tệp văn bản, khi danh sách bệnh nhân và bác sĩ tăng lên đáng kể, tốc độ tìm kiếm, sắp xếp, và truy xuất thông tin sẽ bị chậm lại. Điều này không phù hợp với nhu cầu thực tế của một bệnh viện có quy mô lớn.

Mặc dù đã có chức năng lưu lịch làm việc của bác sĩ, nhưng hệ thống chưa hỗ trợ lập lịch tự động hoặc kiểm tra xung đột lịch trình khi sắp xếp. Điều này có thể dẫn đến tình trạng chồng chéo hoặc bỏ sót lịch hẹn khám. Các chức năng thống kê hiện tại mới chỉ tập trung vào số lượng bệnh nhân theo khoa hoặc loại bệnh. Hệ thống chưa cung cấp các báo cáo chi tiết hơn như tỷ lệ tái khám, thời gian trung bình nằm viện, hoặc chi phí điều trị theo bệnh nhân.

Hệ thống chưa tích hợp tính năng gửi thông báo hoặc nhắc nhở cho bệnh nhân về lịch tái khám hoặc đơn thuốc sắp hết hạn. Đây là một thiếu sót quan trọng, đặc biệt trong môi trường chăm sóc sức khỏe hiện đại.

Hệ thống danh sách bệnh lý hiện tại chỉ mang tính minh họa và chưa bao quát hết các loại bệnh phổ biến trong thực tế. Điều này có thể gây khó khăn khi xử lý các trường hợp bệnh lý phức tạp hoặc mới phát sinh, dẫn đến việc thống kê và phân tích dữ liệu bệnh lý trở nên thiếu chính xác và toàn diện. Để khắc phục vấn đề này, cần triển khai một giải pháp mở rộng danh sách bệnh, cho phép cập nhật linh hoạt dựa trên nhu cầu thực tế. Việc tích hợp hệ thống với các nguồn dữ liệu y tế uy tín hoặc các cơ sở dữ liệu bệnh lý quốc tế cũng là một bước tiến quan trọng, giúp đảm bảo thông tin luôn được cập nhật và hỗ trợ các phân tích chuyên sâu, mang lại hiệu quả cao hơn trong việc quản lý và chăm sóc sức khỏe bệnh nhân.

5.2. Hướng phát triển

Hướng phát triển của **dự án quản lý bệnh viện** tập trung vào việc nâng cao hiệu quả hoạt động và cải thiện trải nghiệm của bệnh nhân. Một trong những định hướng chính là

cá nhân hóa dịch vụ y tế thông qua việc ghi nhớ thông tin lịch sử khám chữa bệnh và các nhu cầu đặc biệt của từng bệnh nhân. Điều này cho phép hệ thống hỗ trợ bác sĩ trong việc đưa ra phác đồ điều trị phù hợp, đồng thời cung cấp cho bệnh nhân các gợi ý về dịch vụ y tế bổ sung. Việc ứng dụng trí tuệ nhân tạo (AI) có thể tự động hóa việc phân loại bệnh nhân, xử lý các yêu cầu khẩn cấp, và đưa ra các cảnh báo thông minh về tình trạng sức khỏe.

Ngoài ra, hệ thống có thể tích hợp các giải pháp không chạm (contactless), như đăng ký khám bệnh trực tuyến, nhận thông tin phòng khám và bác sĩ qua mã QR, hoặc sử dụng công nghệ nhận diện khuôn mặt để xác thực danh tính. Những cải tiến này không chỉ giảm thời gian chờ đợi mà còn nâng cao sự tiện lợi, hiện đại hóa quy trình thăm khám.

Hệ thống cần phát triển một giao diện người dùng thân thiện và hiện đại để phục vụ bệnh nhân, bác sĩ và nhân viên y tế một cách hiệu quả. Giao diện trực quan sẽ giúp dễ dàng quản lý thông tin bệnh nhân, lên lịch khám, và theo dõi lịch làm việc của bác sĩ. Đồng thời, giao diện phải đảm bảo tương thích trên nhiều thiết bị như máy tính, máy tính bảng và điện thoại di động, hỗ trợ tối đa nhu cầu sử dụng trong các tình huống thực tế.

Lưu trữ dữ liệu trong hệ thống cần được tối ưu hóa bằng cách sử dụng các tệp văn bản hoặc tích hợp cơ sở dữ liệu như SQLite hoặc MySQL để quản lý thông tin bệnh nhân, bác sĩ, đơn thuốc, và lịch sử điều trị. Việc sử dụng cơ sở dữ liệu không chỉ cải thiện tốc độ truy xuất mà còn hỗ trợ quản lý số lượng lớn thông tin một cách chính xác và hiệu quả. Bên cạnh đó, cần có cơ chế sao lưu dữ liệu định kỳ để đảm bảo an toàn trong trường hợp xảy ra sự cố.

Quản lý phân quyền là một tính năng quan trọng, cho phép bác sĩ, nhân viên hành chính, và quản lý bệnh viện truy cập vào những thông tin phù hợp với vai trò của họ. Việc này đảm bảo tính bảo mật và kiểm soát tốt hơn các hoạt động trong hệ thống. Đồng thời, các cơ chế bảo mật như mã hóa dữ liệu, xác thực đăng nhập hai lớp, và ghi lại lịch sử truy cập cần được triển khai để ngăn chặn rò rỉ dữ liệu nhạy cảm.

Cuối cùng, việc tối ưu hóa thuật toán trong xử lý dữ liệu sẽ giúp hệ thống vận hành nhanh chóng, chính xác, và tiết kiệm tài nguyên. Các thuật toán thông minh có thể được phát triển để hỗ trợ phân loại bệnh, quản lý lịch khám, dự báo nhu cầu sử dụng phòng bệnh hoặc thuốc men, và thậm chí tối ưu hóa lịch làm việc của bác sĩ. Qua đó, hệ thống không chỉ cải thiện chất lượng dịch vụ y tế mà còn hỗ trợ bệnh viện trong việc tối ưu hóa nguồn lực và gia tăng hiệu quả hoạt động.

TÀI LIỆU THAM KHẢO

- [1] Thomas H. Cormen - Charles E. Leiserson, *Introduce to Algorithm*, năm xuất bản 2009, NXB Thống Kê.
- [2] PGS. TS. Hoàng Nghĩa Tý, *Cấu trúc dữ liệu và thuật toán*, năm xuất bản 2014, NXB Xây Dựng.
- [3] Jon Kleinberg, Éva Tardos, *Algorithm Design*, năm XB 2006, NXB Pearson/Addison-Wesley
- [4] GS. Phạm Văn Ất, *Giáo trình C++ & Lập trình hướng đối tượng*, NXB Bách Khoa Hà Nội

Phụ lục



Hình 5. Mã qr code PBL