# TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**



BÀI BÁO CÁO KẾT THÚC HỌC PHẦN HỌC KỲ I

NĂM HỌC 2023-2024

Học phần: CƠ SỞ DỮ LIỆU NÂNG CAO

**Cơ sở dữ liệu bệnh viện**

**GVHD**  TS. Phan Trọng Nhân

**Nhóm 7** Nguyễn Phương Vinh 3122411247

Trịnh Long Phát 3122411150

Lê Ngọc Hiệp 3122411053

Trần Hảo Điền 3122411042

Ngô Tuấn Anh 3122411007

Thành phố Hồ Chí Minh, Tháng 11/2023

# TRƯỜNG ĐẠI HỌC SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**



BÀI BÁO CÁO KẾT THÚC HỌC PHẦN HỌC KỲ I

NĂM HỌC 2023-2024

Học phần: CƠ SỞ DỮ LIỆU NÂNG CAO

**Cơ sở dữ liệu bệnh viện**

**GVHD**  TS. Phan Trọng Nhân

**Nhóm 7** Nguyễn Phương Vinh 3122411247

Trịnh Long Phát 3122411150

Lê Ngọc Hiệp 3122411053

Trần Hảo Điền 3122411042

Ngô Tuấn Anh 3122411007

Thành phố Hồ Chí Minh, Tháng 11/2023

Lời cam đoan

Tôi xin cam đoan đây là công trình báo cáo của cá nhân tôi và nhóm. Các nội dung nghiên cứu trong báo cáo “Cơ sở dữ liệu bệnh viện” của tôi là trung thực. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được cá nhân thu nhập từ các nguồn tài liệu khác nhau có ghi rõ nguồn gốc. Nếu có phát hiện bất kỳ sự gian lận nào thì tôi xin hoàn toàn chịu trách nhiệm về bài báo cáo của tôi.

Nhóm thực hiện

Nhóm 7

Lời cảm ơn

Lời cảm ơn chân thành đến các đọc giả đã dành thời gian để quan tâm đến bài báo của nhóm 7.

**BÁO CÁO HOẠT ĐỘNG CỦA NHÓM 7**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ và Tên | Nhiệm vụ được giao | Ghi chú |
| 3122411247 | Nguyễn Phương Vinh | - Chịu trách nhiệm lên nội dung cho (E)ERD.  - Thực hiện vẽ và sửa chữa (E)ERD.  - Thực hiện ánh xạ và chuyển đổi mô hình thực thể liên kết sang mô hình quan hệ.  - Cài đặt các chức năng phần mềm.  - Thực hiện chứng minh hiệu quả của chỉ mục. |  |
| 3122411042 | Trần Hảo Điền | - Thực hiện vẽ và sửa chữa (E)ERD.  - Thực hiện ánh xạ và chuyển đổi mô hình thực thể liên kết sang mô hình quan hệ.  - Cài đặt và thiết lập cơ sở dữ liệu.  - Thực hiện chứng minh hiệu quả của chỉ mục.  - Giải quyết vấn đề bảo mật trong cơ sở dữ liệu bệnh viện. |  |
| 3122411007 | Ngô Tuấn Anh | - Thực hiện ánh xạ và chuyển đổi mô hình thực thể liên kết sang mô hình quan hệ.  - Kiểm tra ràng buộc toàn vẹn trong cơ sở dữ liệu.  - Viết bài báo cáo nhóm.  - Giải quyết vấn đề bảo mật trong cơ sở dữ liệu bệnh viện. |  |
| 3122411150 | Trịnh Long Phát | - Thực hiện vẽ và sửa chữa (E)ERD.  - Thực hiện ánh xạ và chuyển đổi mô hình thực thể liên kết sang mô hình quan hệ.  - Giải quyết vấn đề bảo mật trong cơ sở dữ liệu bệnh viện. |  |
| 3122411053 | Lê Ngọc Hiệp | - Thực hiện vẽ và sửa chữa (E)ERD.  - Cài đặt các chức năng phần mềm.  - Thực hiện chứng minh hiệu quả của chỉ mục. |  |

**Mục lục**

[**LỜI MỞ ĐẦU** 1](#_Toc184236609)

[**CHƯƠNG 1. GIỚI THIỆU VỀ ĐỀ TÀI** 6](#_Toc184236610)

[**1.1. Yêu cầu đề tài** 6](#_Toc184236611)

[**1.2. Khái niệm về cơ sở dữ liệu** 10](#_Toc184236612)

[**1.3. Khảo sát giao diện thực tế của đề tài** 10](#_Toc184236613)

[**CHƯƠNG 2. THIẾT KẾ CƠ SỞ DỮ LIỆU** 13](#_Toc184236614)

[**2.1. Thiết kế sơ đồ (E)ER (Enhanced Entity Relationship)** 13](#_Toc184236615)

[**2.1.1. Khái niệm và một số thực thể trong sơ đồ** 13](#_Toc184236616)

[**2.1.2. Khái niệm và thuộc tính của thực thể** 13](#_Toc184236617)

[**2.1.3. Thiết kế mối liên kết giữa các thực thể** 18](#_Toc184236618)

[**2.2. Chuyển từ mô hình thực thể liên kết sang mô hình quan hệ dữ liệu** 27](#_Toc184236619)

[**2.2.1. Cách chuyển đổi** 27](#_Toc184236620)

[**2.2.2. Xác định các bảng và thuộc tính của bảng** 27](#_Toc184236621)

[**2.2.3. Mô hình dữ liệu quan hệ của bệnh viện** 32](#_Toc184236622)

[**2.2.4. Xác định phụ thuộc hàm** 33](#_Toc184236623)

[**2.3. Xác định RBTV (Ràng buộc toàn vẹn)** 36](#_Toc184236624)

[**2.3.1. Định nghĩa** 36](#_Toc184236625)

[**2.3.2. Xác định RBTV (Ràng buộc toàn vẹn)** 38](#_Toc184236626)

[**2.4. Mã nguồn cơ sở dữ liệu SQL (Structured Query Language)** 44](#_Toc184236627)

[**CHƯƠNG 3: XÂY DỰNG ĐỀ TÀI** 72](#_Toc184236628)

[**3.1. Môi trường lập trình** 72](#_Toc184236629)

[**3.2. Yêu cầu về chức năng** 72](#_Toc184236630)

[**3.2.1. Chức năng tìm kiếm bệnh nhân** 72](#_Toc184236631)

[**3.2.2. Chức năng thêm bệnh nhân** 82](#_Toc184236632)

[**3.2.3. Liệt kê tất cả các bệnh nhân được chữa bệnh bởi bác sĩ đó.** 92](#_Toc184236633)

[**3.2.4. Chức năng lập báo cáo về khoản thanh toán.** 96](#_Toc184236634)

[**3.2.5. Chứng minh việc sử dụng chỉ mục đạt hiệu quả.** 103](#_Toc184236635)

[**3.2.6. Giải quyết một trường hợp bảo mật trong cơ sở dữ liệu.** 106](#_Toc184236636)

[**Kết luận** 110](#_Toc184236637)

[**Tài liệu tham khảo** 111](#_Toc184236638)

**Danh sách các hình ảnh**

[Hình 1. 1. Phần mềm quản lý bệnh viện Viettel HIS [2] 11](#_Toc184216142)

[Hình 1. 2. Phần mềm quản lý bệnh viện TCSoft [2] 12](#_Toc184216143)

[Hình 2. 1. Thuộc tính Employee 14](#_Toc184216144)

[Hình 2. 2. Thuộc tính Patient 14](#_Toc184216145)

[Hình 2. 3. Thuộc tính Department 15](#_Toc184216146)

[Hình 2. 4. Thuộc tính Examination 15](#_Toc184216147)

[Hình 2. 5. Thuộc tính Medication 16](#_Toc184216148)

[Hình 2. 6. Thuộc tính Imported\_Medication 16](#_Toc184216149)

[Hình 2. 7. Thuộc tính Provider 17](#_Toc184216150)

[Hình 2. 8. Thuộc tính Treatment 17](#_Toc184216151)

[Hình 2. 9. Mô hình liên kết mở rộng của nhân sự 19](#_Toc184216152)

[Hình 2. 10. Mô hình liên kết mở rộng của bệnh nhân 20](#_Toc184216153)

[Hình 2. 11. Mô hình liên kết giữa phòng ban và nhân sự 21](#_Toc184216154)

[Hình 2. 12. Mô hình liên kết giữa phòng ban và bác sĩ 21](#_Toc184216155)

[Hình 2. 13. Mô hình liên kết giữa bác sĩ và khám bệnh 22](#_Toc184216156)

[Hình 2. 14. Mô hình liên kết giữa khám bệnh và bệnh nhân ngoại trú 22](#_Toc184216157)

[Hình 2. 15. Mô hình liên kết giữa khám bệnh và thuốc men 23](#_Toc184216158)

[Hình 2. 16. Mô hình liên kết giữa nhà cung cấp, nhập thuốc men và thuốc men 23](#_Toc184216159)

[Hình 2. 17. Mô hình liên kết giữa bác sĩ và điều trị 24](#_Toc184216160)

[Hình 2. 18. Mô hình liên kết giữa y tá và điều trị 24](#_Toc184216161)

[Hình 2. 19. Mô hình liên kết giữa điều trị và bệnh nhân nội trú 25](#_Toc184216162)

[Hình 2. 20. Mô hình liên kết giữa điều trị và thuốc men 25](#_Toc184216163)

[Hình 2. 21. Toàn bộ mối liên kết 26](#_Toc184216164)

[Hình 2. 22. Bảng Doctor 27](#_Toc184216165)

[Hình 2. 23. Bảng Nurse 28](#_Toc184216166)

[Hình 2. 24. Bảng Department 28](#_Toc184216167)

[Hình 2. 25. Bảng Paient 28](#_Toc184216168)

[Hình 2. 26. Bảng InPatient 28](#_Toc184216169)

[Hình 2. 27. Bảng OutPatient 29](#_Toc184216170)

[Hình 2. 28. Bảng Medication\_Effect 29](#_Toc184216171)

[Hình 2. 29. Bảng Medication 29](#_Toc184216172)

[Hình 2. 30. Bảng Provider 29](#_Toc184216173)

[Hình 2. 31. Bảng Treatment 30](#_Toc184216174)

[Hình 2. 32. Bảng Examination 30](#_Toc184216175)

[Hình 2. 33. Bảng Treat\_Use 30](#_Toc184216176)

[Hình 2. 34. Bảng Exam\_Use 30](#_Toc184216177)

[Hình 2. 35. Bảng Cure 30](#_Toc184216178)

[Hình 2. 36. Bảng Imported\_Medication 31](#_Toc184216179)

[Hình 2. 37. Bảng Details 31](#_Toc184216180)

[Hình 2. 38. Tổng quan mô hình dữ liệu quan hệ 32](#_Toc184216181)

[Hình 2. 39. Ràng buộc kiểu Default 37](#_Toc184216182)

[Hình 2. 40. Ràng buộc kiểu Not Null 37](#_Toc184216183)

[Hình 2. 41. Ràng buộc kiểu Primary Key (khóa chính) 38](#_Toc184216184)

[Hình 2. 42. Ràng buộc kiểu Foreign Key (khóa ngoại) 38](#_Toc184216185)

[Hình 2. 43. Ràng buộc kiểu Unique (độc quyền) 38](#_Toc184216186)

[Hình 3. 1. Giao diện tìm kiếm 55](#_Toc184216187)

[Hình 3. 2. Giao diện thêm bệnh nhân 65](#_Toc184216188)

[Hình 3. 3. Giao diện liệt kê các bác sĩ 66](#_Toc184216189)

[Hình 3. 4. Giao diện liệt kê các bệnh nhân của bác sĩ 66](#_Toc184216190)

[Hình 3. 5. Giao diện báo cáo các khoản thu phí của bệnh nhân 76](#_Toc184216191)

[Hình 3. 6. Tìm kiếm không có chỉ mục 78](#_Toc184216192)

[Hình 3. 7. Tìm kiếm có chỉ mục 78](#_Toc184216193)

[Hình 3. 8. Sơ đồ bảo mật theo DAC (Discretionary Access Control) [3] 79](#_Toc184216194)

[Hình 3. 9. Quyền truy cập của doctor 81](#_Toc184216195)

[Hình 3. 10. Quyền truy cập của Dean 82](#_Toc184216196)

**Danh sách các bảng biểu**

[Bảng 2. 1. Bảng tầm ảnh hưởng R1 39](#_Toc184216197)

[Bảng 2. 2. Bảng tầm ảnh hưởng R2 39](#_Toc184216198)

[Bảng 2. 3. Bảng tầm ảnh hưởng R3 40](#_Toc184216199)

[Bảng 2. 4. Bảng ầm ảnh hưởng R4 40](#_Toc184216200)

[Bảng 2. 5. Bảng tầm ảnh hưởng R5 40](#_Toc184216201)

[Bảng 2. 6. Bảng tầm ảnh hưởng R6 41](#_Toc184216202)

[Bảng 2. 7. Bảng tầm ảnh hưởng R7 41](#_Toc184216203)

[Bảng 2. 8. Bảng tầm ảnh hưởng R8 42](#_Toc184216204)

[Bảng 2. 9. Bảng tầm ảnh hưởng R9 42](#_Toc184216205)

[Bảng 2. 10. Bảng tầm ảnh hưởng R10 43](#_Toc184216206)

[Bảng 2. 11. Bảng tầm ảnh hưởng R11 43](#_Toc184216207)

[Bảng 2. 12. Bảng tầm ảnh hưởng R12 43](#_Toc184216208)

[Bảng 2. 13. Bảng tầm ảnh hưởng R13 44](#_Toc184216209)

[Bảng 2. 14. Bảng tầm ảnh hưởng R14 44](#_Toc184216210)

**Danh sách từ viết tắt**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Ký tự chữ viết tắt** | **Chữ viết đầy đủ** |
| 1. | (E)ERD | Enhanced Entity Relationship Diagram |
| 2. | (E)ER | Enhanced Entity Relationship |
| 2 | RBTV | Ràng buộc toàn vẹn |
| 3. | CSDL | Cơ sở dữ liệu |

# **LỜI MỞ ĐẦU**

**1. Lý do chọn đề tài**

Trong bối cảnh hiện nay, dữ liệu y tế đóng vai trò thiết yếu trong việc cải thiện chất lượng dịch vụ chăm sóc sức khỏe ở các nước trên thế giới nói chung và ở Việt Nam nói riêng. Đặc biệt, khi nước ta đang dần tiến một kỷ nguyên mới dành cho công nghệ hiện đại, việc giải quyết các bài toán về dữ liệu y tế trở nên dễ dàng hơn nhờ các trợ thủ đắc lực tiên tiến như máy tính, thiết bị khai báo y tế,…

Tuy nhiên, nhận thấy vấn đề rằng, ở Việt Nam, việc quản lý dữ liệu y tế vẫn còn gặp khá nhiều khó khăn trong việc xây dựng và liên kết các dữ liệu với nhau.

**2. Lịch sử nghiên cứu vấn đề và tổng quan**

Những năm 1970 chứng kiến sự ra đời của các hệ thống máy tính đầu tiên trong y tế. Hệ thống quản lý bệnh viện (Hospital Information System – HIS) và cở sở dữ liệu y tế bắt đầu xuất hiện.

Các cơ sở dữ liệu như Oracle, IBM DB2 cũng ra đời và được sử dụng để quản lý thông tin bệnh viện một cách hiệu quả hơn.

Tổng quan của một mô hình quản lý, lưu trữ các thông tin cần thiết đến bệnh nhân, dịch vụ y tế, nhân lực và nghiệp vụ của cơ sở y tế, bao gồm:

* Hồ sơ bệnh nhân (thông tin cá nhân, lịch sử khám chữa bệnh).
* Quản lý thuốc và vật tư y tế.
* Dữ liệu về thuốc men.
* Thông tin nhân sự (bác sĩ, y tá).

**3. Mục đích nghiên cứu**

- Xây dựng hệ thống quản lý thông tin y tế hiệu quả

Tạo ra một cơ sở dữ liệu có cấu trúc rõ ràng và dễ sử dụng để quản lý thông tin bệnh viện, bao gồm hồ sơ bệnh nhân, lịch sử khám chữa bệnh, thông tin nhân sự, và quản lý thuốc men.

- Tăng cường hiệu quả hoạt động của bệnh viện

Hỗ trợ tự động hóa các quy trình như đăng ký khám bệnh, lưu trữ thông tin, và truy xuất dữ liệu, từ đó giảm thời gian và công sức của nhân viên y tế.

- Nâng cao chất lượng dịch vụ chăm sóc sức khỏe

Đảm bảo thông tin bệnh nhân và lịch sử điều trị được quản lý chính xác, giúp bác sĩ đưa ra quyết định điều trị nhanh chóng và đúng đắn.

- Cải thiện tính minh bạch và bảo mật thông tin

Đề xuất giải pháp để đảm bảo dữ liệu y tế được lưu trữ an toàn, tuân thủ các quy định bảo mật thông tin cá nhân, đồng thời có khả năng truy xuất dễ dàng khi cần.

- Phân tích và hỗ trợ ra quyết định

Phát triển các báo cáo và biểu đồ dựa trên dữ liệu, hỗ trợ ban lãnh đạo bệnh viện phân tích hoạt động, đánh giá hiệu quả và đưa ra các quyết định quản lý.

- Ứng dụng công nghệ hiện đại vào quản lý y tế

Tích hợp cơ sở dữ liệu với các công nghệ như hệ thống phần mềm quản lý bệnh viện (HIS), trí tuệ nhân tạo (AI), và xử lý dữ liệu lớn (Big Data) để nâng cao hiệu suất hoạt động.

- Tạo tiền đề cho các nghiên cứu hoặc dự án phát triển sau này

Đề tài đặt nền móng cho việc triển khai các hệ thống thông minh, như hệ thống gợi ý phác đồ điều trị hoặc dự báo tình hình bệnh dịch dựa trên dữ liệu.

- Đáp ứng nhu cầu thực tiễn

Đáp ứng nhu cầu cấp thiết của các cơ sở y tế trong việc số hóa thông tin và quản lý dữ liệu trong thời đại công nghệ 4.0.

**4. Đối tượng nghiên cứu và phạm vi nghiên cứu**

Đối tượng nghiên cứu:

1) Dữ liệu y tế

* Thông tin bệnh nhân.
* Thông tin thuốc men y tế.
* Thông tin nhân sự.
* Thông tin nhà cung cấp.
* Thông tin các buổi khám và điều trị tại bệnh viện.

2) Hệ thống quản lý bệnh viện

* Đăng ký khám bệnh, chưa bệnh.
* Quản lý kê thuốc.
* Quản lý giữa các phòng khoa.

Phạm vi nghiên cứu:

- Phạm vi về nội dung

* Xây dựng cơ sở dữ liệu
* Xây dựng giao diện tương tác tiện dụng cho người dùng
* Quản lý và khai thác dữ liệu
* Bảo mật dữ liệu

- Phạm vi không gian

Hệ thống cơ sở dữ liệu được thiết kế và thử nghiệm trong phạm vi giả định của một bệnh viện cụ thể, không áp dụng ngay cho toàn ngành y tế.

**5. Phương pháp nghiên cứu**

Các phương pháp nghiên cứu nhóm sẽ dùng cho việc nghiên cứu đề tài lần này là:

+ Tra cứu tư liệu.

+ Tham khảo thông tin trên các trang web và diễn đàn khoa học.

**6. Tính mới của đề tài**

1. Quản lý thông tin nhân viên:

- Mỗi nhân viên (bác sĩ hoặc y tá) phải thuộc một phòng ban (department) cụ thể.

- Yêu cầu thông tin của trưởng phòng (dean):

+ Phải là bác sĩ, có chuyên môn (speciality) cụ thể.

+ Có ít nhất 5 năm kinh nghiệm từ khi nhận bằng chuyên môn.

- Cần lưu trữ thông tin chuyên môn của nhân sự bao gồm: tên chuyên môn và năm cấp bằng.

2. Quản lý bệnh nhân:

- Phân loại bệnh nhân: Dựa vào mã bệnh nhân (unique code) để phân biệt:

+ Ngoại trú (Outpatient - "OP").

+ Nội trú (Inpatient - "IP").

Mã bệnh nhân có cấu trúc: gồm tiền tố “OP” hoặc “IP” và 9 chữ số liên tiếp, tự động sinh mã khi lưu trữ thông tin.

# **CHƯƠNG 1. GIỚI THIỆU VỀ ĐỀ TÀI**

## **1.1. Yêu cầu đề tài**

- Bản gốc

A hospital X needs to build a management information system to manage the information of their patients, doctors, and nurses.

The database of hospital X needs to store the information of employees (doctors and nurses) including: a unique code, full name consisting of first name and last name, date of birth, gender, address, start date (first day of work), phone number(s), and speciality with its related name and degree’s year. The hospital has many departments. Each department has a unique code, a title, and a dean who is a doctor. The employees have to belong to a specific department. A department has at least one or many employees. The dean must hold a specific speciality and has had more than 5 years of experience since the date he or she was awarded the speciality degree. The patients have to provide with the hospital their information such as: full name (first name and last name), date of birth, gender, address, and phone number. After receiving their information, the system will store them into the database, and generate a unique code to identify each patient simultaneously. Patients are divided into two types: outpatients and inpatients. The hospital also wishes to use the first two characters to determine the patient type by the unique code. If one is an outpatient, the unique code for him or her starts with “OP,” which is then followed by 9 digits such as “OP000000001.” If one is an inpatient, the unique code for him or her starts with “IP,” which is then followed by 9 digits such as “IP000000001.”

1. For outpatients, the information of the examining doctor needs to be stored. The outpatients can have many examinations with their examining doctor. The hospital needs to store the details of each examination such as: examination date, diagnosis, the next examination date if any, medications, and fee.

2. For inpatients, some information is added such as: date of admission, treating doctors, caring nurse, diagnosis, sickroom, date of discharge, and fee. After admitting to the hospital, a patient can receive treatment from at least one doctor. A doctor can treat many patients at the same time, or sometimes, he has no patients to treat. The hospital needs the details of each treatment such as: treatment period (start date and end date), result, and medications. Each inpatient is taken care of by a nurse; a nurse can take care of many inpatients at the same time. Furthermore, when a patient is recovered and his or her last treatment has been confirmed as “recovered” by the doctor, he or she will be discharged from the hospital. As a result, the discharge date must be recorded by the system.

The information of a medication is also stored in the database. This information consists of a unique code, name of the medication, effects, price, and expirationdate. A medication is provided by one or more providers, and one provider may provide many types of medication. A provider is tracked by its unique number, name, address, and phone. Moreover, the hospital also want to keep the imported medication information including imported date, price, and quantity. In case one medication is out-of-date, it will be automatically marked so in the database.

- Bản dịch thuật (Tiếng Việt)

Bệnh viện X cần xây dựng hệ thống thông tin quản lý để quản lý thông tin của bệnh nhân, bác sĩ và y tá của họ.

Cơ sở dữ liệu của bệnh viện X cần lưu trữ thông tin của nhân viên (bác sĩ và điều dưỡng) gồm: mã số duy nhất, họ và tên đầy đủ gồm họ và tên, ngày sinh, giới tính, địa chỉ, ngày bắt đầu (ngày đầu tiên làm việc), số điện thoại và chuyên ngành có tên liên quan và năm cấp bằng. Bệnh viện có rất nhiều các phòng ban. Mỗi khoa có một mã số duy nhất, một chức danh và một trưởng khoa là bác sĩ. Các nhân viên phải thuộc về một bộ phận cụ thể. Một bộ phận có ít nhất một hoặc nhiều nhân viên. Trưởng khoa phải có chuyên môn cụ thể và có nhiều kinh nghiệm hơn trên 5 năm kinh nghiệm kể từ ngày được cấp chuyên môn bằng cấp.

Người bệnh phải cung cấp cho bệnh viện các thông tin như: họ tên (họ và tên), ngày sinh, giới tính, địa chỉ và số điện thoại. Sau đó nhận được thông tin của họ, hệ thống sẽ lưu trữ chúng vào cơ sở dữ liệu và tạo ra một mã duy nhất để xác định đồng thời từng bệnh nhân. Bệnh nhân được chia thành hai loại: bệnh nhân ngoại trú và bệnh nhân nội trú. Bệnh viện cũng mong muốn được sử dụng lần đầu hai ký tự để xác định loại bệnh nhân bằng mã duy nhất. Nếu một người là một bệnh nhân ngoại trú, mã duy nhất dành cho họ bắt đầu bằng “OP”, sau đó theo sau bằng 9 chữ số chẳng hạn như “OP000000001.” Nếu một người là bệnh nhân nội trú, mã duy nhất dành cho người đó hoặc cô ấy bắt đầu bằng “IP”, sau đó là 9 chữ số, chẳng hạn như “IP000000001”.

1. Đối với bệnh nhân ngoại trú, thông tin của bác sĩ khám bệnh cần được lưu trữ. Các bệnh nhân ngoại trú có thể được khám nhiều lần với bác sĩ khám bệnh của họ. Các bệnh viện cần lưu trữ thông tin chi tiết của từng lần khám như: ngày khám, chẩn đoán, ngày khám tiếp theo nếu có, thuốc và lệ phí.

2. Đối với bệnh nhân nội trú, bổ sung thêm một số thông tin như: ngày nhập viện, điều trị bác sĩ, y tá chăm sóc, chẩn đoán, phòng bệnh, ngày xuất viện và phí. Sau đó khi nhập viện, bệnh nhân có thể được ít nhất một bác sĩ điều trị. Một bác sĩ có thể chữa trị cho nhiều bệnh nhân cùng một lúc, hoặc đôi khi ông ta không có bệnh nhân để điều trị. Bệnh viện cần thông tin chi tiết của từng phương pháp điều trị như: thời gian điều trị (ngày bắt đầu và ngày kết thúc), kết quả và thuốc. Mỗi bệnh nhân nội trú được y tá chăm sóc; một y tá có thể chăm sóc nhiều bệnh nhân nội trú cùng một lúc thời gian. Hơn nữa, khi một bệnh nhân được hồi phục và lần điều trị cuối cùng của họ đã kết thúc. Được bác sĩ xác nhận là đã “hồi phục” thì sẽ được xuất viện bệnh viện. Vì vậy, ngày xả thải phải được hệ thống ghi lại.

Thông tin về thuốc cũng được lưu trữ trong cơ sở dữ liệu. Thông tin này bao gồm một mã duy nhất, tên thuốc, tác dụng, giá cả và hạn sử dụng ngày. Một loại thuốc được cung cấp bởi một hoặc nhiều nhà cung cấp và một nhà cung cấp có thể cung cấp nhiều loại thuốc. Một nhà cung cấp được theo dõi bởi số duy nhất của nó, tên, địa chỉ và số điện thoại. Hơn nữa, bệnh viện cũng muốn giữ nguyên sản phẩm nhập khẩu thông tin thuốc bao gồm ngày nhập khẩu, giá cả và số lượng. Trong trường hợp một thuốc đã hết hạn sử dụng sẽ tự động được đánh dấu vào cơ sở dữ liệu.

## **1.2. Khái niệm về cơ sở dữ liệu**

Trước khi các hệ CSDL ra đời (khoảng đầu những năm 60 của thế kỷ 20), mỗi chương trình ứng dụng đều có một tệp dữ liệu tương ứng và mỗi khi chương trình ứng dụng cần được sửa đổi hoặc mở rộng thì tệp dữ liệu tương ứng cũng phải thay đổi theo. Việc lưu trữ thông tin của một tổ chức trong một hệ xử lý tệp như vậy có những khó khăn như sau:

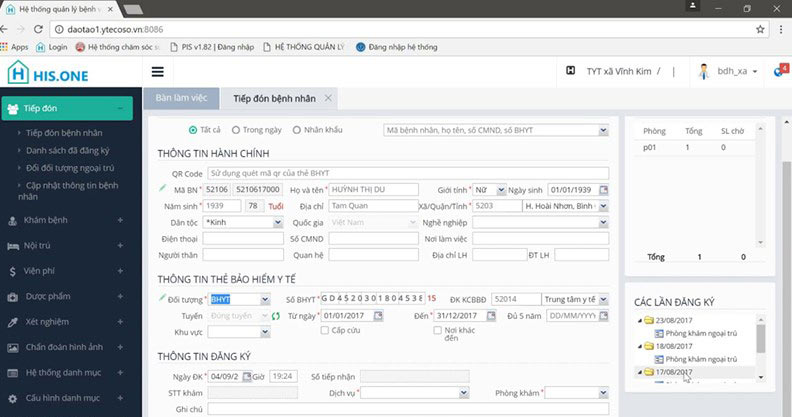
* Dư thừa dữ liệu và không nhất quán.
* Khó khăn trong việc truy cập dữ liệu.
* Các vấn đề toàn vẹn.
* Các vấn đề an toàn.

Một cơ sở dữ liệu là một tập hợp các dữ liệu có liên quan với nhau được lưu trữ trên thiết bị nhớ thứ cấp (như băng từ, đĩa từ,…) để đáp ứng nhu cầu khai thác thông tin của nhiều người sử dụng với nhiều mục đích khác nhau. [1]

Song song, có thể thấy cơ sở dữ liệu bệnh viện là một tập dữ liệu được lưu trữ và quản lý, liên quan đến các hoạt động và đối tượng trong bệnh viện, bao gồm: bệnh nhân, nhân viên y tế (bác sĩ, y tá), thuốc, phòng ban, và các dịch vụ y tế.

## **1.3. Khảo sát giao diện thực tế của đề tài**

Phần mềm Viettel HIS



Hình 1. 1. Phần mềm quản lý bệnh viện Viettel HIS [2]

Giao diện điền thông tin của bệnh nhân

Thông tin lưu trữ:

- Thông tin hành chính

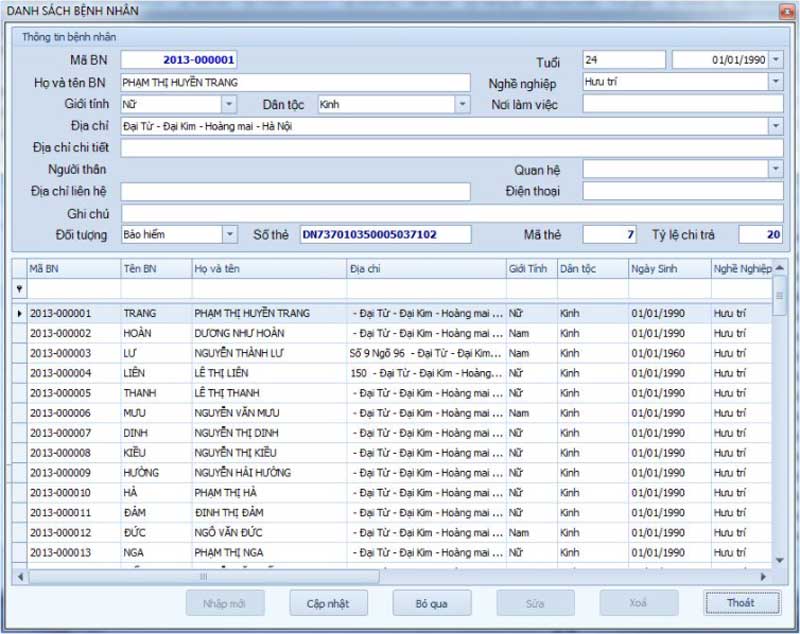
- Thông tin về thẻ bảo hiểm y tế

- Các lần đăng ký khám bệnh

Chức năng:

- Tạo lượt đăng ký khám bệnh của bệnh nhân.

Phần mềm TCSoft



Hình 1. 2. Phần mềm quản lý bệnh viện TCSoft [2]

Giao diện điền thông tin và lưu trữ các bệnh nhân trong cơ sỡ dữ liệu.

# **CHƯƠNG 2. THIẾT KẾ CƠ SỞ DỮ LIỆU**

## **2.1. Thiết kế sơ đồ (E)ER (Enhanced Entity Relationship)**

### **2.1.1. Khái niệm và một số thực thể trong sơ đồ**

- Thực thể là biểu diễn một đối tượng, khái niệm hay sự vật xác

định cụ thể của thế giới thực.

- Thể hiện thực thể là một đối tượng cụ thể của thực thể.

- Ký hiệu: Hình chữ nhật dùng để biểu diễn thực thể.

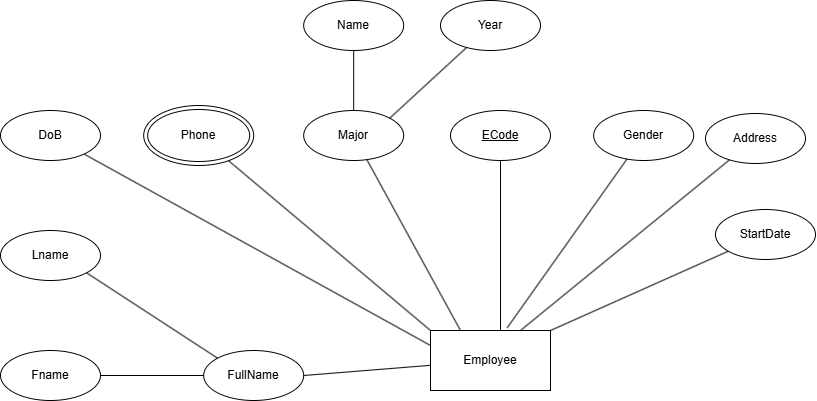
Trong bài báo cáo, các thực thể sẽ bao gồm: Employee, Patient, Department, Examination, Medication, Imported\_Medication, Provider, Treatment.

### **2.1.2. Khái niệm và thuộc tính của thực thể**

- Thuộc tính là những tính chất đặc trưng của thực thể mà giá trị của nó là dữ liệu cần lưu trữ.

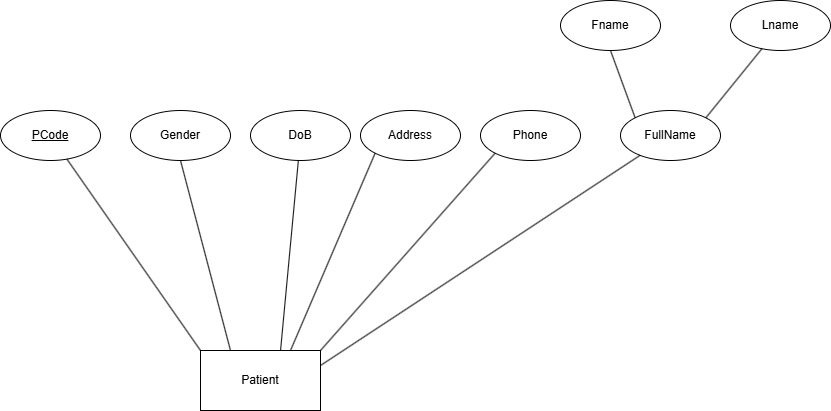
- Ký hiêu: Hình elip dùng để biểu diễn thuộc tính của thực thể.

- Thuộc tính của Employee gồm: Ecode, Gender, Address, StartDate, Major(Name, Year), Phone, DoB, FullName(Lname, Fname).



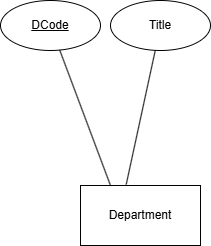
Hình 2. 1. Thuộc tính Employee

- Thuộc tính của Patient gồm: Pcode, Gender, DoB, Address, Phone, Fullname(Fname, Lname).



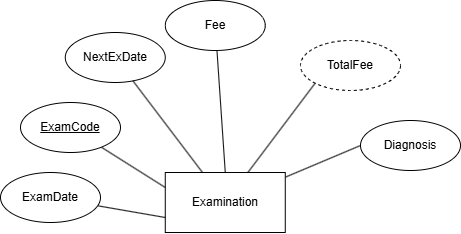
Hình 2. 2. Thuộc tính Patient

- Thuộc tính của Department gồm: Dcode, Tile.



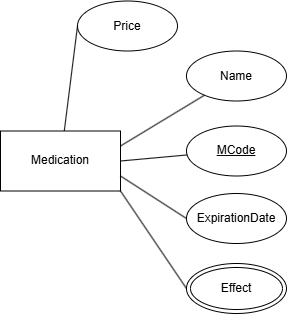
Hình 2. 3. Thuộc tính Department

- Thuộc tính của Examination bao gồm: Examcode, Diagnosis, ExamDate, Fee, NextExDate, TotalFee (Thuộc tính dẫn xuất).



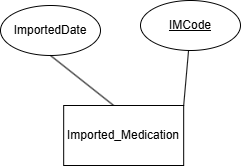
Hình 2. 4. Thuộc tính Examination

- Thuộc tính của Medication bao gồm: MCode, Name, ExpirationDate, Effect, Price.



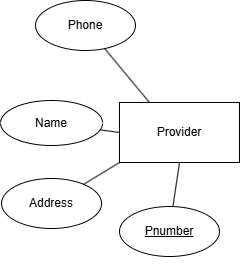
Hình 2. 5. Thuộc tính Medication

- Thuộc tính của Imported\_Medication bao gồm: ImportedDate, IMCode.



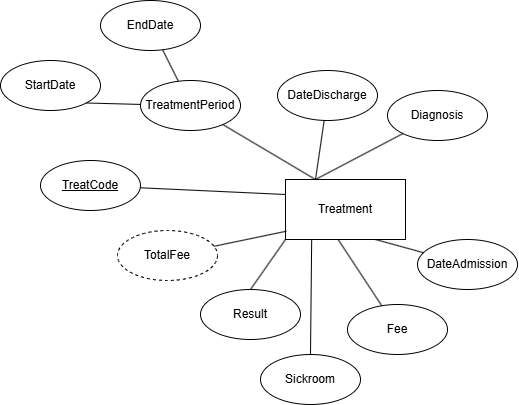
Hình 2. 6. Thuộc tính Imported\_Medication

- Thuộc tính của Provider bao gồm: Phone, Name, Address, Pnumber.



Hình 2. 7. Thuộc tính Provider

- Thuộc tính của Treatment bao gồm: TreatCode, Result, Sickroom, Fee, DateAdmission, TreatmentPeriod(StartDate, EndDate), DateDischarge, Diagnosis.



Hình 2. 8. Thuộc tính Treatment

### **2.1.3. Thiết kế mối liên kết giữa các thực thể**

Các thành phần liên kết giữa những thực thể:

- Hình thoi: Biểu diễn liên kết giữa các thực thể.

- Đường kẻ: Nối thực thể với các thuộc tính của nó hoặc nối thực thể với liên kết.

- Cardinality (Số lượng liên kết):

+ 1:1 (một - một): Một thực thể bên phía này có thể liên kết với một hoặc một thực thể bên phía kia.

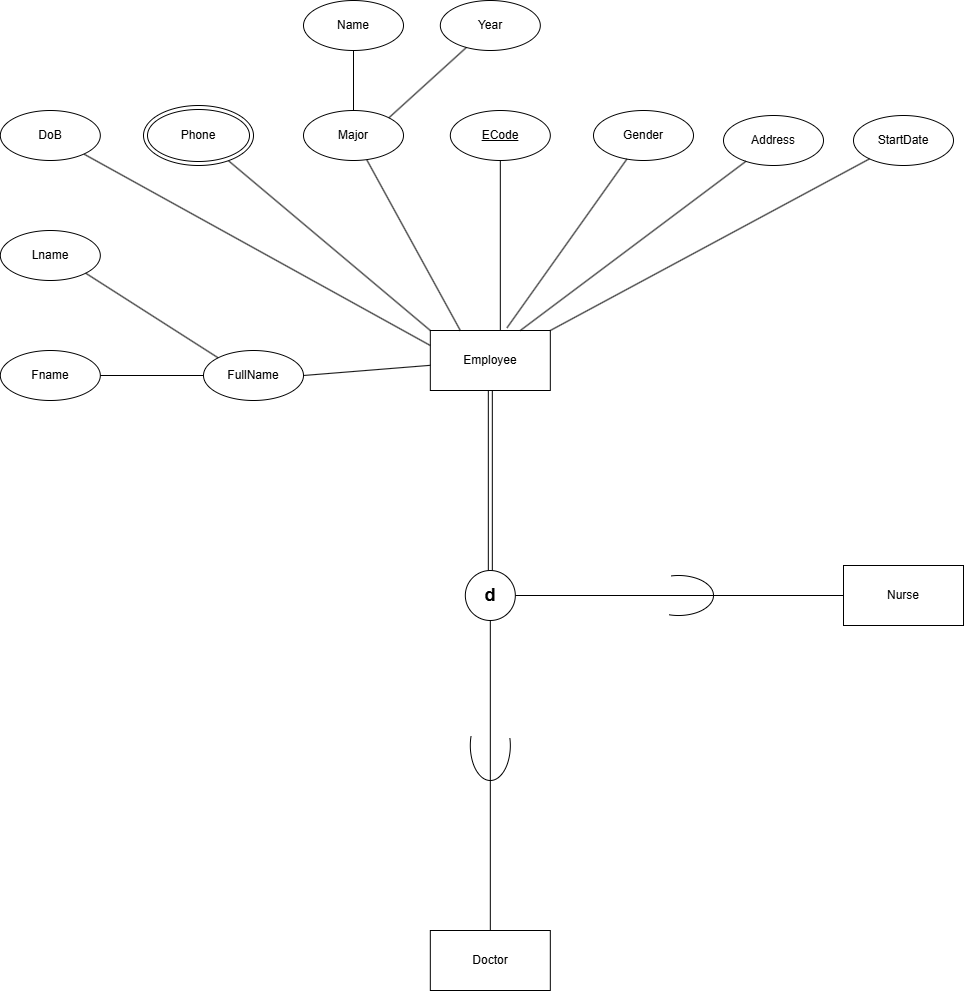
+ 1:N (một - nhiều): Một thực thể bên phía này có thể liên kết với một hoặc nhiều thực thể bên phía kia.

+ 0:N (không - nhiều): Một thực thể bên phía này có thể liên kết với không hoặc nhiều thực thể bên phía kia.

+ 0:1 (không - một): Một thực thể bên phía này có thể liên kết với không hoặc một thực thể bên phía kia.

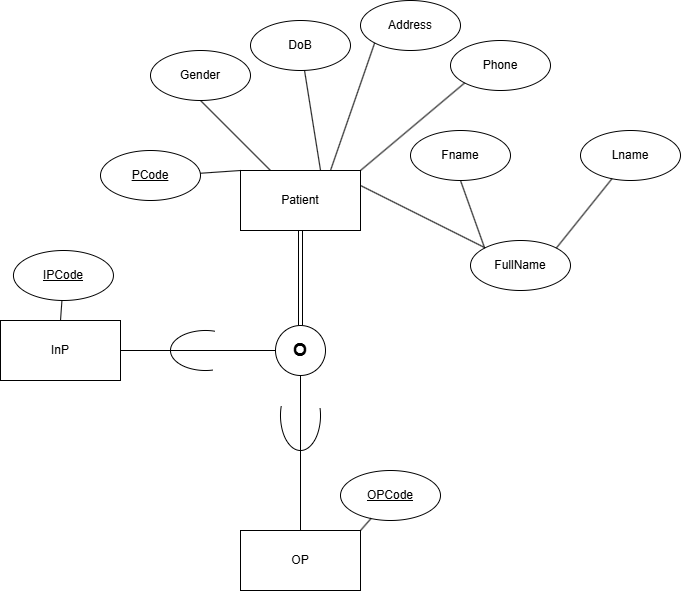
Xác định các mối liên kết mở rộng trong sơ đồ (E)ER:

- Mô hình liên kết mở rộng của nhân sự:



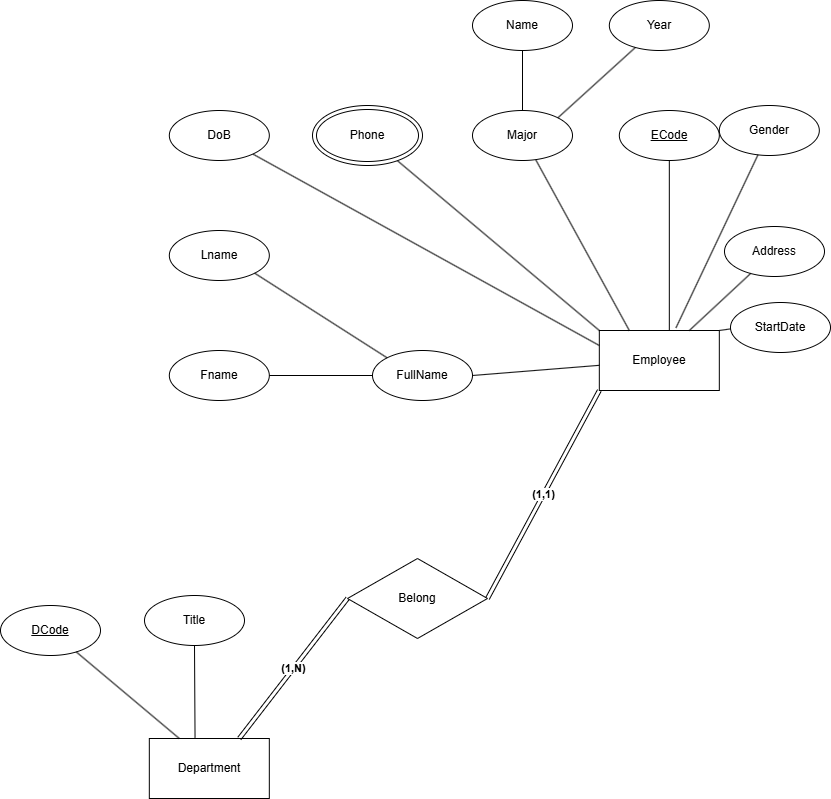
Hình 2. 9. Mô hình liên kết mở rộng của nhân sự

- Mô hình liên kết mở rộng của bệnh nhân



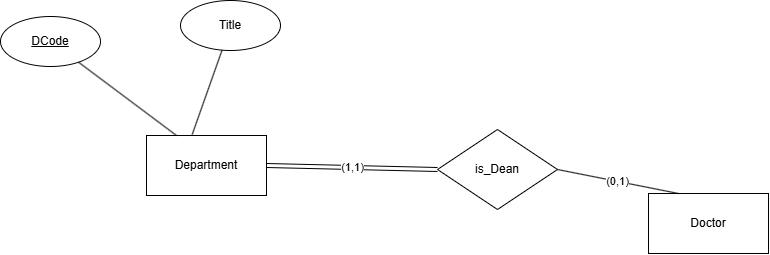
Hình 2. 10. Mô hình liên kết mở rộng của bệnh nhân

- Mô hình liên kết giữa phòng ban và nhân sự



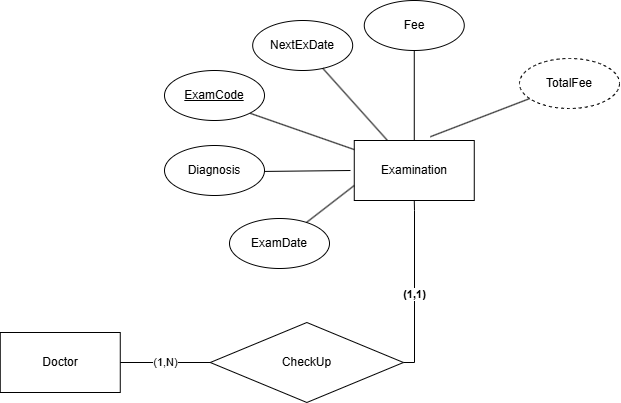
Hình 2. 11. Mô hình liên kết giữa phòng ban và nhân sự

- Mô hình liên kết giữa phòng ban và bác sĩ



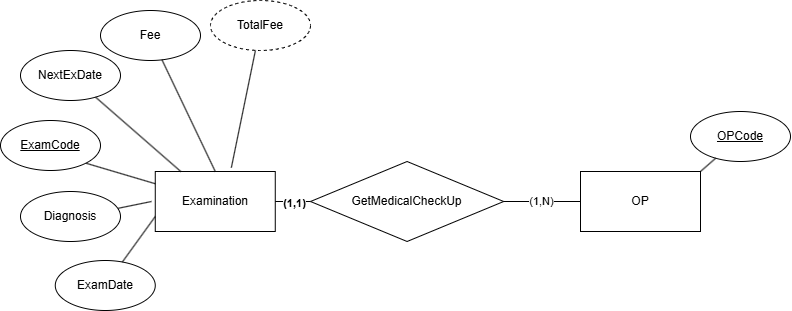
Hình 2. 12. Mô hình liên kết giữa phòng ban và bác sĩ

- Mô hình liên kết giữa bác sĩ và khám bệnh



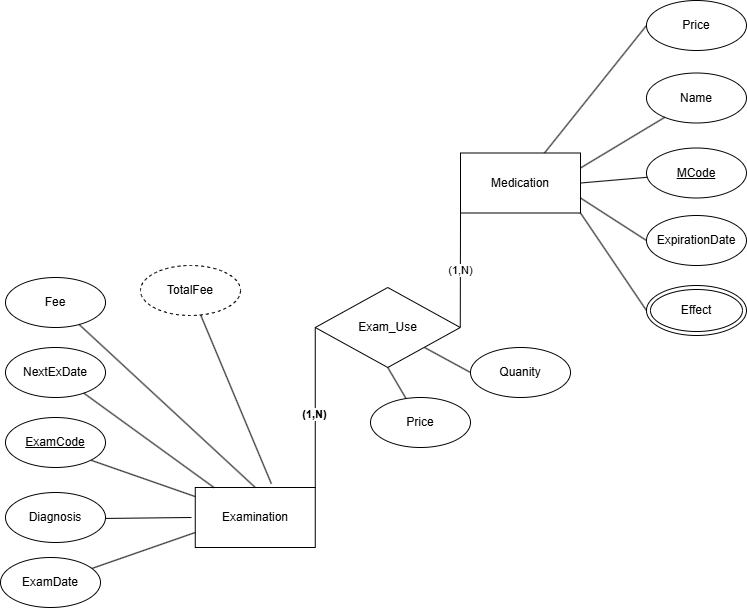
Hình 2. 13. Mô hình liên kết giữa bác sĩ và khám bệnh

- Mô hình liên kết giữa khám bệnh và bệnh nhân ngoại trú



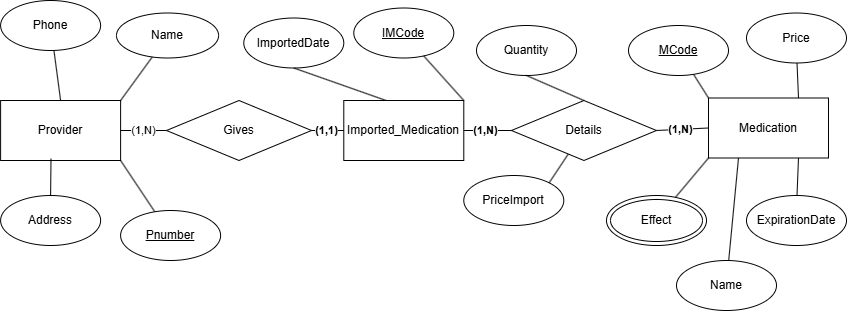
Hình 2. 14. Mô hình liên kết giữa khám bệnh và bệnh nhân ngoại trú

- Mô hình liên kết giữa thuốc men và khám bệnh



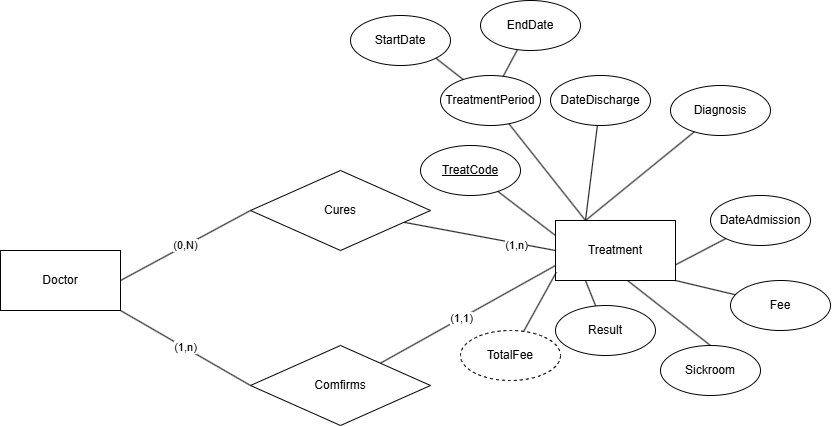
Hình 2. 15. Mô hình liên kết giữa khám bệnh và thuốc men

- Mô hình liên kết giữa nhà cung cấp, nhập thuốc men và thuốc men



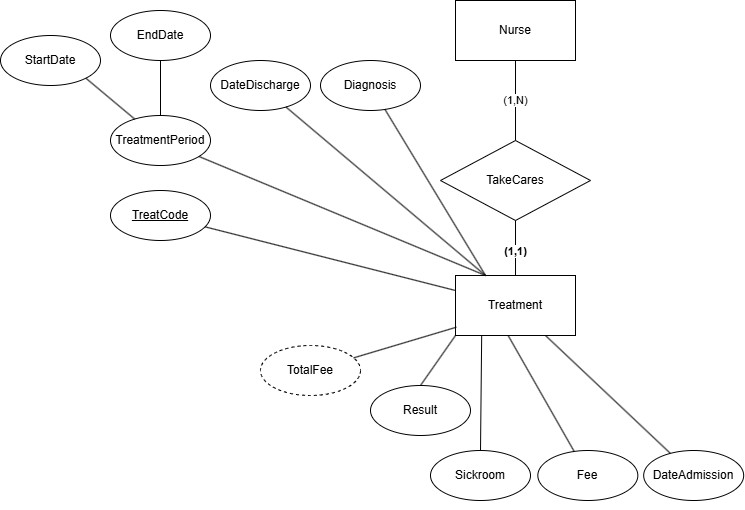
Hình 2. 16. Mô hình liên kết giữa nhà cung cấp, nhập thuốc men và thuốc men

- Mô hình liên kết giữa bác sĩ và điều trị



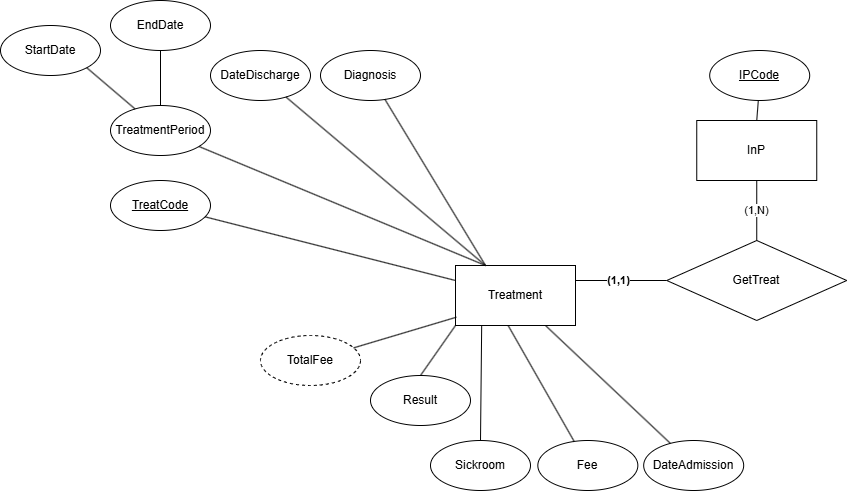
Hình 2. 17. Mô hình liên kết giữa bác sĩ và điều trị

- Mô hình liên kết giữa y tá và điều trị



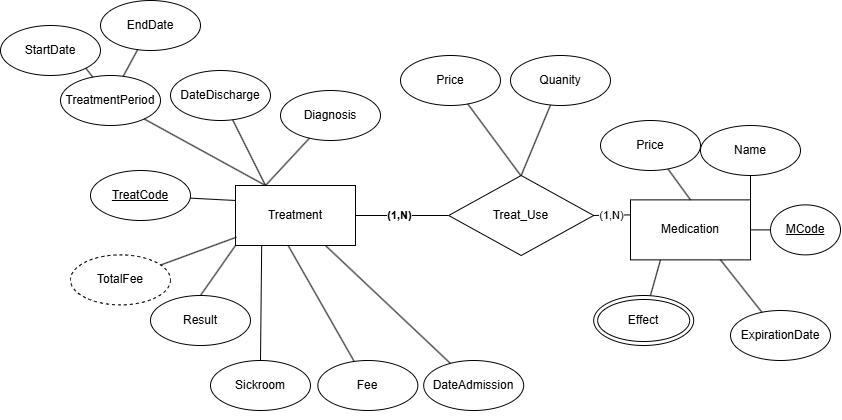
Hình 2. 18. Mô hình liên kết giữa y tá và điều trị

- Mô hình liên kết giữa điều trị và bệnh nhân nội trú



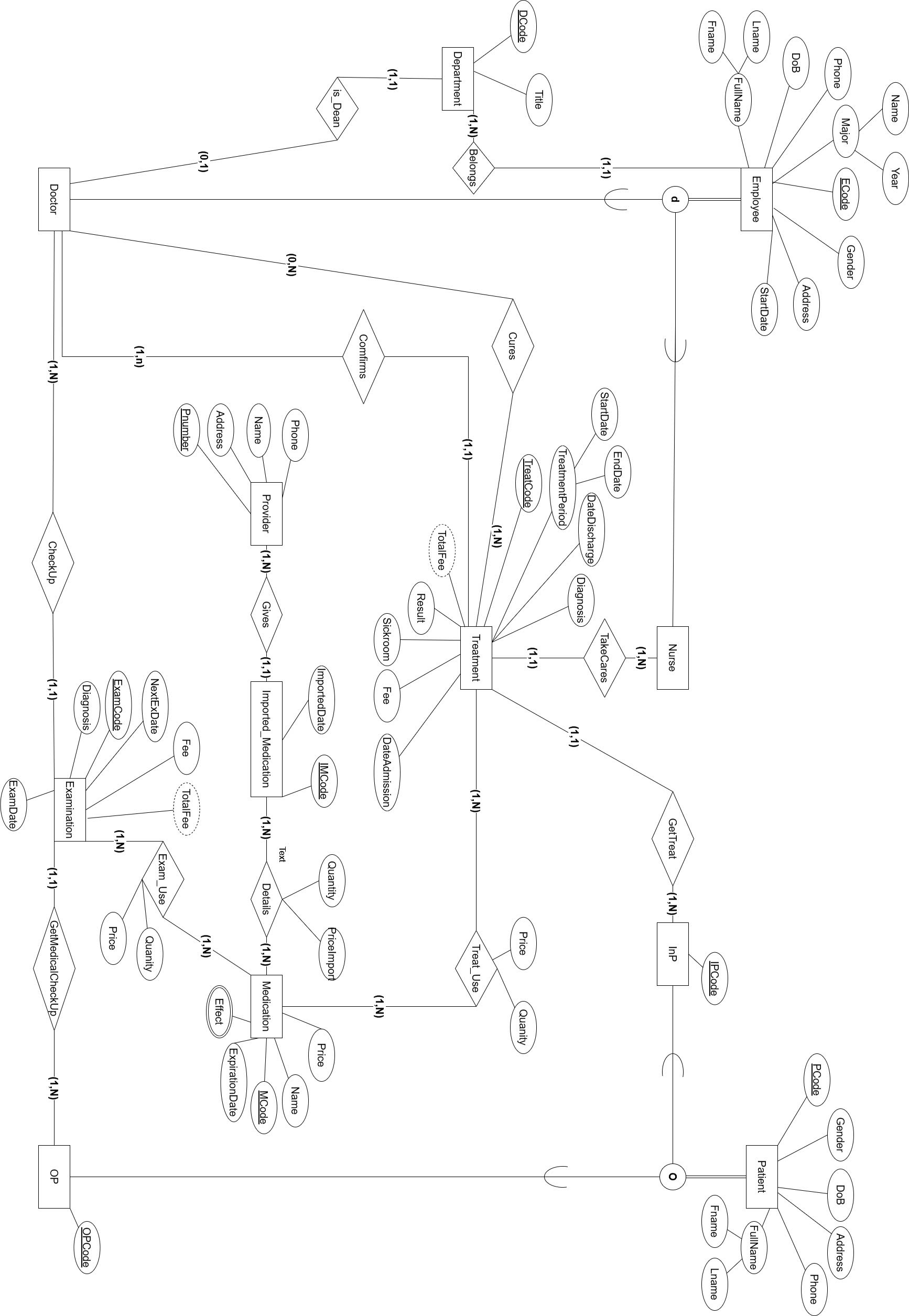
Hình 2. 19. Mô hình liên kết giữa điều trị và bệnh nhân nội trú

- Mô hình liên kết giữa điều trị và thuốc men



Hình 2. 20. Mô hình liên kết giữa điều trị và thuốc men

- Toàn bộ các mối liên kết trong sơ đồ (E)ER



Hình 2. 21. Toàn bộ mối liên kết

## **2.2. Chuyển từ mô hình thực thể liên kết sang mô hình quan hệ dữ liệu**

### **2.2.1. Cách chuyển đổi**

- Các quy tắc chuyển từ mô hình thực thể liên kết sang mô hình quan hệ:

+ Chuyển đổi các tập thực thể: Mỗi thực thể được chuyển thành một quan hệ cùng tên và danh sách thuộc tính. Đối với kiểu thực thể thông thường (regular entitytype): Khóa của quan hệ là khóa của kiểu thực thể. Thuộc tính của quan hệ cũng chính là thuộc tính của kiểu thực thể. Quan hệ không chứa các thuộc tính đa trị mà chỉ chứa thuộc tính thành phần của những thuộc tính phức hợp.

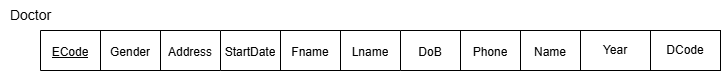
+ Chuyển đổi mối quan hệ 1 – 1: Thuộc tính khóa bên này sẽ làm khóa ngoại của bên kia và ngược lại.

+ Chuyển đổi mối quan hệ 1 – n: Thuộc tính khóa bên 1 làm khóa ngoại bên nhiều.

+ Chuyển đổi mối quan hệ n – n: Chuyển thành quan hệ mới có khóa chính gồm 2 thuộc tính khóa của 2 quan hệ trở lên, thuộc tính mối kết hợp (nếu có) trở thành thuộc tính của quan hệ mới.

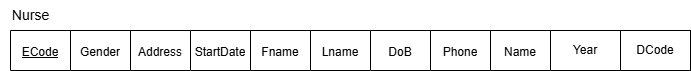
### **2.2.2. Xác định các bảng và thuộc tính của bảng**

- Doctor (ECode, Gender, Address, StartDate, Fname, Lname, DoB, Phone, Name, Year, DCode)



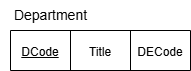
Hình 2. 22. Bảng Doctor

- Nurse (ECode, Gender, Address, StartDate, Fname, Lname, DoB, Phone, Name, Year, DCode)



Hình 2. 23. Bảng Nurse

- Department (DCode, Title, DECode)



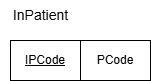
Hình 2. 24. Bảng Department

- Patient (PCode, Gender, DoB, Address, Phone, Fname, Lname)



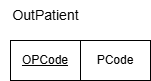
Hình 2. 25. Bảng Paient

- InPatient (IPCode, PCode)



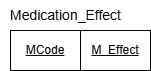
Hình 2. 26. Bảng InPatient

- OutPatient (OPCode, PCode)



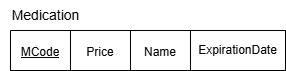
Hình 2. 27. Bảng OutPatient

- Medication\_Effect(MCode, M\_Effect)



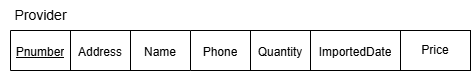
Hình 2. 28. Bảng Medication\_Effect

- Medication (MCode, Price, Name, Expiration)



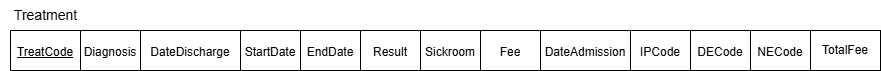
Hình 2. 29. Bảng Medication

- Provider (Pnumber, Address, Name, Phone, Quantity, ImportedData, Price)



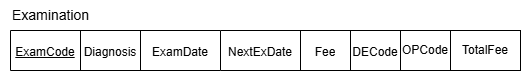
Hình 2. 30. Bảng Provider

- Treatment (TreatCode, Diagnosis, DateDischarge, StartDate, EndDate, Result, Sickroom, Fee, DateAdmission, IPCode, DECode, NECode, TotalFee)



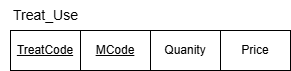
Hình 2. 31. Bảng Treatment

- Examination (ExamCode, Diagnosis, ExamDate, NextExDate, Fee, DECode, OPCode, TotalFee)



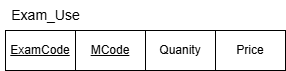
Hình 2. 32. Bảng Examination

- Treat\_Use (TreatCode, MCode, Quantity, Price)



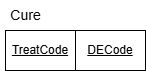
Hình 2. 33. Bảng Treat\_Use

- Exam\_Use (ExamCode, MCode, Quantity, Price)

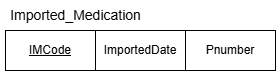


Hình 2. 34. Bảng Exam\_Use

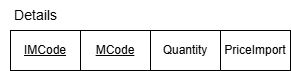
- Cure (TreatCode, DECode)



Hình 2. 35. Bảng Cure

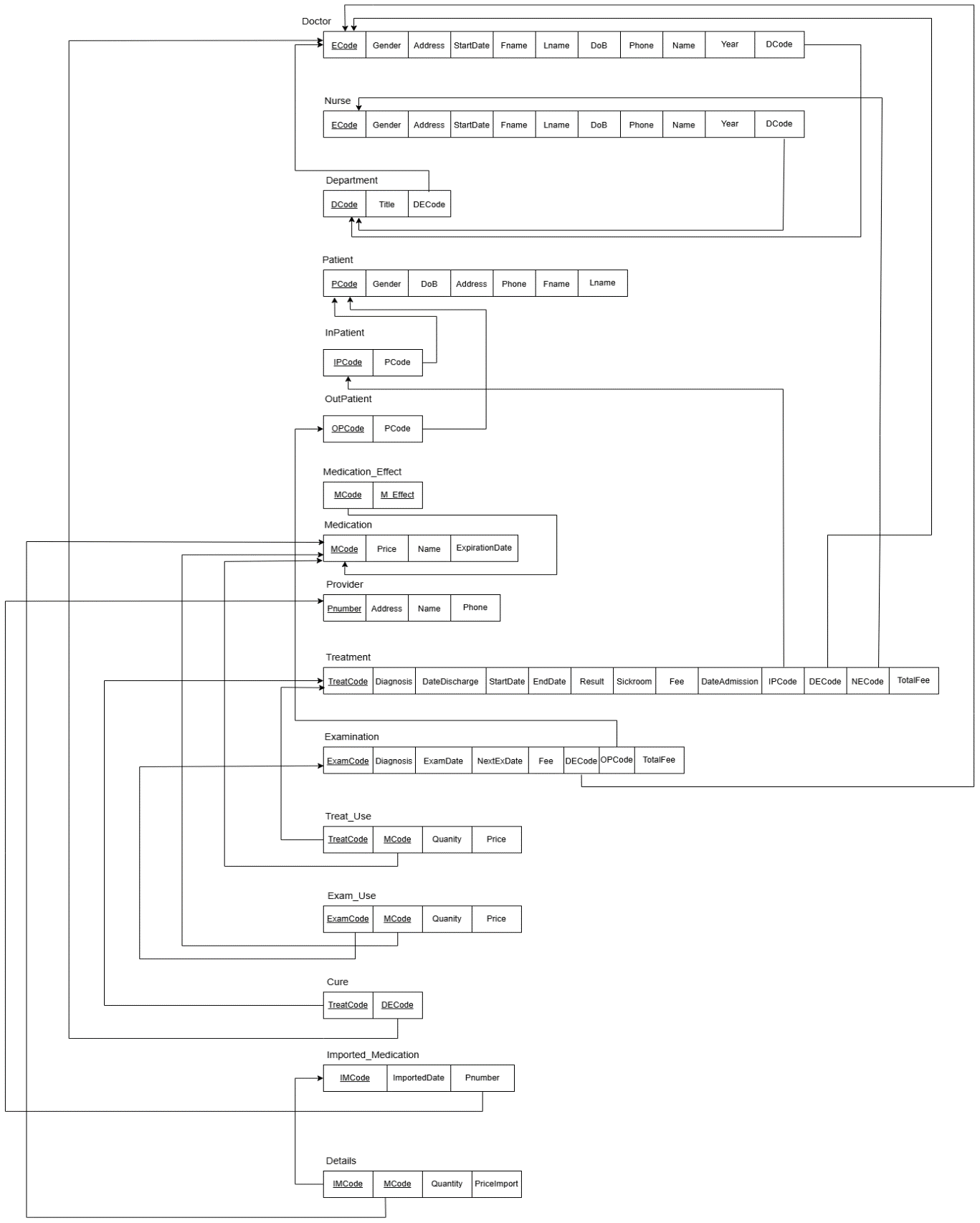
- Imported\_Medication (IMCode, ImportedDate, Pnumber) 

Hình 2. 36. Bảng Imported\_Medication

- Details (IMCode, MCode, Quantity, PriceImport) 

Hình 2. 37. Bảng Details

### **2.2.3. Mô hình dữ liệu quan hệ của bệnh viện**



Hình 2. 38. Tổng quan mô hình dữ liệu quan hệ

### **2.2.4. Xác định phụ thuộc hàm**

Với cơ sở dữ liệu bệnh viện sẽ có những phụ thuộc hàm như sau:

- Doctor (ECode#, Gender, Address, StartDate, Fname, Lname, DoB, Phone, Name, Year, DCode#):

Ecode# → Gender, Address, StartDate, Fname, Lname, DoB, Phone, Name, Year, DCode#

- Nurse (ECode#, Gender, Address, StartDate, Fname, Lname, DoB, Phone, Name, Year, DCode#)

Ecode# → Gender, Address, StartDate, Fname, Lname, DoB, Phone, Name, Year, DCode#

- Department (DCode#, Title, DECode#)

DCode# → Title, DECode

- InPatient (PCode#, Gender, DoB, Address, Phone, Fname, Lname)

PCode# → Gender, DoB, Address, Phone, Fname, Lname

- OutPatient (PCode, Gender, DoB, Address, Phone, Fname, Lname)

PCode# → Gender, DoB, Address, Phone, Fname, Lname

- Medication (MCode#, Price, Name, Expiration, Effect)

MCode# → Price, Name, Expiration, Effect

- Provider (Pnumber#, Address, Name, Phone, Quantity, ImportedData, Price)

Pnumber# → Address, Name, Phone, Quantity, ImportedData, Price

- Treatment (TreatCode#, Diagnosis, DateDischarge, StartDate, EndDate, Result, Sickroom, Fee, DateAdmission, IPCode#, DECode#, NECode#)

TreatCode# → Diagnosis, DateDischarge, StartDate, EndDate, Result, Sickroom, Fee, DateAdmission, IPCode#, DECode#, NECode#

- Examination (ExamCode#, Diagnosis, ExamDate, NextExDate, Fee, DECode#, OPCode#)

ExamCode# → Diagnosis, ExamDate, NextExDate, Fee, DECode#, OPCode#

- Treat\_Use (TreatCode#, MCode#, Quantity, Price)

TreatCode#, MCode# → Quantity, Price

MCode# → Price

TreatCode# → Quantity

- Exam\_Use (ExamCode#, MCode#, Quantity, Price)

ExamCode#, MCode# → Quantity, Price

MCode# → Price

ExamCode# → Quantity

- Imported\_Medication (IMCode#, ImportedDate, Pnumber#)

IMCode# → ImportedDate, Pnumber#

Details (IMCode#, MCode#, Quantity, PriceImport)

IMCode#, MCode# → Quantity, PriceImport

MCode# → PriceImport

IMCode# → Quantity

## **2.3. Xác định RBTV (Ràng buộc toàn vẹn)**

### **2.3.1. Định nghĩa**

Trong một CSDL luôn luôn có nhiều mối liên hệ, nhiều sự ràng buộc qua lại giữa các thuộc tính, giữa các bộ với nhau. Các mối liên hệ ràng buộc này là những điều kiện bất biến mà tất cả các bộ của những quan hệ có liện quan trong CSDL đều phải thỏa mãn ở bất kỳ thời điểm nào. Những điều kiện bất biến đó được gọi là “ràng buộc toàn vẹn”.

Các rèn buộc toàn vẹn dữ liệu được sử dụng trong đề tài gồm:

- Constraint Default

- Constraint Not Null

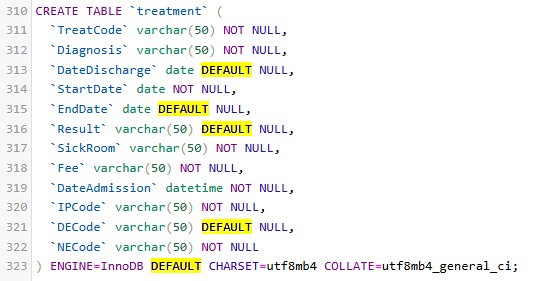
- Constraint Primary Key

- Constraint Foreign Key

- Constraint Unique

**a. Default Constraint**

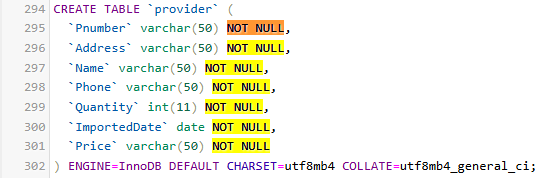
Cột sẽ nhận giá trị mặc định nếu không cung cấp giá trị khi thêm dữ liệu vào bảng.



Hình 2. 39. Ràng buộc kiểu Default

**b. Not Null Constraint**

Cột không được để trống dữ liệu.

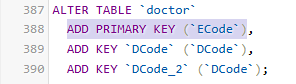


Hình 2. 40. Ràng buộc kiểu Not Null

**c. Primary Key Constraint**

Là loại ràng buộc chỉ xác định duy nhất từng giá trị trong bảng.

Mỗi bảng chỉ chứa duy nhất một khóa chính.



Hình 2. 41. Ràng buộc kiểu Primary Key (khóa chính)

**d. Foreign Key Constraint**

Ràng buộc FOREIGN KEY được sử dụng để ngăn chặn các hành động có thể phá hủy các liên kết giữa các bảng.

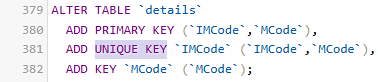
FOREIGN KEY là một trường (hoặc tập hợp các trường) trong một bảng, tham chiếu đến PRIMARY KEY trong một bảng khác.



Hình 2. 42. Ràng buộc kiểu Foreign Key (khóa ngoại)

**e. Unique Constraint**

Đảm bảo rằng giá trị của mỗi cột sẽ khác nhau



Hình 2. 43. Ràng buộc kiểu Unique (độc quyền)

### **2.3.2. Xác định RBTV (Ràng buộc toàn vẹn)**

Gồm các điều kiện của RBTV và biểu diễn của chúng, đồng thời ta lập bảng tầm ảnh hưởng của mỗi RBTV.

Dấu (+): RBTV cần được kiểm tra nguy cơ dẫn tới vi phạm.

Dấu (-): RBTV không có nguy cơ bị vi phạm.

Dấu (-(\*)): RBTV không bị vi phạm vì không được phép sửa đổi.

- R1: Mỗi Employee có ECode khác nhau không trùng với ECode nào khác

∀Employee1, Employee2 € Employee, Employee1 ≠ Employee2

=> Employee1.ECode ≠ Employee2.ECode

Bảng tầm ảnh hưởng của R1

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Employee | +(ECode) | -(\*) | - |

Bảng 2. 1. Bảng tầm ảnh hưởng R1

- R2: mỗi Department có DCode khác nhau không trùng với DCode nào khác

∀D1, D2 € Department, D1 ≠ D2

=> D1.DCode ≠ D2.DCode

Bảng tầm ảnh hưởng của R2

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Department | +(DCode) | -(\*) | - |

Bảng 2. 2. Bảng tầm ảnh hưởng R2

- R3: mỗi Patient có PCode khác nhau không trùng với PCode nào khác

∀P1, P2 € Patient, P1 ≠ P2

=> P1.PCode ≠ P2.PCode

Bảng tầm ảnh hưởng của R3

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Patient | +(PCode) | -(\*) | + |

Bảng 2. 3. Bảng tầm ảnh hưởng R3

- R4: mỗi Medication có MCode khác nhau không trùng với MCode nào khác.

∀M1, M2 € Medication, M1 ≠ M2

=> M1.MCode ≠ M2.MCode

Bảng tầm ảnh hưởng của R4

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Medication | +(MCode) | -(\*) | - |

Bảng 2. 4. Bảng ầm ảnh hưởng R4

- R5: mỗi Examination có ExamCode khác nhau không trùng với ExamCode nào khác

Với mọi EX1, EX2 € Examination, EX1 ≠ EX2

=> EX1.ExamCode ≠ EX2.ExamCode

Bảng tầm ảnh hưởng của R5

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Examination | +(ExamCode) | -(\*) | - |

Bảng 2. 5. Bảng tầm ảnh hưởng R5

- R6: mỗi Treatment có TreatCode khác nhau không trùng với TreatCode nào khác

∀TC1, TC2 € Treatment, TC1 ≠ TC2

=> TC1.TreatCode ≠ TC2.TreatCode

Bảng tầm ảnh hưởng của R6

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Treatment | +(TreatCode) | -(\*) | - |

Bảng 2. 6. Bảng tầm ảnh hưởng R6

- R7: mỗi Imported\_Medication có IMCode khác nhau không trùng với IMCode nào khác.

∀IM1, IM2 € Imported\_Medication, IM1 ≠ IM2

=> IM1.IMCode ≠ IM2.IMCode

Bảng tầm ảnh hưởng của R7

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Imported\_Medication | +(IMCode) | -(\*) | - |

Bảng 2. 7. Bảng tầm ảnh hưởng R7

- R8: mỗi Provider có Pnumber khác nhau không trùng với Pnumber nào khác.

Với mọi Pn1, Pn2 € Provider, Pn1 ≠ Pn2

=> Pn1.Pnumber ≠ Pn2.Pnumber

Bảng tầm ảnh hưởng của R8

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Provider | +(Pnumber) | -(\*) | - |

Bảng 2. 8. Bảng tầm ảnh hưởng R8

- R9: Trưởng khoa phải là một trong các nhân sự thuộc phòng ban đó

∀t ∈ Department (∃u ∈ Doctor(u.ECode = t.DECode ∧ u.DCode = t.DCode))

Bảng tầm ảnh hưởng của R9

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Doctor | - | +(DCode) | + |
| Department | + | +(DECode) | - |

Bảng 2. 9. Bảng tầm ảnh hưởng R9

- R10: Trưởng khoa thì phải có một ngành cụ thể và có ít nhất 5 năm kinh nghiệm.

∀t ∈ Department (∃u ∈ Doctor(u.ECode = t.DECode ∧ u.Year >= 5 ∧ u.Name ≠ null))

Bảng tầm ảnh hưởng của R10

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Doctor | - | +(DCode, u.Name, u.Year) | + |
| Department | +(DCode) | +(DECode) | - |

Bảng 2. 10. Bảng tầm ảnh hưởng R10

- R11: Mỗi nhân sự đều phải thuộc một phòng ban nào đó.

∀t ∈ Department (∀u ∈ Doctor(u.DCode = t.DCode ∧ t.DCode ≠ null))

Bảng tầm ảnh hưởng của R11

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Doctor | + | +(DCode) | - |
| Department | - | +(DCode) | + |

Bảng 2. 11. Bảng tầm ảnh hưởng R11

- R12: Kết quả của chữa trị chỉ tồn tại 2 trạng thái “Recovered” hoặc “Have not recovered”.

∀t ∈ Treatment ,t.Result ∈ {“Recovered”, “Have not recovered”}

Bảng tầm ảnh hưởng của R12

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Treatment | + | +(t.Result) | - |

Bảng 2. 12. Bảng tầm ảnh hưởng R12

- R13: Ngày hết hạn của các loại thuốc không được phép là ngày hôm nay hoặc các ngày trước đó

∀t ∈ Medication, t.ExpirationDate > CurrentDate

Bảng tầm ảnh hưởng của R13

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Medication | +(ExpirationDate) | +(ExpirationDate) | - |

Bảng 2. 13. Bảng tầm ảnh hưởng R13

- R14: Giá của một loại thuốc không thể là số âm

∀t ∈ Medication, t.Price > 0

Bảng tầm ảnh hưởng của R14

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Sửa | Xóa |
| Medication | +(Price) | +(Price) | - |

Bảng 2. 14. Bảng tầm ảnh hưởng R14

## **2.4. Mã nguồn cơ sở dữ liệu SQL (Structured Query Language)**

CREATE TABLE `cure` (

`TreatCode` varchar(50) NOT NULL,

`DECode` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `cure`

--

INSERT INTO `cure` (`TreatCode`, `DECode`) VALUES

('T000000001', 'D123'),

('T000000001', 'D345'),

('T000000002', 'D234'),

('T000000002', 'D567'),

('T000000003', 'D234'),

('T000000003', 'D567');

-- --------------------------------------------------------

--

-- Table structure for table `department`

--

CREATE TABLE `department` (

`DCode` varchar(50) NOT NULL,

`Title` varchar(50) NOT NULL,

`DECode` varchar(50) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `department`

--

INSERT INTO `department` (`DCode`, `Title`, `DECode`) VALUES

('1', 'Cardiology', 'D123'),

('2', 'Ophthalmology', 'D234'),

('3', 'Pharmacy', 'D345');

-- --------------------------------------------------------

--

-- Table structure for table `details`

--

CREATE TABLE `details` (

`IMCode` varchar(50) NOT NULL,

`MCode` varchar(50) NOT NULL,

`Quantity` int(11) NOT NULL,

`PriceImport` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `doctor`

--

CREATE TABLE `doctor` (

`ECode` varchar(50) NOT NULL,

`Gender` varchar(50) NOT NULL,

`Address` varchar(50) NOT NULL,

`StartDate` date NOT NULL,

`Fname` varchar(50) NOT NULL,

`Lname` varchar(50) NOT NULL,

`DoB` date NOT NULL,

`Phone` varchar(50) NOT NULL,

`Name` varchar(50) NOT NULL,

`Year` int(11) NOT NULL,

`DCode` varchar(50) NOT NULL,

`password` varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `doctor`

--

INSERT INTO `doctor` (`ECode`, `Gender`, `Address`, `StartDate`, `Fname`, `Lname`, `DoB`, `Phone`, `Name`, `Year`, `DCode`, `password`) VALUES

('D123', 'Male', '102/6 Hoa Hung', '2024-11-01', 'Tran', 'Hiep', '2024-11-01', '0909930828', 'ORB', 3, '1', 'adcd7048512e64b48da55b027577886ee5a36350'),

('D234', 'Female', '123 An Duong Vuong', '2024-11-22', 'Nguyen', 'Anh', '2024-11-23', '0909930828', 'HHH', 2, '2', 'adcd7048512e64b48da55b027577886ee5a36350'),

('D345', 'Male', '102/6 Hoa Hung', '2024-11-01', 'Trinh', 'Dien', '2024-11-01', '0909930828', 'ORB', 3, '3', 'adcd7048512e64b48da55b027577886ee5a36350'),

('D456', 'Male', '102/6 Hoa Hung', '2024-11-01', 'Ngo', 'Vinh', '2024-11-01', '0909930828', 'ORB', 3, '1', 'adcd7048512e64b48da55b027577886ee5a36350'),

('D567', 'Male', '102/6 Hoa Hung', '2024-11-01', 'Le', 'Phat', '2024-11-01', '1111111111111', 'ORB', 3, '2', 'adcd7048512e64b48da55b027577886ee5a36350');

-- --------------------------------------------------------

--

-- Table structure for table `examination`

--

CREATE TABLE `examination` (

`ExamCode` varchar(50) NOT NULL,

`Diagnosis` varchar(50) NOT NULL,

`ExamDate` datetime NOT NULL,

`NextExDate` date NOT NULL,

`Fee` varchar(50) NOT NULL,

`DECode` varchar(50) NOT NULL,

`OPCode` varchar(50) NOT NULL,

`TotalFee` int(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `examination`

--

INSERT INTO `examination` (`ExamCode`, `Diagnosis`, `ExamDate`, `NextExDate`, `Fee`, `DECode`, `OPCode`, `TotalFee`) VALUES

('EX000000001', 'Headache', '2024-11-23 09:14:03', '2024-11-26', '100', 'D123', 'OP000000001', 310),

('EX000000002', 'Toothache', '2024-11-23 09:15:10', '2024-11-27', '100', 'D123', 'OP000000002', 280),

('EX000000003', '231', '2024-11-27 15:17:26', '0000-00-00', '100', 'D123', 'OP000000003', 130),

('EX000000004', 'sick', '2024-11-27 20:43:35', '0000-00-00', '100', 'D123', 'OP000000001', 110),

('EX000000005', 'Stomache', '2024-11-30 22:18:34', '0000-00-00', '120', 'D234', 'OP000000001', 160);

-- --------------------------------------------------------

--

-- Table structure for table `exam\_use`

--

CREATE TABLE `exam\_use` (

`ExamCode` varchar(50) NOT NULL,

`MCode` varchar(50) NOT NULL,

`quantity` int(10) NOT NULL,

`price` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `exam\_use`

--

INSERT INTO `exam\_use` (`ExamCode`, `MCode`, `quantity`, `price`) VALUES

('EX000000001', 'MED1', 6, 10),

('EX000000001', 'MED2', 6, 20),

('EX000000001', 'MED3', 6, 5),

('EX000000002', 'MED1', 5, 10),

('EX000000002', 'MED2', 5, 20),

('EX000000002', 'MED4', 5, 6),

('EX000000003', 'MED1', 2, 10),

('EX000000003', 'MED3', 2, 5),

('EX000000004', 'MED1', 1, 10),

('EX000000005', 'MED1', 5, 10),

('EX000000005', 'MED2', 5, 20),

('EX000000005', 'MED4', 5, 6);

-- --------------------------------------------------------

--

-- Table structure for table `imported\_medication`

--

CREATE TABLE `imported\_medication` (

`IMCode` varchar(50) NOT NULL,

`ImportedDate` date NOT NULL,

`Pnumber` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `inpatient`

--

CREATE TABLE `inpatient` (

`IPCode` varchar(50) NOT NULL,

`PCode` int(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `inpatient`

--

INSERT INTO `inpatient` (`IPCode`, `PCode`) VALUES

('IP000000003', 8),

('IP000000001', 9),

('IP000000002', 11);

-- --------------------------------------------------------

--

-- Table structure for table `medication`

--

CREATE TABLE `medication` (

`MCode` varchar(50) NOT NULL,

`Price` int(50) NOT NULL,

`Name` varchar(50) NOT NULL,

`ExpirationDate` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `medication`

--

INSERT INTO `medication` (`MCode`, `Price`, `Name`, `ExpirationDate`) VALUES

('MED1', 10, 'ABC', '2024-11-22'),

('MED2', 20, 'DEF', '2024-11-30'),

('MED3', 5, 'GIH', '2024-11-22'),

('MED4', 6, 'JKL', '2024-11-30');

-- --------------------------------------------------------

--

-- Table structure for table `medication\_effect`

--

CREATE TABLE `medication\_effect` (

`MCode` varchar(50) NOT NULL,

`M\_effect` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `medication\_effect`

--

INSERT INTO `medication\_effect` (`MCode`, `M\_effect`) VALUES

('MED1', 'Antibiotic '),

('MED1', 'relieve pain'),

('MED2', 'Allergy '),

('MED2', 'Anesthesia '),

('MED3', 'Aspirin'),

('MED4', 'antipyretic');

-- --------------------------------------------------------

--

-- Table structure for table `nurse`

--

CREATE TABLE `nurse` (

`ECode` varchar(50) NOT NULL,

`Gender` varchar(50) NOT NULL,

`Address` varchar(50) NOT NULL,

`StartDate` date NOT NULL,

`Fname` varchar(50) NOT NULL,

`Lname` varchar(50) NOT NULL,

`DoB` date NOT NULL,

`Phone` varchar(50) NOT NULL,

`Name` varchar(50) NOT NULL,

`Year` int(11) NOT NULL,

`DCode` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `nurse`

--

INSERT INTO `nurse` (`ECode`, `Gender`, `Address`, `StartDate`, `Fname`, `Lname`, `DoB`, `Phone`, `Name`, `Year`, `DCode`) VALUES

('N0001', 'Female', '123 Thanh Thai', '2024-11-06', 'Nguyen', 'A', '2024-11-01', '0909930828', 'OOOO', 3, '2');

-- --------------------------------------------------------

--

-- Table structure for table `outpatient`

--

CREATE TABLE `outpatient` (

`OPCode` varchar(50) NOT NULL,

`PCode` int(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `outpatient`

--

INSERT INTO `outpatient` (`OPCode`, `PCode`) VALUES

('OP000000001', 8),

('OP000000002', 10),

('OP000000003', 11);

-- --------------------------------------------------------

--

-- Table structure for table `patient`

--

CREATE TABLE `patient` (

`PCode` int(50) NOT NULL,

`Gender` varchar(50) NOT NULL,

`DoB` date NOT NULL,

`Address` varchar(50) NOT NULL,

`Phone` varchar(50) NOT NULL,

`Fname` varchar(50) NOT NULL,

`Lname` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `patient`

--

INSERT INTO `patient` (`PCode`, `Gender`, `DoB`, `Address`, `Phone`, `Fname`, `Lname`) VALUES

(8, 'Male', '2004-01-11', '115 An Duong Vuong', '0000000000', 'Tran', 'Dien'),

(9, 'Male', '2004-02-22', '123 Cong Hoa', '11111111111', 'Ngo', 'Anh'),

(10, 'Male', '2004-09-04', '102/6 Hoa Hung', '0909930828', 'Nguyen', 'Vinh'),

(11, 'Male', '2004-03-31', '333 Truong Chinh', '5555555555', 'Hiep', 'Le'),

(12, 'Male', '2004-12-12', '999 Ly Tu Trong', '8888888888', 'Trinh', 'Phat');

-- --------------------------------------------------------

--

-- Table structure for table `provider`

--

CREATE TABLE `provider` (

`Pnumber` varchar(50) NOT NULL,

`Address` varchar(50) NOT NULL,

`Name` varchar(50) NOT NULL,

`Phone` varchar(50) NOT NULL,

`Quantity` int(11) NOT NULL,

`ImportedDate` date NOT NULL,

`Price` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `treatment`

--

CREATE TABLE `treatment` (

`TreatCode` varchar(50) NOT NULL,

`Diagnosis` varchar(50) NOT NULL,

`DateDischarge` date DEFAULT NULL,

`StartDate` date NOT NULL,

`EndDate` date DEFAULT NULL,

`Result` varchar(50) DEFAULT NULL,

`SickRoom` varchar(50) NOT NULL,

`Fee` varchar(50) NOT NULL,

`DateAdmission` datetime NOT NULL,

`IPCode` varchar(50) NOT NULL,

`DECode` varchar(50) DEFAULT NULL,

`NECode` varchar(50) NOT NULL,

`TotalFee` int(20) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `treatment`

--

INSERT INTO `treatment` (`TreatCode`, `Diagnosis`, `DateDischarge`, `StartDate`, `EndDate`, `Result`, `SickRoom`, `Fee`, `DateAdmission`, `IPCode`, `DECode`, `NECode`, `TotalFee`) VALUES

('T000000001', 'Broken arm', '2024-11-29', '2024-11-24', '2024-11-27', 'Recovered', 'R115', '500', '2024-11-23 09:17:12', 'IP000000001', 'D123', 'N0001', 548),

('T000000002', 'Burn', '2024-11-26', '2024-11-23', '2024-11-23', 'Have not recovered', 'R188', '500', '2024-11-23 09:18:49', 'IP000000002', 'D567', 'N0001', 160),

('T000000003', 'Broken arm', NULL, '2024-12-01', NULL, NULL, '115', '2000', '2024-11-30 22:03:45', 'IP000000003', NULL, 'N0001', 2180);

-- --------------------------------------------------------

--

-- Table structure for table `treat\_use`

--

CREATE TABLE `treat\_use` (

`TreatCode` varchar(50) NOT NULL,

`MCode` varchar(50) NOT NULL,

`quantity` int(10) NOT NULL,

`price` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `treat\_use`

--

INSERT INTO `treat\_use` (`TreatCode`, `MCode`, `quantity`, `price`) VALUES

('T000000001', 'MED1', 3, 10),

('T000000001', 'MED4', 3, 6),

('T000000002', 'MED1', 4, 10),

('T000000002', 'MED2', 4, 20),

('T000000002', 'MED3', 4, 5),

('T000000003', 'MED1', 5, 10),

('T000000003', 'MED2', 5, 20),

('T000000003', 'MED4', 5, 6);

--

-- Indexes for dumped tables

--

--

-- Indexes for table `cure`

--

ALTER TABLE `cure`

ADD PRIMARY KEY (`TreatCode`,`DECode`),

ADD KEY `TreatCode` (`TreatCode`),

ADD KEY `DECode` (`DECode`);

--

-- Indexes for table `department`

--

ALTER TABLE `department`

ADD PRIMARY KEY (`DCode`),

ADD KEY `DECode` (`DECode`);

--

-- Indexes for table `details`

--

ALTER TABLE `details`

ADD PRIMARY KEY (`IMCode`,`MCode`),

ADD UNIQUE KEY `IMCode` (`IMCode`,`MCode`),

ADD KEY `MCode` (`MCode`);

--

-- Indexes for table `doctor`

--

ALTER TABLE `doctor`

ADD PRIMARY KEY (`ECode`),

ADD KEY `DCode` (`DCode`),

ADD KEY `DCode\_2` (`DCode`);

--

-- Indexes for table `examination`

--

ALTER TABLE `examination`

ADD PRIMARY KEY (`ExamCode`),

ADD KEY `DECode` (`DECode`),

ADD KEY `OPCode` (`OPCode`);

--

-- Indexes for table `exam\_use`

--

ALTER TABLE `exam\_use`

ADD PRIMARY KEY (`ExamCode`,`MCode`),

ADD KEY `ExamCode` (`ExamCode`),

ADD KEY `MCode` (`MCode`);

--

-- Indexes for table `imported\_medication`

--

ALTER TABLE `imported\_medication`

ADD PRIMARY KEY (`IMCode`),

ADD KEY `Pnumber` (`Pnumber`);

--

-- Indexes for table `inpatient`

--

ALTER TABLE `inpatient`

ADD PRIMARY KEY (`IPCode`),

ADD KEY `PCode` (`PCode`);

--

-- Indexes for table `medication`

--

ALTER TABLE `medication`

ADD PRIMARY KEY (`MCode`);

--

-- Indexes for table `medication\_effect`

--

ALTER TABLE `medication\_effect`

ADD PRIMARY KEY (`MCode`,`M\_effect`),

ADD KEY `MCode` (`MCode`);

--

-- Indexes for table `nurse`

--

ALTER TABLE `nurse`

ADD PRIMARY KEY (`ECode`),

ADD KEY `DCode` (`DCode`);

--

-- Indexes for table `outpatient`

--

ALTER TABLE `outpatient`

ADD PRIMARY KEY (`OPCode`),

ADD KEY `PCode` (`PCode`),

ADD KEY `PCode\_2` (`PCode`);

--

-- Indexes for table `patient`

--

ALTER TABLE `patient`

ADD PRIMARY KEY (`PCode`),

ADD KEY `Phone` (`Phone`),

ADD KEY `Fname` (`Fname`),

ADD KEY `Lname` (`Lname`);

--

-- Indexes for table `provider`

--

ALTER TABLE `provider`

ADD PRIMARY KEY (`Pnumber`);

--

-- Indexes for table `treatment`

--

ALTER TABLE `treatment`

ADD PRIMARY KEY (`TreatCode`),

ADD KEY `IPCode` (`IPCode`),

ADD KEY `NECode` (`NECode`),

ADD KEY `DECode` (`DECode`);

--

-- Indexes for table `treat\_use`

--

ALTER TABLE `treat\_use`

ADD PRIMARY KEY (`TreatCode`,`MCode`),

ADD KEY `TreatCode` (`TreatCode`),

ADD KEY `MCode` (`MCode`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `patient`

--

ALTER TABLE `patient`

MODIFY `PCode` int(50) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=14;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `cure`

--

ALTER TABLE `cure`

ADD CONSTRAINT `cure\_ibfk\_1` FOREIGN KEY (`TreatCode`) REFERENCES `treatment` (`TreatCode`),

ADD CONSTRAINT `cure\_ibfk\_2` FOREIGN KEY (`DECode`) REFERENCES `doctor` (`ECode`);

--

-- Constraints for table `department`

--

ALTER TABLE `department`

ADD CONSTRAINT `department\_ibfk\_1` FOREIGN KEY (`DECode`) REFERENCES `doctor` (`ECode`);

--

-- Constraints for table `details`

--

ALTER TABLE `details`

ADD CONSTRAINT `details\_ibfk\_1` FOREIGN KEY (`IMCode`) REFERENCES `imported\_medication` (`IMCode`),

ADD CONSTRAINT `details\_ibfk\_2` FOREIGN KEY (`MCode`) REFERENCES `medication` (`MCode`);

--

-- Constraints for table `doctor`

--

ALTER TABLE `doctor`

ADD CONSTRAINT `doctor\_ibfk\_1` FOREIGN KEY (`DCode`) REFERENCES `department` (`DCode`);

--

-- Constraints for table `examination`

--

ALTER TABLE `examination`

ADD CONSTRAINT `examination\_ibfk\_1` FOREIGN KEY (`DECode`) REFERENCES `doctor` (`ECode`),

ADD CONSTRAINT `examination\_ibfk\_2` FOREIGN KEY (`OPCode`) REFERENCES `outpatient` (`OPCode`);

--

-- Constraints for table `exam\_use`

--

ALTER TABLE `exam\_use`

ADD CONSTRAINT `exam\_use\_ibfk\_1` FOREIGN KEY (`MCode`) REFERENCES `medication` (`MCode`),

ADD CONSTRAINT `exam\_use\_ibfk\_2` FOREIGN KEY (`ExamCode`) REFERENCES `examination` (`ExamCode`);

--

-- Constraints for table `imported\_medication`

--

ALTER TABLE `imported\_medication`

ADD CONSTRAINT `imported\_medication\_ibfk\_1` FOREIGN KEY (`Pnumber`) REFERENCES `provider` (`Pnumber`);

--

-- Constraints for table `inpatient`

--

ALTER TABLE `inpatient`

ADD CONSTRAINT `inpatient\_ibfk\_1` FOREIGN KEY (`PCode`) REFERENCES `patient` (`PCode`);

--

-- Constraints for table `medication\_effect`

--

ALTER TABLE `medication\_effect`

ADD CONSTRAINT `medication\_effect\_ibfk\_1` FOREIGN KEY (`MCode`) REFERENCES `medication` (`MCode`);

--

-- Constraints for table `nurse`

--

ALTER TABLE `nurse`

ADD CONSTRAINT `nurse\_ibfk\_1` FOREIGN KEY (`DCode`) REFERENCES `department` (`DCode`);

--

-- Constraints for table `outpatient`

--

ALTER TABLE `outpatient`

ADD CONSTRAINT `outpatient\_ibfk\_1` FOREIGN KEY (`PCode`) REFERENCES `patient` (`PCode`);

--

-- Constraints for table `treatment`

--

ALTER TABLE `treatment`

ADD CONSTRAINT `treatment\_ibfk\_1` FOREIGN KEY (`IPCode`) REFERENCES `inpatient` (`IPCode`),

ADD CONSTRAINT `treatment\_ibfk\_2` FOREIGN KEY (`DECode`) REFERENCES `doctor` (`ECode`),

ADD CONSTRAINT `treatment\_ibfk\_3` FOREIGN KEY (`NECode`) REFERENCES `nurse` (`ECode`);

--

-- Constraints for table `treat\_use`

--

ALTER TABLE `treat\_use`

ADD CONSTRAINT `treat\_use\_ibfk\_1` FOREIGN KEY (`TreatCode`) REFERENCES `treatment` (`TreatCode`),

ADD CONSTRAINT `treat\_use\_ibfk\_2` FOREIGN KEY (`MCode`) REFERENCES `medication` (`MCode`);

COMMIT;

# **CHƯƠNG 3: XÂY DỰNG ĐỀ TÀI**

## **3.1. Môi trường lập trình**

- Đề tài được sử dụng trên nền tảng website.

- Với các loại ngôn ngữ chính như: CSS, Javascript, PHP.

## **3.2. Yêu cầu về chức năng**

**Các yêu cầu chức năng sẽ được áp dụng vào đề tài như sau:**

1) Tìm kiếm thông tin bệnh nhân: Kết quả tìm kiếm bao gồm tên, số điện thoại và thông tin về việc điều trị và thăm khám của bệnh nhân.

2) Thêm thông tin cho bệnh nhân mới.

3) Liệt kê chi tiết tất cả các bệnh nhân được bác sĩ điều trị.

4) Lập báo cáo cung cấp đầy đủ thông tin về khoản thanh toán cho mỗi

điều trị hoặc khám bệnh cho bệnh nhân.

5) Chứng minh một trường hợp sử dụng hiệu quả lập chỉ mục trong các tình huống của bạn.

6) Giải quyết một trường hợp sử dụng bảo mật cơ sở dữ liệu trong các tình huống của bạn.

### **3.2.1. Chức năng tìm kiếm bệnh nhân**

$flag = false;

if(isset($\_GET['key']))

{

    $key = trim($\_GET['key']);

    $self .= "?key=".$key;

    if($key != "")

        $flag = true;

}

if ($flag) {

$sql = "SELECT \* FROM `patient` WHERE Phone = '".$key."' OR PCode = '".$key."' OR Fname = '".$key."' OR Lname = '".$key."'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

if ($result->num\_rows < 1) {

$sql = "SELECT \* FROM `patient` WHERE PCode = '".$key."' OR Fname LIKE '%".$key."%' OR Lname LIKE '%".$key."%' OR CONCAT(Fname, ' ', Lname) LIKE '%".$key."%'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

} else {

$sql = "SELECT \* FROM patient";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

**Giao diện của chức năng tìm kiếm**

<div class="table-responsive">

<table id="" class="table table-bordered toggle-circle mb-0" data-page-size="50">

<thead>

<tr>

<th>#</th>

<th>Patient Name</th>

<th>Patient Number</th>

<th>Patient Address</th>

<th>Patient Phone</th>

<th>Action</th>

</tr>

</thead>

<?php

/\*

\* Get details of all patients

\*/

if ($flag) {

$sql = "SELECT \* FROM `patient` WHERE Phone = '".$key."' OR PCode = '".$key."' OR Fname = '".$key."' OR Lname = '".$key."'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

if ($result->num\_rows < 1) {

$sql = "SELECT \* FROM `patient` WHERE PCode = '".$key."' OR Fname LIKE '%".$key."%' OR Lname LIKE '%".$key."%' OR CONCAT(Fname, ' ', Lname) LIKE '%".$key."%'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

} else {

$sql = "SELECT \* FROM patient";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

echo $time;

$cnt = 1;

while ($row = $result->fetch\_array()) {

?>

<tbody>

<tr>

<td><?php echo $cnt; ?></td>

<td><?php echo $row['Fname'] . ' ' . $row['Lname']; ?></td>

<td><?php echo $row['PCode']; ?></td>

<td><?php echo $row['Address']; ?></td>

<td><?php echo $row['Phone']; ?></td>

<td>

<a href="view\_single\_patient.php?pat\_id=<?php echo $row['PCode']; ?>" class="badge badge-success"><i class="mdi mdi-eye"></i> View</a>

<a href="addprescription.php?pat\_number=<?php echo $row['PCode']; ?>" class="badge badge-success"><i class="fas fa-highlighter"></i> Get Examination</a>

<a href="addtreatment.php?pat\_number=<?php echo $row['PCode']; ?>" class="badge badge-success"><i class="fas fa-highlighter"></i> Get Treatment</a>

</td>

</tr>

</tbody>

<?php $cnt++; } ?>

<tfoot>

<tr class="active">

<td colspan="8">

<div class="text-right">

<ul style="list-style-type: none" class="pagination-rounded justify-content-end footable-pagination m-t-10 mb-0">

<?php

$result1 = executeQuery($sql1);

$numrows = $result1->num\_rows;

if ($numrows > 0) {

$maxPage = ceil($numrows / $rowsPerPage);

$gap = 5;

$start = $pageNum - $gap;

$end = $pageNum + $gap;

if ($start <= 0) $start = 1;

if ($end >= $maxPage) $end = $maxPage;

if ($self != "manage\_patient.php") {

$self .= "&";

} else {

$self .= '?';

}

$nav = '';

if ($start > 1) {

$nav .= '<li class="footable-page"><a href="'.$self.'page=1">1</a></li>';

$nav .= '<li class="footable-page disable"><span>...</span></li>';

}

for ($page = $start; $page <= $end; $page++) {

if ($page == $pageNum) {

$nav .= '<li class="footable-page active"><a>'.$pageNum.'</a></li>';

} else {

$nav .= '<li class="footable-page"><a href="'.$self.'page='.$page.'">'.$page.'</a></li>';

}

}

if ($end < $maxPage) {

$nav .= '<li class="footable-page"><span>...</span></li>';

$nav .= '<li class="footable-page"><a href="'.$self.'page='.$maxPage.'">'.$maxPage.'</a></li>';

}

$prev = $pageNum > 1 ? '<li><a href="'.$self.'page='.($pageNum-1).'" aria-label="Previous">&lt;</a></li>' : '';

$next = $pageNum < $maxPage ? '<li><a href="'.$self.'page='.($pageNum+1).'" aria-label="Next">&gt;</a></li>' : '';

echo $prev . $nav . $next;

}

?>

</ul>

</div>

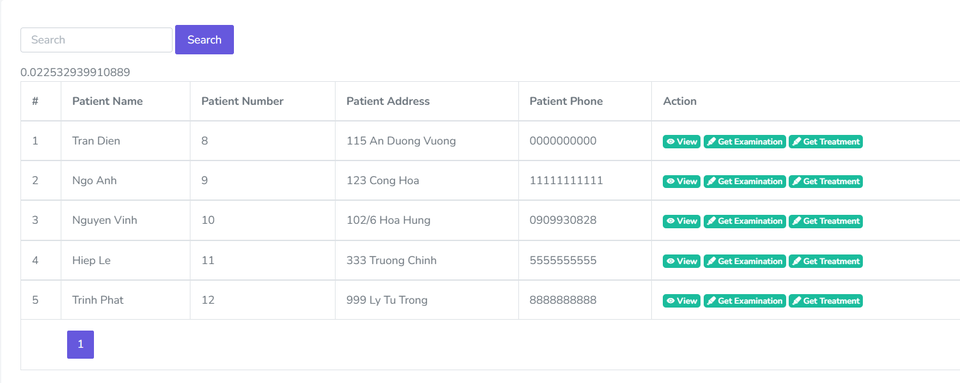
</td>

</tr>

</tfoot>

</table>

</div>



Hình 3. 1. Giao diện tìm kiếm

### **3.2.2. Chức năng thêm bệnh nhân**

- Người dùng sẽ nhấn vào nút button

 <button type="submit" name="add\_patient" class="ladda-button btn btn-primary" data-style="expand-right">Add Patient</button>

- Sau đó dữ liệu sẽ được gửi đi

<div class="table-responsive">

<table id="" class="table table-bordered toggle-circle mb-0" data-page-size="50">

<thead>

<tr>

<th>#</th>

<th>Patient Name</th>

<th>Patient Number</th>

<th>Patient Address</th>

<th>Patient Phone</th>

<th>Action</th>

</tr>

</thead>

<?php

/\*

\* Get details of all patients

\*/

if ($flag) {

$sql = "SELECT \* FROM `patient` WHERE Phone = '".$key."' OR PCode = '".$key."' OR Fname = '".$key."' OR Lname = '".$key."'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

if ($result->num\_rows < 1) {

$sql = "SELECT \* FROM `patient` WHERE PCode = '".$key."' OR Fname LIKE '%".$key."%' OR Lname LIKE '%".$key."%' OR CONCAT(Fname, ' ', Lname) LIKE '%".$key."%'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

} else {

$sql = "SELECT \* FROM patient";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

echo $time;

$cnt = 1;

while ($row = $result->fetch\_array()) {

?>

<tbody>

<tr>

<td><?php echo $cnt; ?></td>

<td><?php echo $row['Fname'] . ' ' . $row['Lname']; ?></td>

<td><?php echo $row['PCode']; ?></td>

<td><?php echo $row['Address']; ?></td>

<td><?php echo $row['Phone']; ?></td>

<td>

<a href="view\_single\_patient.php?pat\_id=<?php echo $row['PCode']; ?>" class="badge badge-success"><i class="mdi mdi-eye"></i> View</a>

<a href="addprescription.php?pat\_number=<?php echo $row['PCode']; ?>" class="badge badge-success"><i class="fas fa-highlighter"></i> Get Examination</a>

<a href="addtreatment.php?pat\_number=<?php echo $row['PCode']; ?>" class="badge badge-success"><i class="fas fa-highlighter"></i> Get Treatment</a>

</td>

</tr>

</tbody>

<?php $cnt++; } ?>

<tfoot>

<tr class="active">

<td colspan="8">

<div class="text-right">

<ul style="list-style-type: none" class="pagination-rounded justify-content-end footable-pagination m-t-10 mb-0">

<?php

$result1 = executeQuery($sql1);

$numrows = $result1->num\_rows;

if ($numrows > 0) {

$maxPage = ceil($numrows / $rowsPerPage);

$gap = 5;

$start = $pageNum - $gap;

$end = $pageNum + $gap;

if ($start <= 0) $start = 1;

if ($end >= $maxPage) $end = $maxPage;

if ($self != "manage\_patient.php") {

$self .= "&";

} else {

$self .= '?';

}

$nav = '';

if ($start > 1) {

$nav .= '<li class="footable-page"><a href="'.$self.'page=1">1</a></li>';

$nav .= '<li class="footable-page disable"><span>...</span></li>';

}

for ($page = $start; $page <= $end; $page++) {

if ($page == $pageNum) {

$nav .= '<li class="footable-page active"><a>'.$pageNum.'</a></li>';

} else {

$nav .= '<li class="footable-page"><a href="'.$self.'page='.$page.'">'.$page.'</a></li>';

}

}

if ($end < $maxPage) {

$nav .= '<li class="footable-page"><span>...</span></li>';

$nav .= '<li class="footable-page"><a href="'.$self.'page='.$maxPage.'">'.$maxPage.'</a></li>';

}

$prev = $pageNum > 1 ? '<li><a href="'.$self.'page='.($pageNum-1).'" aria-label="Previous">&lt;</a></li>' : '';

$next = $pageNum < $maxPage ? '<li><a href="'.$self.'page='.($pageNum+1).'" aria-label="Next">&gt;</a></li>' : '';

echo $prev . $nav . $next;

}

?>

</ul>

</div>

</td>

</tr>

</tfoot>

</table>

</div>

**Giao diện của chức năng thêm bệnh nhân**

<div class="content">

<!-- Start Content-->

<div class="container-fluid">

<!-- start page title -->

<div class="row">

<div class="col-12">

<div class="page-title-box">

<div class="page-title-right">

<ol class="breadcrumb m-0">

<li class="breadcrumb-item"><a href="his\_doc\_dashboard.php">Dashboard</a></li>

<li class="breadcrumb-item"><a href="javascript: void(0);">Patients</a></li>

<li class="breadcrumb-item active">Add Patient</li>

</ol>

</div>

<h4 class="page-title">Add Patient Details</h4>

</div>

</div>

</div>

<!-- end page title -->

<!-- Form row -->

<div class="row">

<div class="col-12">

<div class="card">

<div class="card-body">

<h4 class="header-title">Fill all fields</h4>

<!-- Add Patient Form -->

<form method="post" onsubmit="return checkform()">

<div class="form-row">

<div class="form-group col-md-6">

<label for="inputEmail4" class="col-form-label">First Name</label>

<input type="text" required="required" name="pat\_fname" class="form-control" id="inputEmail4" placeholder="Patient's First Name">

</div>

<div class="form-group col-md-6">

<label for="inputPassword4" class="col-form-label">Last Name</label>

<input required="required" type="text" name="pat\_lname" class="form-control" id="inputPassword4" placeholder="Patient's Last Name">

</div>

</div>

<div class="form-row">

<div class="form-group col-md-6">

<label for="inputEmail4" class="col-form-label">Date Of Birth</label>

<input type="date" required="required" name="pat\_dob" class="form-control" id="inputEmail4" placeholder="DD/MM/YYYY">

</div>

<div class="form-group col-md-6">

<label for="inputAddress" class="col-form-label">Address</label>

<input required="required" type="text" class="form-control" name="pat\_addr" id="inputAddress" placeholder="Patient's Address">

</div>

</div>

<div class="form-row">

<div class="form-group col-md-4">

<label for="inputCity" class="col-form-label">Mobile Number</label>

<input required="required" type="text" name="pat\_phone" class="form-control" id="inputCity">

</div>

<div class="form-group col-md-4">

<label for="inputState" class="col-form-label">Patient's Gender</label>

<select id="inputGender" required="required" name="pat\_gender" class="form-control">

<option value="-1">Choose</option>

<option>Male</option>

<option>Female</option>

</select>

</div>

<div class="form-group col-md-2" style="display:none">

<?php

$length = 5;

$patient\_number = substr(str\_shuffle('0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'), 1, $length);

?>

<label for="inputZip" class="col-form-label">Patient Number</label>

<input type="text" name="pat\_number" value="<?php echo $patient\_number; ?>" class="form-control" id="inputZip">

</div>

</div>

<button type="submit" name="add\_patient" class="ladda-button btn btn-primary" data-style="expand-right">Add Patient</button>

</form>

<!-- End Patient Form -->

</div> <!-- end card-body -->

</div> <!-- end card -->

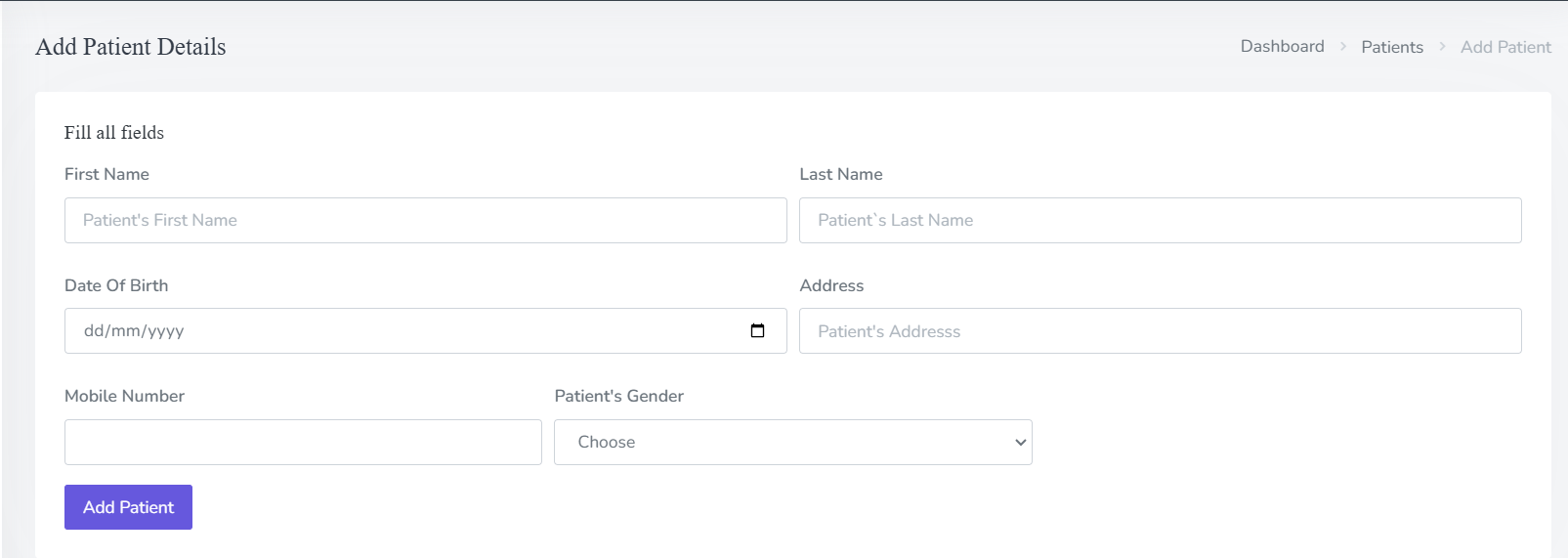
</div> <!-- end col -->

</div>

<!-- end row -->

</div> <!-- container -->

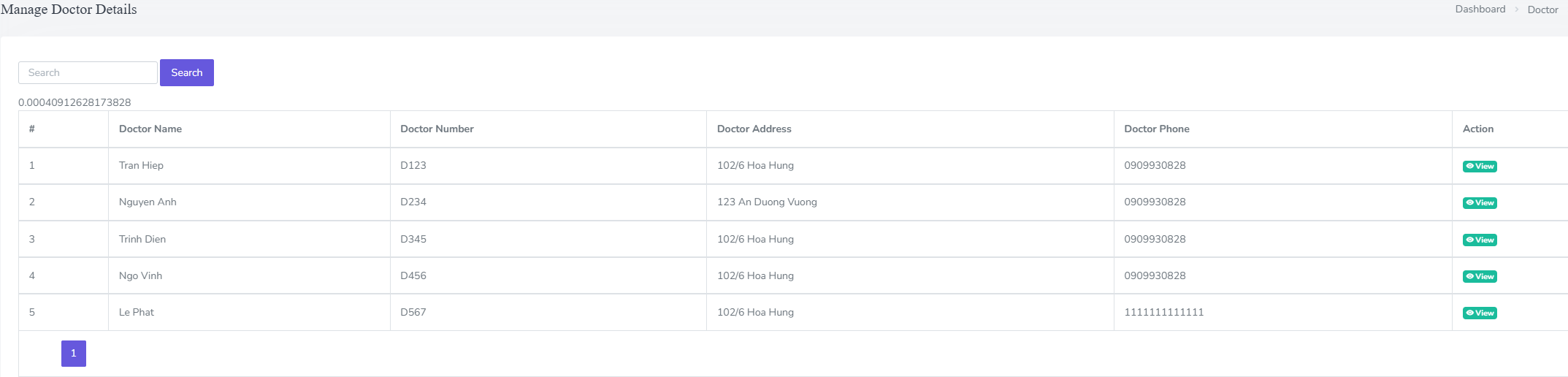
</div> <!-- content -->



Hình 3. 2. Giao diện thêm bệnh nhân

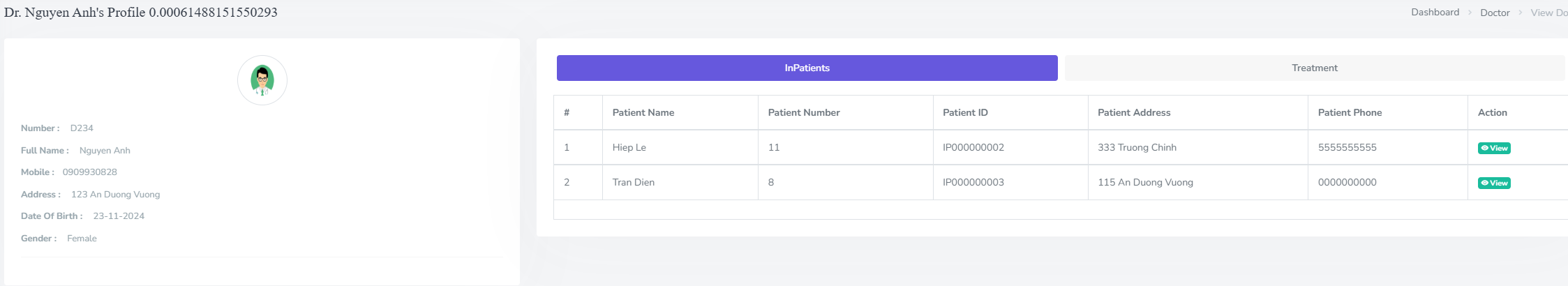
### **3.2.3. Liệt kê tất cả các bệnh nhân được chữa bệnh bởi bác sĩ đó.**

Trước hết, chúng ta cần phải có một mục để quản lý các bác sĩ trong bệnh viện.



Hình 3. 3. Giao diện liệt kê các bác sĩ

Ta sẽ thực hiện bấm vào “View” của từng bác sĩ để liệt kê danh sách các bệnh nhân mà bác sĩ đó đảm nhận.



Hình 3. 4. Giao diện liệt kê các bệnh nhân của bác sĩ

Code giao diện:

<table id="" class="table table-bordered toggle-circle mb-0" data-page-size="7">

<thead>

<tr>

<th>#</th>

<th>Patient Name</th>

<th>Patient Number</th>

<th>Patient ID</th>

<th>Patient Address</th>

<th>Patient Phone</th>

<th>Action</th>

</tr>

</thead>

<?php

/\*

\* Lấy thông tin tất cả bệnh nhân

\*/

$sql = "SELECT \* FROM `treatment` T

JOIN `cure` C ON C.TreatCode = T.TreatCode

JOIN `inpatient` I ON T.IPCode = I.IPCode

JOIN `patient` P ON P.PCode = I.PCode

WHERE C.DECode = '".$pat\_id."'

GROUP BY T.IPCode";

$result = executeQuery($sql);

$cnt = 1;

while ($row = $result->fetch\_array()) {

?>

<tbody>

<tr>

<td><?php echo $cnt; ?></td>

<td><?php echo $row['Fname']; ?> <?php echo $row['Lname']; ?></td>

<td><?php echo $row['PCode']; ?></td>

<td><?php echo $row['IPCode']; ?></td>

<td><?php echo $row['Address']; ?></td>

<td><?php echo $row['Phone']; ?></td>

<td>

<a href="view\_single\_patient.php?pat\_id=<?php echo $row['PCode']; ?>&&pat\_number=<?php echo $row['PCode']; ?>&&pat\_name=<?php echo $row['Fname']; ?>\_<?php echo $row['Lname']; ?>" class="badge badge-success">

<i class="mdi mdi-eye"></i> View

</a>

</td>

</tr>

</tbody>

<?php

$cnt++;

}

?>

<tfoot>

<tr class="active">

<td colspan="8">

<div class="text-right">

<ul class="pagination pagination-rounded justify-content-end footable-pagination m-t-10 mb-0"></ul>

</div>

</td>

</tr>

</tfoot>

</table>

### **3.2.4. Chức năng lập báo cáo về khoản thanh toán.**

 $pres\_id = $\_GET['pres\_id'];

  $ret="SELECT  \* FROM `examination` E JOIN `outpatient` O ON E.OPCode = O.OPCode JOIN `patient` P ON P.PCode = O.PCode  WHERE ExamCode = '".$pres\_id."'";

   $res = executeQuery($ret);

**Giao diện của chức năng báo cáo**

<div class="content-page">

<div class="content">

<!-- Start Content-->

<div class="container-fluid">

<!-- start page title -->

<div class="row">

<div class="col-12">

<div class="page-title-box">

<div class="page-title-right">

<ol class="breadcrumb m-0">

<li class="breadcrumb-item"><a href="his\_doc\_dashboard.php">Dashboard</a></li>

<li class="breadcrumb-item"><a href="javascript: void(0);">Pharmaceuticals</a></li>

<li class="breadcrumb-item active">View Prescriptions</li>

</ol>

</div>

<h4 class="page-title">#<?php echo $row->ExamCode; ?></h4>

</div>

</div>

</div>

<!-- end page title -->

<div class="row">

<div class="col-12">

<div class="card-box">

<div class="row">

<div class="col-xl-5">

<div class="tab-content pt-0">

<div class="tab-pane active show" id="product-1-item">

<img src="assets/images/users/patient.png" alt="" class="img-fluid mx-auto d-block rounded">

</div>

</div>

</div> <!-- end col -->

<div class="col-xl-7">

<div class="pl-xl-3 mt-3 mt-xl-0">

<h2 class="mb-3">Name : <?php echo $row->Fname . " " . $row->Lname; ?></h2>

<hr>

<h4 class="text-primary"><b class="text-dark"> Patient Number : </b><?php echo $row->PCode; ?></h4>

<hr>

<h4 class="text-primary"><b class="text-dark"> Patient ID : </b><?php echo $row->OPCode; ?></h4>

<hr>

<h4 class="text-primary"><b class="text-dark">Date Of Birth : </b><?php echo date("d-m-Y", strtotime($row->DoB)); ?></h4>

<hr>

<h4 class="text-primary"><b class="text-dark">Patient Category : </b>Outpatient</h4>

<hr>

<h4 class="text-primary"><b class="text-dark">Examination Date :</b><?php echo date("d-m-Y", strtotime($row->ExamDate)); ?></h4>

<hr>

<h4 class="text-primary"><b class="text-dark">Diagnosis : </b><?php echo $row->Diagnosis; ?></h4>

<hr>

<h4 class="text-primary"><b class="text-dark">Date of Re-examination : </b><?php

echo ($row->NextExDate == "0000-00-00") ? "NO" : date("d-m-Y", strtotime($row->NextExDate)); ?></h4>

<hr>

<h4 class="text-primary"><b class="text-dark">Examination Date :</b><?php echo $row1['Fname'] . " " . $row1['Lname'] . " - " . $row->DECode; ?></h4>

<hr>

<h2 class="align-centre">Prescription</h2>

</div>

</div> <!-- end col -->

</div>

<!-- end row -->

<div class="table-responsive mt-4">

<table class="table table-bordered table-centered mb-0">

<thead class="thead-light">

<tr>

<th>Medication</th>

<th>Price</th>

<th>Quantity</th>

<th>Total</th>

</tr>

</thead>

<tbody>

<?php

$sql = "SELECT \* FROM `exam\_use` E JOIN `medication` M ON E.MCode = M.MCode WHERE E.ExamCode = '" . $pres\_id . "'";

$result = executeQuery($sql);

while ($row1 = $result->fetch\_array()) {

?>

<tr>

<td><?php echo $row1['Name']; ?></td>

<td><?php echo $row1['price']; ?></td>

<td><?php echo $row1['quantity']; ?></td>

<td>$<?php echo ($row1['quantity'] \* $row1['price']); ?></td>

</tr>

<?php }

$sql1 = "SELECT \* FROM `examination` WHERE `ExamCode` = '" . $pres\_id . "'";

$res1 = executeQuery($sql1);

$row2 = $res1->fetch\_array();

?>

<tr>

<td colspan="3"><h2>Total: </h2></td>

<td><h2>$<?php echo $row2['Fee']; ?></h2></td>

</tr>

</tbody>

</table>

</div>

</div> <!-- end card -->

</div> <!-- end col -->

</div>

<!-- end row -->

</div> <!-- container -->

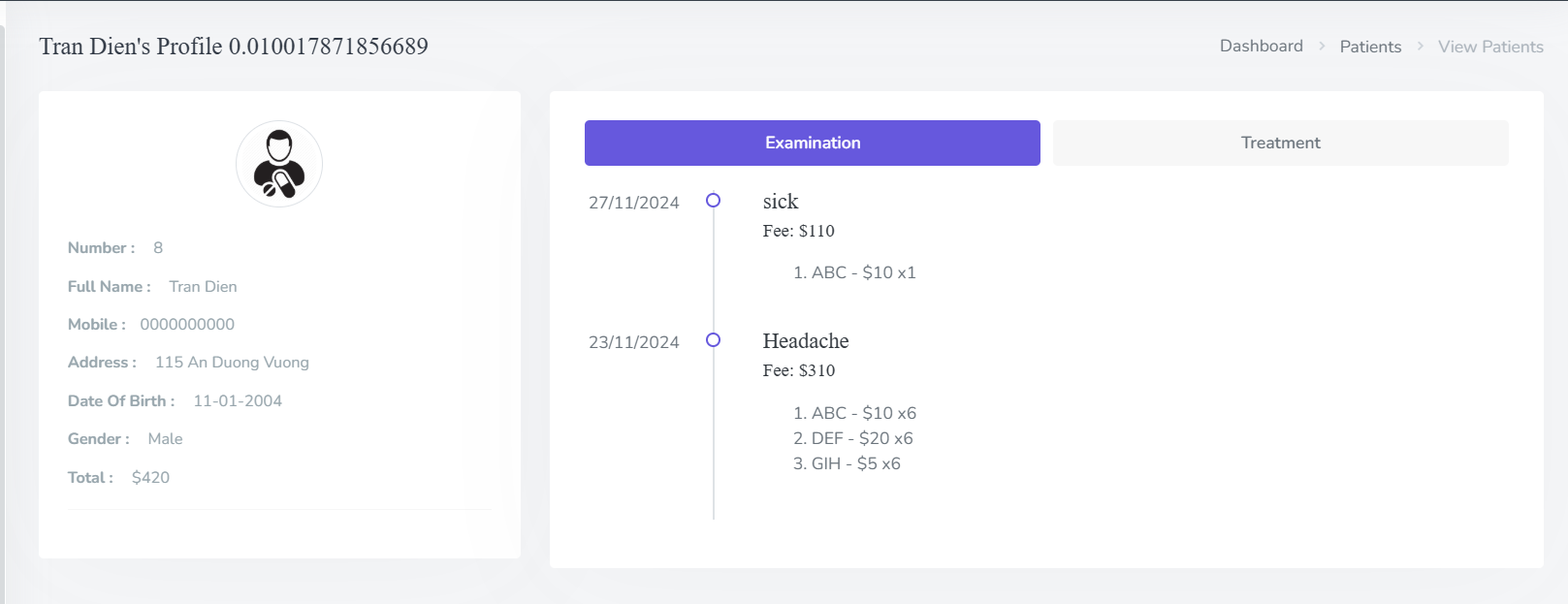
</div> <!-- content -->

<!-- Footer Start -->

<?php include('assets/inc/footer.php'); ?>

<!-- end Footer -->

</div>



Hình 3. 5. Giao diện báo cáo các khoản thu phí của bệnh nhân

### **3.2.5. Chứng minh việc sử dụng chỉ mục đạt hiệu quả.**

Chứng minh bằng thời gian chạy giữa truy vấn không có chỉ mục và có chỉ mục sẽ mang lại sự hiệu quả. Đơn vị thời gian sẽ được tính bằng m/s (mili/giây)

if ($flag) {

$sql = "SELECT \* FROM `patient` WHERE Phone ='" . $key . "' OR PCode = '" . $key . "' OR Fname ='" . $key . "' OR Lname ='" . $key . "'";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

if ($result->num\_rows < 1) {

$sql = "SELECT \* FROM `patient` WHERE PCode = '" . $key . "' OR Fname LIKE '%" . $key . "%' OR Lname LIKE '%" . $key . "%' OR CONCAT(Fname,' ',Lname) LIKE '%" . $key . "%' ";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

$time = microtime(true) - $start;

}

} else {

$sql = "SELECT \* FROM patient";

$sql1 = $sql;

$sql .= " LIMIT $offset, $rowsPerPage";

$start = microtime(true);

$result = executeQuery($sql);

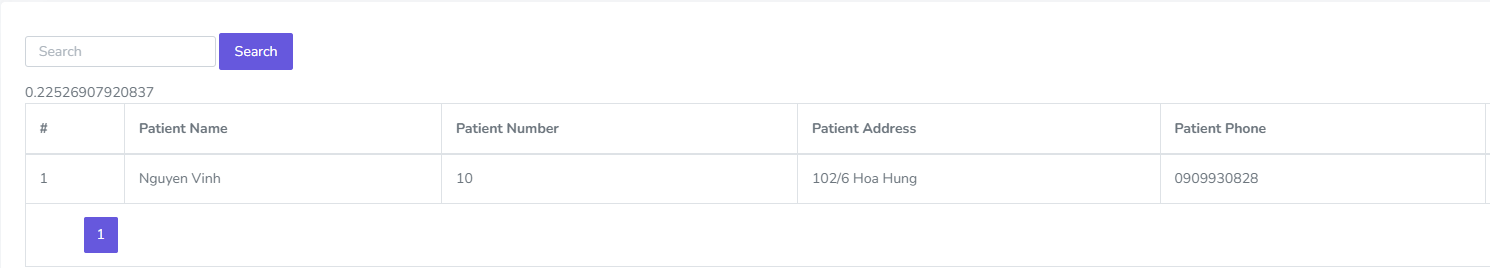
$time = microtime(true) - $start;

}

// SQL code to get the time taken to execute the query

echo $time;

Thời gian tìm kiếm bệnh nhân với tiêu chí là số điện thoại khi chưa gắn chỉ mục.



Hình 3. 6. Tìm kiếm không có chỉ mục

Thời gian tìm kiếm bệnh nhân với tiêu chí là số điện thoại khi đã gắn chỉ mục.



Hình 3. 7. Tìm kiếm có chỉ mục

Chức năng chỉ mục được kiểm định thông qua sự chênh lệch thời gian của cả hai kết quả.

### **3.2.6. Giải quyết một trường hợp bảo mật trong cơ sở dữ liệu.**

Nhóm sẽ tiến hành giải quyết vấn đề bảo mật thông qua Kiểm soát truy cập tùy ý (DAC).

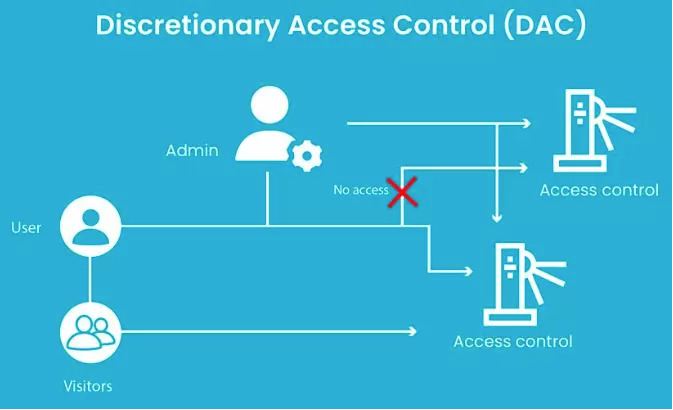
**Nguyên lý hoạt động về DAC (kiểm soát truy cập tùy chọn)**

- Chủ sở hữu sẽ quyết định quyền truy cập đối với tài nguyên.

- Trong hệ thống bệnh viện, người quản trị viên là người có quyền hạn cao nhất, ví dụ như có thể truy cập được toàn bộ mà không bị giới hạn bởi một chức năng nào.

- Tất nhiên, chủ sở hữu cũng có thể cấp quyền truy cập tới chức năng mà mình có thể dùng được cho một người khác.

- Và quyền hạn tối đa mà chủ sở hữu có thể cấp cho người khác là bằng với quyền hạn của chủ sở hữu.



Hình 3. 8. Sơ đồ bảo mật theo DAC (Discretionary Access Control) [3]

**Áp dụng DAC (kiểm soát truy cập tùy chọn) vào đề tài**

Chức năng cấp quyền trong cơ sở dữ liệu

create user 'dean'@'localhost'

identified by 'mypassword';

create user 'doctor'@'localhost'

identified by 'mypassword';

grant select, insert, update, delete

on hospital.\*

to 'dean'@'localhost';

grant select , insert

on hospital.patient

to 'doctor'@'localhost';

grant select

on hospital.\*

to 'doctor'@'localhost';

**Phân biệt giữa hai loại người dùng:**

**1) Tạo kết nối CSDL (Cơ sở dữ liệu) với người dùng là “doctor”.**



Hình 3. 9. Quyền truy cập của doctor

Câu lệnh cấp quyền cho “doctor”:

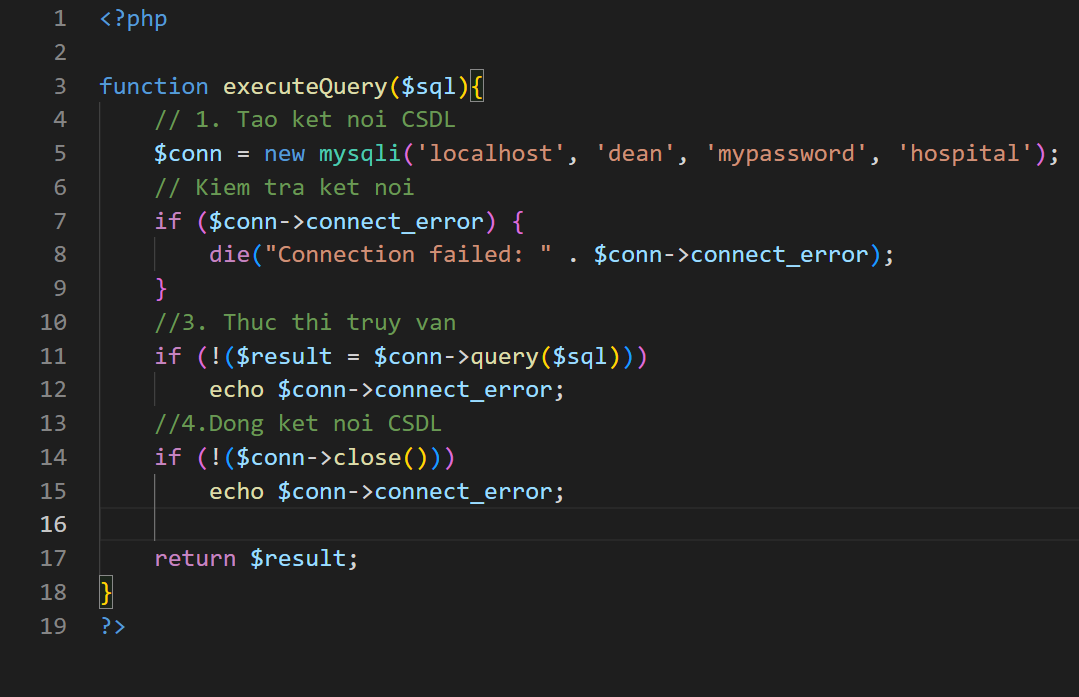
grant select , insert

on hospital.patient

to 'doctor'@'localhost';

**Ý nghĩa:** Tức là tài khoản người dùng thuộc doctor thì không thể làm bất cứ điều gì ngoại trừ có thể thêm và xem thông tin của bệnh nhân.

**2) Tạo kết nối CSDL (Cơ sở dữ liệu) với người dùng là “dean”.**



Hình 3. 10. Quyền truy cập của Dean

Câu lệnh cấp quyền cho “dean”:

grant select, insert, update, delete

on hospital.\*

to 'dean'@'localhost';

**Ý nghĩa:** Tức là tài khoản người dùng thuộc dean thì có thể xem, thêm, xóa, sửa tất cả.

# **Kết luận**

Chương 1 sẽ tóm tắt và giới thiệu về đề tài, đồng thời cũng cho thấy các khó khăn đang gặp phải. Ngoài ra, ở chương này còn cho thấy các khảo sát về giao diện của trang web cơ sở dữ liệu về bệnh viện.

Chương 2 sẽ nêu các thực thể và thuộc tính của chúng. Nhóm sẽ tiến hành ánh xạ và chuyển khái niệm mô hình thực thể liên kết sang mô hình quan hệ. Ngoài ra, nhóm cũng đã có những phân tích về phụ thuộc hàm của chúng. Khi bắt đầu triển khai lập các mối quan hệ giữa các thực thể, ta cần phải kiểm tra tính rèn buộc toàn vẹn của mỗi phần.

Chương 3 sẽ tiến hành cài đặt các chức năng như tìm kiếm, thêm bệnh nhân. Ngoài ra, hiển thị các bệnh nhân được bác sĩ chữa bệnh. Nhóm cũng đã thực hiện chứng minh hiệu quả của chỉ mục trong tìm kiếm và giải quyết vấn đề bảo mật trong cơ sở dữ liệu.

# **Tài liệu tham khảo**

Tiếng Việt

|  |  |
| --- | --- |
| [1] | N. T. Dũng, Nhập môn cơ sở dữ liệu, Huế: Đại học Huế, 2011. |
| [2] | bmdSolution, https://bmdsolutions.vn/phan-mem-quan-ly-benh-vien/. |
| [3] | D. A. C. (DAC), https://viblo.asia/p/kiem-soat-truy-cap-an-toan-ByEZkArq5Q0. |