# **LỜI MỞ ĐẦU**

Ngày nay việc ứng dụng công nghệ thông tin đã trở nên phổ biến trong hầu hết mọi cơ quan, doanh nghiệp, trường học, y tế đặc biệt là việc áp dụng các giải pháp tin học trong công tác quản lý hoặc để xây dựng những hệ thống phục vụ cho một mục đích nào đó.

Công nghệ thông tin là một trong những lĩnh vực quan trọng hàng đầu thế giới. Việc thành lập các công ty, ngân hàng, các tổ chức lớn, cho đến những thành phần nhỏ hơn là các doanh nghiệp, cửa hàng... ít hay nhiều đầu cần đến công nghệ thông tin như là một thành phần không thể thiếu trong việc quản lý, xây dựng và phát triển về sau này. Với sự hỗ trợ đắc lực từ công nghệ thông tin, tốc độ phát triển của xã hội đã tăng lên rất nhiều lần.

Trong ít năm trở lại đây, với tốc độ phát triển như vũ bão, công nghệ thông tin đang dần làm cho cuộc sống của con người trở nên thú vị và đơn giản hơn. Tuy rằng công việc tự học là vô cùng quan trọng, nhưng như thế thôi là chưa đủ. Việc kết hợp với các yếu tố giảng dạy từ những người khác thì tốc độ tiếp thu được kiến thức của chúng ta sẽ tang lên gấp nhiều lần và giúp chúng ta tạo ra những tư duy, tính sáng tạo và kế thừa phát huy những ưu điểm của người giảng dạy. Chính vì vậy để bắt kịp với tốc độ phát triển của xã hội, những kiến thức có được trên giảng đường là vô cùng quan trọng vơi những sinh viên chúng em.

Chúng em xin chân thành cảm ơn các thầy cô người truyền lửa, kiến thức và sự tận tình giảng dạy cho chúng em trong suốt thời gian học tập vừa qua. Nhờ những kiến thức đó mà chúng em có thể xây dựng và hoàn thiện được mô hình web ý tưởng trên những chặng đường cuối của hành trình sinh viên này. Tuy đã có nhiều cố gắng nhưng trong quá trình làm để tài không thể tránh được những sai sót.

Chúng em rất mong nhận được các ý kiến đóng góp từ các thầy cô và bạn bè để chúng em rút kinh nghiệm thực hiện tốt hơn ở những lần sau.

# **MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc11361348)

[MỤC LỤC 2](#_Toc11361355)

[CHƯƠNG I: KHẢO SÁT HỆ THỐNG 4](#_Toc11361356)

[1. Mô tả về môi trường hoạt động: 4](#_Toc11361357)

[2. Khảo sát bài toán 4](#_Toc11361358)

[3. Ưu, nhược điểm của hệ thống cũ: 6](#_Toc11361362)

[4. Yêu cầu của đề tài: 6](#_Toc11361365)

[5. Công cụ lập trình: 6](#_Toc11361366)

[CHƯƠNG II: PHÂN TÍCH HỆ THỐNG 11](#_Toc11361369)

[1. Phân tích chức năng hệ thống: 11](#_Toc11361370)

[1.1. Sơ đồ phân rã chức năng: 11](#_Toc11361371)

[1.2. Sơ đồ mức khung cảnh ( mức 0 ): 12](#_Toc11361372)

[1.3. Sơ đồ mức đỉnh (mức 1): 13](#_Toc11361373)

[1.4. Sơ đồ mức dưới đỉnh (mức 2): 14](#_Toc11361374)

.............................................................

[2. Phân tích dữ liệu hệ thống: 18](#_Toc11361375)

[CHƯƠNG III: THIẾT KẾ HỆ THỐNG 19](#_Toc11361376)

[1. Thiết kế cơ sở dữ liệu: 19](#_Toc11361377)

[2. Thiết kế giao diện: 30](#_Toc11361378)

[CHƯƠNG IV: CÀI ĐẶT VÀ CHẠY THỬ 38](#_Toc11361382)

[1. Cài đặt: 38](#_Toc11361383)

[1.1. Các công cụ cần cài đặt: 38](#_Toc11361384)

[1.2. Chạy các thao tác để cài đặt chương trình: 38](#_Toc11361385)

[2. Chạy thử: 38](#_Toc11361386)

[KẾT LUẬN 39](#_Toc11361387)

[TÀI LIỆU THAM KHẢO 40](#_Toc11361388)

[**CHƯƠNG I: KHẢO SÁT HỆ THỐNG**](#_Toc11361356)

* 1. **Mô tả về môi trường hoạt động:**
* Phục vụ cho mục đích khám chữa bệnh
* Áp dụng cho nhu cầu khám chữa bệnh ở 1 bệnh bệnh viện
* Phục vụ cho 4 đối tượng chính : Quản lí bệnh viện, bác sỹ khám bệnh, bác sỹ xét nghiệm ,bệnh nhân
  1. **Khảo sát bài toán**

**-** Ở bệnh viện chấn thương chỉnh hình ở Tp Hồ Chí Minh quá trình đăng kí khám chữa bệnh online được thực hiện như sau:

**+** Bệnh nhân có thể là cá nhân, bác sĩ, nhân viên trong bệnh viện. Mỗi

bệnh nhân trước khi đăng ki khám bệnh phải có một tài khoản

riêng.

+ Khi đăng kí bệnh nhân có thể có chọn ngày và khung giờ khám

+ Sau khi nhận được phiếu hẹn thì công cuộc khám bệnh sẽ diễn ra

theo quy trình bình thường trong bệnh viện.

**1.3 Ưu, nhược điểm của hệ thống cũ:**

- Ưu điểm   
 + Giải quyết được bài toán quản lý bệnh viện, dễ dàng đáp ứng nhu

cầu khám bệnh tức thời của bệnh nhân.

+ Bệnh nhân không phải xếp hàng dài, chen lấn nhau mà sẽ được

cấp số và gọi khách hàng vào quầy giao dịch theo thứ tự

(vào trước –ra trước ).

+ Bệnh nhân được phục vụ theo thứ tự, ai đến trước sẽ được khám trước đảm bảo sự công bằng.

- Nhược điểm :

+ Chưa giải quyết được tình trạng quá tải bệnh nhân gây nên nhiều áp lực lớn cho việc công các của đội ngũ nhân viên y tế.

+ Chưa đẩy được bác sỹ tự động, kết quả trả về còn sơ sài.

* 1. **Yêu cầu của đề tài:**
* Xây dựng phầm mềm dựa trên nhu cầu đăng kí khám chữa bệnh
* Xây dựng chức năng đăng kí, đăng nhập cho 4 tác nhân chính
* Cập nhật danh mục : người dùng, quyền, phòng ban, khung giờ khám,..
* Tìm kiếm : kết quả, bệnh nhân, bác sĩ, phong ban, phiếu hẹn,…
* Cập nhật phiếu đăng kí, phiếu hẹn,….
* Thống kê bác sĩ, bệnh nhân, phiếu hẹn,….

**1.5 Công cụ lập trình:**

- Ngôn ngữ lập trình: C#,

- Framework: ASP.NET Core 5.0, Knockoutjs,

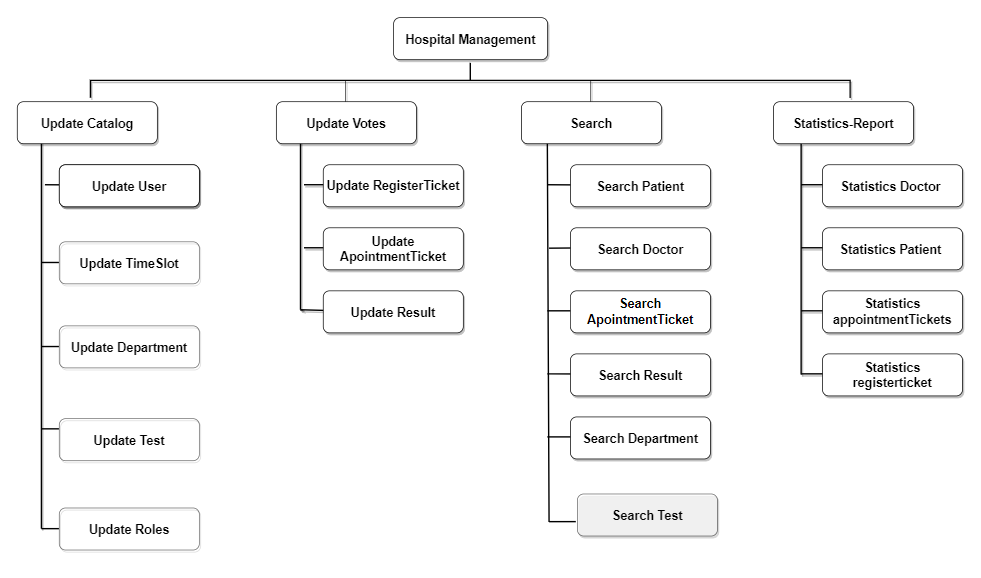
- Mô hình: Backend: Rest full API

- Frontend: mô hình MVC

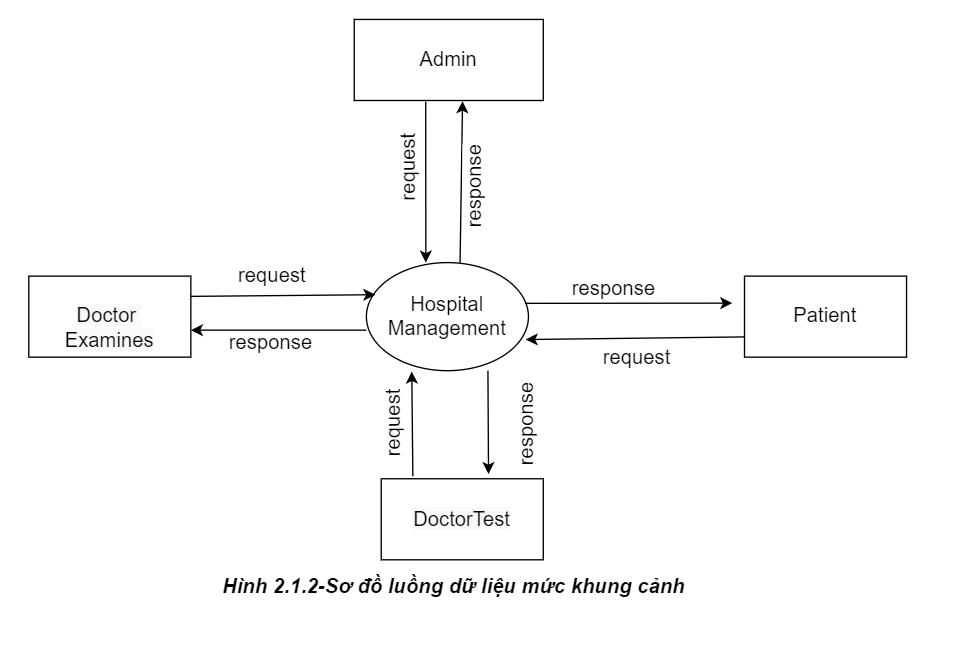
- Cơ sở dữ liệu : SQL Server

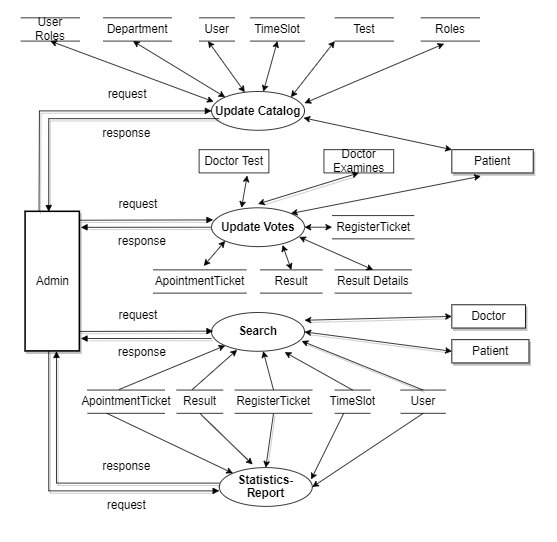
**CHƯƠNG II: PHÂN TÍCH HỆ THỐNG**

**2.1 Phân tích chức năng hệ thống:**

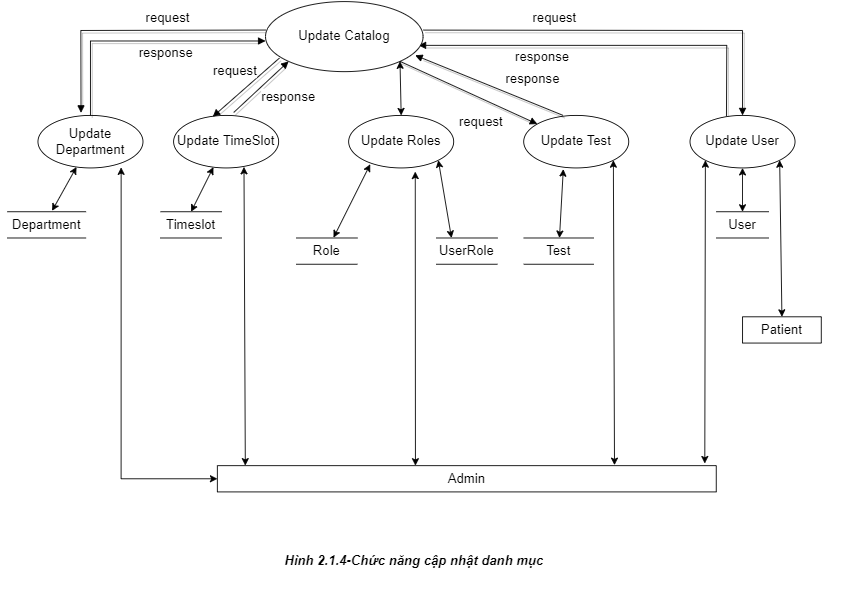
****

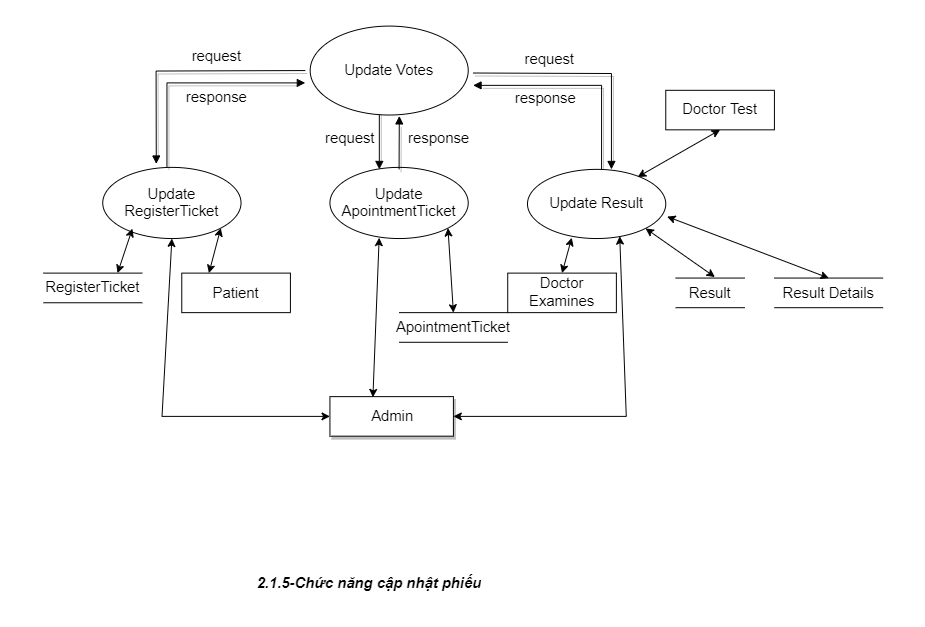
***Hình 2.1.1 -Sơ đồ phân cấp chức năng***

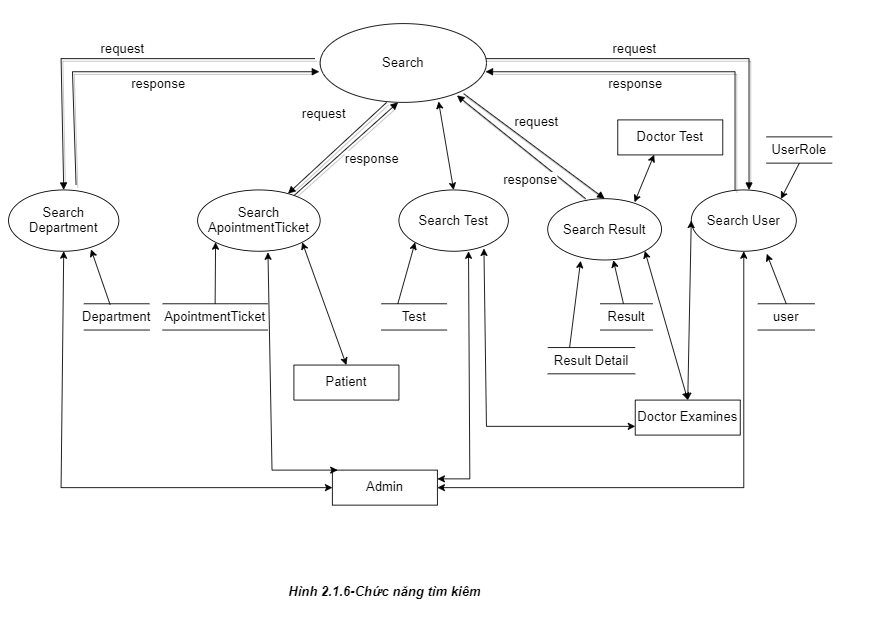
****

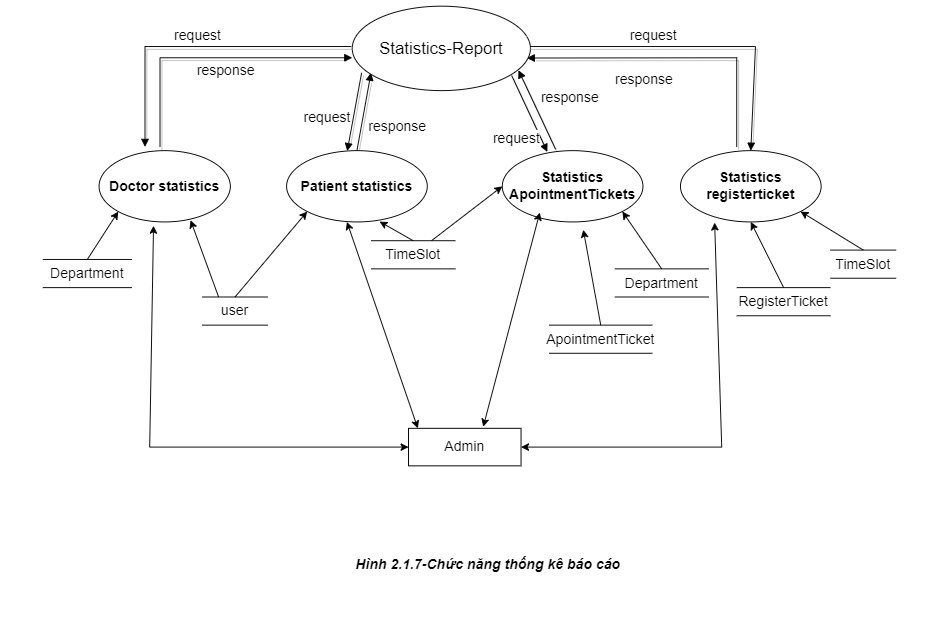
****

***Hình 2.1.3 -Sơ đồ mức đỉnh***

****

****

****

****

**2.2 Phân tích dữ liệu hệ thống**

* + Các dữ liệu đầu vào gồm các bảng sau:
* Test(Id, Name, IdDoctor, Description, Datecreate)
* AspNetUserLogins(Userid, LoginProvider, ProviderKey,

ProviderDisplayName)

* AspNetUserClaims(Id, UserId, ClameType, ClaimValue);
* AspNetRoles(Id, Name, NomalzedValue, ConcurrencyStamp)
* AspNetUserRole(UserId, RoleId)
* AspNetRoleClaims(Id, RoleId, ClameType, ClaimValue);
* Department(Id, Name, Decription, Datecreate, Deleted)
* AspNetUsers(Id, UserName, FirstName, LastName, IdTest,

IdDepartment, Deleted, Email, NormalizedUserName,

NormalizedEmail, EmailConfirmed, PhoneNumber,

PasswordHash, SecurityStamp, ConcurrencyStamp,

PhoneNumber, PhoneNumberConfirmed, LockoutEnd,

TwoFactorEnabled, LockoutEnabled, AccessFailedCount)

* AspNetUserTokens(UserId, LoginProvider, Name, Value)
* Result(Id, IdApointmentTicket, Diagnostic, TheapyRegiment,

DateCreate, Deleted)

* ResultDetails(Idresults, IdDoctortest, DateCreate, DateUpdate, Urlfile,

Diagnostic)

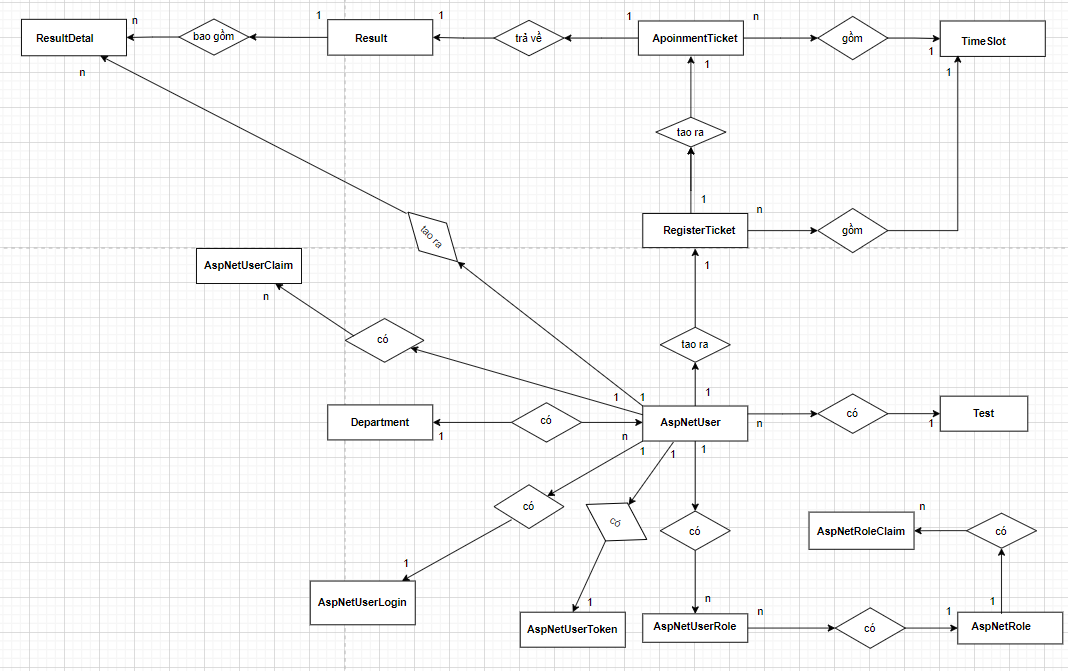
* ApointmentTicket(Id, Idtimemeet, IdRegisterTicket, IdDoctor,

DateMeet, DateCreate, Status, Desciption, Deleted)

* RegisterTicket(Id, IdPatient, IdTimeMeet, Symptom, DateMeet,

DateCreate, Deleted, Status)

* Timeslot(Id, Description, DateCreate, Deleted, Max)
  + Sơ đồ thực thể liên kết

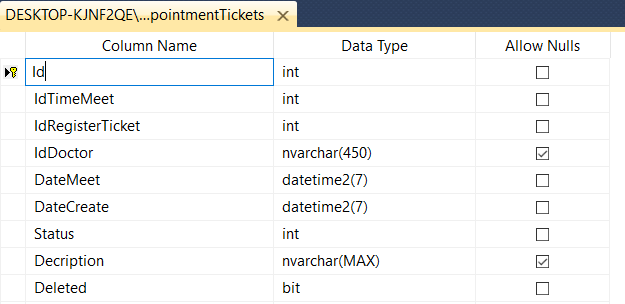


**CHƯƠNG III: THIẾT KẾ HỆ THỐNG**

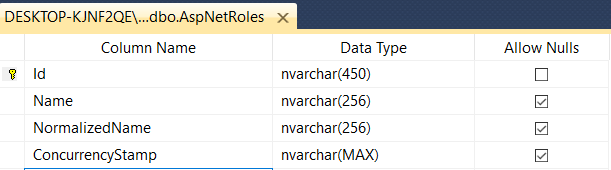
**3.1 Thiết kế cơ sở dữ liệu:**

**3.1.1 Thiết kế cơ sở dữ liệu:**

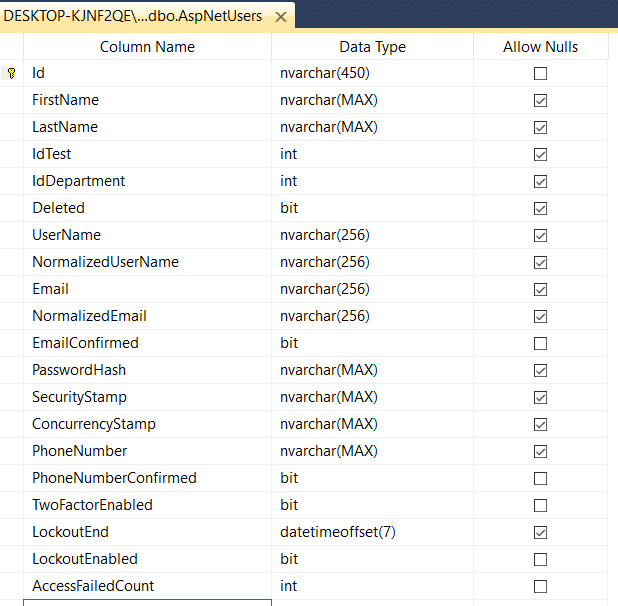
* Bảng ApointmentTicket



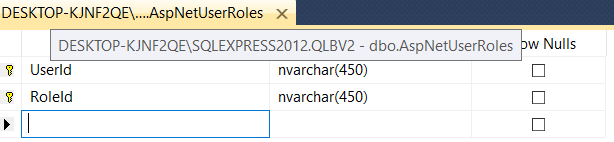
* Bảng AspNetRole



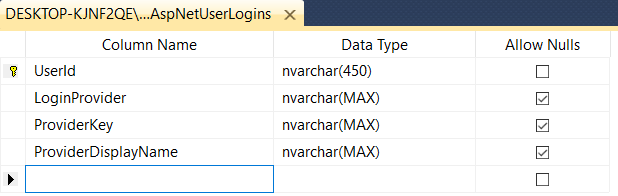
* Bảng AspNetUser



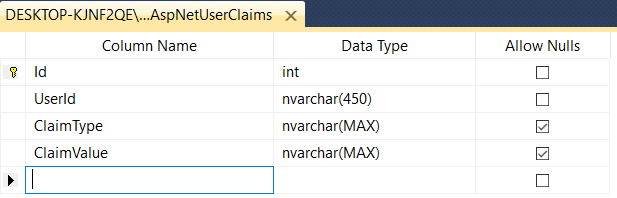
* Bảng AspNetUserRole



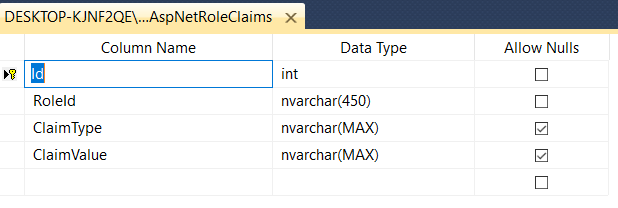
* Bảng AspNetUserLogin



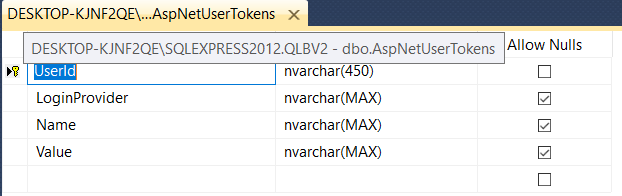
* Bảng AspNetUserClaim



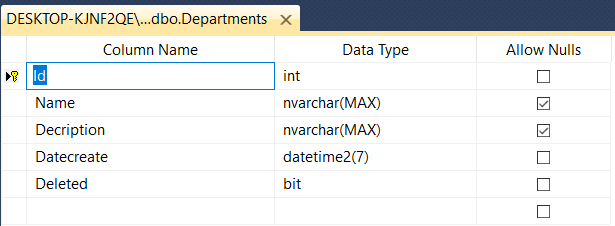
* Bảng AspNetRoleClaim



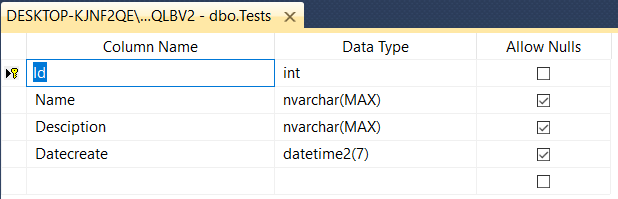
* Bảng AspNetUserToken



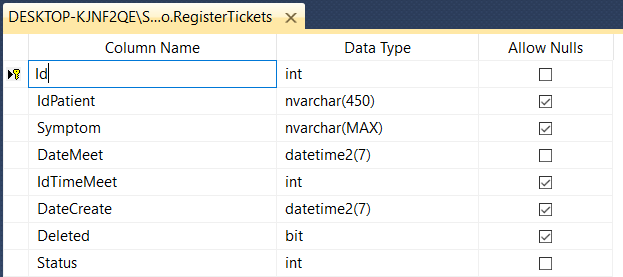
* Bảng Department



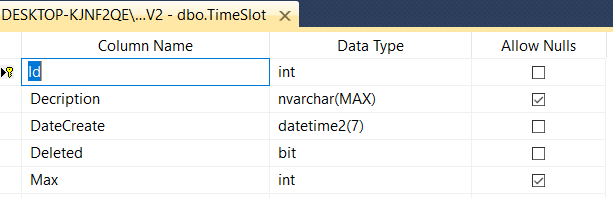
* Bảng Test



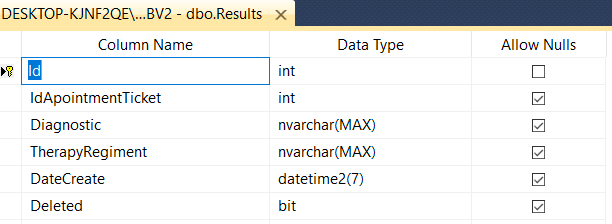
* Bảng RegisterTicket



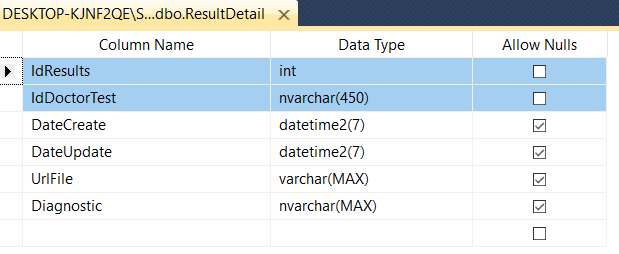
* Bảng TimeSlot



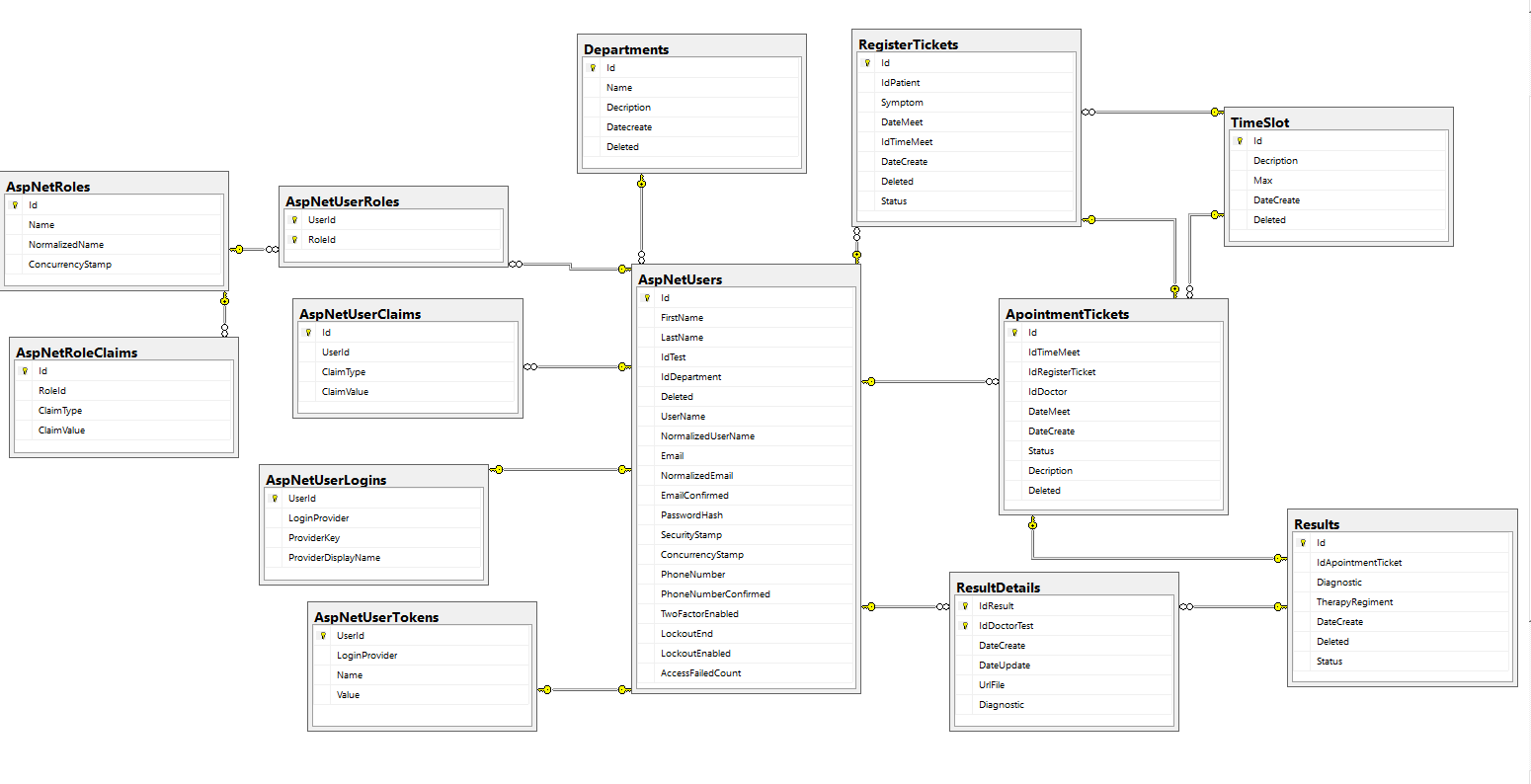
* Bảng Result



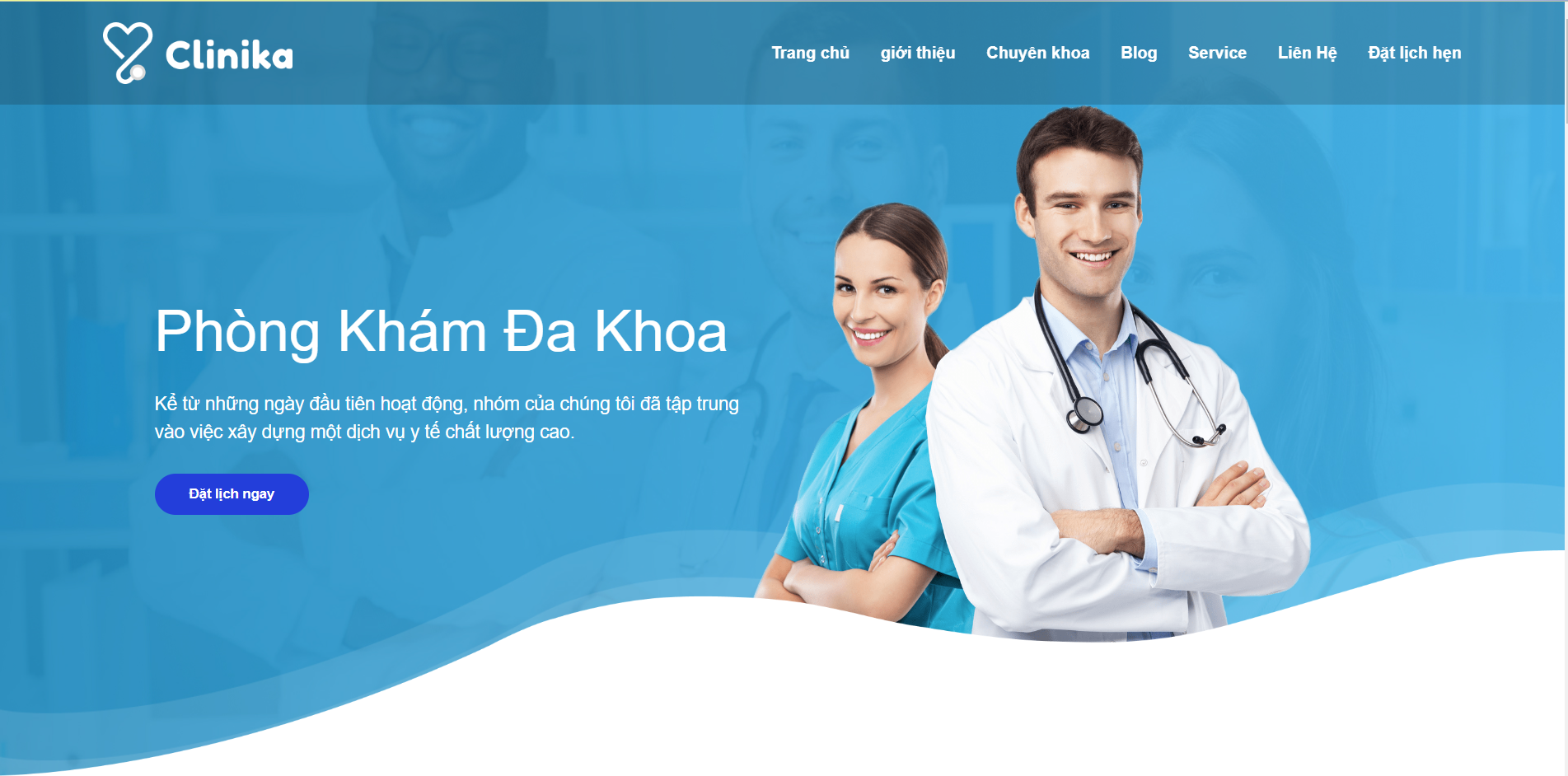
* Bảng ResultDetail

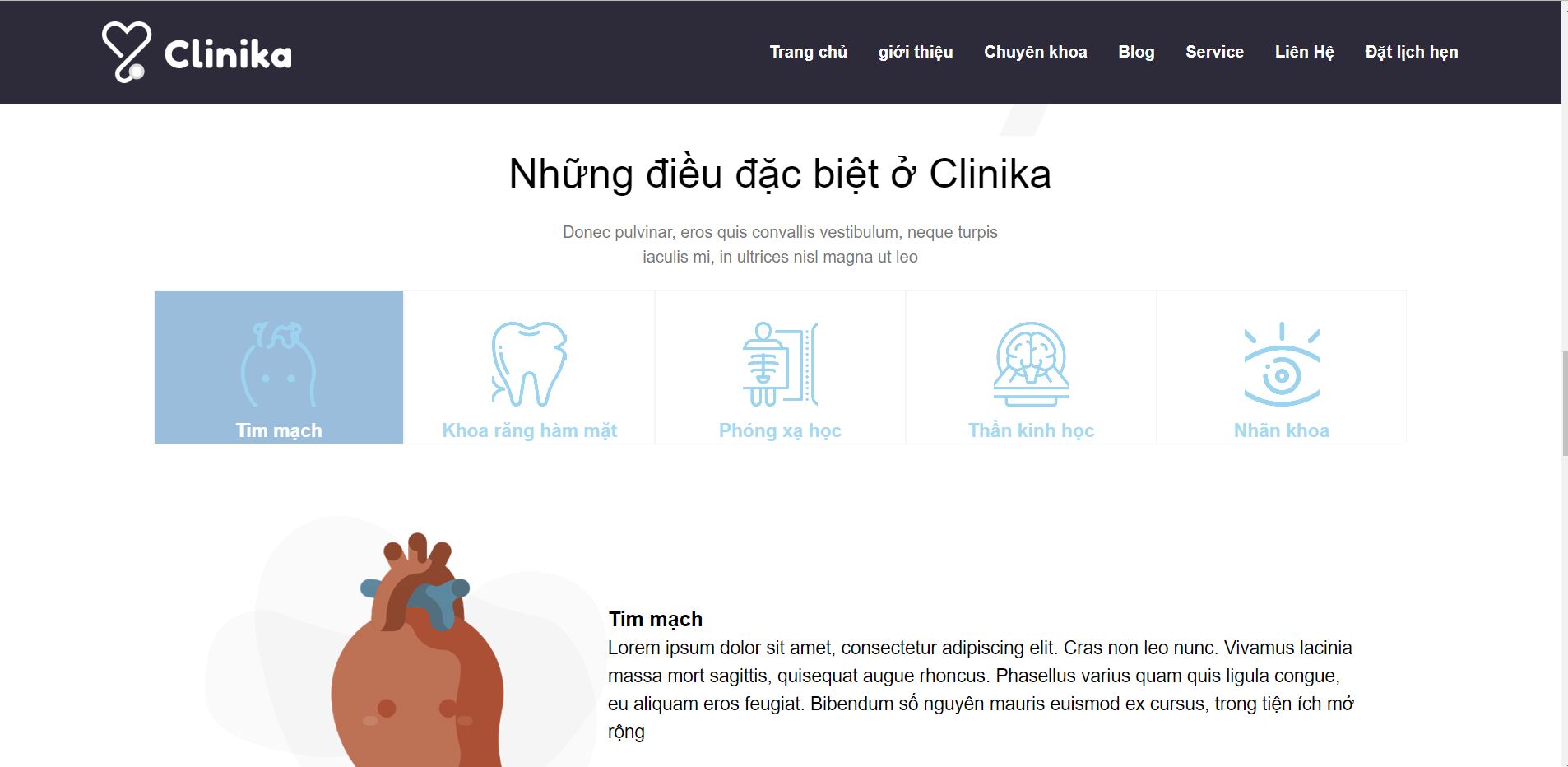


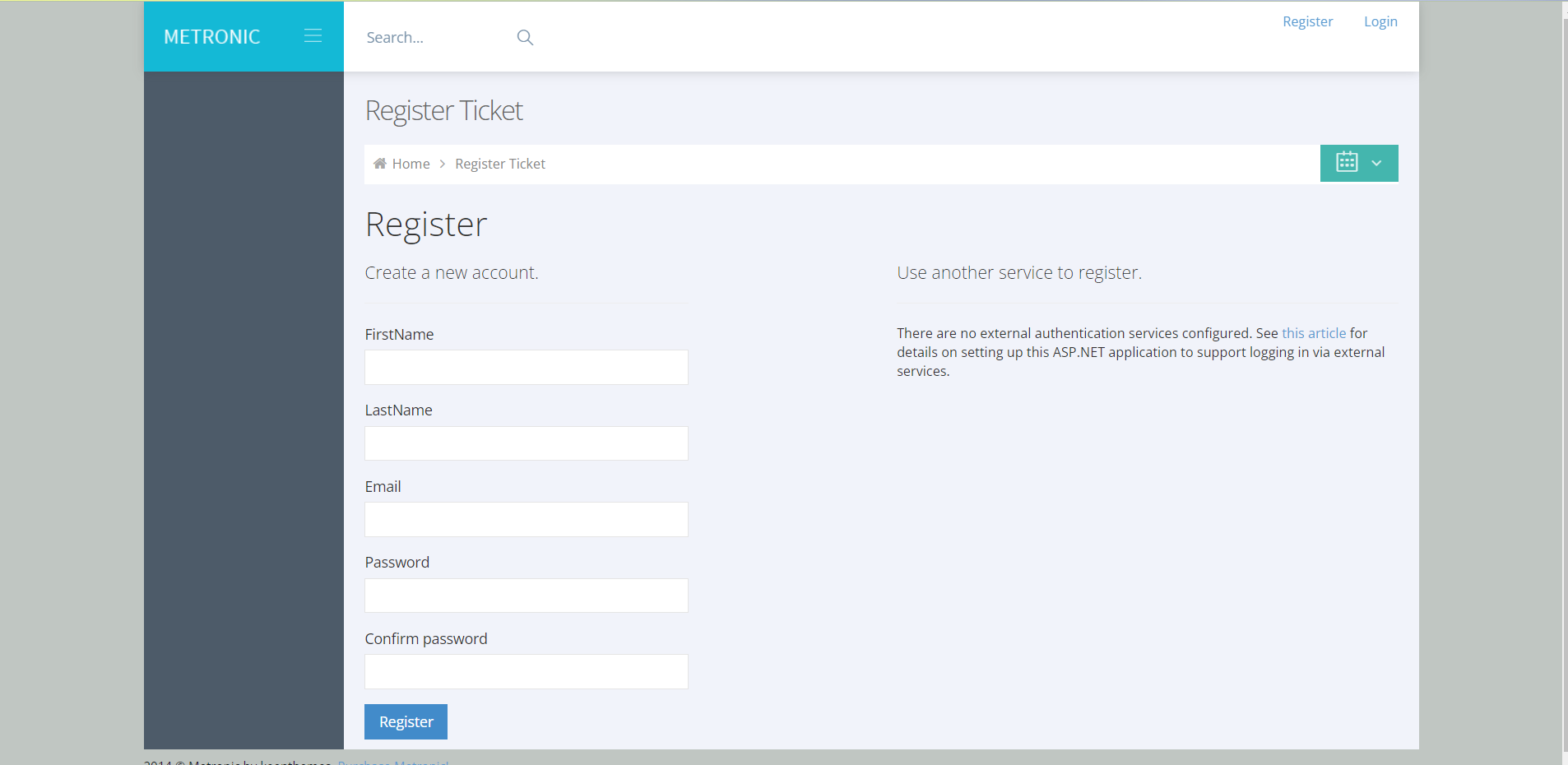
**3.1.2 Sơ đồ quan hệ**

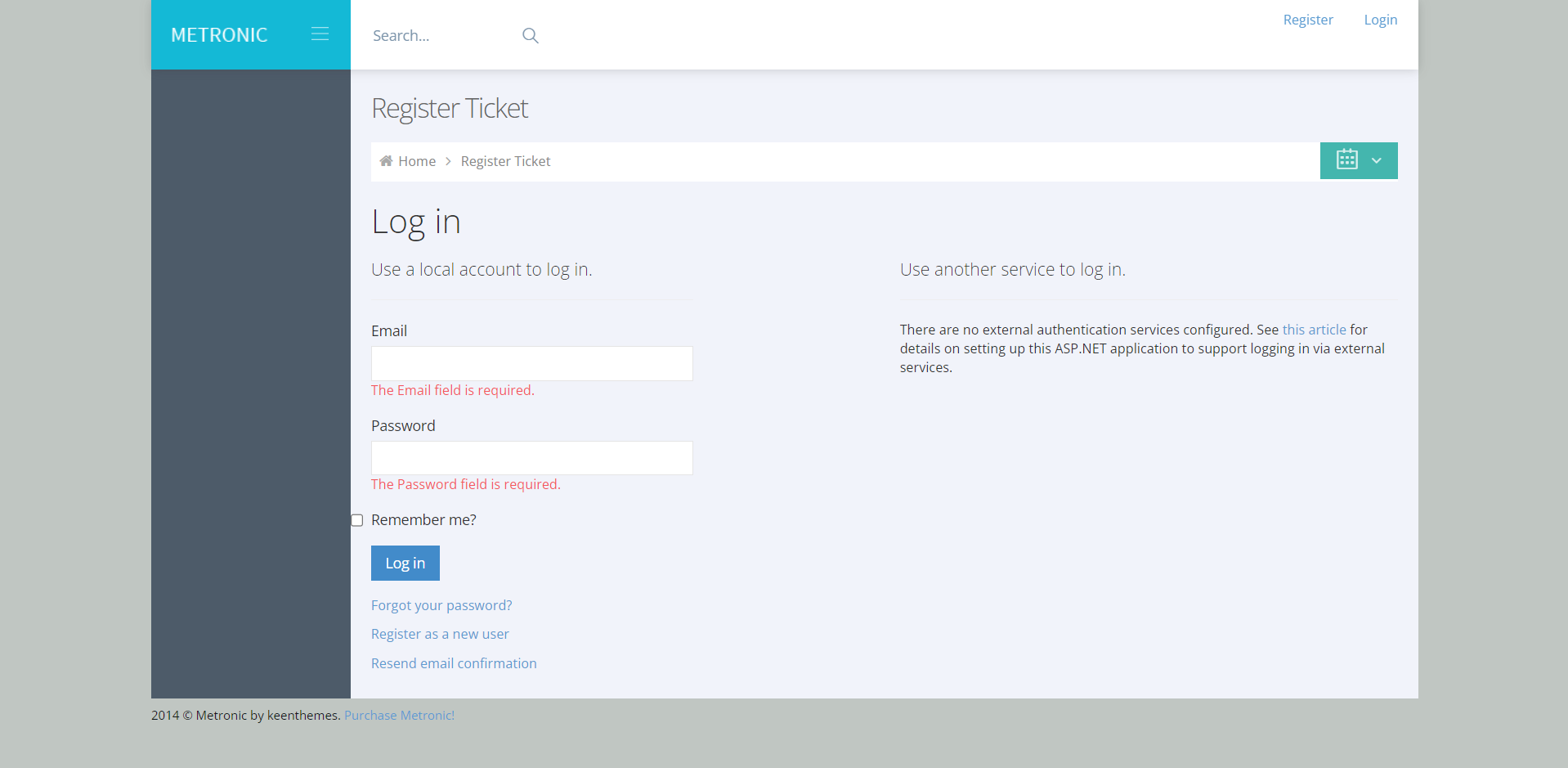
****

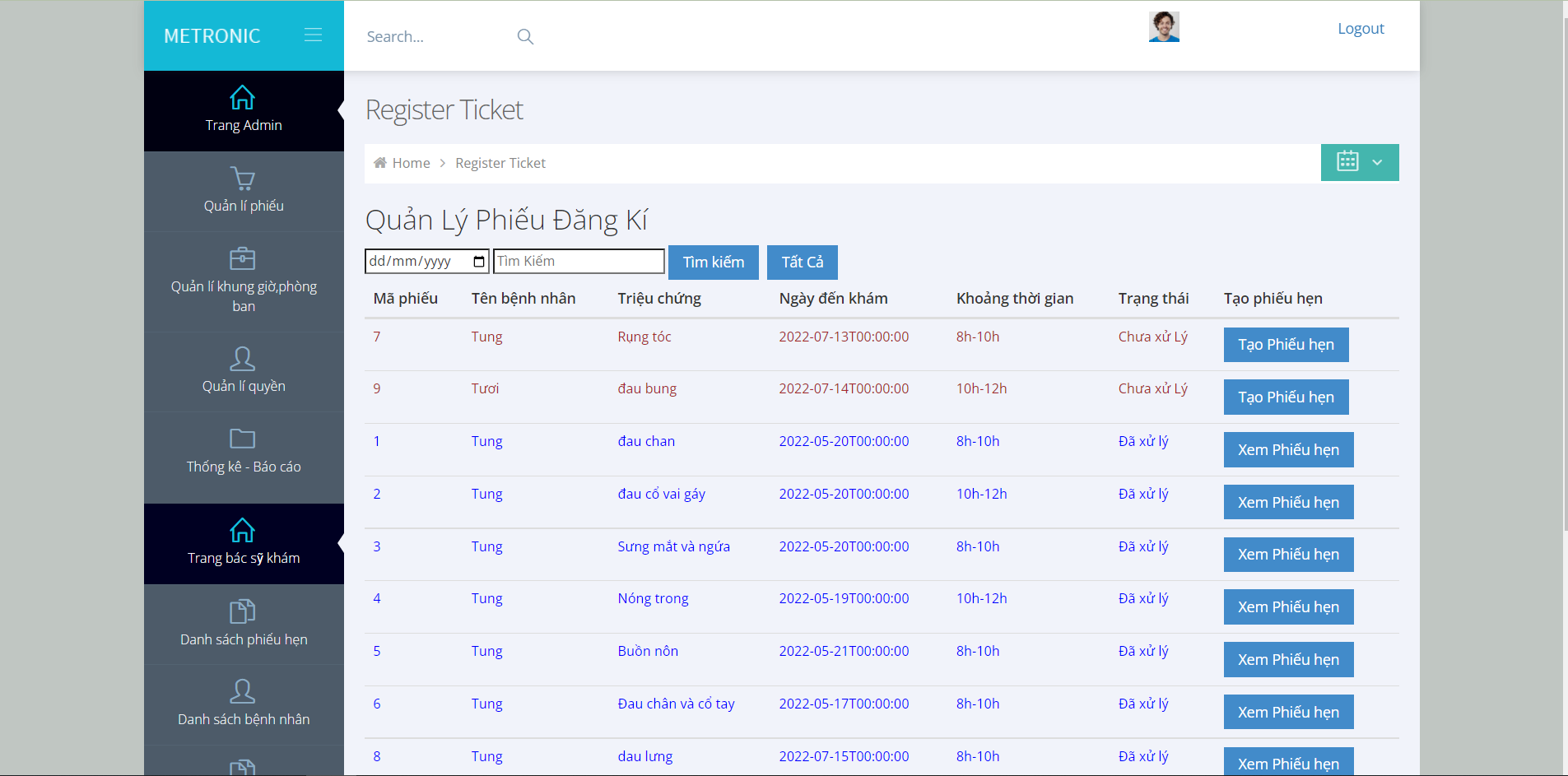
**3.2 Thiết kế giao diện**

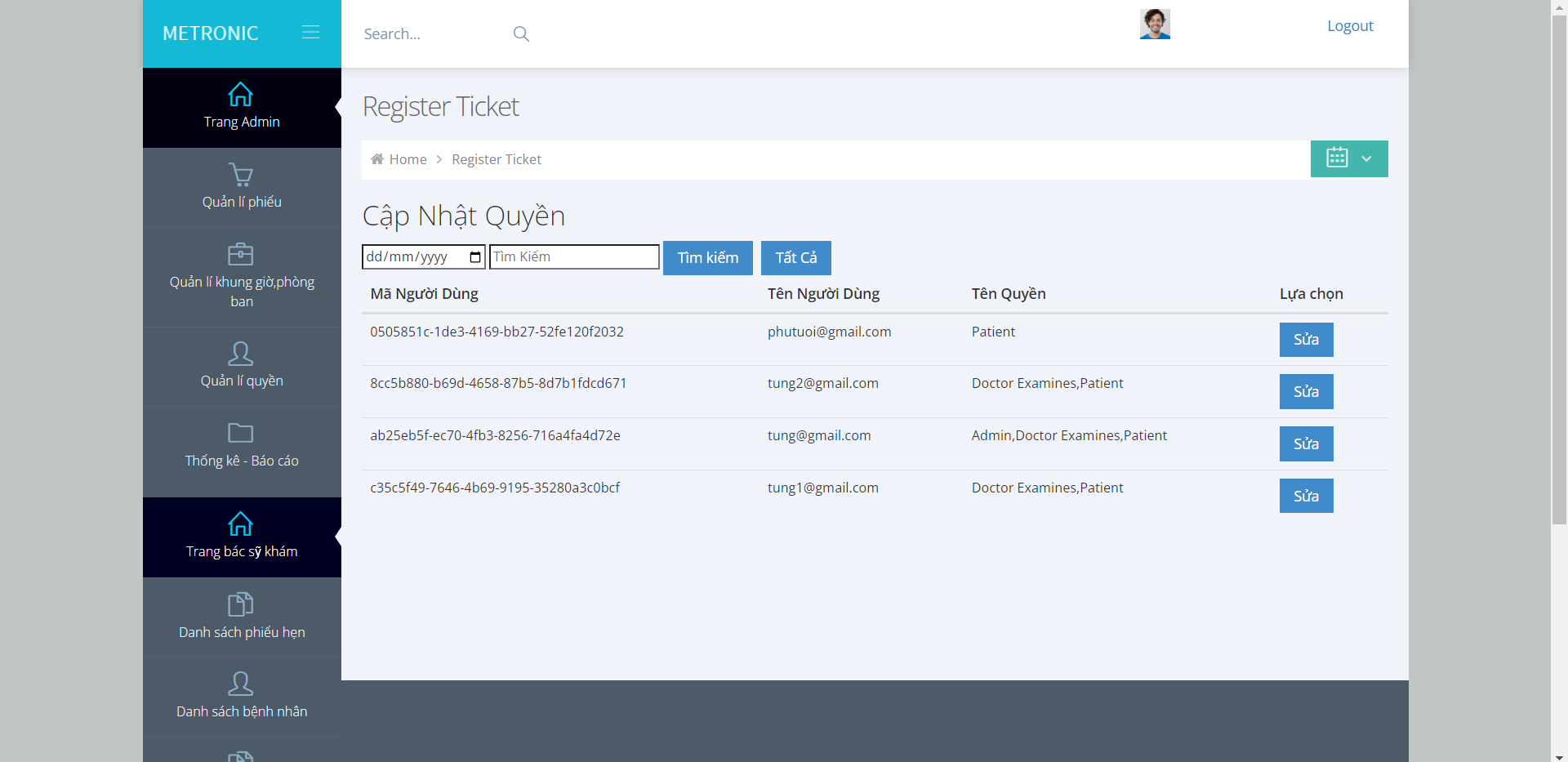
****

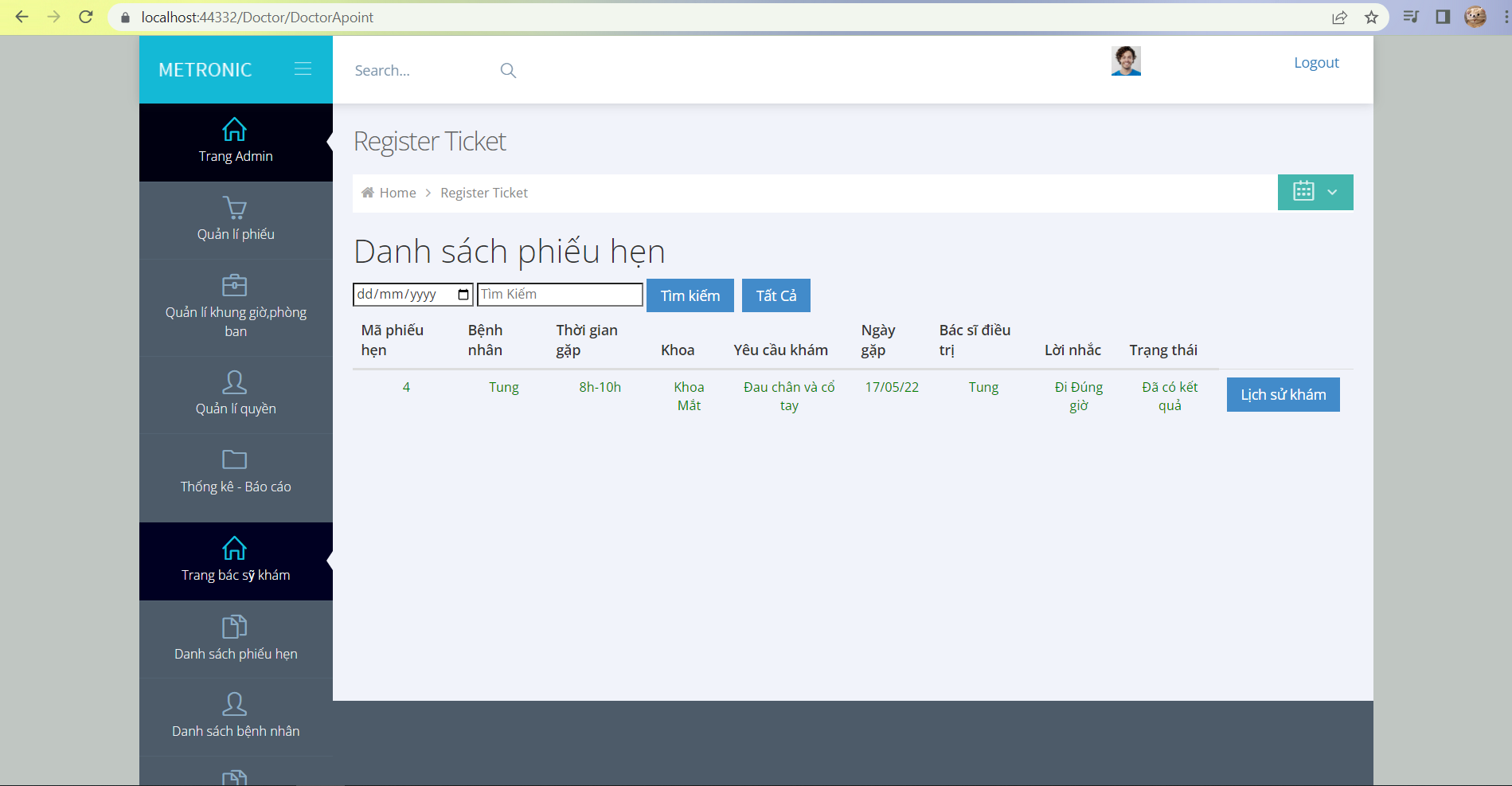
****

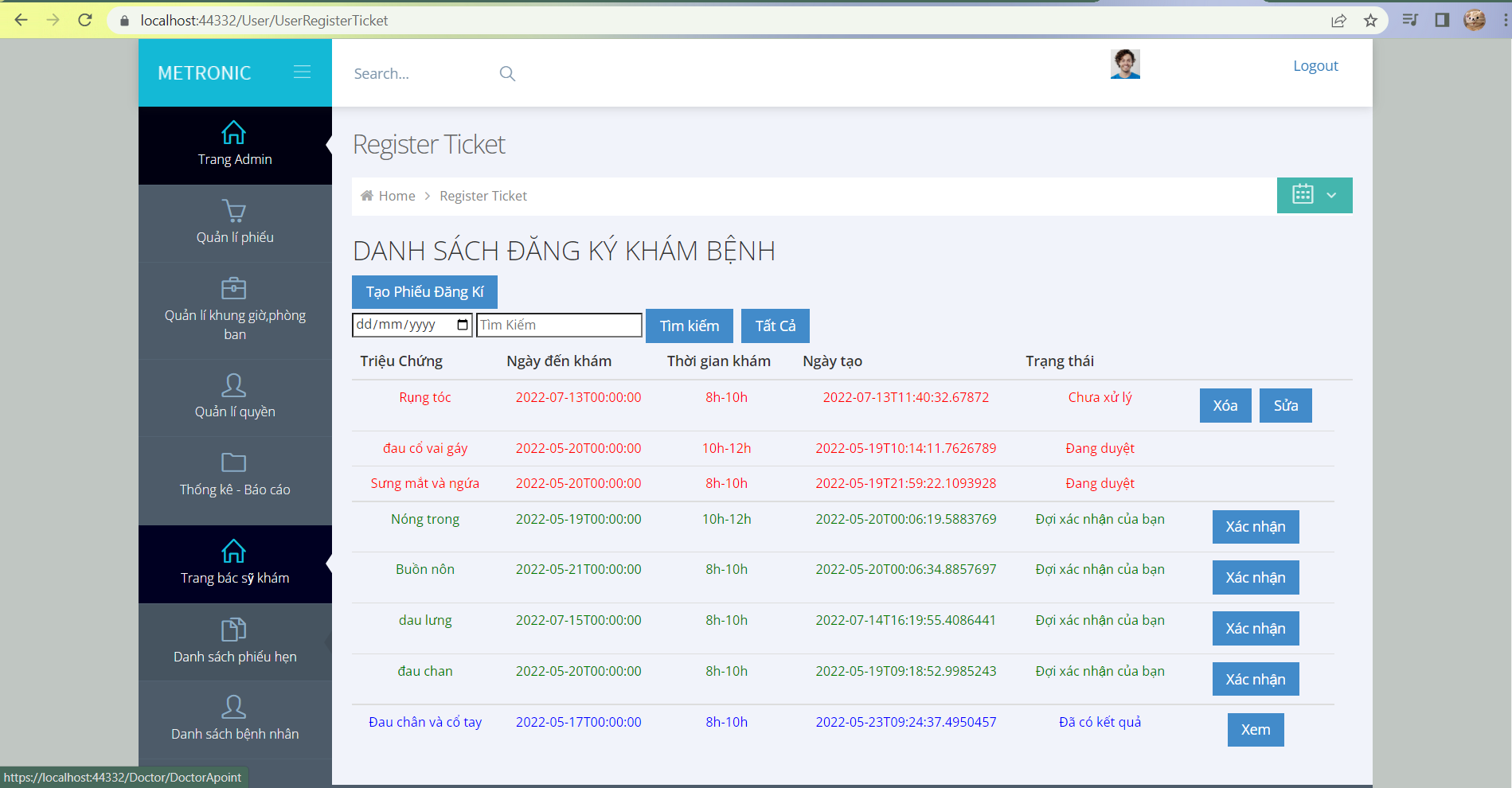
****

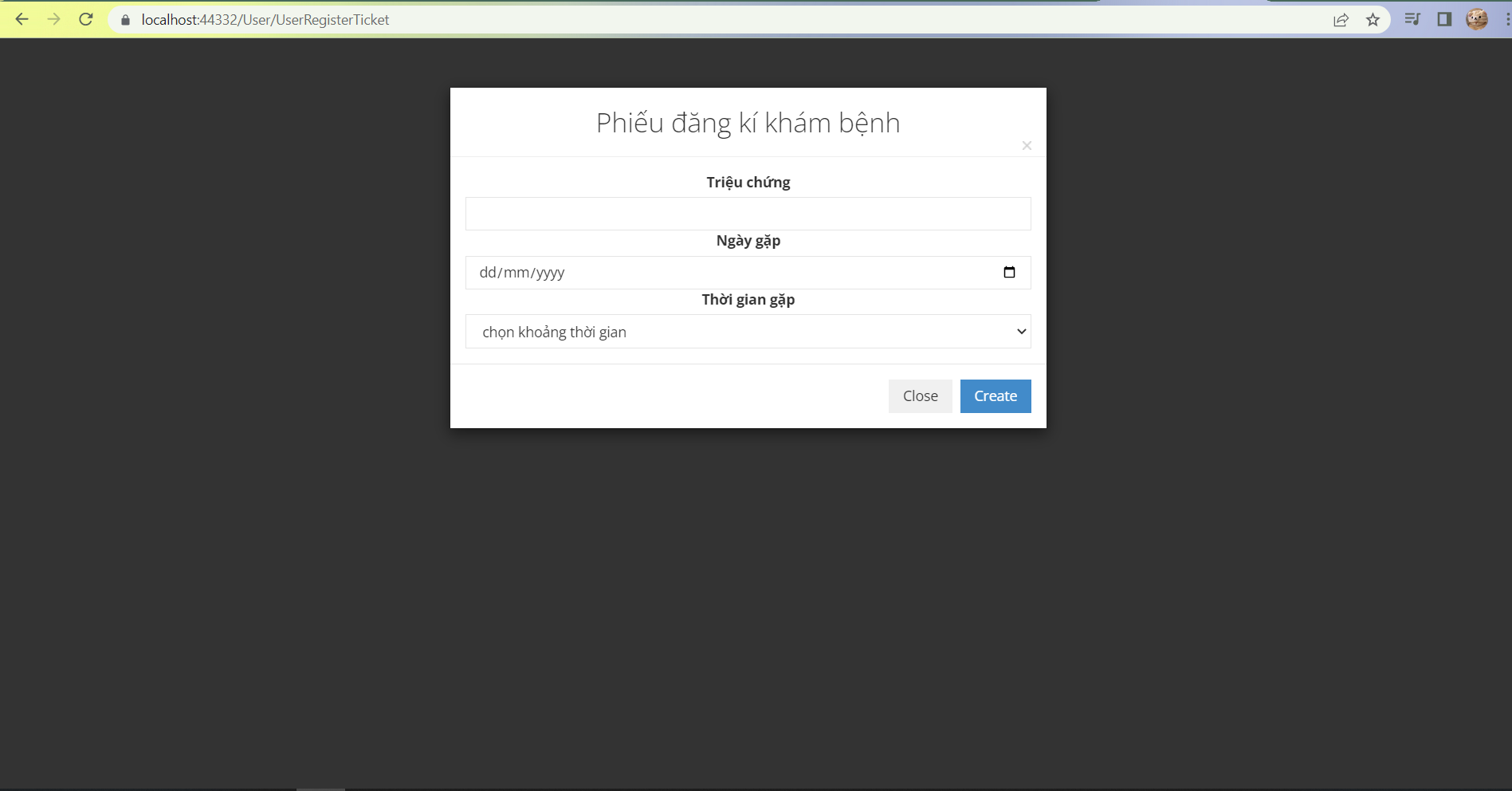
****

****

****

****

****

****

**CHƯƠNG IV: CÀI ĐẶT VÀ KIỂM THỬ**

**4.1 Mô tả về sơ lược về công cụ :**

**4.1.1 Sơ lược về C#**

### **C# là ngôn ngữ đơn giản**

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).  
 Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

### **C# là một ngôn ngữ lập trình thuần hướng đối tượng**

Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

### **C# là một ngôn ngữ ít từ khóa**

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

* Nển tảng .Netcore

.NET Core là một nền tảng phát triển đa mục đích, mã nguồn mở được duy trì bởi Microsoft và cộng đồng .NET trên GitHub. Đó là nền tảng chéo (hỗ trợ Windows, macOS và Linux) và có thể được sử dụng để xây dựng các ứng dụng thiết bị, đám mây và IoT.

* Phát triển ứng dụng web

Ngôn ngữ lập trình C# có thể được sử dụng trên bất kỳ nền tảng nào hiện nay. Bạn có thể xây dựng các trang web động và ứng dụng web với nền tảng .NET hay các nền tảng open-source khác. C# có thể làm cho ứng dụng web chạy trơn chu trên một máy chủ.

**4.1.2 Sơ lược về SQL Server**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (**Transact-SQL)**để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

**4.2 Cài đặt môi trường**

**4.3 Cách cài đặt**