**BỘ GIÁO DỤC ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN: CÔNG NGHỆ PHẦN MỀM**



***Giáo viên hướng dẫn:* Thầy NGUYỄN XUÂN NHỰT**

***Sinh viên thực hiện:* LỚP 17DTHC6**

1. **Nguyễn Ngọc Thuỷ Tiên (nhóm trưởng)**
2. **Châu Văn Hưng**
3. **Cao Hữu Duy**
4. **Lê Đồng Phương**
5. **Nguyễn Hữu Đạt**
6. **Lê Khôi Nguyên**

Ngày 24 tháng 10 năm 2018

**LỜI NHẬN XÉT CỦA GIÁO VIÊN**

**LỜI CẢM ƠN**

Trong suốt thời gian học tập môn Công Nghệ Phần Mềm, chúng em đã nhận được rất nhiều sự quan tâm giúp đỡ của thầy Nguyễn Xuân Nhựt. Chúng em xin gửi lời cảm ơn chân thành đến thầy với sự giúp đỡ tận tình của thầy suốt khoảng thời gian thực hiện đồ án. Mặc dù đã rất cố gắng nhưng do thời gian và kiến thức còn hạn chế nên chúng em đã không có nhiều điều kiện để tìm hiểu về đồ án vì vậy không tránh khỏi thiếu sót, thầy đã giúp chúng em nắm lại các kiến thức đã học và bổ sung thêm kiến thức mà chúng em chưa nắm được. Nhờ thầy mà chúng em đã có thêm nhiều kinh nghiệm làm hành trang cho sự nghiệp của em sắp tới. Đồ án này được hoàn thiện là nhờ vào sự giúp đỡ tận tình từ thầy.

Chúng em xin chân thành cảm ơn các thầy đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập và rèn luyện ở Trường Đại Học Công Nghệ Thành Phố Hồ Chí Minh.

Cuối cùng chúng em xin gửi lời cảm ơn chân thành và lời chúc sức khỏe đến thầy Nguyễn Xuân Nhựt và toàn thể giáo viên khoa Công Nghệ Thông Tin của Trường Đại Học Công Nghệ Thành Phố Hồ Chí Minh.

**MỤC LỤC**

*Trang*

[LỜI NHẬN XÉT CỦA GIÁO VIÊN ii](#_Toc528018609)

[LỜI CẢM ƠN 1](#_Toc528018610)

[MỤC LỤC 1](#_Toc528018611)

[DANH MỤC HÌNH ẢNH 4](#_Toc528018612)

[DANH MỤC CÁC BẢNG 5](#_Toc528018613)

[MỞ ĐẦU 1](#_Toc528018614)

[1. Giới thiệu về đề tài: 1](#_Toc528018615)

[2. Mục đích chọn đề tài: 1](#_Toc528018616)

[3. Đối tượng và phạm vi nghiên cứu: 2](#_Toc528018617)

[4. Phương pháp nghiên cứu: 2](#_Toc528018618)

[CHƯƠNG 1: TỔNG QUAN 3](#_Toc528018619)

[1.1 Giới thiệu ngôn ngữ Java 3](#_Toc528018620)

[1.2 Các đặc trưng của Java 3](#_Toc528018621)

[1.2.1 Giới thiệu về Java Swing: 4](#_Toc528018622)

[Phân cấp các lớp Java Swing 5](#_Toc528018623)

[1.3 Giới thiệu mô hình 3 lớp 5](#_Toc528018624)

[1.3.1 Các thành phần của mô hình 3 lớp 7](#_Toc528018625)

[1.3.1.1 Presentation Layers 7](#_Toc528018626)

[1.3.1.2 Business Logic Layer 7](#_Toc528018627)

[1.3.1.3 Data Layers 8](#_Toc528018628)

[1.4 Hệ quản trị cơ sở dữ liệu SQL Server 2008 8](#_Toc528018629)

[1.4.1 Khái niệm 8](#_Toc528018630)

[1.4.2 Giao diện SQL server 2008 9](#_Toc528018631)

[CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG 10](#_Toc528018632)

[2.1 Khảo sát hiện trạng 10](#_Toc528018633)

[2.2 Nội dung 11](#_Toc528018634)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 12](#_Toc528018635)

[3.1 Xây dựng biểu đồ Use case (Use Case Diagram) 12](#_Toc528018636)

[3.1.1 Biểu đồ Use case tổng quát 13](#_Toc528018637)

[3.1.2 Phân rã biểu đồ Use case 13](#_Toc528018638)

[3.1.2.1 Phân rã Use case : Thêm mới bệnh nhân 13](#_Toc528018639)

[3.1.2.2 Phân rã Use case : Xem thông tin bệnh nhân 14](#_Toc528018640)

[3.1.2.3 Phân rã Use case : Tìm kiếm bệnh nhân 14](#_Toc528018641)

[3.1.2.4 Phân rã Use case : Xóa bệnh nhân 14](#_Toc528018642)

[3.1.2.5 Phân rã Use case : Sửa thông tin bệnh nhân 15](#_Toc528018643)

[3.1.2.6 Phân rã Use case : Đang nhập và đang xuất 15](#_Toc528018644)

[3.1.2.7 Phân rã Use case : Xuất report excel 16](#_Toc528018645)

[3.2 Xây dựng biểu đồ trình tự (Sequence Diagram) 16](#_Toc528018646)

[3.2.1 Thêm bệnh nhân 16](#_Toc528018647)

[3.2.2 Xem thông tin bệnh nhân 17](#_Toc528018648)

[3.2.3 Xóa thông tin bệnh nhân 18](#_Toc528018649)

[3.2.4 Sửa thông tin bệnh nhân 18](#_Toc528018650)

[3.2.5 Xuất excel 19](#_Toc528018651)

[3.3 Biểu đồ hoạt động (Activity Diagram) 19](#_Toc528018652)

[3.4 Mô tả kịch bản: 20](#_Toc528018653)

[3.4.1 Form đăng nhập 20](#_Toc528018654)

[3.4.2 Form xuất file 20](#_Toc528018655)

[3.4.3 Form quản lý bệnh nhân 21](#_Toc528018656)

[3.5 Xây dựng biểu đồ lớp (Class Diagram) 22](#_Toc528018657)

[3.5.1 Biểu đồ lớp 22](#_Toc528018658)

[3.6 Các bảng sử dụng trong chương trình 23](#_Toc528018659)

[3.7 Mô hình dữ liệu quan hệ 24](#_Toc528018660)

[CHƯƠNG 4: PHẦN MỀM THIẾT KẾ 25](#_Toc528018661)

[4.1 Giao diện thêm thông tin bệnh nhân 27](#_Toc528018662)

[4.2 Giao diện Tìm – xóa – sửa. 27](#_Toc528018663)

[4.2.1 Chức năng tìm 28](#_Toc528018664)

[4.2.2 Chức năng xóa 28](#_Toc528018665)

[4.2.3 Chức năng Sửa thông tin bệnh nhân 29](#_Toc528018666)

[4.3 Giao diện xuất excel 30](#_Toc528018667)

[4.4 Giao diện Thông tin phần mềm 30](#_Toc528018668)

[KẾT LUẬN 31](#_Toc528018669)

[DANH MỤC TÀI LIỆU THAM KHẢO 32](#_Toc528018670)

**DANH MỤC HÌNH ẢNH**

[Hình 1. 1 Mô hình 3 lớp 6](#_Toc528016457)

[Hình 1. 2 Cấu trúc chi tiết mô hình 3 lớp 6](#_Toc528016458)

[Hình 3. 1 Biểu đồ Use case tổng quát 13](#_Toc528016445)

[Hình 3. 2 Use case Thêm bệnh nhân 13](#_Toc528016446)

[Hình 3. 3 Use case Xem thông tin bệnh nhân 14](#_Toc528016447)

[Hình 3. 4 Use case Tìm kiếm bệnh nhân 14](#_Toc528016448)

[Hình 3. 5 Use case Xóa bệnh nhân 14](#_Toc528016449)

[Hình 3. 6 Use case Sửa thông tin bệnh nhân 15](#_Toc528016450)

[Hình 3. 7 Use case Đang nhập và đang xuất 15](#_Toc528016451)

[Hình 3. 8 Use case Xuất excel 16](#_Toc528016452)

[Hình 3. 10 Biểu đồ trình tự Thêm bệnh nhân 16](#_Toc528016453)

[Hình 3. 19 Biểu đồ trình tự Xem thông tin bệnh nhân 17](#_Toc528016454)

[Hình 3. 40 Biểu đồ lớp 22](#_Toc528016455)

[Hình 4. 1 Giao diện đăng nhập 25](#_Toc528018527)

[Hình 4. 2 Giao diện chính 26](#_Toc528018528)

[Hình 4. 3 Giao diện thêm thông tin bệnh nhân 27](#_Toc528018529)

[Hình 4. 9 Giao diện tìm-xóa-sửa 27](#_Toc528018530)

[Hình 4. 10 Giao diện tìm 28](#_Toc528018531)

[Hình 4. 11 Giao diện xóa bệnh nhân 28](#_Toc528018532)

[Hình 4. 14 Giao diện sửa thông tin bệnh nhân 29](#_Toc528018533)

[Hình 4. 20 Giao diện xuất excel 30](#_Toc528018534)

[Hình 4. 28 Giao diện Thông tin phần mềm 30](#_Toc528018535)

# DANH MỤC CÁC BẢNG

[Bảng 3. 1 Bảng User 23](#_Toc528016502)

[Bảng 3. 1 Bảng bệnh nhân 23](#_Toc528016503)

[Bảng 3. 1 Bảng bệnh nhân 23](#_Toc528016504)

[Bảng 3. 1 Bảng sử dụng thuốc 24](#_Toc528016505)

[Bảng 3. 13 Mô hình dữ liệu quan hệ 24](#_Toc528016506)

# 

# MỞ ĐẦU

## Giới thiệu về đề tài:

Ngày nay việc sử dụng phần mềm để quản lý bệnh nhân trong các cơ sở khám chữa bệnh rất phổ biến. Nó mang lại nhiều thuận tiện cho việc quản lý bệnh nhân, theo dõi bệnh nhân, quản lý được toàn bộ hệ thống và đặc biệt là có độ chính xác cao. Vì vậy, người tiêu dùng cũng ngày càng quan tâm hơn đến việc sử dụng phần mềm nhằm mang lại nhiều thuận tiện cho công việc của mình. Tuy nhiên, họ còn e ngại sử dụng do giá thành các ứng dụng phần mềm còn cao và đòi hỏi phải có kỹ năng và hiểu biết thì mới sử dụng được. Do đó, với mong muốn mang lại những thuận tiện cho người dùng chúng em chọn đề tài xây dựng phần mềm quản lý bệnh nhân lao với những chức năng đơn giản dễ sử dụng và đáp ứng được các yêu cầu thiết yếu cho việc quản lý bệnh nhân trong trạm y tế.

## Mục đích chọn đề tài:

Phần mềm quản lý bệnh nhân lao giúp cho người quản lý dễ dàng quản lý các bệnh nhân, cung cấp thuốc, xuất nhập thông tin và xuất file excel quyết toán lên tuyến trên một cách hiệu quả. Mục tiêu cụ thể như sau:

* Phần mềm có khả năng quản lý các bệnh nhân đến khám chữa bệnh tại trạm y tế, quản lý bệnh nhân, thay đổi thông tin bệnh nhân, tìm kiếm thông tin bệnh nhân, cung cấp thuốc.
* Phần mềm có thể quản lý bệnh nhân như tên, tuổi, địa chỉ, cân nặng, ngày điều trị, loại bệnh, loại thuốc mà bệnh nhân đang dùng và số lượng viên bao nhiêu.
* Người quản lý có thể dễ dàng xuất file Excel để đi quyết toán thuốc theo định kì mỗi 2 tuần lên tuyến trên.
* Người quản lý có thể sao lưu phục hồi được dữ liệu để đảm bảo an toàn cho cơ sở dữ liệu của mình.
* Phần mềm có đăng nhập và đăng xuất để bảo vệ thông tin người bệnh.

## 

## Đối tượng và phạm vi nghiên cứu:

* Đối tượng nghiên cứu: Các trạm y tế, các cơ sở khám cho bệnh nhân lao trên toàn các tuyến xã trong cả nước.
* Phạm vi nghiên cứu:
* Quản lý bệnh nhân: Nhập, xuất thông tin bệnh nhân.
* Quản lý bệnh nhân: tìm kiếm, sửa, xoá thông tin bệnh nhân.
* Quản lý thống kê – in ấn: Thống kê số lượng viên thuốc, xuất file quyết toán.

## Phương pháp nghiên cứu:

* Phương pháp phân tích, tổng hợp.
* Sưu tầm, nghiên cứu, các phương pháp xây dựng hệ thống phần mềm.
* Thống kê, thu thập thông tin.

# TỔNG QUAN

## Giới thiệu ngôn ngữ Java

Java là một ngôn ngữ lập trình và nền tảng tính toán được phân phối lần đầu tiên bởi Sun Microsystems vào năm 1995. Rất nhiều ứng dụng, trang web đều được viết bằng Java. Java nhanh, bảo mật và đáng tin cậy.

Java là ngôn ngữ [lập trình hướng đối tượng](https://stackjava.com/faq/lap-trinh-huong-doi-tuong-la-gi-uu-nhuoc-diem.html) (OOP).

**Về tốc độ:**

* Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ “biên dịch tại chỗ” – Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn.
* Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần.

**Về quản lý bộ nhớ:**

* Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi [Java Virtual Machine (JVM)](https://stackjava.com/java-basic/su-khac-nhau-giua-jdk-jre-va-jvm.html) bằng cách tự động “dọn dẹp rác”. Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của [JVM](https://stackjava.com/java-basic/su-khac-nhau-giua-jdk-jre-va-jvm.html)) mà người lập trình không đóng các kết nối thì rò rỉ dữ liệu vẫn có thể xảy ra.

## Các đặc trưng của Java

**Hướng đối tượng**

* Mọi thực thể trong chương trình đều là một đối tượng (1 class xác định)
* Các biến, hàm đều nằm trong một class nào đó

**Đơn giản**

* Loại bỏ con trỏ
* Loại bỏ lệnh goto
* Không cho phép đa kế thừa (chuyển sang sử dụng interface)

**Độc lập phần cứng và hệ điều hành**

Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Do đó một chương trình viết bằng Java có thể chạy trên nhiều thiết bị, nhiều hệ điều hành khác nhau.

**Mạnh mẽ**

* Quá trình cấp phát, giải phóng bộ nhớ được thực hiện tự động.
* Yêu cầu chặt chẽ khi khai báo dữ liệu, ép kiểu dữ liệu.
* Tự động phát hiện lỗi lúc biên dịch.
* Không sử dụng con trỏ hoặc các phép toán con trỏ.

**Phân tán**

* Java hỗ trợ lập trình cho các hệ thống phân tán như client-server, RMI… bằng Java web, UDP, TCP…

**Đa luồng**

* Java hỗ trợ lập trình đa luồng (multithreading); việc đồng bộ dữ liệu trong lập trình đa luồng cũng khá đơn giản.

### Giới thiệu về Java Swing:

Java Swing là một phần của Java được sử dụng để tạo các ứng dụng window-based. Nó được xây dựng trên API AWT (Abstract Windowing Toolkit) và được viết hoàn toàn bằng Java.

Không giống như AWT, Java Swing cung cấp các thành phần không phụ thuộc vào nền tảng và nhẹ hơn.

Gói javax.swing cung cấp các lớp cho java swing API như JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser, v.v.

* Ưu điểm của Java Swing:

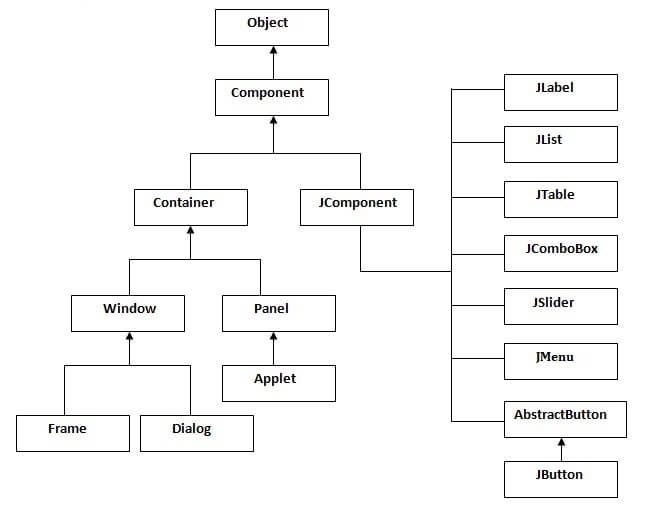
+ Chiếm ít tài nguyên của hệ thống.

+ Hỗ trợ giao diện **Look and Feel.**

+ Dễ dàng mở rộng.

## Phân cấp các lớp Java Swing

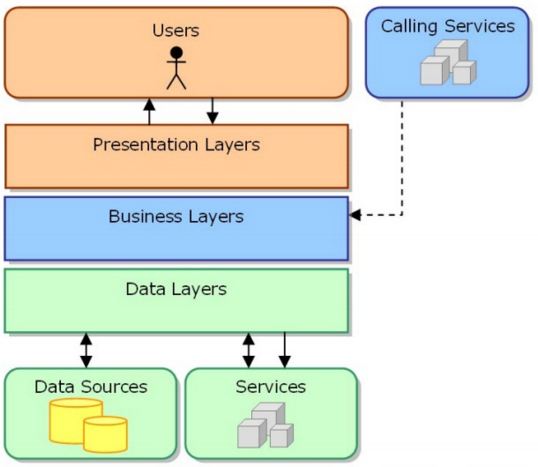
Hệ thống phân cấp của API java swing được đưa ra dưới đây.



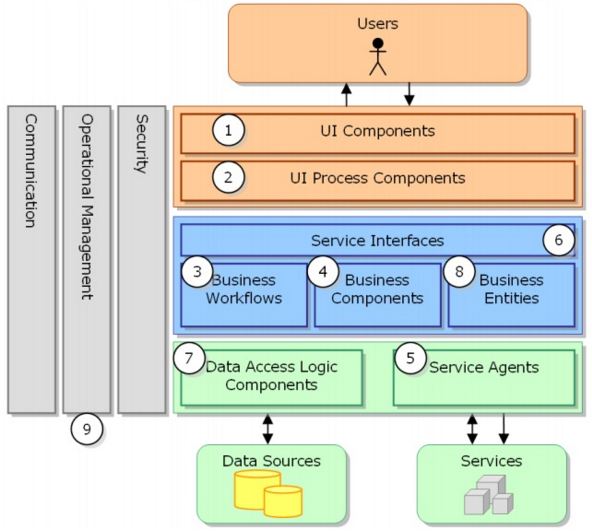
## Giới thiệu mô hình 3 lớp

Khi mới tiếp xúc với Windows Form và ADO.NET, việc lập trình bắt đầu trở nên phức tạp khi dự án lớn dần. Bởi vậy để dễ quản lý các thành phần của hệ thống, cũng như không bị ảnh hưởng bởi các thay đổi, người ta hay nhóm các thành phần có cùng chức năng lại với nhau và phân chia trách nhiệm cho từng nhóm để công việc không bị chồng chéo và ảnh hưởng lẫn nhau. Một trong những mô hình lập trình như vậy đó gọi là mô hình 3 lớp (Three Layers).

Mô hình 3 lớp được cấu thành từ: **Presentation Layers, Business Layers** và **Data Layers**. Các lớp này sẽ giao tiếp với nhau thông qua các dịch vụ (serices) mà mỗi lớp cung cấp để tạo nên ứng dụng, lớp này cũng không cần biết bên trong lớp kia làm gì mà chỉ cần biết lớp kia cung cấp dịch vụ gì cho mình và sử dụng nó.



Hình 1. 1 Mô hình 3 lớp

**

Hình 1. 2 Cấu trúc chi tiết mô hình 3 lớp

### Các thành phần của mô hình 3 lớp

#### ****Presentation Layers****

Lớp này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả dữ liệu thông qua các thành phần trong giao diện người sử dụng. Lớp này sẽ sử dụng các dịch vụ do lớp Business Logic cung cấp. Trong .NET thì bạn có thể dùng **Windows Forms**, **ASP.NET** hay **Mobile Forms** để hiện thực lớp này.

Trong lớp này có 2 thành phần chính là **User Interface Components** và **User Interface Process Components**.

**UI Components:** Là những phần tử chịu trách nhiệm thu thập và hiển thị thông tin cho người dùng cuối. Trong ASP.NET thì những thành phần này có thể là các TextBox, các Button, DataGrid…

**UI Process Components**: Là thành phần chịu trách nhiệm quản lý các qui trình chuyển đổi giữa các UI Components. Ví dụ chịu trách nhiệm quản lý các màn hình nhập dữ liệu trong một loạt các thao tác định trước như các bước trong một Wizard…

#### ****Business Logic Layer****

Lớp này thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do lớp **Data Access** cung cấp, và cung cấp các dịch vụ cho lớp **Presentation**. Lớp này cũng có thể sử dụng các dịch vụ của các nhà cung cấp thứ 3 để thực hiện công việc của mình.

Trong lớp này có các thành phần chính là **Business Components**, **Business Entities** và **Service Interface**.

**Service Interface:** Là giao diện lập trình mà lớp này cung cấp cho lớp **Presentation**sử dụng. Lớp **Presentation**chỉ cần biết các dịch vụ thông qua giao diện này mà không cần phải quan tâm đến bên trong lớp này được hiện thực như thế nào.

**Business Entities:** Là những thực thể mô tả những đối tượng thông tin mà hệ thống xử lý. Các Business Entities này cũng được dùng để trao đổi thông tin giữa lớp **Presentation**và lớp **Data Layers**.

**Business Components:** Là những thành phần chính thực hiện các dịch vụ mà **Service Interface** cung cấp, chịu trách nhiệm kiểm tra các ràng buộc logic (constraints), các qui tắc nghiệp vụ (Business Rules), sử dụng các dịch vụ bên ngoài khác để thực hiện các yêu cầu của ứng dụng.

#### ****Data Layers****

Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng. Thường lớp này sẽ sử dụng các dịch vụ của các hệ quản trị cơ sở dữ liệu như SQL Server, Oracle,… để thực hiện nhiệm vụ của mình. Trong lớp này có các thành phần chính là **Data Access Logic**, **Data Sources**, **Servive Agents**.

**Data Access Logic Components (DAL)** là thành phần chính chịu trách nhiệm lưu trữ vào và truy xuất dữ liệu từ các nguồn dữ liệu – Data Sources như RDMBS, XML, File systems…. Trong .NET Các **DAL**này thường được hiện thực bằng cách sử dụng thư viện ADO.NET để giao tiếp với các hệ cơ sở dữ liệu hoặc sử dụng các O/R Mapping Frameworks để thực hiện việc ánh xạ các đối tượng trong bộ nhớ thành dữ liệu lưu trữ trong CSDL. Chúng ta sẽ tìm hiểu các thư viện O/R Mapping này trong một bài viết khác.

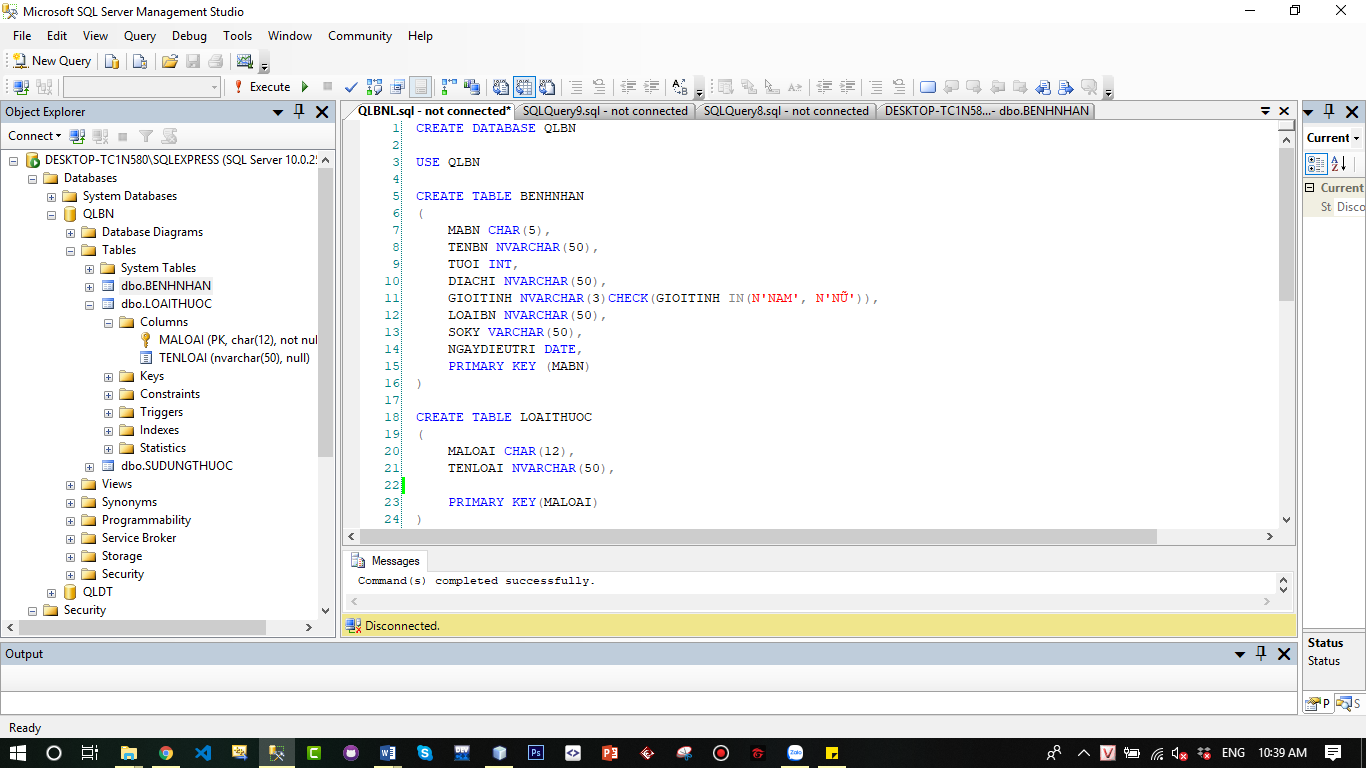
**Service Agents:** Là những thành phần trợ giúp việc truy xuất các dịch vụ bên ngoài một cách dễ dàng và đơn giản như truy xuất các dịch vụ nội tại.

## Hệ quản trị cơ sở dữ liệu SQL Server 2008

### Khái niệm

* SQL (Structured Query Language - ngôn ngữ truy vấn mang tính cấu trúc) là một loại ngôn ngữ máy tính phổ biến để tạo, sửa và lấy dữ liệu từ một hệ quản trị cơ sở dữ liệu quan hệ. Ngôn ngữ này phát triển vượt xa so với mục đích ban đầu là để phục vụ các hệ quản trị cơ sở dữ liệu đối tượng - quan hệ. Nó là một tiêu chuẩn ANSI/ISO.
* Theo Microsoft Việt Nam, nhằm giúp người sử dụng tạo ra những giải pháp tốt nhất cho nhu cầu quản lý dữ liệu, SQL Server 2008 cung cấp nền tảng dữ liệu thông minh, hiệu suất cao, giúp khách hàng quản lý hiệu quả các loại dữ liệu với mọi kích thước được lưu trữ tại chính doanh nghiệp hay trên nền điện toán đám mây.
* Trên cơ sở đó, SQL Server 2008 giúp giảm thời gian cũng như chi phí của việc phát triển và quản lý các ứng dụng, từ đó giúp khách hàng giải quyết hiệu quả các thách thức liên quan đến sự gia tăng khối lượng dữ liệu, đưa ra quyết định nhanh chóng hơn nhằm thúc đẩy hoạt động kinh doanh.

### C:\Users\Tien Nguyen\Desktop\sql2008.pngGiao diện SQL server 2008



# KHẢO SÁT HIỆN TRẠNG

## Khảo sát hiện trạng

Hiện trạng các y sĩ của khoa lao tại Trạm Y Tế hằng ngày mỗi người sẽ làm công việc khám và kê toa phát thuốc cho bệnh nhân lao theo phác đồ điều trị từ tuyến huyện gửi về (thuốc sẽ nhận từ tuyến huyện về).

Các y sĩ phải ghi chép bằng tay và nhập liệu qua ứng dụng Excel, mỗi 2 tuần phải tự tổng hợp số lần uống qua từng ngày của bệnh nhân và phân loại bệnh nhân, để đi quyết toán thuốc xuống huyện và nhận thuốc mới về cấp.

Quá trình nhập liệu bằng tay và mỗi 2 tuần phải tổng hợp, phân loại như vậy khiến cho các y sĩ gặp nhiều khó khăn và mất thời gian. Nên vì vậy họ cần đến một phần mềm đơn giản, dễ quan sát cho họ thuận tiện hơn trong việc quản lý bệnh nhân của mình.

* Sơ đồ chức năng hệ thống

Quản lý hệ thống

Quản lý bệnh nhân

Thống kê - in ấn

## Nội dung

* Xây dựng phần mềm hỗ trợ quản lý bệnh nhân lao trong phòng khám lao của Trạm Y Tế xã Khánh Hoà:
  + - Giải quyết, hổ trợ các công việc liên quan đến hoạt động hằng ngày của khoa lao như: có chức năng đăng nhập – đăng xuất, tiếp nhận bệnh nhân mới, chọn loại thuốc và số lượng viên, chỉnh sửa thông tin bệnh nhân, xoá bệnh nhân đã xong đợt điều trị…v…v…..
    - Ứng dụng công nghệ thông tin vào việc giảm thiểu công sức, tiết kiệm thời gian cho các y sĩ tại Trạm.
    - Phần mềm hổ trợ xuất file Excel theo yêu cầu y sĩ để thuận tiện cho việc quyết toán thuốc xuống tuyến huyện định kỳ 2 tuần/lần.
* Phần mềm phải đáp ứng được các tác vụ cơ bản, đồng thời phải hoạt động ổn định và có hiệu suất cao, đảm bảo được hiệu quả khi số lượng bệnh nhân lưu trữ cao.
* Đặc biệt, phần mềm phải phù hợp với lứa tuổi người đang sử dụng là các cán bộ ý tế trong độ tuổi trung niên khoảng 45 – 58 tuổi. Yêu cầu thiết kế giao diện đơn giản, chữ to rõ dễ nhìn, không cầu kì phức tạp nhiều thứ trên giao diện.
* Phần mềm đồng thời phải có khả năng mở rộng, nâng cấp khi cần thiết.

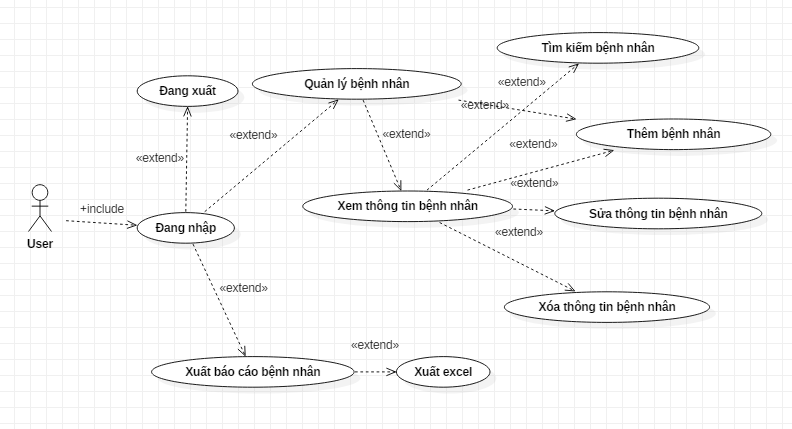
# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Xây dựng biểu đồ Use case (Use Case Diagram)

1. Xây dựng usecase tổng quát:

|  |  |  |
| --- | --- | --- |
| STT | Usecase | Mô tả (vài dòng) |
| 1 | Thêm bệnh nhân | Sau khi người dùng chọn thêm mới, sẽ nhập tất cả các thông tin cần thiết của bệnh nhân, và được lưu xuống cơ sở dữ liệu. |
| 2 | Xem thông tin bệnh nhân. | Xem thông tin của tất cả các bệnh nhân trên màn hình chính. |
| 3 | Xoá thông tin 1 bệnh nhân. | Người dùng có thể xoá thông tin của bất kỳ bệnh nhân nào |
| 4 | Tìm kiếm | Cho người dùng tìm kiếm thông tin bệnh nhân thông qua nhập tên bệnh nhân. |
| 5 | Sửa thông tin 1 bệnh nhân. | Người dùng có thể sửa thông tin của bệnh nhân |
| 6 | Đăng nhập và đăng xuẩt | Đăng nhập vào hệ thống làm việc và đăng xuất khoá phần mềm lại khi không sử dụng. |
| 7 | Xuất file | Xuất ra thông tin bệnh nhân hoặc tất cả các bệnh nhân dưới dạng excel. |

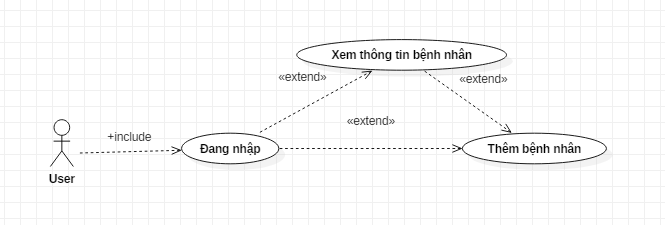
### Biểu đồ Use case tổng quát



Hình 3. 1 Biểu đồ Use case tổng quát

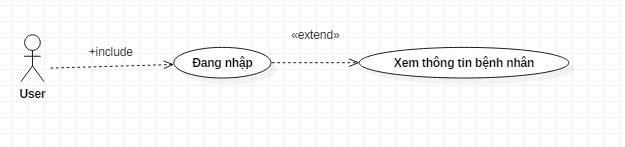
### Phân rã biểu đồ Use case

#### Phân rã Use case : Thêm mới bệnh nhân



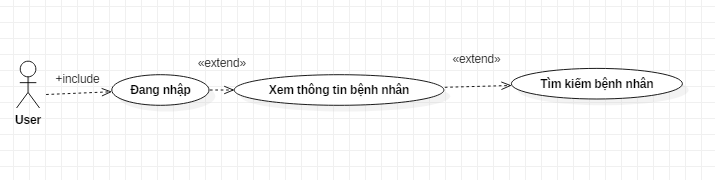
Hình 3. 2 Use case Thêm bệnh nhân

#### Phân rã Use case : Xem thông tin bệnh nhân



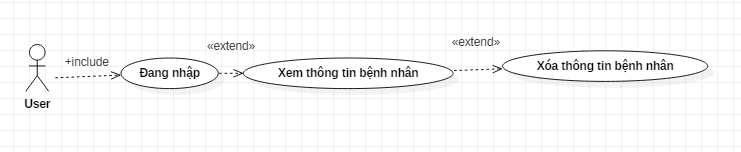
Hình 3. 3 Use case Xem thông tin bệnh nhân

#### Phân rã Use case : Tìm kiếm bệnh nhân



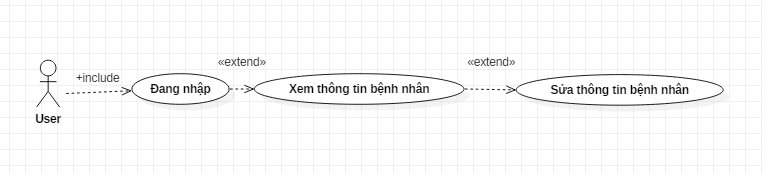
Hình 3. 4 Use case Tìm kiếm bệnh nhân

#### Phân rã Use case : Xóa bệnh nhân



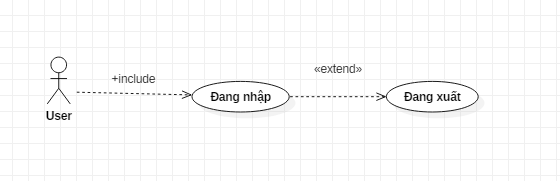
Hình 3. 5 Use case Xóa bệnh nhân

#### Phân rã Use case : Sửa thông tin bệnh nhân



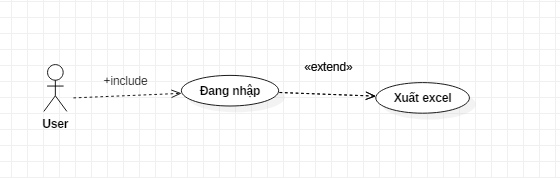
Hình 3. 6 Use case Sửa thông tin bệnh nhân

#### Phân rã Use case : Đang nhập và đang xuất



Hình 3. 7 Use case Đang nhập và đang xuất

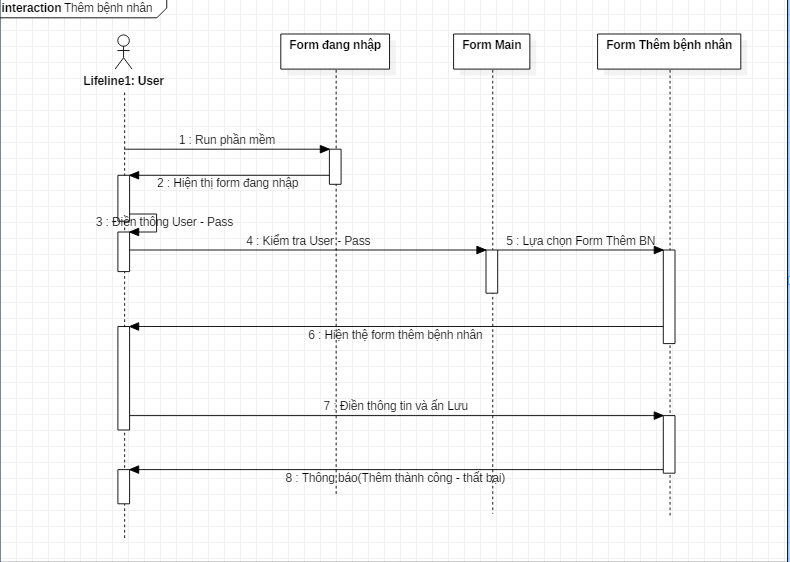
#### Phân rã Use case : Xuất report excel



Hình 3. 8 Use case Xuất excel

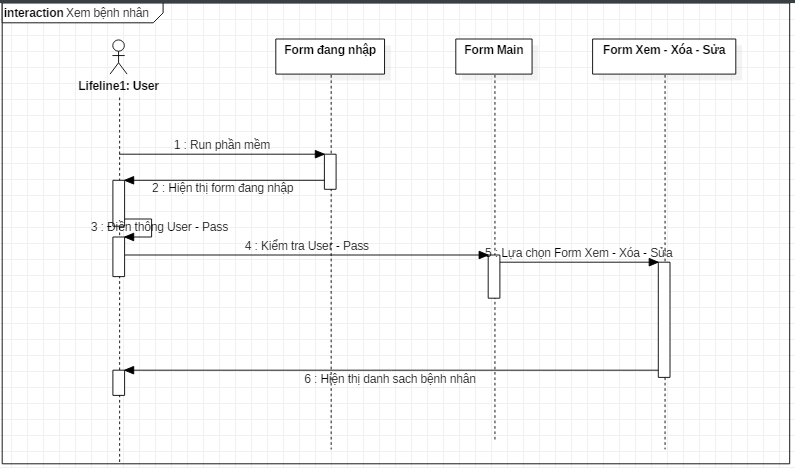
## Xây dựng biểu đồ trình tự (Sequence Diagram)

### Thêm bệnh nhân



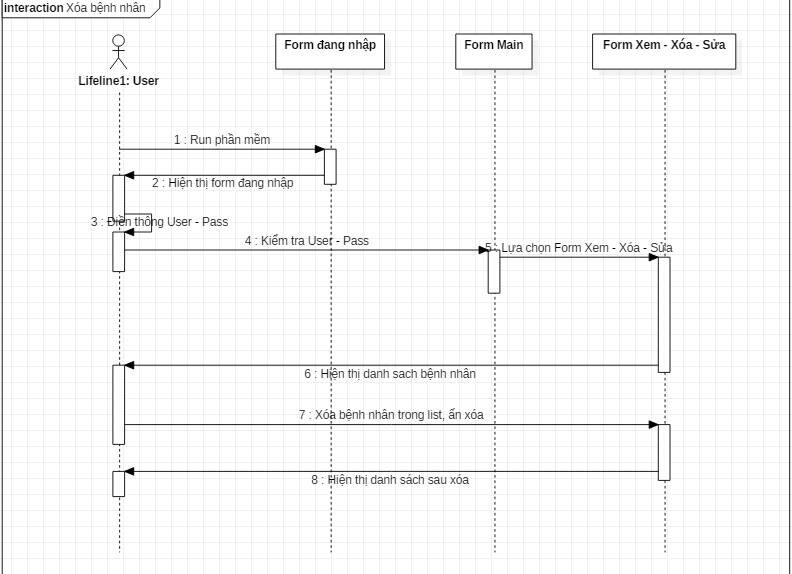
Hình 3. 10 Biểu đồ trình tự Thêm bệnh nhân

### Xem thông tin bệnh nhân



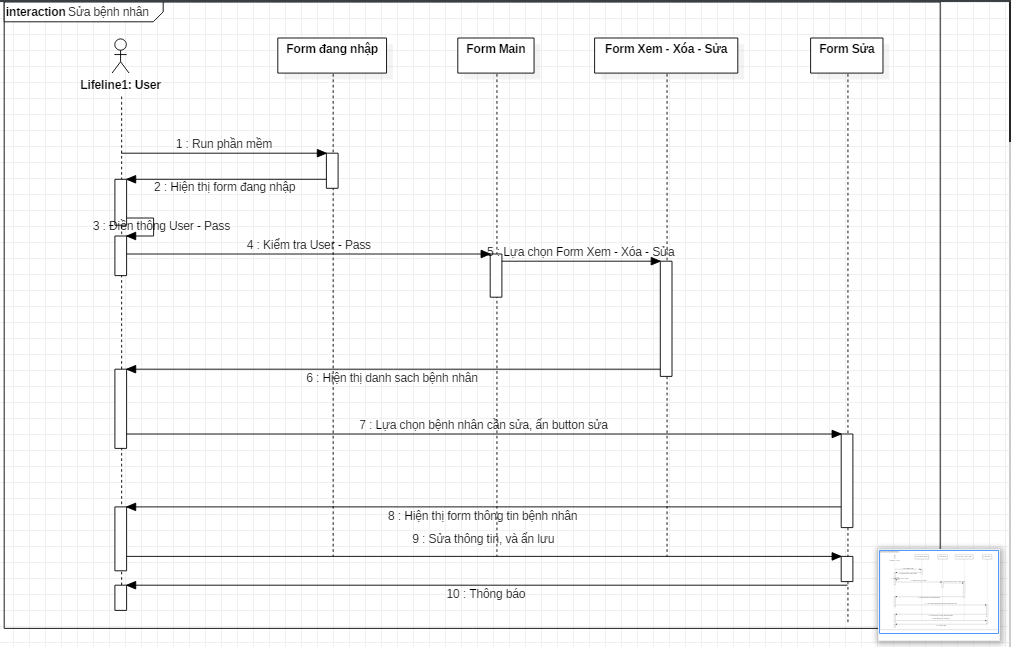
Hình 3. 19 Biểu đồ trình tự Xem thông tin bệnh nhân

### Xóa thông tin bệnh nhân



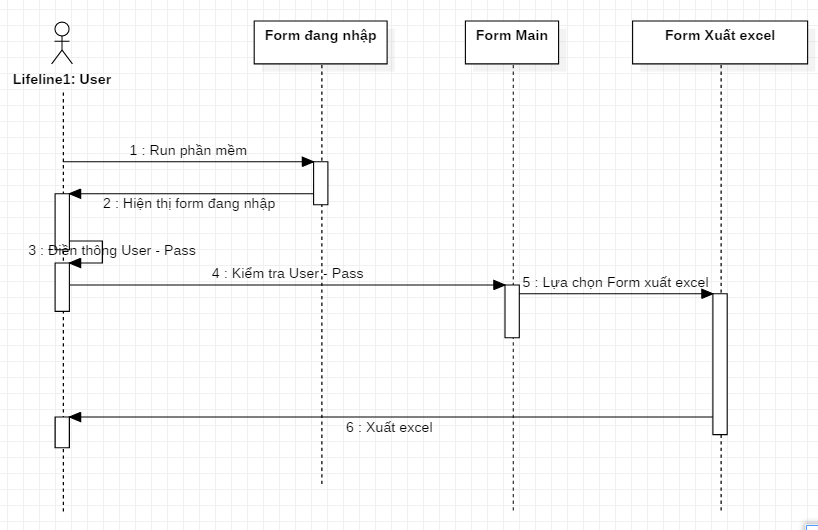
Hình 3. 20 Biểu đồ trình tự Xóa thông tin bệnh nhân

### Sửa thông tin bệnh nhân



Hình 3. 21 Biểu đồ trình tự Sửa thông tin bệnh nhân

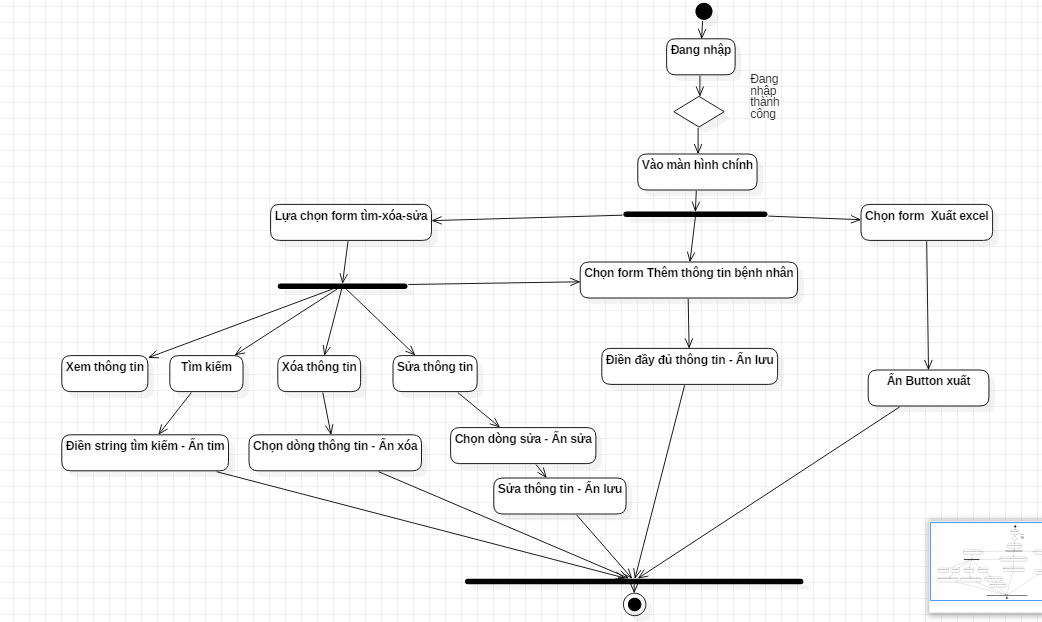
### Xuất excel



Hình 3. 22 Biểu đồ trình tự Xuất excel thông tin bệnh nhân

## Biểu đồ hoạt động (Activity Diagram)

Biểu đồ hoạt động chung :



## Mô tả kịch bản:

### Form đăng nhập

|  |  |
| --- | --- |
| **Tên Use case** | Đăng nhập |
| **Tác nhân chính** | Người dùng hệ thống |
| **Người chịu trách nhiệm** | Người dùng hệ thống |
| **Tiền điều kiện** | Kích hoạt chương trình |
| **Kích hoạt** | Người dùng hệ thống chọn chức năng ***Đăng nhập*** |
| * **Luồng sự kiện chính** * Khi kích hoạt chương trình, sẽ xuất hiện giao diện đăng nhập yêu cầu cung cấp Tài khoản và Mật khẩu. * Người sử dụng sẽ điền thông tin vào giao diện và nhấn **Đăng nhập**. * Hệ thống sẽ kiểm tra Tài khoản và Mật khẩu. * Nếu hợp lệ hệ thống thông báo thành công và đưa vào giao diện chính. * Kết thúc Use case. * **Luồng sự kiện rẽ nhánh** * Luồng 1: * Tại giao diện Đăng nhập, người dùng có thể chọn ***Thoát***. * Kết thúc Use case. * Luồng 2: * Hệ thống kiểm tra thông tin đăng nhập, nếu sai sẽ từ chối đăng nhập. * Hiển thị thông báo. * Kết thúc Use case. * Hệ thống kết nối CSDL để kiểm tra thông tin, nếu kết nối không thành công, không thực hiện kiểm tra. * Thông báo lỗi. * Kết thúc Use case. | |

### Form xuất file

|  |  |
| --- | --- |
| **Tên Use case** | Xuất file báo cáo bệnh nhân |
| **Tác nhân chính** | Người dùng phần mềm |
| **Người chịu trách nhiệm** | Người dùng phần mềm |
| **Tiền điều kiện** | Đã đang nhập, kích hoạt gia diện chính |
| **Kích hoạt** | Người dùng chọn chức năng ***Xuất file excel*** |
| * **Luồng sự kiện chính** * Người dùng chọn Xuất excel từ màn hình chính * Ấn xuất file * Lựa chọn nơi lưu trữ * Thông báo * Kết thúc Use Case | |

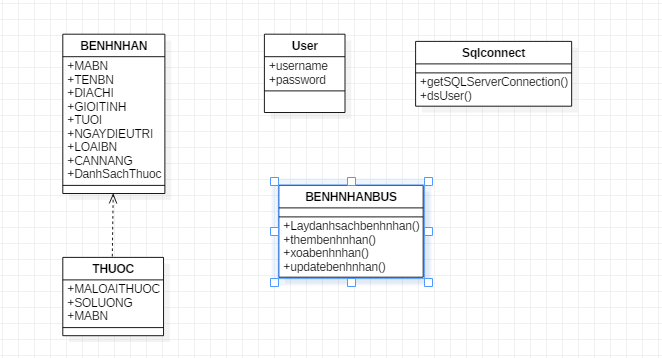
### Form quản lý bệnh nhân

|  |  |
| --- | --- |
| **Tên Use case** | Quản lý bệnh nhân |
| **Tác nhân chính** | Người dùng |
| **Người chịu trách nhiệm** | Các y bác sĩ |
| **Tiền điều kiện** | Kích hoạt giao diện Quản lý bệnh nhân |
| **Đảm bảo tối thiểu** | Loại bỏ các thông tin đã nhập và quay lại bước trước |
| **Đảm bảo thành công** | Thông tin bệnh nhân được lưu vào CSDL |
| **Kích hoạt** | Người dùng chọn chức năng ***Thêm, Xóa, Sửa, xuất file*** |
| * **Luồng sự kiện chính** * Người dùng chọn chức năng Quản lý bệnh nhân trên giao diện chính. * Form Quản lý khách hàng xuất hiện, gồm các thông tin: * Thêm * Xóa * Sửa * Xuất file excel * Kết thúc Use case. * **Luồng sự kiện rẽ nhánh** * Luồng 1: * Tại giao diện Quản lý bệnh nhân, nhân viên quản lý có thể chọn chức năng Thêm, Xóa, Sửa hoặc Thoát. * Kết thúc Use case. * Luồng 2: * Hệ thông kiểm tra thông tin đăng nhập, nếu sai sẽ từ chối đăng nhập. * Hiển thị thông báo yêu cầu đăng nhập lại. * Nếu thành công thì đưa vào CSDL, ngược lại báo lỗi. * Kết thúc Use case. | |

## Xây dựng biểu đồ lớp (Class Diagram)

### Biểu đồ lớp

Các lớp được xây dựng trong chương trình



Hình 3. 40 Biểu đồ lớp

## Các bảng sử dụng trong chương trình

* Bảng User

Bảng 3. 1 Bảng User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu** | **Kích thước** | **Ràng buộc** | **Diễn giải** |
|  | username | char | 20 | Not null, Chính | Tài khoản |
|  | password | char | 100 |  | Mật khẩu |

* Bảng bệnh nhân

Bảng 3. 1 Bảng bệnh nhân

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu** | **Kích thước** | **Ràng buộc** | **Diễn giải** |
|  | MaBN | varchar | 5 | Not null, Chính | Mã bệnh nhân |
|  | TenBN | Nvarchar | 50 |  | Tên bệnh nhân |
|  | TUOI | Int |  |  | Tuổi |
|  | DiaChi | String |  |  | Địa Chỉ |
|  | CanNang | String |  |  | Cân Nặng |
|  | LoaiBenh | String |  |  | Loại bệnh |
|  | NgayDieuTri | Date |  |  | Ngày điều trị |

* Bảng loại thuốc

Bảng 3. 1 Bảng bệnh nhân

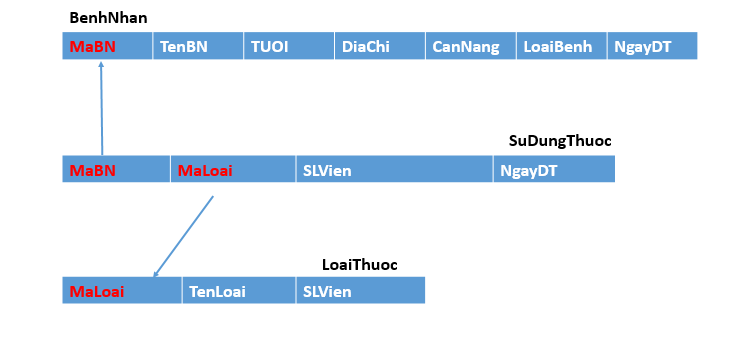
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu** | **Kích thước** | **Ràng buộc** | **Diễn giải** |
|  | MaLoai | Char | 5 | Not null, Chính | Mã loại thuốc |
|  | TenLoai | Nvarchar | 50 |  | Tên đầy đủ loại thuốc |
|  | SLVien | Int |  |  | Số lượng viên còn lại |

* Bảng sử dụng thuốc

Bảng 3. 1 Bảng sử dụng thuốc

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu** | **Kích thước** | **Ràng buộc** | **Diễn giải** |
| 1 | MaLoai | Char | 5 | Not null, Chính | Mã loại thuốc |
| 2 | MaBN | Char | 5 | Not null, Chính | Mã bệnh nhân |
| 3 | SLVien | Int |  |  | Số lượng viên của loại thuốc mà bệnh nhân dùng |
|  | NgayDT | Date |  |  | Ngày điều trị |

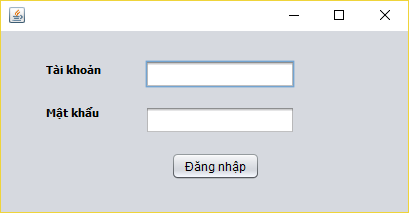
## Mô hình dữ liệu quan hệ



Bảng 3. 13 Mô hình dữ liệu quan hệ

# PHẦN MỀM THIẾT KẾ

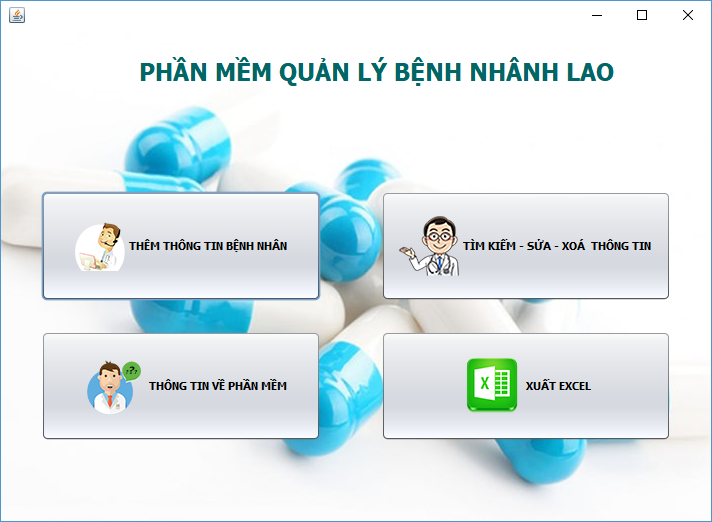
Khi người sử dụng chạy phầm mềm, giao diện đăng nhập sẽ hiển thị như hình 4.1, yêu cầu người sử dụng phải nhập tài khoản, mật khẩu. Nếu nhập đúng mới có thể đăng nhập phần mềm và sử dụng các chức năng khác của phần mềm, nếu nhập sai sẽ không đăng nhập được.



Hình 4. 1 Giao diện đăng nhập

* Tài khoản: Nhập tài khoản của người sử dụng.
* Mật khẩu: Nhập mật khẩu của người sử dụng.
* Chọn nút Đăng nhập để tiến hành đăng nhập.

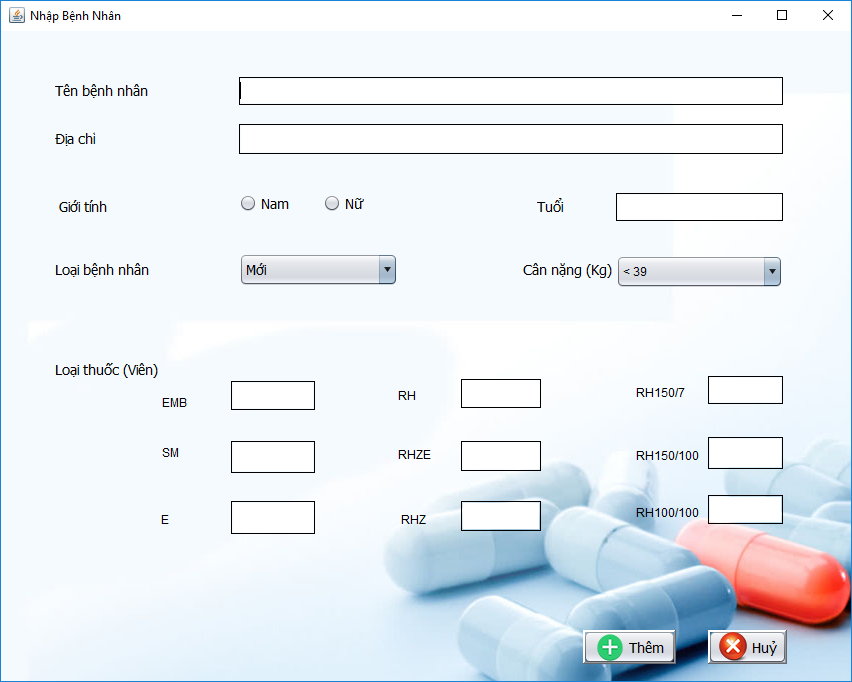
Khi đăng nhập thành công, giao diện chính của phần mềm sẽ như hình 4.2, người sử dụng có thể sử dụng các chức năng tại đây.



Hình 4. 2 Giao diện chính

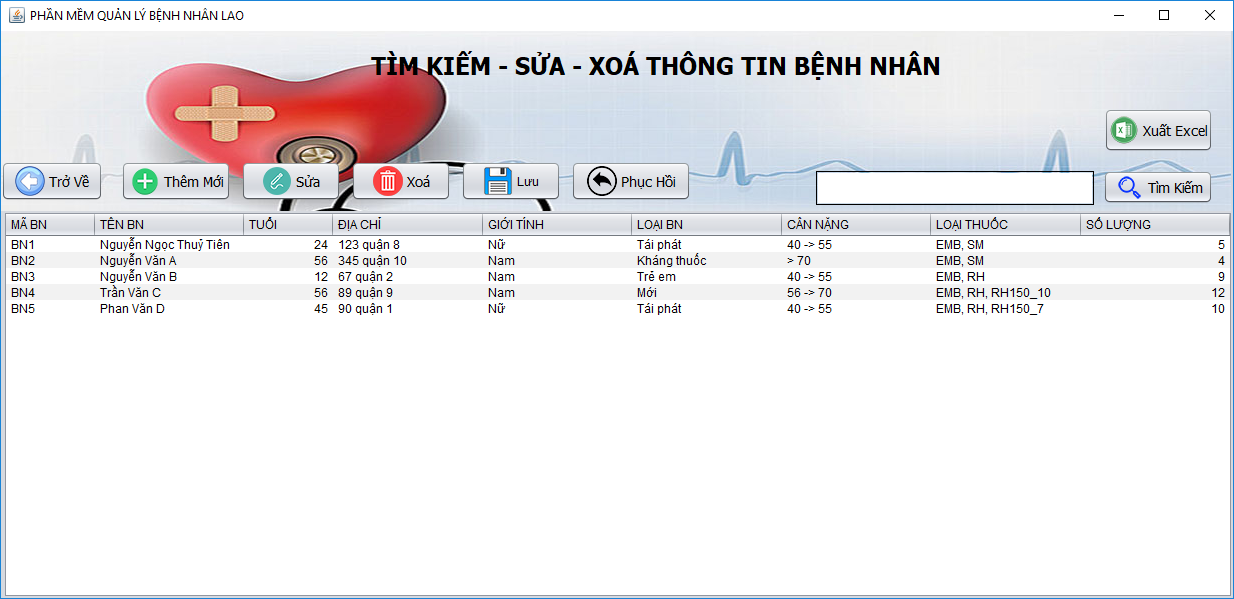
Có 4 tab chính là Thêm thông tin bệnh nhân, Tìm-xóa-sửa, Thông tin phần mềm, Xuất report excel

## Giao diện thêm thông tin bệnh nhân



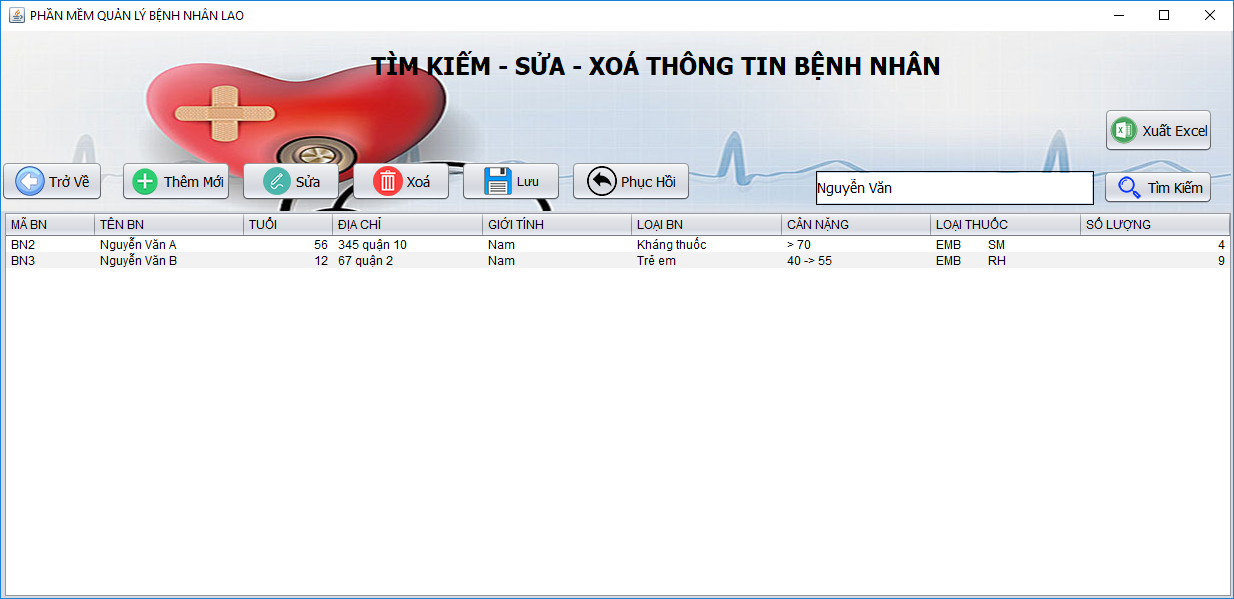
Hình 4. 3 Giao diện thêm thông tin bệnh nhân

## Giao diện Tìm – xóa – sửa.



Hình 4. 9 Giao diện tìm-xóa-sửa

### Chức năng tìm



Hình 4. 10 Giao diện tìm

Chức năng dùng để tìm kiếm các bệnh nhân theo tên:

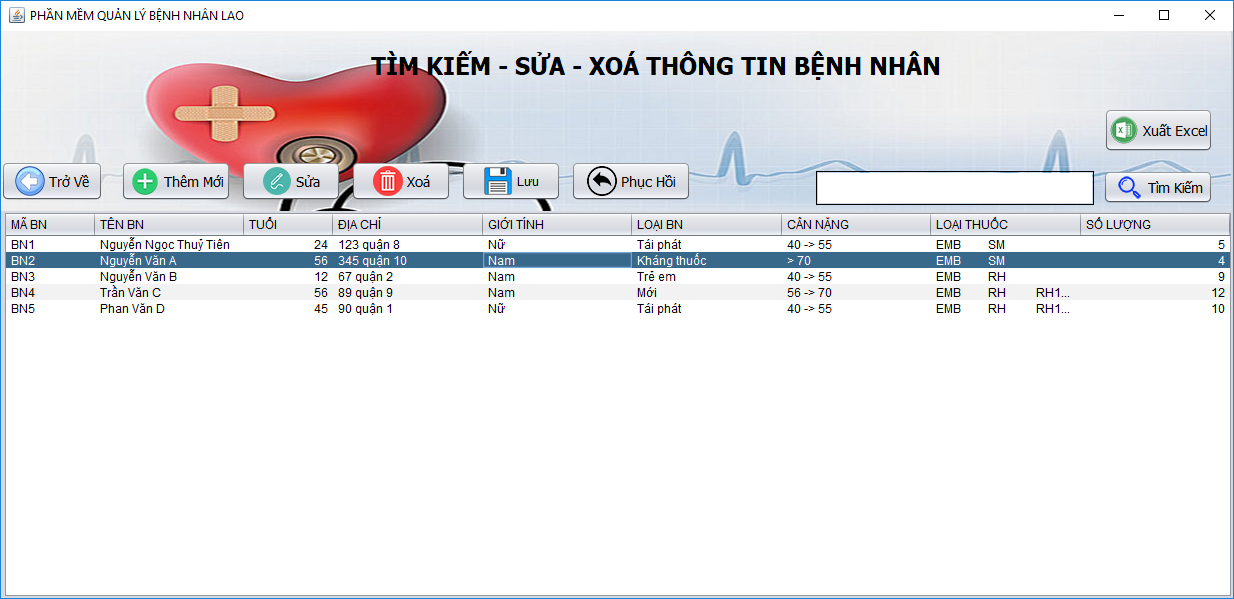
***Bước 1***: Điền đoạn chuỗi muốn tìm trong tên**.**

***Bước 2***: Ấn Tìm kiếm.

***Bước 3***: Form sẽ lọc và hiện thị kết quả.

.

### Chức năng xóa



Hình 4. 11 Giao diện xóa bệnh nhân

Chức năng cho phép xóa thông tin một bệnh nhân ra khỏi phần mềm.

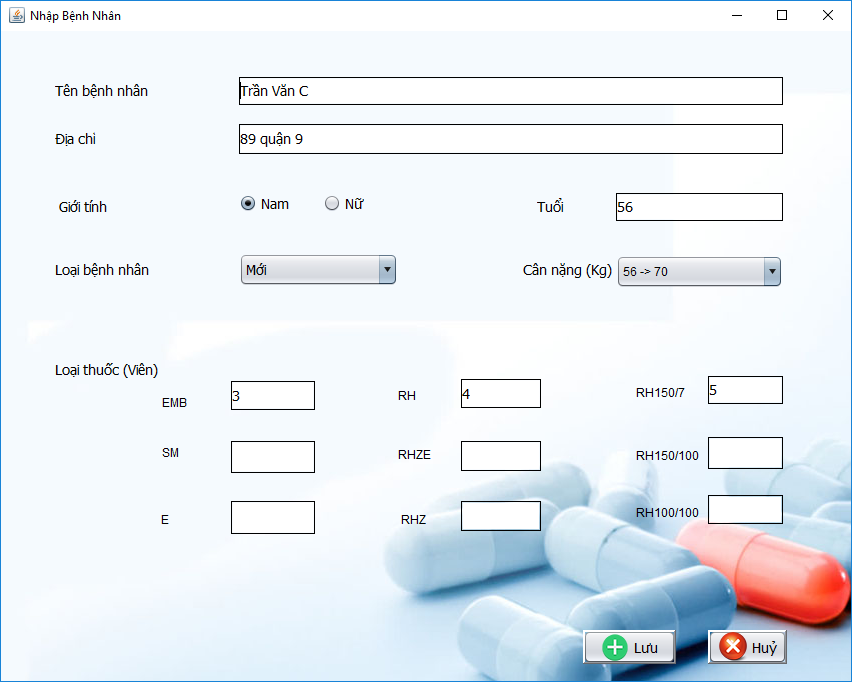
***Bước 1***: Chọn dòng của bệnh nhân cần xóa**.**

***Bước 2:*** Ấn button xóa.

***Bước 3:*** Khi đó form sẽ hiện thị danh sách sau khi xóa, có thể quay lại **Bước 1** để tiếp tục xóa tiếp.

***Bước 4:*** Nhấn button **Lưu** để cập nhật database các thao tác vừa xóa. Hoặc ấn **Phục hồi** để phục hồi các bước vừa xóa, khi chưa ấn **Lưu.**

### Chức năng Sửa thông tin bệnh nhân



Hình 4. 14 Giao diện sửa thông tin bệnh nhân

Chức năng cho phép chỉnh sửa thông tin một bệnh nhân và lưu lại xuống database.

***Bước 1***: Chọn dòng của bệnh nhân cần sửa**.**

***Bước 2***: Sửa thông tin cần sửa.

***Bước 3***: Ấn **Lưu** để cập nhật xuống database, hoặc ấn **Hủy** để hủy bỏ việc chỉnh sửa.

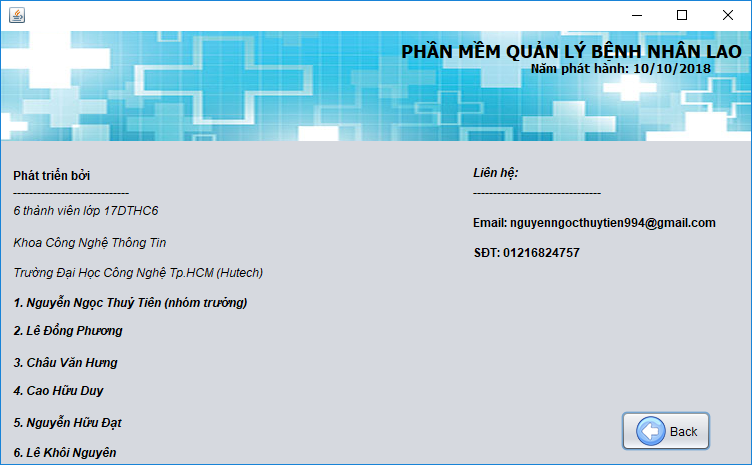
## Giao diện xuất excel



Hình 4. 20 Giao diện xuất excel

Chức năng cho phép xuất report thành file excel

## Giao diện Thông tin phần mềm



Hình 4. 28 Giao diện Thông tin phần mềm

# KẾT LUẬN

1. **Kết luận**

Trong quá trình thực hiện đề tài ***“Xây dựng phần mềm quản lý bệnh nhân lao”:***

Chúng em được tìm hiểu sâu về quy trình để phát triển một phần mềm. Bao gồm các phân tích cần thiết từ giao diện đến hệ thống thật chi tiết. Bên cạnh đó cũng rèn luyện kỹ năng làm việc nhóm, kỹ năng viết và trình bày báo cáo, kỹ năng đọc – nghiên cứu tài liệu có sẵn.

Chúng em mong đây là phần mềm sẽ thật sự hữu ích, hộ trợ tích cực cho các y bác sĩ trong công việc hằng ngày.

1. **Hướng phát triển**

Chúng em sẽ phát triển thêm các chức năng còn thiếu, nhằm đáp ứng đầy đủ, tiện lợi nhất. Cung cấp một ứng dụng hửu ích đến các y bác sĩ.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Tiếng Việt**

[1] Khóa luận tốt nghiệp – Xây dựng phần mềm quản lý Gas – Đoàn Hồng Thạnh – Phạm Thị Thu