**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**quản lý khám bệnh tại phòng khám**

***Nhóm 07 - Sinh viên thực hiện***

1. Tô Công Tuyển – 16013611
2. Phạm Thanh Duy – 16043751

# MỤC LỤC

[MỤC LỤC 1](#_Toc23539577)

[CHƯƠNG 1 : GIỚI THIỆU 4](#_Toc23539578)

[1.1 Tổng quan 4](#_Toc23539579)

[1.2 Mục tiêu đề tài 4](#_Toc23539580)

[1.3 Phạm vi đề tài 4](#_Toc23539581)

[1.4 Mô tả yêu cầu chức năng 5](#_Toc23539582)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 6](#_Toc23539583)

[2.1 Java EE 6](#_Toc23539584)

[2.2 MongoDB 8](#_Toc23539585)

[2.3 RMI (Remote Method Invocatin) 11](#_Toc23539586)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 13](#_Toc23539587)

[3.1 Phân tích yêu cầu bằng UML 13](#_Toc23539588)

[3.1.1 Usecase tổng quát 13](#_Toc23539589)

[3.1.2 Danh sách tác nhân và mô tả 14](#_Toc23539590)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 15](#_Toc23539591)

[3.1.4 Tình huống hoạt động 16](#_Toc23539592)

*[3.1.4.1 UC01 Nhập thông tin bệnh nhân.](#_Toc23539593)* [16](#_Toc23539593)

*[3.1.4.2 UC02 Sửa thông tin bệnh nhân.](#_Toc23539594)* [18](#_Toc23539594)

*[3.1.4.3 UC03 Tìm thông tin bệnh nhân.](#_Toc23539595)* [20](#_Toc23539595)

*[3.1.4.4 UC04 Kê toa thuốc.](#_Toc23539596)* [22](#_Toc23539596)

*[3.1.4.5 UC05 Quản lý bệnh nhân.](#_Toc23539597)* [24](#_Toc23539597)

[3.1.4.5.1 Thêm bệnh nhân. 24](#_Toc23539598)

[Use case: UC05.1\_Thêm bệnh nhân 24](#_Toc23539599)

[3.1.4.5.2 Sửa thông tin bệnh nhân. 26](#_Toc23539600)

*[3.1.4.6 UC06 Quản lý nhân viên.](#_Toc23539601)* [28](#_Toc23539601)

[3.1.4.6.1 Thêm nhân viên. 28](#_Toc23539602)

[Use case: UC06.1\_Thêm nhân viên 28](#_Toc23539603)

[3.1.4.6.2 Sửa nhân viên. 30](#_Toc23539604)

*[3.1.4.7 UC07 Quản lý thuốc.](#_Toc23539605)* [32](#_Toc23539605)

[3.1.4.7.1 Thêm thuốc 32](#_Toc23539606)

[5.1 Nếu thông tin người quản lý nhập vào không hợp lệ. Hệ thống hiển thị thông báo lỗi thông tin nhập không hợp lệ và yêu cầu nhân viên nhập lại thông tin không hợp lệ. 32](#_Toc23539607)

[5.2 Nếu mã thuốc người quản lý nhập vào trùng với mã thuốc đã có trong hệ thống. Hệ thống thông báo thuốc đã có trong hệ thống thuốc. 32](#_Toc23539608)

[3.1.4.7.2 Sửa thông tin thuốc 34](#_Toc23539609)

[3.1.4.7.3 Tìm kiếm thuốc 36](#_Toc23539610)

*[3.1.4.8 UC08 Đăng nhập.](#_Toc23539611)* [38](#_Toc23539611)

[4.1 Nếu nhập sai mật khẩu hoặc mã đăng nhập, hệ thống thông báo nhập sai mã đăng nhập hoặc mật khẩu và yêu cầu nhân viên nhập lại mã đăng nhập hoặc mật khẩu. 38](#_Toc23539612)

[3.2 Class diagram 40](#_Toc23539613)

[3.3 Deployment diagram 41](#_Toc23539614)

[CHƯƠNG 4 : HIỆN THỰC 42](#_Toc23539615)

[4.1 Cấu hình phần cứng, phần mềm 42](#_Toc23539616)

[4.2 Giao diện của hệ thống 43](#_Toc23539617)

[4.2.1 Đăng nhập 43](#_Toc23539618)

[4.2.2 Giao diện chính 43](#_Toc23539619)

[4.2.3 Giao diện quản lý thuốc 43](#_Toc23539620)

[4.2.4 Giao diện quản lý nhân viên 43](#_Toc23539621)

[4.2.5 Giao diện quản lý bệnh nhân 43](#_Toc23539622)

[4.2.6 Giao diện kê toa thuốc 44](#_Toc23539623)

[CHƯƠNG 5 : KẾT LUẬN 44](#_Toc23539624)

[5.1 Kết quả đạt được 44](#_Toc23539625)

[5.2 Hạn chế của đồ án 44](#_Toc23539626)

[5.3 Hướng phát triển 44](#_Toc23539627)

[TÀI LIỆU THAM KHẢO 45](#_Toc23539628)

# : GIỚI THIỆU

## Tổng quan

Sự xuất hiện của Tin học đã tạo một bước đột phá rất lớn trong sự phát triển của khoa học và công nghệ. Chính vì vậy, ngày nay, trong tất cả các ngành, các lĩnh vực ở mọi cấp độ và quy mô khác nhau vai trò ứng dụng của Tin học là vô cùng quan trọng và tất yếu. Trong lĩnh vực Y tế, các phòng mạch tư ngày càng mở rộng về số lượng lẫn quy mô. Chính vì thế, công tác quản lý các thông tin liên quan đến bệnh nhân cũng như việc khám chữa bệnh các phòng mạch tư là rất cần thiết. Việc ứng dụng Tin học trong quá trình xử lý, giải quyết hiệu quả vấn đề trên là không thể thiếu. Đó chính là nguyên nhân xuất phát ý tưởng cần phải xây dựng một phần mềm quản lý phòng mạch tư để đáp ứng hiệu quả với nhu cầu thực tế được đặt ra như trên.

## Mục tiêu đề tài

* Xây dựng ứng dụng giúp nhân viên nhận thông tin từ bệnh nhân gửi đến bác sĩ.
* Bác sĩ nhận thông tin bệnh nhân từ nhân viên và kê toa thuốc cho bệnh nhân.
* Người quản lý thực hiện thêm, xoá, sửa bệnh nhân, nhân viên.

## Phạm vi đề tài

Ứng dụng được sử dụng trong phạm vi quản lý khám bệnh của một bệnh viện, thực hiện các chức năng quản lý thuốc, quản lý nhân viên, nhập thông tin bệnh nhân, sử thông tin bênh nhân, tìm thông tin bệnh nhân, kê toa thuốc đối tượng phục vụ chính là nhân viên và bác sĩ kê toa thuốc.

Thực hiện với các công nghệ:

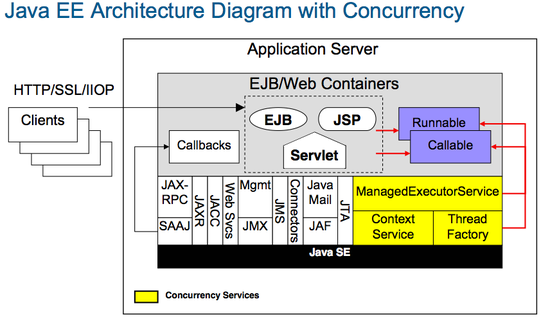
* Sử dụng JavaEE để làm giao diện cho ứng dụng.
* Sử dụng hệ cơ sở dữ liệu MongoDB để tạo cơ sở dữ liệu cho ứng dụng.
* Triển khai ứng dụng với RMI (Remote Method Invocation) và JPA (Java Persistence API).

## Mô tả yêu cầu chức năng

Nhân viên nhận bệnh: Đăng nhập hệ thống, xem thông tin của mình, đổi mật khẩu khi cần Nhập thông tin bệnh nhân khi đến khám và đưa vào danh sách khám của bác sĩ mà bệnh nhân chọn. (nếu bệnh nhân đã khám thì không cần nhập lại). Sửa thông tin bệnh nhân khi có yêu cầu từ bệnh nhân. Tìm thông tin bệnh nhân khi biết mã hoặc tên.b. Bác sĩ: Đăng nhập hệ thống, xem thông tin của mình, đổi mật khẩu khi cần Nhận bệnh nhân từ danh sách chờ. Thực hiện khám bệnh bao gồm: ghi nhận thông tin tình trạng của bệnh nhân, cho toa thuốc.Nếu bệnh nhân cần xét nghiệm để biết thêm thông tin thì BS sẽ chuyển qua bộ phận xét nghiệm. Khi thực hiện xong xét nghiệm, bệnh nhân sẽ quay về phòng BS để cập nhật các thông tin xét nghiệm. Nhân viên phát thuốc: Đăng nhập hệ thống, xem thông tin của mình, đổi mật khẩu khi cần Nhận danh sách bệnh nhân đã khám và phát thuốc dựa trên toa thuốc bác sĩ đã kê trước đó. Quản lý: Quản lý thông tin bệnh nhân: xem, thêm. Quản lý các nhân viên tham gia hệ thống. Quản lý các toa thuốc.

# : CƠ SỞ LÝ THUYẾT

## Java EE

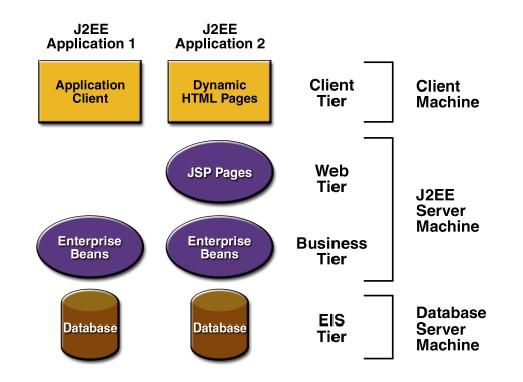
"Java EE" tên đầy đủ là "Java Enterprise Edition" dịch sang tiếng việt có nghĩa là là "Java Phiên Bản Doanh Nghiệp", nó là một nền tảng (platform) dành cho việc xây dựng các ứng dụng (application) cấp doanh nghiệp.

*Hình 2-1: Kiến trúc javaEE*

Java EE tập hợp nhiều công nghệ kỹ thuật và phối hợp chúng với nhau, việc này sẽ làm giảm đáng kể chi phí và độ phức tạp cho việc xây dựng, phát triển, triển khai, quản lý các tầng làm việc (managing multi-tier), cho các ứng dụng máy chủ trung tâm.

Java EE được xây dựng dưa trên nền tảng Java SE và có thêm các tập APIs (application programming interfaces) để phát triển các ứng dựng phái máy chủ một cách nhanh chóng(running), linh động(portable), mạnh mẽ(rob), khả năng mở rộng cao, đáng tinh cậy và bảo mật.

Hệ nền JEE sử dụng một mô hình ứng dụng theo kiểu đa tầng cho các ứng dụng doanh nghiệp. Về mặt logic ứng dụng được chia theo các thành phần theo chức năng, và các thành phần ứng dụng khác nhau tạo nên một ứng dụng JEE được cài đặt trên các máy khác nhau tùy thuộc vào các tầng trong môi trường phân tầng của JEE nơi mà các thành phần ứng dụng thuộc về.



*Hình 2-2: Kiến trúc javaEE theo kiểu đa tầng*

## MongoDB



*Hình 2‑3 NoSQL với MongoDB*

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là cơ sở dữ liệu thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng

So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

Cấu trúc lệnh cơ bản trên MongoDB so với MySQL

| **CSDL** | **MySQL** | **MongoDB** |
| --- | --- | --- |
| Tạo CSDL | CREATE DATABASE test; | use test; |
| Tạo bảng | CREATE TABLE students (ten\_cot - kieu\_du\_lieu); | db.createCollection('students'); |
| Tạo bản ghi | INSERT INTO studetns ('name', 'gender') VALUES('thanh', 'male'); | db.students.insert({ name:'thanh', gender: 'male'}); |
| Cập nhật | UPDATE students SET name = 'thanh update' WHERE id = 1; | db.students.update({ \_id: 1 },{$set:{ name: 'thanh update' }}); |
| Xóa bản ghi | DELETE FROM students Where id = 1; | db.students.remove({ \_id: 1}); |
| Tìm kiếm all | SELECT \* FROM students; | db.students.find({}); |
| Tìm kiếm | SELECT \* FROM students WHERE name = 'thanh'; | db.students.find({ name: 'thanh' }); |

*Bảng 2-1: So sánh cấu trúc lệnh giữa MongoDB và MySQL*

* Ưu điểm của MongoDB:
  + Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.
  + Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.
  + MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster.
  + Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
  + Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
  + Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB sẽ nhanh hơn so với MySQL.
* Nhược điểm:
* Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.
* Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.
* Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điêù này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

## RMI (Remote Method Invocatin)

RMI - Remote Method Invocation là một kĩ thuật cài đặt các đối tượng phân tán trong Java. RMI là một phần của bộ J2SDK và là hàm thư viện hỗ trợ các lời gọi phương thức từ xa và trả về giá trị cho các ứng dụng tính toán phân tán. Chúng ta giả sử rằng ngôn ngữ Java được sử dụng ở cả hai phía gọi và phía bên phương thức được gọi.

* Đặc tính của RMI:

+ RMI là mô hình đối tượng phân tán của Java, RMI giup cho việc giao tiếp giữa các đối tượng phân tán trong môi trường internet trể nên dễ dàng hơn.

+ RMI là API bậc cao được xây dựng dựa trên lập trình Socket.

+ RMI không những cho phép chúng ta truyền dữ liệu giữa các đối tượng trên các hệ thống máy tính khác nhau, mà còn triệu gọi các phương thức trong các đối tượng ở xa (Remote Object).

+ Việc truyền dữ liệu giữa các máy khác nhau được xử lý một cách trong suốt bởi máy ảo Java (Java virtual machine).

+ Tương tự như mô hình Client/Server, RMI vẫn lấy/duy trì khái niệm của Client và Server, tuy nhiên cách tiếp cận (approach) của RMI linh hoạt hơn, mềm dẻo hơn so với môt hình Client/Server.

+ Một điều thuận lợi quan trong nhất của RMI là nó cung cấp cớ chế callbacks, nó cho phép Server triệu gọi các phương thức ở Client.

* Truyền tin trong RMI:

+ RMI sử dụng lớp trung gian để truyền tin Skeleton và Stub

+ Lớp Stub dùng ở client

+ Lớp Skeleton dùng ở Server

+ Java sử dụng rmic.exe để tạo ra các lớp trung gian

+ TCP Socket

* Cách thức hoạt động của RMI:

+ Server RMI phải đăng ký với 1 dịch vụ tra tìm và đăng ký tên miền.

+ Sau khi server được đăng ký, nó sẽ chờ các yêu cầu của RMI client.

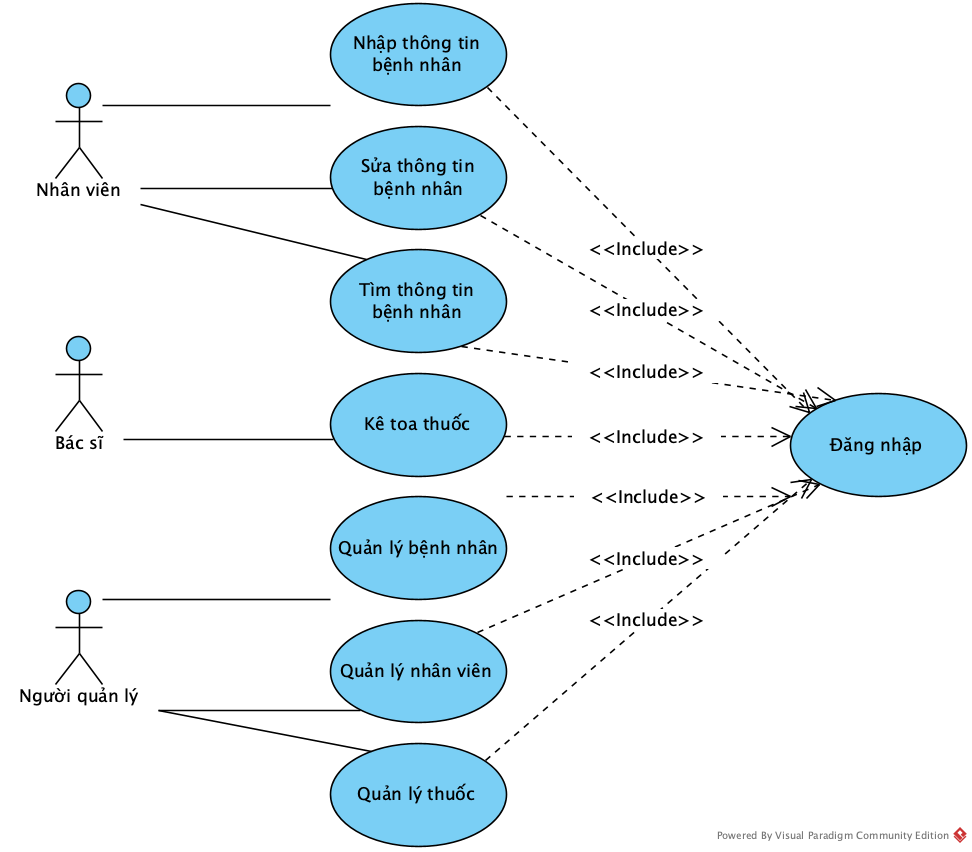
+ Các client RMI sẽ gửi thông điệp RMI để gọi một phương thức trên một đối tượng từ xa.

+ Ứng dụng client yêu cần một tên dịch vụ cụ thể và nhận một URL trỏ tới tài nguyên từ xa.

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát

******

*Hình 3-1: Use case tổng quát*

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Nhân viên nhận bệnh | Nhập thông tin bệnh nhân khi đến khám và đưa vào danh sách khám của bác sĩ mà bệnh nhân chọn.  Sửa thông tin bệnh nhân khi có yêu cầu từ bệnh nhân.  Tìm thông tin bệnh nhân khi biết mã hoặc tên. |
| Nhân viên phát thuốc | Nhận danh sách bệnh nhân đã khám và phát thuốc dựa trên toa thuốc bác sĩ đã kê trước đó. |
| Bác sĩ | Nhận bệnh nhân từ danh sách chờ.  Thực hiện khám bệnh bao gồm: ghi nhận thông tin tình trạng của bệnh nhân, cho toa thuốc. |
| Người quản lý | Quản lý thông tin bệnh nhân: xem, thêm, tìm kiếm bệnh nhân.  Quản lý các nhân viên tham gia hệ thống.  Quản lý các toa thuốc. |

*Bảng 3‑1 Danh sách tác nhân và mô tả*

### Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Nhập thông tin bệnh nhân |
| UC02 | Sửa thông tin bệnh nhân |
| UC03 | Tìm thông tin bệnh nhân |
| UC04 | Kê toa thuốc |
| UC05 | Quản lý bệnh nhân |
| UC06 | Quản lý nhân viên |
| UC07 | Quản lý bác sĩ |
| UC08 | Đăng nhập |

*Bảng 3‑2 Danh sách các tình huống trong hệ thống*

### Tình huống hoạt động

#### *UC01 Nhập thông tin bệnh nhân.*

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case:** UC01\_Nhập thông tin bệnh nhân | |
| Mục đích: | Thêm thông tin bệnh nhân. |
| Mô tả: | Use case thực hiện việc thêm thông tin bệnh nhân. |
| Tác nhân: | Nhân viên nhận bệnh. |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng quyền nhân viên. |
| Điều kiện sau: | Lưu thông tin bệnh nhân nếu chưa có trong hệ thống |
| Luồng sự kiện chính (Basic flows) | 1. Nhân viên nhập mã bệnh nhân 2. Hệ thống kiểm tra thông tin 3. Hệ thống hệ hiện thông tin bệnh nhân. |
| Luồng sự kiện phụ (Alternative Flows): | * 1. Hệ thông báo không tìm thấy.   2. Nhân viên nhận bệnh nhập thông tin bệnh nhân.   3. Chọn lưu. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3‑3 đặc tả nhập thông tin bệnh nhân*

Activity diagram.



*Hình 3-2: activity nhập thông tin bệnh nhân*

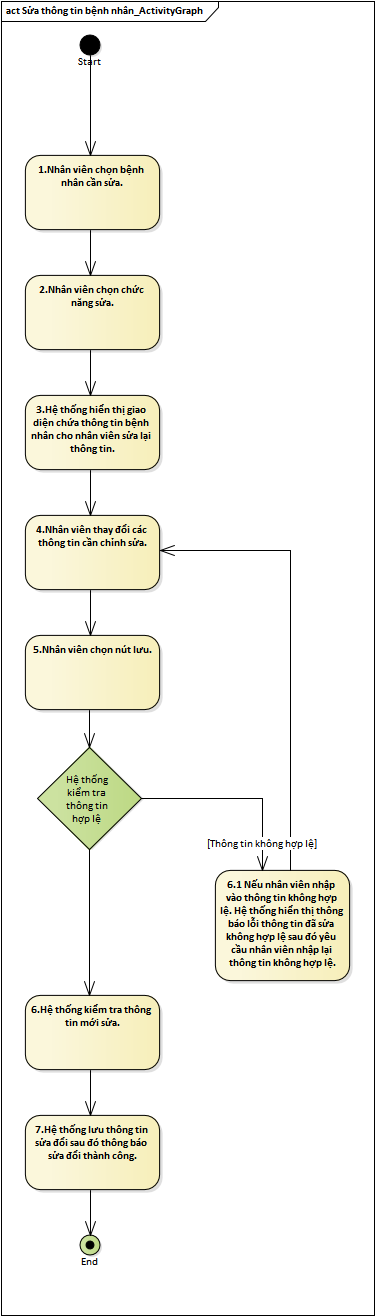
#### *UC02 Sửa thông tin bệnh nhân.*

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case: UC02\_Sửa thông tin bệnh nhân.** | |
| Mục đích: | Chỉnh sửa và lưu lại thông tin bệnh nhân vào hệ thống của quầy thuốc. |
| Mô tả: | Nhân viên thực hiện các thay đổi về thông tin bệnh nhân có trong hệ thống sau đó lưu lại các thông tin đã thay đổi. |
| Tác nhân: | Nhân viên nhận bệnh |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng nhân viên nhân viên và đã thực hiện use case tìm kiếm bệnh nhân. |
| Điều kiện sau: | Thông tin của bệnh nhân được cập nhật thành công trong hệ thống. |
| Luồng sự kiện chính (Basic flows) | 1. Nhân viên chọn bệnh nhân cần sửa. 2. Nhân viên chọn chức năng sửa. 3. Hệ thống hiển thị giao diện chứa thông tin bệnh nhân cho nhân viên sửa lại thông tin. 4. Nhân viên thay đổi các thông tin cần chỉnh sửa. 5. Nhân viên chọn nút lưu. 6. Hệ thống kiểm tra thông tin mới sửa. 7. Hệ thống lưu thông tin sửa đổi sau đó thông báo **s**ửa đổi thành công. |
| Luồng sự kiện phụ (Alternative Flows): | Không có. |
| Luồng sự kiện ngoại lệ (Exeption Flows): | 6.1 Nếu nhân viên nhập vào thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi thông tin đã sửa không hợp lệ sau đó yêu cầu nhân viên nhập lại thông tin không hợp lệ. Quay lại bước |

*Bảng 3-4 đặc tả sửa thông tin bệnh nhân*

Activity diagram.



*Hình 3-3: activity sử thông tin bệnh nhân*

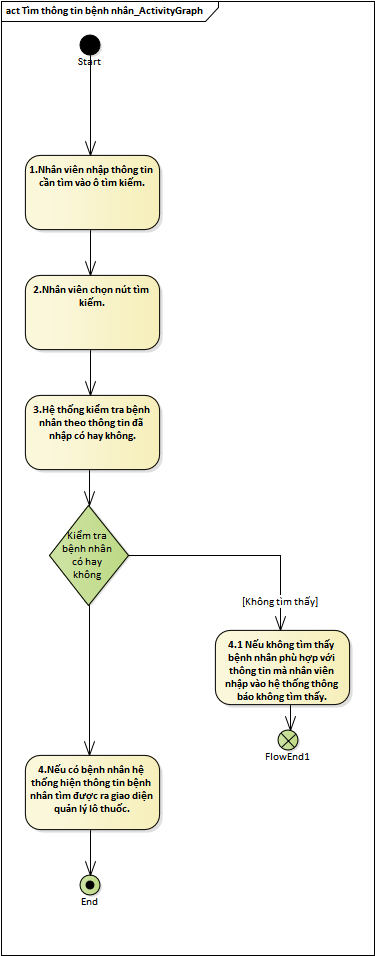
#### *UC03 Tìm thông tin bệnh nhân.*

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case:** UC03\_Tìm kiếm bệnh nhân | |
| Mục đích: | Tìm kiếm thông tin bệnh nhân có trong hệ thống. |
| Mô tả: | Use case thực hiện việc tìm kiếm thông tin bệnh nhân có trong hệ thống bằng cánh dùng mã, hoặc tên bệnh nhân để tìm kiếm. |
| Tác nhân: | Nhân viên nhận bệnh. |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng nhân viên nhân viên và đang ở trong giao diện quản lý bệnh nhân. |
| Điều kiện sau: | Tìm thành công và hiện ra thông tin bệnh nhân được tìm thấy. |
| Luồng sự kiện chính (Basic flows) | 1. Nhân viên nhập thông tin cần tìm vào ô tìm kiếm. 2. Nhân viên chọn nút tìm kiếm. 3. Hệ thống kiểm tra bệnh nhân theo thông tin đã nhập có hay không. 4. Nếu có bệnh nhân hệ thống hiện thông tin bệnh nhân tìm được ra giao diện quản lý lô thuốc. |
| Luồng sự kiện phụ (Alternative Flows): | 4.1 Nếu không tìm thấy bệnh nhân phù hợp với thông tin mà nhân viên nhập vào hệ thống thông báo không tìm thấy. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3-5: đặc tả tìm kiếm bệnh nhân*

Activity diagram.



*Hình 3-4: activity tìm kiếm thông tin bệnh nhân*

#### *UC04 Kê toa thuốc.*

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case:** UC04\_Kê toa thuốc | |
| Mục đích: | Thêm thông tin toa thuốc cho bệnh nhân. |
| Mô tả: | Use case thực hiện việc nhập thông tin toa thuốc cho bệnh nhân. |
| Tác nhân: | Bác sĩ. |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng quyền quản lý bác sĩ. |
| Điều kiện sau: | Lưu thông tin toa thuốc cho bệnh nhân |
| Luồng sự kiện chính (Basic flows) | 1. Bác sĩ nhận thông tin bệnh nhân từ danh sách chờ. 2. Bác sĩ nhập thông tin toa thuốc cho bệnh nhân. 3. Hệ thống kiểm tra dữ liệu hợp lệ 4. Hệ thống thông báo lưu thành công. |
| Luồng sự kiện phụ (Alternative Flows): | 4.1 Hệ thống báo lỗi dữ liệu không hợp lệ. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3-6: đặc tả kê toa thuốc*

Activity diagram.



*Hình 3-5: activity kê toa thuốc*

#### *UC05 Quản lý bệnh nhân.*

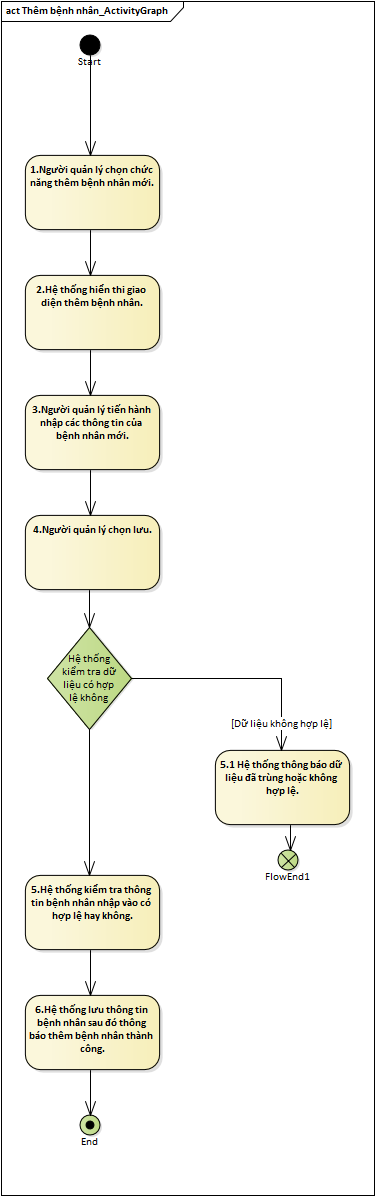
##### Thêm bệnh nhân.

Đặc tả use case.

|  |  |
| --- | --- |
| Use case: UC05.1\_Thêm bệnh nhân | |
| Mục đích: | Thêm bệnh nhân mới vào hệ thống. |
| Mô tả: | Use case thực hiện việc thêm bệnh nhân mới vào hệ thống |
| Tác nhân: | Người quản lý. |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng nhân viên người quản lý. |
| Điều kiện sau: | Các thông tin của bệnh nhân mới được thêm thành công vào hệ thống. |
| Luồng sự kiện chính (Basic flows) | 1. Người quản lý chọn chức năng thêm bệnh nhân mới. 2. Hệ thống hiển thi giao diện thêm bệnh nhân. 3. Người quản lý tiến hành nhập các thông tin của bệnh nhân mới. 4. Người quản lý chọn lưu. 5. Hệ thống kiểm tra thông tin bệnh nhân nhập vào có hợp lệ hay không. 6. Hệ thống lưu thông tin bệnh nhân sau đó thông báo thêm bệnh nhân thành công. |
| Luồng sự kiện phụ (Alternative Flows): | 5.1 Hệ thống thông báo dữ liệu đã trùng hoặc không hợp lệ. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3-7: đặc tả thêm bệnh nhân*

Activity diagram.



*Hình 3-6: activity thêm bệnh nhân*

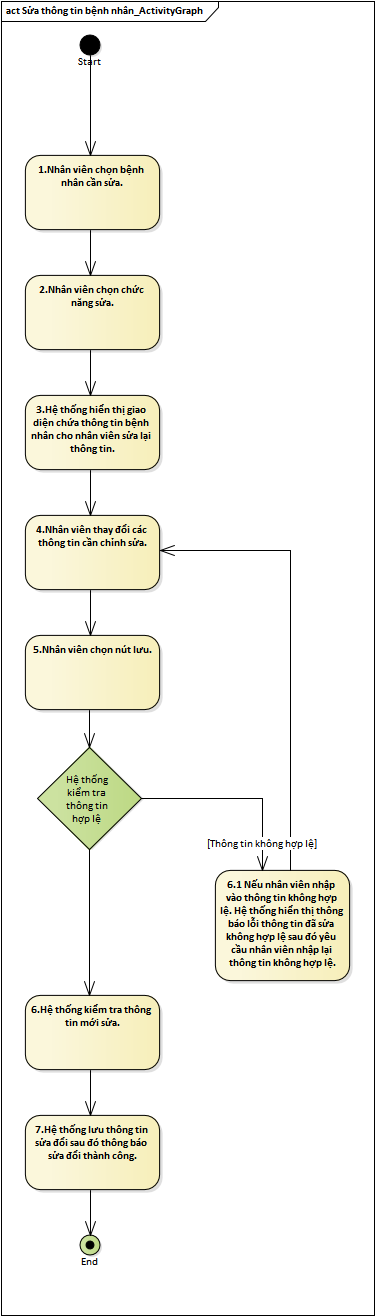
##### Sửa thông tin bệnh nhân.

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case: UC05.2\_Sửa thông tin bệnh nhân.** | |
| Mục đích: | Chỉnh sửa và lưu lại thông tin bệnh nhân vào hệ thống của quầy thuốc. |
| Mô tả: | Người quản lý thực hiện các thay đổi về thông tin bệnh nhân có trong hệ thống sau đó lưu lại các thông tin đã thay đổi. |
| Tác nhân: | Người quản lý |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng nhân viên Người quản lý và đã thực hiện use case tìm kiếm bệnh nhân. |
| Điều kiện sau: | Thông tin của bệnh nhân được cập nhật thành công trong hệ thống. |
| Luồng sự kiện chính (Basic flows) | 1. Người quản lý chọn bệnh nhân cần sửa.  2. Người quản lý chọn chức năng sửa.  3. Hệ thống hiển thị giao diện chứa thông tin bệnh nhân cho nhân viên sửa lại thông tin.  4. Người quản lý thay đổi các thông tin cần chỉnh sửa.  5. Người quản lý chọn nút lưu.  6. Hệ thống kiểm tra thông tin mới sửa.  7. Hệ thống lưu thông tin sửa đổi sau đó thông báo **s**ửa đổi thành công. |
| Luồng sự kiện phụ (Alternative Flows): | Không có. |
| Luồng sự kiện ngoại lệ (Exeption Flows): | 6.1 Nếu nhân viên nhập vào thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi thông tin đã sửa không hợp lệ sau đó yêu cầu nhân viên nhập lại thông tin không hợp lệ. Quay lại b4 |

*Bảng 3-8: đặc tả sửa thông tin bệnh nhân*

Activity diagram.



*Hình 3-7: activity sử thông tin bệnh nhân*

#### *UC06 Quản lý nhân viên.*

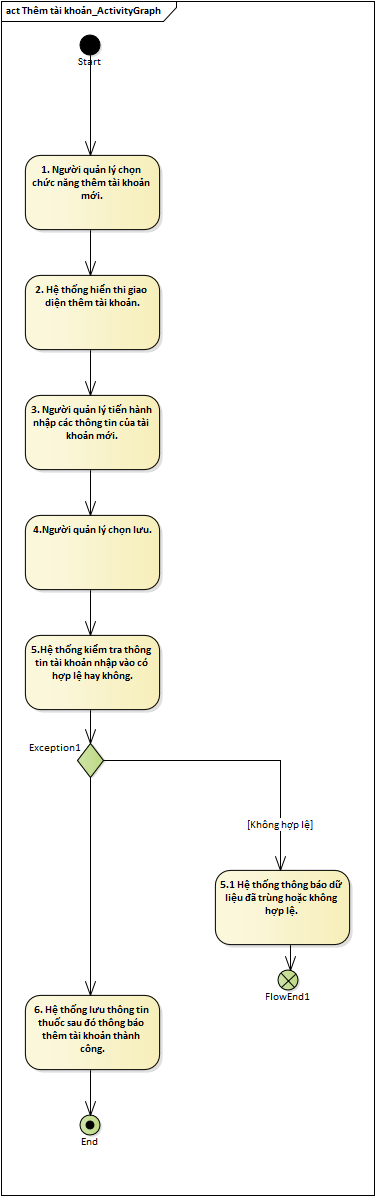
##### Thêm nhân viên.

Đặc tả use case.

|  |  |
| --- | --- |
| Use case: UC06.1\_Thêm nhân viên | |
| Mục đích: | Thêm nhân viên mới vào hệ thống. |
| Mô tả: | Use case thực hiện việc thêm nhân viên mới vào hệ thống |
| Tác nhân: | Người quản lý. |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng nhân viên người quản lý. |
| Điều kiện sau: | Các thông tin của nhân viên mới được thêm thành công vào hệ thống. |
| Luồng sự kiện chính (Basic flows) | 1. Người quản lý chọn chức năng thêm nhân viên mới.  2. Hệ thống hiển thi giao diện thêm nhân viên.  3. Người quản lý tiến hành nhập các thông tin của nhân viên mới.  4.Người quản lý chọn lưu.  5.Hệ thống kiểm tra thông tin nhân viên nhập vào có hợp lệ hay không.  6. Hệ thống lưu thông tin thuốc sau đó thông báo thêm nhân viên thành công. |
| Luồng sự kiện phụ (Alternative Flows): | 5.1 Hệ thống thông báo dữ liệu đã trùng hoặc không hợp lệ. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3-9: đặc tả thêm nhân viên*

Activity diagram.



*Hình 3-8: activity thêm nhân viên*

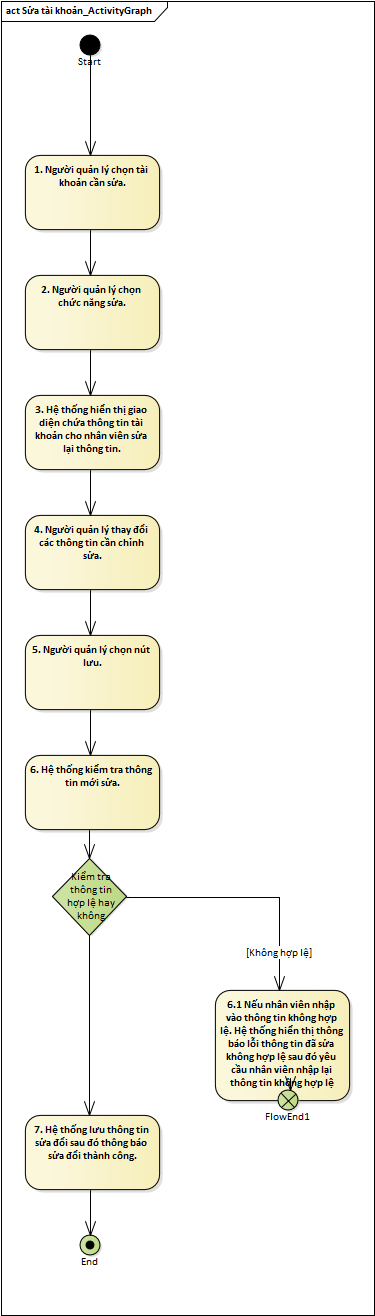
##### Sửa nhân viên.

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case: UC06.2\_Sửa thông tin nhân viên.** | |
| Mục đích: | Chỉnh sửa và lưu lại thông tin nhân viên vào hệ thống của quầy thuốc. |
| Mô tả: | Người quản lý thực hiện các thay đổi về thông tin nhân viên có trong hệ thống sau đó lưu lại các thông tin đã thay đổi. |
| Tác nhân: | Người quản lý |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng nhân viên Người quản lý và đã thực hiện use case tìm kiếm nhân viên. |
| Điều kiện sau: | Thông tin của nhân viên được cập nhật thành công trong hệ thống. |
| Luồng sự kiện chính (Basic flows) | 1. Người quản lý chọn nhân viên cần sửa.  2. Người quản lý chọn chức năng sửa.  3. Hệ thống hiển thị giao diện chứa thông tin nhân viên cho nhân viên sửa lại thông tin.  4. Người quản lý thay đổi các thông tin cần chỉnh sửa.  5. Người quản lý chọn nút lưu.  6. Hệ thống kiểm tra thông tin mới sửa.  7. Hệ thống lưu thông tin sửa đổi sau đó thông báo **s**ửa đổi thành công. |
| Luồng sự kiện phụ (Alternative Flows): | Không có. |
| Luồng sự kiện ngoại lệ (Exeption Flows): | 6.1 Nếu nhân viên nhập vào thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi thông tin đã sửa không hợp lệ sau đó yêu cầu nhân viên nhập lại thông tin không hợp lệ. |

*Bảng 3-10: đặc tả sửa nhân viên*

Activity diagram.



*Hình 3-9: activity sửa nhân viên*

#### *UC07 Quản lý thuốc.*

##### Thêm thuốc

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case:** UC07.1\_Thêm thuốc | |
| Mục đích: | Thêm thành công thuốc mới vào hệ thống. |
| Mô tả: | Khi muốn thêm thuốc mới, nhân viên tiến hành nhập thông tin thuốc đó vào hệ thống. |
| Tác nhân: | Người quản lý. |
| Điều kiện trước: | Ở giao diện quản lý thuốc. |
| Điều kiện sau: | Các thông tin của thuốc mới được thêm thành công vào hệ thống thuốc. |
| Luồng sự kiện chính (Basic flows) | 1. Người quản lý chọn chức năng thêm thuốc mới. 2. Hệ thống cho phép nhân viên nhập thông tin. 3. Người quản lý tiến hành nhập các thông tin của thuốc mới: mã thuốc, tên thuốc, nhóm thuốc, thành phần thuốc. 4. Người quản lý nhấn lưu. 5. Hệ thống kiểm tra thông tin nhân viên nhập vào có hợp lệ hay không. 6. Nếu thông tin hợp lệ hệ thống lưu thông tin thuốc sau đó thông báo thêm thuốc thành công. |
| Luồng sự kiện phụ (Alternative Flows): | 5.1 Nếu thông tin người quản lý nhập vào không hợp lệ. Hệ thống hiển thị thông báo lỗi thông tin nhập không hợp lệ và yêu cầu nhân viên nhập lại thông tin không hợp lệ.5.2 Nếu mã thuốc người quản lý nhập vào trùng với mã thuốc đã có trong hệ thống. Hệ thống thông báo thuốc đã có trong hệ thống thuốc. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3-11: đặc tả thêm thuốc*

Activity diagram.

****

*Hình 3-10: activity thêm thuốc*

##### Sửa thông tin thuốc

Đặc tả use case

|  |  |
| --- | --- |
| **Use case: UC07.2\_Sửa thông tin thuốc.** | |
| Mục đích: | Chỉnh sửa và lưu lại thông tin thuốc vào hệ thống thuốc của quầy thuốc. |
| Mô tả: | Nhân viên thực hiện các thay đổi về thông tin thuốc có trong hệ thống sau đó lưu lại các thông tin thuốc đã thay đổi. |
| Tác nhân: | Người dùng |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng tài khoản nhân viên và đã thực hiện use case tìm kiếm thuốc. |
| Điều kiện sau: | Thông tin của thuốc được cập nhật thành công trong hệ thống. |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng chọn thuốc cần sửa. 2. Người dùng chọn chức năng sửa. 3. Hệ thống cho phép nhân viên sửa lại thông tin. 4. Người dùng thay đổi các thông tin cần chỉnh sửa. 5. Người dùng chọn nút lưu. 6. Hệ thống kiểm tra thông tin mới sửa. 7. Nếu thông tin hợp lệ hệ thống lưu thông tin sửa đổi sau đó thông báo **s**ửa đổi thành công. |
| Luồng sự kiện phụ (Alternative Flows): | Không có. |
| Luồng sự kiện ngoại lệ (Exception Flows): | 6.1 Nếu người dùng nhập vào thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi thông tin đã sửa không hợp lệ sau đó yêu cầu nhân viên nhập lại thông tin không hợp lệ. |

*Bảng 3-12: đặc tả sửa thông tin thuốc*

Activity diagram



*Hình 3-11: activity sửa thông tin thuốc*

##### Tìm kiếm thuốc

Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** UC004\_Tìm kiếm thuốc | |
| Mục đích: | Tìm kiếm thông tin thuốc có trong hệ thống. |
| Mô tả: | Use case thực hiện tìm kiếm thông tin thuốc có trong hệ thống thuốc của quầy thuốc. Nhân viên có thể dùng mã thuốc hoặc tên thuốc để tìm kiếm. |
| Tác nhân: | Người dùng |
| Điều kiện trước: | Phải đăng nhập vào hệ thống bằng tài khoản nhân viên và đang ở trong giao diện quản lý thuốc. |
| Điều kiện sau: | Tìm thành công và hiện ra thông tin thuốc được tìm thấy. |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng nhập thông tin cần tìm vào ô tìm kiếm. 2. Người dùng chọn nút tìm kiếm. 3. Hệ thống kiểm tra thuốc theo thông tin đã nhập xem có hay không. 4. Nếu có thuốc hệ thống hiện thông tin thuốc tìm được ra giao diện quản lý thuốc. |
| Luồng sự kiện phụ (Alternative Flows): | 4.1 Nếu không tìm thấy thuốc phù hợp với thông tin mà người dùng nhập vào hệ thống thông báo không tìm thấy. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

*Bảng 3-13: đặc tả tìm kiếm thuốc*

Activity diagram



*Hình 3-12: activity tìm kiếm thuốc*

#### *UC08 Đăng nhập.*

Đặc tả use case.

|  |  |
| --- | --- |
| **Use case:** *UC09\_Đăng nhập* | |
| Mục đích: | Đăng nhập vào hệ thống để thực hiện các chức năng của ứng dụng. |
| Mô tả: | Use case này mô tả các tác nhân đăng nhập vào ứng dụng. |
| Tác nhân: | Người quản lý, nhân viên, bác sĩ |
| Điều kiện trước: | Phải chọn trước người dùng vào hệ thống. |
| Điều kiện sau: | Nếu đăng nhập thành công thì tác nhân bắt đầu sử dụng ứng. |
| Luồng sự kiện chính (Basic flows) | 1. Hệ thống hiển thị giao diện đăng nhập chứa mã đăng nhập và mật khẩu. 2. Tác nhân nhập tên đăng nhập và mật khẩu của mình. 3. Tác nhân chọn nút đăng nhập. 4. Hệ thống kiểm tra thông tin đăng nhập. 5. Đăng nhập thành công và hiển thị giao diện quản lý hệ thống. |
| Luồng sự kiện phụ (Alternative Flows): | 4.1 Nếu nhập sai mật khẩu hoặc mã đăng nhập, hệ thống thông báo nhập sai mã đăng nhập hoặc mật khẩu và yêu cầu nhân viên nhập lại mã đăng nhập hoặc mật khẩu. |
| Luồng sự kiện ngoại lệ (Exception Flows): | Không có. |

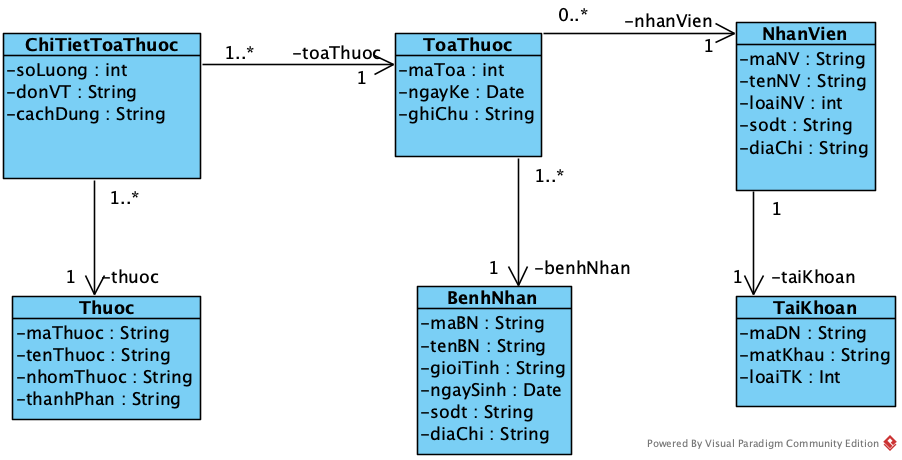
*Bảng 3-14: đặc tả đăng nhập*

Activity diagram.

****

*Hình 3-13: activity đăng nhập*

## Class diagram



*Hình 3-14: mô hình class diagram*

## Deployment diagram



*Hình 3-15: mô hình triển khai*

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

* Phần cứng:

|  |  |  |  |
| --- | --- | --- | --- |
| CPU | RAM | HDD | Architecture |
| Intel Core i7, 2.3 GHz | 8 GB | 1 TB | 64 bit |

*Bảng 4-1: cấu hình phần cứng*

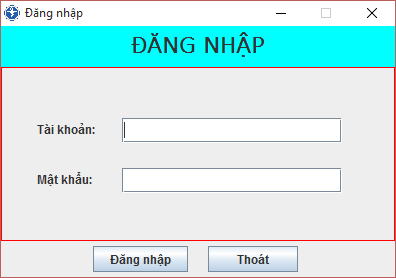
* Phần mềm:

|  |  |  |
| --- | --- | --- |
| Tên phần mềm | Phiên bản | Loại |
| Eclipse IDE for Java EE Developers | 4.9.0 | IDE cho Java |
| MongDB | 4.0 | Hệ quản trị cơ sở dữ liệu No SQL |
| Microsoft Windows 10 | 10 | Hệ điều hành |
| JDK | 1.8.0 |  |

*Bảng 4-1: cấu hình phần mềm*

## Giao diện của hệ thống

### Đăng nhập



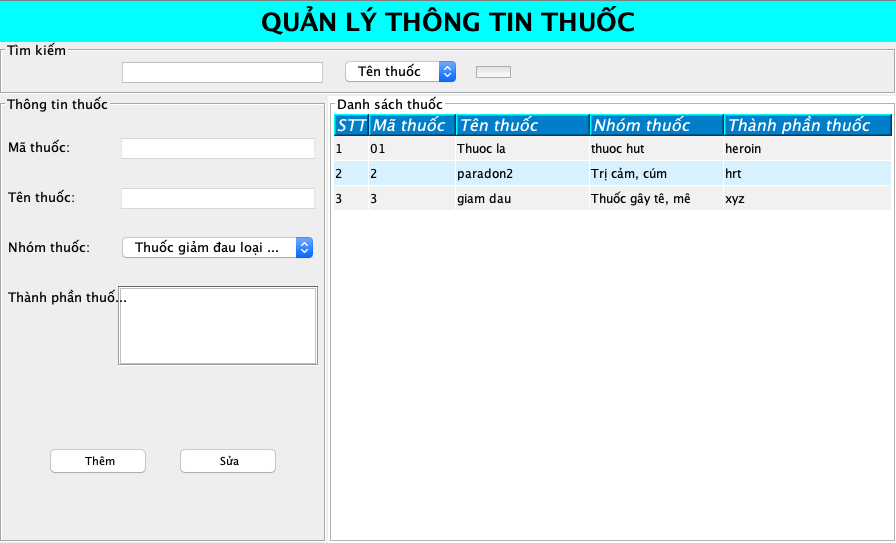
*Hình 4-1: giao diện đăng nhập*

### Giao diện chính



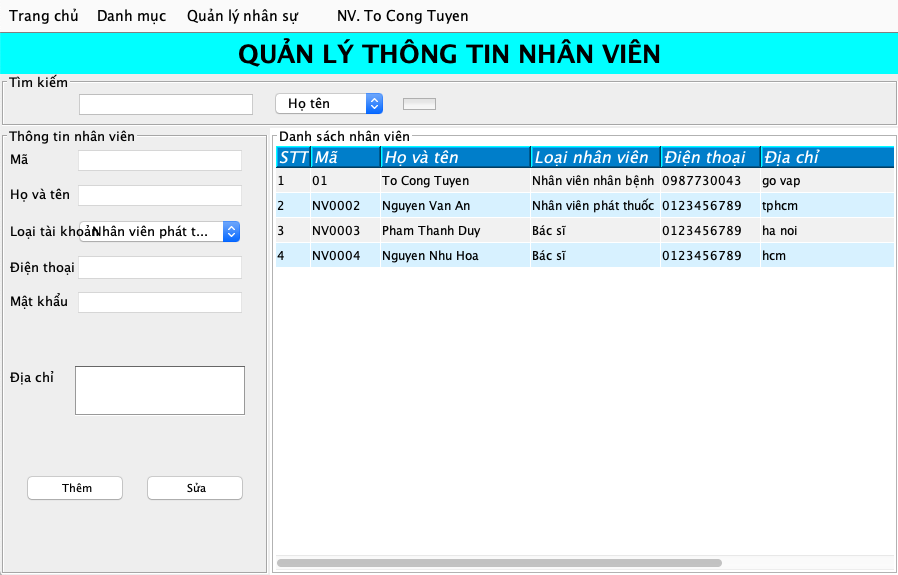
*Hình 4-2: giao diện màn chính*

### Giao diện quản lý thuốc



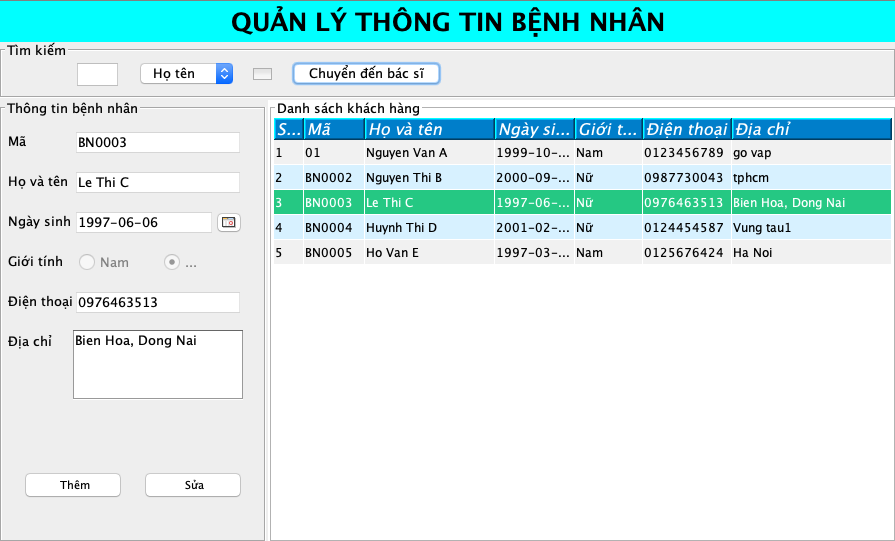
*Hình 4-3: giao diện quản lý thuốc*

### Giao diện quản lý nhân viên



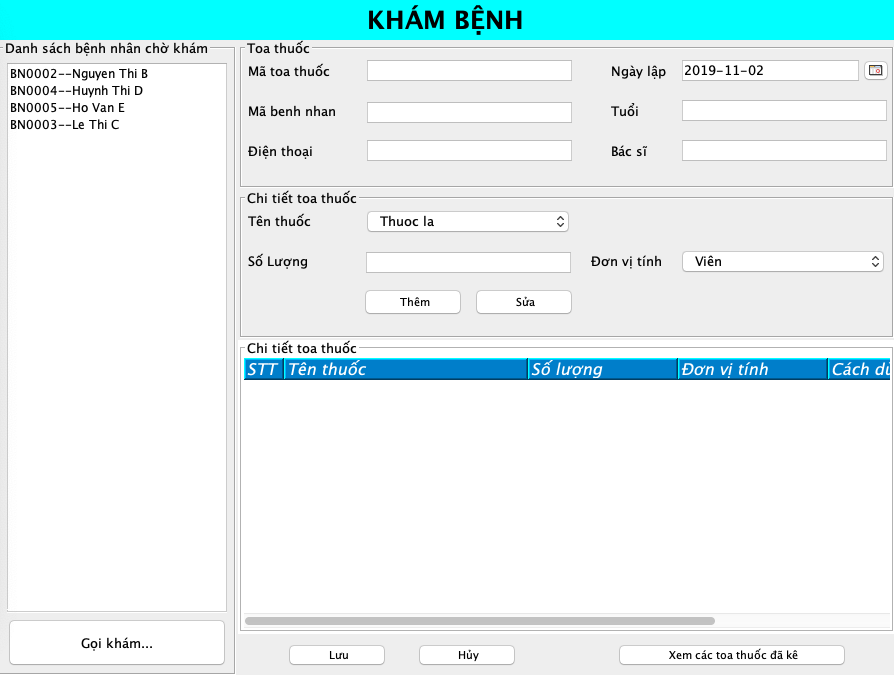
*Hình 4-4: giao diện quản lý nhân viên*

### Giao diện quản lý bệnh nhân



*Hình 4-5: giao diện quản lý bệnh nhân*

### Giao diện kê toa thuốc



*Hình 4-6: giao diện kê toa thuốc*

# : KẾT LUẬN

## Kết quả đạt được

* Đã triển khai được ứng dụng với RMI (Remote Method Invocation) với JPA (Java Persistence API) dùng để tạo bảng trong MongoDB.
* Sử dụng JavaEE kết hợp với MongoDB cơ sở dữ liệu No SQL .
* Ứng dụng đã xây dựng được với các chức năng cơ bản của ứng dụng quản lý phòng khám của bệnh viện: quản lý thuốc, quản lý nhân viên, nhập thông tin bệnh nhân, sử thông tin bênh nhân, tìm thông tin bệnh nhân, kê toa thuốc.

## Hạn chế của đồ án

* Dữ liệu lưu trữ của đề tài trong Mongo còn hạn chế.
* Chưa thực hiện chức năng kê toa.

## Hướng phát triển

* Tạo nhiều dữ liệu vào MongoDB.
* Thực hiện hoàn thiệt chức năng kê toa thuốc.

# TÀI LIỆU THAM KHẢO

Các tài liệu từ Internet

1. <https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N>

2. <https://www.mongodb.com/>

3. <https://o7planning.org/vi/10265/huong-dan-cai-dat-va-cau-hinh-database-mongodb>

4. <https://viblo.asia/p/java-rmi-va-ung-dung-phan-tan-don-gian-OEqGj50KM9bL>