**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN**



Ứng dụng bệnh viện cho mọi người

Nhóm: Funny guys

Phạm Công Anh – Lê Quang Linh – Hoàng Văn Phú

Nguyễn Minh Quang – Nguyễn Quang Huy

Giáo viên: Vũ Diệu Hương

HÀ NỘI - 2018

MỤC LỤC

[1. Mô tả chung: 2](#_Toc509230983)

[a) Mục tiêu: 2](#_Toc509230984)

[b) Lợi ích: 2](#_Toc509230985)

[c) Phạm vi: 2](#_Toc509230986)

[d) Danh sách thuật ngữ và các từ viết tắt: 2](#_Toc509230987)

[e) Công nghệ sử dụng: 2](#_Toc509230988)

[f) Tài liệu tham khảo: 4](#_Toc509230989)

[2. Mô tả hệ thống hiện tại: 4](#_Toc509230990)

[3. Yêu cầu hệ thống tương lai: 4](#_Toc509230991)

[a) Tổng quan: 4](#_Toc509230992)

[b) Chức năng: 5](#_Toc509230993)

[ Diễn đàn: 5](#_Toc509230994)

[ Thông tin: 5](#_Toc509230995)

[ Đặt lịch và đánh giá: 5](#_Toc509230996)

[ Hồ sơ bệnh án: 5](#_Toc509230997)

[ Cập nhật và chỉnh sửa: 5](#_Toc509230998)

[c) Phi chức năng: 5](#_Toc509230999)

[4. Mô tả hệ thống: 6](#_Toc509231000)

[5. Mô hình trực quan: 7](#_Toc509231001)

[a) Mô hình quan hệ: 7](#_Toc509231002)

[b) Mô hình use-case: 8](#_Toc509231003)

[6. Đặc tả chi tiết: 9](#_Toc509231004)

# **Mô tả chung:**

## Mục tiêu:

Mục tiêu của tài liệu này là cung cấp mô tả chi tiết về các yêu cầu cho phần mềm “bệnh viện cho mọi người”. Nó sẽ minh họa mục đích cụ thể và thông tin chi tiết cho việc phát triển hệ thống.

## Lợi ích:

Tài liệu này sẽ trở thành tài liệu tham khảo cho các giai đoạn thiết kế, lập trình kiểm thử trong quy trình sản xuất phần mềm này.

## Phạm vi:

Phần mềm sẽ giúp người dùng có thể thu thập các thông tin về y tế, đặt lịch khám và đánh giá về chất lượng dịch vụ y tế đồng thời cung cấp cho bệnh viện công cụng quản lý các hồ sơ của người dùng.

## Danh sách thuật ngữ và các từ viết tắt:

|  |  |  |
| --- | --- | --- |
| Thuật ngữ | Định nghĩa | Giải thích |
| MVC | model-view-control | Một loại mô hình thiết kế được sử dụng trong kỹ thuật phần mềm |
| MVVM | model-view-view model | Một loại mô hình thiết kế được sử dụng trong kỹ thuật phần mềm |

## Công nghệ sử dụng:

* Cơ sở dữ liệu: MongoDB.



+ MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở thuộc học NoSQL. Nó được thiết kế theo kiểu hướng đối tượng, các bảng trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ trên bảng không cần tuân theo một cấu trúc nhất định nào cả (điều này rất thích hợp để làm big data).

+ MongoDB lưu trữ dữ liệu theo hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh. Một số điểm mạnh của MongoDB:

* Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau.
* Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng.
* Sử dụng bộ nhớ nội tại, nên truy vấn sẽ rất nhanh.
* MongoDB rất dễ mở rộng.
* Không có các join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên mongoDB.
* MongoDB phù hợp cho các ứng dụng realtime.
* Back-end: Node.js



Node.js là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thuật điều khiển theo sự kiện, nhập/xuất không đồng bộ để tối tiểu tổng chi phí và tối đại khả năng mở rộng. Node.js bao gồm có V8 JavaScript engine của Google, libUV, và vài thư viện khác.

Node.js định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

Khác với hầu hết các chương trình Javascript, Nodejs không chạy trên một trình duyệt mà chạy trên Server. Node.js sử dụng nhiều chi tiết kỹ thuật của CommonJS. Nó cung cấp một môi trường REPL cho kiểm thử tương tác.

* Front-end: Angular.js



AngularJS là một web framework JavaScript được phát triển và tài trợ bởi Google và cộng đồng để giải quyết các vấn đề gặp phải trong việc phát triển ứng dụng đơn trang. Nó được phát triển từ AngularJS do nhu cầu phát triển về tốc độ, đơn giản và sự linh hoạt. AngularJs có các thành phần bổ sung cho Cordova, framework sử thường dùng để viết các ứng dụng di động. Nó nhằm mục đích để đơn giản hóa cả phát triển và thử nghiệm của các ứng dụng bằng cách cung cấp một framework với kiến trúc cho client side đó là MVC (model-view-control) và MVVM (model-view-view model).

## Tài liệu tham khảo:

[1] Object-Oriented Software Development with UML (Zhiming Liu).

[2] SOFTWARE ENGINEERING (Ninth Edition).

# **Mô tả hệ thống hiện tại:**

+ Nhiều bệnh viện và phòng khám chỉ quản lý hồ sơ bệnh nhân trên sổ sách làm việc kiểm soát bệnh án của bệnh nhân trở nên khó khăn.

+ Việc thông tin cho bệnh nhân về dịch bệnh, các nghiên cứu mới ra hay cá mẹo vặt giúp ích cho sức khỏe không được bệnh viện chú tâm đến.

+ Việc khám chữa bệnh phải đến bệnh viện mới có thể đặt lịch, không thể hẹn lịch để bác sĩ tới nhà.

+ Bệnh nhân không có cái nhìn tổng quan về cách làm việc của bác sĩ vì không có đánh giá của các bệnh nhân đã khám chữa bệnh trước đó.

# **Yêu cầu hệ thống tương lai:**

## Tổng quan:

Phần mềm sẽ khắc phục các yếu điểm của hệ thống hiện tại và bổ sung các tính năng như diễn đàn, thông tin, đặt lịch – đánh giá và hồ sơ bệnh án giúp cho việc quản lý và tương tác với người dùng tốt hơn.

## Chức năng:

### Trao đổi kiến thức trên diễn đàn diễn đàn:

+) Tin tức:

1. Bản tin y học: Các bài báo về nghiên cứu khoa học và thành tựu y học đáng chú ý trong nước và ngoài nước.
2. Mẹo nhỏ và cảnh báo: Cập nhật các tin tức mới nhất về dịch bệnh và các căn bệnh nguy hiểm cùng các mẹo nhỏ cũng như cách phòng ngừa.

+) Giải đáp thắc mắc: Người dùng sẽ viết lên những thắc mắc hay câu hỏi để trao đổi với cộng đồng hoặc bác sĩ để lấy lời khuyên. Đây cũng là nơi chia sẻ các kinh nghiệm từ cộng đồng giúp bảo về sức khỏe cho mọi người.

### Cung cấp thông tin:

+) Cung cấp các thông tin về các khoa, phòng (Tên, chức năng, …).

+) Thông tin các nhân viên, bác sĩ (tên, chức vị, kinh nghiệm, ...).

+) Thông tin của người dùng.

+) Tìm kiếm thông tin.

+) Vị trí map và giới thiệu về bệnh viện.

### Đặt lịch và đánh giá:

+) Cho phép khách hàng đặt lịch hẹn gặp mặt với bác sĩ theo form (Nếu bác sĩ có thể đi được thì yêu cầu sẽ được chấp nhận).

+) Sau khi khám khách hàng sẽ đánh giá về thái độ, nghiệp vụ của nhân viên ý tế qua một form.

### Thực hiện các nghiệp vụ liên quan hồ sơ bệnh án:

+) Lưu trữ thông tin cơ bản của bệnh nhân (Thời gian nhập bênh, lịch sử bệnh án, vị trí giường bệnh, phòng đang nằm và các nhân viên phụ trách, ….).

+) Xuất hồ sơ bệnh án (theo số CMND và số điện thoại đăng ký để xác minh).

### Cập nhật và chỉnh sửa:

Cập nhật các thông tin và chỉnh sửa (bên phía admin).

## c) Phi chức năng:

(1) Hiệu năng hoạt động: Hệ thống hỗ trợ tới 200 người dùng truy xuất cơ sở dữ liệu trung tâm đồng thời bất kỳ lúc nào, độ trễ của các phép truy vấn tới cơ sở dữ liệu không quá 10s.

(2) Tương thích, Đa dụng: Hệ thống phải chạy tốt trên các browser cơ bản như Google Chrome/ Firefox/ Opera/

(3) Tính khả dụng:

+ Giao diện phải thân thiện với người dùng, các chức năng phải hiện ngay trên trang nhất nơi dễ nhìn thấy nhất và phải ghi rõ tên để người dùng dễ dàng sử dụng ngay cả trong lần đầu người dùng sử dụng phần mềm.

+ Phần mềm phải được bảo trì hằng ngày, tránh xảy ra các lỗi nghiêm trọng trong quá trình vận hành như mất mát dữ liệu hoặc tắc nghẽn đường truyền do server quá tải.

(4) Tính tin cậy: Hệ thống phải được sao lưu lại định kỳ 1 lần/ngày để hỗ trợ cho việc phục hồi dữ liệu khi xảy ra mất mát, lỗi hay sai sót trong quản lý hoặc quá trình vận hành.

(5) An toàn thông tin: Hệ thống phải bảo mật, phân quyền chặt chẽ cho các cấp người dùng. Đồng thời chỉ có nhân viên quản lý mới có quyền thay đổi và cập nhập thông tin khoa phòng, thông tin nhân viên.

(6) Tính ổn định: Hệ thống đảm bảo chạy ổn định liên tục 24/24 thời giản nghỉ không quá 10%.

(7) Sự hỗ trợ: Hệ thống phải dễ dàng cập nhật và nâng cấp khi các công nghệ sử dụng nâng cấp phiên bản mới.

(8) Các ràng buộc thiết kế: Hệ thống hỗ trợ ngôn ngữ Việt Nam.

# **Mô tả hệ thống:**

*a. Các thực thể:*

+ Mỗi bài đăng gồm: id, hình ảnh minh họa, nội dung, số lượt thích, không thích, tiêu đề, tag.

+ Mỗi bình luận gồm: id, nội dung, hình ảnh minh họa.

+ Khoa gồm: tên khoa, mô tả chức năng, biểu tượng khoa, thời gian thành lập và địa chỉ.

+ Tài khoản: tên đăng nhập, mật khẩu.

+ Người dùng: id, họ và tên, số CMND, ngày tháng năm sinh, giới tính, số điện thoại, địa chỉ, ảnh chân dung.

+ Nhân viên gồm: (chức vụ) học vấn, kinh nghiệm (thời gian đã làm việc), điểm đánh giá

+ Bệnh nhân: yêu cầu khám bệnh.

+ Hồ sơ bệnh án: Thời gian nhập viện, thời gian xuất viện, số giường bệnh và phòng đang nằm, nội dung tình trạng, cách chữa.

+ Loại bệnh: id, tên.

+ Loại thuốc: id, tên, công dụng.

+ Lịch hẹn: id, thời gian hẹn, trạng thái chấp nhận, mô tả.

+ Đánh giá: id, điểm (số sao), nội dung đánh giá, id bác sĩ, id bệnh nhân.

*b. Quan hệ:*

+ Mỗi bài đăng có nhiều bình luận, mỗi bình luận chỉ thuộc một bài đăng.

+ Mỗi bài đăng thuộc một tài khoản, mỗi tài khoản có nhiều bài đăng.

+ Mỗi bình luận chỉ có thể thuộc tối đa một tài khoản, mỗi tài khoản có nhiều bình luận.

+ Mỗi khoa gồm nhiều nhân viên, mỗi khoa có một trưởng khoa là nhân viên giữ chức vụ bác sĩ, mỗi nhân viên chỉ thuộc một khoa.

+ Bác sĩ, bệnh nhân và tài khoản có quan hệ ràng buộc chồng chéo với người dùng.

+ Mỗi bệnh nhân gồm nhiều hồ sơ bệnh án, mỗi hồ sơ bệnh án thuộc một bệnh nhân.

+ Mỗi hồ sơ bệnh án có nhiều loại bệnh, mỗi loại bệnh thuộc nhiều hồ sơ bệnh án.

+ Mỗi loại bệnh gồm nhiều loại thuốc, mỗi loại thuốc thuộc nhiều loại bệnh.

+ Mỗi bác sĩ có thể nhận nhiều hồ sơ bệnh án, mỗi hồ sơ bệnh án chỉ do một bác sĩ phụ trách.

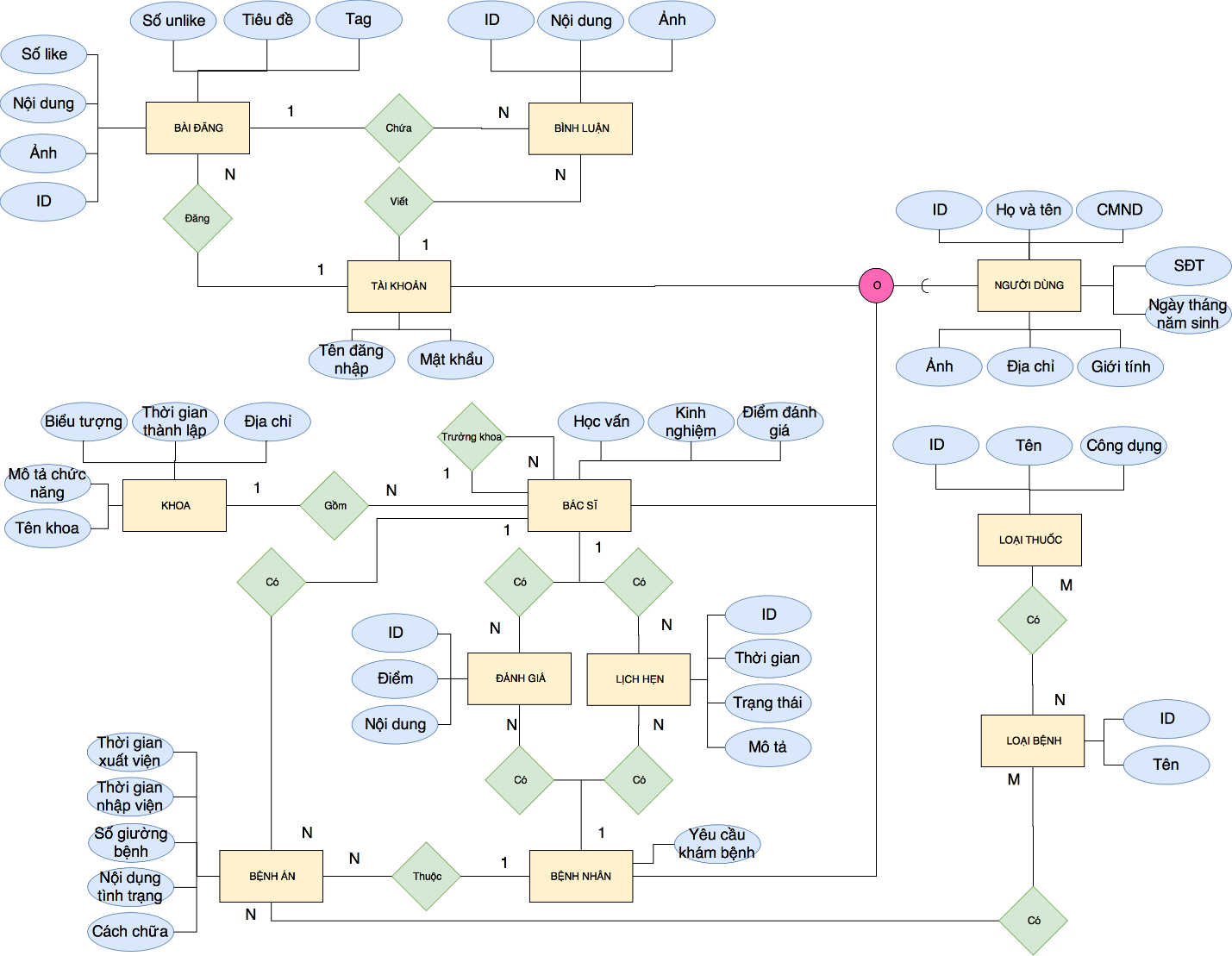
+ Mỗi lịch hẹn chỉ có một bác sĩ và một tài khoản là bệnh nhân, mỗi bác sĩ và bệnh nhân sẽ có nhiều lịch hẹn.

+ Mỗi đánh giá thuộc một tài khoản là bệnh nhân và một bác sĩ, mỗi bác sĩ và bệnh nhân có nhiều đánh giá.

+ Mỗi đối tượng sẽ có thời gian update mới nhất và trạng thái hoạt ẩn hay hiện dữ liệu (nếu ẩn thì chỉ có người có quyền truy cập cao nhất mới dược truy cập).

# **Mô hình trực quan:**

## Mô hình quan hệ:



Mô hình quan hệ của hệ thống

## Mô hình USECASE chính:



Mô hình ca sử dụng của hệ thống

# **Đặc tả chi tiết:**

* 1. Login/Logout
     1. Mô tả tóm tắt

Use case này mô tả cách một người dùng, quản lý đăng nhập vào hệ thống giao tiếp bệnh viện.

* + 1. Dòng sự kiện

+) Dòng sự kiên chính

Use case này bắt đầu khi một actor muốn đăng nhập vào hệ thống giao tiếp bệnh viện, kết thúc khi actor đăng xuất khỏi hệ thống

1 – Hệ thống yêu cầu actor nhập tên và mật khẩu.

2 – Hệ thống kiểm chứng tên và mật khẩu được nhập

3 – Thống báo thành công và cho phép actor đăng nhập vào hệ thống, hệ thống sẽ phân quyền người dùng theo từng loại nhân viên.

4- Hệ thống yêu cầu actor xác nhận đăng xuất.

+) Các dòng sự kiên khác

Nếu trong dòng sự kiện chính, actor nhập sai tên hoặc mật khẩu, hệ thống sẽ hiển thị một thông báo lỗi. Actor có thể chọn trở về đầu của dòng sự kiến chính hoặc hủy bỏ việc đăng nhập, lúc này usecase hệ thống kết thúc

* + 1. Các yêu cầu đặc biệt

Không có.

* + 1. Điều kiện tiên quyết

Không có.

* + 1. Hậu điều kiện

Nếu use case thành công, actor lúc này đã đăng nhập vào hệ thống. Nếu không trạng thái hệ thống sẽ không thay đổi.

* + 1. Điểm mở rộng

Không có.

* 1. Add the Post
     1. Mô tả tóm tắt

Use case này cho phép actor (user: nhân viên, khách hàng) đăng một tin tức, viết bình luận trong mục tin tức, đặt câu hỏi và trả lời trong mục diễn đàn y tế.

* + 1. Dòng sự kiện

+) Dòng sự kiến chính

Use case này bắt đầu khi actor muốn thêm một vài viết, bình luận, hay đặt câu hỏi, trả lời.

1 – Hệ thống hiển thị trực quan yêu cầu của actor.

2 – Sau khi actor thực hiện một thao tác chọn chức năng, một trong các luồng phụ sau được thực hiện

Write New: Viết một tin tức, bài báo trong mục tin tức

1-Hệ thống yêu cầu actor cung cấp một bài viết dưới dạng tài liệu word.

2-Hệ thống sẽ lưu trữ bài viết vào cơ sơ dữ liệu bao gồm các thông tin: tên người dùng và nội dung.

3-Hệ thống sẽ hiển thị kết quả là bài báo được đăng tại mục tin tức

Write Comment: Viết bình luận trong một bài báo, tin tức

1-Hệ thống yêu cầu actor nhập vào chuỗi kí tự bình luận cho bài viết.

2-Hệ thống sẽ lưu trữ bình luận cho chính xác bài viết bao gồm các thông tin: tên người dùng và nội dung.

3-Hệ thống hiển thị bình luận cho bài viết ở phần bình luận của bài viết.

Add the Question: Thêm câu hỏi, thắc mắc trong mục Hỏi và trả lời.

1-Hệ thống yêu cầu actor nhập vào thắc mắc

2-Hệ thống thêm câu hỏi của khách hàng vào Cơ sở dữ liệu, bao gồm tên người dùng, nội dung và tag và hiển thị câu hỏi trên mục hỏi đáp.

Add the Answer: Giải đáp thắc mắc cho câu hỏi của actor.

1-Hệ thống yêu cầu actor nhập vào chuỗi kí tự bình luận cho bài viết.

2-Hệ thống sẽ lưu trữ câu trả lời cho chính xác câu hỏi bao gồm các thông tin: tên người dùng và nội dung.

3-Hệ thống hiển thị câu trả lời cho câu hỏi ở phần giải đáp của câu hỏi.

* + 1. Các yêu cầu đặc biệt

Không có

* + 1. Điều kiện tiên quyết

Actor cần phải đăng nhập vào hệ thống trước khi use case bắt đầu.

* + 1. Hậu điều kiện

Nếu use case thực hiện thành công, hệ thống sẽ cập nhập về bài báo, tin tưc, bình luận, câu hỏi, đáp vấn cho cơ sở dữ liệu. Ngược lại, trạng thái của hệ thống không thay đổi.

* + 1. Điểm mở rộng

Không có

* 1. Edit the Post

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.4 Vote the Post

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.5 Delete the post

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.6 Search the post

6.3.1 Tóm tắ

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.6 Resgister

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.7 Manage Account

6.3.1 Tóm tắt

Use case này cho phép người quản lý, xem, xóa tài khoản của người dùng.

6.3.2 Các dòng sự kiện

+) Dòng sự kiện chính:

Use case này bắt đầu khi actor muốn xem hay xóa tài khoản người dùng.

1 – Hệ thống cho phép actor chọn chức năng muốn thực hiện là xem, xóa tài khoản người dùng.

2 – Có các luồng sự kiện sau sẽ được thực hiện

Nếu actor (Người quản lý) chọn xem, luồng phụ ‘Xem tài khoản’ được thực hiện

Nếu actor (người quản lý) chọn xóa, luồng phụ ‘Xóa tài khoản’ được thực hiện.

+) Xem tài khoản:

Hệ thống cho phép actor xem hoạt động, hành vi của tài khoản này.

+) Xóa tài khoản:

Hệ thống cho phép actor kiểm tra lần cuối về thông tin của tài khoản và xóa tài khoản khỏi hệ thống.

6.3.3 Các yêu cầu đặc biệt

Các tài khoản trong thời gian quá 6 thánh không hoạt động cần được xóa để giảm bớt dữ liệu cho hệ thống.

6.4.4 Điều kiện tiên quyết

Người quản lý phải cần phải đăng nhập với quyền quản trị hệ thống

6.2.5 Hậu điều kiện

Hệ thống sẽ thay đổi thi actor thực hiện thành công use case với luồng xóa tìa khoản. Nếu không, hệ thống không đổi

6.2.6 Điểm mở rộng

6.8 Search Departments

6.3.1 Tóm tắt

Use case này cho phép người dùng tra cứu thông tin khoa, phòng của bệnh viện.

6.3.2 Các dòng sự kiện

+) Dòng sự kiện chính:

Use case này bắt đầu khi một người dùng tra cứu thông tin khoa phòng của bệnh viện

1 – Hệ thống truy xuất và hiển thị thông tin phòng cho người dùng

2 – Khi người dùng xem xong thông tin khoa, phòng thì use case kết thúc

+) Dòng sự kiện khác:

- Không có thông tin về khoa, phòng.

Nếu trong dòng sự kiện chính hệ thống không thể tìm thấy thông tin khoa phòng trong danh sách của bệnh viện, một thông báo sẽ được hiển hị. Use case kết thúc khi người dùng xem xong thông báo này.

6.3.3 Các yêu cầu đặc biệt

Không có

6.4.4 Điều kiện tiên quyết

Người dùng đã đăng nhập hệ thống

6.2.5 Hậu điều kiện

Trạng thái của hệ thống không thay đổi sau khi thực hiện use case này.

6.2.6 Điểm mở rộng

Không có.

6.10 Display Staff’s Information

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.11 Display Location

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.12 Search Staff’s

6.3.1 Tóm tắt

Use case này cho phép người dùng tra cứu thông tin nhân viên của bệnh viện.

6.3.2 Các dòng sự kiện

+) Dòng sự kiện chính:

Use case này bắt đầu khi một người dùng tra cứu thông tin nhân viên của bệnh viện

1 – Hệ thống truy xuất và hiển thị thông tin nhân viên cho người dùng

2 – Khi người dùng xem xong thông tin nhân viên thì use case kết thúc

+) Dòng sự kiện khác:

- Không có thông tin về nhân viên.

Nếu trong dòng sự kiện chính hệ thống không thể tìm thấy thông tin nhân viên trong danh sách của bệnh viện, một thông báo sẽ được hiển hị. Use case kết thúc khi người dùng xem xong thông báo này.

6.3.3 Các yêu cầu đặc biệt

Không có

6.4.4 Điều kiện tiên quyết

Người dùng đã đăng nhập hệ thống

6.2.5 Hậu điều kiện

Trạng thái của hệ thống không thay đổi sau khi thực hiện use case này.

6.2.6 Điểm mở rộng

Không có.

6.13 Edit Location

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.14 Edit Department

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.15 Edit Staff’s Information

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.17 Evaluate Staff

6.3.1 Tóm tắt

Use case này cho phép khách hàng đánh giá về thái độ, nghiệp vụ của nhân viên ý tế trong bệnh viện

6.3.2 Các dòng sự kiện

Use case bắt đầu khi người dùng xem thông tin của nhân viên y tế

+) Dòng sự kiện chính:

1 – Hệ thống đưa ra một biểu mẫu đánh giá cho khách hàng.

2 – Hệ thống ghi nhận các đánh giá của khách hàng vào cơ sở dữ liệu

6.3.3 Các yêu cầu đặc biệt

Khách hàng có thể đánh giá duy nhất về nhân viên, việc đánh giá thêm sẽ là chỉnh sửa sự đánh giá về nhân viên

6.4.4 Điều kiện tiên quyết

Actor phải là khách hàng và có sự đăng nhập trước khi sử dụng use case này.

6.2.5 Hậu điều kiện

Hệ thống sẽ ghi nhận sự đánh giá của khách hàng khi use case kết thúc.

6.2.6 Điểm mở rộng

Không có.

6.18 Start Appointment

6.3.1 Tóm tắt

Use case cho phép khách hàng đăng kí yêu cầu lịch hẹn riêng về sưc khỏe với một bác sĩ, và bác sĩ có thể trả lời yêu cầu hẹn riêng.

6.3.2 Các dòng sự kiện

Use case bắt đầu khi khách hàng muốn yêu cầu lịch hẹn về sức khỏe với nhân viên y tế.

1 – Khách hàng nhập tìm chính xác về nhân viên y tế trong hệ thống qua biểu mẫu hệ thống cung cấp.

2 – Khách hàng nhập thông tin cơ bản của mình vào form xác nhận.

6.3.3 Các yêu cầu đặc biệt.

3 – Khách hàng nhập vào thời gian dự kiến cho cuộc hẹn.

4 – Hệ thống lưu lại thông tin đăng kí cuộc hẹn của khách hàng với trạng thái đang chờ

5 – Hệ thông gửi thông tin về lịch hẹn đến tài khoản của nhân viên y tế.

6 – Nhân viên y tế quyết định phản hồi về cuộc hẹn y tế.

7 – Hệ thống thay đổi trạng thái của cuộc hẹn: Sẵn sàng hoặc bị hủy bỏ. Trả kết quả về cho khách hàng.

6.4.4 Điều kiện tiên quyết

Khách hàng phải đăng nhập vào hệ thống để sư dụng use case.

6.2.5 Hậu điều kiện

Nếu use case không đủ 3 bước đầu, hệ thống sẽ vẫn giữ nguyên và use case kết thúc. Nếu use case thành công, thông tin về cuộc hẹn y tế sẽ được ghi nhận.

6.2.6 Điểm mở rộng

Không có.

6.19 Search Medical Record

6.3.1 Tóm tắt

Use case này cho phép nhân viên bệnh viện tìm kiếm và xem thông tin hồ sơ bệnh án của bệnh nhân.

6.3.2 Các dòng sự kiện

+) Dòng sự kiện chính:

Use case bắt đầu khi actor muốn tìm kiếm và xem thông tin hồ sơ bệnh án.

1 – Hệ thống truy xuất và hiển thị thông tin về các bản hồ sơ bệnh án

2 – Khi actor thoát khỏi chức năng này thì use case kết thúc

+) Các dòng sự kiện khác:

- Không có thông tin về bản hồ sơ.

Nếu trong dòng sự kiến chính hệ thống không tìm thấy hồ sơ bệnh án của bệnh nhân trong hệ thống bệnh viện. Một thông báo sẽ được hiển thị, use case kết thúc khi actor xem xong thông báo này.

6.3.3 Các yêu cầu đặc biệt

Không có

6.4.4 Điều kiện tiên quyết

Nhân viên y tế cần đăng nhập đúng phân quyền vào hệ thống để sử dụng use case này.

6.2.5 Hậu điều kiện

Trạng thái của hệ thống không thay đổi sau khi use case này được thực hiện

6.2.6 Điểm mở rộng

Không có

6.20 Delete Medical Record

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.21 Update Medical Record

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng

6.22 Create Medical Record

6.3.1 Tóm tắt

6.3.2 Các dòng sự kiện

6.3.3 Các yêu cầu đặc biệt

6.4.4 Điều kiện tiên quyết

6.2.5 Hậu điều kiện

6.2.6 Điểm mở rộng