**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:

### 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.

### 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.

### 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 phải kiểm soát ít nhất 100 yêu cầu trong 1s.

(2) Tương thích: Hệ thống phải chạy tốt trong môi trường webbrowser với HTML và CSS;

(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, toàn vẹn thông tin của người dùng bằng việc đăng nhập với mật khẩu do người dùng tự đặt ra.

(6) Các yêu cầu phi chức năng khác: Hệ thống phải dễ dàng cập nhật và nâng cấp khi các công nghệ sử dụng nâng cấp phiên bản mới.

# Mô tả hệ thống:

*a. Các đối tượng:*

+ 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.

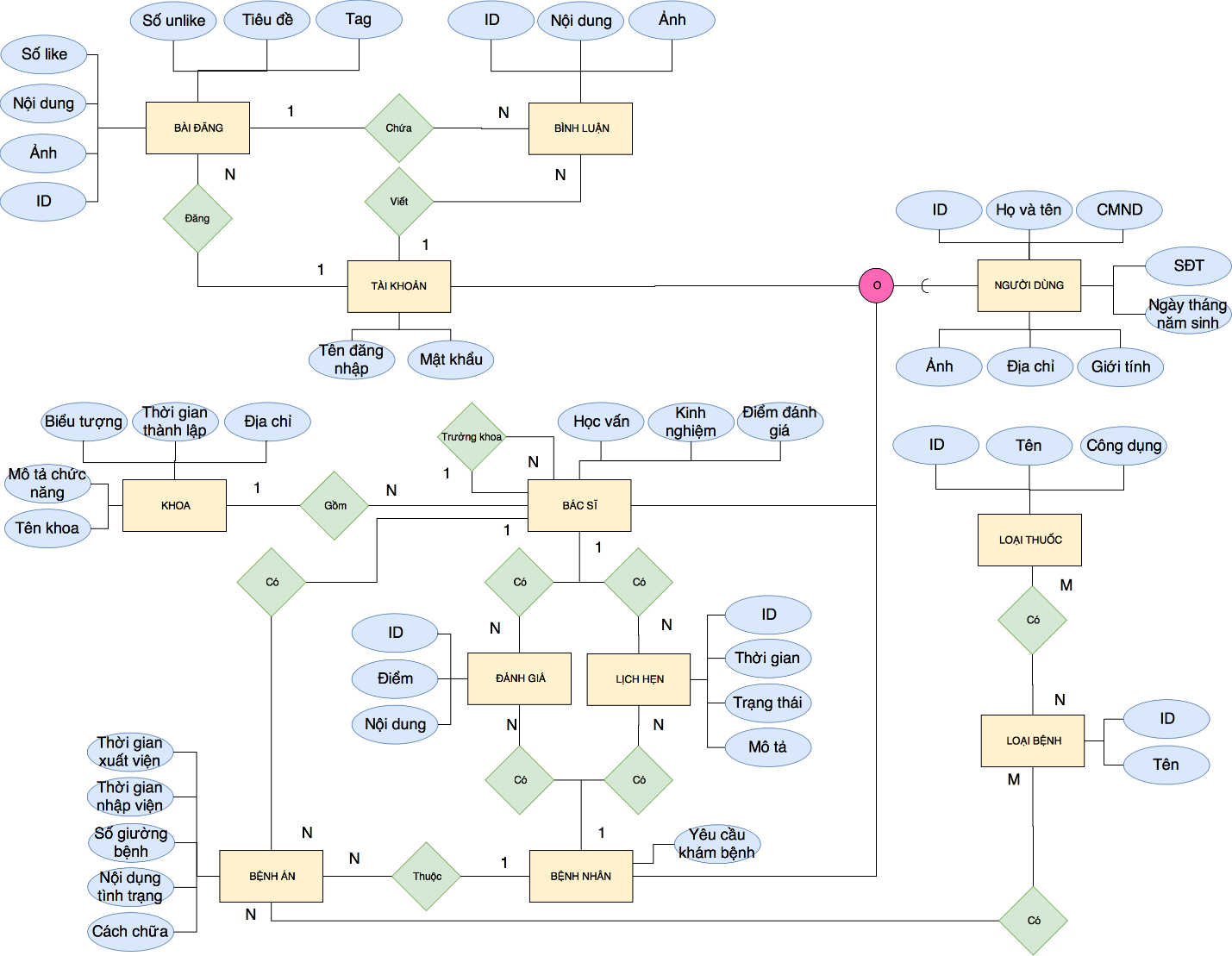
+ Mỗi lịch hẹn chỉ có một bác sĩ và một tài khoản là bệnh nhân, mỗi bác sĩ và bệnh nhân sẽ có nhiều lịch hẹn.

+ 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 use-case:



Mô hình ca sử dụng của hệ thống

# Đặc tả chi tiết: