**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**





**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI:**

**XEM VÀ TRUY XUẤT**

**ẢNH DICOM TRÊN WEB**

**GVHD :** ***PGS TS. Huỳnh Trung Hiếu***

**Sinh viên thực hiện : *Nguyễn Huỳnh Phúc***

**MSSV : *12139731***

**Lớp : *DHTH8A***

**TP. HCM, Tháng 08 năm 2017**

# **LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành đến thầy ***Huỳnh Trung Hiếu*** đã hỗ trợ em trong quá trình làm đồ án, đưa ra những nhận xét cũng như là góp ý để em có thể tiếp cận và giải quyết vấn đề một cách dễ dàng hơn, tạo điều kiện cho em được hoàn thành tốt nhất đồ án chuyên ngàng này.

Đồ án này còn nhiều phần thiếu xót, nhưng em đã cố gắng hoàn thành những mục tiêu mà đồ án cần tiếp cận. Đồ án này giúp em có cái nhìn về ảnh dicom trong việc phục vụ các chuẩn đoán y tế, xong để hướng đến một sản phẩm hoàn thiện em còn phải cải thiện thêm nhiều nhiều thành phần trong thời gian sau.

Cuối cùng, chúc thầy thật khỏe mạnh và có thêm nhiều thành tựu trong nghiên cứu khoa học.

TP. Hồ Chí Minh, ngày 18 tháng 08 năm 2017

Sinh viên thực hiện

Nguyễn Huỳnh Phúc

# **NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

TP. Hồ Chí Minh, ngày....tháng....năm 2017

Chữ kí của GVHD.

# **NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 1**

TP. Hồ Chí Minh, ngày....tháng....năm 2017

Chữ kí của GVPB.

# **NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 2**

TP. Hồ Chí Minh, ngày....tháng....năm 2017

Chữ kí của GVPB.

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc489050780)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 3](#_Toc489050781)

[NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 1 4](#_Toc489050782)

[NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 2 5](#_Toc489050783)

[MỤC LỤC 6](#_Toc489050784)

[DANH MỤC HÌNH 9](#_Toc489050785)

[PHẦN MỞ ĐẦU 10](#_Toc489050786)

[1. Lý do chọn đề tài 10](#_Toc489050787)

[2. Mục đích, nhiệm vụ của đề tài 10](#_Toc489050788)

[3. Ứng dụng của ảnh DICOM trên trình duyệt web 10](#_Toc489050789)

[4. Phạm vi của đề tài 11](#_Toc489050790)

[PHẦN NỘI DUNG 11](#_Toc489050791)

[PHẦN 1 – KHÁI QUÁT VỀ ẢNH DICOM 11](#_Toc489050792)

[1.1. Khái niệm cơ bản về định dạng ảnh DICOM 11](#_Toc489050793)

[1.1.1. Định dạng ảnh DICOM là gì ? 11](#_Toc489050794)

[1.1.2. Định dạng dữ liệu DICOM 12](#_Toc489050795)

[PHẦN 2 – GIỚI THIỆU CƠ BẢN VỀ ORTHANC, NODE.JS, EXPRESS.JS, ANGULARJS VÀ CORNERSTONE 13](#_Toc489050796)

[2.1. Orthanc - DICOM server 13](#_Toc489050798)

[2.1.1. Orthanc là gì ? 13](#_Toc489050799)

[2.1.2. Giao thức mạng DICOM 14](#_Toc489050800)

[2.1.3. Tham số của một máy chủ DICOM 15](#_Toc489050801)

[2.1.4. Cấu hình Orthanc 16](#_Toc489050802)

[2.1.5. Sử dụng giao thức HTTP thay vì giao thức DICOM 17](#_Toc489050803)

[2.2. Giới thiệu Node.js 18](#_Toc489050804)

[2.2.1. Node.js là gì ? 18](#_Toc489050805)

[2.2.2. Đặc điểm của Node.js 18](#_Toc489050806)

[2.2.3. Express framework 19](#_Toc489050807)

[2.3. AngularJS 20](#_Toc489050808)

[2.3.1. AngularJS là gì ? 20](#_Toc489050809)

[2.3.2. Các tính năng của AngularJS 21](#_Toc489050810)

[2.3.3. Đặc điểm của AngularJS 21](#_Toc489050811)

[2.4. Cornerstone 22](#_Toc489050812)

[2.4.1. Giới thiệu thư viện Cornerstone 22](#_Toc489050813)

[2.4.2. Các tính năng chính 22](#_Toc489050814)

[PHẦN 3 – MÔ HÌNH CLIENT–SERVER 23](#_Toc489050815)

[3.1. Khái niệm về REST 23](#_Toc489050816)

[3.2. Mô hình client-server của đề tài 24](#_Toc489050817)

[3.2.1. Mô hình phía server 24](#_Toc489050818)

[3.3.1. Mô hình phía client 26](#_Toc489050819)

[3.3.2. Mục đích của mô hình 28](#_Toc489050820)

[3.3.3. Giao diện chương trình 29](#_Toc489050821)

[PHẦN KẾT LUẬN 30](#_Toc489050822)

[1. Tóm tắt kết quả 30](#_Toc489050823)

[2. Đề xuất, kiến nghị 30](#_Toc489050824)

[DANH MỤC TÀI LIỆU THAM KHẢO 31](#_Toc489050825)

# **DANH MỤC HÌNH**

*Hình 1. Sử dụng trình duyệt web trên các thiết bị khác nhau*

*Hình 2. Cấu trúc một tập tin DICOM cơ bản*

*Hình 3. Kiến trúc lưu trữ và truyền dẫn tập tin DICOM của Oracle*

*Hình 4. Truy vấn thông qua giao thức DICOM*

*Hình 5. Các lớp của một mô hình truyền dẫn DICOM*

*Hình 6. Cơ chế Event-Loop của Node.js*

*Hình 7. Các thành phần cơ bản của AngularJS 2*

*Hình 8. Mô hình client-server của đề tài*

*Hình 9. Lược đồ quan hệ cơ sở dữ liệu của ảnh DICOM trong Orthanc*

*Hình 10. Các thành phần chính của thư viện Cornerstone được sử dụng trong đề tài*

*Hình 11. Giao diện hiển thị chi tiết thông tin về các Study của các bệnh nhân*

*Hình 12. Giao diện hiển thị ảnh DICOM*

# **PHẦN MỞ ĐẦU**

1. ***Lý do chọn đề tài***

Hiện nay sử dụng ảnh chụp từ các máy chụp cắt lớp vi tính CT (Computerized Tomography) hoặc chụp cộng hưởng từ MRI (Magnetic Resonance Tomography) trong chuẩn đoán các bệnh lý trong y tế rất quang trọng và phổ biến rộng rãi trong y học. Do yêu cầu cao về độ chính xác và thống về các tiêu chuẩn chung trong y tế, nên định dạng ảnh DICOM (Digital Imaging and Communications in Medicine) ra đời.

Ảnh DICOM thường được sử dụng trên các hệ thống nội bộ của bệnh viện thông qua phương thức TCP/IP và các ứng dụng desktop để xem ảnh DICOM. Tuy nhiên, với sự bùng nổ của Internet việc các bác sĩ có thể ở bất kì đâu với chiếc điện thoại thông minh, máy tính bảng hoặc laptop chạy trình duyệt web (web browser) điều có thể xem được ảnh, để đưa ra các nhận định sớm nhất về tình trạng của bệnh nhân.

***2***. ***Mục đích, nhiệm vụ của đề tài***

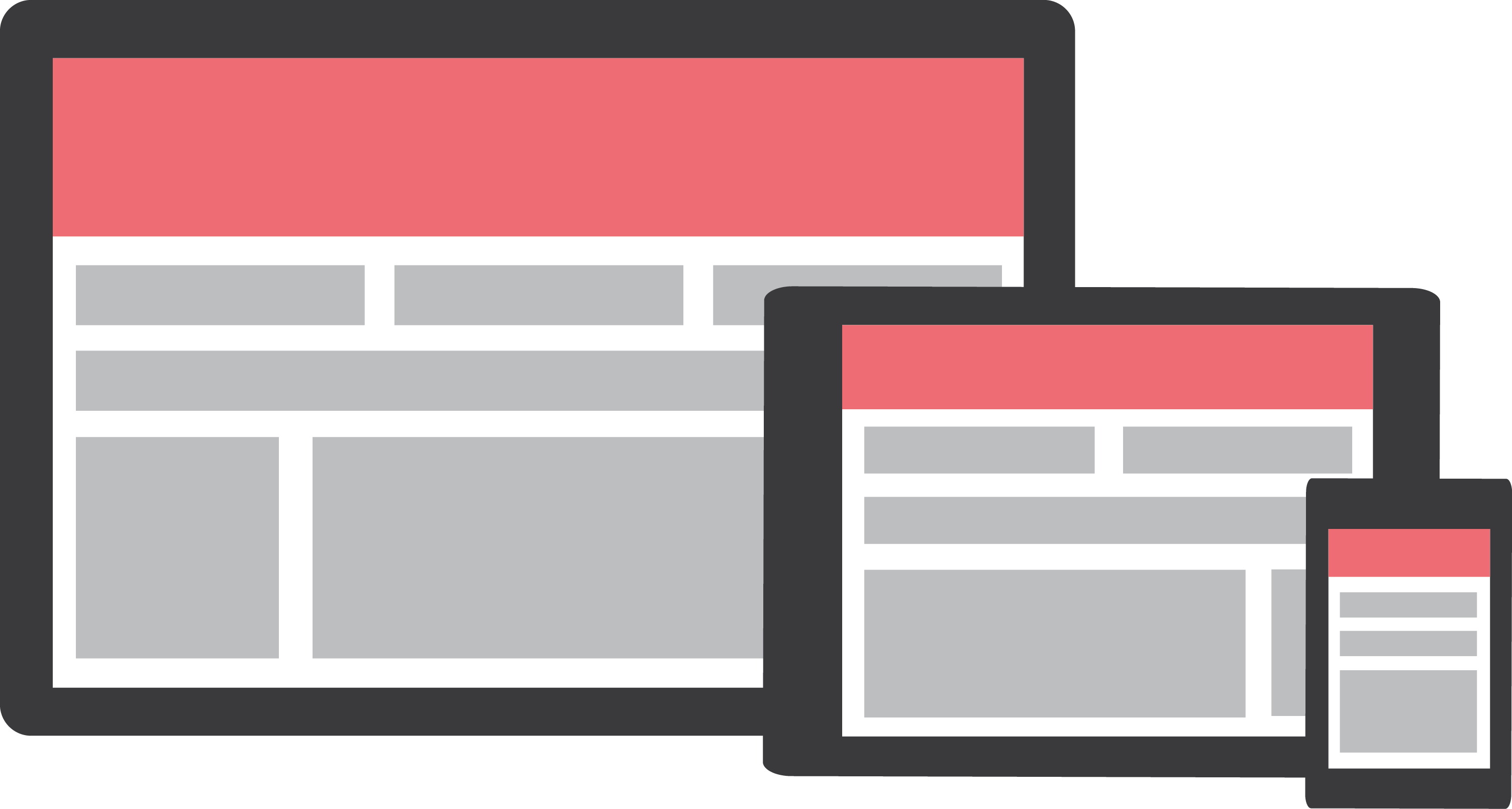
Để giúp chúng ta có cái nhìn tổng quan về việc sử dụng định dạng ảnh DICOM trong y tế, ứng dụng của ảnh DICOM để phục vụ các bác sĩ giải quyết các vấn đề về chuẩn đoán bệnh.

Ngoài ra, cùng với nhu cầu phát triển của Internet và các thiết bị di động, các bác sĩ có thể linh động hơn trong công việc, và đưa ra các chuẩn đoán kịp thời cho bệnh nhân.

***3. Ứng dụng của ảnh DICOM trên trình duyệt web***

Trong thời đại hiện nay, các thiết bị di động hay máy tính đều có thể truy cập đến các nguồn tài liệu ở bắt kì đâu thông qua mạng Internet. Các thiết bị di động hay máy tình cần có một phần mềm gọi là trình duyệt web cho phép người sử dụng xem và tương tác với các văn bản, hình ảnh, đoạn phim, nhạc, trò chơi và các thông tin khác ở trên một trang web thông qua một địa chỉ web trên mạng toàn cầu hoặc mạng nội bộ. Trình duyệt web cho phép người sử dụng truy cập các thông tin trên các trang web một cách nhanh chóng và dễ dàng thông qua các liên kết đó. Trình duyệt web đọc định dạng HTML để hiển thị, do vậy một trang web có thể hiển thị khác nhau trên các trình duyệt khác nhau.

Mặc khác các ảnh định dạng DICOM cũng được coi như một dạng tài liệu chứa các thông tin như dữ liệu hình ảnh, thông tin về bệnh nhân, về loại máy tạo ra bức ảnh… Nên có thể sử dụng trình duyệt web để xem các thông tin này thông qua sử dụng các ngôn ngữ như JavaScript, HTML, CSS chạy trên trình duyệt web.



Hình 1. Sử dụng trình duyệt web trên các thiết bị khác nhau

***4. Phạm vi của đề tài***

Phạm vi của đề tài nhầm hiện thực hóa việc truyền ảnh DICOM từ server lên trình duyệt web và biểu diễn ảnh DICOM trên trình duyệt web.

# **PHẦN NỘI DUNG**

## **PHẦN 1 – KHÁI QUÁT VỀ ẢNH DICOM**

* 1. ***Khái niệm cơ bản về định dạng ảnh DICOM***

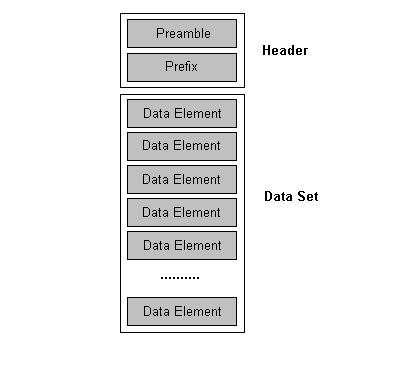
#### **1.1.1. Định dạng ảnh DICOM là gì ?**

Digital Imaging and Communications in Medicine (DICOM) là tiêu chuẩn để xử lý, lưu trữ, in ấn và thu/nhận hình ảnh trong y tế. Tiêu chuẩn này bao gồm cả việc định nghĩa cấu trúc tập tin và giao thức truyền thông tin. Giao thức truyền thông tin là một giao thức ứng dụng sử dụng nền tảng TCP/IP để giao tiếp lẫn nhau giữa các hệ thống. Các tập tin DICOM có thể được trao đổi lẫn nhau giữa các hệ thống khi các hệ thống này có khả năng thu nhận hình ảnh và dữ liệu bệnh nhân theo định dạng DICOM. Hiệp hội các nhà sản xuất điện-điện tử Hoa kỳ năm giữ bản quyền của tiêu chuẩn (DICOM broschure). Tiêu chuẩn này được phát triển bởi Ủy ban tiêu chuẩn DICOM, với các thành viên thuộc Hiệp hội Các nhà sản xuất Điện-Điện tử Hoa Kỳ.

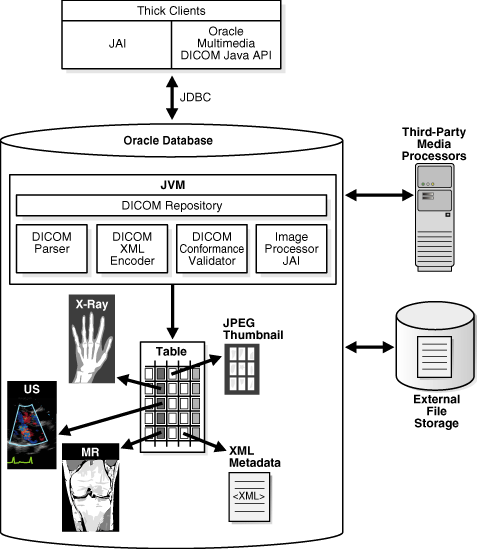
Tiêu chuẩn DICOM cho phép việc tích hợp dễ dàng các máy thu nhận hình ảnh, server, trạm làm việc (workstation), máy in và các thiết bị phần cứng khác có nối mạng từ các nhà sản xuất khác nhau vào trong hệ thống PACS. Các thiết bị khác nhau được đi kèm một bảng đáp ứng các tiêu chuẩn DICOM để làm rõ các lớp dịch vụ mà thiết bị này hỗ trợ. DICOM đã dần dần được chấp nhận rộng rãi ở các bệnh viện và phòng khám.

#### **1.1.2. Định dạng dữ liệu DICOM**

Định dạng dữ liệu DICOM khác so với các định dạng khác là các nhóm thông tin được tích hợp vào bên trong tập tin DICOM. Do đó, nếu một tập tin X-quang phổi theo định dạng DICOM sẽ chứa các thông tin như tên bệnh nhân, mã ID bệnh nhân, vì vậy hình ảnh sẽ không bao giờ bị thất lạc thông tin. Một đối tượng dữ liệu DICOM sẽ bao gồm các thuộc tính như họ và tên, mã ID,v.v. và cũng bao gồm các thuộc tính đặc biệt chứa dữ liệu Pixel của hình ảnh.



*Hình 2. Cấu trúc một tập tin DICOM cơ bản*



*Hình 3. Kiến trúc lưu trữ và truyền dẫn tập tin DICOM của Oracle*

## **PHẦN 2 – GIỚI THIỆU CƠ BẢN VỀ ORTHANC, NODE.JS, EXPRESS.JS, ANGULARJS VÀ CORNERSTONE**

1. 1. ***Orthanc – DICOM server***

#### **2.1.1. Orthanc là gì ?**

Orthanc nhằm mục đích cung cấp một máy chủ DICOM độc lập và mạnh mẽ . Nó được thiết kế để cải thiện phương thức truy cập và lưa trữ ảnh DICOM trong bệnh viện và hỗ trợ nghiên cứu về phân tích tự động các hình ảnh y tế. Orthanc cho phép người dùng tập trung vào nội dung của các tập tin DICOM, che giấu sự phức tạp của định dạng và giao thức tập tập tin DICOM.

Orthanc có thể chạy dưới bất kỳ máy tính cài hệ điều hành Windows, Linux hoặc OS X thành một nơi lưu trữ tập tin DICOM (nói cách khác là một hệ thống mini-PACS). Kiến trúc của nó thì nhẹ và độc lập, có nghĩa là không yêu cầu quản trị cơ sở dữ liệu một cách phức tạp và không cần phụ thuộc vào một bên thứ ba.

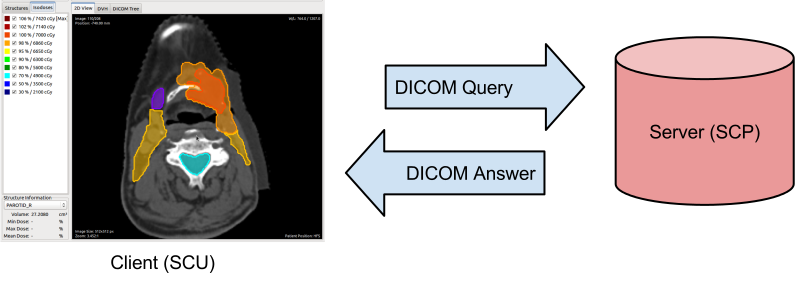
Điều làm cho Orthantc đặc biệt là nó cung cấp một API RESTful. Nhờ tính năng chính này, bạn có thể điều khiển Orthanc từ bất kỳ ngôn ngữ máy tính nào. Các tập tin DICOM của các hình ảnh y tế đã lưu trữ có thể được tải xuống dưới dạng tập tin JSON. Hơn nữa, các hình ảnh chuẩn PNG có thể được tạo ra từ các tập tin DICOM được lưu trữ trên Orthanc.

Orthanc cung cấp một cơ chế bổ sung để thêm các mô đun mới, mở rộng các tính năng cốt lõi REST API của nó. Có thể tích hợp vào nó một chương trình xem ảnh dưới dạng Web, có thể kết nối nó với cơ sở dữ liệu PostgreSQL hoặc nó như là một plugin hỗ trợ xây dựng các trang DICOMweb.

#### **2.1.2. Giao thức mạng DICOM**

Giao thức DICOM là một trong những ví dụ sớm nhất của [các dịch vụ Web](https://en.wikipedia.org/wiki/Web_service), rất lâu trước khi [SOAP](https://en.wikipedia.org/wiki/SOAP) (Simple Object Access Protocol) hay [REST](https://en.wikipedia.org/wiki/Representational_state_transfer) (Representational state transfer) ra đời. Nó cho phép:

1. Kiểm tra kết nối giữa hai thiết bị ( [C-Echo](http://book.orthanc-server.com/dicom-guide.html#dicom-echo) ).
2. Gửi hình ảnh từ thiết bị hình ảnh địa phương đến một thiết bị từ xa ( [C-Store](http://book.orthanc-server.com/dicom-guide.html#dicom-store) ).
3. Tìm kiếm nội dung của một thiết bị từ xa ( [C-Find](http://book.orthanc-server.com/dicom-guide.html#dicom-find) ).
4. Lấy hình ảnh từ một thiết bị từ xa ( [C-Move](http://book.orthanc-server.com/dicom-guide.html#dicom-move) ).



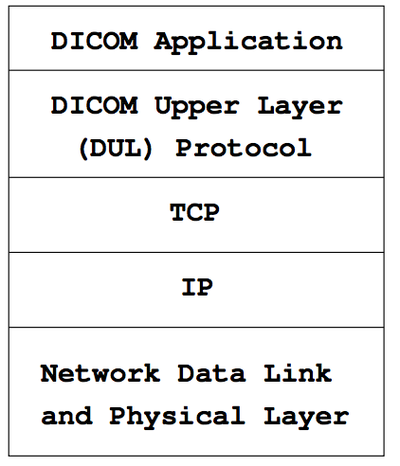
*Hình 4. Truy vấn thông qua giao thức DICOM*

Khách hàng sử dụng dịch vụ DICOM được gọi là người sử dụng dịch vụ SCU (service class user), và máy chủ xử lý yêu cầu được gọi là nhà cung cấp dịch vụ SCP (service class provider). Máy khách gửi một yêu cầu được mã hoá dưới dạng tệp DICOM (lệnh - command) và máy chủ trả lời với tệp DICOM.

Kết nối của một khách hàng DICOM với một máy chủ DICOM được gọi là một sự kết hợp (association). Một sự kết hợp như vậy bắt đầu bằng một cái bắt tay, nơi khách hàng và máy chủ đồng ý về những lệnh nào có thể được trao đổi giữa chúng, và những [cú pháp chuyển tiếp](http://book.orthanc-server.com/dicom-guide.html#dicom-pixel-data) nào được hỗ trợ. Kết quả của đàm phán này được gọi là ngữ cảnh trình bày (presentation context). Một khi sự kết hợp này được chấp nhận lẫn nhau, một kênh giao tiếp dược tạo ra, có thể được sử dụng để liên tiếp gửi nhiều lệnh độc lập.

#### **2.1.3. Tham số của một máy chủ DICOM**

Giao thức DICOM được thiết kế để làm việc trên [các liên kết](https://en.wikipedia.org/wiki/Point-to-point_(telecommunications)) point-to-point. Giao thức DICOM được sử dụng thông qua phương thức [TCP/IP](https://en.wikipedia.org/wiki/Internet_protocol_suite).



*Hình 5. Các lớp của một mô hình truyền dẫn DICOM*

Điều này hàm ý rằng một máy chủ DICOM có thể được xác định bằng cách xác định các thông số từ socket của nó như:

1. Địa chỉ IP của nó.
2. Cổng TCP của nó (cổng DICOM tiêu chuẩn là 104, nhưng theo mặc định, Orthanc sử dụng cổng 4242).

Hơn nữa, mỗi thiết bị sử dụng hình ảnh (có thể là client hoặc server) phải được liên kết với một tên biểu tượng được gọi là tên tiêu đề thực thể ứng dụng (AET- Application Entity Title) . Các AET được giả định là duy nhất bên trong Intranet của bệnh viện. Để có sự tương thích tốt nhất giữa các nhà cung cấp, AET chỉ nên được tạo bởi các ký tự chữ và số ở dạng chữ hoa và ký tự " -" và " \_" và độ dài của nó phải dưới 16 ký tự.

Với địa chỉ IP, cổng TCP và AET mô tả tất cả các thông số của máy chủ DICOM. Các quản trị viên quản lí các hình ảnh y tế cần theo dõi cẩn thận các thông số này cho từng thiết bị hình ảnh và phải xác định chính sách của bệnh viện để chỉ định AET cho các thiết bị mới.

#### **2.1.4. Cấu hình Orthanc**

Orthanc có thể hoạt động như là một máy khách DICOM (DICOM client) hoặc như là một máy chủ DICOM (DICOM server), tùy thuộc vào các thông số trong [tập tin cấu hình](http://book.orthanc-server.com/users/configuration.html#configuration) của nó . Để cấu hình máy chủ DICOM của Orthanc, các tùy chọn sau đây đặc biệt quan trọng:

1. *DicomServerEnabled* phải được đặt thành true.
2. *DicomAet* phải được đặt thành AET được dành riêng cho Orthanc.
3. *DicomPort* chỉ định cổng TCP của máy chủ DICOM.

Để cấu hình Orthanc như là một máy khách DICOM, bạn phải liệt kê các máy chủ DICOM từ xa được biết là Orthanc vào *DicomModalitiestùy* chọn này. Đối với mỗi máy chủ từ xa, bạn phải thực hiện theo thứ tự sau:

1. Đặt một các tên tượng trưng cho máy chủ sẽ được hiển thị bởi [Orthanc Explorer](http://book.orthanc-server.com/users/cookbook.html#orthanc-explorer) (có thể là AET của nó).
2. AET của máy chủ từ xa.
3. Địa chỉ IP hoặc tên máy chủ của nó.
4. Cổng DICOM (có thể là 104 hoặc 4242).

Tất nhiên, sau khi thay đổi cấu hình của Orthanc, phần mềm phải được khởi động lại để đưa các thông số mới vào tài khoản.

#### **2.1.5. Sử dụng giao thức HTTP thay vì giao thức DICOM**

Thực tế là các giao thức DICOM vốn nhắm vào mục tiêu là các Intranet của một bệnh viện đơn lẻ, không phải là Internet hoặc đám mây (Clound). Giao thức này có thể bị chặn bởi tường lửa đi ra và không có quy ước đảm bảo rằng các AET là duy nhất trên toàn cầu hoặc trên tất cả các bệnh viện. Hơn nữa, mặc dù giao thức DICOM hỗ trợ mã hóa TLS, tính năng này ít khi được kích hoạt.

Tùy thuộc vào các ứng dụng khác nhau, chúng ta có thể sử dụng giao thức HTTP (Hypertext Transfer Protocol) trong việc truy xuất ảnh DICOM. Giao thức này dựa trên môi trường Web, có thể quen thuộc với các bác sĩ, nhà vật lý, kỹ sư phần mềm để dễ dàng làm việc hơn, có thể được mã hóa minh bạch ( HTTPS) và có thể tương thích với tình huống kết nối nhiều bệnh viện với nhau. Vì vậy, chúng ta có hai khả năng:

1. Sử dụng cơ chế [ngang hàng Orthanc](http://book.orthanc-server.com/faq/features.html#peers) (Orthanc peer). Bởi vì mỗi máy chủ Orthanc đi kèm với [REST API](http://book.orthanc-server.com/users/rest.html#rest) được xây dựng cho nó, các hệ thống từ xa có thể có được một chương trình kiểm soát đầy đủ đối với nội dung của Orthanc. Truy cập này có thể được bảo vệ bằng xác thực HTTP và [mã hóa SSL](http://book.orthanc-server.com/faq/https.html#https) (Secure Sockets Layer). [Orthanc Explorer](http://book.orthanc-server.com/users/cookbook.html#orthanc-explorer) với giao diện rõ ràng có thể sử dụng cơ chế ngang hàng này để gửi các tập tin DICOM qua HTTP hoặc HTTPS.
2. Sử dụng [theo tiêu chuẩn DICOMweb](http://www.dicomweb.org/), là một phần mở rộng của tiêu chuẩn DICOM, xác định cách truy cập nội dung của một máy chủ DICOM từ xa thông qua HTTP hoặc HTTPS. Bởi vì cơ chế ngang hàng bị ràng buộc với Orthanc, DICOMweb cung cấp truy cập ít hơn nhưng có tính năng truy cập tổng quát hơn cho các máy chủ từ xa. Điều quan trọng là, một ứng dụng hoặc một trang Web có thể gắn một [plugin DICOMweb để truy cập đến máy chủ Orthanc](http://www.orthanc-server.com/static.php?page=dicomweb) bất kì lúc nào.

***2.2. Giới thiệu Node.js***

#### **2.2.1. Node.js là gì ?**

Node.js là một nền tảng Server side được xây dựng dựa trên Javascript Engine (V8 Engine). Node.js được phát triển bởi Ryan Dahl năm 2009.

Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

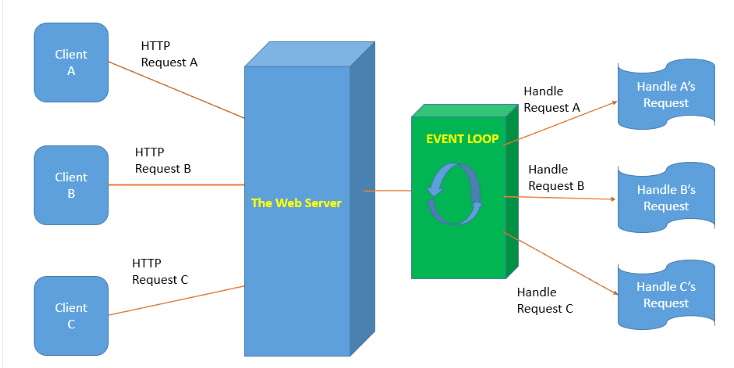
Node.js là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng Node.js được viết bằng Javascript và có thể chạy trong môi trường Node.js trên hệ điều hành Window, Linux...

Node.js cũng cung cấp cho chúng ta các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node.js với các phần mở rộng.

#### **2.2.2. Đặc điểm của Node.js**

*Chạy rất nhanh:* Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.

*Không đồng bộ và Phát sinh sự kiện (Event Driven):* Tất các các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking (khóa). Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dự liệu. Server chuyển sang một API sau khi gọi nó và có cơ chế thông báo về Sự kiện của Node.js giúp Server nhận đựa phản hồi từ các API gọi trước đó.

****

*Hình 6. Cơ chế Event-Loop của Node.js*

*Không đệm:* Ứng dụng Node.js không lưu trữ các dữ liệu buffer.

Dưới đây là các lĩnh vực mà Node.js được sử dụng:

* Các ứng dụng về I/O
* Các ứng dựng về luồng dữ liệu
* Các ứng dụng về dữ liệu hướng đến thời gian thực
* Các ứng dụng dựa vào JSON APIs
* Các ứng dụng Single Page Application

Node.js không nên sử dụng ở các ứng dụng đòi hỏi cao về mặc tính toán hoặc hiệu năng của CPU.

#### **2.2.3. Express framework**

Express là một framework nhỏ và tiện ích để xây dựng các ứng dụng web, cung cấp một lượng lớn của tính năng mạnh mẽ để phát triển các ứng dụng web và mobile. Nó rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Dưới đây là các tính năng cơ bản của Express framework.

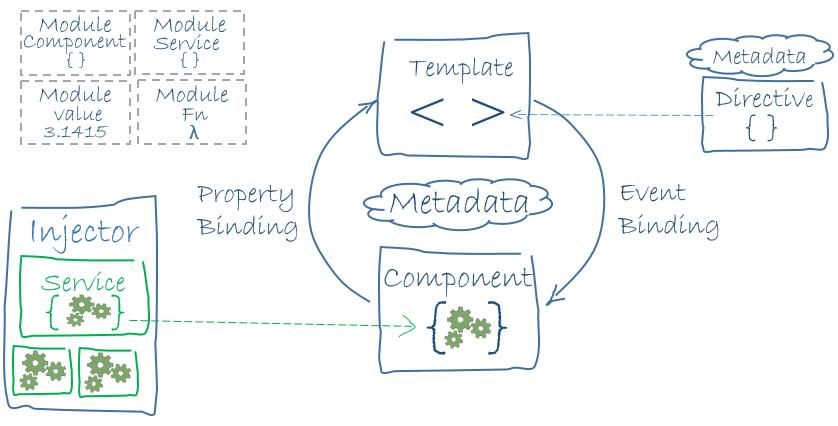
* Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
* Định nghĩ bảng routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

***2.3. AngularJS***

#### **2.3.1. AngularJS là gì ?**

AngularJS là một framework ứng dụng web mã nguồn mở. Nó được phát triển lần đầu năm 2009 bởi Misko Hevery và Adam Abrons. Hiện tại nó được duy trì bởi Google.

AngularJS là một framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: *Data binding* và *Dependency injection* của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Nó xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.



*Hình 7. Các thành phần cơ bản của AngularJS 2*

#### **2.3.2. Các tính năng của AngularJS**

* AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).
* AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.
* Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự động xử lý mã JavaScript để phù hợp với mỗi trình duyệt.
* AngularJS có mã nguồn mở, miễn phí hoàn toàn, được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0.

Nhìn chung, AngularJS là một framework để tạo các ứng dụng lớn, các hiệu năng cao trong khi giữ cho chúng có thể dễ dàng duy trì.

#### **2.3.3. Đặc điểm của AngularJS**

* AngularJS cung cấp khả năng tạo ra các *Single Page Application* một cách rất rõ ràng và dễ dàng để duy trì.
* AngularJS cung cấp khả năng *Data binding* tới HTML do đó giúp người dùng cảm giác linh hoạt, thân thiện.
* AngularJS code dễ dàng khi *unit test*.
* AngularJS sử dụng *dependency injection*.
* AngularJS cung cấp khả năng tái sử dụng các *component (thành phần)*.
* Với AngularJS, lập trình viên sẽ viết ít code hơn, với nhiều chức năng hơn.
* Với AngularJS, *view* là thành phần trong trang HTML thuần, trong khi *controller* được viết bởi JavaScript với quá trình xử lý nghiệp vụ.

Ứng dụng AngularJS có thể chạy trên hết các trình duyệt web, trên các nền tảng Android và iOS.

Mặc dù AngularJS có thể kể đến rất nhiều các ưu điểm, nhưng đến thời điểm này, nó vẫn có một số điểm yếu sau:

* Không an toàn: Là một JavaScript framework, ứng dụng được viết bởi AngularJS không an toàn. Phải có các tính năng bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
* Nếu người sử dụng ứng dụng của bạn vô hiệu hóa JavaScript thì họ chỉ nhìn được trang cơ bản, không thấy gì thêm.

***2.4. Cornerstone***

#### **2.4.1. Giới thiệu thư viện Cornerstone**

Cornerstone (Core) là một dự án mã nguồn mở với mục tiêu cung cấp nền tảng hình ảnh y tế dựa trên Web. Cornerstone là một thư viện JavaScript nhẹ để hỗ trợ hiển thị các hình ảnh y tế trong các trình duyệt web hiện đại và hỗ trợ các thành phần canvas HTML5.

#### **2.4.2. Các tính năng chính**

* HTML5/Javascript dựa trên thư viện này có dễ dàng thêm hình ảnh y học tương tác vào các ứng dụng web
* Phục vụ như một nền tảng để xây dựng các ứng dụng hình ảnh y tế phức tạp hơn
* Hỗ trợ tất cả trình duyệt dựa trên HTML5 bao gồm điện thoại di động, máy tính bảng và máy tính để bàn
* Hiển thị tất cả các định dạng hình ảnh y học phổ biến (ví dụ như màu xám 8 bit, màu xám 16 bit, màu RGB)
* Hiển thị hình ảnh với hiệu suất cao
* Truy xuất hình ảnh từ các hệ thống khác nhau với các giao thức khác nhau thông qua thiết kết Image Loader plugin
* Hỗ trợ API để thay đổi các thuộc tính khung nhìn (vd: ww/wc, zoom, pan, invert)

## **PHẦN 3 – MÔ HÌNH CLIENT–SERVER**

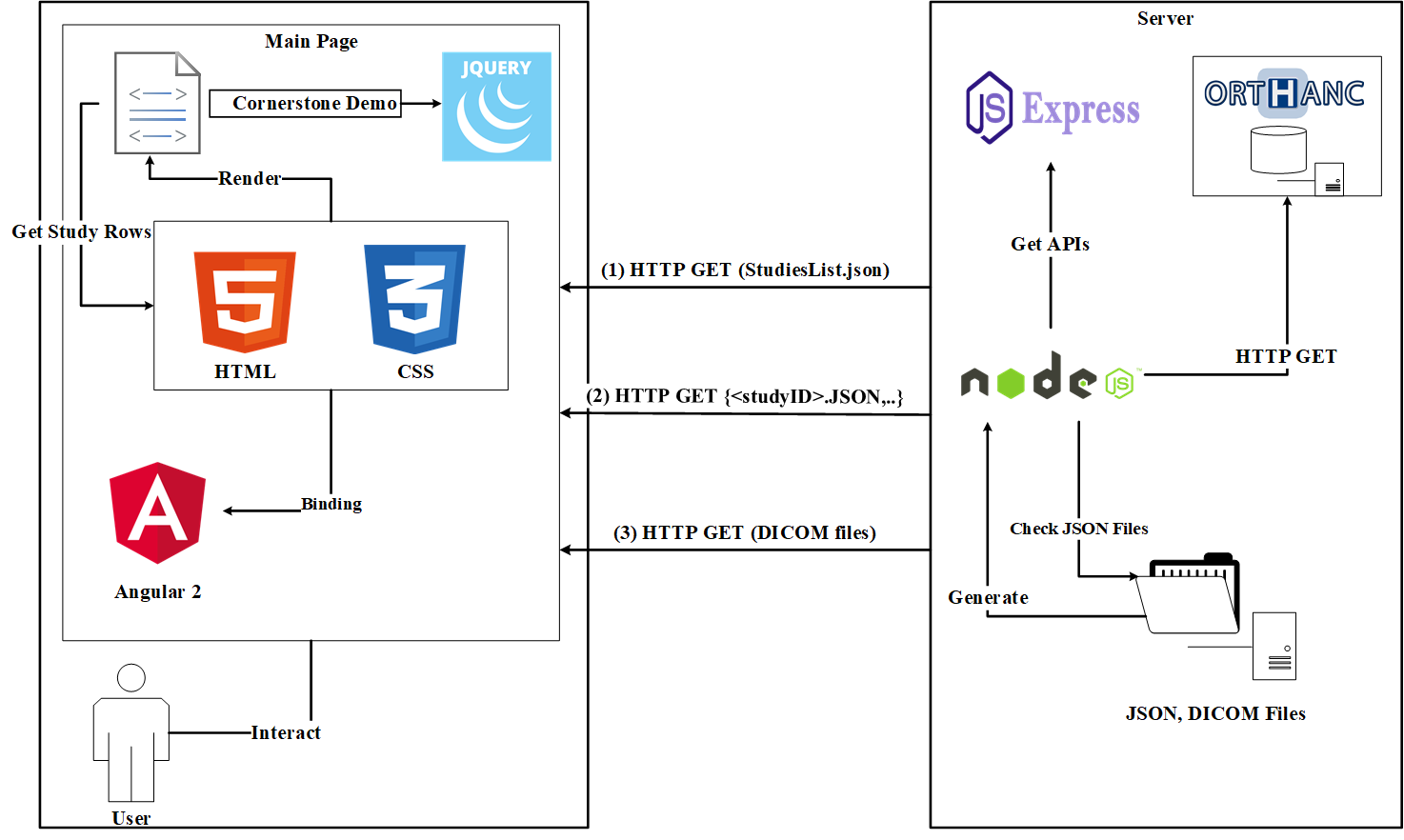
### ***3.1. Khái niệm về REST***

*Representational state transfer (REST)* hoặc các dịch vụ RESTful là một cách để cung cấp khả năng tương tác giữa các hệ thống máy tính trên Internet. Các dịch vụ Web tuân thủ REST yêu cầu hệ thống phải truy cập và thao tác trên văn bản của tài nguyên Web, sử dụng một bộ tiêu chuẩn và được xác định trước tập của các hoạt động không trạng thái. Các hình thức khác của dịch vụ Web tồn tại các tập hợp riêng tùy ý của các hoạt động như WSDL (Web Service Definition Language) và SOAP (Simple Object Access Protocol).

Tài nguyên Web lần đầu tiên được định nghĩa trên World Wide Web như là các tài liệu hoặc các tệp được xác định bởi các URL (Uniform Resource Locator) của chúng, nhưng ngày nay chúng có một định nghĩa chung chung và trừu tượng hơn bao gồm mọi thứ hoặc thực thể có thể được xác định, đặt tên, truy cập hoặc xử lý bằng cách nào đó trên Web. Trong dịch vụ RESTful Web, các yêu cầu được thực hiện đến URI của tài nguyên sẽ thực hiện một phản hồi được chứa trong XML, HTML, JSON hoặc một số định dạng được xác định khác. Phản hồi có thể khẳng định rằng một số thay đổi đã được thực hiện đối với tài nguyên đã lưu trữ và nó có thể cung cấp liên kết siêu văn bản tới các tài nguyên có liên quan hoặc bộ sưu tập các tài nguyên. Sử dụng HTTP thường là cách phổ biến, vì nó cung cấp các phương thức GET, POST, PUT, DELETE... đã được định nghĩa trước.

Với việc sử dụng các giao thức không trạng thái (stateless) và các hoạt động tiêu chuẩn, các hệ thống REST đạt được hiệu năng, độ tin cậy và khả năng phát triển nhanh chóng bằng cách tái sử dụng các thành phần có thể được quản lý và cập nhật mà không ảnh hưởng đến toàn bộ hệ thống, ngay cả khi nó đang chạy.

### ***3.2. Mô hình client-server của đề tài***

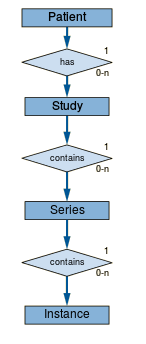


*Hình 8. Mô hình client-server của đề tài*

#### **3.2.1. Mô hình phía server**

Ở đây chúng ta sử dụng ExpressJS là một Framework được sử dụng trên trên NodeJS, mục đích của Framework này để thiết lập định tuyến (Routing) khi người dùng thực hiện gửi một yêu cầu HTTP request đến máy chủ. Khi máy chủ xác định được yêu cầu của người dùng từ phương thức HTTP được gửi đến, nó sẽ gửi ngược lại các thành phần cấu trúc nên giao diện và chức năng người dùng như HTML, CSS, JavaScript. Ngoài ra, ExpressJS cho phép người dùng được truy cập đến các tài nguyên cố định ở máy chủ, ở đây là thư mục chửa các tập tin JSON, DICOM được hình thành từ NodeJS.

Chúng ta thấy Orthanc vừa là một Web Service, vừa là là một cơ sở dữ liệu quản lí các tập tin ảnh DICOM. Chúng ta thực hiện các truy vấn trên Orthanc thông qua các phương thức được cung cấp từ REST API của nó. Vì vậy, NodeJS có thể sử dụng các API của Orthanc để giúp chúng ta lấy ảnh DICOM và các thông tin của ảnh đó. Ở đây chúng ta thực hiện các truy vấn thông qua các Study (triệu chứng hoặc bộ phận cơ thể) của một bệnh nhân.



*Hình 9. Lược đồ quan hệ cơ sở dữ liệu của ảnh DICOM trong Orthanc*

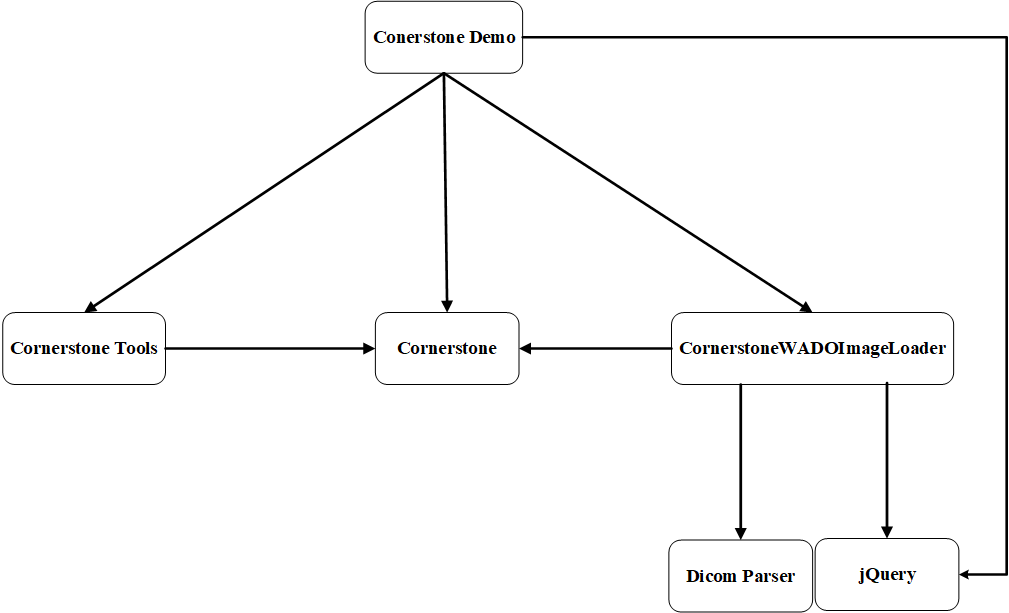
Hình trên mô tả tương ứng một bệnh nhân (Patient) sẽ có nhiều phần chụp các bộ phận hoặc triệu chứng khác nhau (Study), mỗi bộ phận hoặc triệu chứng có thể được chụp ở thành nhiều chuỗi khác nhau (Series), mỗi chuỗi tương với nhiều thể hiện (Instance).

NodeJS sẽ sử dụng các thông tin về ảnh DICOM được lưu trữ trên Orthanc, để hình thành các tập tin JSON và lấy ảnh DICOM theo Study để lưa trữ trên một thư mục tĩnh được ExpressJS cung cấp. Việc thay đổi ảnh DICOM trên Orthanc được NodeJS kiểm tra liên tục, nhằm cập nhật lại các thông tin của các tập tin JSON và ảnh DICOM được lưu trữ trước đó trên thư mục mà ExpressJS cung cấp.

#### **3.3.1. Mô hình phía client**

AngularJS có nhiệm vụ ràng buộc (binding) dữ liệu tiếp nhận được lên giao diện chương trình thông qua cơ chế HTML directive. Tập tin StudyList.json được AngularJS lấy về thông qua phương thức HTTP GET từ thư mục tĩnh ở server (được ExpressJS cung cấp). Khi dữ liệu ở Server thay đổi, người dùng có thể cập nhật dữ liệu trên trang chính (Main Page) mà không cần tải lại trang (cơ chế Single Page của Angular).

Cornerstone giúp hiển thị ảnh DICOM trên trình duyệt Web.



*Hình 10. Các thành phần chính của thư viện Cornerstone được sử dụng trong đề tài*

Ở sơ đồ trên chúng ta thực sử dụng các thành phần của thư viện Cornerstone như:

* Cornerstone (Core) được Cornerstone Tools và CornerstoneWADOImageLoader gọi các phương thức để thực hiện các chứ năng hiển thị và các tác vụ truyền tải ảnh.
* Cornerstone Tools là thư viện được xây dựng trên nền tảng Cornerstone (Core), cung cấp các công cụ phổ biến và cần thiết để làm việc với hình ảnh y tế.

Một số tính năng được cung cấp trong Cornerstone Tools như:

* Công cụ làm việc với một ảnh đơn: thay đổi độ sáng và độ tương phản (WW/WL), phóng to thu nhỏ (Zoom), di chuyển hình ảnh trên khu nhìn (Pan), xác định chiều dài của hai điểm trên ảnh (Length), Rectangle ROI(xác định đối tượng bằng kích thước hình chữ), Elliptical ROI (xác địng đối tượng bằng kích thức hinh Ellipse), xác định giá trị màu của Pixel (Pixel Probe), xác định gốc (Angle)
* Công cụ làm việc với nhìn ảnh trong ngăn xếp (Stack): cuộn ảnh (Scroll), tự động trình chiếu các ảnh liên tục (Cine / Playing Clips), các đường thẳng để tham chiếu trên ảnh (Cross reference lines)
* Làm việc với chuỗi thời gian (Timeseries)
* Quản lý đo lường
* Công cụ đồng bộ hóa: Xếp chồng theo chỉ mục hình ảnh, xếp chồng theo vị trí hình ảnh, thể hiện ảnh bởi Zoom và Pan
* Hỗ trợ ràng buộc mỗi công cụ bởi các phím nhấn chuột khác nhau (mouse inputs): nút chuột trái, nút chuột giữa, nút chuột phải, con lăn chuột
* Hỗ trợ màng hình cảm ứng: kéo (Drag), vuốt (Pinch)
* Cung cấp một Framework về công cụ được sử dụng để đơn giản hóa việc phát triển các công cụ mới
* Cung cấp API để truy cập các dữ liệu do đạt cho các mục đích tuần tự (Serialization purposes)
* CornerstoneWADOImageLoader là một trình tải các ảnh DICOM loại P10 thông qua HTTP (WADO-URI) hoặc DICOMWeb (WADO-RS). Điều này có thể được sử dụng để tích hợp nền tảng với các máy chủ WADO-URI, máy chủ DICOMWeb hoặc các máy chủ khác cung cấp phương thức HTTP dựa trên việc trả lại các ảnh DICOM loại P10. CornerstoneWADOImageLoader phụ thuộc vào các thư viện bên ngoài như:
* jQuery: là một thư viện của JavaScript, được tạo bởi John Resig vào năm 2006. jQuery làm đơn giản hóa việc truyền tải HTML, xử lý sự kiện, tạo hiệu ứng động và tương tác Ajax để thực hiện các tác vụ truyền và nhận dữ liệu
* dicomParser: là một thư viện nhỏ cung cấp tổi thiểu các chức năng dùng để đọc, phân tích, sửa đổi, soạn thảo các dòng dữ liệu byte DICOM P10 trong các trình duyệt web dựa trên HTML5. DicomParser dễ sử dụng và không phụ thuộc bên ngoài.
* Cornerstone (Core)

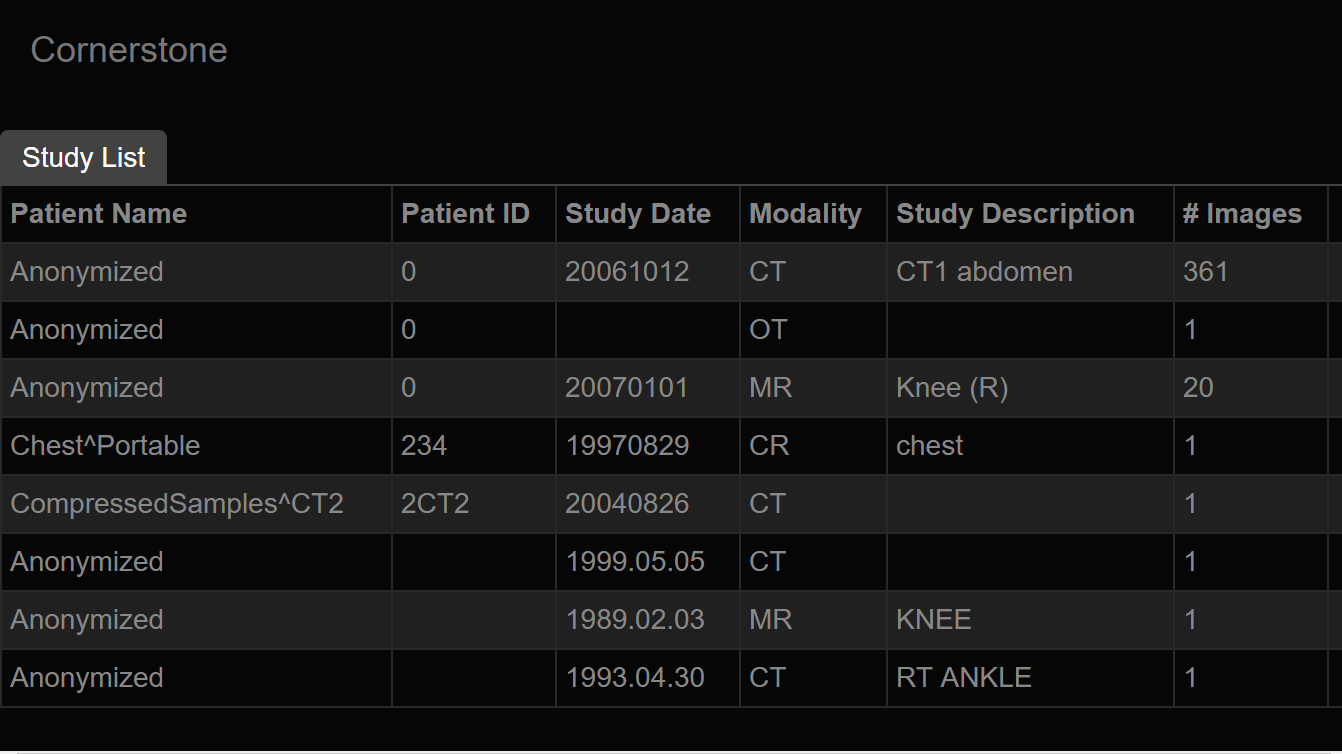
Một số tính năng được cung cấp trong cornerstoneWADOImageLoader như:

* Thực hiện tải các hình DICOM P10 cho Cornerstone (Core) thông phương thức HTTP GET: có thể được sử dụng với một máy chủ WADO-URI, máy chủ Orthanc, các máy chủ nào trả về các trường hợp DICOM P10 thông qua HTTP GET
* Thực hiện tải ảnh cho Cornerstone thông qua WADO-RS (DICOMWeb)
* Hỗ trợ giải mã đa dạng các loại ảnh DICOM bị nén như JPEG, JPEG2000, JPEG-LS, RLE
* Framework để thực hiện các tác vụ chuyên sâu trên CPU như: giải mã hình ảnh, các tác vụ xử lý ảnh (Image Processing)
* Cornerstone Demo: sẽ hiện thực gắn sự kiện “click” lên các dòng (row) của bảng hiện thị thông tin về các Study từ tập tin studiesList.json. Khi người dùng thực hiện “click” để yêu cầu xem các ảnh DICOM của bất kì Study nào, thì một Tab sẽ hiển thị ra sẽ tải các phần giao diện (load template) viewport và Study viewer để hiển thị ảnh DICOM đã qua quá trình xử lí từ các thành phần Cornerstone Tools, Cornerstone (Core) và CornerstoneWADOImageLoader.

#### **3.3.2. Mục đích của mô hình**

Tách bạch hai phần client và server độc lập với nhau. Mặc khác các thành phần bên trong server hoặc client chủ yếu gọi các thư viện cung cấp các API, nên việc phát triển ứng dụng có thể trở nên nhanh hơn. Về lâu dài, do các thành phần độc lập nhau nên việc bảo trì hệ thống sẽ không tốn quá nhiều chi phí.

#### **3.3.3. Giao diện chương trình**



*Hình 11. Giao diện hiển thị chi tiết thông tin về các Study của các bệnh nhân*



*Hình 12. Giao diện hiển thị ảnh DICOM*

# **PHẦN KẾT LUẬN**

1. ***Tóm tắt kết quả***

Việc hiển thị ảnh DICOM trên trình duyệt web thông qua việc sử dụng phương thức HTTP để lấy dữ liệu từ server rất khả quan. Khi đó các bác sĩ có thể xem ảnh chụp của bệnh nhân ở trên bất kì thiết bị nào có hỗ trợ trình duyệt web như điện thoại thông minh, laptop, máy tính để bàn… ở bất kì đâu có mạng Internet.

Mô hình trên của đề tài cho phép người dùng truy cập được dữ liệu trên thư mục của server mà không có kiểm soát chặt chẽ, nên độ an toàn của hệ thống chưa cao, những người khác có ý đồ xấu thể lấy các tài nguyên này.

Cornerstone chỉ cho phép hiển thị các ảnh định dạng DICOM P10, nên hạn chế việc hiển thị thông tin.

1. ***Đề xuất, kiến nghị***

Nếu tiếp tục phát triển hệ thống, chúng ta nên giải quyết các khâu về bảo mật thông tin ảnh có thể bằng việc mã hóa thông tin, tích hợp thêm việc đăng nhập hệ thống thông qua user và password và đảm bảo các yêu cầu về nghiệp vụ y tế.

Dữ liệu ảnh DICOM khá lớn, nếu mạng Internet có tốc độ chậm sẽ ảnh hưởng đến việc truyền tải và hiển thị ảnh DICOM, khi đó chúng ta có thể nén ảnh khi gửi, cắt nhỏ các tập tin DICOM để giảm bớt lưu lượng trên đường truyền.

Việc xử lý ảnh DICOM đồi hỏi nhiều tài nguyên về CPU và bộ nhớ, nó có thể ảnh hưởng nhiều đến các thiết bị di động vì bị hạn chế nhiều về phần cứng, trong tương lai chúng ta cần tối ưu hệ thống nhiều hơn nữa để đáp ứng nhu cầu của các thiết bị di động.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | Understanding DICOM with Orthanc tại địa chỉ http://book.orthanc-server.com /dicom-guide.html vào ngày 29 tháng 07 năm 2017 |
| [2] | An [Orthanc](http://www.orthanc-server.com/) REST API client designed for NodeJS tại địa chỉ <https://www.npmjs> .com/package/orthanc-client vào ngày 29 tháng 07 năm 2017 |
| [3] | Cornerstone Core tại địa chỉ <https://github.com/chafey/cornerstone> vào ngày 29 tháng 07 năm 2017 |
| [4] | Cornerstone Tools tại địa chỉ <https://github.com/chafey/cornerstoneTools> vào ngày 29 tháng 07 năm 2017 |
| [5] | Cornerstone WADO Image Loader tại địa chỉ <https://github.com/chafey/> cornerstone WADOImageLoader vào ngày 29 tháng 07 năm 2017 |
| [6] | Dicom Parser tại địa chỉ <https://github.com/chafey/dicomParser> vào ngày 29 tháng 07 năm 2017 |
| [7] | Angular 2 tại địa chỉ <https://angular.io/docs> vào ngày 29 tháng 07 năm 2017 |
| [8] | Representational state transfer tại địa chỉ <https://en.wikipedia.org/wiki/>Represen tational\_state\_transfer vào ngày 29 tháng 07 năm 2017 |
| [9] | Node.js v6.11.1 Documentation tại địa chỉ [https://nodejs.org/dist/latest-v6.x/docs /api/](https://nodejs.org/dist/latest-v6.x/docs%20/api/) vào ngày 29 tháng 07 năm 2017 |
| [10] | jQuery API tại địa chỉ <http://api.jquery.com/> vào ngày 29 tháng 07 năm 2017 |
| [11] | DICOM tại địa chỉ <https://en.wikipedia.org/wiki/DICOM> vào ngày 29 tháng 07 năm 2017 |
| [12] | ExpressJS tại địa chỉ <https://expressjs.com/en/guide/routing.html> vào ngày 29 tháng 07 năm 2017 |
| [13] | HTML và CSS tại địa chỉ <https://www.w3schools.com/html/html_css.asp> vào ngày 29 tháng 07 năm 2017 |
| [14] | Template giao diện tại địa chỉ <http://chafey.github.io/cornerstoneDemo/> vào ngày 29 tháng 07 năm 2017 |