

**TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA CÔNG NGHỆ THÔNG TIN**

SỐ PHÁCH.....



**ĐỀ TÀI**

**Xây dựng ứng dụng Web ngữ nghĩa để quản lý  
Khoá luận tốt nghiệp**

**TÊN LỚP HỌC PHẦN - MÃ HỌC PHẦN**

**WEB NGỮ NGHĨA – TIN4423**

**GIẢNG VIÊN HƯỚNG DẪN: PGS. TS Hoàng Hữu Hạnh**

**HUẾ, THÁNG 1 NĂM 2023**

**TRƯỜNG ĐẠI HỌC KHOA HỌC**  
**KHOA CÔNG NGHỆ THÔNG TIN**

SỐ PHÁCH.....



**ĐỀ TÀI**

**TÊN LỚP HỌC PHẦN - MÃ HỌC PHẦN**  
**WEB NGỮ NGHĨA – TIN4423**

**Giảng viên hướng dẫn: PGS. TS Hoàng Hữu Hạnh**

**Sinh viên thực hiện: Hoàng Ngọc Thành**

**Mã sinh viên: 18T1021285**

**HUẾ, THÁNG 1 NĂM 2023**

## MỤC LỤC

<b>LỜI MỞ ĐẦU .....</b>	<b>5</b>
<b>I. Tổng quan về Web ngữ nghĩa .....</b>	<b>6</b>
1. Web ngữ nghĩa là gì? .....	6
2. Lí do lựa chọn đề tài? .....	8
2.1. Máy có thể hiểu được thông tin trên Web .....	8
2.2. Thông tin được tìm kiếm nhanh chóng và chính xác hơn .....	8
2.3. Dữ liệu liên kết động .....	8
2.4. Hỗ trợ công cụ tự động hóa .....	9
3. Kiến trúc của Web ngữ nghĩa .....	9
4. Phương pháp nghiên cứu .....	10
5. Đối tượng nghiên cứu .....	10
<b>II. Nội dung nghiên cứu .....</b>	<b>11</b>
1. Tổng quan về RDF .....	11
1.1. Khái niệm .....	11
1.2. Mục tiêu của RDF hướng tới gồm: .....	11
1.3. Những khái niệm cơ bản .....	11
1.4. Ứng dụng .....	14
2. Tổng quan về ngôn ngữ truy vấn RDF .....	14
2.1. Ngôn ngữ truy vấn SPARQL .....	14
2.2. Cú pháp của SPARQL .....	15
2.3. Các khái niệm cơ bản .....	16
3. Giới thiệu về Jena .....	16
3.1. Jena là gì? .....	16
3.2. Kiến trúc khung ứng dụng Jena .....	17
<b>III. Mô phỏng bài toán .....</b>	<b>20</b>
1. Mô tả dữ liệu .....	20
2. Truy vấn SPARQL .....	20
2.1. Hiển thị giáo viên hướng dẫn: .....	20
2.2. Hiển thị ngày báo cáo khóa luận .....	20
2.3. Hiển thị tên đề tài báo cáo khóa luận .....	21
3. Lập trình Jena .....	21
3.1. Tạo mô hình RDF .....	21
3.2. Xuất dữ liệu theo mô hình bộ ba .....	23
<b>IV. Kết Luận .....</b>	<b>24</b>
1. Kết Quả Đạt Được .....	24

2. Hạn chế .....	24
3. Hướng phát triển .....	24
<i>V. Tài liệu bài tiểu luận</i> .....	25
1. Link github project.....	25
Github: ThanhHoang76/SementicWeb .....	25
2. Tài liệu tham khảo .....	25

## LỜI MỞ ĐẦU

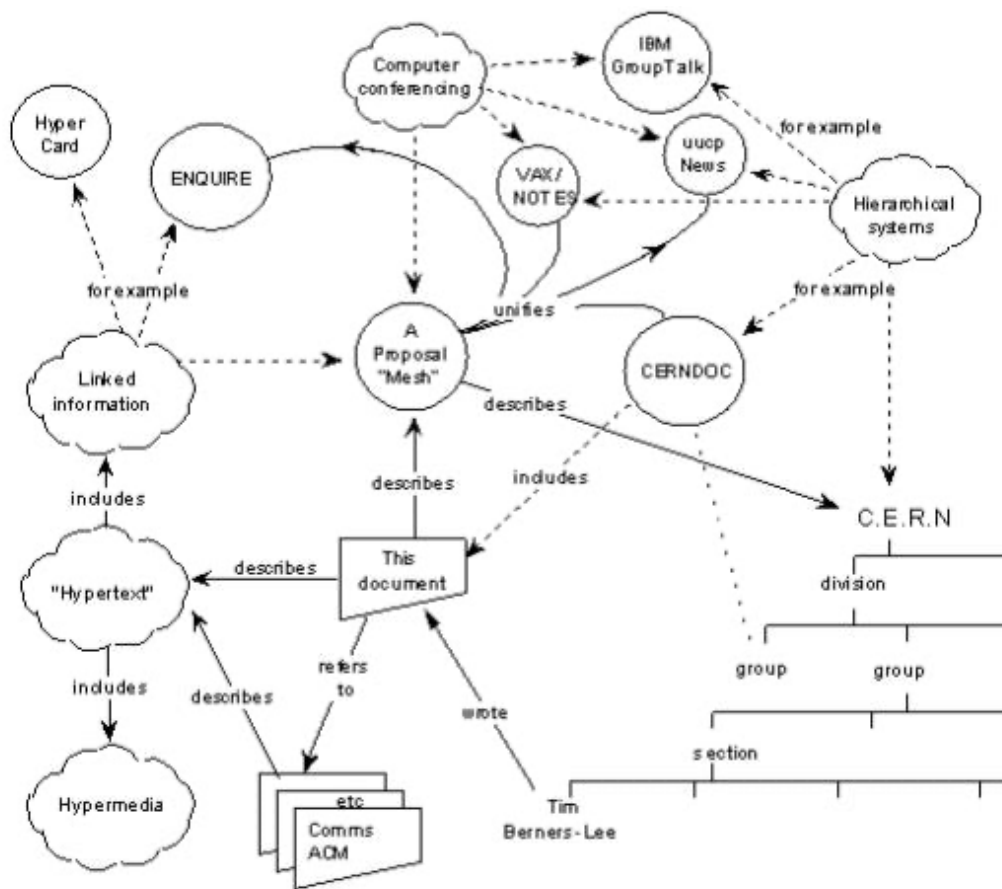
Ngày nay, sự bùng nổ tri thức cùng với các vấn đề giao lưu hội nhập quốc tế khiến mỗi chúng ta phải biết tận dụng những thành tựu của khoa học – công nghệ. Những thành tựu nổi bật nhất của công nghệ thông tin (CNTT) trong giáo dục và đào tạo hiện nay chính là dạy học trên Website. Việc các ứng dụng của CNTT đặc biệt là Internet - Website học tập góp phần rèn luyện khả năng tự học. Đây đã thực sự trở thành cầu nối giữa giáo viên (GV) người học (học viên). Nhưng với sự đơn giản của Web hiện nay đã dẫn đến một số mặt hạn chế. Chẳng hạn khi muốn tìm kiếm một thông tin nào đó, chúng ta có thể dễ dàng bị lạc hay phải đối đầu với một lượng lớn thông tin không hợp lý và không liên quan được trả về từ kết quả tìm kiếm trên Web. Như vậy câu hỏi đặt ra là làm thế nào để chúng ta có được một kết quả tìm kiếm chính xác và nhanh chóng theo đúng những gì mà chúng ta mong muốn?. Với kỹ thuật Web hiện nay rõ ràng đã cản trở khả năng phát triển thông tin của nó, gây khó khăn trong việc tìm kiếm, rút trích, bảo trì, và phát sinh thông tin. Máy tính chỉ được dùng như những thiết bị gửi và trả thông tin, chúng không thể truy xuất được những nội dung thật sự cần và do đó chúng chỉ hỗ trợ ở một mức giới hạn nào đó trong việc truy xuất và xử lý thông tin. Kết quả tất yếu là con người (người sử dụng) phải gánh trên vai trách nhiệm không những truy cập và xử lý thông tin mà còn rút trích và thông dịch mọi thông tin. Để khắc phục các yếu điểm của Web hiện tại khái niệm “Semantic Web” đã ra đời.

## I. Tổng quan về Web ngữ nghĩa

### 1. Web ngữ nghĩa là gì?

Semantic Web không là Web riêng biệt nhưng là một sự mở rộng của Web hiện tại, theo cách thông tin được xác định ý nghĩa tốt hơn, nó cho phép máy tính và người cộng tác với nhau tốt hơn.

Semantic Web được hình thành từ ý tưởng của Tim Berners-Lee, người phát minh ra WWW, URI, HTTP, và HTML.



*Đề xuất đầu tiên về WWW của Tim Berners-Lee năm 1989*

*Semantic Web* là một mạng lưới các thông tin được liên kết sao cho chúng có thể được xử lý dễ dàng bởi các máy tính ở phạm vi toàn cầu. Nó được xem là cách mô tả thông tin rất hiệu quả trên World Wide Web, và cũng được xem là một cơ sở dữ liệu có khả năng liên kết toàn cầu.

Semantic Web là một phương pháp cho phép định nghĩa và liên kết dữ liệu một cách có ngữ nghĩa hơn nhằm phục vụ cho máy tính có thể “hiểu” được. Semantic Web còn cung cấp một môi trường chia sẻ và xử lý dữ liệu một cách tự động bằng máy tính.

## 2. Lí do lựa chọn đề tài?

### 2.1. Máy có thể hiểu được thông tin trên Web

Internet ngày nay dựa hoàn toàn vào nội dung. Web hiện hành chỉ cho con người đọc chứ không dành cho máy hiểu. Semantic Web sẽ cung cấp ý nghĩa cho máy hiểu.

Ví dụ như:

- The Beatles là một ban nhạc nổi tiếng của Liverpool.
- John Lennon là một thành viên của The Beatles.
- Bản nhạc “Hey Dude” do nhóm The Beatles trình bày.

Semantic Web là tất cả những gì về cách tạo một Web mà cả người và máy có thể hiểu. Người dùng máy tính sẽ vẫn có thông tin trình bày theo cách trước đây, nhưng đối với máy tính, Semantic Web là ánh sáng soi rọi vào màn đêm của Web hiện hành. Bây giờ, máy không phải suy luận dựa vào ngữ pháp và các ngôn ngữ đánh dấu (Markup Language) nữa vì cấu trúc ngữ nghĩa của văn bản (text) thực sự đã chứa nó rồi.

### 2.2. Thông tin được tìm kiếm nhanh chóng và chính xác hơn

Với Semantic Web, việc tìm kiếm sẽ dễ dàng nếu mọi thứ được đặt trong ngữ cảnh. Ý tưởng chính yếu là toàn bộ ngữ cảnh mà người sử dụng được biết đến.

Mục tiêu của Semantic Web là phát triển các tiêu chuẩn và kỹ thuật để giúp máy hiểu nhiều thông tin trên Web hơn, để chúng tìm ra các thông tin dồi dào hơn, tích hợp, duyệt dữ liệu, và tự động hóa các thao tác.

Với Semantic Web, chúng ta không những nhận được những thông tin chính xác hơn khi tìm kiếm thông tin từ máy tính, mà máy tính còn có thể tích hợp thông tin từ nhiều nguồn khác nhau, biết so sánh các thông tin với nhau.

### 2.3. Dữ liệu liên kết động

Với Semantic Web, chúng ta có thể kết hợp các thông tin đã được mô tả và giàu ngữ nghĩa với bất kỳ nguồn dữ liệu nào.

Ví dụ, bằng cách thêm các metadata (dữ liệu về dữ liệu) cho các tài liệu khi tạo ra nó, chúng ta có thể tìm kiếm các tài liệu mà metadata cho biết tác giả là Eric Miller. Cũng thế, với metadata chúng ta có thể tìm kiếm chỉ những tài liệu thuộc loại tài liệu nghiên cứu.

Với Semantic Web, chúng ta không chỉ cung cấp các URI cho tài liệu như đã làm trong quá khứ mà còn cho con người, các khái niệm, các mối liên hệ. Như trong ví dụ trên, bằng cách cung cấp những định danh duy nhất cho mỗi con người như vai trò của



‘tác giả’ và khái niệm ‘tài liệu nghiên cứu’, chúng ta đã làm rõ người ở đây là ai và mối quan hệ tương ứng của người này với một tài liệu nào đó. Ngoài ra, bằng cách làm rõ người mà chúng ta đang đề cập chúng ta có thể phân biệt những tài liệu của Eric Miller với những tài liệu của những người khác. Chúng ta cũng có thể kết hợp những thông tin đã được mô tả ở nhiều site khác nhau để biết thêm thông tin về người này ở những ngữ cảnh khác nhau ví dụ như vai trò của anh ta ra sao khi anh ta là tác giả, nhà quản lý, nhà phát triển, ...

## 2.4. Hỗ trợ công cụ tự động hóa

Ngoài ra, chúng còn cung cấp các loại dịch vụ tự động từ nhiều vùng khác nhau: từ gia đình và các thư viện kỹ thuật số cho đến các dịch vụ kinh doanh điện tử và dịch vụ sức khỏe.v.v.

Semantic Web còn cung cấp các phương tiện để thêm các thông tin chi tiết lên Web nhằm hỗ trợ sự tự động hóa cho các dịch vụ.

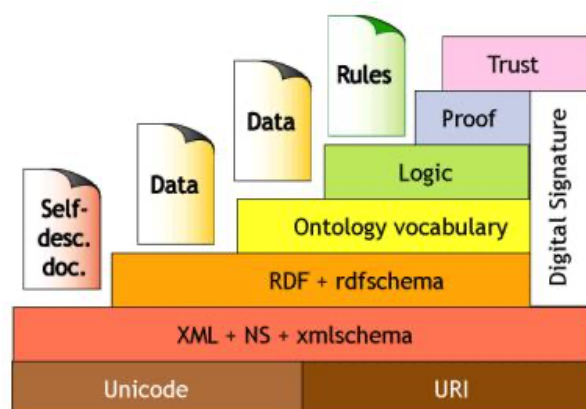
## 3. Kiến trúc của Web ngữ nghĩa

Web ngữ nghĩa được xây dựng trên nền Web truyền thống. Web ngữ nghĩa được xem như sự mở rộng của Web truyền thống, có bổ sung thêm ngữ nghĩa.

Hình dưới mô tả cấu trúc phân tầng của Web ngữ nghĩa, do Tim Berners-Lee đưa ra. Từ kiến trúc đó ta thấy hệ thống Web truyền thống của chúng ta đang ở tầng thứ hai.

- Tầng Unicode & URI: nhằm bảo đảm việc sử dụng tập ký tự quốc tế và cung cấp lớp định danh các tài nguyên trong Web ngữ nghĩa.
- Tầng XML cùng với các định nghĩa về không gian tên và lược đồ bảo

đảm rằng chúng ta có thể tích hợp các định nghĩa Web ngữ nghĩa với các chuẩn dựa trên XML khác.



*Kiến trúc của Web ngữ nghĩa được tạo ra năm 2001*

- Tầng RDF và RDF Schema (RDFS): tạo các phát biểu (statement) để mô tả các đối tượng với những từ vựng và định nghĩa của URI; các đối tượng này có thể được tham chiếu đến bởi những từ vựng và định nghĩa của URI ở trên. Đây cũng là lớp mà chúng ta có thể gán các kiểu (type) cho các tài nguyên và liên kết. Tầng Ontology Vocabulary: bộ từ vựng được xây dựng trên cơ sở tầng RDF và RDFS, cho phép thể hiện ngữ nghĩa hình thức, định nghĩa mối liên hệ giữa các khái niệm với nhau.
- Tầng Logic: xây dựng tập luật, được xem như là cơ sở luật trên Web ngữ nghĩa hỗ trợ cho quá trình suy diễn.
- Tầng Proof: chứng minh suy diễn của hệ thống bằng cách liên kết các sự kiện. Hiện nay, các nhà nghiên cứu đang xây dựng các ngôn ngữ luật cho nó như: SWRL, RuleML.
- Tầng Trust: đảm bảo tính tin cậy của các ứng dụng trên Web ngữ nghĩa, giúp ứng dụng quyết định nên hay không nên tin tưởng tài nguyên.
- Tầng Digital Signature: chữ ký điện tử, dùng để xác định chủ thể của tài nguyên.

#### 4. Phương pháp nghiên cứu

- Chúng ta sẽ nghiên cứu tài liệu trong giáo trình, sách... những nguồn tin cậy để lấy kiến thức và phục vụ trong quá trình nghiên cứu.
- Bên cạnh đó lượng kiến thức trên internet và những video bài học của người đi trước là vô cùng quan trọng, bổ sung những gì còn thiếu cho chúng ta.
- Trao đổi kiến thức từ giảng viên và học tập từ các anh, chị, bạn bè cùng nhóm học phần và cùng trong khoa công nghệ thông tin.

#### 5. Đối tượng nghiên cứu

- Ứng dụng “Quản Lý Khóa Luận Tốt Nghiệp”.
- Những thành phần trong đó
  - Mô tả dữ liệu đã đúng hay chưa
  - Lập trình với Jena
  - Đã truy vấn được dữ liệu hay chưa

## II. Nội dung nghiên cứu

### 1. Tổng quan về RDF

#### 1.1. Khái niệm

Nội dung thông tin Web được phục vụ chủ yếu cho con người, và máy móc không thể đọc và hiểu được nội dung này. Do đó, rất khó để tự động hóa bất cứ nội dung nào trên Web, ít nhất trên quy mô lớn. Hơn nữa, với lượng thông tin khổng lồ trên Web, chúng ta không thể xử lý chúng chỉ bằng phương pháp thủ công. Vì vậy, W3C đề xuất một giải pháp để mô tả dữ liệu trên Web và có thể được hiểu bởi máy móc, đó chính là RDF. RDF (viết tắt từ Resource Description Framework, tạm dịch là Framework Mô tả Tài nguyên) có nguồn gốc tạo ra từ đầu năm 1999 bởi tổ chức W3C như là 1 tiêu chuẩn để mã hóa siêu dữ liệu (metadata). Tên RDF được giới thiệu chính thức trong các tài liệu đặc tả của W3C với nội dung sơ lược.

#### 1.2. Mục tiêu của RDF hướng tới gồm:

- Xây dựng một mô hình dữ liệu đơn giản
- Xây dựng ngữ nghĩa hợp thức, có khả năng chứng minh
- Sử dụng từ vựng mở rộng dựa trên URI
- Sử dụng cú pháp dựa trên XML
- Hỗ trợ việc sử dụng các kiểu dữ liệu lược đồ XML
- Cho phép thực hiện các phát biểu về bất kỳ nguồn tài nguyên nào.

#### 1.3. Những khái niệm cơ bản

RDF bao gồm các khái niệm chính sau: Mô hình dữ liệu biểu đồ; Từ vựng dựa trên IRI; Kiểu dữ liệu; Chữ ký hiệu (Literal); Kế thừa theo thứ tự (Entailment).

• Mô hình dữ liệu biểu đồ: Mọi tài nguyên đều có những thuộc tính và các thuộc tính đó đều có giá trị. Những tài nguyên đó có thể được mô tả bằng những phát biểu để xác định những thuộc tính và những giá trị đó. RDF sử dụng một thuật ngữ đặc biệt đó là bộ ba (triple) để nói về những thành phần khác nhau của phát biểu. Trong đó một bộ ba gồm có:

+ Chủ đề (Subject): Thành phần này xác định đối tượng mà phát biểu đề cập (trong ví dụ là trang web), được gọi là chủ ngữ.

+ Vị ngữ (Predicate): Thành phần này xác định thuộc tính hay những đặc trưng của chủ ngữ của phát biểu xác định (người tạo, ngày tạo, hay ngôn ngữ trong ví dụ này).

+ Tân ngữ (Object): Thành phần này xác định giá trị của thuộc tính.

Ví dụ một phát biểu như sau:

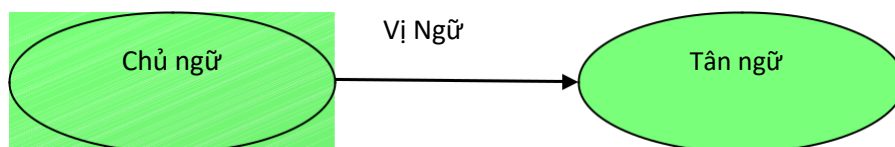
<http://www.example.org/index.html> has a creator whose value is John Smith

- Chủ ngữ là URL: <http://www.example.org/index.html>
- Vị ngữ là: creator
- Tân ngữ là: John Smith

Tuy nhiên, phát biểu trên được viết ở ngôn ngữ tiếng Anh, nó có lợi trong việc giao tiếp giữa những người sử dụng tiếng Anh, còn RDF thì tạo ra những phát biểu để máy có thể xử lý. RDF sử dụng những URI như nền tảng các cơ chế của nó để xác định những chủ ngữ, vị ngữ và tân ngữ trong những phát biểu. Như phần trước đã nói rằng RDF dựa trên ý tưởng của việc mô tả những phát biểu đơn giản về những tài nguyên, nơi mà mỗi phát biểu gồm có một chủ ngữ, vị ngữ và tân ngữ. Vì vậy trong RDF, các thành phần chủ ngữ, vị ngữ, tân ngữ của phát biểu ở trên ánh xạ sang URI có dạng:

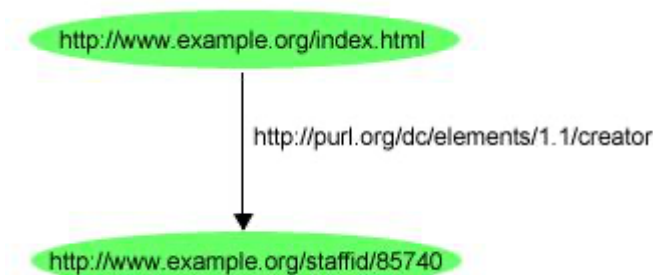
- Chủ ngữ: <http://www.example.org/index.html>.
- Vị ngữ: <http://purl.org/dc/elements/1.1/creator> (creator).
- Tân ngữ: <http://www.example.org/staffid/85740> (John Smith).

Tập các bộ ba hợp lại tạo thành đồ thị RDF. Các nút trong đồ thị có thể là các chủ ngữ và tân ngữ trong bộ ba và các cung trong đồ thị là vị ngữ. Cho nên một bộ ba còn có thể được mô tả dưới dạng nút-cung-nút. Hướng của các đồ thị rất quan trọng và nó luôn bắt đầu từ chủ ngữ đến tân ngữ



*Mối quan hệ giữa các thành phần của một phát biểu*

Vì thế phát biểu RDF ở trên sẽ được biểu diễn bằng đồ thị như hình sau:





- Chữ ký hiệu (Literal): Xác định các giá trị như những con số hay ngày tháng, thể hiện ý nghĩa hiển thị một từ vựng. Bất cứ cái gì được biểu thị bởi một giá trị Literal cũng có thể biểu diễn dưới dạng một URI. Literal có hai kiểu: Plain Literal và Typed Literal, trong đó:

+ Plain Literal: Là một chuỗi kết hợp với một thẻ (tag) ngôn ngữ tùy chọn. Điều này có thể được sử dụng cho văn bản thông thường trong một ngôn ngữ tự nhiên.

+ Typed Literal: Được tạo ra bằng cách kết hợp một chuỗi với một định danh URI để biểu diễn một kiểu dữ liệu đặc biệt nào đó. Kết quả trả về là một nút trong đồ thị tương tự như kiểu Plain Literal. Kiểu giá trị được biểu diễn bằng Typed Literal sẽ được ánh xạ đến kiểu giá trị đặc biệt mà ta đã khai báo thông qua URI

- Kiểu dữ liệu: Xác định các giá trị như là giá trị nguyên, giá trị ngày tháng, giá trị thập phân. RDF không cung cấp cơ chế định nghĩa kiểu dữ liệu mới.

- Sự kế thừa theo thứ tự (Entailment): Từ vựng trong lược đồ RDF hỗ trợ sự kế thừa theo thứ tự trong các quan hệ lớp con, quan hệ nội miền.

- Từ vựng dựa trên IRI: IRI (Internationalized Resource Identifier) là một thuộc tính xác định tài nguyên quốc tế trên web. RDF sử dụng IRI như nền tảng để xác định những chủ ngữ, vị ngữ, tân ngữ trong các phát biểu.

## 1.4. Ứng dụng

RDF là một ngôn ngữ trình diễn thông tin về tài nguyên trên web, trình diễn dữ liệu đặc tả về các nguồn tài nguyên trên web như là tiêu đề, tác giả và ngày chỉnh sửa trên một trang web, quyền tác giả và thông tin cấp phép về một tài liệu trên web... Do đó, RDF được sử dụng rộng rãi trong các ứng dụng, hệ thống thông tin nhằm khai thác, tìm kiếm và trao đổi thông tin trên web. Trong Thông tư số 01/2011/TT-BTTTT ngày 04/01/2011 của Bộ trưởng Bộ Thông tin và Truyền thông Công bố Danh mục tiêu chuẩn kỹ thuật về ứng dụng công nghệ thông tin trong cơ quan nhà nước quy định Khuyến nghị áp dụng tiêu chuẩn RDF và được xếp vào nhóm Tiêu chuẩn về tích hợp dữ liệu.

## 2. Tổng quan về ngôn ngữ truy vấn RDF

### 2.1. Ngôn ngữ truy vấn SPARQL

SPARQL (Simple Protocol And RDF Query Language) là một khuyến nghị về ngôn ngữ truy vấn và giao thức trao đổi dữ liệu RDF của W3C. SPARQL được phát

triển dựa trên các ngôn ngữ RDF trước đây là rdfDB, RDQL và SeRQL, và bổ sung một số đặc tính mới.

SPARQL có thể được sử dụng để tạo các truy vấn trên các nguồn dữ liệu đa dạng, cho dù dữ liệu được lưu giữ một cách đơn thuần như RDF hay được xem như là RDF thông qua phần mềm trung gian.

SPARQL được phát triển bởi nhóm RDF Data Access Working Group-một phần trong hoạt động của Semantic Web và đã được W3C khuyến nghị sử dụng vào tháng 01 năm 2008.

## 2.2. Cú pháp của SPARQL

Cú pháp

# định nghĩa các tiền tố

PREFIX ns: <namespaceURI>

PREFIX :<.>

# xác định kết quả

SELECT variables

# định nghĩa tập dữ liệu

[FROM <dataURI>] [FROM

NAMED <dataURI>]

# mẫu truy vấn

# điều chỉnh truy vấn

[ORDER BY variables] [OFFSET/LIMIT n]

[DISTINCT]

Một câu truy vấn bao gồm hai mệnh đề chính, mệnh đề SELECT và mệnh đề WHERE cùng các thành phần khác. Mệnh đề SELECT định danh các biến mà ứng dụng quan tâm và mệnh đề WHERE bao gồm các mẫu bộ ba (triple pattern), các thành phần khác sẽ được đề cập đến trong các phần tiếp theo.

Ngôn ngữ truy vấn SPARQL lấy thông tin từ các đồ thị RDF, nó cung cấp các tính năng sau:

- Trích xuất thông tin dưới dạng các URI, các nút trắng, các literal thường và literal có kiểu.
- Chiết xuất các đồ thị con RDF.

- Xây dựng các đồ thị RDF mới dựa trên thông tin của các đồ thị truy vấn.

Kết quả của câu truy vấn là tất cả các lời giải mà một câu truy vấn có thể phù hợp với đồ thị được truy vấn. Một câu truy vấn có thể có không, một hay nhiều kết quả.

### 2.3. Các khái niệm cơ bản

- Mẫu bộ ba (triple pattern) là một bộ ba RDF nhưng mỗi thành phần (subject, predicate hay object) đều có thể là một biến truy vấn.
- Một mẫu đồ thị cơ sở (basic graph pattern) là một tập các mẫu bộ ba.
- Ngôn ngữ SPARQL dựa trên cơ sở so sánh các mẫu đồ thị (graph pattern). Mẫu đồ thị đơn giản nhất là các mẫu bộ ba.
- Một định danh URI được đặt trong cặp dấu  $\langle \rangle$ . Một giá trị literal được đặt trong cặp dấu nháy kép ("").
- Biến trong ngôn ngữ truy vấn có giá trị toàn cục. Biến được bắt đầu với kí tự ? hoặc kí tự \$. Trong một câu truy vấn, \$x và ?x đều là biến.
- SPARQL cung cấp một cơ chế viết tắt. Tiền tố (prefix) có thể được định nghĩa và một QName sẽ cung cấp một dạng viết làm cho URI có thể ngắn gọn.
- Một nút trắng có thể xuất hiện trong một mẫu truy vấn (query pattern). Nó giữ vai trò như một biến, mặc dù nó không được đề cập trong kết quả của câu truy vấn hay bất kỳ nơi nào ở trong mẫu đồ thị.

## 3. Giới thiệu về Jena

### 3.1. Jena là gì?

Jena là tên gọi tắt cả Apache Jena (<http://jena.apache.org/>) là một khung ứng dụng Java cho việc phát triển các ứng dụng Web ngữ nghĩa và Linked Data (Chương 7). Jena cung cấp các thư viện Java giúp cho các nhà phát triển lập trình xử lý RDF, RDFS, RDFa, OWL và SPARQL theo các khuyến nghị của W3C. Jena cũng bao gồm một cơ chế suy luận dựa trên luật để thực hiện các suy diễn trên các ontology viết bằng OWL và RDFS; cũng như cung cấp nhiều giải pháp lưu trữ các bộ ba RDF trong bộ nhớ hoặc trên các bộ lưu trữ bền vững, như đĩa cứng.

Jena với tên gọi đầu tiên là Jena Semantic Framework (<http://jena.sf.net/>) là sản phẩm được phát triển từ phòng nghiên cứu Web ngữ nghĩa của HP (Hewlett-Packard) tại Bristol, Vương quốc Anh từ năm 2000. Jena luôn là một dự án mã nguồn mở và được sử dụng rất rộng rãi trong các ứng dụng và minh họa Web ngữ nghĩa.



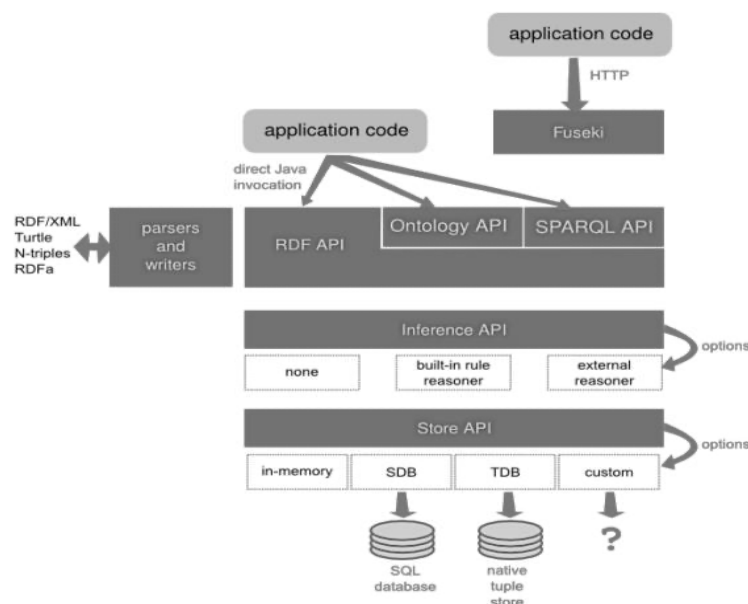
Vào năm 2009, HP quyết định tập trung các hoạt động phát triển khác hơn là tập hỗ trợ trực tiếp phát triển Jena, dù vẫn duy trì hỗ trợ các mục tiêu của dự án. Nhóm dự án Jena đã được chuyển đổi và hỗ trợ chính thức của Tổ chức phần mềm Apache (Apache Software Foundation) vào tháng 11 năm 2010. Từ đó Jena được ươm mầm phát triển tiếp tục với Apache và chính thức trở thành một dự án thuộc hàng đầu của Apache vào tháng 4 năm 2012.

Khung ứng dụng Jena bao gồm các thành phần sau:

- API sử dụng cho việc đọc, xử lý và ghi dữ liệu RDF theo các dạng XML, N-Triples và Turtle;
- Ontology API để xử lý các ontology dạng OWL và RDFS;
- Cơ chế suy luận dựa trên luật để suy diễn với các nguồn dữ liệu RDF và OWLsources;
- Lưu trữ một số lượng lớn các bộ ba RDF một cách hiệu quả lên đĩa hoặc trong các hệ quản trị CSDL phổ biến hiện nay;
- Một cơ chế truy vấn tương thích với các đặc tả SPARQL Update;
- Là máy chủ cho phép xuất bản dữ liệu RDF đến các ứng dụng khác với nhiều giao thức khác nhau, bao gồm cả SPARQL.

### 3.2. Kiến trúc khung ứng dụng Jena

Tại trung tâm của khung ứng dụng, Jena lưu các thông tin là các bộ ba RDF dưới dạng các đồ thị có hướng và cho phép mã chương trình bổ sung, xoá, sửa đổi, lưu trữ và xuất bản thông tin đó. Kiến trúc của Jena được thể hiện trong hình dưới



*Kiến trúc khung ứng dụng Jena*

Các bộ ba và đồ thị RDF và các thành phần khác được truy xuất thông qua giao diện lập trình RDF API của Jena. Các trừu tượng hóa đặc thù ở đây là Resource biểu diễn tài nguyên RDF (có định danh URI hoặc không), Literal cho các giá trị dữ liệu (số, chuỗi, ngày, ...), Statement biểu diễn một bộ ba RDF và Model biểu diễn toàn bộ đồ thị RDF. RDF API có các công cụ cơ bản để bổ sung và xóa các bộ ba trong đồ thị và tìm các bộ ba khớp với một mẫu xác định nào đó. Chúng ta cũng có thể đọc RDF từ các nguồn bên ngoài chẳng hạn từ các tập tin, địa chỉ URL, và mô tả một đồ thị theo các định dạng chuẩn. Cả giá trị đầu vào và đầu ra đều hỗ trợ tất cả các cú pháp phổ biến của RDF.

Jena cung cấp giao diện lập trình với Model khá phong phú, đồng thời cho phép đồ thị RDF lưu trữ dưới một tên đối tượng Graph đơn giản hơn thông qua Store API. Điều này cho phép Jena sử dụng nhiều các cách thức lưu trữ khác nhau một cách tương đương, miễn là nó hợp với giao diện Graph. Ngoài ra, Jena có thể lưu trữ một đồ thị trong bộ nhớ, hoặc trong một CSDL SQL, hoặc các hệ thống lưu trữ khác. Giao diện đồ thị cũng là một điểm mở rộng thuận lợi cho việc kết nối các hệ thống lưu trữ khác vào Jena, chẳng hạn như LDAP, bằng cách viết một ứng dụng nhỏ cho phép lời gọi từ Graph API làm việc trên hệ thống lưu trữ đó.

Đặc tính quan trọng của các ứng dụng Web ngữ nghĩa đó là các luật ngữ nghĩa của RDF, RDFS và OWL có thể được dùng để suy diễn thông tin từ một phát biểu không rõ ràng trong đồ thị. Ví dụ, nếu một lớp C là lớp con của lớp B, và B là lớp con của lớp A thì suy ra rằng C là lớp con của A. API suy diễn (inference API) cung cấp phương tiện để tạo ra các bộ ba được suy diễn từ những bộ ba đang lưu trữ như là các bộ ba được bổ sung bình thường khác. API suy diễn cung cấp một số bộ suy diễn luật để thực hiện điều này bằng cách sử dụng các tập luật có sẵn cho OWL và RDFS, hoặc các luật riêng của mỗi ứng dụng. Ngoài ra, API này có thể được kết nối với các bộ suy diễn ngoài chẳng hạn suy diễn logic mô tả để thực hiện công việc này với các giải thuật suy diễn khác và đặc thù.

Một tập các chuẩn định nghĩa các công nghệ Web ngữ nghĩa bao gồm SPARQL – ngôn ngữ truy vấn cho RDF (Chương 6). Jena tương thích tất cả các chuẩn đã được ban hành và luôn thể hiện các sự thay đổi và cập nhật vào các phiên bản phát triển liên tục. Để xử lý SPARQL, của truy vấn và cập nhật, là trách nhiệm của SPARQL API của Jena. Ontology cũng là quan trọng với nhiều ứng dụng Web ngữ nghĩa. Các

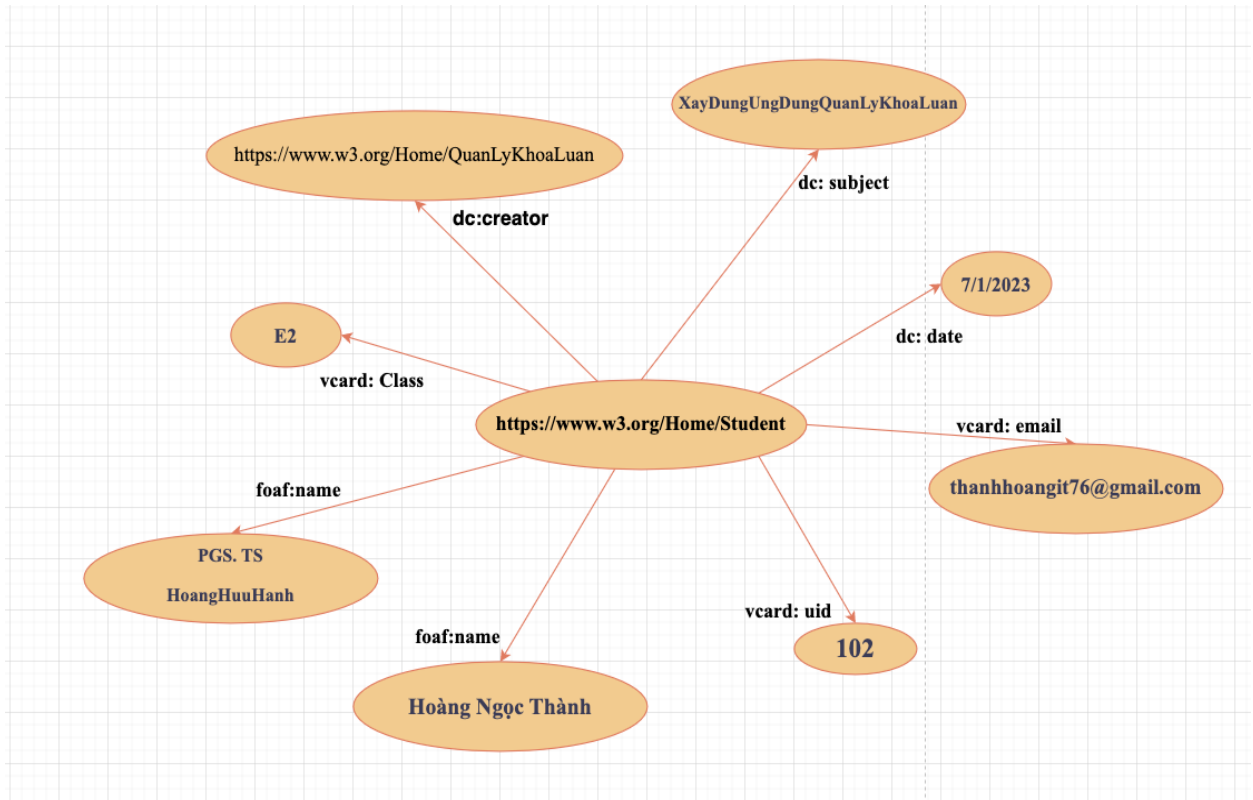
ontology là các mô tả logic hình thức, hoặc là các mô hình, của vài khía cạnh của thế giới thực mà các ứng dụng phải xử lý. Các ontology được chia sẻ giữa các nhà phát triển và các nhà nghiên cứu, tạo nên một nền tảng tốt cho việc xây dựng các ứng dụng linked-data. Ngôn ngữ ontology cho RDF gồm RDFS và OWL. OWL mạnh hơn về tính biểu diễn tri thức so với RDFS. Cả hai ngôn ngữ đều được Jena hỗ trợ thông qua Ontology API.

Trong khi các khả năng kể trên được thiết kế cho các ứng dụng thông qua các Java API, thì việc xuất bản dữ liệu qua Internet cũng là một yêu cầu chung trong các ứng dụng hiện đại. Fuseki, tên gọi trước đây là Joseki (<http://joseki.sf.net>), là một ứng dụng máy chủ cho việc xuất bản dữ liệu, có thể biểu diễn và cập nhật các mô hình RDF qua môi trường Web bằng cách sử dụng SPARQL và HTTP.

Ngoài ra, Jena còn có công cụ để thực hiện các lệnh thông qua giao diện dòng lệnh, các cơ chế lập chỉ mục đặc biệt cho viên tìm kiếm free-text, ... Thông tin chi tiết hơn có thể tìm ở website chính thức của Jena: <http://jena.apache.org>.

### III. Mô phỏng bài toán

#### 1. Mô tả dữ liệu



#### 2. Truy vấn SPARQL

##### 2.1. Hiển thị giáo viên hướng dẫn:

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name

WHERE { ?x foaf:name ?name }

name
PGS.TS HoangHuuHanhh

##### 2.2. Hiển thị ngày báo cáo khóa luận

PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?date

WHERE { ?x dc:date ? date }

date
7/1/2023

### 2.3. Hiện thị tên đề tài báo cáo khóa luận

PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?subject

WHERE { ?x dc:subject? subject }

subject
Xaydungungdungquanlykhoaluan

## 3. Lập trình Jena

### 3.1. Tạo mô hình RDF

```
1.  /*
2.  *@author Thanhhoang
3.  @date Jan 5, 2023
4.  */
5.  package com.husc.hoangngocthanh.semanticweb;
6.
7.  import org.apache.jena.rdf.model.Model;
8.  import org.apache.jena.rdf.model.ModelFactory;
9.  import org.apache.jena.rdf.model.Resource;
10. import org.apache.jena.sparql.vocabulary.FOAF;
11. import org.apache.jena.vocabulary.DC;
12. import org.apache.jena.vocabulary.SKOS;
13. import org.apache.jena.vocabulary.VCARD;
14.
15. public class TaoMoHinhRDF {
16.     public static Model taoModel() {
17.         // tạo một Model rỗng
18.         Model model = ModelFactory.createDefaultModel();
19.         // set các namespace
20.         model.setNsPrefix("foaf", FOAF.getURI());
21.         model.setNsPrefix("dc", DC.getURI());
22.         model.setNsPrefix("skos", SKOS.getURI());
23.         model.setNsPrefix("vcard", VCARD.getURI());
24.         // tạo tài nguyên
25.         Resource subject = model.createResource("https://www.w3.org/Home/QuanLyKhoaLuan");
26.         // Bổ sung thuộc tính
27.         subject.addProperty(DC.creator, "https://www.w3.org/Home/Student");
28.         Resource subject1 = model.createResource("https://www.w3.org/Home/Student");
29.         subject1.addProperty(FOAF.name, "HoangNgocTHanh");
30.         subject1.addProperty(FOAF.name, "PGS. Ts HoangHuuHanH");
31.         subject1.addProperty(DC.subject, "XayDungUngDungQuanLyKhoaLuan");
32.         subject1.addProperty(VCARD.UID, "102");
33.         subject1.addProperty(DC.date, "7/1/2023");
34.         subject1.addProperty(VCARD.CLASS, "E2");
35.         subject1.addProperty(VCARD.EMAIL, "thanhhoangit76@gmail.com");
36.         return model;
37.     }
38.
39.     public static void main(String[] args) {
40.         Model model = taoModel();
41.         model.write(System.out);
42.     }
43. }
```

Kết quả đạt được sau khi chạy chương trình ta có mô hình RDF:

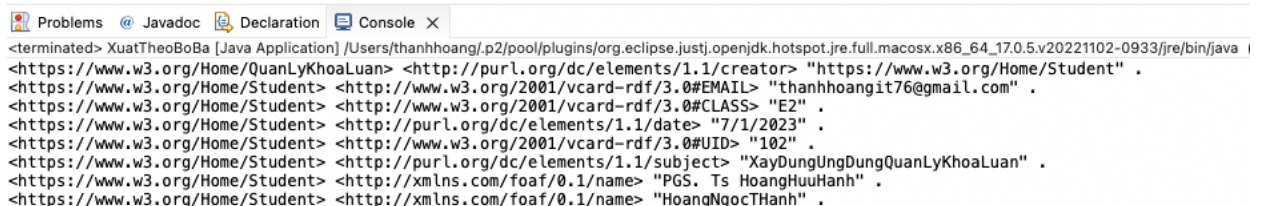
```
Problems @ Javadoc Declaration Console X
<terminated> TaoMoHinhRDF [Java Application] /Users/thanhhoang/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.5.v20221102-0933/jre/bin/java
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  <rdf:Description rdf:about="https://www.w3.org/Home/QuanLyKhoaLuan">
    <dc:creator>https://www.w3.org/Home/Student</dc:creator>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.w3.org/Home/Student">
    <vcard:EMAIL>thanhhoangit76@gmail.com</vcard:EMAIL>
    <vcard:CLASS>E2</vcard:CLASS>
    <dc:date>7/1/2023</dc:date>
    <vcard:UID>102</vcard:UID>
    <dc:subject>XayDungUngDungQuanLyKhoaLuan</dc:subject>
    <foaf:name>PGS. Ts HoangHuuHanh</foaf:name>
    <foaf:name>HoangNgocThanh</foaf:name>
  </rdf:Description>
</rdf:RDF>
```

```
1. <rdf:RDF
2.   xmlns:dc="http://purl.org/dc/elements/1.1/"
3.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.   xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
5.   xmlns:skos="http://www.w3.org/2004/02/skos/core#"
6.   xmlns:foaf="http://xmlns.com/foaf/0.1/"
7.   <rdf:Description rdf:about="https://www.w3.org/Home/QuanLyKhoaLuan">
8.     <dc:creator>https://www.w3.org/Home/Student</dc:creator>
9.   </rdf:Description>
10.  <rdf:Description rdf:about="https://www.w3.org/Home/Student">
11.    <vcard:EMAIL>thanhhoangit76@gmail.com</vcard:EMAIL>
12.    <vcard:CLASS>E2</vcard:CLASS>
13.    <dc:date>7/1/2023</dc:date>
14.    <vcard:UID>102</vcard:UID>
15.    <dc:subject>XayDungUngDungQuanLyKhoaLuan</dc:subject>
16.    <foaf:name>PGS. Ts HoangHuuHanh</foaf:name>
17.    <foaf:name>HoangNgocThanh</foaf:name>
18.  </rdf:Description>
19. </rdf:RDF>
```

## 3.2. Xuất dữ liệu theo mô hình bộ ba

```
1.  /*
2.  * @author Thanhhoang
3.  * @date Jan 5, 2022
4.  */
5.  package com.husc.hoangngocthanh.semanticweb;
6.
7.  import org.apache.jena.rdf.model.Model;
8.  import org.apache.jena.riot.Lang;
9.  import org.apache.jena.riot.RDFDataMgr;
10.
11. public class XuatTheoBoBa {
12.     public static void KetQua() {
13.         Model model = TaoMoHinhRDF.taoModel();
14.         RDFDataMgr.write(System.out, model, Lang.NTRIPLES);
15.     }
16.
17.     public static void main(String[] args) {
18.         KetQua();
19.     }
20. }
```

### Kết quả sau khi chạy chương trình



```
<terminated> XuatTheoBoBa [Java Application] /Users/thanhhoang/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.5.v20221102-0933/jre/bin/java
<https://www.w3.org/Home/QuanLyKhoaLuan> <http://purl.org/dc/elements/1.1/creator> "https://www.w3.org/Home/Student" .
<https://www.w3.org/Home/Student> <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL> "thanhhoangit76@gmail.com" .
<https://www.w3.org/Home/Student> <http://www.w3.org/2001/vcard-rdf/3.0#CLASS> "E2" .
<https://www.w3.org/Home/Student> <http://purl.org/dc/elements/1.1/date> "7/1/2023" .
<https://www.w3.org/Home/Student> <http://www.w3.org/2001/vcard-rdf/3.0#UID> "102" .
<https://www.w3.org/Home/Student> <http://purl.org/dc/elements/1.1/subject> "XayDungUngDungQuanLyKhoaLuan" .
<https://www.w3.org/Home/Student> <http://xmlns.com/foaf/0.1/name> "PGS. Ts HoangHuuHanH" .
<https://www.w3.org/Home/Student> <http://xmlns.com/foaf/0.1/name> "HoangNgocTHanh" .
```

1. <https://www.w3.org/Home/QuanLyKhoaLuan> <http://purl.org/dc/elements/1.1/creator> "https://www.w3.org/Home/Student" .
2. <https://www.w3.org/Home/Student> <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL> "thanhhoangit76@gmail.com" .
3. <https://www.w3.org/Home/Student> <http://www.w3.org/2001/vcard-rdf/3.0#CLASS> "E2" .
4. <https://www.w3.org/Home/Student> <http://purl.org/dc/elements/1.1/date> "7/1/2023" .
5. <https://www.w3.org/Home/Student> <http://www.w3.org/2001/vcard-rdf/3.0#UID> "102" .
6. <https://www.w3.org/Home/Student> <http://purl.org/dc/elements/1.1/subject> "XayDungUngDungQuanLyKhoaLuan" .
7. <https://www.w3.org/Home/Student> <http://xmlns.com/foaf/0.1/name> "PGS. Ts HoangHuuHanH" .
8. <https://www.w3.org/Home/Student> <http://xmlns.com/foaf/0.1/name> "HoangNgocTHanh" .

## **IV. Kết Luận**

### **1. Kết Quả Đạt Được**

➤ Từ những hướng dẫn tận tình của thầy, cùng những gì đã học được ở trường, em đã phân tích và xây dựng phần mềm với các kết quả đạt được:

- Bước đầu tìm hiểu được Web ngữ nghĩa là gì và áp dụng nó vào xây dựng ứng dụng
- Lập trình với jena tạo mô hình RDF và xuất dữ liệu theo bộ ba được
- Truy vấn SPARQL thành công
- Biểu diễn dữ liệu mô hình theo RDF

### **2. Hạn chế**

• Tuy nhiên, do thời gian và khả năng có hạn, nên em chưa đi sâu tìm hiểu được thêm về ứng dụng, vì vậy giao diện ứng dụng vẫn còn sơ sài. Rất mong các thầy cô giáo và các bạn tận tình giúp đỡ để ứng dụng áp dụng ngữ nghĩa vào để xây dựng ngày càng được hoàn thiện hơn.

### **3. Hướng phát triển**

- Tiếp tục nâng cao kiến thức để có thể áp dụng vào xây dựng ứng dụng



## **V. Tài liệu bài tiểu luận**

### ***1. Link github project***

***Github:*** [ThanhHoang76/SementicWeb](https://github.com/ThanhHoang76/SementicWeb)

### ***2. Tài liệu tham khảo***

- HoangHuuHanh, LeManhThanh (2012). GiaoTrinhWebNN\_v1.1

**PHIẾU ĐÁNH GIÁ TIỂU LUẬN**

**Học kỳ 1 Năm học 2022 - 2023**

<b>Cán bộ chấm thi 1</b>	<b>Cán bộ chấm thi 2</b>
<b>Nhận xét:</b> .....	<b>Nhận xét:</b> .....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
<b>Điểm đánh giá của CBChT1:</b>	<b>Điểm đánh giá của CBChT2:</b>
Bằng số: .....	Bằng số: .....
Bằng chữ: .....	Bằng chữ: .....

**Điểm kết luận:** Bằng số..... Bằng chữ:.....

**CBChT1**  
(Ký và ghi rõ họ tên)

*Thừa Thiên Huế, ngày ..... tháng ..... năm 20...*

**CBChT2**  
(Ký và ghi rõ họ tên)

