

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài

**XÂY DỰNG ỨNG DỤNG CHUẨN ĐOÁN BỆNH TÂM
DỰA TRÊN ONTOLOGY**

Sinh viên: Nguyễn Hoàng Tuấn

Mã số: B1606858

Khóa: K42

Cần Thơ, 01/2020

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài

**XÂY DỰNG ỨNG DỤNG CHUẨN ĐOÁN BỆNH TÂM
DỰA TRÊN ONTOLOGY**

Người hướng dẫn
TS. Trần Công Án

Sinh viên thực hiện
Nguyễn Hoàng Tuấn
Mã số: B1606858
Khóa: K42

Cần Thơ, 01/2020

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành sâu sắc đến thầy Trần Công Ân đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình hoàn hành luận văn.

Xin cảm ơn quý thầy cô khoa CNTT và TT đã truyền đạt những kiến thức giúp em có những kiến thức nền tảng để có thể hoàn thành luận văn.

Xin cảm ơn Ban Giám Hiệu, tất cả thầy cô và nhân viên thuộc trường Đại học Cần Thơ đã tạo điều kiện cho em hoàn thành chương trình đào tạo.

Cảm ơn gia đình và bạn bè đã luôn luôn bên cạnh ủng hộ, động viên, giúp đỡ em trong những lúc khó khăn nhất.

Em xin chân thành cảm ơn.

Cần Thơ, ngày 01 tháng 01 năm 2021

Tác giả luận văn

Nguyễn Hoàng Tuấn

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

ii

[illegible]

iii

[illegible]

iv

MỤC LỤC

DANH MỤC BẢNG.....	vii
DANH MỤC HÌNH.....	viii
DANH MỤC TỪ VIẾT TẮT	ix
TÓM TẮT	x
ABSTRACT	xi
PHẦN 1: GIỚI THIỆU.....	1
I. Đặt vấn đề	1
II. Những nghiên cứu liên quan.....	1
III. Mục tiêu	2
IV. Đối tượng và phạm vi nghiên cứu.....	3
1. Đối tượng nghiên cứu	3
2. Phạm vi nghiên cứu.....	3
V. Phương pháp nghiên cứu	3
VI. Nội dung nghiên cứu	3
VII. Bố cục quyền luận văn.....	4
PHẦN 2: NỘI DUNG.....	5
CHƯƠNG 1: ĐẶC TẢ YÊU CẦU	5
I. Yêu cầu chức năng	5
1. Chức năng chẩn đoán bệnh	5
2. Chức năng tra cứu.....	5
II. Yêu cầu phi chức năng.....	5
CHƯƠNG 2: THIẾT KẾ GIẢI PHÁP.....	6
I. Cơ sở lý thuyết	6
1. Ontology.....	6
2. RDF.....	6
3. Jena Fuseki.....	7
4. SPARQL	7
5. Xampp.....	8
6. Apache	8
7. Ngôn ngữ PHP	8
8. Ngôn ngữ HTML.....	8
9. Ngôn ngữ CSS	9

10.	Bootstrap	10
11.	Ngôn ngữ JavaScript	10
12.	JQUERY	10
II.	Mô hình dữ liệu	11
1.	Lớp - Class	11
2.	Liên kết – Object	12
3.	Thuộc tính – Property	12
4.	Axixom	12
5.	Tập luật – SWRL	13
6.	Các phương thức	13
CHƯƠNG 3: CÀI ĐẶT GIẢI PHÁP		15
I.	Server Jena Fuseki	15
II.	Thư viện SparqlLib	17
III.	Chuẩn đoán bệnh	18
1.	Lưu đồ dữ liệu	18
2.	Nguyên lý hoạt động	19
IV.	Tra cứu	20
1.	Tra cứu bệnh	20
2.	Tra cứu triệu chứng	21
3.	Tra cứu thuốc	22
CHƯƠNG 4: ĐÁNH GIÁ KIỂM THỬ		23
I.	Mục tiêu kiểm thử	23
II.	Kịch bản kiểm thử	23
III.	Kết quả kiểm thử	24
PHẦN 3: KẾT LUẬN		29
I.	Kết quả đạt được	29
1.	Kết quả	29
2.	Hạn chế	29
II.	Hướng phát triển	29
TÀI LIỆU THAM KHẢO		30

DANH MỤC BẢNG

Bảng 1. Một số thẻ trong HTML	8
Bảng 2. Danh sách các lớp	11
Bảng 3. Danh sách các liên kết	12
Bảng 4. Danh sách các thuộc tính.....	12
Bảng 5. Thuộc tính của các quan hệ	12
Bảng 6 . Danh sách các luật	13
Bảng 7. Các phương thức.....	13
Bảng 8. Một số hàm / phương thức trong thư viện sparqllib	17
Bảng 9. Kịch bản kiểm thử	23
Bảng 10. Kết quả kiểm thử kịch bản	24

DANH MỤC HÌNH

Hình 1. Giao diện chọn triệu chứng của Ứng dụng ESforRPD2	1
Hình 2. Giao diện kết quả của Ứng dụng ESforRPD2	2
Hình 3. Giao diện chi tiết loại bệnh của Ứng dụng ESforRPD2	2
Hình 4. Sơ đồ chức năng.....	5
Hình 5. Mô hình cơ sở dữ liệu	11
Hình 6. Cách lấy dữ liệu axixom	13
Hình 7. Sơ đồ nguyên lý hoạt động của server Jena Fuseki	15
Hình 8. Tạo dataset trên server Jena Fuseki	15
Hình 9. Tải file ontology lên server	16
Hình 10. Truy vấn câu lệnh sparql trên Jena Fuseki	16
Hình 11. Import thư viện sparqllib	17
Hình 12. Lấy kết quả từ CSDL	17
Hình 13. Lưu đồ dữ liệu chẩn đoán bệnh	18
Hình 14. Phương thức POST	18
Hình 15. Danh sách các triệu chứng	19
Hình 16. Kết quả chuẩn đoán sơ bộ.....	19
Hình 17. Kết quả chẩn đoán nâng cao	19
Hình 18. Lưu đồ tra cứu bệnh.....	20
Hình 19. Danh sách các bệnh.....	20
Hình 20. Lưu đồ dữ liệu tra cứu triệu chứng.....	21
Hình 21. Danh sách triệu chứng	21
Hình 22. Lưu đồ dữ liệu tra cứu thuốc	22
Hình 23. Danh sách các loại thuốc	22
Hình 24. Giao diện chuẩn đoán bệnh.....	24
Hình 25. Giao diện kết quả chuẩn đoán sơ bộ.....	25
Hình 26. Giao diện kết quả chẩn đoán nâng cao	25
Hình 27. Giao diện tra cứu bệnh.....	26
Hình 28. Giao diện thông tin chi tiết bệnh	26
Hình 29. Giao diện tra cứu triệu chứng	27
Hình 30. Giao diện thông tin chi tiết triệu chứng.....	27
Hình 31. Giao diện tra cứu thuốc.....	28
Hình 32. Giao diện thông tin chi tiết thuốc	28

DANH MỤC TỪ VIẾT TẮT

Ký hiệu viết tắt	Giải thích
RDF	Resource Description Framework
RDFS	Resource Description Framework Shema
CSDL	Cơ sở dữ liệu
OWL	Web Ontology Language
SWRL	Semantic Web Rule Language
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
CSS	Cascading Style Sheets
JS	JavaScript
XML	Extensible Markup Language

TÓM TẮT

Hiện nay, Việt Nam là một trong những quốc gia xuất khẩu tôm lớn nhất trên thế giới, đem lại nguồn ngoại tệ ngày càng lớn cho đất nước. Với thời đại công nghệ số hiện nay, việc nuôi tôm cũng nên được hỗ trợ bởi công nghệ, điển hình như một phần mềm tìm ra bệnh tôm để dễ dàng tìm ra thuốc chữa trị cũng như phương pháp xử lý hiệu quả nhất. Đề tài luận văn mong muốn thiết kế một trang web để dàng sử dụng giúp chuẩn đoán bệnh tôm dựa trên một số triệu chứng tôm mắc phải. Ứng dụng giúp người nuôi tôm đắc lực trong việc tìm ra bệnh tôm cũng như cách phòng chống, đặc biệt là những người mới bắt đầu nuôi tôm chưa có kinh nghiệm hay những triệu chứng không xảy ra phổ biến.

Với ontology, máy tính có thể tổng hợp cùng lúc nhiều nguồn dữ liệu cùng lúc cũng như có thể hiểu được trực tiếp nguồn dữ liệu do có cấu trúc ngữ nghĩa từ đó thông tin sẽ được tìm kiếm nhanh chóng và chính xác hơn. Đặc biệt mô hình phân cấp của nó rất phù hợp cho việc chẩn đoán. Ứng dụng chẩn đoán sử dụng cơ sở dữ liệu là ontology với ngôn ngữ chính là PHP chạy trên máy chủ Apache, được xây dựng trên công cụ Visual Studio Code. Các công nghệ khác được sử dụng như HTML, CSS, JavaScript, Bootstrap, SPARQL, jQuery.

Từ khóa: ontology, SPARQL, PHP.

ABSTRACT

Currently, Vietnam is one of the largest shrimp exporting countries in the world, bringing more and more foreign currency to the country. With today's digital technology, shrimp farming should also be supported by technology, typically a software to find shrimp diseases to easily find the cure and the most effective treatment method. The thesis aims to design an easy-to-use website to help diagnose shrimp diseases based on some symptoms of shrimp acquired. The application helps shrimp farmers effectively in finding out shrimp diseases as well as how to prevent them, especially those who have just started farming inexperienced shrimp or the symptoms do not occur commonly.

With ontology, the computer can synthesize many data sources at the same time as well as directly understand the data source due to the semantic structure from which information will be searched more quickly and accurately. Especially its hierarchical model is very suitable for diagnostics. The diagnostic application uses the database ontology with the main language PHP running on the Apache server, built on the Visual Studio Code engine. Other technologies used are HTML, CSS, JavaScript, Bootstrap, SPARQL, jQuery.

Keyword: ontology, SPARQL, PHP.

PHẦN 1: GIỚI THIỆU

I. Đặt vấn đề

Đường bờ biển nước ta dài 3260 km kéo dài qua 28 tỉnh thành từ Bắc vào Nam. Đây là điều kiện rất thuận lợi để phát triển ngành đánh bắt nuôi trồng thủy hải sản. Trong đó, ngành nuôi tôm đang được đầu tư và có những phát triển vượt bậc, mức tăng trưởng tăng đều qua từng năm, trở thành một trong những quốc gia xuất khẩu tôm lớn nhất thế giới, đem lại nguồn ngoại tệ khá lớn cho quốc gia. Tuy nhiên, việc chăm sóc tôm thật sự không dễ dàng, đặc biệt đối với những người mới bắt đầu chưa có nhiều kinh nghiệm, cụ thể là không biết được tôm bị bệnh gì. Ngay cả những người nuôi tôm có khi nghiệm khi gặp phải những bệnh hiếm gặp cũng sẽ khá lúng túng trong việc xử lý. Do đó, nhu cầu chẩn đoán bệnh tôm là cần thiết.

Bên cạnh đó, công nghệ thông tin dường như phủ sóng khắp mọi nơi, khắp mọi ngành nghề và nó có thể được ứng dụng vào để hỗ trợ cho việc chăm sóc tôm tốt hơn nhằm tăng năng suất, đạt hiệu quả cao hơn. Vừa hay, cơ sở dữ liệu phân cấp *ontology* rất phù hợp cho việc chẩn đoán tìm ra bệnh tôm dựa trên tập triệu chứng, cũng như tra cứu các loại bệnh và các loại thuốc đặc trị tương ứng. Do đó, ý tưởng xây dựng ứng dụng chẩn đoán bệnh dựa trên ontology đã được hình thành.

II. Những nghiên cứu liên quan

Hệ thống chẩn đoán bệnh cho cây lúa ESforRPD2

Hệ thống EsforRPD2 giúp chẩn đoán bệnh ở lúa dựa trên những đặc điểm hình thái, với dữ liệu đầu vào là danh sách các triệu chứng sẽ cho ra danh sách các bệnh.

<input type="checkbox"/>	38	Vết ướt ở mép lá hoặc phần bị thương của lá.
<input type="checkbox"/>	39	Lá khô héo trở nên thối rữa
<input type="checkbox"/>	40	Lá héo như bông
<input type="checkbox"/>	41	Lá trở nên nhăn nheo
<input type="checkbox"/>	42	Triệu chứng héo ở cây non
<input type="checkbox"/>	43	Vô trùng số lượng lớn (vô trùng)
<input type="checkbox"/>	44	Lá úa vàng đến cam từ đầu lá về phía gốc
<input type="checkbox"/>	45	Số lượng máy xới giảm
<input type="checkbox"/>	46	Có đốm nâu sẫm trên hạt
<input type="checkbox"/>	47	Các đốm nâu được nhìn thấy từ vết thủng của côn trùng truyền nhiễm trên lá già
<input type="checkbox"/>	48	Trông giống như một chiếc lá non

Trước 1 Kế tiếp

Hình 1. Giao diện chọn triệu chứng của Ứng dụng ESforRPD2

Đầu tiên, người dùng chọn những triệu chứng lúa đang mắc phải và gửi chẩn đoán.

Kết quả chẩn đoán

Các triệu chứng đã chọn	
Không	Tên triệu chứng
1	Hạt gạo trở thành quả cầu
2	Trong một bông chỉ có một lượng nhỏ cùi bị nhiễm bệnh

Hiển thị các tính toán

Dựa trên các triệu chứng đã chọn, chẩn đoán chính xác nhất là:

Bệnh đạo ôn / đạo ôn (Pyricularia oryzae), đốm nâu / đốm nâu. (Helminthosporium oryzae / Drechslera oryzae), Bệnh đốm nâu hẹp (Cercospora oryzae), Bệnh cháy lá / Bệnh khô vằn / Bệnh thối thân / Bệnh khô vằn và bệnh thối thân. (Rhizoctonia solani Kuhn), Smudge giả / Smut giả. (Ustilagoidea virens), Cò lùn / Cò lùn. (Nilaparvata Lugens Stal), Kresiek / Bệnh cháy lá / bệnh cháy lá do vi khuẩn, BLB. (Xanthomonas campestris pv. Oryzae), Tungro

Với mức độ tin cậy là 72% .

Phương pháp điều khiển

Hình 2. Giao diện kết quả của Ứng dụng ESforRPD2

Kết quả sẽ cho ra danh sách các bệnh và tính toán mức độ tin cậy toàn thể.

Chẩn đoán

Đạo ôn / Đạo ôn. (Pyricularia oryzae)

Thông tin

Bệnh đạo ôn được biết đến ở tất cả các nước trồng lúa và được coi là bệnh quan trọng nhất. Năm 1913, bệnh này cùng với đốm nâu xuất hiện ở nhiều vườn ươm ở vùng Surabaya và Madura, tuy không gây thiệt hại lớn. Bệnh đạo ôn với các triệu chứng tương tự như cây bị thối khí nóng. Bệnh này khác với bệnh đốm nâu (Drechslera oryzae), bệnh đạo ôn xuất hiện nhiều hơn trên các loại cây màu, do đó bệnh này thường được coi là bệnh của nhà giàu, với việc thâm canh nông nghiệp ở Indonesia ngày càng tăng thì thiệt hại do bệnh đạo ôn cũng ngày càng nhiều (Semangun, Năm 1990).

Điều khiển

1. Luân canh với cây trồng không phải là lúa. 2. Trồng các giống kháng như Sentani, Danau Bawah, Tondano, Singkarak, Arias, Ranau, Maninjau, Danau Atas ở vùng cao, trong khi các giống Dodokan và Jangkak kháng nhẹ. Các giống Asahan và PM 38 kháng bệnh đạo ôn trên ruộng lúa. 3. Phương pháp trồng cải tiến: - Ngâm rom bệnh làm phân trộn, - Khoảng cách cây, - Bón phân cân đối (N, P, K, S, và phân vi lượng) - Cải thiện hệ thống cấp nước, - Làm cỏ 4. Chăm sóc hạt giống thuốc diệt nấm khuyến cáo 5. Phun cây bằng thuốc diệt nấm khuyến cáo. Đối với những vùng thường bị bệnh (vùng đặc hữu), thời gian bón là thời điểm để nhánh cực đại, giai đoạn mang thai và trở sớm (ra hoa 5-10%). Trong khi đó, đối với những diện tích không phải đặc hữu, việc bón chỉ được tiến hành khi bắt đầu ra hoa nếu những lần quan sát trước đó có đốm lá. 6. Sử dụng tác nhân sinh học Paenibacillus polymixa. (Tổng cục Bảo vệ cây trồng lương thực, 1989)

Hình 3. Giao diện chi tiết loại bệnh của Ứng dụng ESforRPD2

Sau khi tìm ra loại bệnh, người dùng có thể xem chi tiết danh sách loại bệnh.

III. Mục tiêu

Đề tài mong muốn sẽ tổng hợp các loại bệnh, các triệu chứng ở tôm và hệ thống chúng đưa vào phần mềm chẩn đoán giúp tìm ra loại bệnh tôm đang mắc phải cũng như cách phòng chống dựa trên các triệu chứng mắc phải. Đề tài còn mong muốn xây dựng chức năng tra cứu giúp người dùng tìm thấy thông tin của danh sách các bệnh, các triệu chứng và thuốc.

IV. Đối tượng và phạm vi nghiên cứu

1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm nguồn dữ liệu về bệnh tôm (các loại bệnh tôm, triệu chứng, thuốc) và cơ sở dữ liệu “Ontology” bao gồm các thành phần liên quan:

- Ontology: cơ sở dữ liệu phân cấp
- SPARQL: ngôn ngữ truy vấn dữ liệu trong ontology
- sparqllib: thư viện giúp kết nối dữ liệu và thực hiện các truy vấn SPARQL được viết bằng ngôn ngữ PHP

Ngoài ra, còn có các công nghệ về thiết kế web.

2. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài chủ yếu là làm việc với:

- Phần mềm Protege để thiết kế cơ sở dữ liệu
- Phần mềm Jena Fuseki để tạo endpoint giúp kết nối với trang web đồng thời chuyển mô hình dữ liệu về dạng RDF
- Phần mềm Visual Studio Code để thiết kế trang web

Ngoài ra, còn có các thư viện hỗ trợ như:

- Thư viện Bootstrap: hỗ trợ thiết kế web
- Thư viện jQuery: tạo các sự kiện, tương tác giữa các phần tử web
- Thư viện sparqllib: kết nối và truy vấn dữ liệu RDF

V. Phương pháp nghiên cứu

Do nội dung đề tài khá mới mẻ nên phương pháp nghiên cứu chủ yếu dựa vào các tài liệu online trên mạng như trang chủ của các thư viện, youtube, các trang chia sẻ kiến thức. Đối với mỗi thành phần của đối tượng nghiên cứu sẽ có phương pháp:

- Ontology: tìm hiểu các nội dung liên quan đến ở trang chủ W3C, đọc tài liệu online tìm kiếm trên google
- SPARQL: đọc tài liệu trên các trang chia sẻ kiến thức
- Thư viện sparqllib: tải thư viện về và đọc, nghiên cứu các hàm, phương thức bên trong
- Dữ liệu bệnh tôm: đọc các tài liệu chẩn đoán bệnh ở tôm

VI. Nội dung nghiên cứu

Nội dung nghiên cứu bao gồm:

- Tìm hiểu về các loại bệnh ở tôm cũng như các triệu chứng liên quan, tác nhân gây bệnh, cách phòng chống và thuốc đặc trị
- Tìm hiểu về ontology, cách sử dụng phần mềm thiết kế ontology (Protege), mô hình dữ liệu RDF và ngôn ngữ truy vấn dữ liệu RDF (SPARQL)
- Tìm hiểu các công nghệ hỗ trợ cho việc thiết kế website như Bootstrap, jQuery.
- Tìm hiểu, tham khảo các đề tài liên quan để tìm ra giải pháp tốt nhất cho phần mềm.

VII. Bố cục quyền luận văn

Bố cục luận văn bao gồm các phần và các chương sau:

Phần 1: Giới thiệu:

- Đặt vấn đề
- Những nghiên cứu liên quan
- Mục tiêu
- Đối tượng và phạm vi nghiên cứu
- Phương pháp nghiên cứu
- Nội dung nghiên cứu
- Bố cục quyền luận văn

Phần 2: Nội dung:

Chương 1: Đặc tả yêu cầu

- Yêu cầu chức năng
- Yêu cầu phi chức năng

Chương 2: Thiết kế giải pháp

- Cơ sở lý thuyết
- Mô hình dữ liệu

Chương 3: Cài đặt giải pháp

- Server Jena Fuseki
- Thư viện SparqlLib
- Chuẩn đoán bệnh
- Tra cứu

Chương 4: Đánh giá kiểm thử

- Mục tiêu kiểm thử
- Kịch bản kiểm thử
- Kết quả kiểm thử

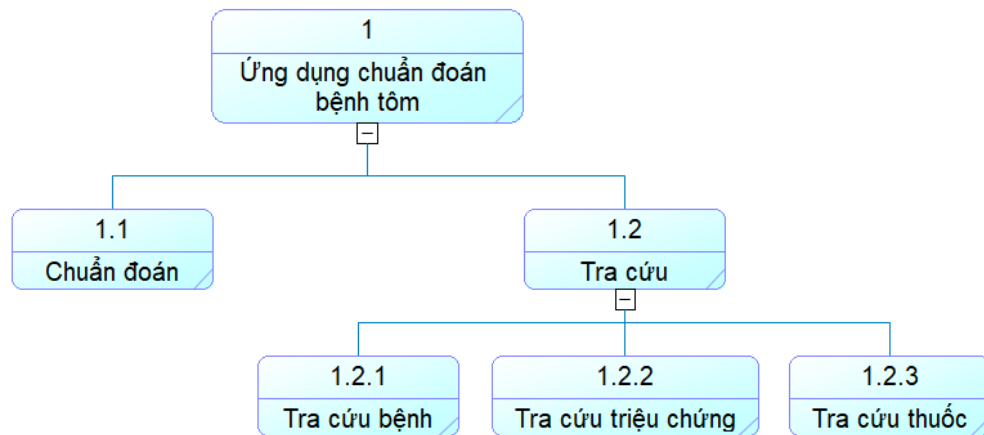
Phần 3: Kết luận:

- Kết luận
- Hướng phát triển

PHẦN 2: NỘI DUNG

CHƯƠNG 1: ĐẶC TẢ YÊU CẦU

I. Yêu cầu chức năng



Hình 4. Sơ đồ chức năng

Sơ đồ chức năng Ứng dụng chuẩn đoán bệnh tâm gồm có 2 chức năng chính là chuẩn đoán và tra cứu. Chức năng chuẩn đoán gồm có chuẩn đoán sơ bộ và chuẩn đoán nâng cao. Chức năng tra cứu gồm có tra cứu bệnh, tra cứu thuốc và tra cứu triệu chứng.

1. Chức năng chẩn đoán bệnh

Chức năng chuẩn đoán bệnh được chia làm 2 giai đoạn:

- Tìm những bệnh có một trong các triệu chứng đã chọn
- Từ danh sách bệnh đó, tìm các triệu chứng liên quan khác và tiến hành chọn tiếp trong các triệu chứng đó để giới hạn lại danh sách bệnh tìm được. Giai đoạn 2 có thể lặp lại đến khi nào không còn triệu chứng liên quan nữa

2. Chức năng tra cứu

Chức năng tra cứu gồm có 3 phần:

- Tra cứu bệnh: từ danh sách các bệnh có thể xem chi tiết bệnh đó, các triệu chứng của bệnh, các thuốc đặc trị cũng như thông tin về tác nhân gây bệnh và cách phòng chống
- Tra cứu triệu chứng: từ danh sách triệu chứng có thể tra ra mắc triệu chứng đó sẽ có thể bị những bệnh nào
- Tra cứu thuốc: từ danh sách thuốc, chọn một loại thuốc sau đó tìm ra các bệnh mà thuốc đặc trị

II. Yêu cầu phi chức năng

Các yêu cầu phi chức năng gồm:

- Giao diện đơn giản, thân thiện, dễ dùng.
- Ứng dụng chỉ hỗ trợ chạy tốt nhất trên màn hình desktop.

CHƯƠNG 2: THIẾT KẾ GIẢI PHÁP

I. Cơ sở lý thuyết

1. Ontology

Ontology là một mô hình dữ liệu biểu diễn một lĩnh vực nào đó và được sử dụng để suy luận về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng.

Ontology cung cấp bộ từ vựng bao gồm các khái niệm, các thuộc tính và các định nghĩa.

❖ Các phần tử trong ontology:

Các lớp (Class)

Các lớp là các nhóm, tập hợp các đối tượng gọi là thể hiện. Chúng có thể chứa các thể hiện, chứa cả lớp khác hay cả hai cùng lúc đều được. Một lớp có thể xếp gộp vào một lớp khác để trở thành lớp con của lớp đó. Mối quan hệ giữa các lớp xếp gộp tạo ra một cấu trúc có thứ bậc. Thường trong một ontology, tất cả các lớp được đặt trong một lớp gốc gọi là Thing.

Các thể hiện (Individuals)

Các thể hiện là các đối tượng, là thành phần cơ bản của một ontology. Các thể hiện là những đối tượng như con người, động vật, đồ vật, địa điểm,... Thậm chí là các đối tượng trừu tượng như thành viên, các từ,...

Các thuộc tính (Properties)

Các thuộc tính mô tả các đặc trưng, đặc điểm, tính chất của các thể hiện trong ontology. Mỗi thuộc tính có tên và đều có một giá trị đối với mỗi thể hiện. Các thuộc tính dùng để lưu trữ các thông tin mà mỗi thể hiện có thể có.

Các quan hệ (Relation)

Các quan hệ là mối liên kết giữa các thể hiện trong ontology. Một quan hệ cũng như một thuộc tính nhưng có giá trị là một thể hiện khác trong ontology.

2. RDF

RDF (Khung mô tả tài nguyên) là viết tắt của Resource Description Framework, là một mô hình dữ liệu. Tài nguyên trong mô hình này được biểu diễn bằng các thẻ tương tự giống như HTML. Khối xây cơ bản của nó là bộ ba đối tượng-thuộc tính-giá trị và được gọi là một câu lệnh. RDF được cung cấp một cú pháp trong XML và kế thừa những lợi ích liên quan đến XML. Tuy nhiên, các biểu diễn cú pháp quan trọng khác không dựa trên XML nên cú pháp dựa trên XML không phải là thành phần nhất thiết phải có trong mô hình RDF.

RDF độc lập với miền, người dùng có thể tùy ý xác định một thuật ngữ riêng của mình trong ngôn lược đồ được gọi là RDF Schema (RDFS). Trong RDFS, chúng ta có thể tự xác định từ vựng, chỉ định thuộc tính nào áp dụng cho loại đối tượng nào và quyết định chúng có thể nhận những giá trị nào.

3. Jena Fuseki

Jena Fuseki là một khung dùng để xây dựng các ứng dụng web ngữ nghĩa, bao gồm API để đọc, xử lý và ghi dữ liệu RDF. Nó đóng vai trò là máy chủ giúp tạo endpoint từ file ontology đã tạo bằng phần mềm Protégé để kết nối với ứng dụng.

Jena còn cung cấp cho các nhà phát triển một tập thư viện để xử lý RDF, RDFS, OWL và SPARQL.

4. SPARQL

SPARQL là một ngôn ngữ truy vấn RDF, gồm 4 kiểu truy vấn khác nhau:

- Select: trích xuất các giá trị thô từ SPARQL endpoint, các kết quả được trả về trong một định dạng bảng
- Construct: trích xuất thông tin từ SPARQL endpoint và chuyển kết quả thành dạng RDF hợp lệ
- Ask: cung cấp các kết quả dạng True/False đơn giản cho các truy vấn trên SPARQL endpoint
- Describe: trích xuất một đồ thị RDF từ SPARQL endpoint, các nội dung đó được đưa tới endpoint để quyết định dựa trên những thông tin có ích

Trong câu truy vấn SPARQL, biến được chỉ định bởi tiền tố “?” hoặc “\$”. Câu truy vấn SPARQL cũng dựa trên bộ ba như trong RDF và được đặt trong mệnh đề WHERE, các thành phần cần truy xuất được đặt trong mệnh đề SELECT và được thay thế bằng biến.

Một câu truy vấn SPARQL hoàn chỉnh gồm 2 mệnh đề chính, mệnh đề SELECT và mệnh đề WHERE cùng các thành phần phụ như: PREFIX, FROM, FILTER, OPTIONAL, ORDER BY, GROUP BY, DISTINCT,... Cú pháp tổng quát của SPARQL được liệt kê ví dụ như sau:

```
PREFIX name: <namespaceURI>
SELECT ?a ?b
[FROM <dataURI>]
[FROM NAMED <dataURI>]
WHERE {
    ?a name: ?b .
    [FILTER] [OPTIONAL]
}
[ORDER BY]
[GROUP BY]
[OFFSET/LIMIT]
[DISTINCT]
```

5. Xampp

XAMPP là viết tắt của Cross-Platform (đa nền tảng-X), Apache (A), MariaDB (M), PHP (P) và Perl (P), là phần mềm mã nguồn mở rất phổ biến dùng để xây dựng các dự án website theo ngôn ngữ PHP được chạy trên localhost của máy.

XAMPP là bộ phần mềm tuyệt vời, có giao diện quản lý khá dễ sử dụng, tích hợp nhiều chức năng vô cùng tiện ích. Chỉ cần cài XAMPP, người dùng có thể sử dụng bộ ba tiện ích thiết kế web quyền lực:

- Apache: Server dùng để xử lý các file php
- PHP: ngôn ngữ viết cho server
- MySql: cơ sở dữ liệu thông dụng dùng trong thiết kế web

6. Apache

Apache là phần mềm web server miễn phí mã nguồn mở giúp đưa nội dung lên web. Apache như một người giao hàng, khi có người gửi yêu cầu đến server Apache sẽ gửi file tương ứng với yêu cầu xuống client.

Apache là một phần mềm đa nền tảng, xử lý tốt các file dưới dạng ngôn ngữ PHP, Python, Java,... Apache biến những ngôn ngữ thành một phần của HTML và có thể hiển thị trên trình duyệt cho người dùng thấy được.

7. Ngôn ngữ PHP

PHP là viết tắt của Hypertext Preprocessor, là một ngôn ngữ lập trình dùng để phát triển các ứng dụng viết cho máy chủ. Nó thường được dùng để làm web do có thể dễ dàng nhúng và trang HTML, tốc độ nhanh, nhỏ gọn, dễ lập trình, cú pháp khá dễ dùng và đạt hiệu quả cao.

Ngoài ra, PHP còn có rất nhiều các thư viện hỗ trợ đặc lực cho việc lập trình web.

Ưu điểm của PHP

- Cú pháp dễ học, dễ tiếp cận, có nhiều nguồn tài liệu tham khảo
- Tương thích với hầu hết mọi trình duyệt
- Có thể viết code trên bất kỳ trình soạn thảo văn bản nào
- Có tốc độ xử lý rất nhanh
- Tính cộng đồng cao, nhiều thư viện hỗ trợ

8. Ngôn ngữ HTML

HTML là viết tắt của Hypertext Markup Language, không phải là ngôn ngữ lập trình, chỉ bao gồm các thẻ giúp trình bày các nội dung văn bản.

Bảng 1. Một số thẻ trong HTML

Tên thẻ	Mô tả
<html>	Bao bọc toàn bộ code của trang web
<head>	Chứa các thẻ định dạng tiêu đề, icon, thẻ nhúng các thư viện

<body>	Chứa nội dung trang web
<meta>	Định dạng bảng mã sử dụng
<title>	Gán tiêu đề cho trang web
<link>	Thẻ giúp chèn vào các file javascript hay css
<div>	Dùng nhóm các phần tử lại với nhau
<style>	Chứa các định dạng css
<script>	Chứa nội dung javascript
<a>	Định nghĩa là một đường dẫn
	Hiển thị hình ảnh dựa vào thuộc tính src
<p>	Chứa văn bản
	Danh sách theo thứ tự
	Danh sách không theo thứ tự
	Phần tử trong danh sách
<table>	Định dạng bảng
<input>	Biểu diễn trường có thể nhập liệu vào, có nhiều kiểu để lựa chọn như: text, checkbox, radio, password,...
<button>	Tạo nút có thể click đặt trong form

Tất cả nội dung trang web thường nằm trong thẻ <html></html>. Cấu trúc tổng thể của một trang web gồm thẻ <html></html> chứa 2 thẻ chính bên trong gồm thẻ <head></head> và thẻ <body></body>. Cấu trúc được liệt kê như sau:

```
<html>
<head>
<!-- META INFORMATION -->
</head>
<body>
<!-- PAGE CONTENT -->
</body>
</html>
```

9. Ngôn ngữ CSS

CSS là viết tắt của Cascading Style Sheets, là ngôn ngữ giúp thay đổi định dạng nội dung của các phần tử HTML như là font chữ, màu sắc, vị trí, cách hiển thị,... Trang web có đẹp mắt, thân thiện người dùng hay không phần lớn là nhờ vào CSS.

CSS có thể được viết trực tiếp trong HTML bằng cách đặt trong các thẻ của HTML với thuộc tính style hoặc đặt trong thẻ <style>. Ngoài ra, CSS cũng có thể viết thành file riêng biệt và nhúng vào HTML bằng thẻ <link>.

CSS có thể sử dụng một cách dễ dàng, có thể được viết trên bất kỳ trình soạn thảo văn bản nào, có cú pháp rất dễ tiếp cận và dễ học.

10. Bootstrap

Bootstrap là một framework bao gồm các HTML, CSS, JavaScript template dùng để thiết kế giao diện web nhanh hơn, chuyên nghiệp hơn. Bootstrap được dùng chủ yếu dựa vào định danh của các thẻ qua *class* hoặc *id*.

Bootstrap được sử dụng rất dễ dàng với tính tùy chỉnh cao, người lập trình có thể tùy ý chỉnh sửa các định dạng sẵn có của bootstrap một cách nhanh chóng và đơn giản.

Tóm lại, bootstrap có những lợi ích sau:

- Phát triển website trở nên nhanh chóng
- Sự tương thích với trình duyệt và thiết bị
- Tương tác tốt với smartphone
- Giao diện đầy đủ, sang trọng
- Dễ dàng tùy biến

Tuy nhiên, cũng có một số hạn chế như:

- Có nhiều code thừa
- Làm giảm tính sáng tạo

11. Ngôn ngữ JavaScript

JavaScript được viết tắt là JS, là ngôn ngữ kịch bản, được thông dịch và thực thi trực tiếp bởi trình duyệt ở client, có thể hoạt động trên nhiều trình duyệt, nền tảng khác nhau và có tốc độ xử lý rất nhanh. JS là ngôn ngữ lập trình rất phổ biến, được sử dụng một cách rất linh hoạt, hiệu quả giúp tăng tính tương tác trên website.

JS có thể được nhúng trực tiếp vào trang web qua thẻ `<script>` hoặc được tham chiếu qua một file .js riêng. JS có thể dễ dàng thay đổi nội dung của các thẻ HTML dựa vào id hay class và hơn nữa là có thể thay đổi các định dạng CSS của thẻ.

JS là ngôn ngữ phía client nên làm giảm bớt lượng truy cập đến server, vừa tăng tốc độ xử lý vừa làm giảm gánh nặng cho server.

12. JQUERY

jQuery là một thư viện được viết bằng ngôn ngữ JavaScript, giúp xử lý các thuộc tính của đối tượng HTML, tạo các hiệu ứng hay xử lý sự kiện.

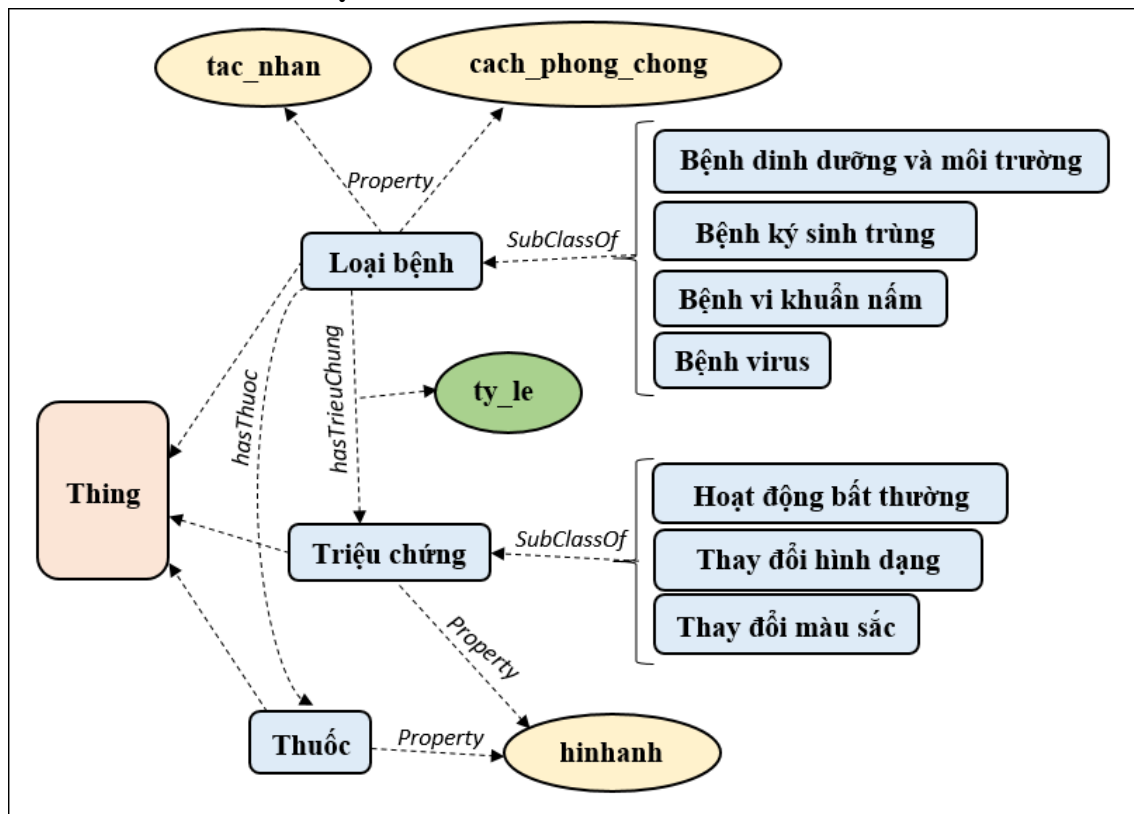
Mục đích của jQuery là làm cho việc sử dụng JavaScript trở nên dễ dàng hơn, giúp website có tính tương tác và hấp dẫn hơn. Với jQuery, chúng ta có thể làm được nhiều việc hơn mà ít tốn công sức và thời gian hơn.

Ngoài ra, jQuery còn đơn giản hóa việc sử dụng AJAX và DOM.

Một số ưu điểm của jQuery:

- Nhanh, nhỏ và nhẹ
- Tốn ít công sức mà hiệu quả là khá cao
- Tương thích với nhiều loại trình duyệt
- Chứa rất nhiều tính năng

II. Mô hình dữ liệu



Hình 5. Mô hình cơ sở dữ liệu

Toàn bộ dữ liệu được bao bọc bởi một lớp mặc định (*Thing*), bên trong chứa các lớp và các lớp con, thuộc tính của các lớp cùng với liên kết giữa các lớp.

1. Lớp - Class

Bảng 2. Danh sách các lớp

Class	SubClass
Loại bệnh	Bệnh dinh dưỡng và môi trường
	Bệnh ký sinh trùng
	Bệnh vi khuẩn nấm
	Bệnh virus
Triệu chứng	Hoạt động bất thường
	Thay đổi hình dáng
	Thay đổi màu sắc
Thuốc	

Hệ thống gồm có 3 lớp chính gồm: *Loại bệnh*, *Triệu chứng* và *Thuốc*. Trong lớp *Loại bệnh* lại có 4 lớp con gồm: *Bệnh virus*, *Bệnh vi khuẩn nấm*, *Bệnh ký sinh trùng* và *Bệnh dinh dưỡng và môi trường*. Lớp *Triệu chứng* tương tự cũng có 3 lớp con: *Hoạt động bất thường*, *Thay đổi hình dạng* và *Thay đổi màu sắc*.

2. Liên kết – Object

Bảng 3. Danh sách các liên kết

Object	Domain	Ranges
hasTrieuChung	Loại bệnh	Triệu chứng
TrieuChungof	Triệu chứng	Loại bệnh
hasThuoc	Loại bệnh	Thuốc
Thuocof	Thuốc	Loại bệnh

Liên kết *hasTrieuChung* nhằm mô tả mỗi loại bệnh có một số triệu chứng và liên kết *TrieuChungof* mô tả một triệu chứng có thể mắc một số loại bệnh. Liên kết *hasThuoc* mô tả một loại bệnh được chữa trị bởi một số loại thuốc và *Thuocof* mô tả một loại thuốc đặc trị một số loại bệnh.

3. Thuộc tính – Property

Bảng 4. Danh sách các thuộc tính

Property	Domain	Ranges
tac_nhan	Loại bệnh	xsd:string
cach_phong_chong	Loại bệnh	xsd:string
hinh_anh	Triệu chứng, Thuốc	xsd:string
cach_su_dung	Thuốc	xsd:string

Thuộc tính *tac_nhan* xác định những tác nhân gây nên bệnh. Thuộc tính *cach_phong_chong* lưu trữ cách phòng chống cho từng loại bệnh khác nhau. Thuộc tính *hinh_anh* dùng lưu đường dẫn của triệu chứng hoặc thuốc. Thuộc tính *cach_su_dung* lưu trữ cách sử dụng của từng loại thuốc.

4. Axiom

Bảng 5. Thuộc tính của các quan hệ

Axiom Type	Source	Object	Target	Type
ty_le	Loại bệnh	hasTrieuChung	Triệu chứng	xsd:interger

Để giải quyết vấn đề tay ba giữa Loại bệnh, Triệu chứng và *hasTrieuChung*, hệ thống sử dụng giải pháp là axiom. Axiom giúp ta tự định nghĩa các thuộc tính của mỗi bộ ba, có nghĩa là đối với mỗi loại bệnh liên kết tương đương với một triệu chứng sẽ có thêm thông tin là tỷ lệ mắc bệnh. Tỷ lệ này tổng sẽ là 100% đối với mỗi loại bệnh.

```
[ a                                owl:Axiom ;
  rdfs:ty_le                       ?tyle;
  owl:annotatedProperty benhtom:hasTrieuChung ;
  owl:annotatedSource     ?loaibenh ;
  owl:annotatedTarget     ?trieuchung
] .
```

Hình 6. Cách lấy dữ liệu axiôm

Cách lấy dữ liệu axiôm khác với lấy theo bộ ba như thông thường mà sẽ được định danh theo 4 thành phần gồm: `ty_le` là tên của thuộc tính axiôm, `annotatedProperty` là liên kết giữa 2 đối tượng `annotatedSource` và `annotatedTarget`.

5. Tập luật – SWRL

Bảng 6. Danh sách các luật

Ký hiệu	Tập luật	Giải thích
S1	<code>hasTrieuChung(?x, ?y) -> TrieuChungof(?y, ?x)</code>	Nếu bệnh x liên kết <code>hasTrieuChung</code> với y thì đồng thời y sẽ liên kết <code>TrieuChungof</code> với x
S2	<code>hasThuoc(?x, ?y) -> Thuocof(?y, ?x)</code>	Nếu bệnh x liên kết <code>hasThuoc</code> với y thì đồng thời y sẽ liên kết <code>Thuocof</code> với x

Các luật sẽ ràng buộc dữ liệu giúp tạo liên kết cho các đối tượng dựa trên những liên kết đã có.

Ngoài ra, tập luật còn giúp đảm bảo tính đúng và đầy đủ của các liên kết.

6. Các phương thức

Bảng 7. Các phương thức

Tên phương thức	Ý nghĩa tham số	Diễn giải
<code>get_subClassOf_trieuchung()</code>		Lấy ra các lớp con của lớp <code>trieuchung</code>
<code>get_trieuchung(\$subclass)</code>	Lớp con của lớp <code>trieuchung</code>	Lấy ra các triệu chứng trong lớp con <code>\$subclass</code>
<code>get_loaibenh()</code>		Lấy ra các loại bệnh từ các triệu chứng đã chọn
<code>get_label_trieuchung()</code>		Lấy ra nhãn của các triệu chứng
<code>get_trieuchung2()</code>		Lấy ra các triệu chứng liên quan

get_loaibenh_result()		Lấy ra loại bệnh được lọc lại bởi triệu chứng liên quan
get_label_trieuchung_result()		Lấy ra nhãn của tất cả các triệu chứng đã chọn
get_trieuchung_lienquan_result()		Lấy ra các triệu chứng liên quan sau dựa trên các bệnh đã lọc
get_detail_benh_trieuchung()		Lấy ra các triệu chứng của một loại bệnh nào đó
Get_detail_benh_thuoc()		Lấy ra danh sách các loại thuốc trị loại bệnh đó
get_detail_trieuchung()		Lấy ra nhãn và hình ảnh của triệu chứng
get_detail_trieuchung_benh()		Lấy ra các bệnh có một triệu chứng nào đó
get_detail_benh()		Lấy ra nhãn, tác nhân, cách phòng chống của một loại bệnh
get_subClassOf_benh()		Lấy ra các lớp con của lớp loaibenh
get_all_benh(\$name_subclass)	Lớp con của lớp loaibenh	Lấy ra các thể hiện của lớp \$name_subclass
get_all_trieuchung()		Lấy ra nhãn, hình ảnh của tất cả triệu chứng
get_all_thuoc()		Lấy tất cả các loại thuốc
Get_detail_thuoc()		Lấy ra thông tin của thuốc
Get_detail_thuoc_benh()		Lấy ra danh sách các loại bệnh mà thuốc đặc trị

Các phương thức trong bảng được dùng để truy vấn đến cơ sở dữ liệu và lấy ra các thông tin của những đối tượng như loại bệnh, triệu chứng, thuốc.

Tất cả các phương thức đều trả về dữ liệu dạng bảng giúp dễ dàng sử dụng để chèn vào nội dung của các phần tử web.

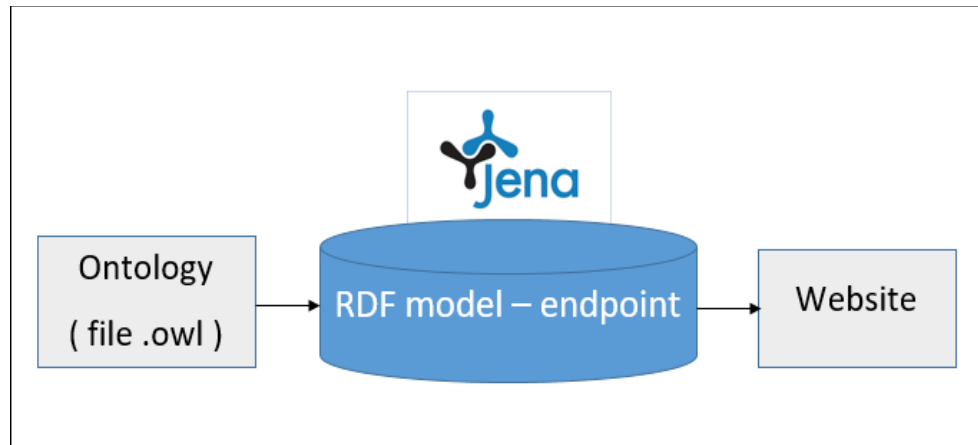
Các phương thức đã sử dụng đa số có 2 thành phần chính:

- Điểm endpoint để website kết nối với cơ sở dữ liệu
- Câu lệnh SPARQL giúp truy vấn dữ liệu

Ngoài ra, một số phương thức còn dùng các biến được truyền qua phương thức POST hoặc GET để truy vấn lọc dữ liệu.

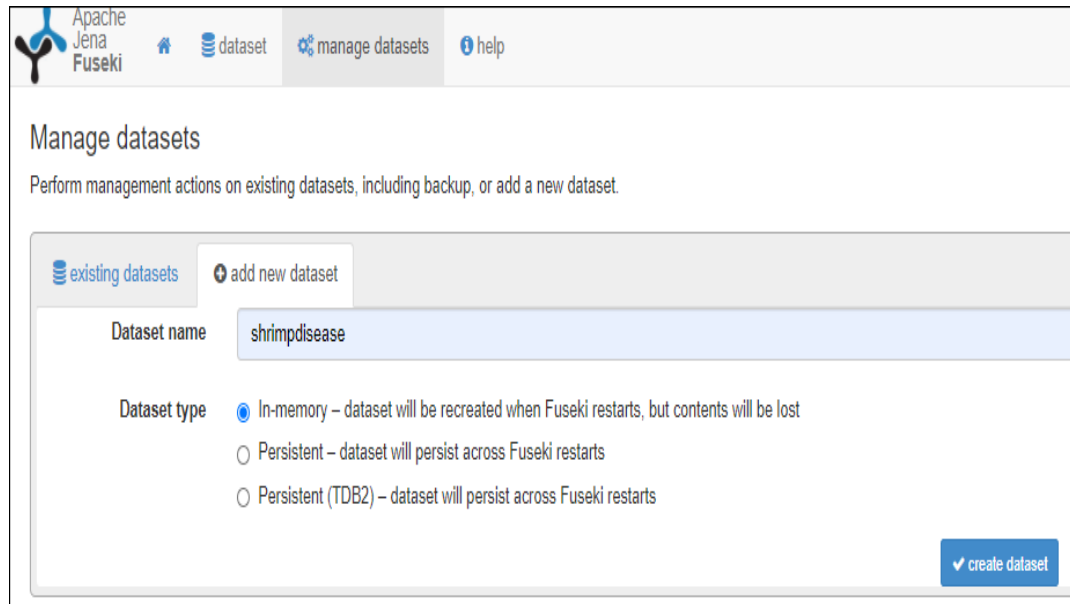
CHƯƠNG 3: CÀI ĐẶT GIẢI PHÁP

I. Server Jena Fuseki



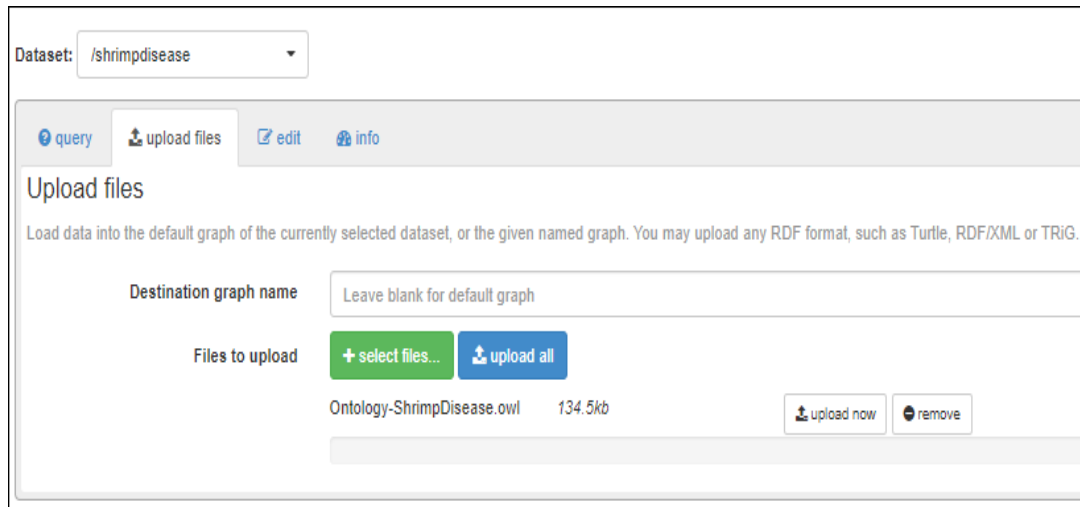
Hình 7. Sơ đồ nguyên lý hoạt động của server Jena Fuseki

Server Jena Fuseki nhận vào một file định dạng .owl và sẽ chuyển dữ liệu thành mô hình RDF gồm các bộ ba và giúp tạo endpoint (điểm truy cập) để website có thể kết nối vào truy cập dữ liệu. Jena vừa là môi trường để liên kết với website vừa là phần mềm quản trị mô hình dữ liệu RDF.



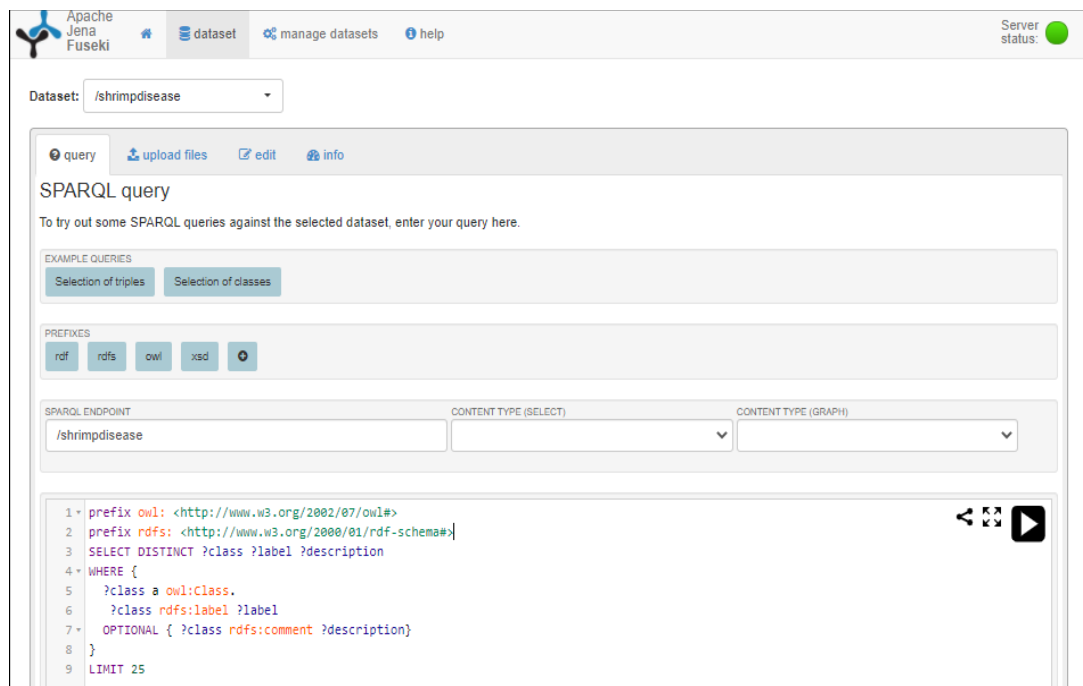
Hình 8. Tạo dataset trên server Jena Fuseki

Trước tiên, ta tạo một dataset (endpoint), là tên giúp website định danh để truy cập đến. Để khởi tạo dataset, cần nhập tên dataset và lựa chọn loại lưu trữ cho dataset là sẽ tạo lại dữ liệu hay giữ nguyên qua mỗi lần khởi động lại server.



Hình 9. Tải file ontology lên server

Từ dataset đã tạo, ta tải file ontology lên để điền dữ liệu vào dataset, đồng thời dữ liệu sẽ chuyển đổi sang mô hình RDF để có thể truy cập từ website thông qua các ngôn ngữ truy vấn RDF.



Hình 10. Truy vấn câu lệnh sparql trên Jena Fuseki

Trên server Jena Fuseki, người dùng có thể nhập câu lệnh SPARQL và thực thi thử nghiệm các câu truy vấn sparql trước khi sử dụng truy vấn dữ liệu vào website. Server có công cụ ví dụ các câu truy vấn đơn giản như lấy ra các lớp, lớp con và các thông số như nhãn, ghi chú.

II. Thư viện Sparqllib

```
<?php
require_once( "sparqllib.php" );
require_once( "../connect_sparql.php" );
?>
```

Hình 11. Import thư viện sparqllib

Dùng hàm `require_once(<ten_thu_vien>)` để nhúng thư viện `sparqllib` và các thư viện php cần thiết khác vào. Các hàm và phương thức được viết trong thư viện `connect_sparql.php`.

```
$data = sparql_get($endpoint, $sparql);
return $data;
```

Hình 12. Lấy kết quả từ CSDL

Hàm `sparql_get` giúp kết nối vào endpoint tại địa chỉ `$endpoint` và thực hiện câu truy vấn sparql là `$sparql` sau đó trả về dữ liệu dạng bảng.

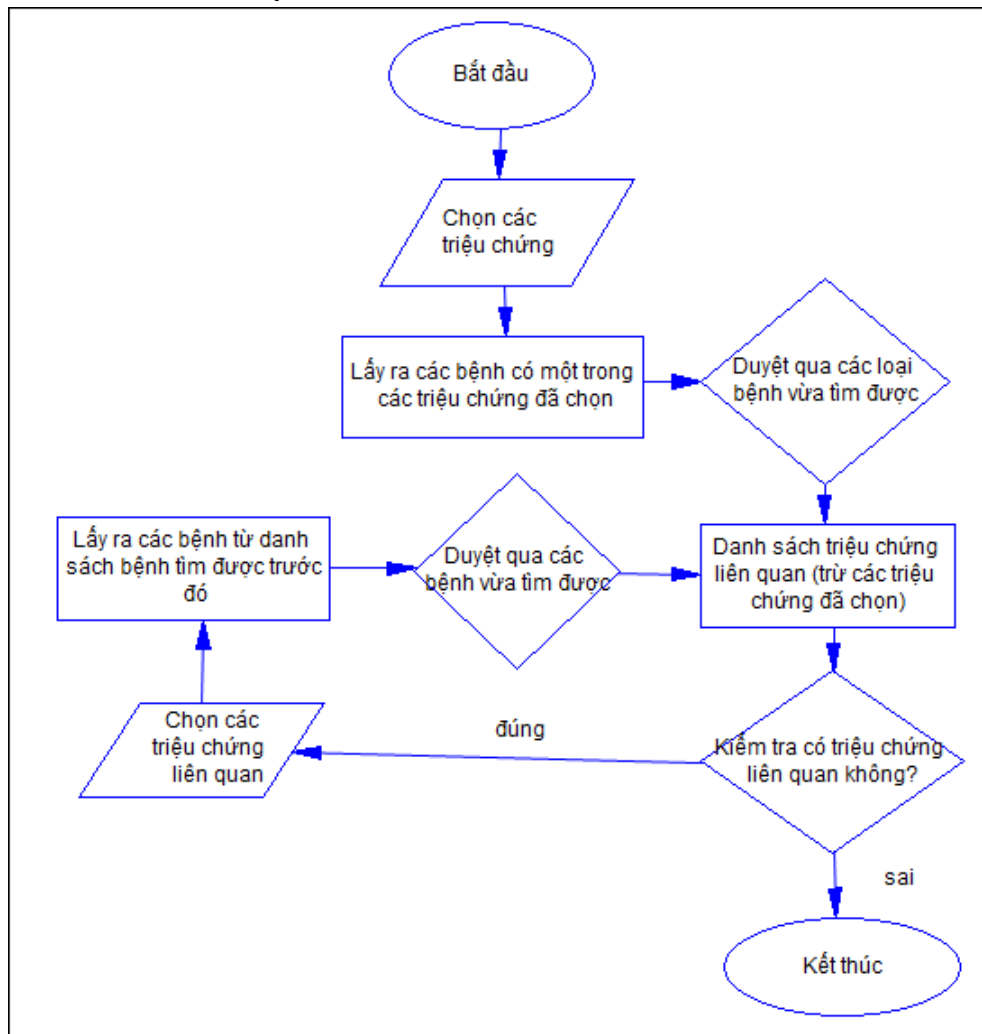
Bảng 8. Một số hàm / phương thức trong thư viện sparqllib

Tên hàm / phương thức	Diễn giải
<code>sparql_connect(\$sep)</code>	Kết nối đến cơ sở dữ liệu có url là <code>\$sep</code>
<code>sparql_errno(\$db)</code>	Trả về lỗi khi kết nối vào endpoint <code>\$db</code>
<code>sparql_get(\$sep, \$sparql)</code>	Kết nối đến endpoint <code>\$sep</code> và thực hiện câu truy vấn <code>\$sparql</code> và trả về kết quả dạng bảng
<code>sparql_query(\$sparql)</code>	Giúp thực hiện câu truy vấn dữ liệu sparql và trả về dữ liệu dạng bảng
<code>sparql_fetch_all(\$result)</code>	Lấy ra tất cả các dòng từ kết quả dạng bảng
<code>sparql_fetch_array(\$result)</code>	Lấy ra từng dòng dữ liệu trong bảng
<code>sparql_num_rows(\$result)</code>	Đếm số dòng trong bảng
<code>sparql_field_array(\$result)</code>	Lấy ra tất cả các cột
<code>sparql_field_name(\$result, \$i)</code>	Lấy ra dữ liệu ở cột <code>\$i</code> trong kết quả dạng bảng <code>\$result</code>

Các hàm và phương thức trên được dùng để truy xuất dữ liệu RDF trong cơ sở dữ liệu. Kết quả trả về là dạng bảng, có thể lấy theo từng dòng hay từng cột tùy ý.

III. Chuẩn đoán bệnh

1. Lưu đồ dữ liệu



Hình 13. Lưu đồ dữ liệu chẩn đoán bệnh

Đầu tiên, lưu đồ dữ liệu chẩn đoán bệnh nhận vào danh sách các triệu chứng (tc), từ đó lấy ra được danh sách các bệnh (lb).

Sau đó, từ danh sách bệnh, sẽ tìm ra được các triệu chứng liên quan (tc2).

Tiếp theo, ta chọn từ các triệu chứng liên quan, hệ thống sẽ giới hạn lại các bệnh (dựa vào lb và tc2) và danh sách lb sẽ được cập nhật kèm theo tc2 cũng được cập nhật. Đến khi nào tc2 rỗng, hệ thống sẽ dừng lại.

```
if (isset($_POST['checkbox'])) {  
    foreach($_POST['checkbox'] as $value) {
```

Hình 14. Phương thức POST

Hệ thống chủ yếu sử dụng phương thức POST để truyền dữ liệu qua các trang.

2. Nguyên lý hoạt động

<input type="checkbox"/> Nổi trên mặt ao, dạt vào bờ	<input type="checkbox"/> Nằm yên ở đáy	<input type="checkbox"/> Lờ đờ, hôn mê
<input type="checkbox"/> Hoạt động yếu	<input type="checkbox"/> Kéo đàn bơi vòng vòng	<input type="checkbox"/> Phát triển nhanh bất thường, đột ngột ngừng ăn, ruột rỗng

Hình 15. Danh sách các triệu chứng

Từ những triệu chứng được lấy từ cơ sở dữ liệu, người dùng chọn một số triệu chứng và tiến hành chuẩn đoán giai đoạn thứ nhất.

<div>80 Bệnh mềm vỏ ⓘ</div> <div>25 Bệnh hoại tử mắt ⓘ</div> <div>15 Bệnh đục cơ ⓘ</div> <div>10 Bệnh vi khuẩn Vibrio ⓘ</div>	<div>Triệu chứng liên quan</div> <div><input type="checkbox"/> Bị nhiều sinh vật bám</div> <div><input checked="" type="checkbox"/> Hoạt động yếu</div> <div><input type="checkbox"/> Các vùng trong mắt hoại tử</div> <div><input type="checkbox"/> Đối thành màu tối: Xanh, đỏ, đen, xám</div> <div><input type="checkbox"/> Mang xanh hoặc nâu</div> <div><input type="checkbox"/> Lờ đờ, hôn mê</div>
---	---

Hình 16. Kết quả chuẩn đoán sơ bộ

Kết quả của chuẩn đoán sơ bộ là danh sách các bệnh kèm tỷ lệ mắc bệnh cùng với danh sách triệu chứng liên quan (nếu có). Chọn tiếp triệu chứng liên quan (nếu có) và tiến hành chẩn đoán tiếp.

<div>90 Bệnh mềm vỏ ⓘ</div> <div>20 Bệnh đục cơ ⓘ</div>	<div>Triệu chứng liên quan</div> <div><input type="checkbox"/> Bị nhiều sinh vật bám</div> <div><input type="checkbox"/> Kén ăn, chậm lớn</div>
---	---

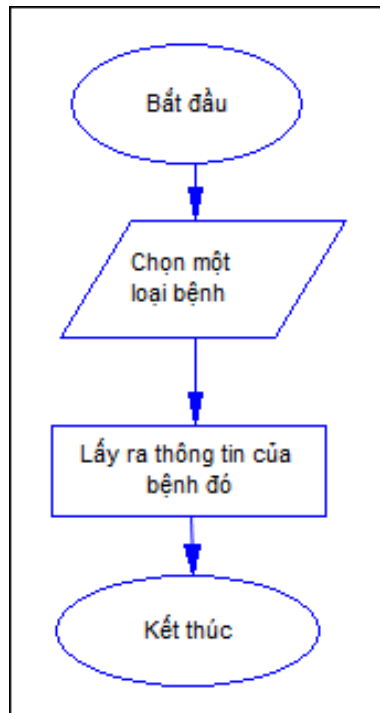
Hình 17. Kết quả chẩn đoán nâng cao

Ở bước này, các loại bệnh và triệu chứng liên quan được giới hạn lại và tiếp tục chọn cho đến khi không còn triệu chứng liên quan hoặc chỉ còn một loại bệnh thì dừng lại.

IV. Tra cứu

1. Tra cứu bệnh

Lưu đồ tra cứu bệnh

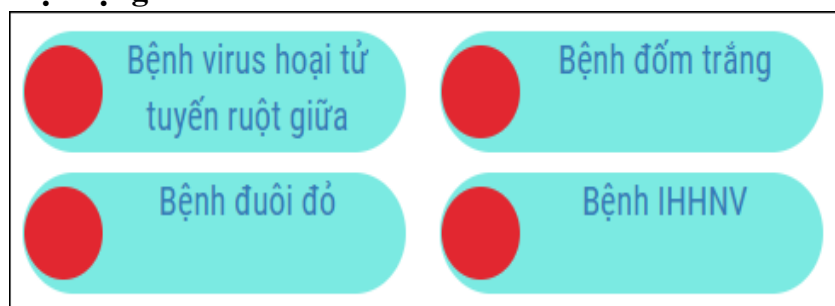


Hình 18. Lưu đồ tra cứu bệnh

Đầu tiên chọn một bệnh trong danh sách các bệnh. Sau đó, hệ thống lấy ra các thông tin của bệnh đó.

Chức năng tra cứu bệnh sử dụng phương thức GET để truyền dữ liệu qua các trang.

Nguyên lý hoạt động



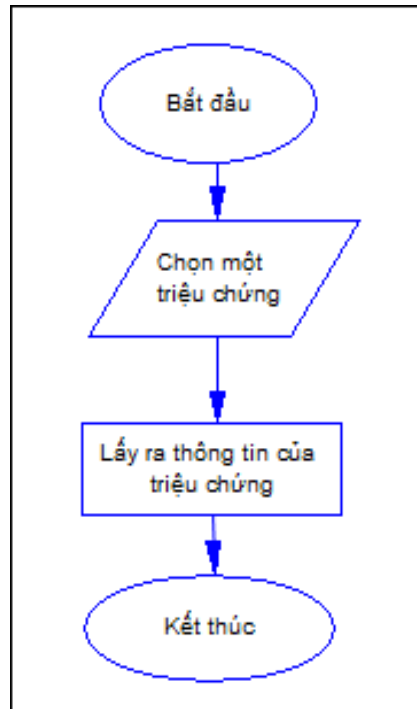
Hình 19. Danh sách các bệnh

Đầu tiên, hệ thống sẽ hiển thị danh sách tất cả bệnh cho người dùng chọn một trong số chúng.

Người dùng sau khi chọn một bệnh, hệ thống sẽ lấy ra tất cả thông tin về bệnh đó gồm tác nhân gây bệnh, cách phòng chống, thuốc đặc trị và các triệu chứng của bệnh.

2. Tra cứu triệu chứng

Lưu đồ dữ liệu



Hình 20. Lưu đồ dữ liệu tra cứu triệu chứng

Hệ thống nhận vào một triệu chứng được chọn, sau đó lấy ra các thông tin của triệu chứng kèm theo danh sách các bệnh tương ứng.

Chức năng tra cứu triệu chứng sử dụng phương thức GET để truyền dữ liệu qua các trang.

Nguyên lý hoạt động

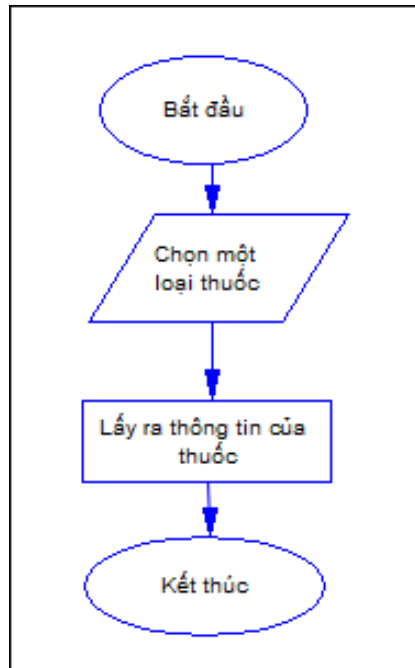


Hình 21. Danh sách triệu chứng

Đầu tiên, hệ thống hiển thị danh sách các triệu chứng để người dùng chọn. Sau khi chọn triệu chứng, hệ thống sẽ lấy ra thông tin của triệu chứng đó và danh sách các bệnh có thể mắc phải.

3. Tra cứu thuốc

Lưu đồ dữ liệu



Hình 22. Lưu đồ dữ liệu tra cứu thuốc

Hệ thống nhận vào một loại thuốc, sau đó tìm và lấy ra các thông tin của thuốc kèm danh sách các bệnh mà thuốc có thể đặc trị.

Chức năng tra cứu thuốc sử dụng phương thức GET để truyền dữ liệu qua các trang.

Nguyên lý hoạt động



Hình 23. Danh sách các loại thuốc

Đầu tiên, hệ thống sẽ hiện tất cả loại thuốc lên màn hình và có công cụ để người dùng có thể tìm kiếm nhanh hơn. Sau khi chọn một loại thuốc, hệ thống sẽ tìm trong cơ sở dữ liệu và lấy ra các thông tin của thuốc và danh sách các bệnh mà thuốc điều trị.

CHƯƠNG 4: ĐÁNH GIÁ KIỂM THỬ

I. Mục tiêu kiểm thử

Mục tiêu chính của kiểm thử chủ yếu là chức năng chẩn đoán bệnh tâm bao gồm các tiêu chuẩn sau:

- Kiểm tra xem ứng dụng có hoạt động tốt không
- Có phát sinh lỗi gì không
- Kết quả trả về có đúng hay không

Nhìn chung, mục tiêu cuối cùng vẫn là tìm ra các lỗi để khắc phục, sửa chữa trong tương lai.

II. Kịch bản kiểm thử

Bảng 9. Kịch bản kiểm thử

Ký hiệu kịch bản	Kịch bản kiểm thử
KB001	Truy cập vào chức năng kiểm thử, kiểm tra xem có đúng danh sách triệu chứng, có xuất hiện lỗi không?
KB002	Tiến hành chẩn đoán sơ bộ và kiểm tra kết quả gồm danh sách bệnh có đúng không, có đúng tỷ lệ không, các triệu chứng liên quan có đúng không?
KB003	Tiến hành chuẩn đoán nâng cao, kiểm tra loại bệnh có ít lại không, tỷ lệ bệnh có đúng không, các triệu chứng liên quan còn lại có đúng không?
KB004	Truy cập vào chức năng tra cứu bệnh, sau đó chọn loại bệnh và kiểm tra kết quả xem có đúng thông tin của bệnh không, các triệu chứng và loại thuốc có đúng không?
KB005	Truy cập vào chức năng tra cứu triệu chứng, sau đó chọn triệu chứng và kiểm tra kết quả xem có đúng thông tin triệu chứng chưa, những loại bệnh có thể mắc phải có đúng không?
KB006	Truy cập vào chức năng tra cứu thuốc, sau đó chọn loại thuốc và kiểm tra kết quả xem có đúng thông tin thuốc không, danh sách loại bệnh có đúng không?

Tiến hành kiểm thử các kịch bản ở bảng bên trên.

Môi trường kiểm thử

Thiết bị kiểm thử: Laptop Dell 3443 Inspiron

Cấu hình thiết bị:

- Chip Intel(R) Core(TM) i7-5500U
- RAM 4 GB
- Hệ điều hành Windows 10
- Trình duyệt Google Chrome

III. Kết quả kiểm thử

Bảng 10. Kết quả kiểm thử kịch bản

Kịch bản	Mức độ đạt yêu cầu
KB001	Đạt
KB002	Đạt
KB003	Đạt
KB004	Đạt
KB005	Đạt
KB006	Đạt

Nhìn chung, các kết quả kiểm thử đều đạt yêu cầu.

➤ Kết quả kiểm thử

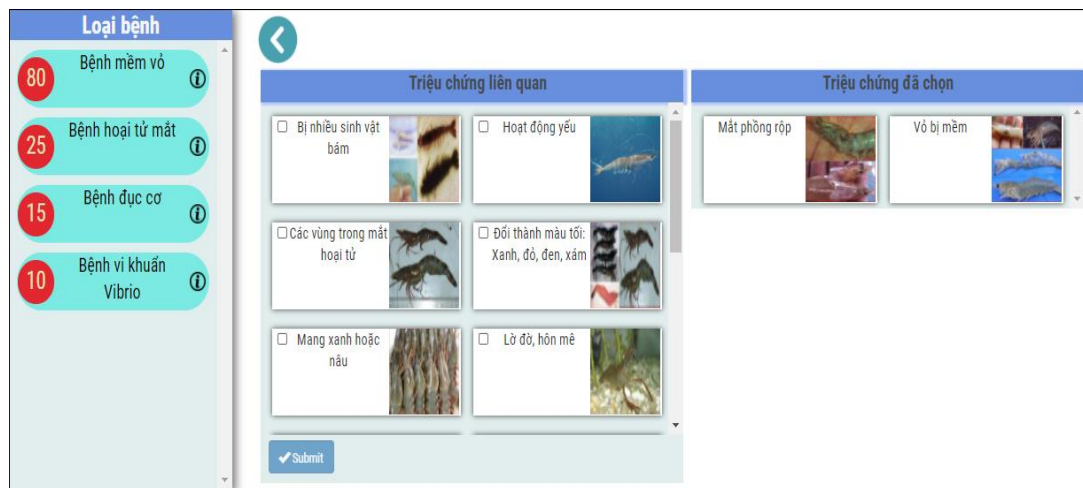
Giao diện chuẩn đoán bệnh

The screenshot displays a web-based interface for diagnosing shrimp diseases. At the top, there is a header 'Chọn Triệu Chứng' (Choose Symptoms). Below it, there are three main categories of symptoms, each with a corresponding color-coded bar: 'Bộ phận bị biến dạng' (Deformed parts) in light blue, 'Thay đổi màu sắc' (Color change) in medium blue, and 'Hoạt động bất thường' (Abnormal activity) in dark blue. Under the 'Hoạt động bất thường' category, there are seven checkboxes, each accompanied by a small image of a shrimp and a description of the symptom: 'Nổi trên mặt ao, dạt vào bờ' (Floating on the surface of the pond, drifting to the shore), 'Nằm yên ở đáy' (Lying quietly at the bottom), 'Lờ đờ, hôn mê' (Lethargic, comatose), 'Kén ăn, chậm lớn' (Picky eating, slow growth), 'Hoạt động yếu' (Weak activity), 'Kéo đàn bơi lòng vòng' (Dragging the group swimming in circles), and 'Phát triển nhanh bất thường, đột ngột ngừng ăn, ruột rỗng' (Abnormal rapid development, suddenly stopping eating, empty gut). At the bottom left, there is a 'Submit Diagnosis' button with a checkmark icon, and at the bottom right, there is a checkbox labeled 'all'.

Hình 24. Giao diện chuẩn đoán bệnh

Khi vào giao diện chuẩn đoán bệnh, hệ thống sẽ hiện ra các nhóm bệnh để người dùng lựa chọn. Sau khi chọn nhóm bệnh, giao diện chuẩn đoán bệnh hiện tất cả các triệu chứng của nhóm cho người dùng lựa chọn.

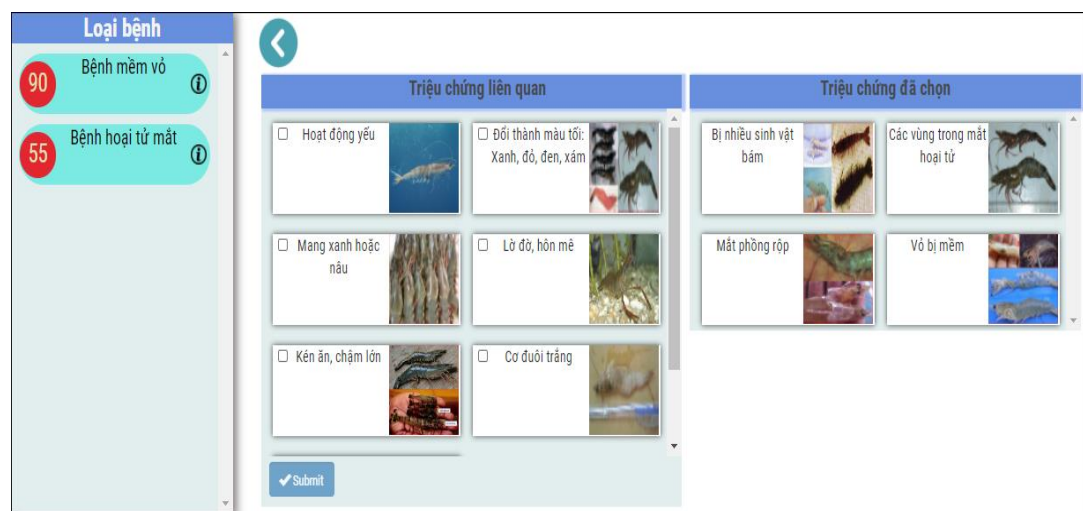
Giao diện kết quả chuẩn đoán sơ bộ



Hình 25. Giao diện kết quả chuẩn đoán sơ bộ

Giao diện chuẩn đoán sơ bộ hiển thị kết quả các bệnh kèm theo tỷ lệ mắc bệnh, đồng thời tìm ra các triệu chứng liên quan đến bệnh.

Giao diện kết quả chẩn đoán nâng cao

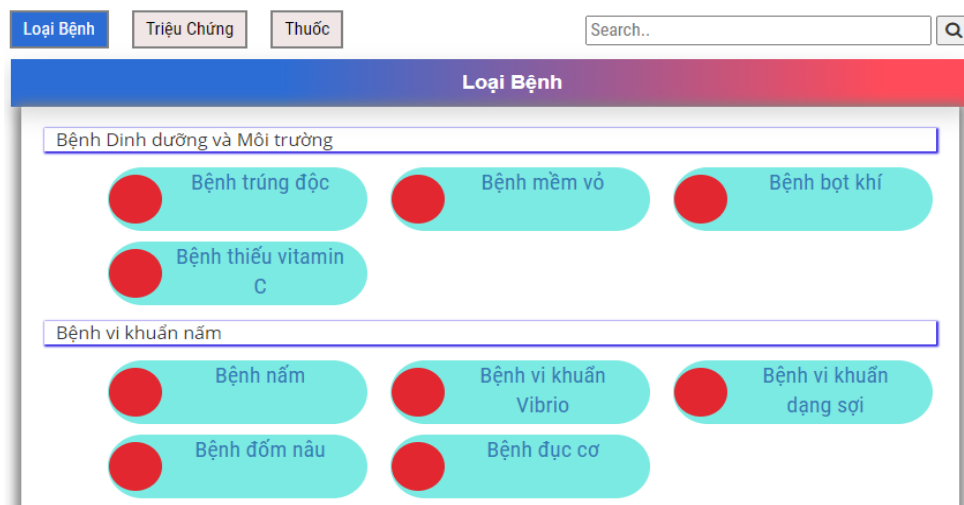


Hình 26. Giao diện kết quả chẩn đoán nâng cao

Giao diện chuẩn đoán nâng cao sẽ hiển thị các loại bệnh đã được giới hạn lại và làm mới các triệu chứng liên quan để tiến hành chọn tiếp.

Ở bước này, hệ thống có thể thực hiện lặp lại cho đến khi nào không còn triệu chứng liên quan nữa. Đến lúc đấy, loại bệnh sẽ chỉ còn lại một bệnh duy nhất và tỷ lệ sẽ 100%.

Giao diện tra cứu bệnh



Hình 27. Giao diện tra cứu bệnh

Giao diện tra cứu bệnh hiển thị tất cả các loại bệnh để người dùng chọn. Sau khi chọn sẽ xem được các thông tin về bệnh đó.

Giao diện thông tin chi tiết bệnh



Hình 28. Giao diện thông tin chi tiết bệnh

Giao diện thông tin chi tiết bệnh hiển thị thông tin về bệnh như tác nhân gây bệnh, cách phòng trị. Ngoài ra, còn lọc ra danh sách các thuốc đặc trị và các triệu chứng của bệnh đó.

Giao diện tra cứu triệu chứng

Loại Bệnh

Triệu Chứng

Thuốc

Triệu Chứng

Bộ phận bị biến dạng

Mắt phồng rộp	Vỏ bị mềm	Các vùng trong mắt hoại tử	Vỏ xù xì, biến dạng
Nắp mang phồng lên	Vỏ và phần phụ bị ăn mòn, gãy, rụng	Bị nhiều sinh vật bám	Bọt khí bám vào các phần phụ và mang
Đuôi hoại tử	Đuôi bị đứt	Các phần phụ và vỏ kitin hoại tử	Đuôi phồng
Trầy biến dạng, sợi anten xoắn queo	Gan, tụy teo lại hoặc hoại tử		

Hình 29. Giao diện tra cứu triệu chứng

Giao diện tra cứu triệu chứngs hiển thị tất cả các triệu chứng được phân loại theo nhóm để người dùng có thể chọn.

Giao diện thông tin chi tiết triệu chứng

Trầy biến dạng, sợi anten xoắn queo



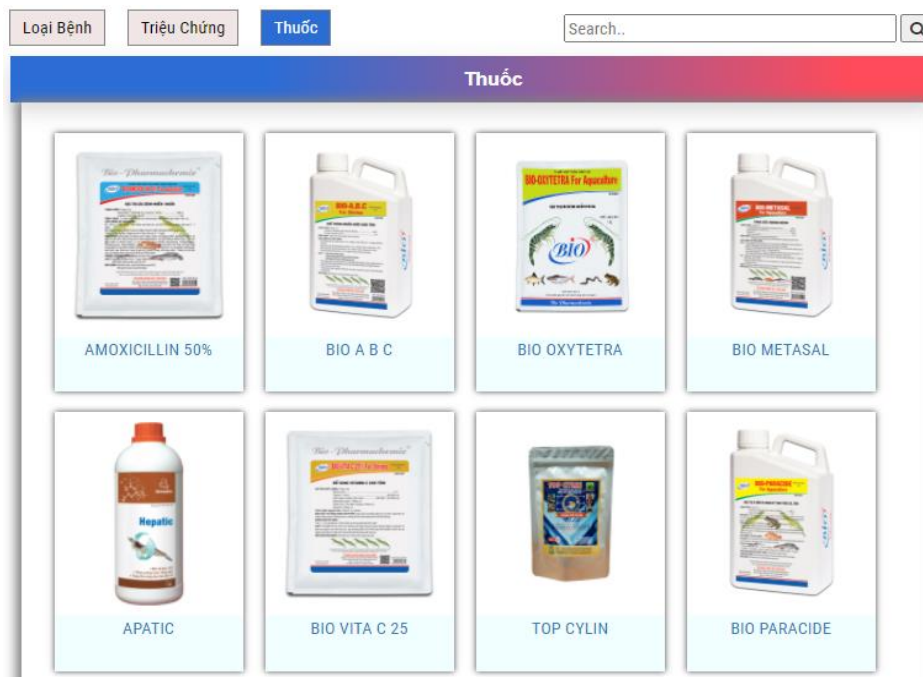
Loại Bệnh

Bệnh IHNV

Hình 30. Giao diện thông tin chi tiết triệu chứng

Giao diện thông tin chi tiết triệu chứng hiển thị hình ảnh triệu chứng và tìm ra danh sách các bệnh có thể mắc phải.

Giao diện tra cứu thuốc



Hình 31. Giao diện tra cứu thuốc

Giao diện tra cứu thuốc sẽ hiển thị tất cả các loại thuốc để người dùng xem và tiến hành chọn. Ngoài ra, người dùng có thể nhập tên thuốc ở ô phía trên góc phải để tìm kiếm một số loại thuốc.

Giao diện thông tin chi tiết thuốc



Hình 32. Giao diện thông tin chi tiết thuốc

Giao diện thông tin chi tiết thuốc hiển thị hình ảnh thuốc, hướng dẫn sử dụng và danh sách các bệnh chữa trị.

PHẦN 3: KẾT LUẬN

I. Kết quả đạt được

1. Kết quả

Hoàn thành trọng tâm nghiên cứu là cơ sở dữ liệu phân cấp ontology bao gồm cách thiết kế, cách sử dụng và ý nghĩa của các thành phần bên trong.

Xây dựng được ứng dụng web chẩn đoán bệnh tâm với các chức năng như:

- Chuẩn đoán bệnh
- Tra cứu:
 - Tra cứu bệnh
 - Tra cứu triệu chứng
 - Tra cứu thuốc

2. Hạn chế

Ứng dụng có một số hạn chế:

- Cơ sở dữ liệu hiện là cơ sở dữ liệu tĩnh, không quản trị trực tiếp trên website được mà phải qua phần mềm thiết kế ontology
- Về phần giao diện, vẫn còn chưa được bắt mắt và chuyên nghiệp

II. Hướng phát triển

Trong tương lai, ứng dụng có thể phát triển thêm về:

- Ứng dụng có thể phát triển thêm chức năng quản trị, giúp cập nhật các danh sách loại bệnh, triệu chứng và thuốc cùng những thông tin chi tiết liên quan một cách nhanh chóng và chính xác hơn
- Có thể ứng dụng các công nghệ hỗ trợ thiết kế để giao diện tối ưu hơn

TÀI LIỆU THAM KHẢO

- [1]. Ontology là gì. [Online]. Có sẵn tại: http://thuvien.tcdktcnsl.edu.vn/files/products/khai_niem_ve_ontology_L63.pdf. [Truy cập tháng 10/2020].
- [2] Wikipedia, “Ontology”. [Online]. Có sẵn tại: [https://vi.wikipedia.org/wiki/Bản_thể_học_\(khoa_học_thông_tin\)](https://vi.wikipedia.org/wiki/Bản_thể_học_(khoa_học_thông_tin)). Truy cập tháng 10/2020.
- [3] Wikipedia, “RDF”. [Online]. Có sẵn tại: <https://vi.wikipedia.org/wiki/RDF>. [Truy cập tháng 10/2020].
- [4] W3C, “SPARQL Query Language for RDF”. [Online]. Có sẵn tại: <https://www.w3.org/TR/rdf-sparql-query/>. [Truy cập tháng 11/2020].
- [5] SPARQL RDF Library for PHP. [Online]. Có sẵn tại: <http://graphite.ecs.soton.ac.uk/sparqllib/>. [Truy cập tháng 11/10/2020].
- [6] W3schools, “CSS”. [Online]. Có sẵn tại: <https://www.w3schools.com/w3css/default.asp>. [Truy cập tháng 12/2020].
- [7] W3schools, “JS”. [Online]. Có sẵn tại: <https://www.w3schools.com/js/DEFAULT.asp>. [Truy cập tháng 12/2020].
- [8] W3schools, “HTML”. [Online]. Có sẵn tại: <https://www.w3schools.com/html/>. [Truy cập tháng 12/2020].
- [9] TS. Bùi Quang Tề, “Bệnh của tôm nuôi và biện pháp phòng trị”, nhà xuất bản Nông nghiệp, Hà Nội, 2003.