ABSTRACT

In this project, we are creating a Decision Support System related to healthcare of the elderly people. For this, we created a domain level ontology based on Handbook of Geriatrics and then mapped it to an upper level ontology, BFO. The knowledge base was created with the help of the Handbook of Geriatrics. Geriatrics is basically a collective term used for diseases related to the elderly people. In this project, we have classified these diseases with their symptoms and precautions and with the help of an android app made a Decision support system based on this knowledge base.

Furthermore, we have also used this Handbook of Geriatrics to represent the knowledge base in two other frameworks. We have developed a UML structure and a Prolog structure in addition to the ontology structure. We have then compared these three structures and recorded some advantages and disadvantages of the same. We also used the Prolog structure to create a sample of the Decision support system in Prolog.

Finally, we have created an Android Application and a Web Application, based on this ontological framework. Both the applications are able to display the ontology structure and can predict the disease based on the symptoms entered.

INDEX

CHAPTER 1

INTRODUCTION

1.1 Semantic Web

The web content as seen today is scattered all over the network. Although, this content is human understandable but has a huge downside. This web content cannot be efficiently accessed by machines. To make the web content over the network machine-processible Semantic Web Technologies like RDF, OWL, Protégé and many more are used.

The automation of processing of web content is necessary because it would allow efficient ways for Searching, Extracting, Maintaining, Uncovering and Viewing information.

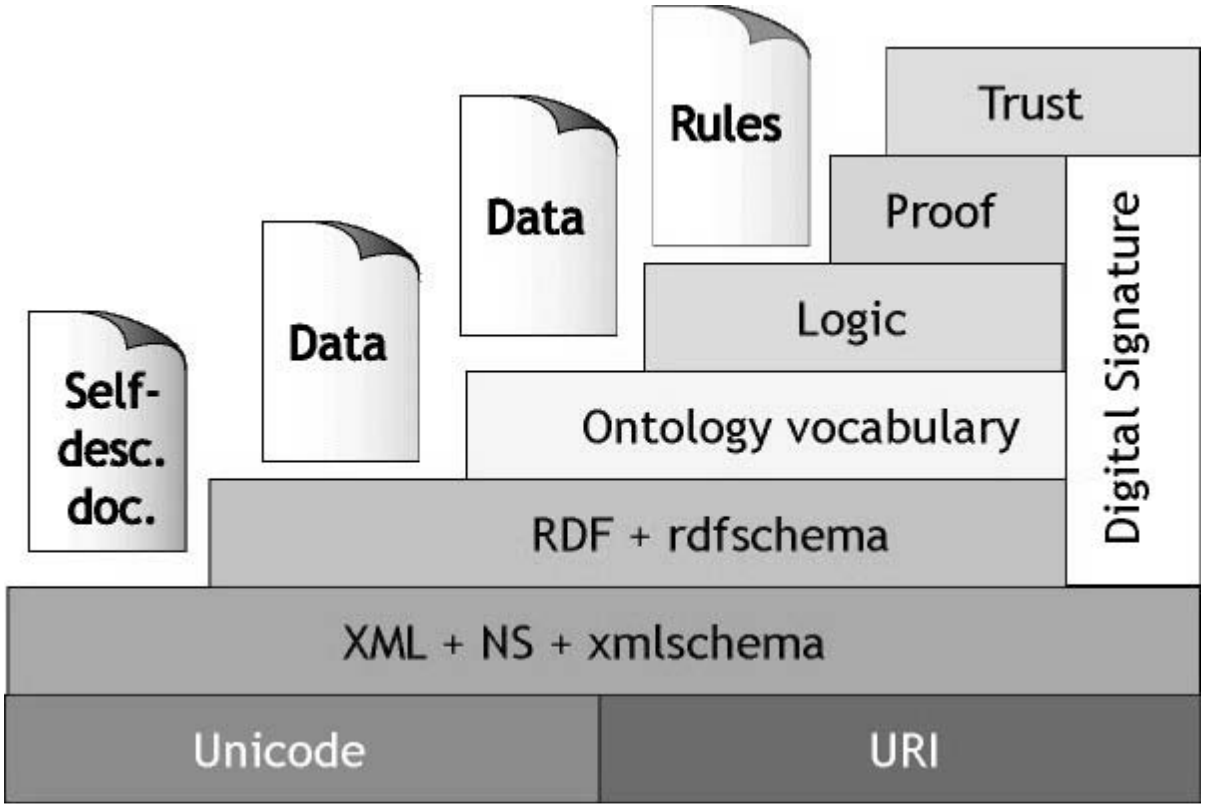


Fig. 1.1.1 – Layered Architecture of Semantic Web

CHAPTER 2

OBJECTIVE

- The main objective of this project is to develop an ontology based on the knowledge base of the Handbook of Geriatrics. This ontology is then used to create a Decision support system for the elderly people.

- In addition to the ontology structure of the knowledge base, two other structures are also developed. The UML structure is a class diagram for the Decision support system and the Prolog structure is used to create a sample of the Decision support system.

- A comparison is made between the three structures thus created, which yields the advantages and disadvantages of the structures.

- The ontology created is a domain level ontology and lacks the dynamic structure. For this purpose, the ontology is mapped to an upper level ontology, BFO, which adds dynamics to the whole structure.

- An Android application and a Web application are created to implement the Decision support system based on the ontology structure.

CHAPTER 3

UML STRUCTURE

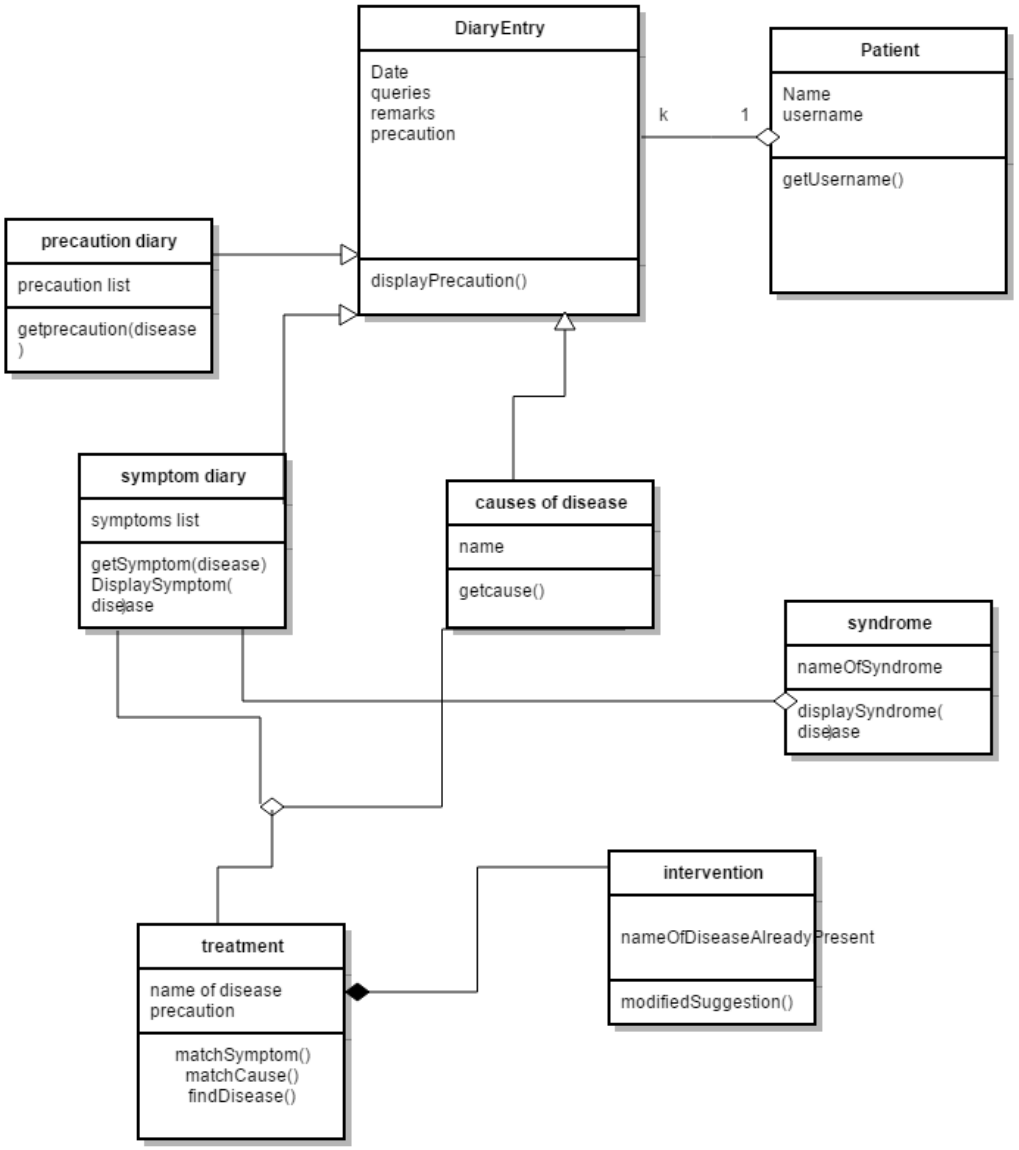


Fig. 3.1 – UML structure of the knowledge base

A UML structure of a knowledge base depicts the entire working of the application. It can be used to show how the knowledge base can be implemented on different frameworks.
The entire knowledge base is divided into classes and functions and these are named under different modules. The modules are connected to show the relationship between the data in the knowledge base.

CHAPTER 4

PROLOG STRUCTURE

Prolog is logic based programming language generally used for Artificial Intelligence and Decision making by computers. The language follows first order logic and formal logic and the relations are represented as facts and rules.

For the purposes of our project, we have extracted the vital information from the Handbook of Geriatrics and converted this into facts and rules.

There are basically three parts of Prolog

- Knowledge Base
- Fact base
- Inference Engine

Knowledge base is represented as:

```
disease(Patient, pneumoconiosis):-
        symptom(Patient, chronic_cough),
        symptom(Patient, shortness_of_breath).
```

Fact base contains facts used to match against the antecedent part of rules stored in the knowledge base.

Inference engine is used to deduce new facts.

The following snapshots of the Prolog editor show the facts and rules, and also depict some of the output scenarios for the Decision support system.

File Edit Browse Compile Prolog Pce Help

dise.pl

```
go :-
    write('What is the patient''s name? '),
    read(Patient),
    disease(Patient,Disease),
    write_list([Patient,' probably has ',Disease,'.']),nl,
    inst(Disease).


go :-
    write('Sorry, I don''t seem to be able to'),nl,
    write('diagnose the disease.'),nl.

symptom(Patient,dyseptic_without_ulcer) :-
    write_list(['Does ',Patient,' have a dyseptic_without_ulcer (y/n) ?']),
    response(Reply),
    Reply='y'.

symptom(Patient,chronic_gastro_dudenitis) :-
    write_list(['Does ',Patient,' have a chronic_gastro_dudenitis (y/n) ?']),
    response(Reply),
    Reply='y'.

symptom(Patient,gastric_ulcer) :-
    write_list(['Does ',Patient,' have a gastric_ulcer (y/n) ?']),
    response(Reply),
    Reply='y'.

symptom(Patient,headache) :-
    write_list(['Does ',Patient,' have a headache (y/n) ?']),
    response(Reply),
    Reply='y'.
```

Fig. 4.1 – Facts and Rules in Prolog

dise.pl

File Edit Browse Compile Prolog Pce Help

dise.pl

```
symptom(Patient, reflux_of_food),
symptom(Patient, vomitting).



inst(hypertension):-
        write('PRECAUTION----take rest').
inst(non_ulcer_dyspepcia):-
        write('PRECAUTION:------No definitive diagnosis because of psychological factor,gastrointestinal dymotility,infection and the
rapy of peptic ulcer').

inst(osteoArthiritis):-
        write('PRECAUTION:------diagnosis is limited to symptomatic relief with analegiscs and physiotherapy').

inst(cancer_of_gi_tract):-write('PRECAUTION:------1.colorectum exams for screening for cancer.2.digital rectum exams 3.checking stool
 for acute blood').
inst(hiatus_hernia_and_gastroerophageal_reflux):-
        write('PRECAUTION:------1.lose weight 2.take proper diet 3.avoid tea cofee and cola 4.reduce use of saturated fat 5.take anta
cid/skimmilk for hot burn').

write_list([]).
write_list([Term| Terms]) :-
    write(Term),
    write_list(Terms).

response(Reply) :-
    get_single_char(Code),
    put_code(Code), nl,
    char_code(Reply, Code).
```

Fig. 4.2 – Facts and Rules in Prolog

dise.pl

File Edit Browse Compile Prolog Pce Help

dise.pl

```
symptom(Patient,immobility),
symptom(Patient,have_fever).

disease(Patient, hypertension):-
symptom(Patient,headache),
symptom(Patient,shortness_of_breath),
symptom(Patient,nosebleed).
disease(Patient, non_ulcer_dyspepcia):-
symptom(Patient,dyseptic_without_ulcer),
        symptom(Patient,chronic_gastro_dudenitis),
        symptom(Patient,gastric_ulcer).

disease(Patient,osteoArthiritis):-
symptom(Patient,pain),
symptom(Patient,stiffness),
symptom(Patient,bone_swelling),
symptom(Patient,loss_of_movement),
symptom(Patient,instability).
disease(Patient, cancer_of_gi_tract):-
symptom(Patient, change_of_bowel_habit),
symptom(Patient, constipation),
symptom(Patient, diarrhoea),
symptom(Patient, decreased_size_of_stool),
symptom(Patient, loss_of_appetite),
symptom(Patient, weight_loss).
disease(Patient, hiatus_hernia_and_gastroerophageal_reflux):-
symptom(Patient, heartburn),
symptom(Patient, dysphagia),
symptom(Patient, pain_in_lower_sternum),
symptom(Patient, belching),
symptom(Patient, reflux_of_food),
symptom(Patient, vomitting).
```

Fig. 4.3 – Facts and Rules in Prolog

SWI-Prolog (AMD64, Multi-threaded, version 7.2.2)

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.2.2)
Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- go.
What is the patient's name? viraj
|: .
viraj  starting diagnosis

Does viraj have a headache (y/n) ?y
Does viraj have a shortness_of_breath (y/n) ?y
Does viraj have a nosebleed (y/n) ?y
viraj probably has hypertension.
PRECAUTION----take rest
true .

2 ?- go.
What is the patient's name? viru
|: .
viru  starting diagnosis

Does viru have a headache (y/n) ?n
Does viru have a dyseptic_without_ulcer (y/n) ?y
Does viru have a chronic_gastro_dudenitis (y/n) ?y
Does viru have a gastric_ulcer (y/n) ?y
viru probably has non_ulcer_dyspepcia.
PRECAUTION:------No definitive diagnosis because of psychological factor,gastrointestinal dymotility,infection and therapy of peptic ulcer
true .

3 ?- go.
What is the patient's name? shubh.
shubh  starting diagnosis

Does shubh have a headache (y/n) ?n
Does shubh have a dyseptic_without_ulcer (y/n) ?n
Does shubh have a pain (y/n) ?n
Does shubh have a change_of_bowel_habit (y/n) ?n
Does shubh have a   heartburn (y/n) ?n
Sorry, I don't seem to be able to
diagnose the disease.
true.

4 ?-
```

Fig. 4.4 – Output in Prolog

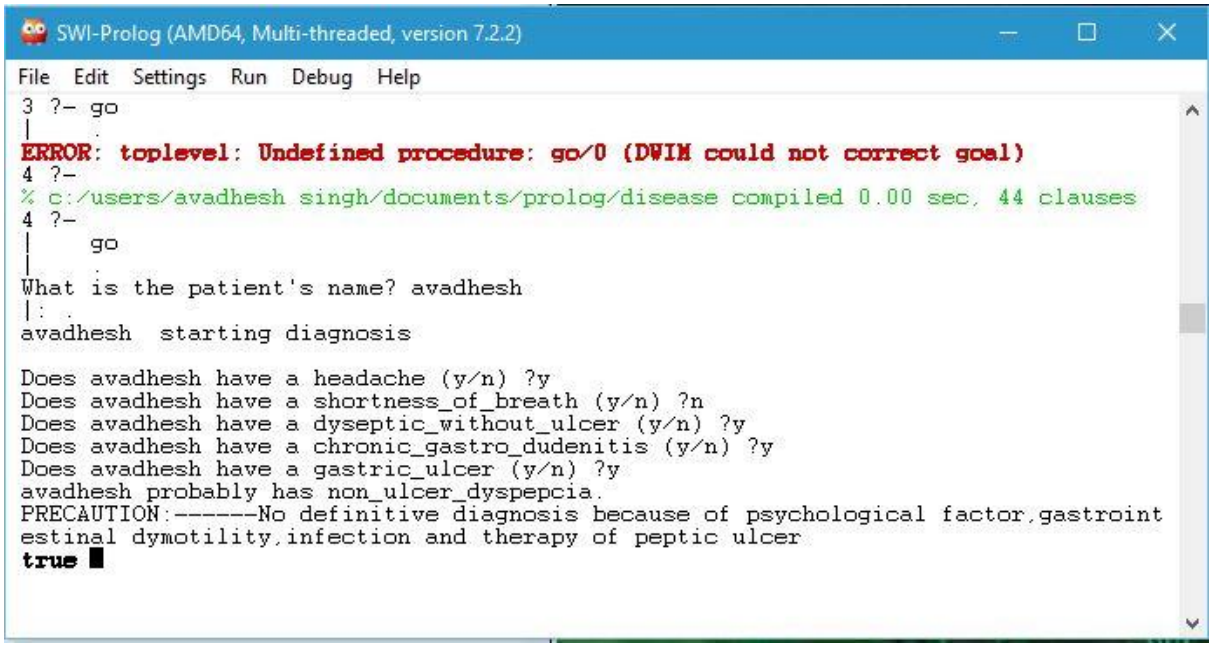


Fig. 4.5 – Output in Prolog

CHAPTER 5

ONTOLOGY STRUCTURE

5.1 Introduction

Ontology is a specification of a conceptualization; this means that through ontology the conceptual data can be easily specified. This is done with the help of classes and properties for these classes.

The classes can be considered as various domains over which our knowledge base is spread, and the properties are used to connect these classes. This imparts a semantic to the ontological structure which can be efficiently used for various purposes.

The data in ontology is stored in the form of triplets, contrary to the conventional data store, in which the data is stored in tuples. Data is represented as Subject, Object and Predicate. The predicate part connects the subject and the object.

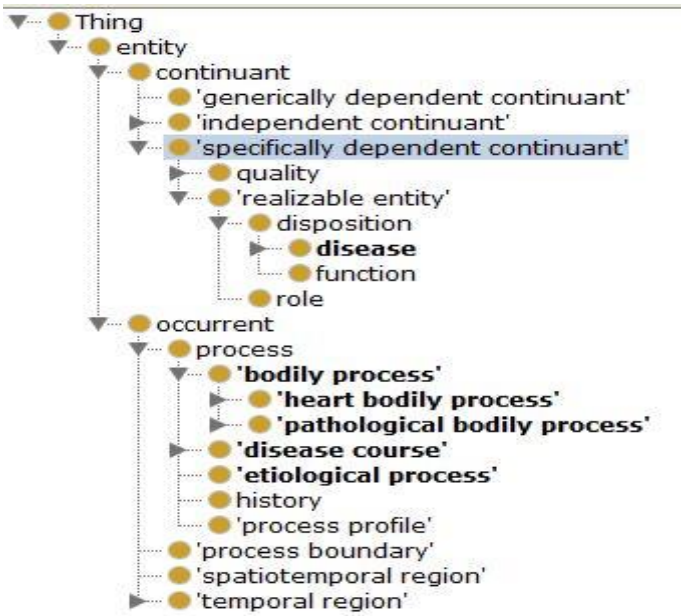


Fig. 5.1.1 – Class hierarchy for ontology

The figure shown above depicts the class hierarchy for ontology. The uppermost class is the Thing class, and every class is a subclass of the Thing class.

The properties in an ontological structure are of two types

- Object Properties
- Data Properties

Fig. 5.1.2 – Object properties for ontology

Fig.5.1.3 – Data properties for ontology

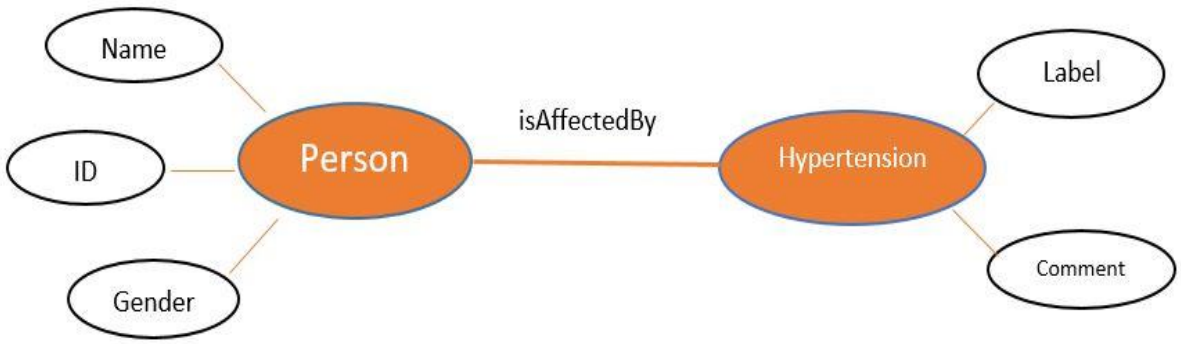


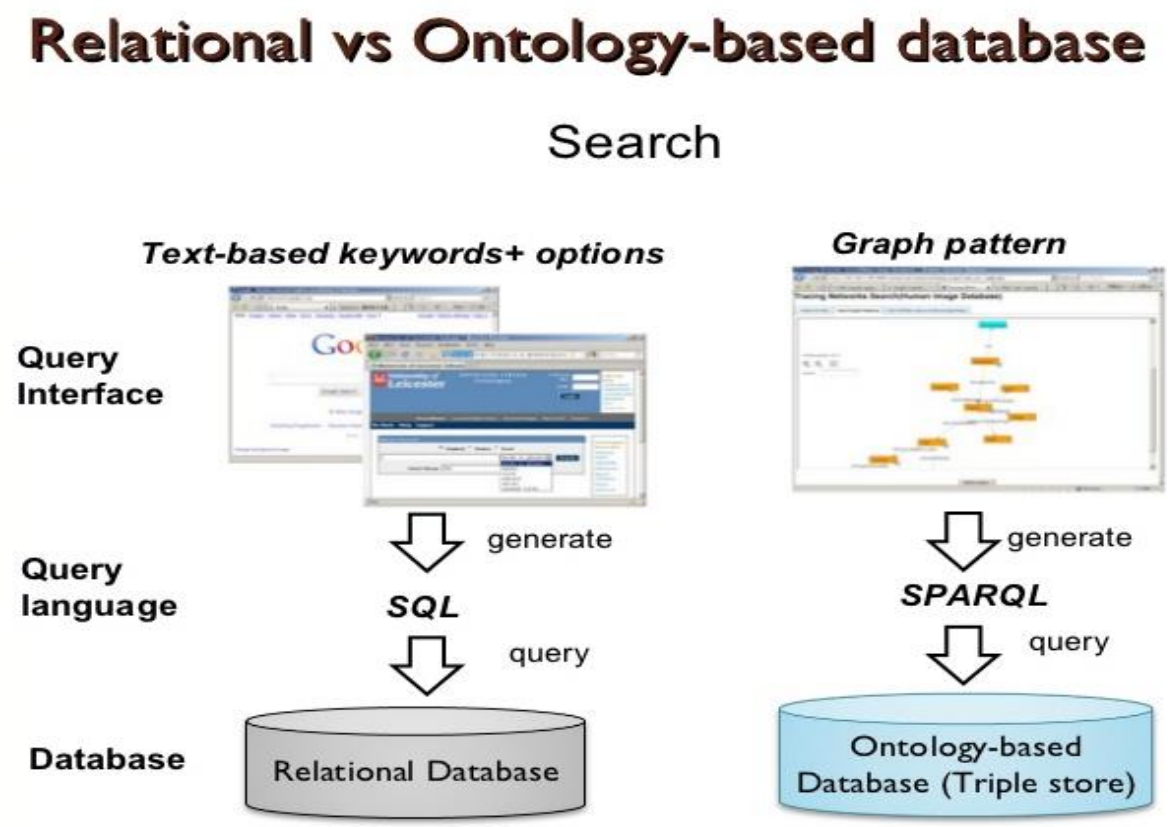
Fig.5.1.4 – Graphical Structure of the Ontology

As shown in the figure above, Person and Hypertension are classes, isAffectedBy is the object properties and Name, Gender are the Data properties.

5.2 Difference between the three Structures

DB-Schema	Prolog Structure	Ontology
Focus on Data	Focus on Rules and Facts	Focus on Meaning
Can not infer new information semantically	Can infer new information semantically	Can infer new information semantically
Not reusable	Not reusable	Reusable
Difficult to understand in layman terms	Difficult to understand in layman terms	Easy to understand in layman terms
DB is preferred when schema is small/simple/not used at query time	Prolog is preferred whenever we need to develop an AI system	Ontology is preferred when schema is large/complex/used at query time.
DB preferred when complete information is available e.g. Booking system		Ontology is preferred when it is not possible/reasonable to assume complete information e.g. Medical

5.3 Difference between Relational Database and Ontology



5.4 Basic Formal Ontology (BFO)

Basic Formal Ontology is an upper level ontology which is used by domain level ontologies for mapping. The main drawback of the domain level ontologies is that they are not dynamic, this means that the entities do not have constraints for time, space and theme. To remove this passive state of the domain level ontology, we map it to an upper level ontology, BFO, hence making it dynamic.

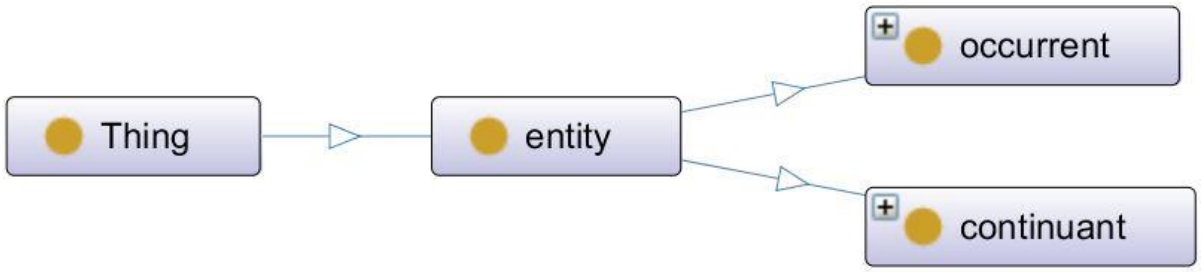


Fig. 5.4.1 – Top Hierarchy of BFO

The class on top of the hierarchical structure of any ontology is always the Thing class, and every class is the subclass of the Thing class. In BFO, one other class is also included just below the Thing class, which is the Entity class. The Entity class has no other sibling and is second in the hierarchical structure.

The Entity class is then classified into two classes as shown in the above figure

- Occurent
- Continuant

Continuant is something existing at an instant in time e.g. a person or a disease.

Occurrent is something existing through a span of time. E.g. human life, diagnostic session.

Basic Formal Ontology defines a basic structure of classes and properties. This structure is then used by domain level ontologies to enhance their structure and follow a predefined complete structure. The figure on the next page illustrates hierarchical structure of the BFO 2.0.

BFO also has benefits regarding to time, space and theme of any ontology. These benefits are depicted pictorially on next page.

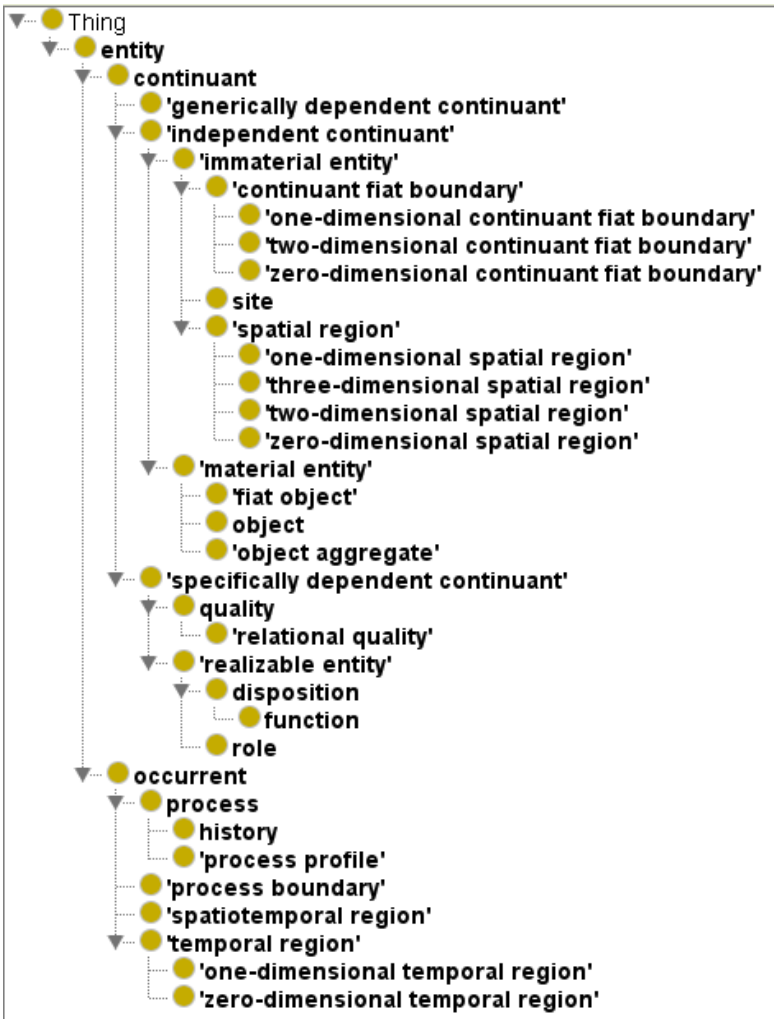


Fig. 5.4.2 – Visualization of BFO hierarchy

Fig. 5.4.3 – Special constraints of BFO

CHAPTER 6

DATA MODEL AND QUERYING

6.1 Proposed Data Model

The proposed data model has been constructed from the Handbook of Geriatrics. All the diseases and their symptoms have been placed under various classes as specified in the BFO.

The diseases as shown in the figure below have been placed inside the continuant class, because a disease is an entity that exists in particular time. The diseases do not span over a period of time. Under continuant class these are kept under specifically dependent continuant class, this is because these diseases depend upon one or more independent continuant. Now, under specifically dependent continuant the diseases are kept under the realizable entity class, this is because they are realized onto the independent continuant class. Finally, these are kept under the disposition class, because these diseases are triggered in the independent continuant class.

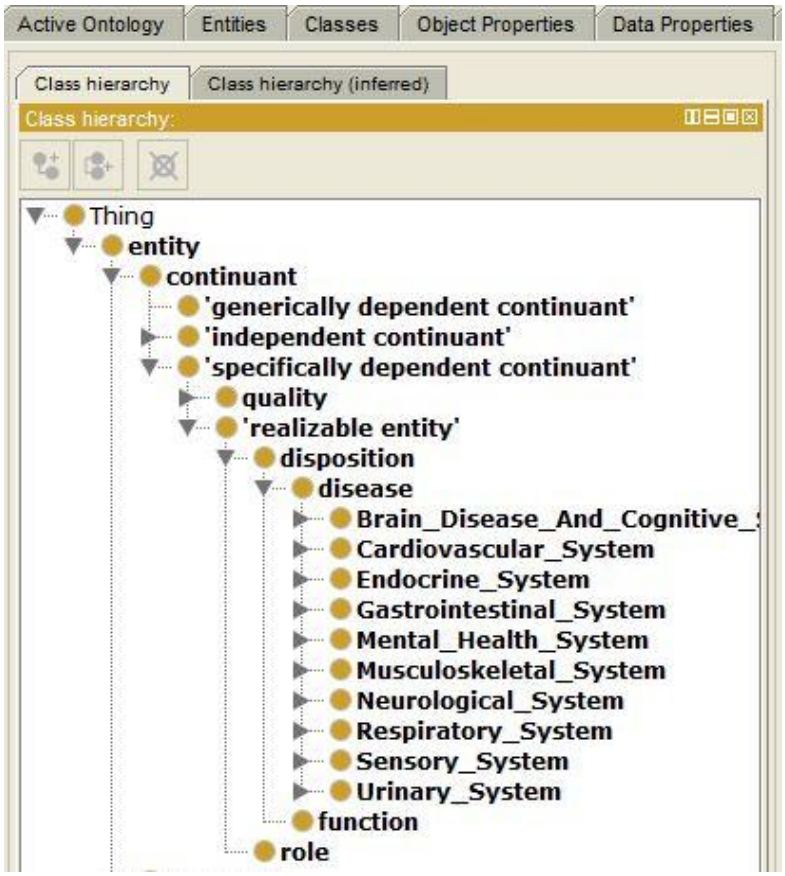


Fig. 6.1.1 – Diseases in the Ontology

The symptoms are the type of entity which might span over a period of time, hence they are kept under the occurent class. The occurent class has another subclass process, and under this class the symptoms are arranged.

The symptoms are further divided into two classes

- Common symptoms
- Specific symptoms

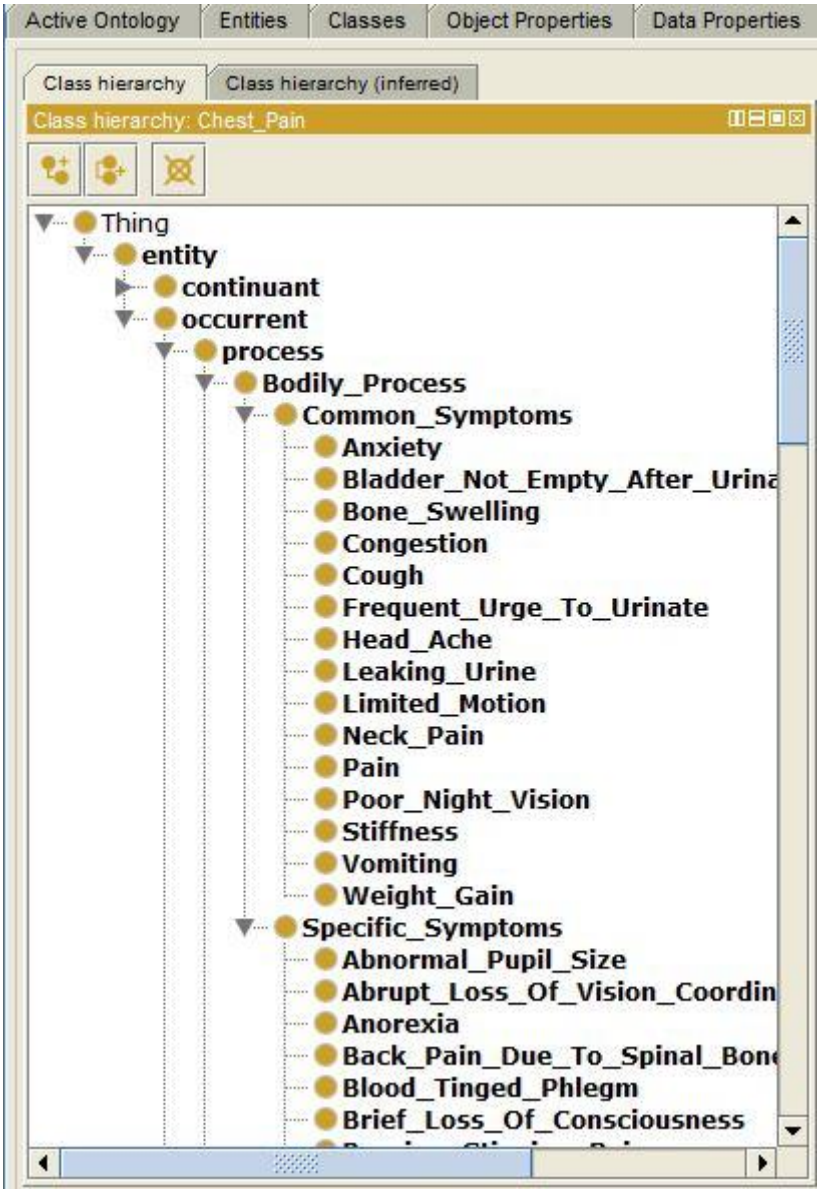


Fig. 6.1.2 – Symptoms in the Ontology

The organs of the body are kept under the continuant class because they exist at a point of time. Furthermore these organs are kept under the independent continuant class as the organs are independent entities.

Finally the organs are classified under the material entity class and not in the immaterial entity class. This is because the organs are specific things and occupy some space in the human body.

The figure below shows the hierarchy of the organs and their details such as annotations and object property assertions.

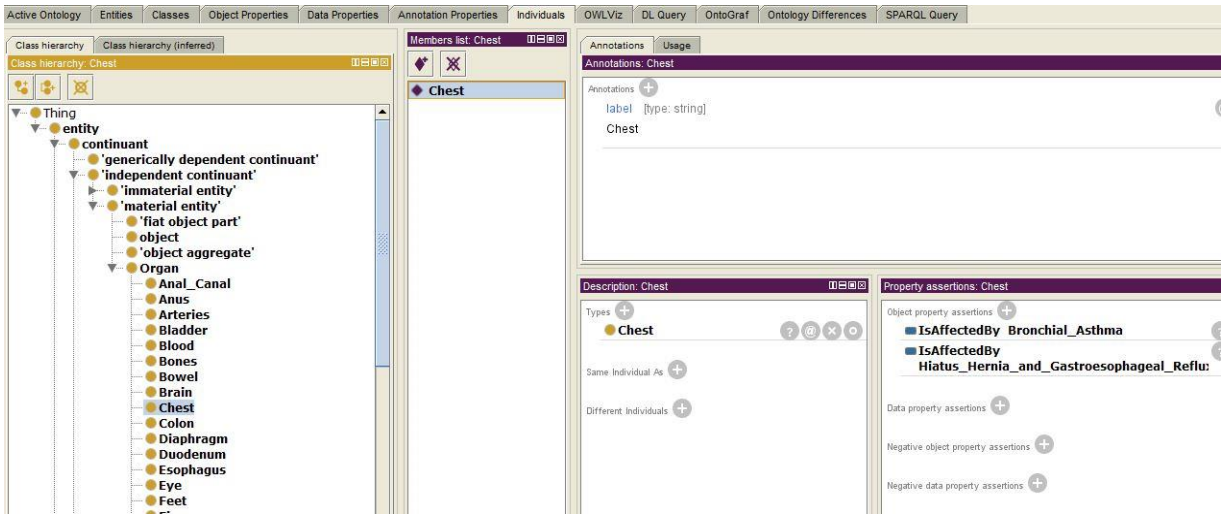


Fig. 6.1.3 – Organs in the Ontology

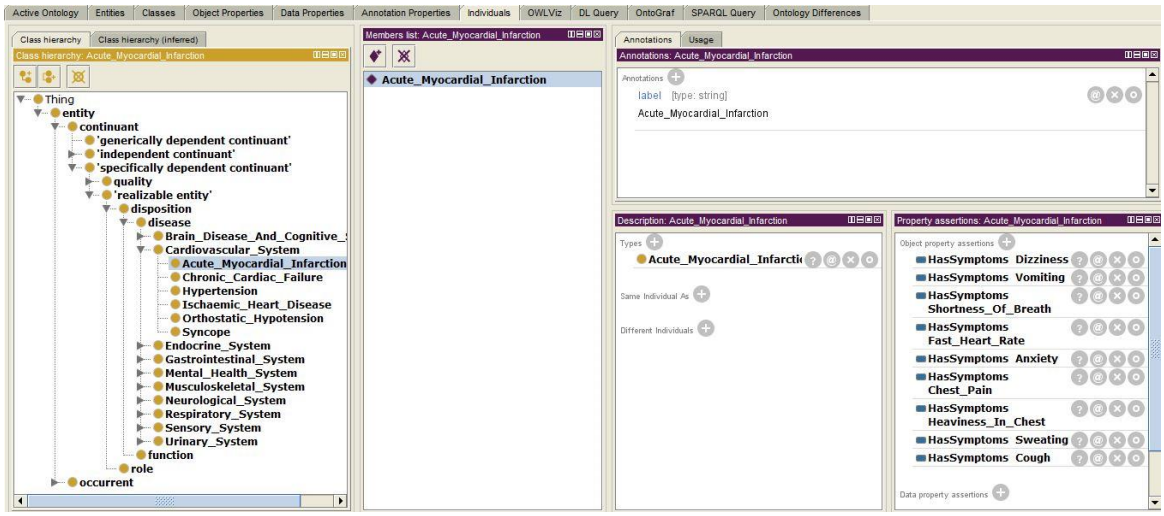


Fig. 6.1.4 – Details of the Diseases

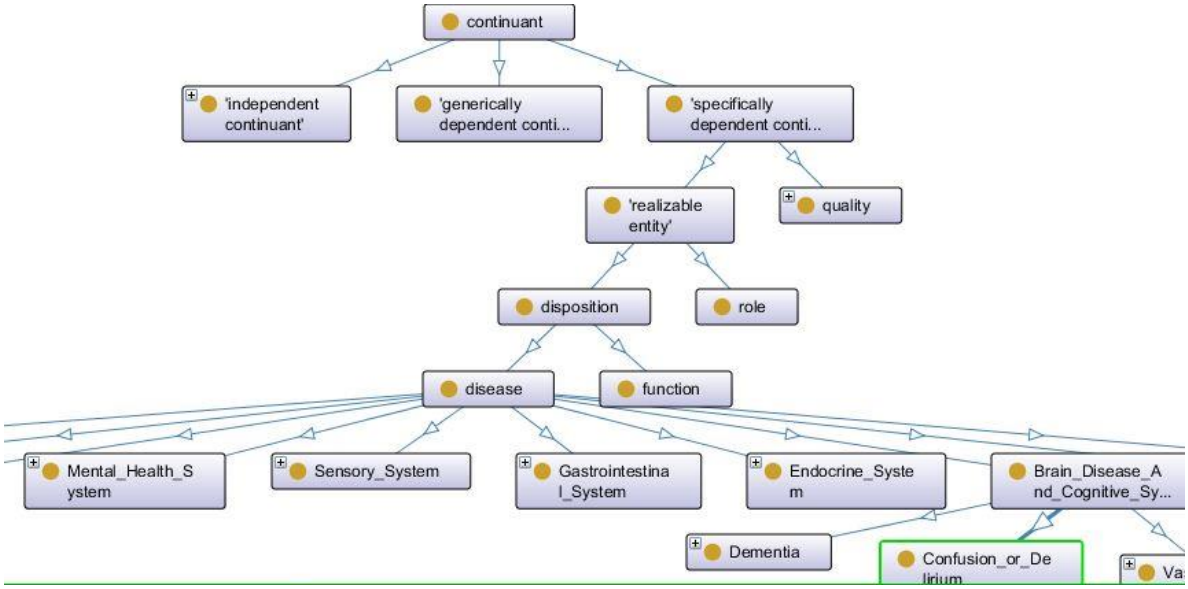


Fig. 6.1.5 – Hierarchy of the continuant class

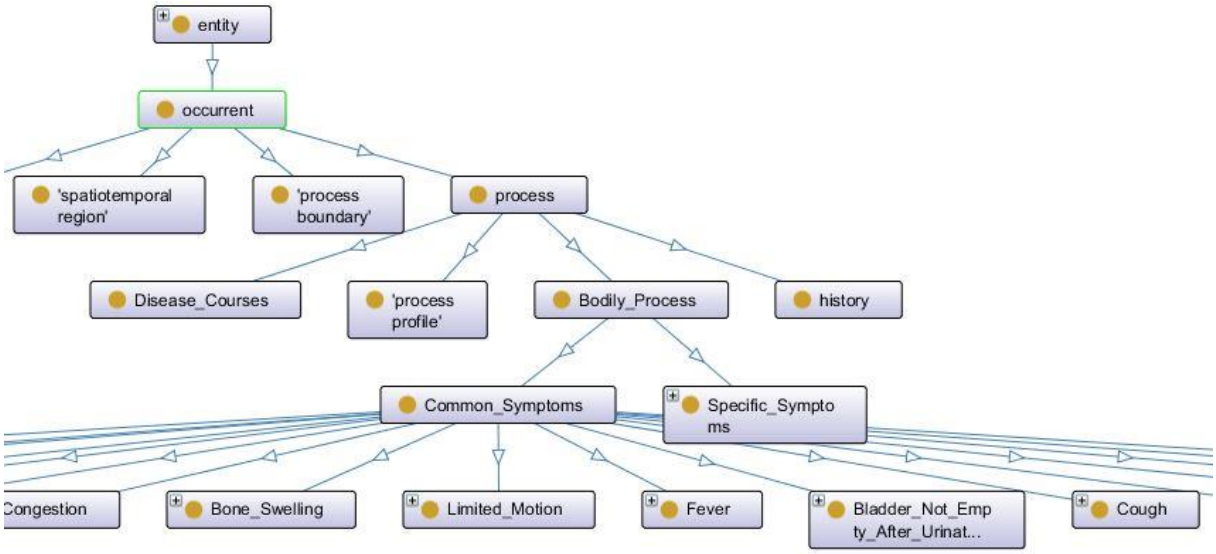


Fig. 6.1.6 – Hierarchy of the occurrent class

The object properties in the ontology are used to create a link between classes. These links serve as predicate between the object and the subject.

The object properties used in the proposed ontological framework are shown in the figure below. These object properties are classified into different classes based on their functions.

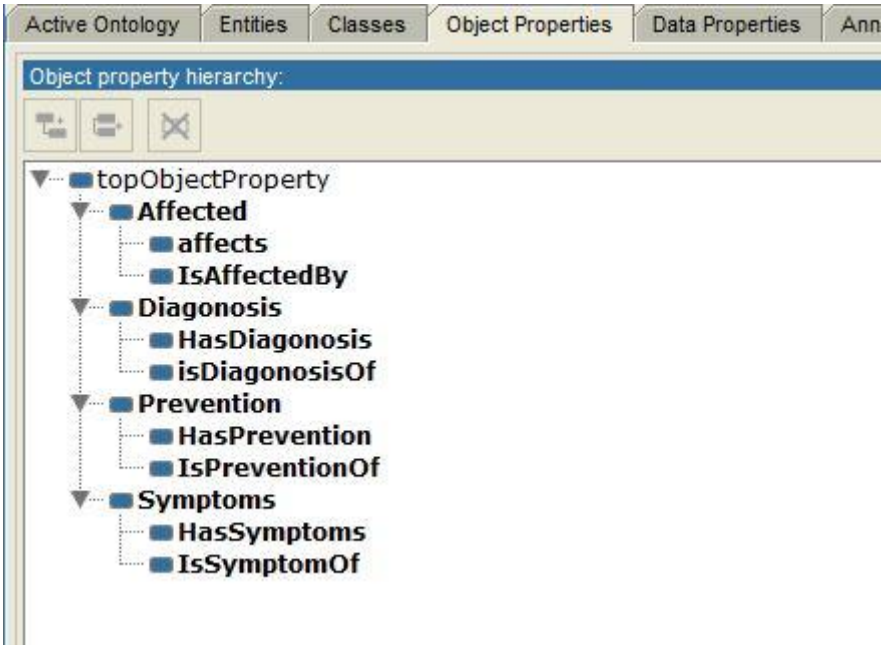


Fig. 6.1.7 – Object Properties in the Ontology

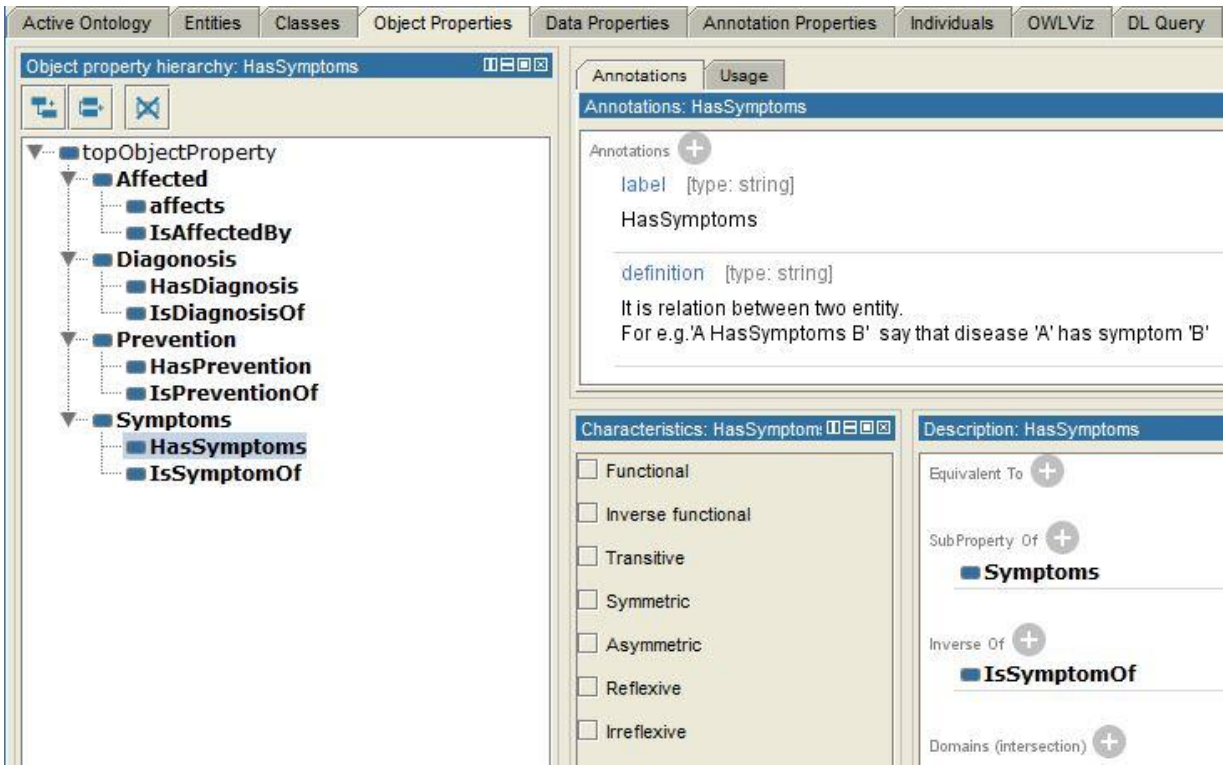


Fig. 6.1.8 – Details of Object Properties

The Data Properties in the ontology are used to define the constant values for individuals. These can be labels, names, definitions and other annotations required by the individuals.

The figure below shows the data properties used in the proposed ontology and its properties as well.

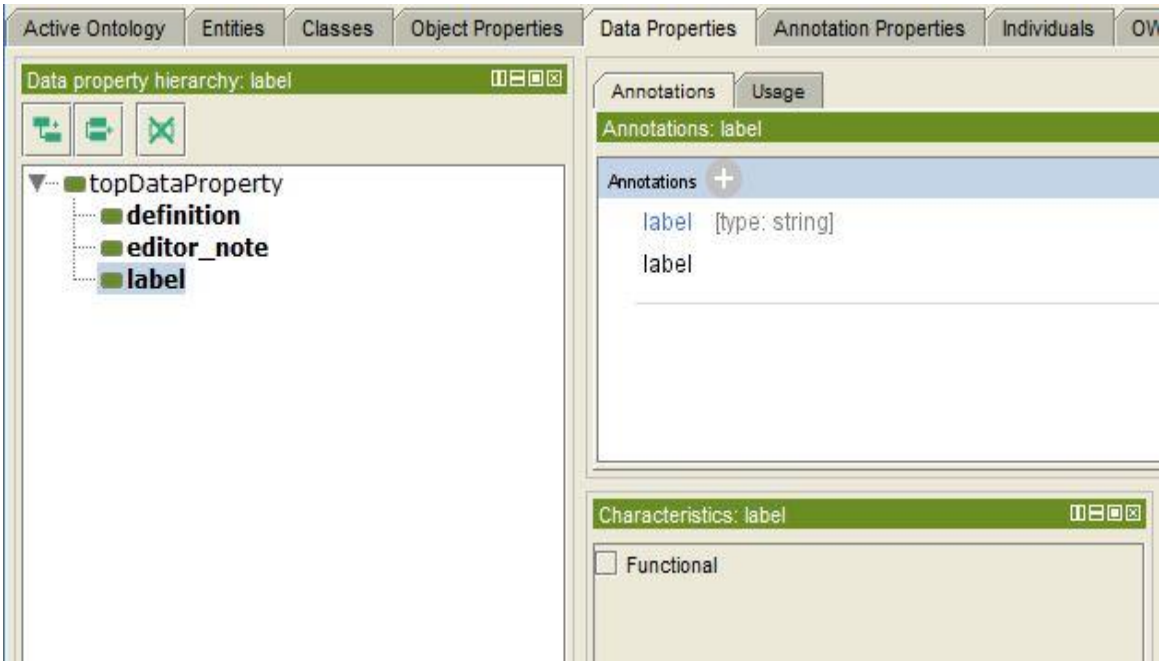


Fig. 6.1.9 – Details of Data Properties

6.2 Description Logic (DL) Queries

The proposed model thus created can now be used to extract information by using different querying methods. In this section Description Logic is used for querying purposes.

The following figures illustrate these queries and the results they show.

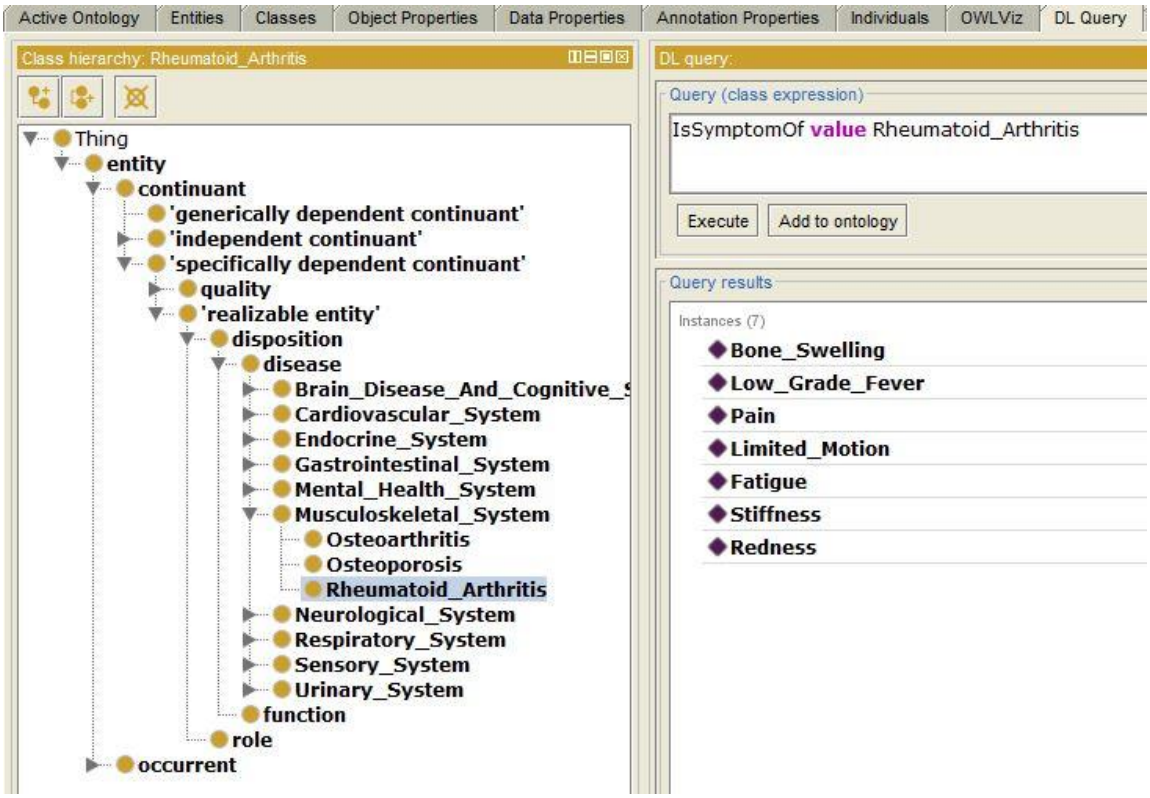


Fig. 6.2.1 – DL Query to identify symptoms of Rheumatoid Arthritis

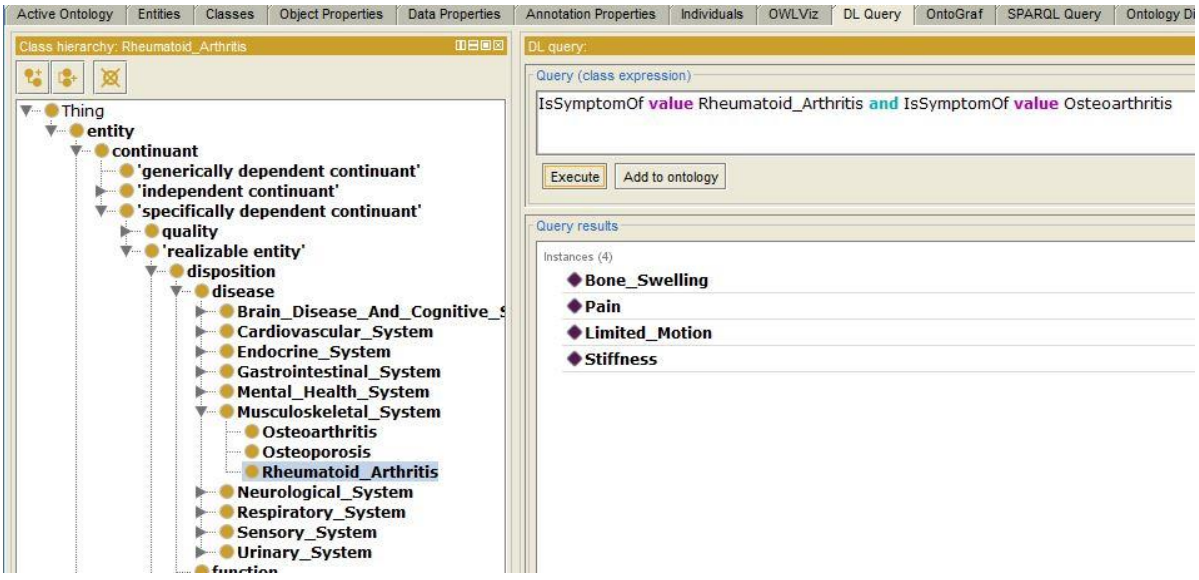


Fig. 6.2.2 – DL Query for common symptoms of Rheumatoid Arthritis and Osteoarthritis

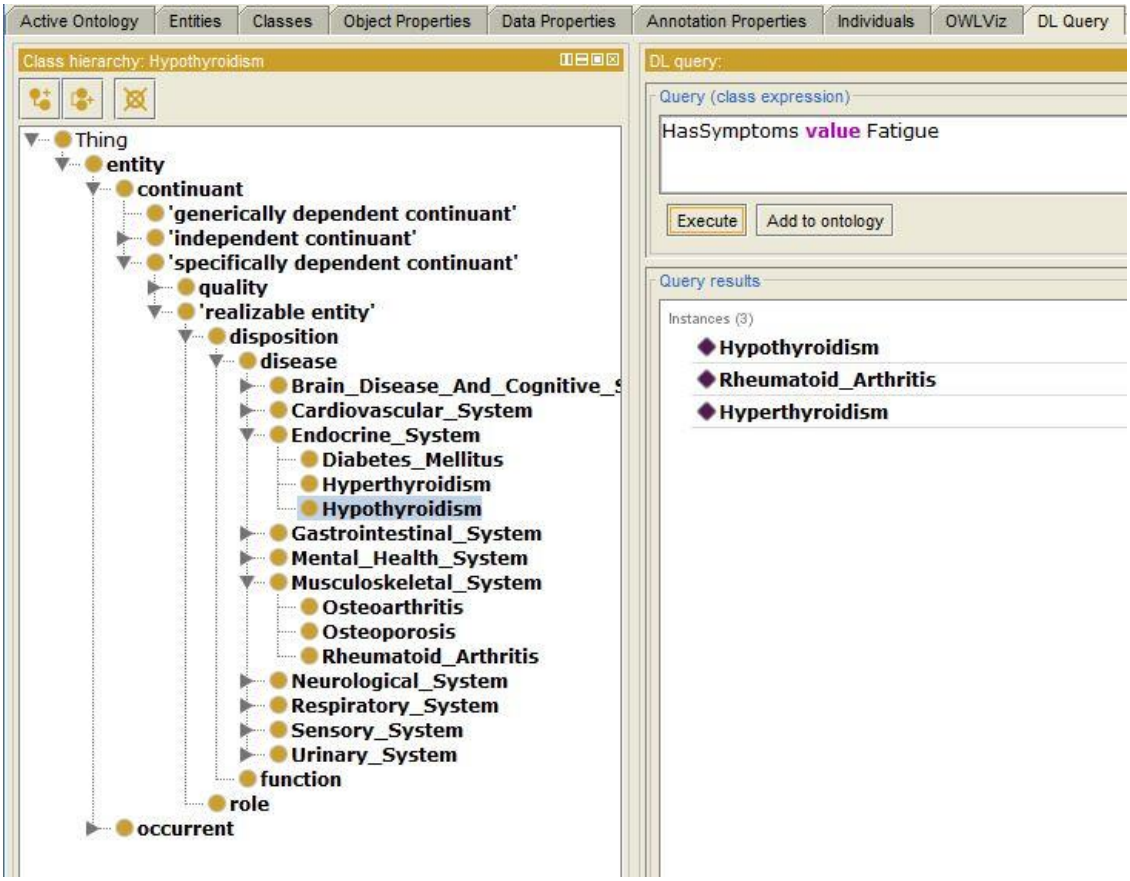


Fig. 6.2.3 – DL Query for Diseases with symptom Fatigue

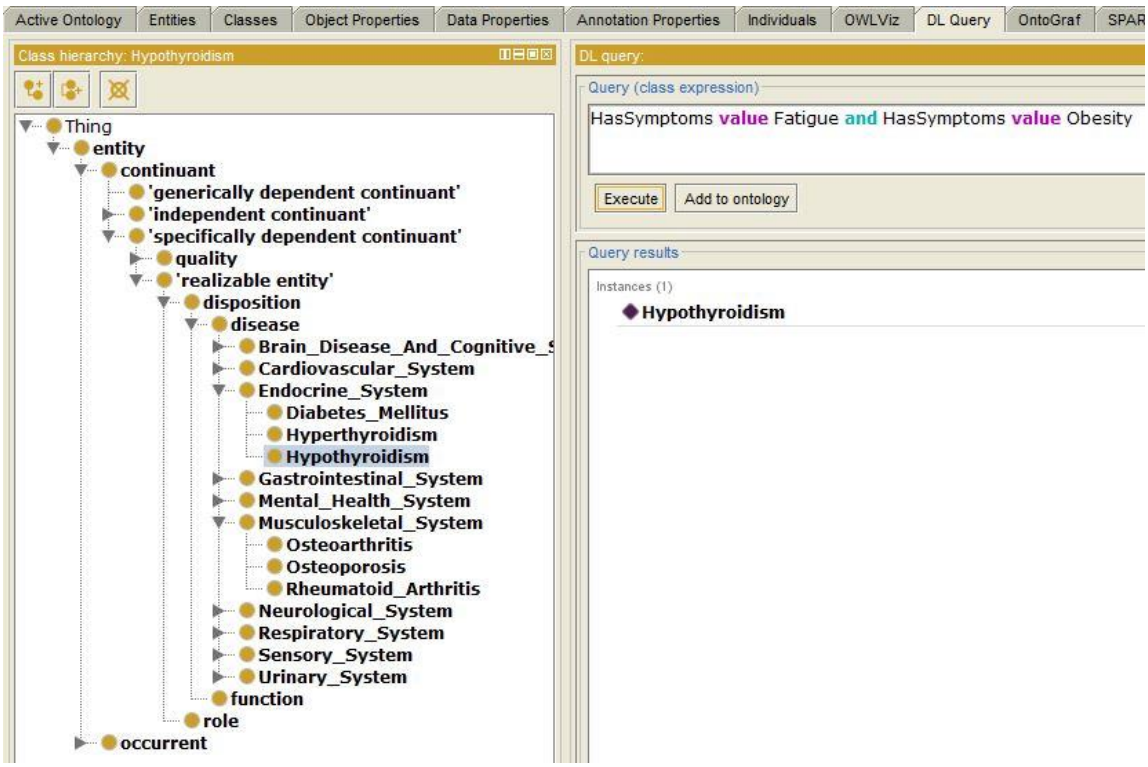


Fig. 6.2.4 – DL Query for Diseases with symptom Fatigue and Obesity

6.3 SPARQL Queries

The SPARQL queries provide a more robust way to query the ontology. The SPARQL query works on the first order predicate logic. First order logic quantifies the objects of a sentence, so that semantics of this sentence can be extracted.

In this querying methodology, the object, the predicate and the subject all are used to extract information from the ontology. And whichever of the three is left as a variable outputs the desired result.

The following figures illustrate the use of SPARQL queries and the results they output.

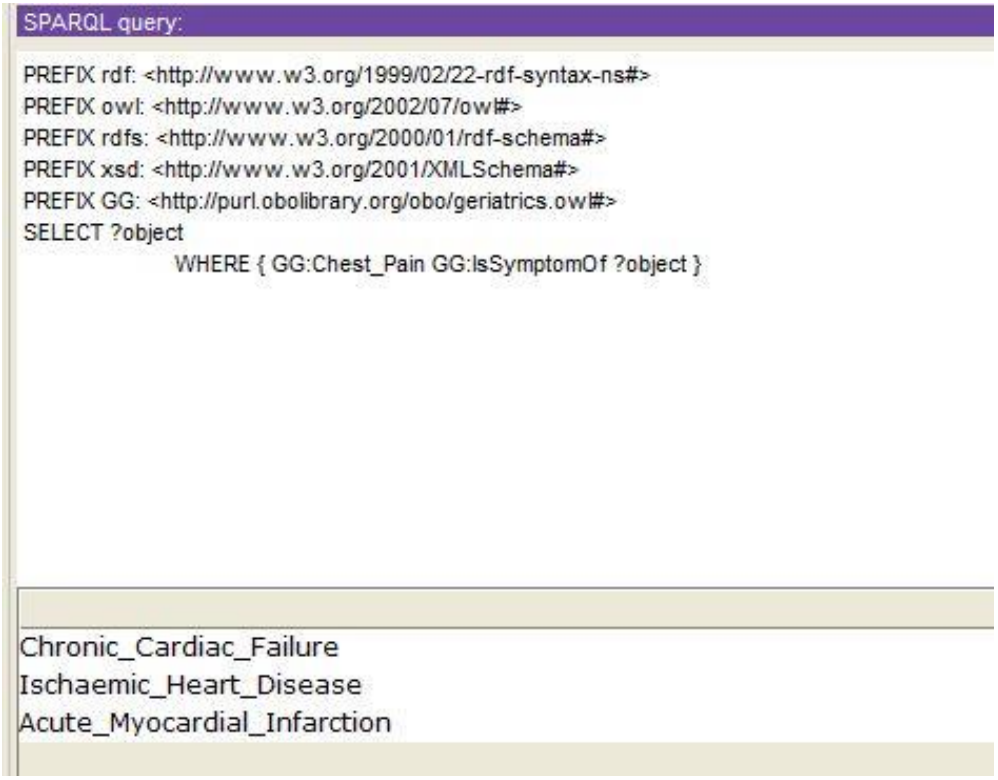


Fig. 6.3.1 – SPARQL Query for Diseases with symptom Chest Pain

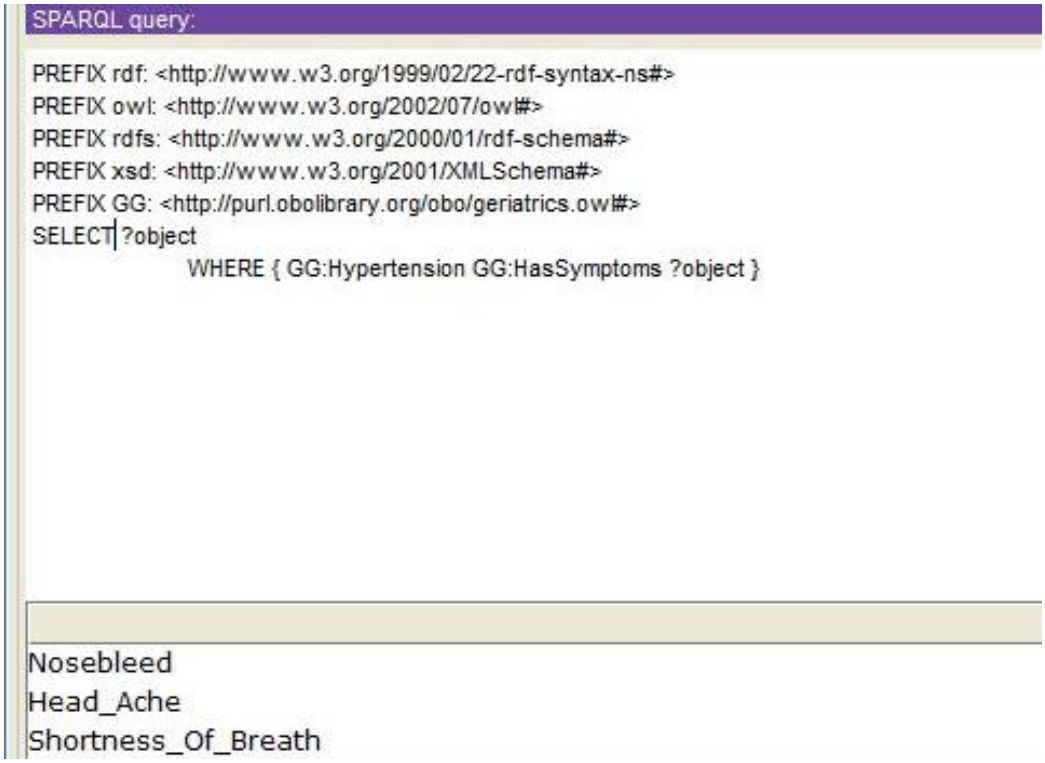


Fig. 6.3.2 – SPARQL Query for symptoms of Hypertension

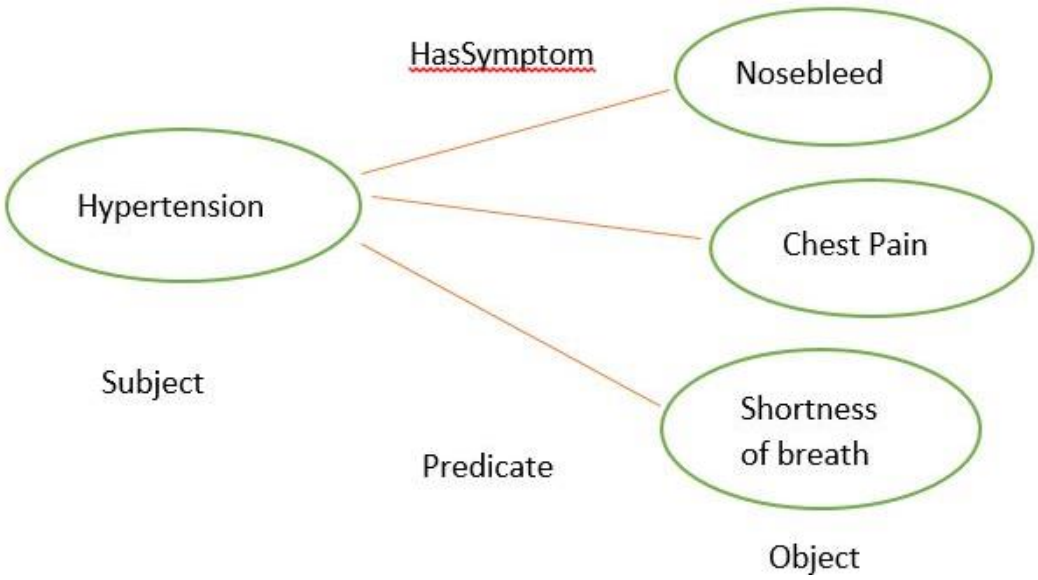


Fig. 6.3.3 – SPARQL Query for symptoms of Hypertension

The above two figures illustrate the functioning of the SPARQL query. As shown in the figure, Hypertension is the subject, HasSymptom is the predicate and Nosebleed is the object. Now in the query, the object is left as a variable and other two are specified, hence the output is all the symptoms of the disease Hypertension.

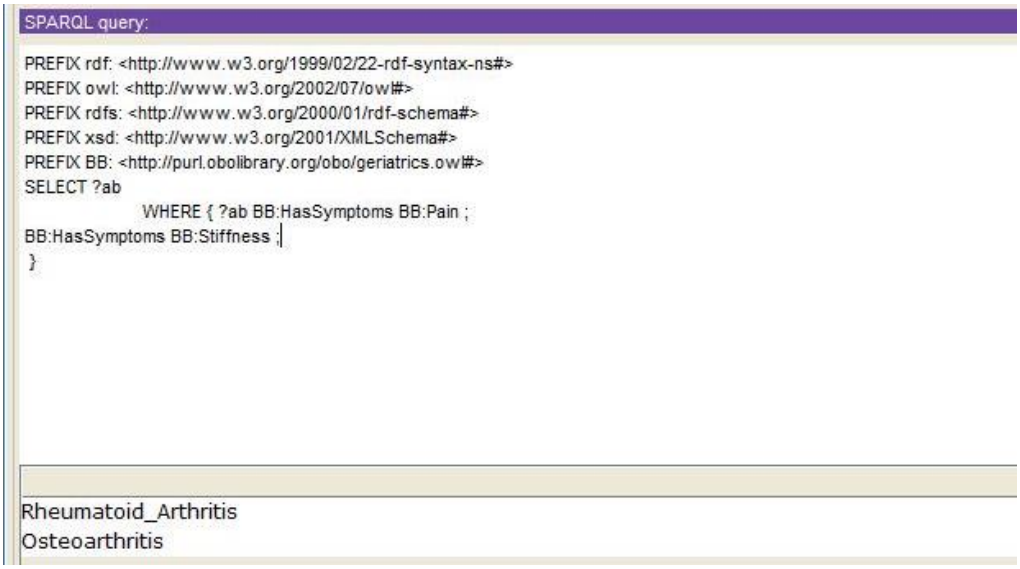


Fig. 6.3.4 – SPARQL Query for Diseases with symptom Pain and Stiffness

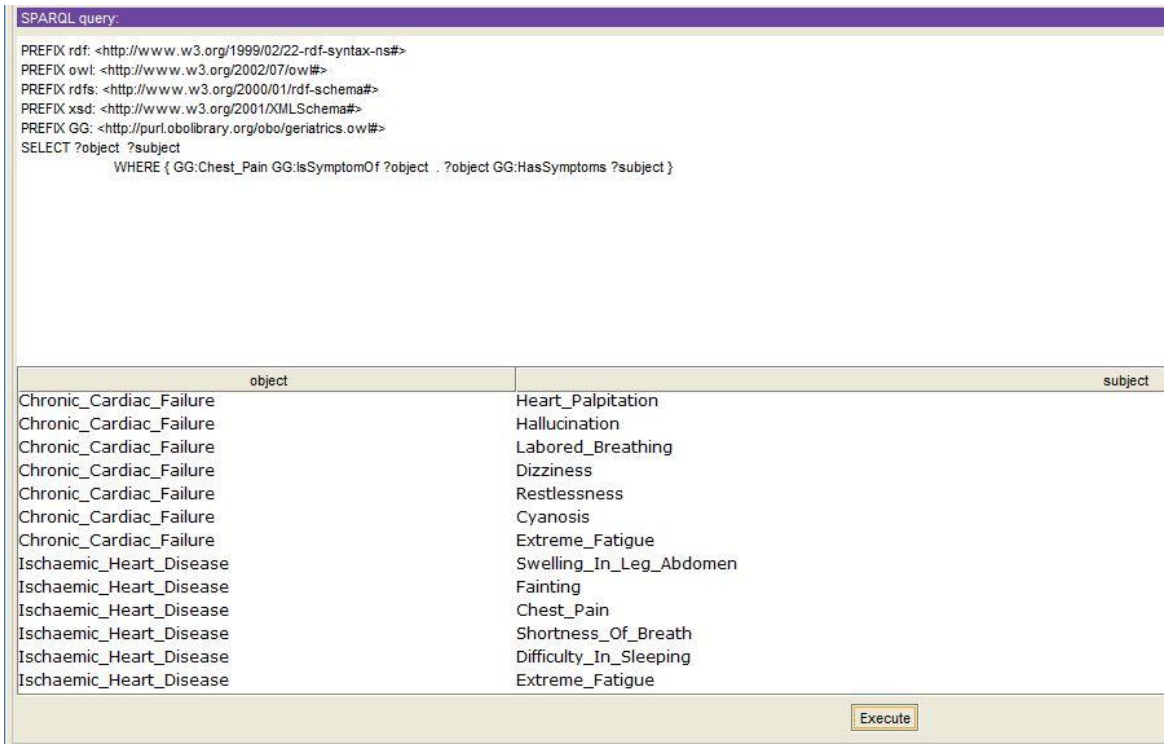


Fig. 6.3.5 – A complex SPARQL Query for Diseases with various symptoms

CHAPTER 7

WEB APPLICATION

Based on the proposed data model a web application is developed keeping the ontology as the backend database.

The web application of the Decision support system consists of three main parts. These are as follows

- Symptom Checker
- Disease Lookup
- Ontology Browser

The Symptom Checker is used to predict the probable diseases based on the symptoms entered by the user.

The Disease Lookup is the Knowledge base of all the diseases related to Geriatrics. Their information, symptoms and diagnosis are shown in this section.

The Ontology Browser is an embedded browser for the ontology created. This is based on the jOWL library of JavaScript.

The following snapshots show the working and design of the web application.

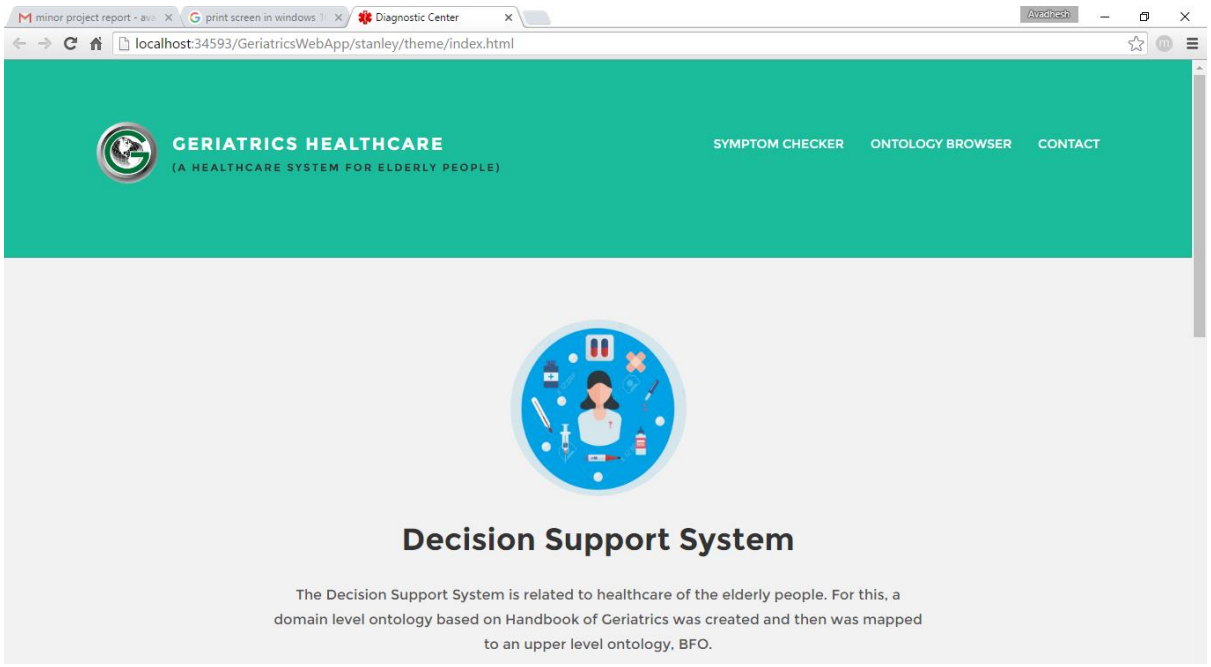


Fig. 7.1 – Landing Page of the Web Application

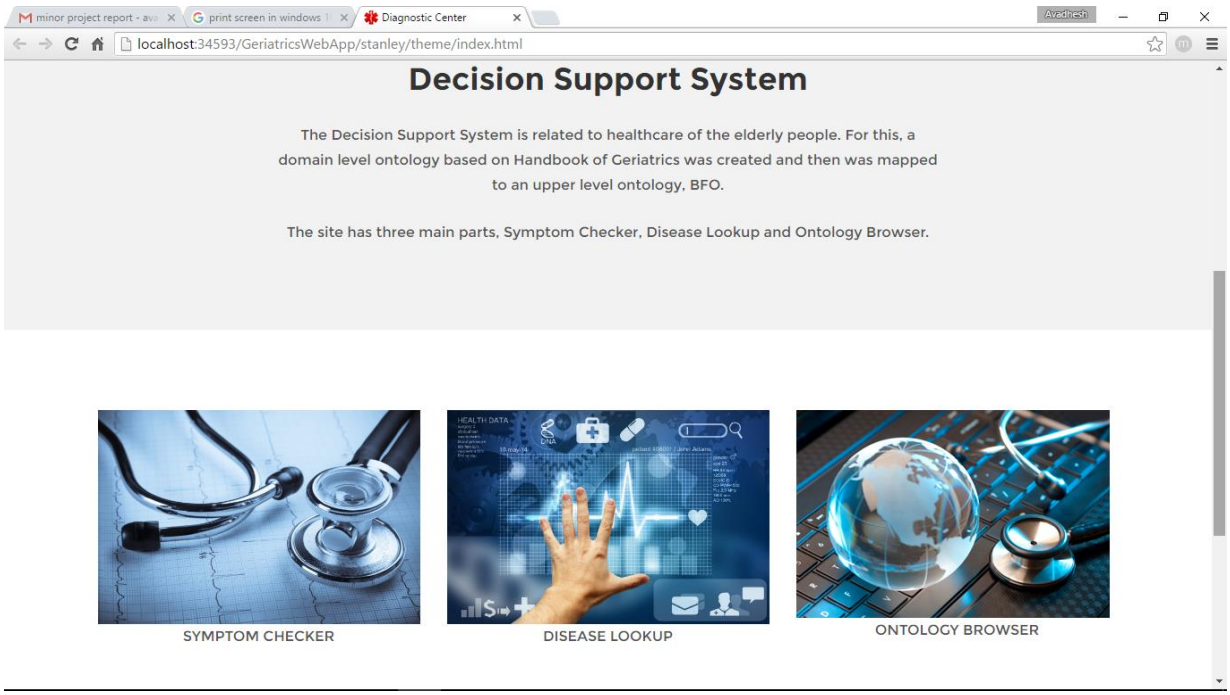


Fig. 7.2 – The three parts of the Web Application

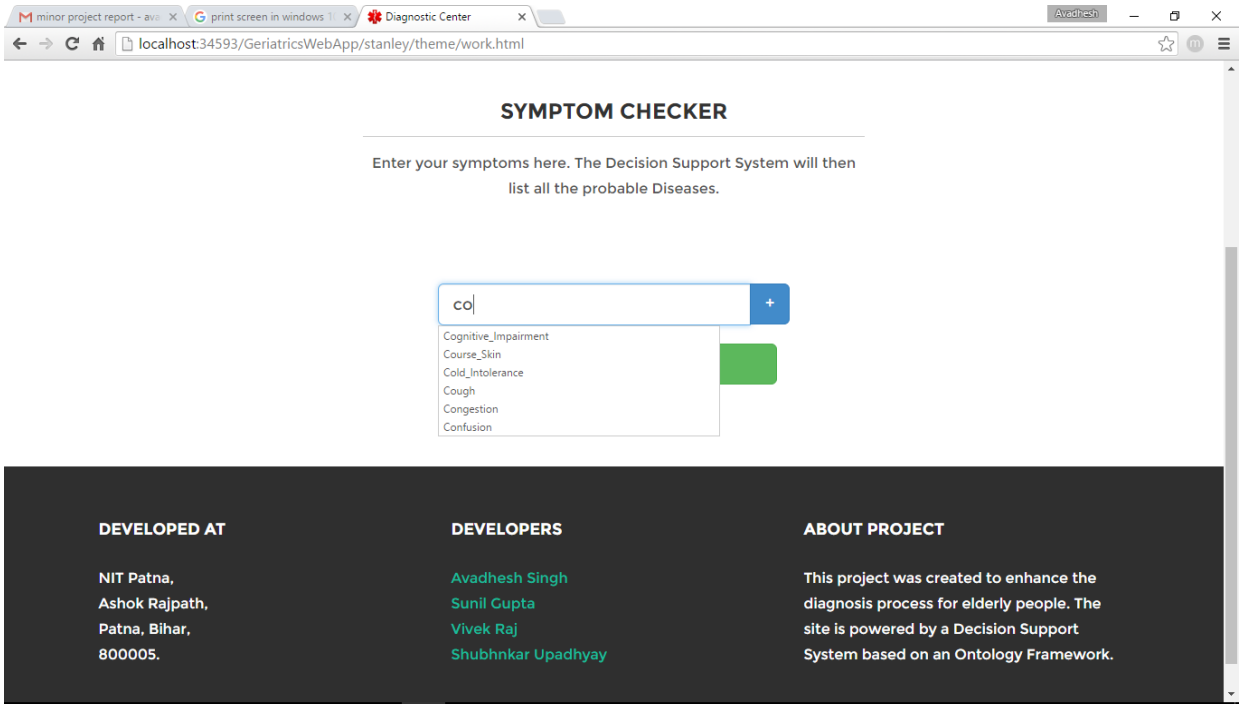


Fig. 7.3 – Symptom Checker of the Web Application

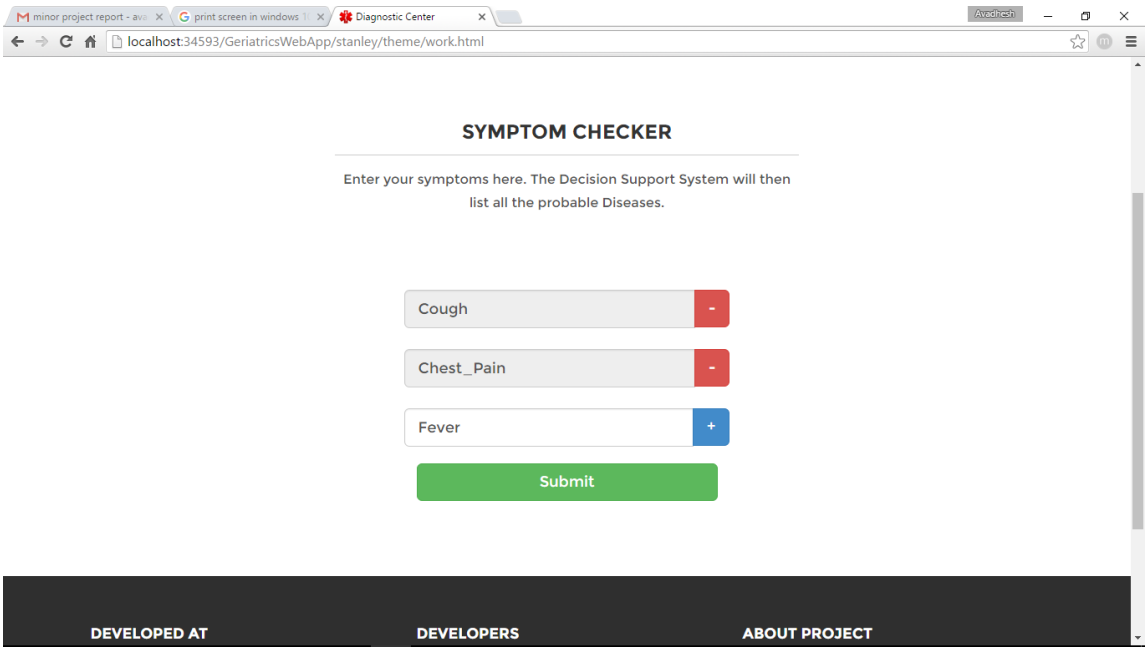


Fig. 7.4 – Symptom Checker of the Web Application

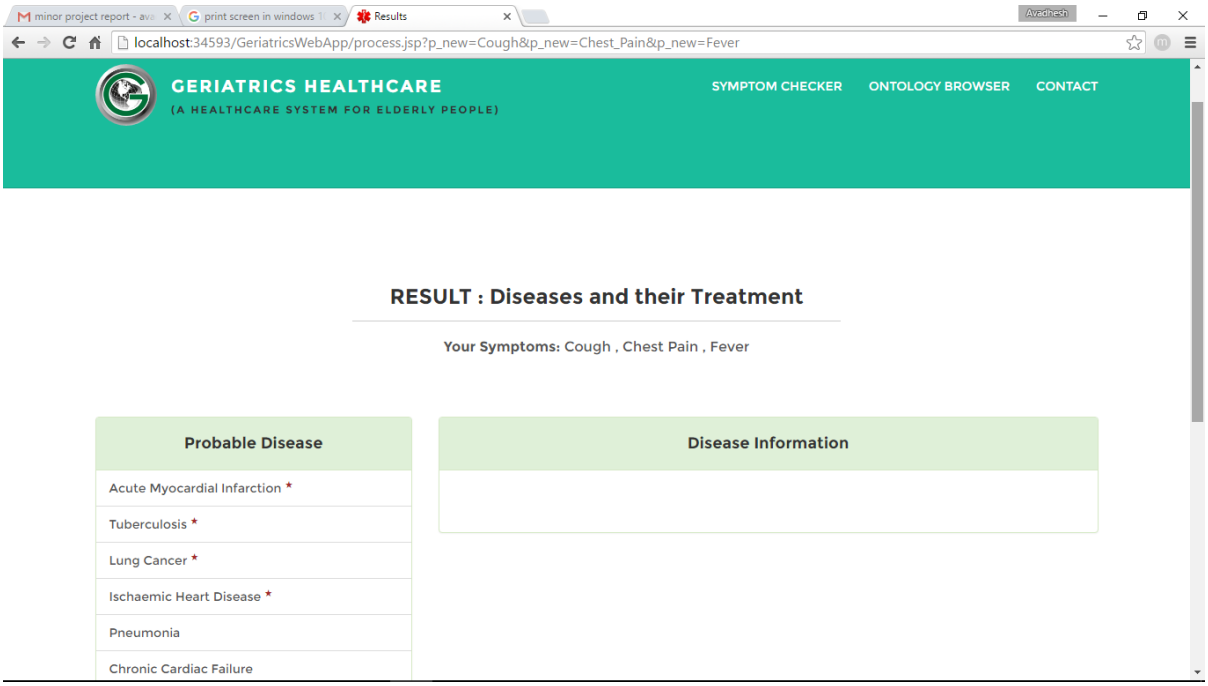


Fig. 7.5 –Result of Symptom Checker based on the symptoms

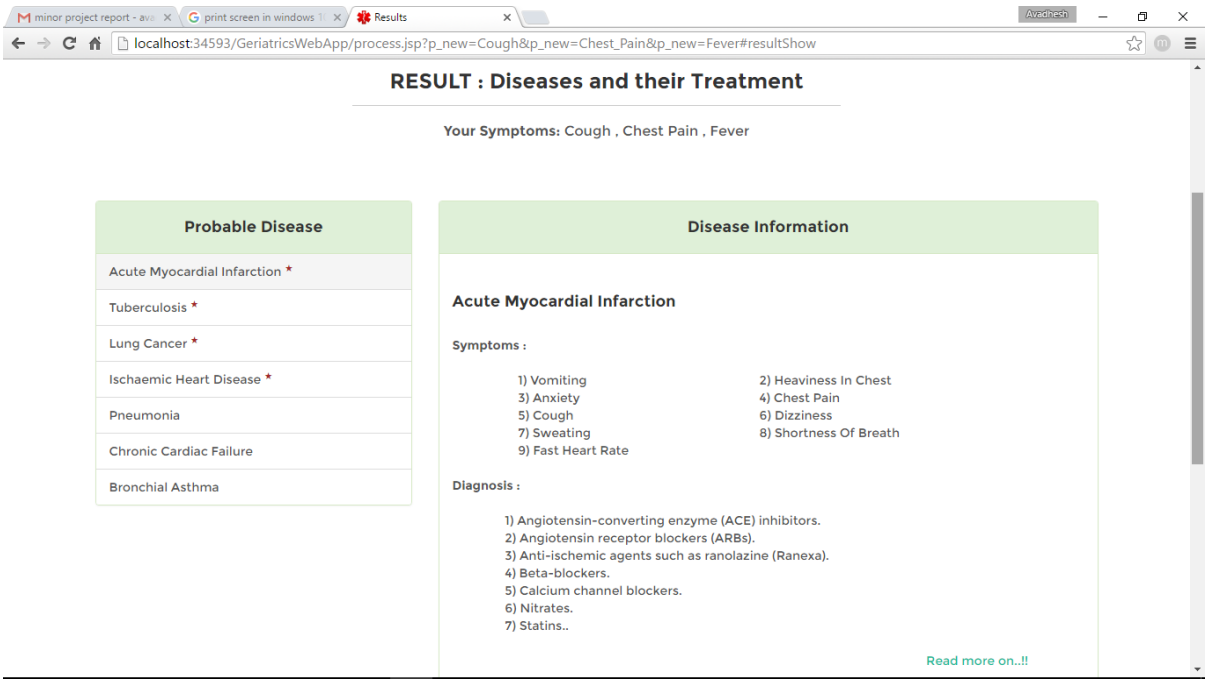


Fig. 7.6 –Information related to the probable disease

Fig. 7.7 –Further information related to the probable disease

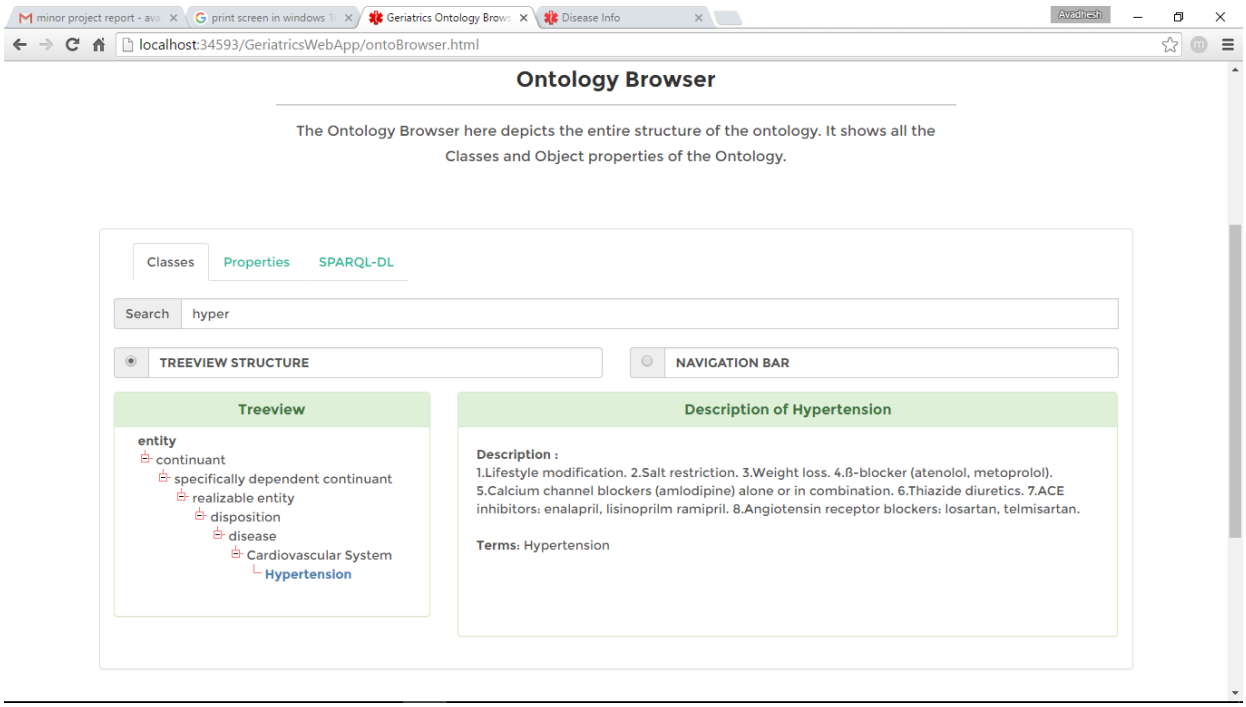


Fig. 7.8 – Ontology Browser of the Web Application

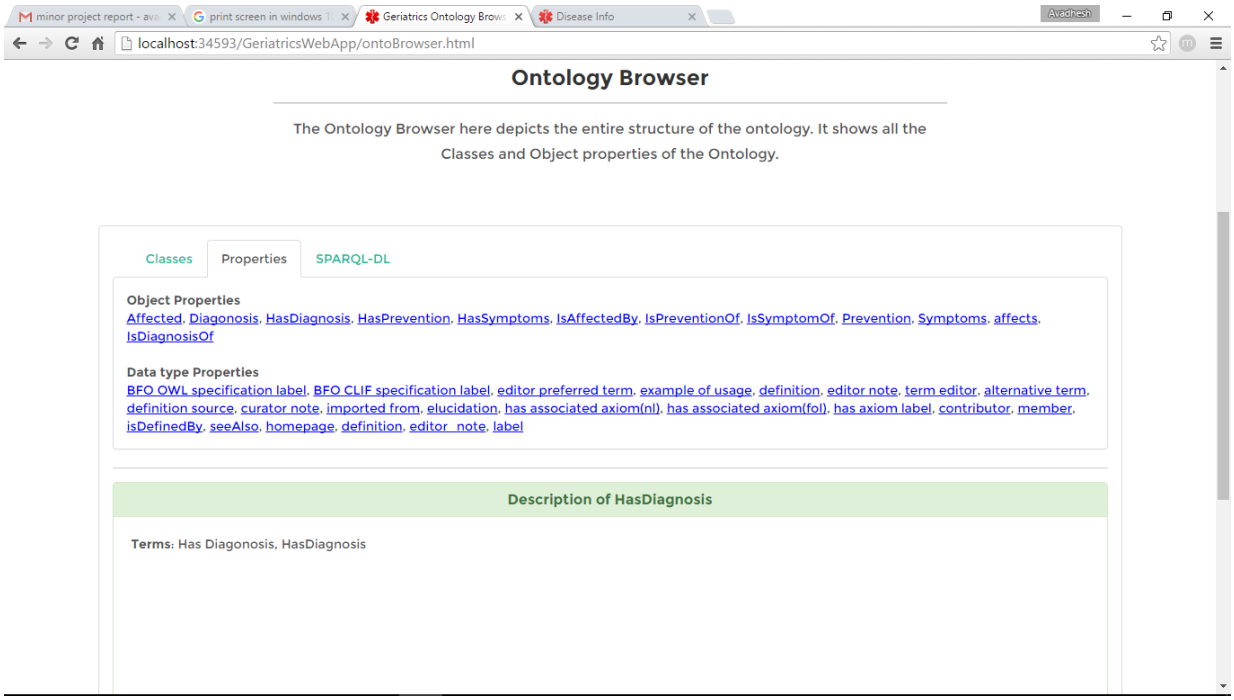


Fig. 7.9 – Ontology Browser of the Web Application

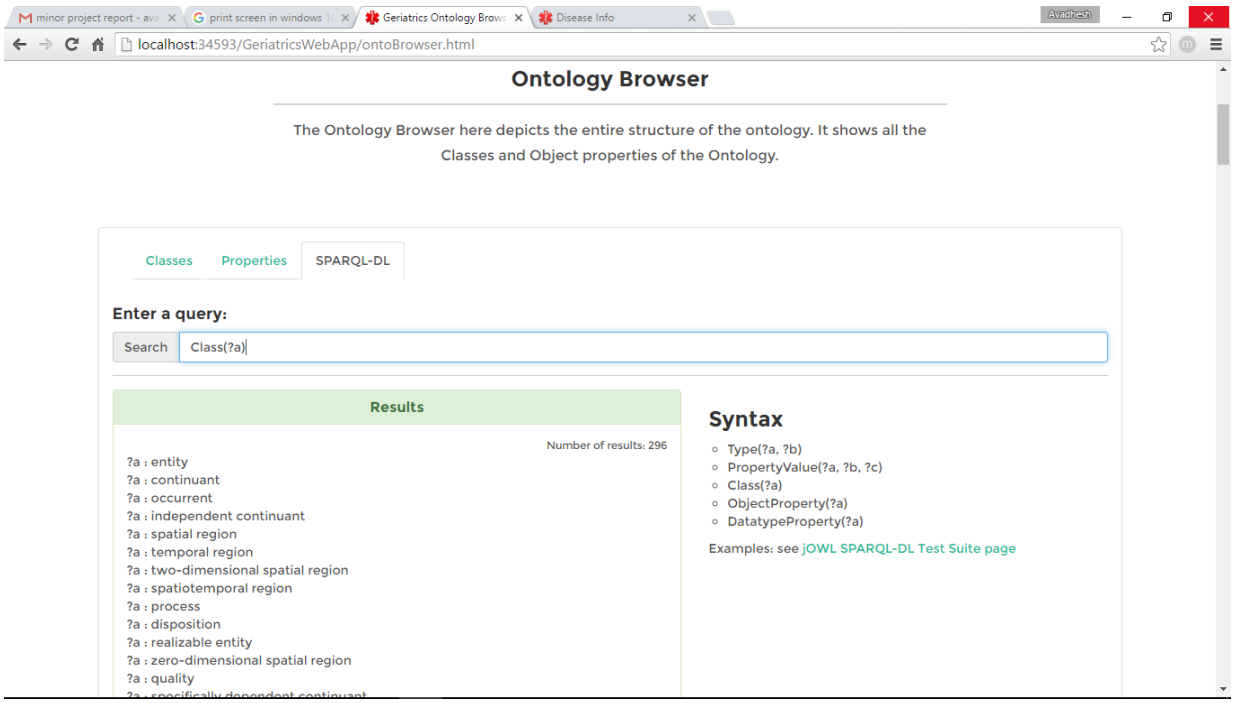


Fig. 7.10 – Ontology Browser of the Web Application

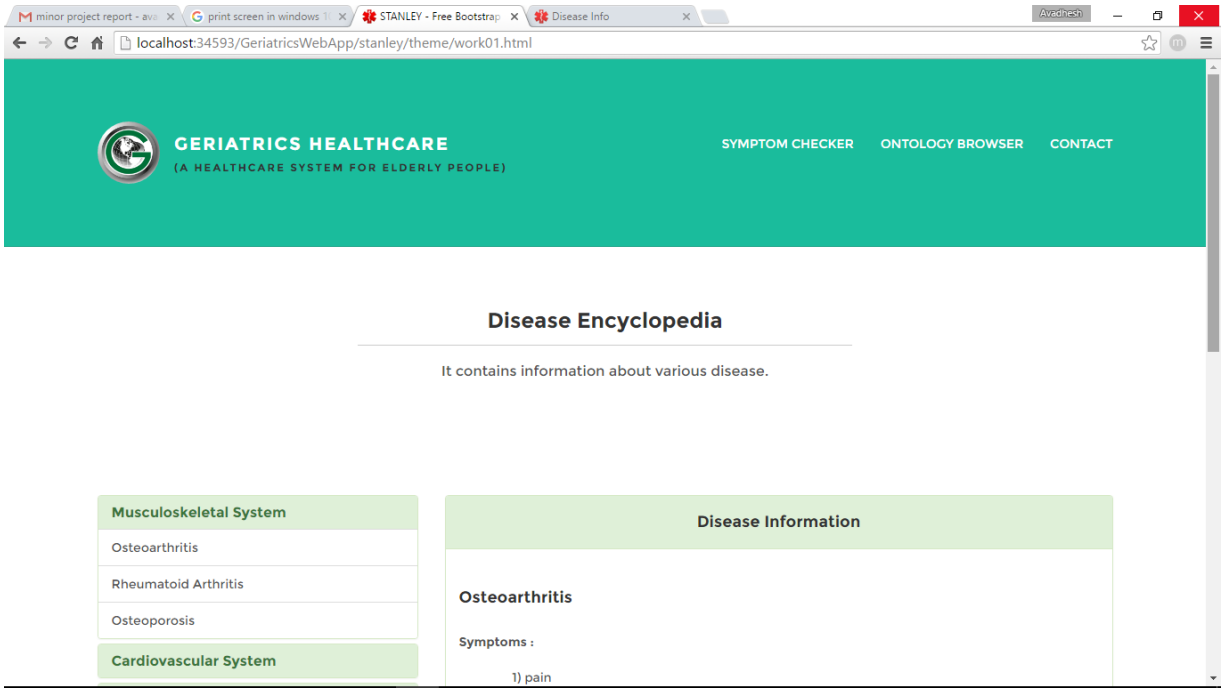


Fig. 7.11 – Disease Encyclopaedia of the Web Application

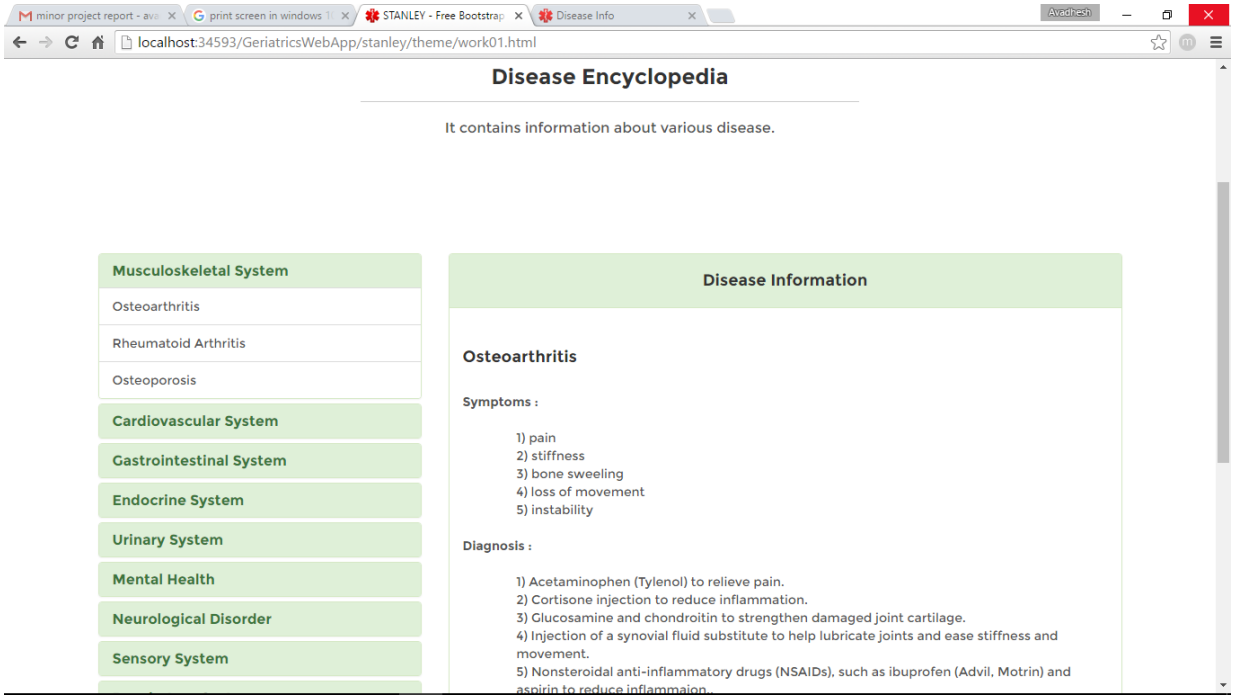


Fig. 7.12 – Disease Encyclopaedia of the Web Application

CHAPTER 8

ANDROID APPLICATION

8.1 Control flow of the Application

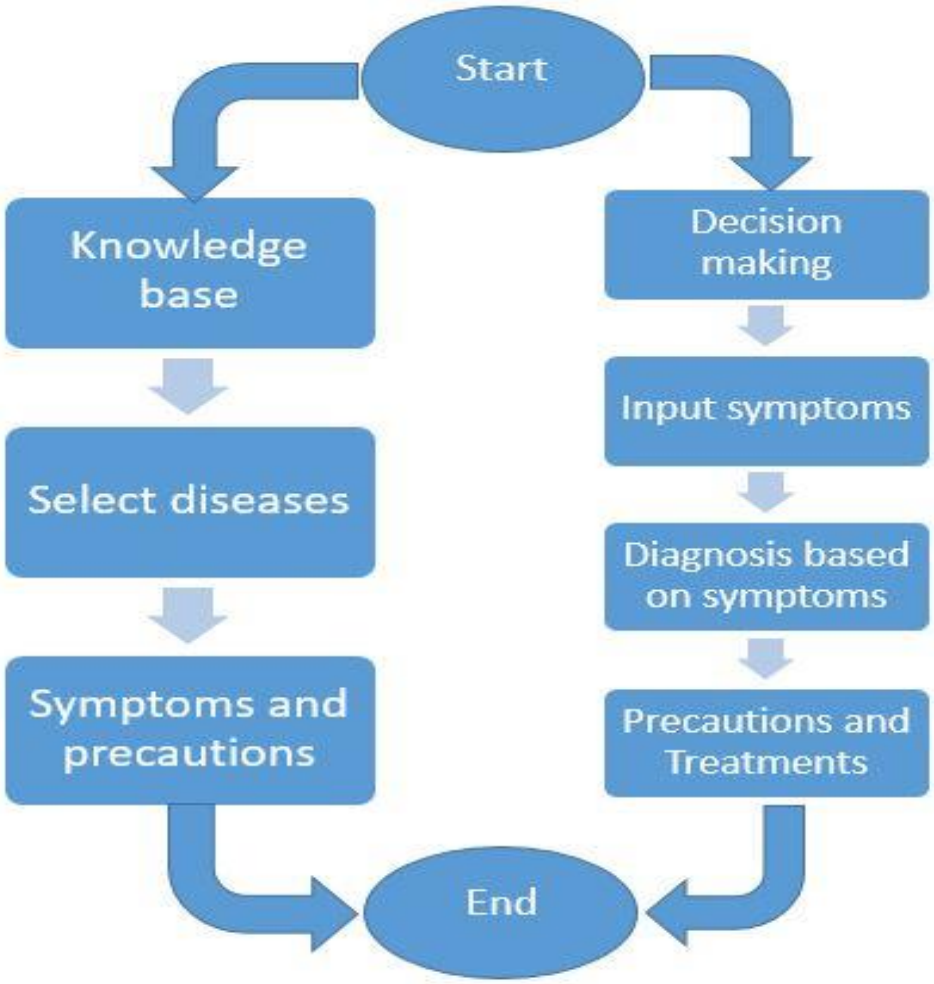


Fig. 8.1.1 – Control flow for the Application

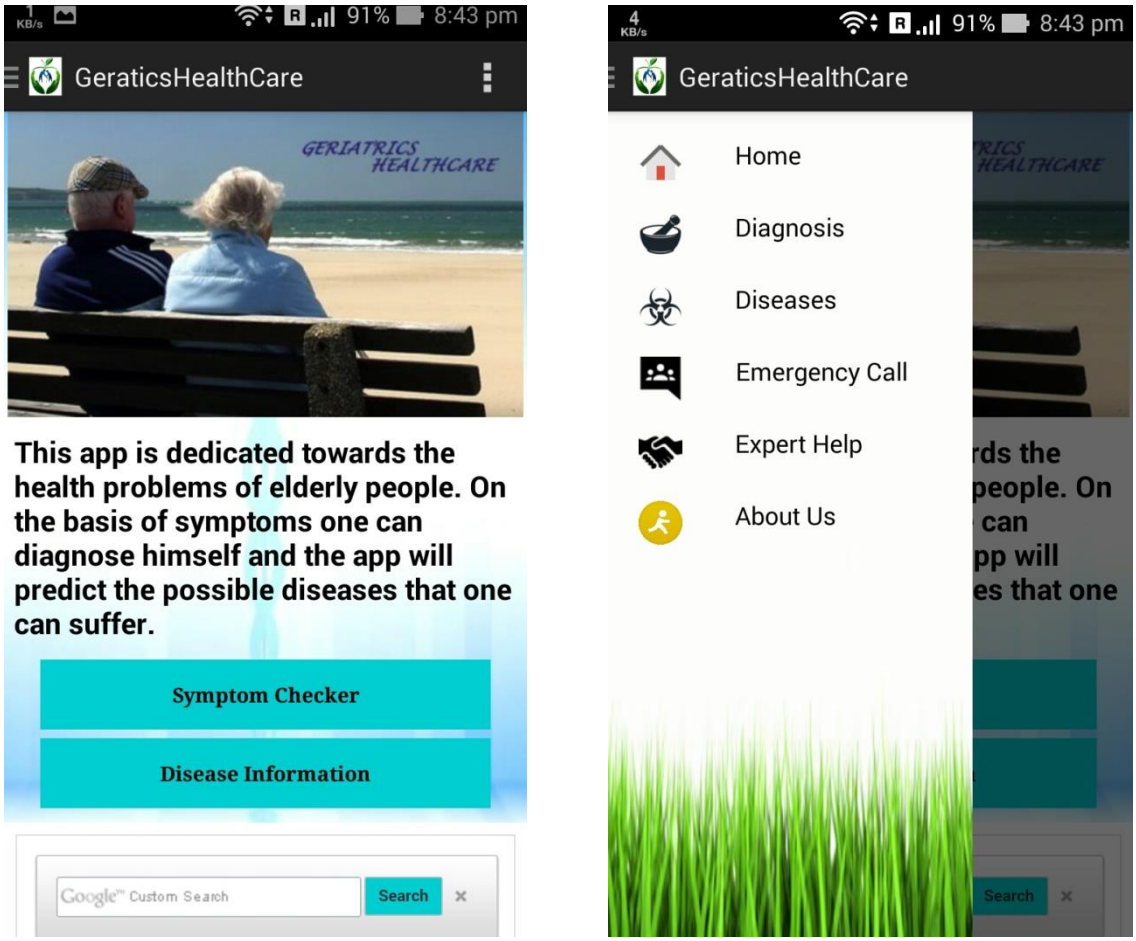
The above figure illustrates the control flow of the Android application. The application is divided into two parts

- Knowledge base part
- Decision making part

8.2 Knowledge base

The Knowledge Base part consists of all the diseases included in the Handbook of Geriatrics. The hierarchical structure is developed and is used for information retrieval of various diseases. It acts as a data store for displaying data on various diseases.

Following figures are various screenshots of the Android Application.



(a) (b)

Fig. 8.2.1 (a,b) – Landing pages for the application

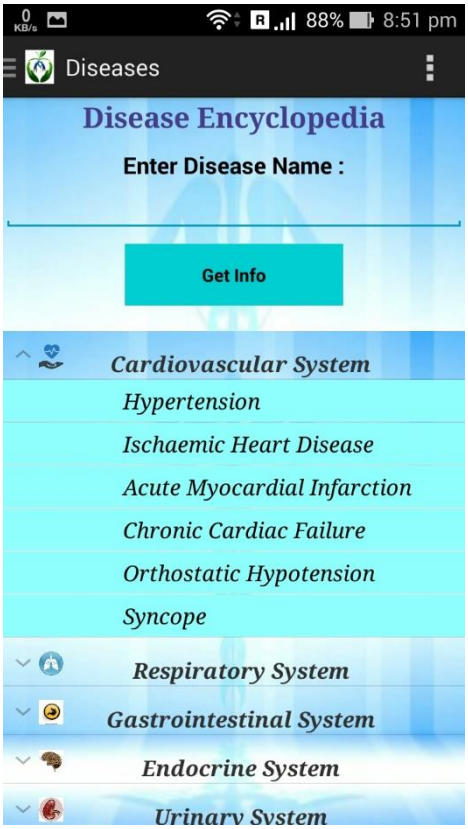
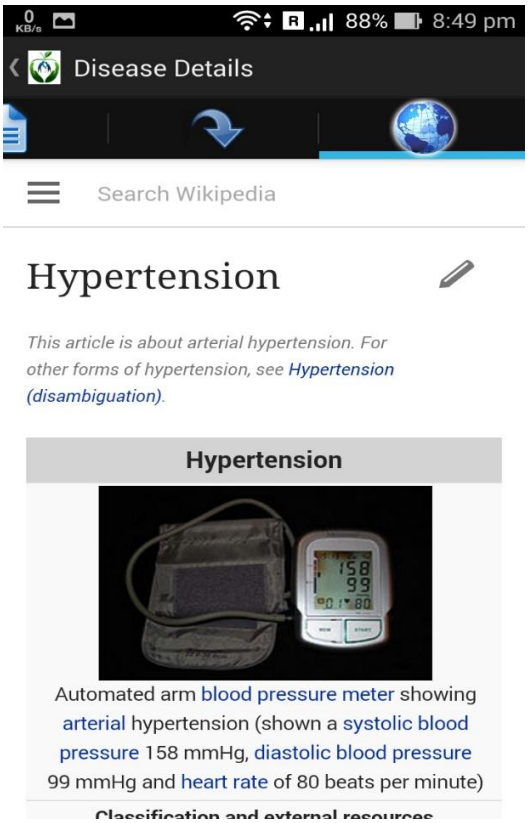


Fig. 8.2.2 – Hierarchical structure in app

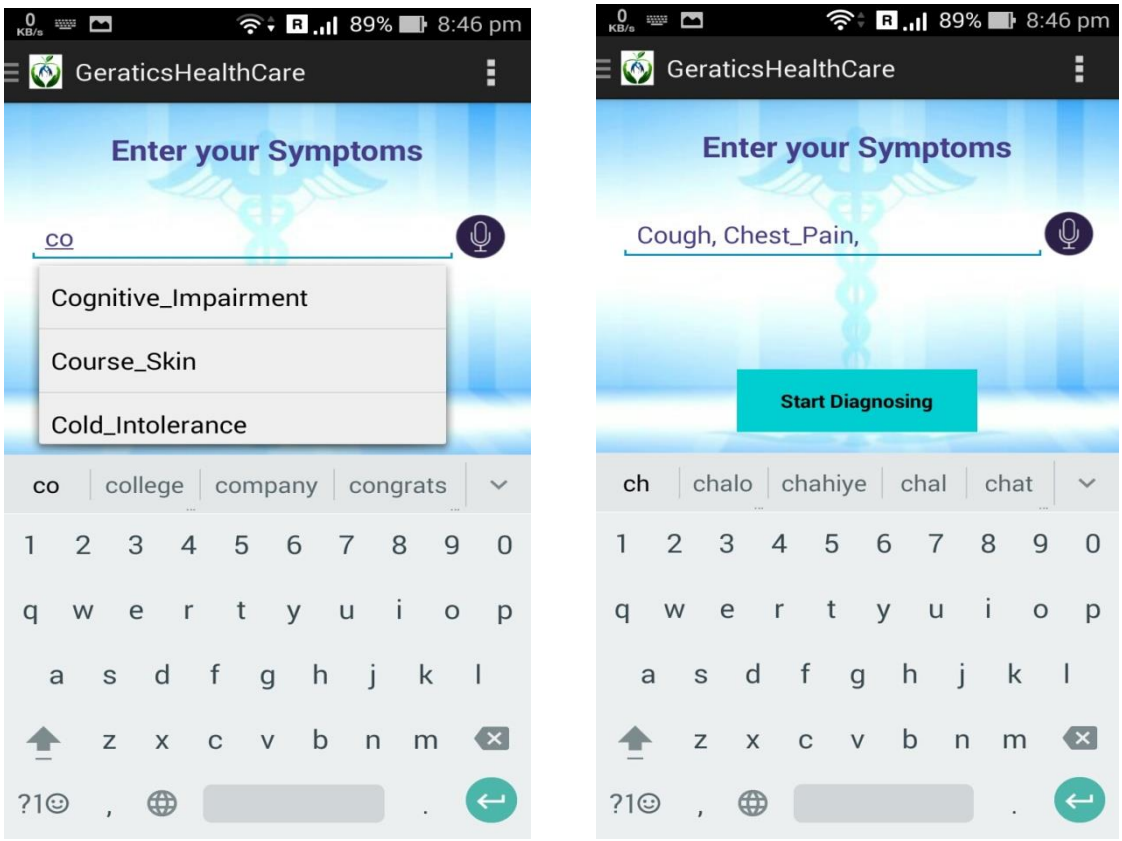


(a)information from internal resource (b)information from internet

Fig. 8.2.3(a)(b) – Information of Diseases

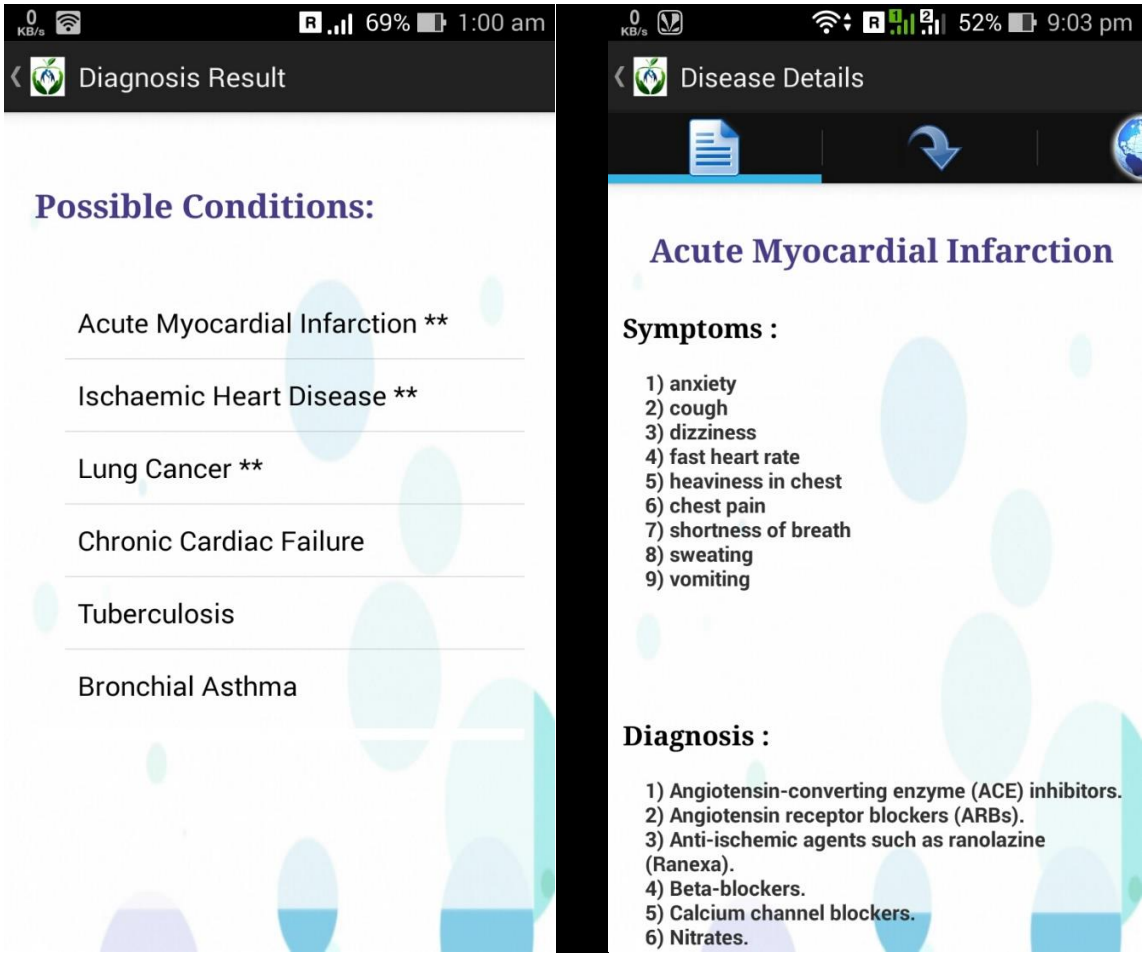
8.3 Decision making:

Decision making part takes symptoms as input. User can enter symptom through voice or through keyboard. In figure 8.3.1 user enters the symptoms Cough and Chest_Pain. Based on these symptoms decision making system shows the all the probable diseases and related information as shown in figure 8.3.2.



(a) (b)

Fig. 8.3.1. (a)(b) – Searching for symptoms



(a) (b)

Fig. 8.3.2. (a)(b) – Output for corresponding symptoms entered.

REFERENCES

[1] Arp, Robert, Barry Smith, and Andrew D. Spear. *Building ontologies with basic formal ontology*. Mit Press, 2015.

[2] Arp, Robert, and Barry Smith. *Function, role, and disposition in basic formal ontology*. *Nature* 2 (2008): 1-4.

[3] AB Dey *Handbook of Geriatrics* Paras Publication 2007

[4] R.G.Smith, Gorel Wachtmeister, P. Schwed *European Manual of Internal Medicine GERIATRIC MEDICINE,* 2001

[5] Brian Christopher Misiaszek**,** *Geriatric Medicine Survival Handbook,* 2008

[6] The Merck manual of geriatrics. Whitehouse Station, NJ: Merck Research Laboratories, 2000.

[7] Opearational Guideline , National Programme for Healthcare of The Elderly, Directorate General of Health Services Ministry of Health & Family Welfare Government of India , 2011

[8] Research at Mayo Clinic - Mayo Clinic Research www.mayo.edu/research