package Project\_AI;

import java.io.InputStream;

import java.util.List;

import java.util.Scanner;

import org.apache.jena.iri.impl.Main;

import com.hp.hpl.jena.query.Query;

import com.hp.hpl.jena.query.QueryExecution;

import com.hp.hpl.jena.query.QueryExecutionFactory;

import com.hp.hpl.jena.query.QueryFactory;

import com.hp.hpl.jena.query.QuerySolution;

import com.hp.hpl.jena.query.ResultSet;

import com.hp.hpl.jena.query.ResultSetFormatter;

import com.hp.hpl.jena.rdf.model.Model;

import com.hp.hpl.jena.rdf.model.ModelFactory;

import com.hp.hpl.jena.rdf.model.RDFNode;

import com.hp.hpl.jena.util.FileManager;

public class FindMyEvent

{

static String source,destination;

static void moodOfUser(String name)

{

Model model = FileManager.get().loadModel("/Users/kanikakapoor/Desktop/Protege-AI/myontology/AI\_project.owl");

System.out.println("Here is the list of Events ");

String queryString =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName" +"\n"+

"WHERE "

+ "{ ?bi "+" my:attend ?EventName."

+ "?bi rdf:type my:"+name+"}";

Query query = QueryFactory.create(queryString);

QueryExecution qexec = QueryExecutionFactory.create(query, model);

ResultSet Results = qexec.execSelect();

ResultSetFormatter.out(System.out,Results,query);

//oncampus or off campus

Scanner inaa=new Scanner(System.in);

System.out.println("Do you want to attend oncampus event or offcampus");

System.out.println("Press Number");

System.out.println("1. OnCampus");

System.out.println("2. OffCampus");

int choicelocation=inaa.nextInt();

System.out.println("Do you want to have Food at Event");

System.out.println("Press Number");

System.out.println("1. Yes");

System.out.println("2. No");

int choicefood = inaa.nextInt();

//oncampus and yes foood

if(choicelocation==1 && choicefood==1)

{

String queryString1 =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName ?hasFood" +"\n"+

"WHERE "

+ "{ ?bi my:attend ?EventName."

+ "?bi rdf:type my:"+name+"."

+ "?EventName rdf:type my:OnCampus."

+ "?EventName my:has\_Food ?hasFood."

+ "FILTER(?hasFood = true)}";

Query query1 = QueryFactory.create(queryString1);

QueryExecution qexec1 = QueryExecutionFactory.create(query1, model);

ResultSet Results1 = qexec1.execSelect();

ResultSetFormatter.out(System.out,Results1,query1);

}

//off campus and yes food

else if (choicelocation==2 && choicefood==1 )

{

String queryString1 =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName ?hasFood" +"\n"+

"WHERE "

+ "{ ?bi my:attend ?EventName."

+ "?bi rdf:type my:"+name+"."

+ "?EventName rdf:type my:OffCampus."

+ "?EventName my:has\_Food ?hasFood ."

+ "FILTER(?hasFood = true)}";

Query query1 = QueryFactory.create(queryString1);

QueryExecution qexec1 = QueryExecutionFactory.create(query1, model);

ResultSet Results1 = qexec1.execSelect();

ResultSetFormatter.out(System.out,Results1,query1);

}

//oncampus and no food

else if (choicelocation==1 && choicefood==2 )

{

String queryString1 =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName ?hasFood" +"\n"+

"WHERE "

+ "{"

+ "?bi my:attend ?EventName."

+ "?bi rdf:type my:"+name+"."

+ "?EventName rdf:type my:OnCampus."

+ "?EventName my:has\_Food ?hasFood ."

+ "FILTER(?hasFood= false)}";

Query query1 = QueryFactory.create(queryString1);

QueryExecution qexec1 = QueryExecutionFactory.create(query1, model);

ResultSet Results1 = qexec1.execSelect();

ResultSetFormatter.out(System.out,Results1,query1);

}

//offcampus and no food

else if (choicelocation==2 && choicefood==1 )

{

String queryString1 =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName ?hasFood" +"\n"+

"WHERE "

+ "{ ?bi my:attend ?EventName."

+ "?bi rdf:type my:"+name+"."

+ "?EventName rdf:type my:OffCampus."

+ "?EventName my:has\_Food ?hasFood ."

+ "FILTER(?hasFood = false)}";

Query query1 = QueryFactory.create(queryString1);

QueryExecution qexec1 = QueryExecutionFactory.create(query1, model);

ResultSet Results1 = qexec1.execSelect();

ResultSetFormatter.out(System.out,Results1,query1);

}

else

{

System.out.println("Invalid");

System.exit(0);

}

}

public static void main(String[] args)

{

//FileManager.get().addLocatorClassLoader(Main.class.getClassLoader());

// create an empty Model

//Model model = ModelFactory.createDefaultModel();

// use the FileManager to find the input file

//InputStream in = FileManager.get().open("/Users/kanikakapoor/Desktop/Protege-AI/myontology/AI\_project.owl");

//if (in == null)

//{

//throw new IllegalArgumentException(

// "File: " + "/Users/kanikakapoor/Desktop/Protege-AI/myontology/AI\_project.owl" + " not found");

//}

// read the RDF/XML file

//model.read(in, null);

String personName;

Scanner scannerObject = new Scanner(System.in);

// Reads a single line from the console

// and stores into personname variable

System.out.println("Hey ! How are you doing? May I know your name ");

personName = scannerObject.nextLine();

System.out.println("Hi "+personName+" Welcome to 'Find my Event' Application. \n"

+ "Today I will route you to your favorable event depending upon your mood !");

/\*

System.out.println("Do you want to attend oncampus event or offcampus");

System.out.println("Press Number");

System.out.println("1. OnCampus");

System.out.println("2. OffCampus");

int choi=scannerObject.nextInt();

System.out.println("Do you want to have Food at Event");

System.out.println("1. Yes");

System.out.println("2. No");

int choi2=scannerObject.nextInt();

if(choi==1)

{

if(choi2==1)

{

//oncampus and yes food

String queryString5 =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName ?Value" +"\n"+

"WHERE "

+ "{ "

+ "?EventName rdf:type my:OnCampus."

+ "?EventName my:has\_Food ?Value."

+ "FILTER (?Value=true)}"

+ "}";

Query query5 = QueryFactory.create(queryString5);

QueryExecution qexec5 = QueryExecutionFactory.create(query5, model);

ResultSet Results5 = qexec5.execSelect();

ResultSetFormatter.out(System.out,Results5,query5);

break;

}

else if(choi2==2)

{

//oncampus and No food

String queryString6 =

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX my: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?EventName ?Value" +"\n"+

"WHERE "

+ "{ "

+ "?EventName rdf:type my:OnCampus."

+ "?EventName my:has\_Food ?Value."

+ "FILTER (?Value=false)}"

+ "}";

Query query5 = QueryFactory.create(queryString5);

QueryExecution qexec5 = QueryExecutionFactory.create(query5, model);

ResultSet Results5 = qexec5.execSelect();

ResultSetFormatter.out(System.out,Results5,query5);

break;

}

}

if(choi==2)

{

if(choi2==1)

{

}

if(choi2==2)

{

}

}

\*/

System.out.println("So How are you feeling today ?");

System.out.println("Press any number");

System.out.println("1. Bored ");

System.out.println("2. Studious ");

System.out.println("3. Helpful ");

System.out.println("4. Peaceful ");

System.out.println("5. Happy ");

System.out.println("6. Neutral ");

String str;

int choice = scannerObject.nextInt();

switch(choice)

{

case 1:

str = "Bored";

moodOfUser(str);

//leisure\_subclass();

break;

case 2:

str = "Studious";

moodOfUser(str);

// music\_subclass();

break;

case 3:

str = "Helpful";

moodOfUser(str);

// places();

break;

case 4:

str = "Peaceful";

moodOfUser(str);

//attraction\_subClass();

break;

case 5:

str = "Happy";

moodOfUser(str);

//museum();

break;

case 6:

str="Neutral";

moodOfUser(str);

//abc();

break;

default:

System.out.println("Invalid");

System.exit(0);

break;

}

/\*

String sparqlQuery=

"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +

"PREFIX owl: <http://www.w3.org/2002/07/owl#>\n"+

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n"+

"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n"+

"PREFIX event: <http://www.semanticweb.org/kanikakapoor/ontologies/2015/4/untitled-ontology-18#>"+

"\n"+

"SELECT ?Event\_Name ?FoodAvailable \n"

+ "WHERE {\n"

+ "?Event\_Name event:has\_Food ?FoodAvailable \n"

+ "}";

com.hp.hpl.jena.query.Query query=QueryFactory.create(sparqlQuery);

QueryExecution qe=QueryExecutionFactory.create(query,model);

ResultSet results = qe.execSelect();

ResultSetFormatter.out(System.out,results,query);

//ResultSetFormatter.outputAsXML(System.out,results);

/\*MyOutputStream myOutput=new MyOutputStream();

ResultSetFormatter.out(myOutput, results, query);

String sparqlResults=myOutput.getString();

List vars=results.getResultVars();

while(results.hasNext())

{

QuerySolution qs=results.nextSolution();

System.out.println("----------solution-----");

for(int i=0; i< vars.size();i++)

{

String var = vars.get(i).toString();

RDFNode node = qs.get(var);

System.out.println(var + "\t" + node.toString());

}

}

\*/

/\*

// list the statements in the Model

StmtIterator iter = model.listStatements();

// print out the predicate, subject and object of each statement

while (iter.hasNext()) {

Statement stmt = iter.nextStatement(); // get next statement

Resource subject = stmt.getSubject(); // get the subject

Property predicate = stmt.getPredicate(); // get the predicate

RDFNode object = stmt.getObject(); // get the object

System.out.print("Subject:" + subject.toString());

System.out.print("Predicate:" + " " + predicate.toString() + " ");

if (object instanceof Resource) {

System.out.print("Object:" + object.toString());

} else {

// object is a literal

System.out.print("Object as Literal:" + " \"" + object.toString() + "\"");

}

System.out.println(" .");

}

// write it to standard out

//model.write(System.out);

\*/

}

}