

AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
ICHEP – Senior 2 Level – CESS
CSE488: Ontologies and The Semantic Web
Spring 2024



Football Ontology Project

Final Submission

Submitted to:

Dr. Ensaf Hussein

Eng. Eman Khaled

Submitted by:

Team (1)

Youssef George Fouad (19P9824)

Omar Helmy Ibrahim Elbanna (19P3904)

Kerollos Wageeh Youssef (19P3468)

Marc Nagy Nasry (19P3041)

GitHub Repository Submission Link: github.com/youssef7/football-knowledge-base

Table of Contents

Abstract.....	4
1.0 Modelling The Ontology	5
2.0 Populating The Ontology	8
3.0 Querying The Ontology	9
3.1 List All instances of Class Player	9
3.2 List All Instances of Class Coach	10
3.3 List All Instances of Class Referee	11
3.4 List All Instances of Class Match	12
3.5 List All Instances of Class Team	13
3.6 List All Instances of Class Championship	14
3.7 Name of All Teams That Participates in Laliga and Their Coach Names	15
3.8 Name of All Teams That Participate in Premier League and their Coach Names.....	16
3.9 Name and Age of Male Players in Real Madrid	17
3.10 How Many Players in Each Club	19
3.11 How Many Players That Plays in a Team That Participates in Premier League	20
3.12 Players Younger Than 23 Years Old	21
3.13 Name, Age, and Teams of Belgian Players	22
3.14 Players Whose Names Start With ‘A’ and Younger Than 25 Years Old	23
3.15 Player Name and Optionally His Team and Coach	24
3.16 Construct Query	25
3.17 Ask Query to Find If There is a Referee for Clasico Match.....	26
3.18 Describe Spanish Players in Real Madrid.....	27
4.0 Manipulating The Ontology Using Jena	28
4.1 Jena 1	28
4.2 Jena 2	29
4.3 Jena 3	31
4.4 Jena 4	32
4.5 Jena 5	34

4.6	Jena 6	35
5.0	Java Application.....	38
5.1	Core Components.....	38
5.1.1	Ontology Management.....	38
5.1.2	GUI Components	39
5.1.3	Filtering Logic	40
5.2	Functional Workflow	40
5.3	Data Flow Diagram (DFD)	40
5.3.1	System Components.....	41
5.3.2	Data Flows	41
5.4	Sample Tests	42

List of Figures

Figure 1 OWLViz Modelling Visualization	7
Figure 2 OntoGraph Modelling Visualization	7
Figure 3 PELLET Reasoner Output.....	8
Figure 4 Query 1 Response.....	9
Figure 5 Query 2 Response.....	10
Figure 6 Query 3 Response.....	11
Figure 7 Query 4 Response.....	12
Figure 8 Query 5 Response.....	13
Figure 9 Query 6 Response.....	14
Figure 10 Query 7 Response.....	15
Figure 11 Query 8 Response.....	16
Figure 12 Query 9 Response.....	17
Figure 13 Query 9 Response.....	18
Figure 14 Query 10 Response.....	19
Figure 15 Query 11 Response.....	20
Figure 16 Query 12 Response.....	21
Figure 17 Query 13 Response.....	22
Figure 18 Query 14 Response.....	23
Figure 19 Query 15 Response.....	24
Figure 20 Query 16 Response.....	25
Figure 21 Query 17 Response.....	26

Figure 22 Query 18 Response	27
Figure 23 Jena 1 Code Snippet	28
Figure 24 Jena 1 Output	28
Figure 25 Jena 2 Get Query Function Code Snippet	29
Figure 26 Jena 2 Code Snippet	30
Figure 27 Jena 2 Output	30
Figure 28 Jena 3 Code Snippet	31
Figure 29 Jena 3 Output	31
Figure 30 Jena 4 Has Match Function Code Snippet.....	32
Figure 31 Jena 4 Code Snippet	32
Figure 32 Jena 4 Found Match Output	33
Figure 33 Jena 4 Not Found Match Output	33
Figure 34 Jena 5 Code Snippet	34
Figure 35 Jena 5 Output	34
Figure 36 Jena 6 Rule 1 Implementation	35
Figure 37 Jena 6 Rule 3 Implementation	35
Figure 38 Jena 6 Rule 3 Implementation	36
Figure 39 Jena 6 Rule 1 Output	36
Figure 40 Jena 6 Rule 2 Output	37
Figure 41 Jena 6 Rule 3 Output	37
Figure 42: Java Application DFD	40

Abstract

This is a documentation of the Ontologies and The Semantic Web course project. This project aims to create an ontology, using the Protégé editor. The objective of this phase is only modelling the ontology, populating the ontology, and querying the ontology.

This is the objective with this project, without delving into the technical details, as they will be thoroughly explained and assessed throughout the whole document anyway.

The methods towards achieving such a goal will be clearly detailed as well as thoroughly described in terms of implementation throughout our development process. That is the purpose of this documentation, to know how to plan such a project and implement it as efficiently and accurately as possible.

1.0 Modelling The Ontology

We have decided to choose the football domain to model our ontology. The ontology consists of classes which represent different types of entities within the football domain. These include:

1. **Championship:** Represents football tournaments or leagues.
2. **Person:** Represents individuals involved in football, such as players, coaches, and referees.
3. **Coach:** Represents individuals who coach football teams. Coaches are subclassed from the more general class of Person.
4. **Match:** Represents individual football matches between two teams.
5. **Player:** Represents individuals who play football professionally or recreationally. Players are subclassed from the Person class.
6. **Referee:** Represents individuals who officiate football matches. Referees are subclassed from the Person class.
7. **Team:** Represents football teams or clubs.

Each class has its own data properties which capture attributes or characteristics of entities in the ontology. These may include:

1. Properties related to **championship** details such as name, country, type, and year.
2. Properties related to **match** details such as date, location, title, and result.
3. Properties related to **person** details such as name, age, gender, and nationality.
4. Properties related to **team** details such as name and country.

There exist object properties that connect classes to each other:

1. **between_teams**: Relates a match to the teams competing in it.
2. **coaches_in**: Connects a coach to the team they coach.
3. **competes_in**: Establishes the participation of a team in a match.
4. **contains_teams**: Associates a championship with the teams participating in it.
5. **has_coach**: Links a team to its coach.
6. **has_player**: Connects a team to its players.
7. **has_referee**: Indicates the referee officiating a match.
8. **participates_in**: Establishes the participation of a team in a championship.
9. **plays_in**: Connects a player to the team they play for.
10. **refereed**: Indicates matches officiated by a referee.

Each class has specific restrictions and constraints:

- A coach must coach exactly one team.
- A match must involve exactly two teams and one referee.
- A player can play for at most one team.
- A team must have at least one coach, between 23 and 30 players, and compete in multiple matches.
- A team must participate in at least one championship
- A referee must officiate at least one match.
- A match must have exactly one referee

The following figure shows the visualization of the ontology using OWLViz plugin:

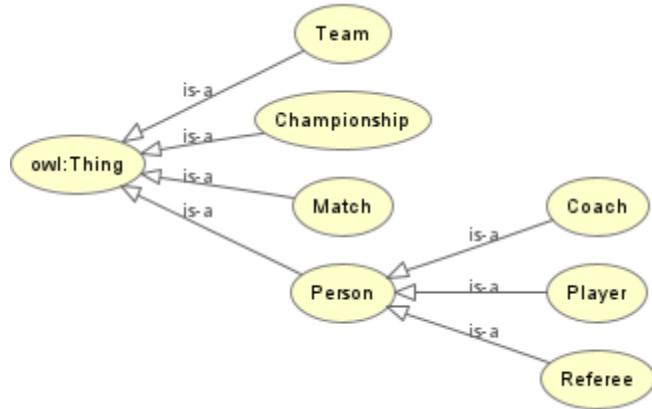


Figure 1 OWLViz Modelling Visualization

The following figure shows the visualization of the ontology using OntoGraph plugin:

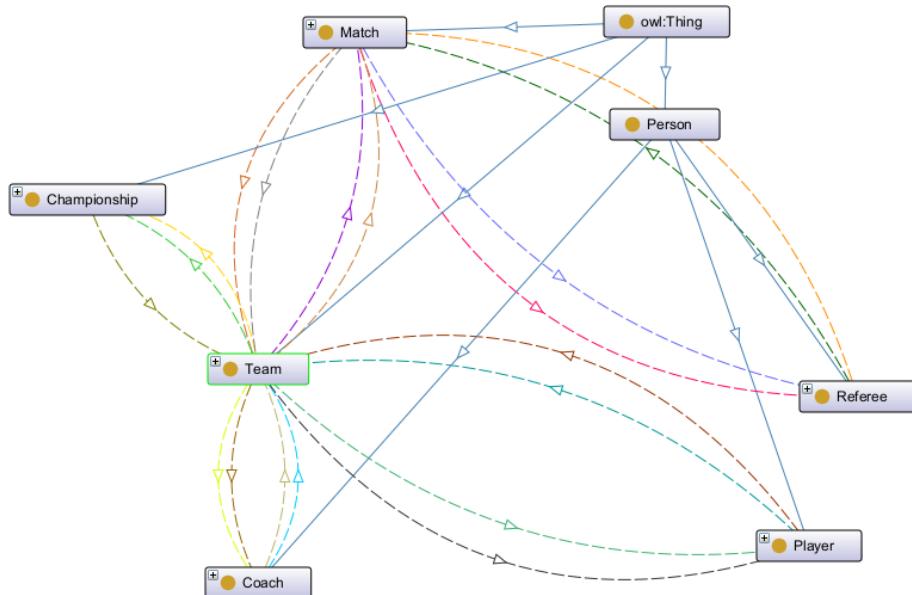


Figure 2 OntoGraph Modelling Visualization

2.0 Populating The Ontology

First, we've made two major leagues: LaLiga and the Premier League. Then, we made four teams: Real Madrid, Barcelona, Manchester City, and Liverpool. Each team has some players, at least 23 each. And of course, every team needs a coach, so we've got one for each team.

We've got two referees, we've created two matches: one between Real Madrid and Barcelona, and another between Manchester City and Liverpool.

After populating the ontology we checked for the consistency of the ontology using PELLET reasoner and there were no conflicts.

```
INFO 23:35:35 ----- Running Reasoner -----
INFO 23:35:35 Pre-computing inferences:
INFO 23:35:35   - class hierarchy
INFO 23:35:35   - object property hierarchy
INFO 23:35:35   - data property hierarchy
INFO 23:35:35   - class assertions
INFO 23:35:35   - object property assertions
INFO 23:35:35   - data property assertions
INFO 23:35:35   - same individuals
INFO 23:35:36 Ontologies processed in 603 ms by Pellet
INFO 23:35:36
INFO 23:35:36 [GitRepo] Git repository detected: D:\Github\football-knowledge-base\.git
INFO 23:35:36 [GitRepo] On branch: main
```

Figure 3 PELLET Reasoner Output

3.0 Querying The Ontology

3.1 List All instances of Class Player

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?player
```

```
WHERE {
```

```
    ?player rdf:type :Player .
```

```
}
```

player	
1	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Alex_Collado>
2	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Alvaro_Odriozola>
3	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Andriy_Lunin>
4	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Ansu_Fati>
5	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Casemiro>
6	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Clément_Lenglet>
7	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Dani_Carvajal>
8	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/David_Alaba>
9	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Eden_Hazard>
10	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Eder_Militao>
11	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Eduardo_Camavinga>
12	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Emerson_Royal>
13	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Federico_Valverde>
14	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Ferland_Mendy>
15	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Frenkie_de_Jong>
16	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Gareth_Bale>
17	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Gerard_Deulofeu>
18	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Gerard_Piqué>
19	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Iscó>
20	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Jordi_Alba>
21	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Iude_Bellinham>

Figure 4 Query 1 Response

3.2 List All Instances of Class Coach

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?player
```

```
WHERE {
```

```
    ?player rdf:type :Coach .
```

```
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The URL is `localhost:3030/#/dataset/football/query`. The query entered is:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT ?player
WHERE {
    ?player rdf:type :coach .
}
```

The results table shows four entries:

player
<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Xavi_Hernandez>
<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Carlo_Ancelotti>
<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Jurgen_Klopp>
<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Pep_Guardiola>

Figure 5 Query 2 Response

3.3 List All Instances of Class Referee

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?player
```

```
WHERE {
```

```
    ?player rdf:type :Referee .
```

```
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The title bar says "Apache Jena Fuseki UI". The address bar shows "localhost:3030/#/dataset/football/query". The main area is titled "SPARQL Query" and contains the following SPARQL query:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>
3
4 SELECT ?player
5 WHERE {
6     ?player rdf:type :Referee .
7 }
```

Below the query, there are tabs for "Selection of triples" and "Selection of classes", with "Selection of triples" being selected. There are also tabs for "Content Type (SELECT)" (JSON) and "Content Type (GRAPH)" (Turtle). A "Prefixes" section includes buttons for rdf, rdfs, owl, and xsd. At the bottom, there are buttons for "Table" and "Response", and a message "2 results in 0.011 seconds". The results table shows two entries:| player |
| --- |
| 1 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Hernandez_Hernandez> |
| 2 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#Michael_Oliver> |

At the bottom right, there are navigation buttons for page 1 of 1.

Figure 6 Query 3 Response

3.4 List All Instances of Class Match

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?player
```

```
WHERE {
```

```
    ?player rdf:type :Match .
```

```
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The title bar says "Apache Jena Fuseki UI". The address bar shows "localhost:3030/#/dataset/football/query". The main area is titled "SPARQL Query" and contains the following SPARQL query:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>
3
4 SELECT ?player
5 WHERE {
6     ?player rdf:type :Match .
7 }
```

The "Content Type (SELECT)" dropdown is set to "JSON". The "Content Type (GRAPH)" dropdown is set to "Turtle". Below the query editor, the results are displayed in a table:

player
1 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Classico>
2 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#Week_32>

At the bottom, it says "Showing 1 to 2 of 2 entries".

Figure 7 Query 4 Response

3.5 List All Instances of Class Team

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?player
WHERE {
  ?player rdf:type :Team . }
```

The screenshot shows the Apache Jena Fuseki UI interface. The title bar says "Apache Jena Fuseki UI" and the address bar shows "localhost:3030/#/dataset/football/query". The main area is titled "SPARQL Query" and contains the following text:

```
To try out some SPARQL queries against the selected dataset, enter your query here.
```

Example Queries: Selection of triples (selected), Selection of classes

SPARQL Endpoint: /football/query

Content Type (SELECT): JSON

Content Type (GRAPH): Turtle

Prefixes: rdf, rdfs, owl, xsd

Query text:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX :
3
4 SELECT ?player
5 WHERE {
6   ?player rdf:type :Team .
7 }
```

Results table:

player
1 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Barcelona>
2 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Real_Madrid>
3 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Liverpool>
4 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/Manchester_City>

Buttons at the bottom: Table, Response, 4 results in 0.014 seconds, Simple view, Ellipse, Filter query results, Page size: 50, etc.

Figure 8 Query 5 Response

3.6 List All Instances of Class Championship

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?player
WHERE {
  ?player rdf:type :Championship . }
```

The screenshot shows the Apache Jena Fuseki UI interface. At the top, it displays the URL `localhost:3030/#/dataset/football/query`. Below the header, there are tabs for `query`, `add data`, `edit`, and `info`. The `query` tab is selected. The main area is titled `SPARQL Query` with the sub-instruction `To try out some SPARQL queries against the selected dataset, enter your query here.`. Below this, there are two tabs: `Selection of triples` (selected) and `Selection of classes`. The `SPARQL Endpoint` field contains the URL `/football/query`. The `Content Type (SELECT)` dropdown is set to `JSON`. The `Prefixes` section includes buttons for `rdf`, `rdfs`, `owl`, and `xsd`. The `Content Type (GRAPH)` dropdown is set to `Turtle`. The query text area contains the following SPARQL code:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>
3
4 SELECT ?player
5 WHERE {
6   ?player rdf:type :Championship .
7 }
```

Below the query text, there is a note: `Press CTRL + <spacebar> to autocomplete`. At the bottom of the query area, there are buttons for `Table` and `Response`. The `Response` section indicates `2 results in 0.011 seconds`. The results table has one column labeled `player`. The data rows are:

1 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/La_Liga>
2 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#Premier_League>

At the bottom right of the response area, there are buttons for `Simple view`, `Ellipsis`, `Filter query results`, and `Page size: 50`.

Figure 9 Query 6 Response

3.7 Name of All Teams That Participates in Laliga and Their Coach Names

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT ?teamName ?coachName

WHERE {

?team rdf:type :Team ;

:participates_in ?championship ;

:has_coach ?coach .

?championship :championship_name "La Liga" .

?team :team_name ?teamName .

?coach :person_name ?coachName .

}

The screenshot shows the Apache Jena Fuseki UI interface. The top navigation bar includes tabs for 'query', 'add data', 'edit', and 'info'. Below the navigation is a search bar with the URL 'localhost:3030/#/dataset/football/query'. The main area is titled 'SPARQL Query' and contains the following SPARQL code:

```
PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>
SELECT ?teamName ?coachName
WHERE {
  ?team rdf:type :Team ;
  :participates_in ?championship ;
  :has_coach ?coach .
  ?championship :championship_name "La Liga" .
  ?team :team_name ?teamName .
  ?coach :person_name ?coachName .
}
```

The 'Content Type (SELECT)' dropdown is set to 'JSON'. The 'Content Type (GRAPH)' dropdown is set to 'Turtle'. Below the query editor, there are buttons for 'Table' and 'Response'. The response section shows 2 results in 0.011 seconds:

teamName	coachName
1 Barcelona	Xavi Hernandez
2 Real Madrid	Carlo Ancelotti

Figure 10 Query 7 Response

3.8 Name of All Teams That Participate in Premier League and their Coach Names

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?teamName ?coachName
```

```
WHERE {
```

```
?team rdf:type :Team ;  
:participates_in ?championship ;  
:has_coach ?coach .
```

```
?championship :championship_name "Premier League" .
```

```
?team :team_name ?teamName .
```

```
?coach :person_name ?coachName .
```

```
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The top navigation bar indicates the dataset is 'localhost:3030/#/dataset/football/query'. The main area is titled 'SPARQL Query' with the sub-instruction 'To try out some SPARQL queries against the selected dataset, enter your query here.' Below this, there are tabs for 'Selection of triples' (selected) and 'Selection of classes'. The SPARQL Endpoint is set to '/football/query'. The Content Type (SELECT) is set to 'JSON', and the Content Type (GRAPH) is set to 'Turtle'. The query text is pasted from the previous code block. The results section shows a table with two rows. The first row has 'teamName' as 'Liverpool' and 'coachName' as 'Jurgen Klopp'. The second row has 'teamName' as 'Manchester City' and 'coachName' as 'Pep Guardiola'. There are buttons for 'Table' and 'Response', and a note that there are '2 results in 0.01 seconds'. At the bottom right, there are buttons for 'Simple view', 'Ellipsis', 'Filter query results', and 'Page size: 50'.

teamName	coachName
Liverpool	Jurgen Klopp
Manchester City	Pep Guardiola

Figure 11 Query 8 Response

3.9 Name and Age of Male Players in Real Madrid

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT ?playerName ?playerAge

WHERE {

```
?player rdf:type :Player ;  
  :plays_in :Real_Madrid ;  
  :person_name ?playerName ;  
  :person_age ?playerAge ;  
  :person_gender "Male" .
```

}

The screenshot shows the Apache Jena Fuseki UI interface. The top navigation bar includes tabs for 'Selection of triples' and 'Selection of classes', and dropdowns for 'Content Type (SELECT)' set to 'JSON' and 'Content Type (GRAPH)' set to 'Turtle'. Below the header, the SPARQL endpoint is specified as '/football/query'. The main area displays the SPARQL query code. At the bottom, there are buttons for 'Table' and 'Response', with 'Table' selected. The response section shows 23 results in 0.019 seconds. The results are presented in a table with two columns: 'playerName' and 'playerAge'. The data is as follows:

playerName	playerAge
Alvaro Odriozola	"26"^^<http://www.w3.org/2001/XMLSchema#integer>
Andriy Lunin	"24"^^<http://www.w3.org/2001/XMLSchema#integer>
Casemiro	"30"^^<http://www.w3.org/2001/XMLSchema#integer>
Dani Carvajal	"31"^^<http://www.w3.org/2001/XMLSchema#integer>
David Alaba	"31"^^<http://www.w3.org/2001/XMLSchema#integer>
Eden Hazard	"32"^^<http://www.w3.org/2001/XMLSchema#integer>
Eder Militao	"24"^^<http://www.w3.org/2001/XMLSchema#integer>
Eduardo Camavinga	"20"^^<http://www.w3.org/2001/XMLSchema#integer>

Figure 12 Query 9 Response

playerName	playerAge
1 Alvaro Odriozola	"26"^^< http://www.w3.org/2001/XMLSchema#integer >
2 Andriy Lunin	"24"^^< http://www.w3.org/2001/XMLSchema#integer >
3 Casemiro	"30"^^< http://www.w3.org/2001/XMLSchema#integer >
4 Dani Carvajal	"31"^^< http://www.w3.org/2001/XMLSchema#integer >
5 David Alaba	"31"^^< http://www.w3.org/2001/XMLSchema#integer >
6 Eden Hazard	"32"^^< http://www.w3.org/2001/XMLSchema#integer >
7 Eder Militao	"24"^^< http://www.w3.org/2001/XMLSchema#integer >
8 Eduardo Camavinga	"20"^^< http://www.w3.org/2001/XMLSchema#integer >
9 Federico Valverde	"24"^^< http://www.w3.org/2001/XMLSchema#integer >
10 Ferland Mendy	"26"^^< http://www.w3.org/2001/XMLSchema#integer >
11 Gareth Bale	"33"^^< http://www.w3.org/2001/XMLSchema#integer >
12 Isco	"31"^^< http://www.w3.org/2001/XMLSchema#integer >
13 Jude Bellingham	"20"^^< http://www.w3.org/2001/XMLSchema#integer >
14 Karim Benzema	"35"^^< http://www.w3.org/2001/XMLSchema#integer >
15 Lucas Vazquez	"32"^^< http://www.w3.org/2001/XMLSchema#integer >
16 Luka Modric	"37"^^< http://www.w3.org/2001/XMLSchema#integer >
17 Marco Asensio	"26"^^< http://www.w3.org/2001/XMLSchema#integer >
18 Mariano Diaz	"29"^^< http://www.w3.org/2001/XMLSchema#integer >
19 Raphael Varane	"29"^^< http://www.w3.org/2001/XMLSchema#integer >
20 Rodrygo Goes	"22"^^< http://www.w3.org/2001/XMLSchema#integer >
21 Thibaut Courtois	"29"^^< http://www.w3.org/2001/XMLSchema#integer >
22 Toni Kroos	"34"^^< http://www.w3.org/2001/XMLSchema#integer >

Figure 13 Query 9 Response

3.10 How Many Players in Each Club

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT ?clubName (COUNT(?player) AS ?playerCount)

WHERE {

```
?player rdf:type :Player ;
:plays_in ?club ;
:person_name ?playerName .
?club :team_name ?clubName .
```

}

GROUP BY ?clubName

The screenshot shows the Apache Jena Fuseki UI interface. The top navigation bar includes tabs for 'Apache Jena Fuseki UI' and 'localhost:3030/#/dataset/football/query'. Below the navigation is a search bar and a 'Content Type (SELECT)' dropdown set to 'JSON'. The main area contains the SPARQL query code, which is identical to the one provided above. The results are displayed in a table:

clubName	playerCount
Manchester City	"24"^^<http://www.w3.org/2001/XMLSchema#integer>
Real Madrid	"23"^^<http://www.w3.org/2001/XMLSchema#integer>
Barcelona	"24"^^<http://www.w3.org/2001/XMLSchema#integer>
Liverpool	"25"^^<http://www.w3.org/2001/XMLSchema#integer>

At the bottom, it says 'Showing 1 to 4 of 4 entries'.

Figure 14 Query 10 Response

3.11 How Many Players That Plays in a Team That Participates in Premier League

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT (COUNT(?player) AS ?playerCount)
```

```
WHERE {
```

```
    ?team rdf:type :Team ;
```

```
        :participates_in ?premierLeague.
```

```
    ?premierLeague :championship_name "Premier League" .
```

```
    ?player :plays_in ?team.
```

```
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The title bar says "Apache Jena Fuseki UI". The address bar shows "localhost:3030/#/dataset/football/query". The main area is titled "SPARQL Query" and contains the following SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT (COUNT(?player) AS ?playerCount)
WHERE {
    ?team rdf:type :Team ;
        :participates_in ?premierLeague.
    ?premierLeague :championship_name "Premier League" .
    ?player :plays_in ?team.
}
```

The "Content Type (SELECT)" dropdown is set to "JSON". The "Content Type (GRAPH)" dropdown is set to "Turtle". Below the query, it says "1 result in 0.011 seconds". The result table shows one entry:

playerCount
1 "49"^^<http://www.w3.org/2001/XMLSchema#integer>

At the bottom, it says "Showing 1 to 1 of 1 entries".

Figure 15 Query 11 Response

3.12 Players Younger Than 23 Years Old

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT ?playerName ?age

WHERE {

```
?player rdf:type :Player ;  
    :person_name ?playerName ;  
    :person_age ?age .
```

FILTER (?age < 23)

}

	playerName	age
1	Alex Collado	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
2	Ansu Fati	"19"^^<http://www.w3.org/2001/XMLSchema#integer>
3	Eduardo Camavinga	"20"^^<http://www.w3.org/2001/XMLSchema#integer>
4	Jude Bellingham	"20"^^<http://www.w3.org/2001/XMLSchema#integer>
5	Pedri	"19"^^<http://www.w3.org/2001/XMLSchema#integer>
6	Riqui Puig	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
7	Rodrygo Goes	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
8	Sergiño Dest	"21"^^<http://www.w3.org/2001/XMLSchema#integer>
9	Curtis Jones	"20"^^<http://www.w3.org/2001/XMLSchema#integer>
10	Doku	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
11	Ferran Torres	"21"^^<http://www.w3.org/2001/XMLSchema#integer>
12	Hervey Elliott	"18"^^<http://www.w3.org/2001/XMLSchema#integer>
13	Ibrahim Konaté	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
14	Joe Hardy	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
15	Liam Delap	"18"^^<http://www.w3.org/2001/XMLSchema#integer>
16	Neco Williams	"20"^^<http://www.w3.org/2001/XMLSchema#integer>
17	Phil Foden	"21"^^<http://www.w3.org/2001/XMLSchema#integer>

Figure 16 Query 12 Response

3.13 Name, Age, and Teams of Belgian Players

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

SELECT ?playerName ?playerAge ?teamName

WHERE {

```
?player rdf:type :Player ;  
    :person_name ?playerName ;  
    :person_age ?playerAge ;  
    :person_nationality "Belgian" ;  
    :plays_in ?team .  
?team :team_name ?teamName .
```

}

The screenshot shows the Apache Jena Fuseki UI interface. The top navigation bar includes tabs for 'Apache Jena Fuseki UI', 'localhost:3030/#/dataset/football/query', and a search bar. Below the navigation is a 'SPARQL Query' section with a text area containing the SPARQL code. The code defines a query to select playerName, playerAge, and teamName for Belgian players, using prefixes for rdf and the local ontology. The 'Content Type (SELECT)' dropdown is set to 'JSON'. The 'Content Type (GRAPH)' dropdown is set to 'Turtle'. On the right side, there are download and copy/share buttons. The bottom section displays the query results in a table format. The table has three columns: 'playerName', 'playerAge', and 'teamName'. The results are as follows:

playerName	playerAge	teamName
Eden Hazard	"32"^^<http://www.w3.org/2001/XMLSchema#integer>	Real Madrid
Thibaut Courtois	"29"^^<http://www.w3.org/2001/XMLSchema#integer>	Real Madrid
Divock Origi	"26"^^<http://www.w3.org/2001/XMLSchema#integer>	Liverpool
Doku	"22"^^<http://www.w3.org/2001/XMLSchema#integer>	Manchester City
Kevin De Bruyne	"30"^^<http://www.w3.org/2001/XMLSchema#integer>	Manchester City

Figure 17 Query 13 Response

3.14 Players Whose Names Start With ‘A’ and Younger Than 25 Years Old

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
SELECT ?playerName ?playerAge
```

```
WHERE {
```

```
    ?player rdf:type :Player ;  
        :person_name ?playerName ;  
        :person_age ?playerAge .
```

```
FILTER (regex(?playerName, "^\u0410", "i") && ?playerAge < 25)
```

```
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The title bar says "Apache Jena Fuseki UI" and the address bar shows "localhost:3030/#/dataset/football/query". The main area is titled "SPARQL Query" with the sub-instruction "To try out some SPARQL queries against the selected dataset, enter your query here." Below this, there are tabs for "Selection of triples" and "Selection of classes", with "Selection of triples" being active. There are also tabs for "Prefixes" (with "rdf" selected), "Content Type (SELECT)" (set to "JSON"), and "Content Type (GRAPH)" (set to "Turtle"). The query editor contains the SPARQL code provided in the text above. To the right of the query editor is a "Run" button. Below the query editor is a table titled "Table" showing the results of the query. The table has two columns: "playerName" and "playerAge". The results are:

playerName	playerAge
1 Alex Collado	"22"^^<http://www.w3.org/2001/XMLSchema#integer>
2 Andriy Lunin	"24"^^<http://www.w3.org/2001/XMLSchema#integer>
3 Ansu Fati	"19"^^<http://www.w3.org/2001/XMLSchema#integer>

At the bottom of the table, it says "Showing 1 to 3 of 3 entries".

Figure 18 Query 14 Response

3.15 Player Name and Optionally His Team and Coach

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>
SELECT ?playerName ?teamName ?coachName
WHERE {
  ?player rdf:type :Player ;
    :person_name ?playerName .
  OPTIONAL {
    ?player :plays_in ?team .
    ?team :team_name ?teamName .
  }
  OPTIONAL {
    ?team :has_coach ?coach .
    ?coach :person_name ?coachName .
  }
}
```

playerName	teamName	coachName
1 Alex Collado	Barcelona	Xavi Hernandez
2 Alvaro Odriozola	Real Madrid	Carlo Ancelotti
3 Andriy Lunin	Real Madrid	Carlo Ancelotti
4 Ansu Fati	Barcelona	Xavi Hernandez
5 Casemiro	Real Madrid	Carlo Ancelotti
6 Clément Lenglet	Barcelona	Xavi Hernandez
7 Dani Carvajal	Real Madrid	Carlo Ancelotti
8 David Alaba	Real Madrid	Carlo Ancelotti
9 Eden Hazard	Real Madrid	Carlo Ancelotti
10 Eder Militao	Real Madrid	Carlo Ancelotti
11 Eduardo Camavinga	Real Madrid	Carlo Ancelotti
12 Emerson Royal	Barcelona	Xavi Hernandez
13 Federico Valverde	Real Madrid	Carlo Ancelotti
14 Ferland Mendy	Real Madrid	Carlo Ancelotti
15 Frenkie de Jong	Barcelona	Xavi Hernandez
16 Gareth Bale	Real Madrid	Carlo Ancelotti
17 Gerard Deulofeu	Barcelona	Xavi Hernandez
18 Gerard Piqué	Barcelona	Xavi Hernandez
19 Isco	Real Madrid	Carlo Ancelotti
20 Jordi Alba	Barcelona	Xavi Hernandez
21 Jude Bellingham	Real Madrid	Carlo Ancelotti

Figure 19 Query 15 Response

3.16 Construct Query

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

CONSTRUCT {

```
?player :plays_in ?team ;
:person_name ?playerName ;
:person_age ?playerAge .
?team :team_name ?teamName .
```

}

WHERE {

```
?player rdf:type :Player ;
:plays_in ?team ;
:person_name ?playerName ;
:person_age ?playerAge .
?team :team_name ?teamName .
```

}

subject	predicate	object
1 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"26"^^<http://www.w3.org/2001/XMLSchema#integer>
2 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Zack Steffen"^^<http://www.w3.org/2001/XMLSchema#string>
3 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...
4 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"30"^^<http://www.w3.org/2001/XMLSchema#integer>
5 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Virgil van Dijk"^^<http://www.w3.org/2001/XMLSchema#string>
6 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...
7 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"23"^^<http://www.w3.org/2001/XMLSchema#integer>
8 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Trent Alexander-Arnold"^^<http://www.w3.org/2001/XMLSchema#string>
9 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...
10 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"31"^^<http://www.w3.org/2001/XMLSchema#integer>
11 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Thiago Alcântara"^^<http://www.w3.org/2001/XMLSchema#string>
12 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...
13 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"27"^^<http://www.w3.org/2001/XMLSchema#integer>
14 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Takumi Minamino"^^<http://www.w3.org/2001/XMLSchema#string>
15 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...
16 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"35"^^<http://www.w3.org/2001/XMLSchema#integer>
17 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Scott Carson"^^<http://www.w3.org/2001/XMLSchema#string>
18 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...
19 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"29"^^<http://www.w3.org/2001/XMLSchema#integer>
20 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"Sadio Mané"^^<http://www.w3.org/2001/XMLSchema#string>
21 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	"24"^^<http://www.w3.org/2001/XMLSchema#integer>
22 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football#...	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/f...	

Figure 20 Query 16 Response

3.17 Ask Query to Find If There is a Referee for Clasico Match

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

```
ASK {  
    ?referee rdf:type :Referee ;  
    :refereed ?match .  
    ?match :match_title "Clasico" .  
}
```

The screenshot shows the Apache Jena Fuseki UI interface. The URL is `localhost:3030/#/dataset/football/query`. The query entered is:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
2 PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>  
3  
4 ASK {  
5     ?referee rdf:type :Referee ;  
6         :refereed ?match .  
7     ?match :match_title "Clasico" .  
8 }  
9
```

The Content Type (SELECT) is set to JSON, and the Content Type (GRAPH) is set to Turtle. The response is shown as a table with one row, indicating a true result.

True
True

Figure 21 Query 17 Response

3.18 Describe Spanish Players in Real Madrid

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

PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>

DESCRIBE ?player

WHERE {

```
?player rdf:type :Player ;  
    :plays_in ?team ;  
    :person_nationality "Spanish".  
  
?team :team_name "Real Madrid" .
```

}

subject	predicate	object
1 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Spanish"^^<http://www.w3.org/2001/XMLSchema#string>
2 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Isco"^^<http://www.w3.org/2001/XMLSchema#string>
3 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Male"^^<http://www.w3.org/2001/XMLSchema#string>
4 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"31"^^<http://www.w3.org/2001/XMLSchema#integer>
5 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/fo...
6 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
7 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/fo...
8 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Spanish"^^<http://www.w3.org/2001/XMLSchema#string>
9 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Marco Asensio"^^<http://www.w3.org/2001/XMLSchema#string>
10 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Male"^^<http://www.w3.org/2001/XMLSchema#string>
11 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"26"^^<http://www.w3.org/2001/XMLSchema#integer>
12 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/fo...
13 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
14 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/fo...
15 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Spanish"^^<http://www.w3.org/2001/XMLSchema#string>
16 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Lucas Vazquez"^^<http://www.w3.org/2001/XMLSchema#string>
17 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Male"^^<http://www.w3.org/2001/XMLSchema#string>
18 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"32"^^<http://www.w3.org/2001/XMLSchema#integer>
19 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/fo...
20 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
21 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/fo...
22 <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	<http://www.semanticweb.org/omarelbanna/ontologies/2024/3/footbal...>	"Spanish"^^<http://www.w3.org/2001/XMLSchema#string>

Figure 22 Query 18 Response

4.0 Manipulating The Ontology Using Jena

4.1 Jena 1

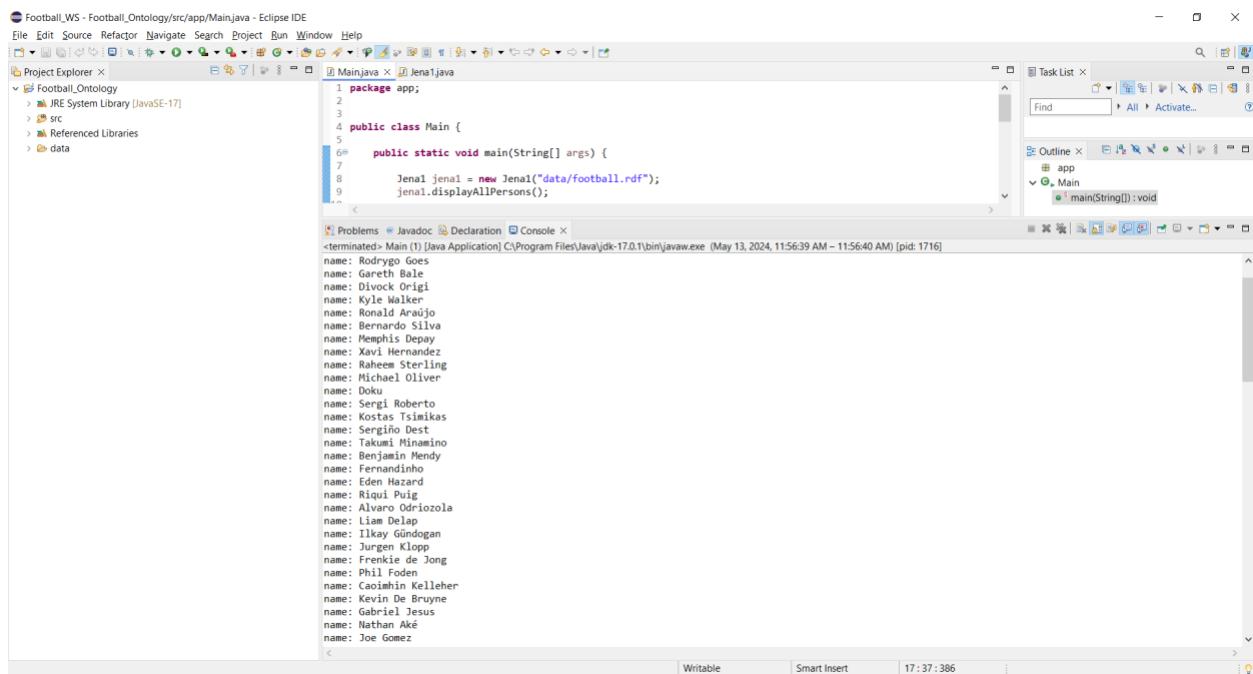
Create a java program (Jena1.java) that loads the ontology and displays all the Persons (without using queries, without inference).



```
app - Jena1.java

25 public void displayAllPersons() {
26     OntClass cl = ontmodel.getOntClass(namespace + "Person");
27     Property person_name = ontmodel.getProperty(namespace+"person_name");
28     for (ExtendedIterator i = cl.listInstances(); i.hasNext();) {
29         OntResource c = (OntResource) i.next();
30         System.out.println("name: " + c.getProperty(person_name).getString());
31     }
}
```

Figure 23 Jena 1 Code Snippet



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Football_WS - Football_Ontology/src/app/Main.java - Eclipse IDE
- Toolbar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Project Explorer:** Shows the project structure with a folder named "Football_Ontology".
- Code Editor:** Displays the Java code for Main.java, which calls the Jena1 class.
- Console:** Shows the command-line output of the Java application, listing numerous names of football players.
- Output:** Shows the names of the players listed sequentially.

```
name: Rodrigo Goes
name: Gareth Bale
name: Divock Origi
name: Kyle Walker
name: Ronald Araújo
name: Bernardo Silva
name: Memphis Depay
name: Xavi Hernandez
name: Raheem Sterling
name: Michael Oliver
name: Doku
name: Sergi Roberto
name: Kostas Tsimikas
name: Sergio Dest
name: Takumi Minamino
name: Benjamin Mendy
name: Fernandinho
name: Eric Maxim Choupo-Moting
name: Rúben Puls
name: Alvaro Odriozola
name: Liam Delap
name: Ilkay Gündogan
name: Jürgen Klopp
name: Frenkie de Jong
name: Phil Foden
name: Caoimhin Kelleher
name: Kevin De Bruyne
name: Gabriel Jesus
name: Nathan Aké
name: Joe Gomez
```

Figure 24 Jena 1 Output

4.2 Jena 2

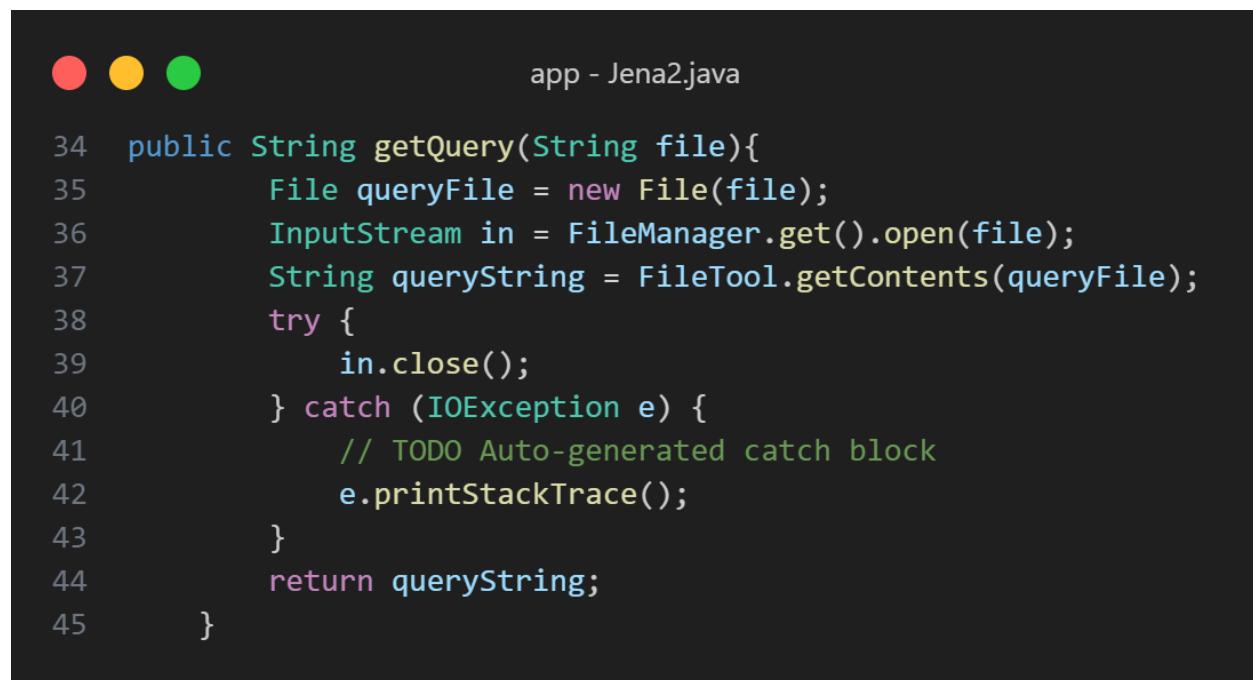
Create a java program (Jena2.java) that loads the ontology and displays all the Persons (using a query, without inference). Create the used query in text file under the data folder.

First, we created this query in data/query.txt

```
PREFIX : <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?name
WHERE
{
    ?per rdf:type :Person .
    ?per :person_name ?name.
}
```

Then we used this function to read the query

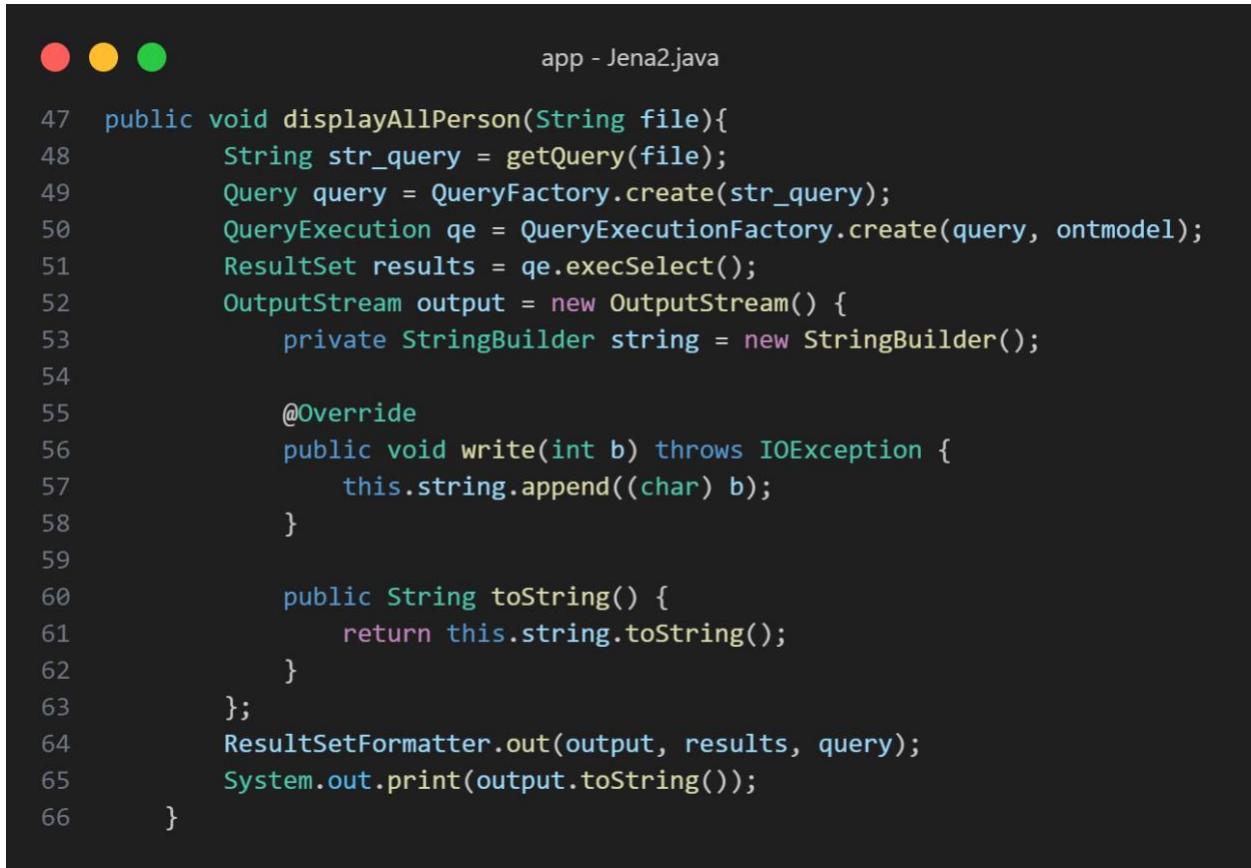


The screenshot shows a code editor window with a dark theme. At the top left, there are three colored circular icons (red, yellow, green). The title bar on the right says "app - Jena2.java". The code itself is a single method named "getQuery" with the following content:

```
34 public String getQuery(String file){
35     File queryFile = new File(file);
36     InputStream in = FileManager.get().open(file);
37     String queryString = FileTool.getContents(queryFile);
38     try {
39         in.close();
40     } catch (IOException e) {
41         // TODO Auto-generated catch block
42         e.printStackTrace();
43     }
44     return queryString;
45 }
```

Figure 25 Jena 2 Get Query Function Code Snippet

Then we used this function to display all persons using the defined query in data/query.txt



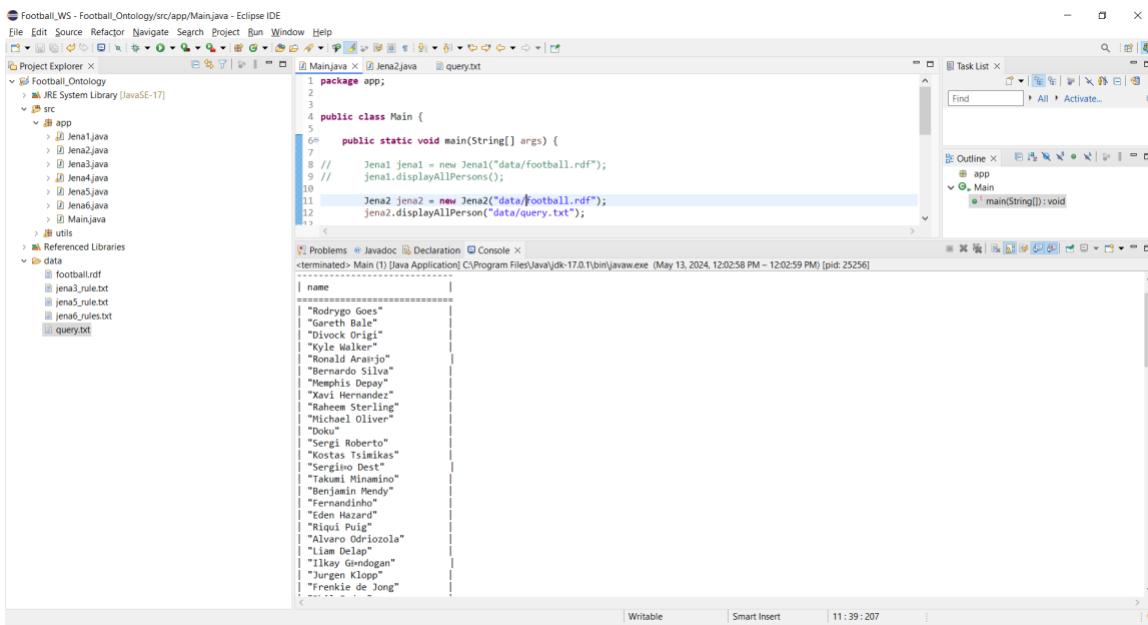
```

app - Jena2.java

47 public void displayAllPerson(String file){
48     String str_query = getQuery(file);
49     Query query = QueryFactory.create(str_query);
50     QueryExecution qe = QueryExecutionFactory.create(query, ontmodel);
51     ResultSet results = qe.execSelect();
52     OutputStream output = new OutputStream() {
53         private StringBuilder string = new StringBuilder();
54
55         @Override
56         public void write(int b) throws IOException {
57             this.string.append((char) b);
58         }
59
60         public String toString() {
61             return this.string.toString();
62         }
63     };
64     ResultSetFormatter.out(output, results, query);
65     System.out.print(output.toString());
66 }

```

Figure 26 Jena 2 Code Snippet



The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure with files like Main.java, Jena1.java, Jena2.java, Jena3.java, Jena4.java, Jena5.java, and Main.java.
- Code Editor:** Displays the Main.java file containing Java code to run Jena1 and Jena2.
- Console:** Shows the output of the Java application, listing names of football players such as "Rodrygo Goes", "Gareth Bale", "Divock Origi", etc.

Figure 27 Jena 2 Output

4.3 Jena 3

Create a java program (Jena3.java) that loads the ontology and displays all the Coaches (without using queries, using inference). To load the inferred model, use the JenaEngine.readInferredModelFromRuleFile method and use owl rules

First we defined this rule in data/jena3_rule.txt

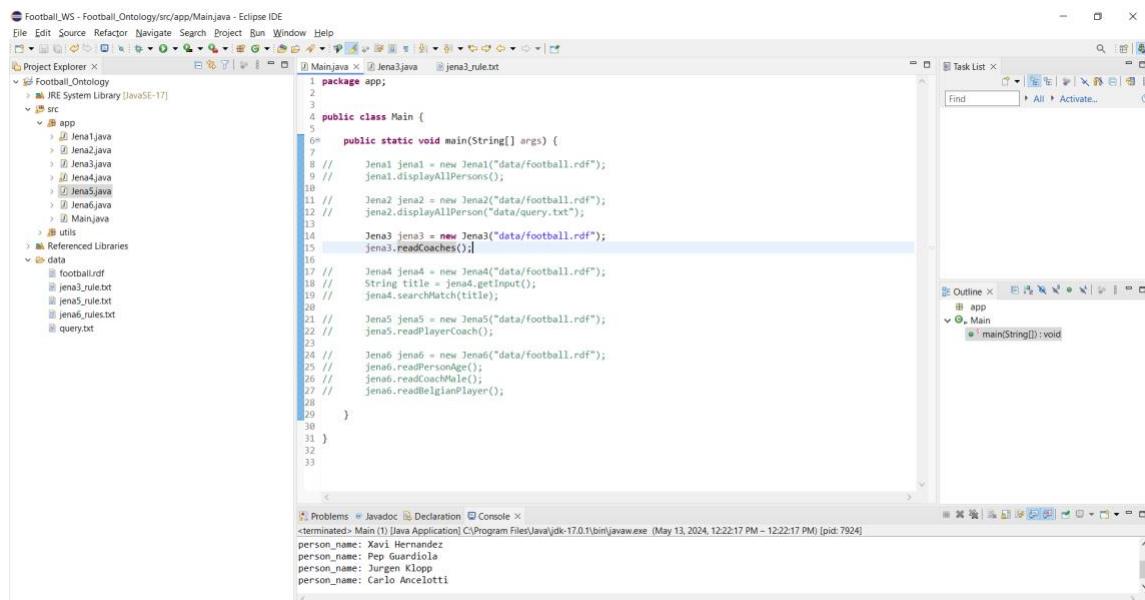
```
@include <OWLMicro>.  
@prefix ont: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.  
[rule: (?per rdf:type ont:Coach) -> (?per rdf:type ont:A) ]
```

Then we used this function to read the inferred model and display coaches



```
app - Jena3.java  
  
21 public void readCoaches(){  
22     Model ourmodel = JenaEngine.readInferredModelFromRuleFile(model,"data/jena3_rule.txt");  
23     Property rdfType = ourmodel.getProperty("http://www.w3.org/1999/02/22-rdf-syntax-ns#", "type");  
24     Resource A = ourmodel.getResource(namespace+'A');  
25     ResIterator iter = ourmodel.listResourcesWithProperty(rdfType,A);  
26     for (; iter.hasNext();) {  
27         Resource i = iter.next();  
28         JenaEngine.readRsDataTpe(ourmodel, namespace, i, "person_name");  
29     }  
30 }
```

Figure 28 Jena 3 Code Snippet



Football_WS - Football_Ontology/src/app/Main.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer X Football_Ontology JRE System Library [JavaSE-17] src app Jena1.java Jena3.java Jena5.java Jena6.java Main.java utils Referenced Libraries football.rdf jena3_rule.txt jena5_rule.txt jena6_rules.txt query.txt

Main.java X Jena3.java jena3_rule.txt

```
1 package app;  
2  
3 public class Main {  
4  
5     public static void main(String[] args) {  
6         Jena1 jena1 = new Jena1("data/football.rdf");  
7         jena1.displayAllPersons();  
8     }  
9     Jena2 jena2 = new Jena2("data/football.rdf");  
10    jena2.displayAllPerson("data/query.txt");  
11  
12     Jena3 jena3 = new Jena3("data/football.rdf");  
13     jena3.readCoaches();  
14  
15     Jena4 jena4 = new Jena4("data/football.rdf");  
16     String title = jena4.getInput();  
17     jena4.searchMatch(title);  
18  
19     Jena5 jena5 = new Jena5("data/football.rdf");  
20     jena5.readPlayerCoach();  
21  
22     Jena6 jena6 = new Jena6("data/football.rdf");  
23     jena6.readPersonAge();  
24     jena6.readCoachMale();  
25     jena6.readBelgianPlayer();  
26  
27 }  
28  
29  
30 }
```

Task List X Find All Activate...
Outline X app Main mainString():void

Console X
<terminated> Main (1) Java Application C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (May 13, 2024, 12:22:17 PM - 12:22:17 PM) [pid: 7924]
person_name: Xavi Hernandez
person_name: Gerard Pique
person_name: Jurgen Klopp
person_name: Carlo Ancelotti

Figure 29 Jena 3 Output

4.4 Jena 4

4. Create a java program (Jena4.java) that:

- a. Reads a name of a Match
- b. If it doesn't exist displays an error message
- c. Else, display its teams, result, location and referee

We used this function to check if the model has the input match



```
app - Jena4.java

40 public boolean hasMatch(String title) {
41     Resource rs = model.getResource(namespace + title);
42     Property ptitle = model.getProperty(namespace + "match_title");
43     if (rs != null && ptitle != null) {
44         if(rs.getProperty(ptitle) != null )
45             return true;
46         else
47             return false;
48     }
49     return false;
50 }
```

Figure 30 Jena 4 Has Match Function Code Snippet

Then we search for the match using this function



```
app - Jena4.java

27 public void searchMatch(String title) {
28     if (hasMatch(title)) {
29         Resource rs = model.getResource(namespace + title);
30         System.out.println(rs.getLocalName());
31         JenaEngine.readRsDataType(model, namespace, rs, "match_result");
32         JenaEngine.readRsDataType(model, namespace, rs, "match_location");
33         JenaEngine.readObjectType(model, namespace, title, "between_teams");
34         JenaEngine.readObjectType(model, namespace, title, "has_referee");
35     } else {
36         System.out.println("Error: Wrong title!");
37     }
38 }
```

Figure 31 Jena 4 Code Snippet

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays the project structure for "Football_WS - Football_Ontology/src/app/Mainjava".
- Main.java (Main4.java):** The code implements a Main class with a main method that creates various Jena models (Jena1 to Jena6) and performs search operations.
- Console Output:** Shows the results of a search query for "Classico". The output includes:


```

<terminated> Main () [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (May 13, 2024, 12:25:52 PM - 12:25:57 PM) [pid: 7340]
Enter a name of a match:
Classico
Classico
match_result: 2-0
match_location: Camp Noe
Classico has_teams:
team_name: Real Madrid
team_name: Barcelona
Classico has_referee:
person_name: Hernandez Hernandez
      
```
- Outline View:** Shows the class structure with a main method.

Figure 32 Jena 4 Found Match Output

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays the project structure for "Football_WS - Football_Ontology/src/app/Mainjava".
- Main.java (Main4.java):** The code is identical to Figure 32, implementing a Main class with a main method that creates various Jena models (Jena1 to Jena6) and performs search operations.
- Console Output:** Shows the results of a search query for "Match1". The output includes error messages and a final error message:


```

<terminated> Main () [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (May 13, 2024, 12:29:48 PM - 12:29:53 PM) [pid: 17324]
log4j:WARN No loggers could be found for logger (org.apache.jena.util.fileManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Enter a name of a match:
Match1
Error: Wrong title!
      
```
- Outline View:** Shows the class structure with a main method.

Figure 33 Jena 4 Not Found Match Output

4.5 Jena 5

Create a java program (Jena5.java) that displays all persons that are actors and directors. Do this using a rule that defines a new class ActorDirector. The rule file must be saved in the data folder.

First we defined this rule in data/jena5_rule.txt

```
@include <OWLMicro>.
@prefix ont: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
[rule1: (?per rdf:type ont:Player) (?per rdf:type ont:Coach)->(?per rdf:type ont:PlayerCoach)]
```

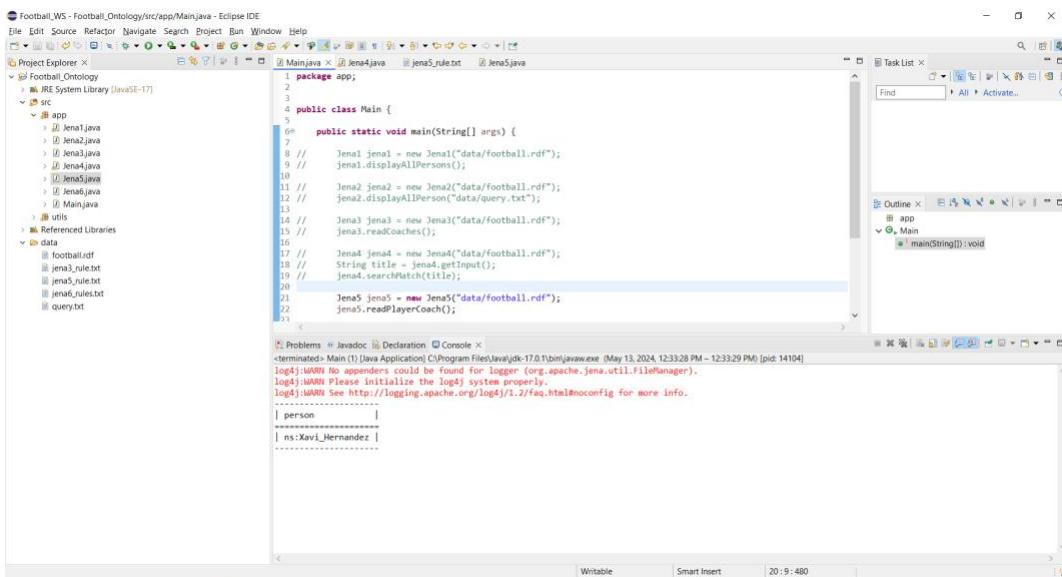
Then we used this function to find all individuals with type PlayerCoach



```
app - Jena5.java

18 public void readPlayerCoach(){
19     this.model = JenaEngine.readInferredModelFromRuleFile(model, "data/jena5_rule.txt");
20
21     String query = "PREFIX ns: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>" +
22         + "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
23         + "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>" +
24         + "SELECT ?person " +
25         + "WHERE{"
26         + "?person rdf:type ns:PlayerCoach. " + "}";
27     System.out.println(JenaEngine.executeQuery(model, query));
28 }
```

Figure 34 Jena 5 Code Snippet



```
Football_WS - Football_Ontology/src/app/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
  Football_Ontology
    > Main System Library [javaSE-17]
      > app
        > Jena1Java
        > Jena2Java
        > Jena3Java
        > Jena4Java
        > Jena5Java
        > MainJava
        > Rule
        > Rule
      > References
      > data
        > football.rdf
        > jena3.rule.txt
        > jena5.rule.txt
        > jena6.rules.txt
        > query.txt
  Main.java X Jena4.java jena5_rule.txt Jena5.java
  package app;
  ...
  public class Main {
  ...
    public static void main(String[] args) {
    ...
      // Jena1 jena1 = new Jena1("data/football.rdf");
      // jena1.displayAllPersons();
      ...
      // Jena2 jena2 = new Jena2("data/football.rdf");
      // jena2.displayAllPerson("data/query.txt");
      ...
      // Jena3 jena3 = new Jena3("data/football.rdf");
      // jena3.readCoaches();
      ...
      // Jena4 jena4 = new Jena4("data/football.rdf");
      // String title = jena4.getInput();
      // jena4.searchMatch(title);
      ...
      Jena5 jena5 = new Jena5("data/football.rdf");
      jena5.readPlayerCoach();
    }
  }
  Problems Javadoc Declaration Console X
  <no file> (1 Java file) Java Platform SE binary JDK 17.0.1 bin\javaw.exe (May 13, 2024, 12:33:28 PM - 12:33:29 PM) [pid: 14104]
  log4j:WARN No loggers could be found for logger (org.apache.jena.util.fileManager).
  log4j:WARN Please initialize the log4j system properly.
  log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
  | person |
  | ns:Xavi_Hernandez |
```

Figure 35 Jena 5 Output

4.6 Jena 6

Specify 3 different rules and implement them in a java program (Jena6.java)

First we defined 3 rules in data/jena6_rules.txt

```
@include <OWLMicro>.  
@prefix ont: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>.  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.  
  
[rule1: (?per rdf:type ont:Player) (?per ont:person_age ?age) lessThan(?age, 22)->  
 (?per rdf:type ont:PersonAge)]  
[rule2:(?per rdf:type ont:Coach) (?per ont:person_gender "Male")-> (?per rdf:type  
 ont:CoachMale)]  
[rule3: (?per rdf:type ont:Player) (?per ont:person_nationality "Belgian") -> (?per  
 rdf:type ont:BelgianPlayer)]
```

Then used these functions implement the rules respectively



```
app - Jena6.java  
  
20 public void readPersonAge() {  
21     String query = "PREFIX ns: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>"  
22         + "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"  
23         + "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>"  
24         + "SELECT ?name ?age "  
25         + "WHERE{"  
26             + "?person rdf:type ns:PersonAge. ?person ns:person_name ?name. ?person ns:person_age ?age." + "}";  
27     System.out.println(JenaEngine.executeQuery(model, query));  
28 }
```

Figure 36 Jena 6 Rule 1 Implementation



```
app - Jena6.java  
  
30 public void readCoachMale() {  
31     String query = "PREFIX ns: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>"  
32         + "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"  
33         + "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>"  
34         + "SELECT ?name ?gender "  
35         + "WHERE{"  
36             + "?person rdf:type ns:CoachMale. ?person ns:person_name ?name. ?person ns:person_gender ?gender."  
37             + "}";  
38     System.out.println(JenaEngine.executeQuery(model, query));  
39 }
```

Figure 37 Jena 6 Rule 3 Implementation

```

app - Jena6.java

41 public void readBelgianPlayer() {
42     String query = "PREFIX ns: <http://www.semanticweb.org/omarelbanna/ontologies/2024/3/football/>" +
43         "+ PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
44         "+ PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>" +
45         "+ SELECT ?name ?age ?nationality "
46         "+ WHERE{" +
47             "+ ?player rdf:type ns:BelgianPlayer. ?player ns:person_name ?name. ?player ns:person_age ?age. ?player ns:person_nationality ?nationality ." +
48         "+ }"; +
49     System.out.println(JenaEngine.executeQuery(model, query));
50 }

```

Figure 38 Jena 6 Rule 3 Implementation

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Displays files like Main.java, Jena1.java, Jena2.java, Jena3.java, Jena4.java, Jena5.java, and Main.java.
- Code Editor:** Shows the Main.java file with Java code for reading RDF data and executing queries.
- Console:** Displays the output of the Java application, showing a table of names and ages.
- Table Output:**

name	age
"Curtis Jones"	20
"Harvey Elliott"	18
"Ferran Torres"	21
"Sergio Dest"	21
"Neco Williams"	20
"Phil Foden"	21
"Fernando	19
"Liam Delap"	18
"Eduardo Camavinga"	20
"Jude Bellingham"	20
"Ansu Fati"	19

Figure 39 Jena 6 Rule 1 Output

Football_WS - Football_Ontology/src/app/MainJava - Eclipse IDE

```

1 // Jena6.java
2 jena2.displayAllPerson("data/query.txt");
3
4 // Jena3 jena3 = new Jena3("data/football.rdf");
5 // jena3.readCoaches();
6
7 // Jena4 jena4 = new Jena4("data/football.rdf");
8 // String title = jena4.getInput();
9 // jena4.searchMatch(title);
10
11 // Jena5 jena5 = new Jena5("data/football.rdf");
12 // jena5.readPlayerCoach();
13
14 // Jena6 jena6 = new Jena6("data/football.rdf");
15 // jena6.readPersonAge();
16 // jena6.readCoachMale();
17 // jena6.readBelgianPlayer();
18
19 }
20
21 }
22
23 }
24
25 }
26
27 }
28
29 }
30
31 }
32

```

Output:

name	gender
"Jürgen Klopp"	"Male"
"Carlo Ancelotti"	"Male"
"Pep Guardiola"	"Male"
"Xavi Hernandez"	"Male"

Figure 40 Jena 6 Rule 2 Output

Football_WS - Football_Ontology/src/app/MainJava - Eclipse IDE

```

1 // Jena6.java
2 jena2.displayAllPerson("data/query.txt");
3
4 // Jena3 jena3 = new Jena3("data/football.rdf");
5 // jena3.readCoaches();
6
7 // Jena4 jena4 = new Jena4("data/football.rdf");
8 // String title = jena4.getInput();
9 // jena4.searchMatch(title);
10
11 // Jena5 jena5 = new Jena5("data/football.rdf");
12 // jena5.readPlayerCoach();
13
14 // Jena6 jena6 = new Jena6("data/football.rdf");
15 // jena6.readPersonAge();
16 // jena6.readCoachMale();
17 // jena6.readBelgianPlayer();
18
19 }
20
21 }
22
23 }
24
25 }
26
27 }
28
29 }
30
31 }
32

```

Output:

name	age	nationality
"Thibaut Courtois"	29	"Belgian"
"Doku"	22	"Belgian"
"Eden Hazard"	32	"Belgian"
"Kevin De Bruyne"	30	"Belgian"
"Divock Origi"	26	"Belgian"

Figure 41 Jena 6 Rule 3 Output

5.0 Java Application

The Java application implemented utilizes the OWLAPI and OpenPellet reasoner to provide a sophisticated interface for managing and viewing sports matches based on selected filters from all available teams and referees.

The application is designed to enable users to dynamically include or exclude teams and referees to filter the view of sports matches. It uses a graphical user interface (GUI) to facilitate user interaction, allowing users to make selections.

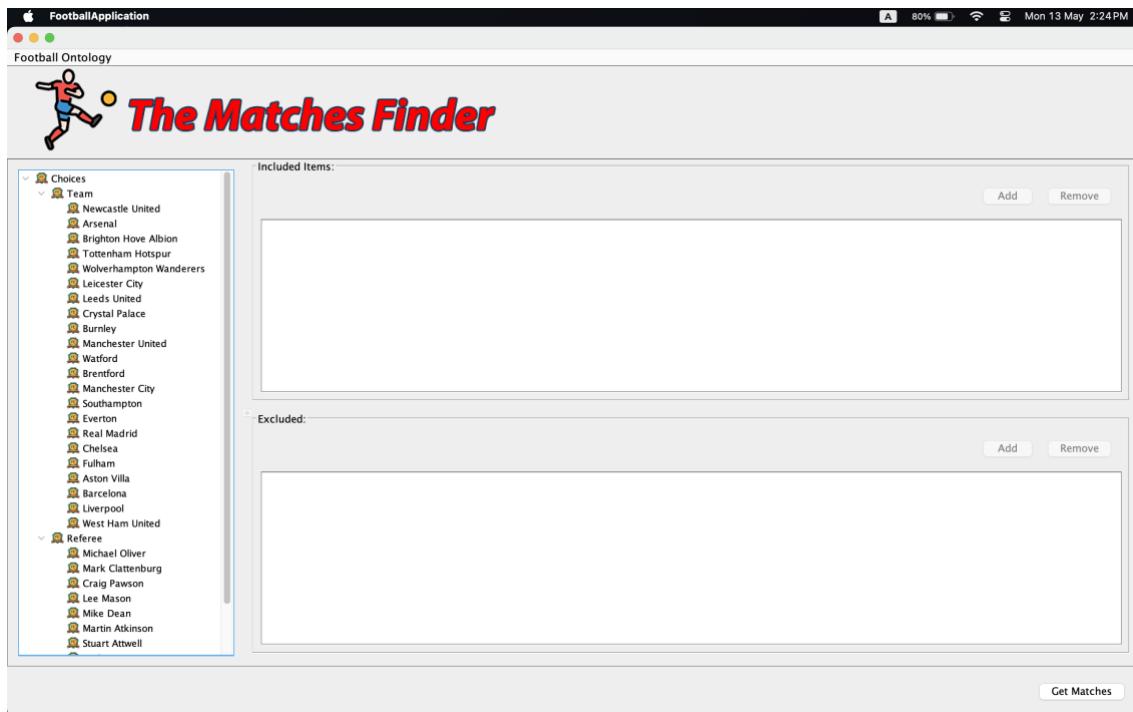
5.1 Core Components

5.1.1 Ontology Management

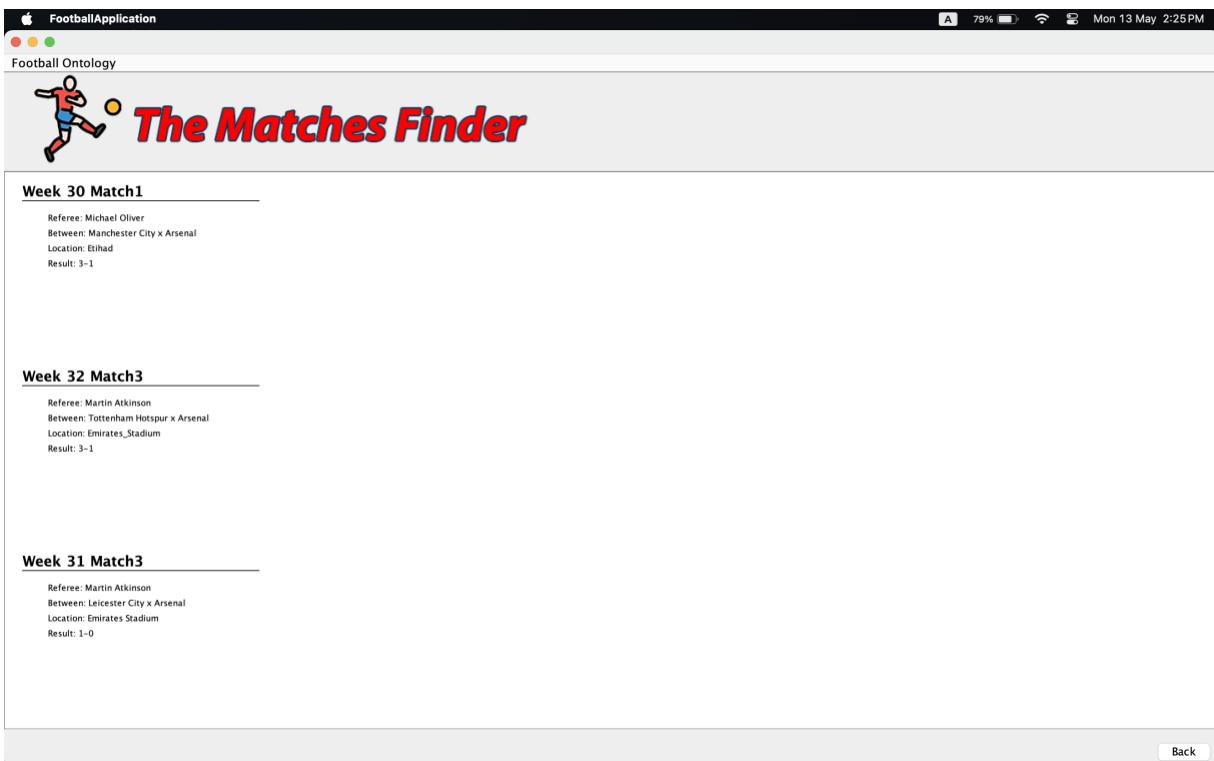
- **OWLAPI Integration:** The application integrates OWLAPI java package to manipulate and query the football ontology that contains detailed information about matches, teams, referees, championships, players, etc.
- **Pellet Reasoner:** Utilizes the OpenPellet reasoner package to infer relationships and validate constraints within the ontology, ensuring that the displayed matches adhere to the selected filters.

5.1.2 GUI Components

- **Selection Panel:** Users select teams and referees to include / exclude from matches.



- **Matches Display Panel:** List the matches that meet the filtering criteria. Matches are listed with details such as date, location, participants, and result, referee, and title.



5.1.3 Filtering Logic

- **Dynamic Property Assignment:** Depending on whether a team or referee is being included or excluded, the application dynamically assigns properties within the ontology to filter matches. For example, properties like ‘betweenTeams’ or ‘hasReferee’ are used.
- **Flexible Querying:** Users can filter by multiple teams and referees simultaneously. The application supports complex queries, combining various inclusion and exclusion criteria.

5.2 Functional Workflow

1. **Initialization:** Upon launch, the application loads the ontology from the specified location of the “football.rdf” file and initializes the OpenPellet reasoner.
2. **User Interaction:** Users select teams and referees they wish to include or exclude from the matches and then presses the “Get Matches” button to trigger the query.
3. **Query in action:** The selected teams and referees triggers a query that utilizes the reasoner to filter out matches based on the specified criteria.
4. **Results Display:** The filtered matches details are listed in the Results Panel.

5.3 Data Flow Diagram (DFD)

The DFD illustrates how data is handled and processed within the system, including the interactions between different system components. It highlights the flow of data between the user interface, the ontology manager, and the reasoner, detailing the operations performed at each stage.

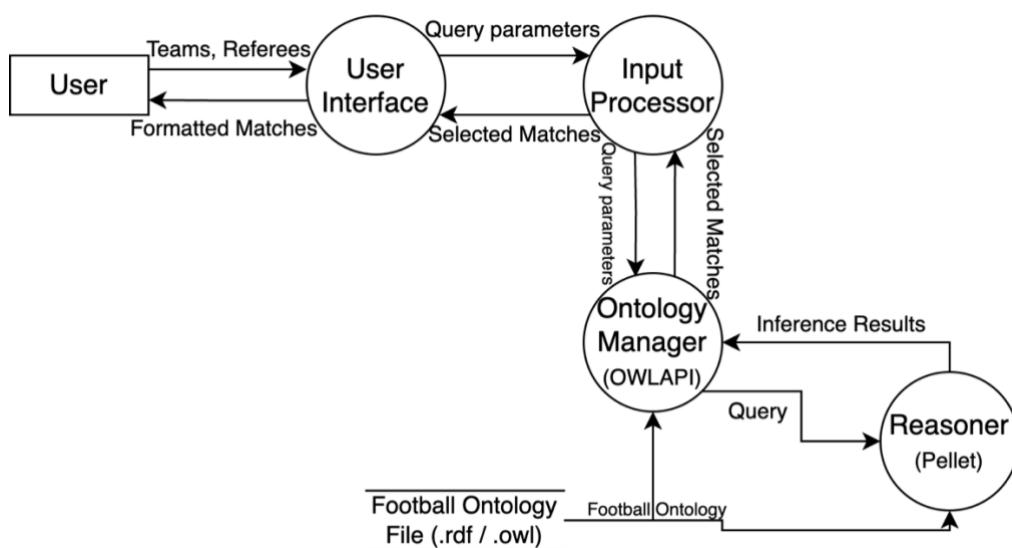


Figure 42: Java Application DFD

5.3.1 System Components

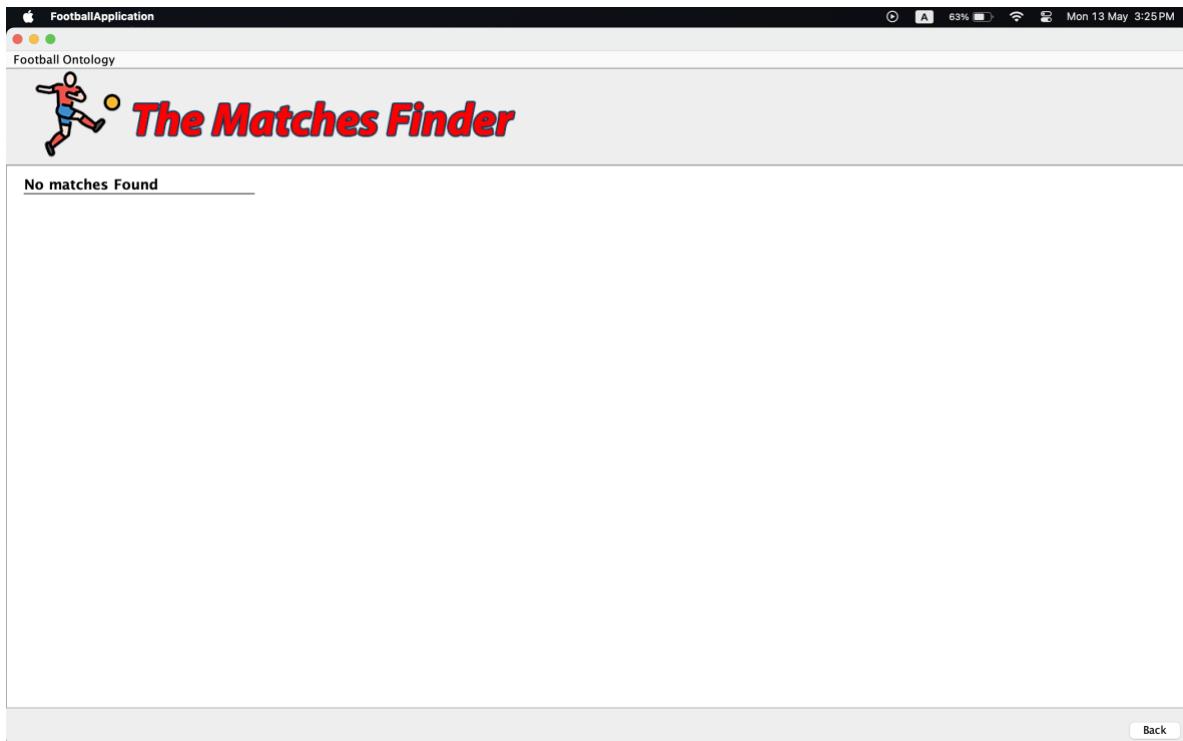
- **User Interface (UI):** It serves as the interaction layer for end-users. It includes various graphical components that allow users to specify their preferences for included or excluded teams, or referees. It also displays the results of matches that meet these criteria.
- **Ontology Manager (OWLAPI):** It functions as the mediator between the UI and the Reasoner. It handles the processing of user inputs, formulates queries based on the user's filtration choices, and interprets the results received from the Reasoner to be relayed back to the User Interface.
- **Reasoner (OpenPellet):** The Reasoner is tasked with logical inference and reasoning over the ontology. It uses the data within the ontology to deduce new information, verify consistency, and respond to queries initiated by the Ontology Manager.

5.3.2 Data Flows

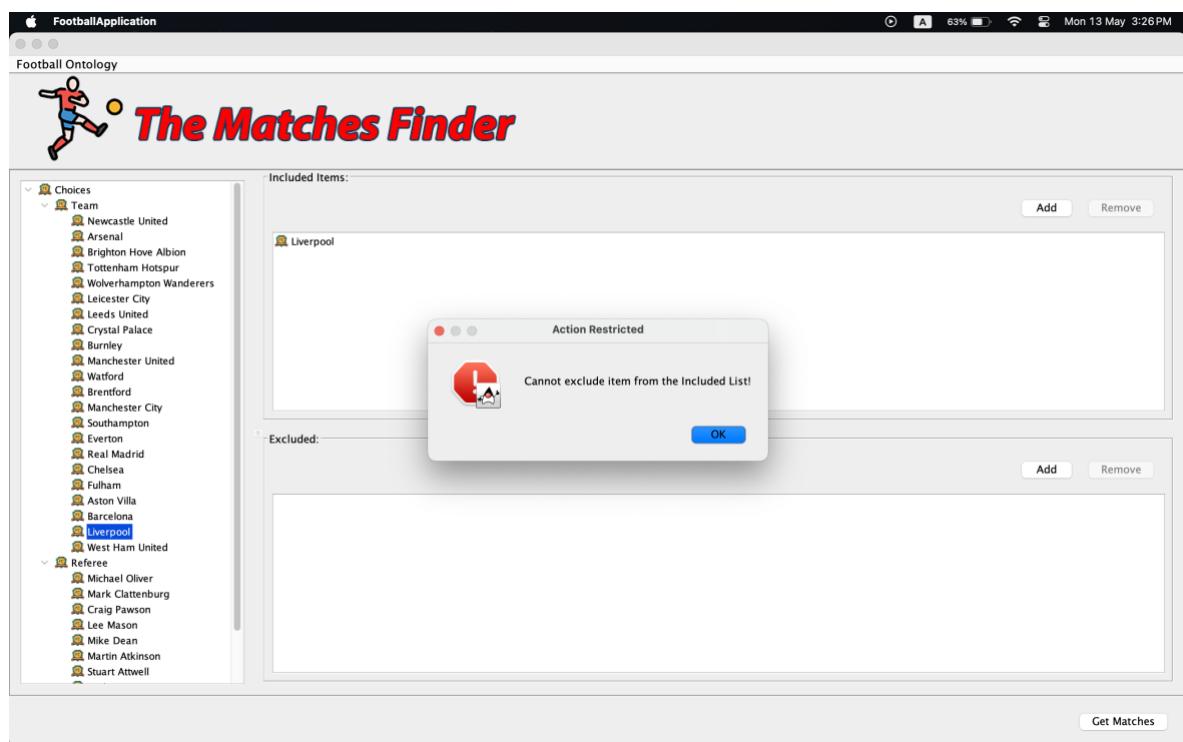
- **User Inputs:** Inputs received from the user, such as selections of teams, or referees, to include or exclude from match results.
- **Query Requests:** Requests sent from the Ontology Manager to the Reasoner, asking it to process specific queries for reasoning or consistency checks.
- **Inference Results:** The results returned from the Reasoner to the Ontology Manager. These include the outcomes of queries and the status of the ontology's consistency after any updates.
- **Match Results Display:** The formatted results of the matches, processed by the Ontology Manager, which are then displayed to the user through the User Interface.

5.4 Sample Tests

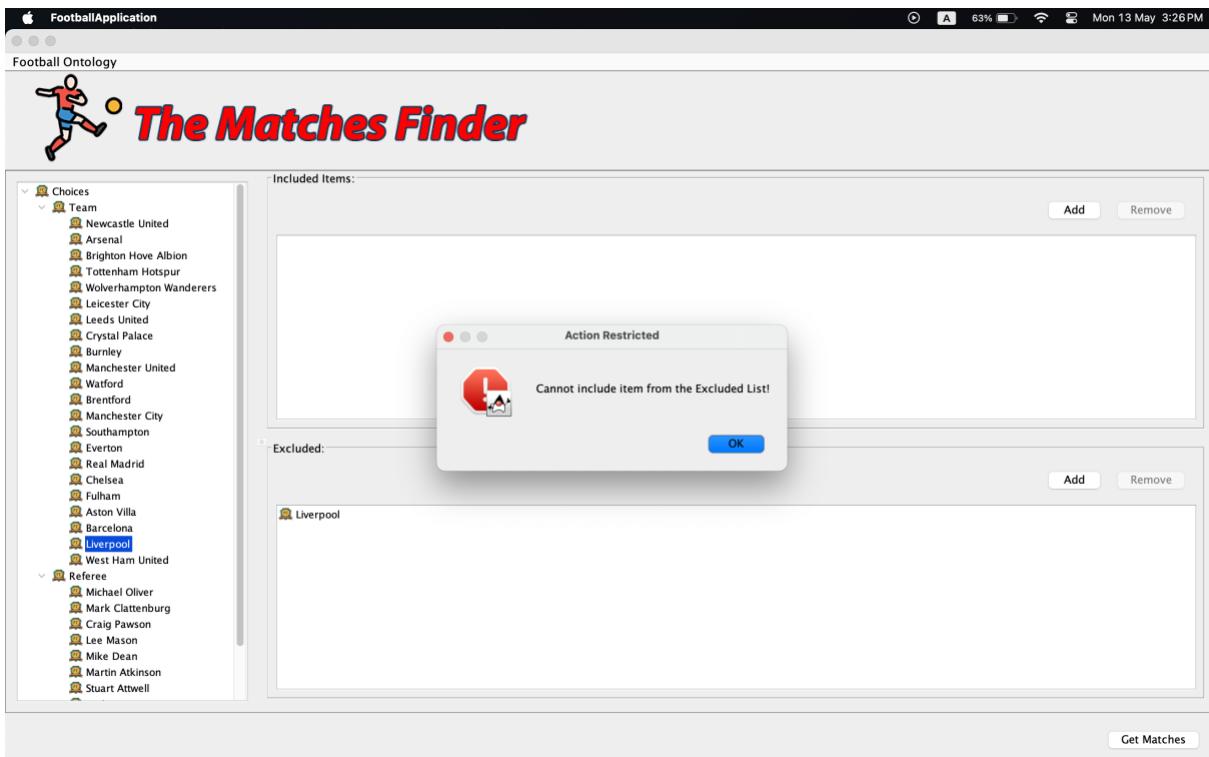
- No included or excluded teams/referees before pressing “Get Matches” button.



- User attempts to exclude a team he already added to the included list.



- User attempts to include a team he already added to the excluded list.



- User searches for “Liverpool”, “Real Madrid” matches that doesn’t include “Michael Oliver” as a referee.

FootballApplication

Football Ontology

The Matches Finder



Included Items:

- Liverpool
- Real Madrid

Excluded:

- Michael Oliver

Choices:

- Team**
 - Newcastle United
 - Arsenal
 - Brighton Hove Albion
 - Tottenham Hotspur
 - Wolverhampton Wanderers
 - Leicester City
 - Leeds United
 - Crystal Palace
 - Burnley
 - Manchester United
 - Watford
 - Brentford
 - Manchester City
 - Southampton
 - Everton
 - Real Madrid
 - Chelsea
 - Fulham
 - Aston Villa
 - Barcelona
 - Liverpool
 - West Ham United
- Referee**
 - Michael Oliver
 - Mark Clattenburg
 - Craig Pawson
 - Lee Mason
 - Mike Dean
 - Martin Atkinson
 - Stuart Attwell

Get Matches

FootballApplication

Football Ontology

The Matches Finder



Classico

Referee: Hernandez Hernandez
 Between: Barcelona x Real Madrid
 Date: 2022-03-11T22:00:00
 Location: Camp Noe
 Result: 2-0

Week 30 Match2

Referee: Anthony Taylor
 Between: Liverpool
 Location: Anfield
 Result: 2-0

Week 31 Match2

Referee: Anthony Taylor
 Between: Liverpool x Manchester United
 Location: Anfield
 Result: 3-2

Back

- Get all matches where “Anthony Taylor” is the referee.

FootballApplication Football Ontology

The Matches Finder

Included Items:

- Anthony Taylor

Add Remove

Excluded:

Add Remove

Get Matches

FootballApplication Football Ontology

The Matches Finder

Week 32 Match2

Referee: Anthony Taylor
Between: Manchester United x Chelsea
Location: Old_Trafford
Result: 1-0

Week 30 Match2

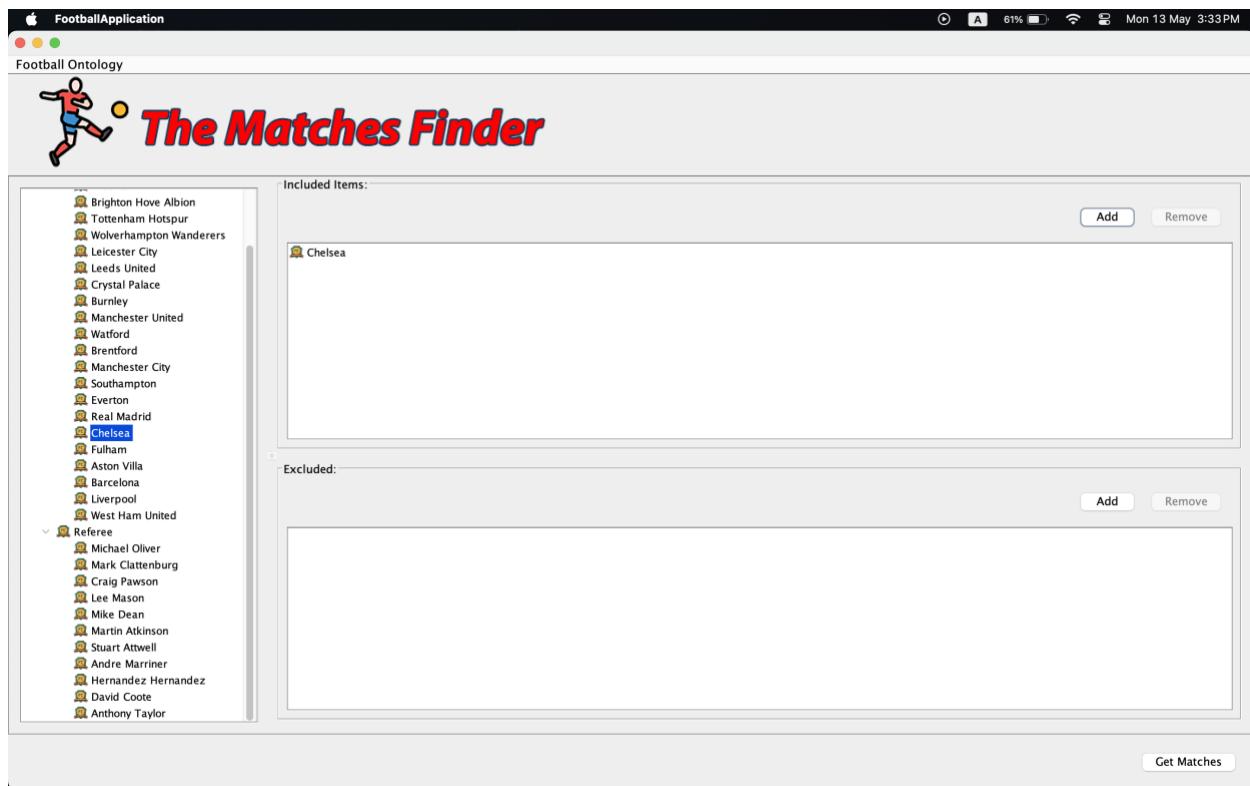
Referee: Anthony Taylor
Between: Liverpool
Location: Anfield
Result: 2-0

Week 31 Match2

Referee: Anthony Taylor
Between: Liverpool x Manchester United
Location: Anfield
Result: 3-2

Back

- Get all Chelsea matches.



Week 32 Match2

Referee: Anthony Taylor
Between: Manchester United x Chelsea
Location: Old_Trafford
Result: 1-0

Week 31 Match1

Referee: Michael Oliver
Between: Manchester City x Chelsea
Location: Etihad
Result: 2-1

Week 30 Match3

Referee: Martin Atkinson
Between: Chelsea x Tottenham Hotspur
Location: Stamford Bridge
Result: 2-2

Back