iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.47222

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Semantic Web Application: Tourist Assisting Web Application

Abhyas Mall[1], Shoibam Amritraj[2], Nitish Hans[3], J Rakesh[3]
[1, 2, 3, 4]*B.Tech, CSE, SRM Institute of Science & Technology, Kattankulathur, Tamil Nadu, India*

Abstract: ***In addition to traditional web technologies, semantic web technology allows the implementation of the "web of things" and "web of linked data," enabling both machines and humans to access the web. The ultimate objective is to allow machines to do increasingly complex tasks and to design systems that can support network interactions. This goal may be realized by merging several linked data technologies, such as RDF, SPARQL, OWL, and SKOS (W3C Org, 2014). In this project, the authors integrated semantic web technologies into the software to illustrate their use. This study describes in detail the design and functionality of the apps, as well as how the semantic web technology principles were implemented. The authors followed the principles and best practices outlined in W3C documents and directed by the first author. The authors exerted a great deal of work to evaluate the acquired data and have begun the project by locating suitable datasets from various data sources (heterogeneous data sources). Following this, more attempts were made to cleanse and verify the data. This generated dataset was subsequently pushed to the selected triple store on application start-up. The ontology design and project architecture are described in full in this study. The project's stack and implementation has been discussed in this report.***
Keywords: JSON-LD, Apache-Jena, Graph-DB, Portege, Semantic Web, Ontology, RDFs, OWL.

I. INTRODUCTION

The application is able to provide information to travelers that are interested in travelling around France. Travelers may search various cities depending on their interests, and the program will show available railway stations, bike stations, and hospitals along with other relevant information. The data on the railway station is rather thorough, and it includes information such as the schedule of a specific station, as well as the position and direction of the station. Through the use of the SNCF API gateway, the data on the timetables are obtained in real time. In addition, the data collected from bike stations display, in real time, the number of bikes that are accessible at each bike station as well as the precise position of each station. The users will now be aware of which bike station they need to go to in order to either pick up a bike or put one off in order to continue their journey. Because it is required to know which vacant slot (rack) there is at the station in order to leave a bike there, the station is called a rack. The hospital data is shown depending on the city that was searched for, along with some helpful information such as the specialty division of a hospital, its location on a map, contact details, and the hospital that corresponds to the head hospital. On the map, you can see all of this information along with the icons that correspond to it. The location of the user will initially be shown on the map, which will do so depending on the location service of the client device. In the event that the client device does not have a location service installed, the information provided by the nearby service providers will be used to make an educated guess as to the user's position. In addition, the travelers will be provided with access to real-time weather information, which will be of use to them when it comes time to organize their journey.

II. LITERATURE OVERVIEW

Information for tourists may often be found on the official websites that are maintained by tourism organizations. Tourists, on the other hand, are often confronted with challenging challenges in terms of locating the information that they need as a result of the sheer amount and variety of material. It is required to build a system that will guarantee that the data is kept in a form that is comprehensible to machines in order to decrease the disintegration of data. This is a need in order to prevent the loss of data. If you do this, you will guarantee the integration of tourist resources, which will lead to the modification of portal content to end users in order to make it easier for you to discover information. Utilizing the technologies of the Semantic Web is required in order to ensure that the data is saved in a format that can be read and understood by machines. Because they are at the center of the Semantic Web, ontologies have the potential to enhance the quality of information searches in a way that is more responsive to the needs of end users.

©IJRASET: All Rights are Reserved

Following is an analysis of several studies that discuss the role that semantic technologies play in enhancing the quality of e-business in the field of tourism. The primary focus of the author in [10] is to generate the ontology for the state of Tamil Nadu in India, which will include sufficient knowledge of tourism. This is done so that users can find the desired destination by specifying various search criteria such as housing, costs, activities, shopping, and so on. A user profile is made with the intention of gleaning information about the person, as well as their requirements and pursuits. The JENA interface is used to save the data that is put into the form in the database. This data is then used to build up searches over the DL ontology in order to acquire the results that are wanted. The Travel Guides online platform is detailed in the article [11], which tries to unify all of the arrangements from numerous travel agents into one centralized location. The PROTON ontology was used in order to meet the requirements of this system. This ontology is not specific to any one area and has over 300 classes and 100 characteristics. It is quite straightforward to extend, as it incorporates concepts that stand in for real-world terminology like locations, organizations, and people, amongst others. Given site demonstrates how the Semantic Web may affect the growth of interoperability in electronic tourism, how to give superior user interaction and systems, and how to make intelligent reasoning available in tourist. Java is the programming language that is used to build the system. A web framework, such as Struts or Hibernate, and the JENA are used for interface with the database. A semantic search for hotels all across the globe is provided by the Reisewissen project, which was established in [12]. This method improves the overall quality of tourism services while also optimizing the search process. The material was organized by the authors into two primary ontologies. The meaning of a diverse selection of hotels is semantically represented by using one as the foundation. This ontology comprises not just ideas for expressing general and contact information, but also information about the costs associated with staying at the hotel. Other ontologies include descriptions of people and the topics that interest them. In addition, the authors constructed three sub ontologies in order to define the qualities of the hotel, the geographic information around the hotel, and the various modes of transportation that may be used to go to the hotel. Java was employed in the development of the site, and JENA is being put to use in the deployment of semantic technology. The majority of travel agents have a significant obstacle when it comes to providing the necessary information. A mobile application known as STAAR was built and described in the publication [13]. The use of ontologies as a means of knowledge representation and the provision of travel-related information are at the heart of this system, which was developed with the intention of establishing connections with linked data repositories. On the basis of an ontology, the system that is described in [14] automatically generates a package that makes use of the data that has been previously annotated. The terms "annotation" and "metadata" are often contrasted with one another. This offers a service that, depending on the user's preferences, selects the appropriate mode of transportation. The author is of the opinion that the criteria for tourism are not well defined, and that the ontology has the potential to make up for this shortcoming. The answer that the Semantic Web offers in order to ensure interoperability is not yet fully defined, but the semantically based information systems are unquestionably ushering in a new age in the tourist industry. Within this process, ontologies play a very significant role. They may increase the quality of representation and search data, as well as establish the ideas of tourism and their interrelationships. Through the use of and collaboration on a shared lexicon, interoperability will be made possible thanks to the ontology.

III.ONTOLOGY ARCHITECTURE

The knowledge network of the project is explained in the ontology diagram that can be seen below in the Figure 1. The city, the bike stations, the train stations, the hospitals, and the weather were the five primary entities. In addition to this, the project has a schedule object for railway stations that is labelled as a blank-node since it was originally intended to be a blank-node in the RDF graph (Blank nodes does not contain any IRI). Within the context of the diagram, these entities are shown as having an oval form. The city was the primary connection between all of the entities. This demonstrates that the connected data is created with cities as the primary focus most of the time. The edges represent the attributes that are associated with each item. In the RDF network, the rectangular representations are nothing more than literals. Each edge illustrates the connection that exists between entities and characteristics, in addition to the relationships that exist with other entitles. In the RDF graph, they are referred to as predicates.

IV.STACK

A. Front-End Technologies

1) *Angular 10:* In order to build the client-side web interface, the authors made use of the Angular framework. This framework provides comprehensive support for TypeScript.
2) *Bootstrap:* Additionally, the authors built the user interface using components that have a decent appearance by using Bootstrap.
3) *Webpack:* In the end, the project utilized webpack in order to bundle the JavaScript application. This piece of software will handle the "typings," "uglifying," and "minifying" processes for you.

4) *JSON-LD:* JSON-LD is a small package that makes it easier to create webpages using the Linked Data Format in HTML. JSON, which is now the most prevalent content format, serves as the foundation for the whole thing. People now have the ability to build Linked-Data on the web thanks to JSON-LD. Rich content extraction and search engine optimization are two further purposes that search engines put this to use for. (JSON-LD org, 2018)

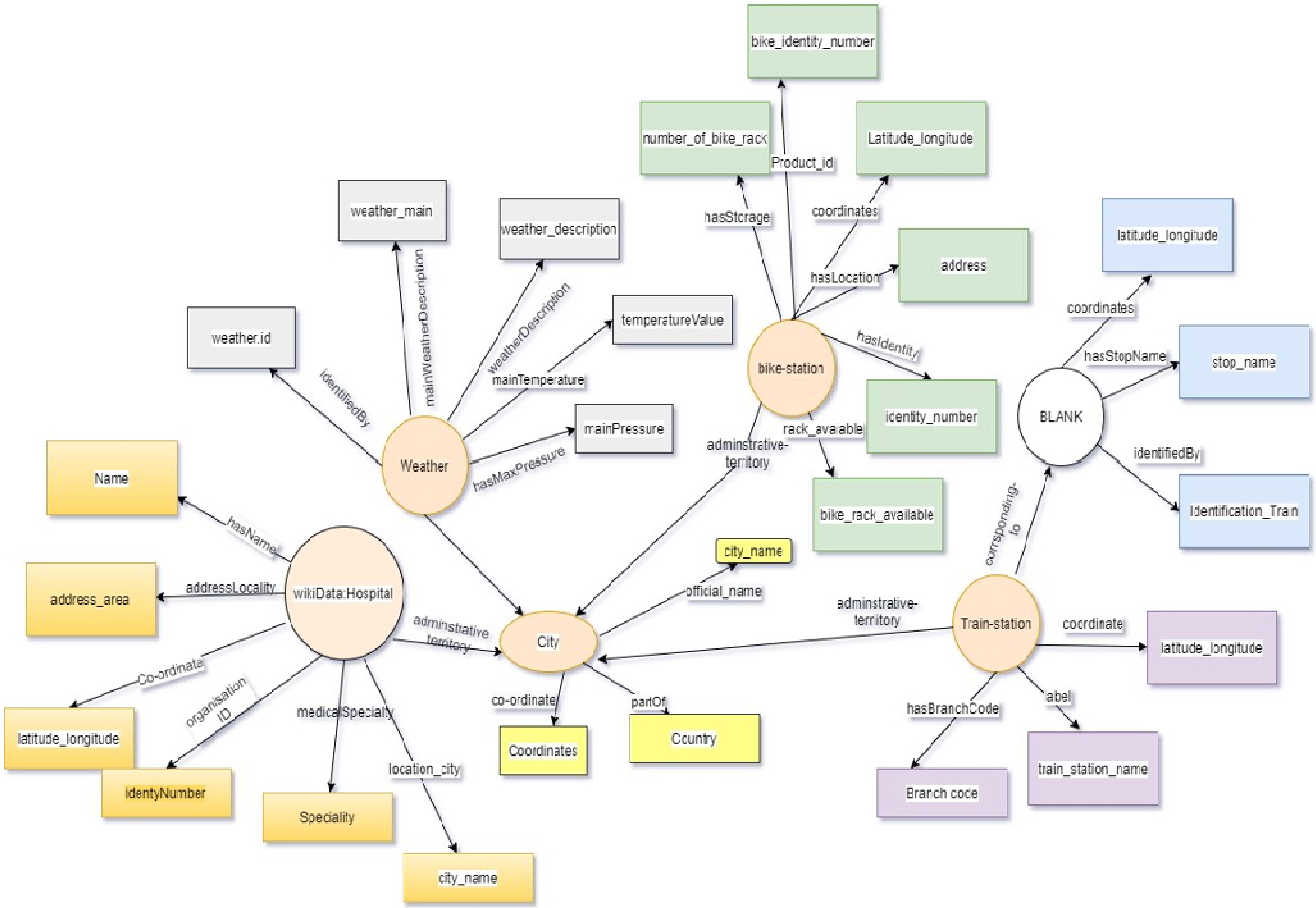


Fig. 1. Ontology Architecture



Fig. 2. Stack of the Project

B. Web Server Technologies

1) *Java –Spring Boot:* Developing server applications that leverage the Rest API is accomplished using Spring boot. This is a framework that was designed using Java as its primary programming language, and it offers pre-set implementations of the REST API.
2) *Swagger:* The API Documentation that the authors provide is generated using Swagger (and can be seen at http://localhost:8080/swagger-ui.html).
3) *Apache-Jena:* Apache Jena is a free and open-source Java framework that enables capabilities to build semantic web and linked-data applications. The library was developed by the Apache Software Foundation. (Apache jena org, 2020).

C. Triple Store

Graph-DB: Ontotex Graph DB is a triple store graph database that also supports RDF and SPARQL. It is the most efficient graph database available. It comes equipped with an integrated SPARQL endpoint, which makes it possible for users to query RDF data by using SPARQL. GraphDB gives users the ability to store RDF data in a variety of formats, including turtle, N-Triples, and N-Quads, among others (GraphDB organisation, year 2020).

D. Portege

Protege is an ontology editing tool that is free and open source, and it enables users to develop OWL ontologies. Using Protege, the authors were able to construct ontology-based applications that are both straightforward and comprehensive. It offers a more intuitive graphical user interface, which makes the process of creating and editing ontologies much easier. With the help of this programme, the authors were able to construct their very own ontology network using Web Ontology Language (OWL). (The Protégé squad in the year 2020)

E. GitHub

Managing the application's source code has been done via the usage of GitHub, which is a mechanism for controlling versions. GitHub maintains a record of all the alterations that have been made to files as well as the history of the various versions of the code files. The developers are able to work together, submit their ideas, and seamlessly incorporate their contributions into a centralised repository of the project (BROWN, 2019).

There are three different repositories that have been developed for this project.

1) *Server-Application:* This repository has the Java code that is used by the "Web-Server" application.
2) *Front-App:* The code for the front end of the angular application is stored in this repository.

a) *Server-Application:* Server application is a Java application that is responsible for storing data in triple store and retrieving it as well as handling client requests and replies that have been sent from the user interface (Front-ap). This application has two primary responsibilities to fulfil. The first goal is to extract RDF data from the triple store based on the search criteria and return it to the front app. The second duty is to prepare RDF data and save it in the triple store. In the area of this study devoted to the implementation, the functions were examined in further detail.

b) *Front-App:* An integrated application with a graphical user interface that has been constructed on the client-side utilizing Angular framework is referred to as a front-app. Angular is a library written in JavaScript that makes it possible to construct front-end Single Page Application (SPA) programs. This framework provides comprehensive support for TypeScript. Additionally, the authors built the user interface using components that have a decent appearance by using Bootstrap. In conclusion, the authors bundled the JavaScript application with the help of a webpack. This tool will handle the "uglifying" or "minifying" procedure as well as the construction of the front-end application.

F. HTTP, WS (Web Socket)

HTTP, WS (WebSocket) Communication between the "front-app" and the "server application" is carried out using the standard protocol known as HTTP. In extended use cases, in addition to HTTP, the WebSocket API is also used. Instead of having a stateless communication channel, this WS protocol enables to open up and provide interactive communication channels in both directions (HTTP are stateless communication). With the help of this particular communication protocol, it is possible to transmit an asynchronous answer from the server to the client at any moment. (MDN contributors, 2020)

V. SEQUENCE DIAGRAMS

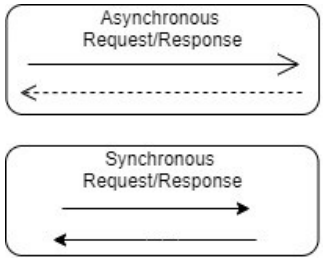


Fig. 3. Request reply arrows

A. Methodology to Search City

The graphic that follows illustrates the flow of functional calls and the execution of the city search procedure. When a user enters a city's name into the search bar of the front-end application and then hits the search button, this flow of the procedure is initiated. A query parameter containing the city name will be sent to the server application in the case of an asynchronous request (Asynchronous requests are shown in open arrowhead- [2]). After receiving the request along with the city's name, the web server will continue to query GRAPHDB using the SPARQL API ([3]). After that, the results will be sent back to the server application from the GraphDB. The RDF graph will be processed further by the server application, which will then construct a suitable object list, and the result will be sent to the front app, where it will be shown in the graphical interface.

The search function of an entity representing a railway station may be seen in the sequence diagram that was just shown. This exact same series of functional steps will be used to the search for additional entities, such as hospitals and bike stations.

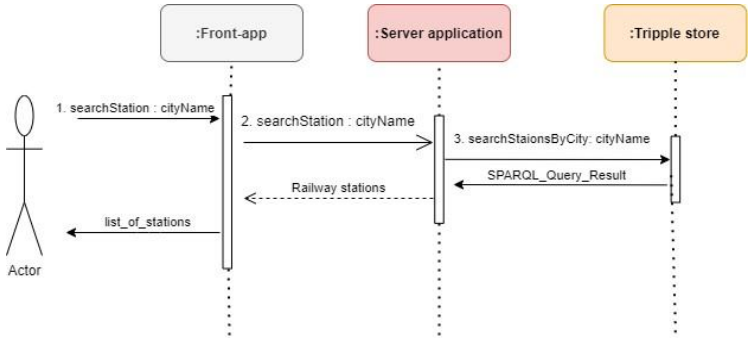


Fig. 4. Sequence diagram to query railway station

B. Methodology to Inquire Railway Station Timetable

The Timetable API service's sequence diagram is shown in the figure that may be seen below. The "showTimeTable" method will be invoked [1] when the user clicks on a specific railway station on the table. This method will send an asynchronous get request to the SNCF API gateway, providing that portal with the branch code of the station that was chosen [2]. This single station's schedule will be returned by the SNCF service (not the schedule for all stations) [3]. The user will be presented with the result that was obtained. After then, the findings will be sent to the programme running on the server. This information is going to be saved in the GraphDB as an RDF Blank-node [6]. If everything works out as planned, the server application will provide the front application with a response with the status code 200 [7].

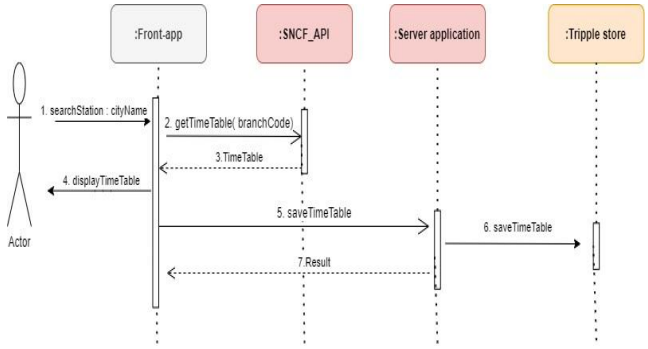


Fig. 5. Sequence diagram for real-time timetable request

C. Methodology to Query Bike Stations real Time Data

The API service sequence diagram for the Rennes bike station is shown in the graphic that may be seen below. The "getAvailability" method will be activated as soon as the user searches for the city of "Rennes" [2]. Using this method will cause an asynchronous GET request to be sent to the Bike station API gateway [2]. The application programming interface for bike stations will provide data on those stations in real time, including the number of bikes available for hire at each of the several bike stations located across the city of Rennes [3]. The received results are going to be forwarded to the application running on the server [4]. This information is going to be saved to the GraphDB in the form of an RDF Blank-node [5].

If everything works out as planned, the server application will provide the front application with a response with the status code 200 [6]. Then, the front app will once again send an asynchronous GET request to the server application [7]. The "Rennes" city's real-time data will be retrieved from GraphDB by the "Rennes" server application, which will then provide the results. The request will be handled by the programme running on the server, which will then construct the relevant list of items. Then, the front app will get this list after it has been processed [10]. The result will be shown in table [11] by the front end app.

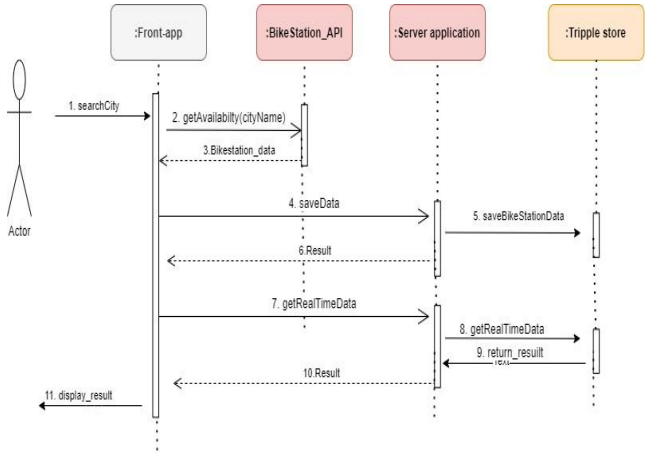


Fig. 6. Sequence diagram for real-time bike station data

D. Methodology to query Weather Data

The sequence diagram of the Weather API service is shown in the graphic that may be seen below. The "getWeather" method will be activated [2] whenever the user searches for a certain city. The city coordinates and the city name will be sent in an async get request to the Weather API gateway by this function [2]. The Weather API will provide the current weather conditions as well as an accurate prediction for that specific city [3]. The user will be presented with the result that was obtained. After then, the findings will be sent to the programme running on the server. This information is going to be saved in the GraphDB as an RDF Blank-node [6]. If everything works out as planned, the server application will provide the front application with a response with the status code 200 [7].

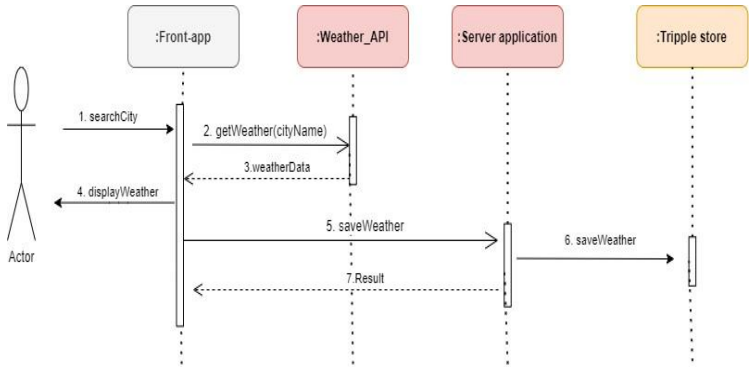


Fig. 7. Sequence diagram for real-time weather data

E. Methodology to Create a new city RDF Data

Using the graphical user interface, the user would be able to generate fresh data for the city. The user will be asked to enter various data depending on the text fields that are provided, such as the name of the city, the city's IRI, the city's coordinates, and so on. This information will be sent to the programme running on the server [2]. The RDF graph will then be created by the server application based on the data that was provided, and it will be stored in the GraphDB [3]. If everything works out as planned, the server application will reply with a status code of 200, and the user will be presented with a message congratulating them on their achievement.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

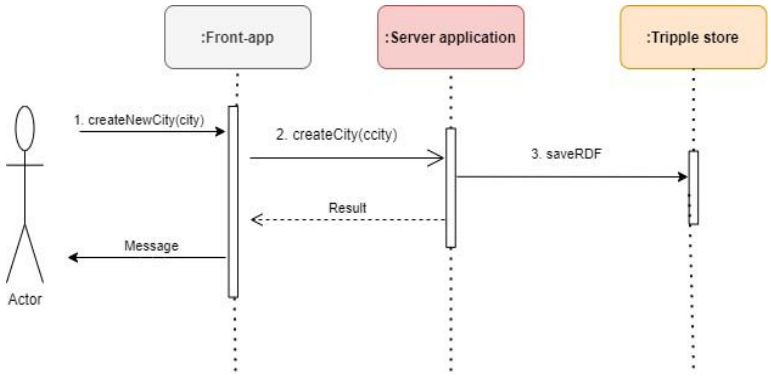


Fig. 8. Sequence diagram to generate RDF data from GUI

VI. IMPLEMENTATION OF THE WEB ONTOLOGY LANGUAGE (OWL): THE BACKBONE OF SEMANTIC WEB

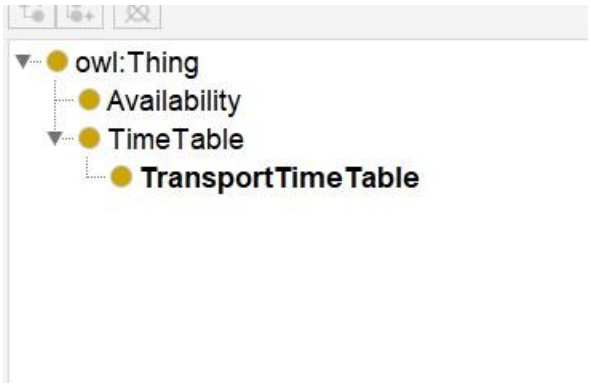


Fig. 9. Ontology Schema: classes

Through the use of the Protege tool, the authors were able to define three different owl classes for the project. The "TimeTable" class has a subclass called the "TransportTimeTable." This class includes the attributes that are unique to the transit timetable, whereas the "Timetable" class contains the features that are relevant to general timetables.

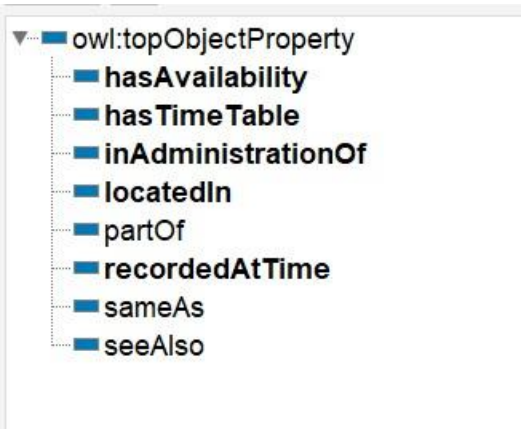


Fig. 10. OWL: Object properties

In the authors custom ontology, these are the object attributes that have been specified. Each attribute is associated with a distinct schema class. The ontology file that was supplied may be used to have access to all of the available information on these attributes. These object attributes are used to establish a linkage between two "Entities." [Creative Commons] (Subject and Object both are entities).

©IJRASET: All Rights are Reserved

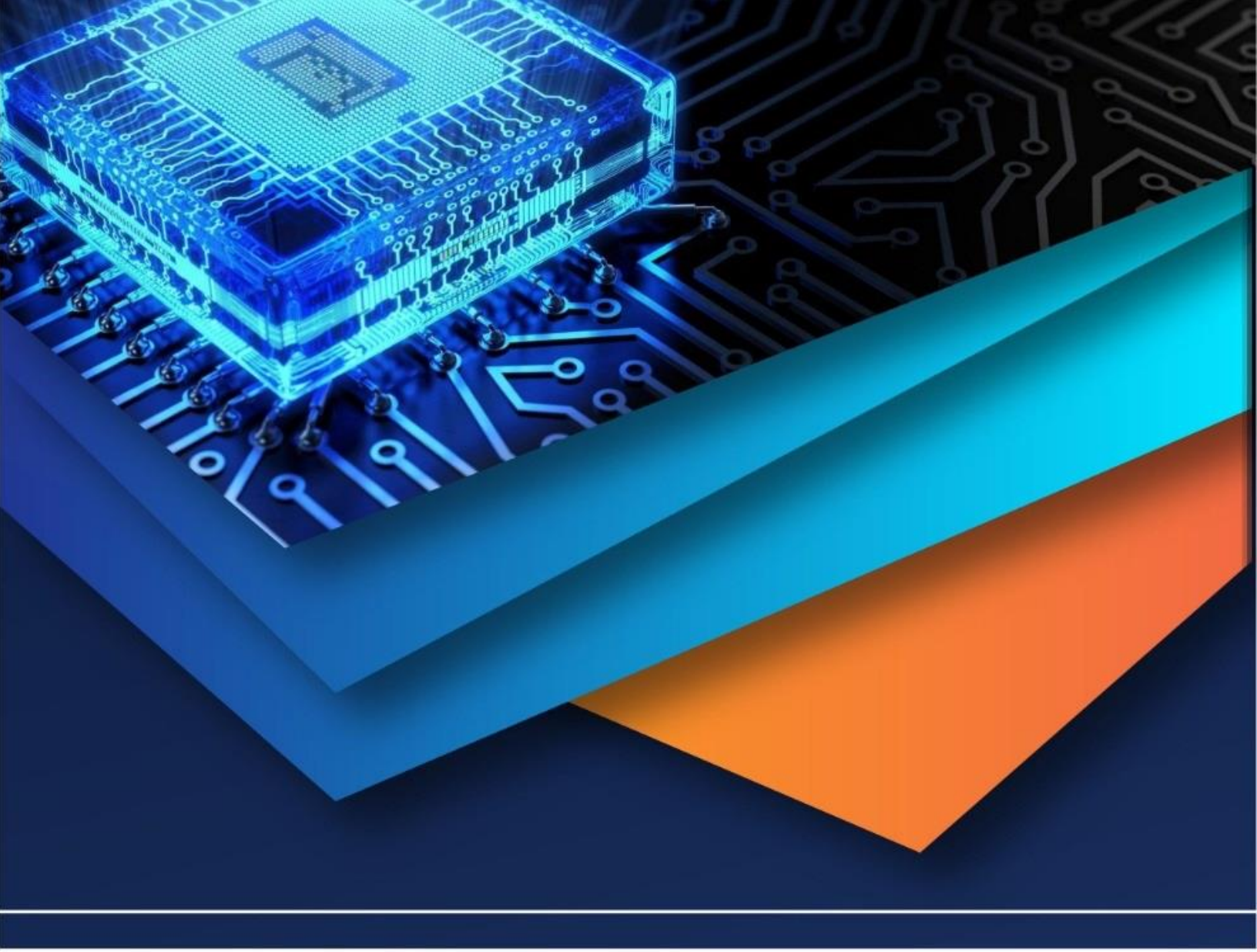


Fig. 11. OWL: Datatype properties

The authors specified a variety of data type attributes, each of which assists in establishing a connection between "Entity" (the subject) and "Literals" (Object).

REFERENCES

[1] JSON LD and Angular medium blog: https://medium.com/javascript-in-plain-english/how-to-usejson-ld-for-advanced-seo-in-angular-63528c98bb91
[2] Apache Jena to GraphDB documentation: https://graphdb.ontotext.com/documentation/free/usinggraphdb-with-jena.html
[3] Apache jena org. (2020). Apache jena. Retrieved from Apache jena : https://jena.apache.org/
[4] BROWN, K. (2019, 11 13). What Is GitHub, and What Is It Used For? Retrieved from How to geek: https://www.howtogeek.com/180167/htg-explains-what-is-github-andwhat-do-geeks-use-it-for/
[5] GraphDB org. (2020). GraphDB General. Retrieved from Ontotext GraphDB: https://graphdb.ontotext.com/documentation/free/index.html
[6] JSON-LD org. (2018). JSON for Linking Data. Retrieved from JSON-LD: https://jsonld.org/
[7] MDN contributors. (2020, 03 01). The WebSocket API (WebSockets). Retrieved from MDN Web docs: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API 8. Protege team. (2020). Protege. Retrieved from Protege: https://protege.stanford.edu/
[8] Protege team. (2020). Protege. Retrieved from Protege: https://protege.stanford.edu/
[9] W3C Org. (2014). SEMANTIC WEB. Retrieved from W3C: https://www.w3.org/standards/semanticweb/#:~:text=The%20term%20%E2%80%9CSemantic%20Web%E2%80%9D%20refers,SPARQL%2C%20OWL%2C%20and%20SKO S.
[10] Ananthapadmanaban, K R, Srimathi, H. and Srivatsa S. K., Tourism Information System- Integration and Information Retrieval of Tourism Information Systems using Semantic Web Services. International Journal of Computer Applications 52(14):13-20, Published by Foundation of Computer Science, New York, USA, August 2012.
[11] Damljanović, D., Inteligentni web portal u oblastiturizma, Magistarskateza, Fakultetorganizacionihnauka, Univerzitet u Beogradu, 2007.
[12] Niemann, M., Mochol, M. and Tolksdorf, R., Enhancing hotel search with semantic web technologies, Journal of Theoretical and Applied Electronic Commerce Research, v.3 n.2, p.82-96, August 2008
[13] Tuan-Dung, C., Thanh-Hien, P. and Anh-Duc, N., An Ontology Based Approach to Data Representation and Information Search in Smart Tourist Guide System, kse, pp.171-175, 2011 Third International Conference on Knowledge and Systems Engineering, 2011
[14] Cardoso, J., Developing Dynamic Packaging Systems using Semantic Web Technologies, Transactions on InfromationSceince and Applications, Vol. 3(4), pp. 729736, ISSN:1970-0832, April 2006.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)