Echipa:

Aghebanoae Oana

Andries Simona

Gabor Silvia

Minea Iuliana

**1.   Introduction**

The purpose of ”Backpacker Recommender” is to help the backpackers who travel a lot through foreign countries, often get lost or don't know how to get around in an unfamiliar place.1

The application offers the user the possibility to choose if he/she wants to search restaurant/hospitals/hotels nearby its current location or another selected city.

The application offers the possibility to monitor the travel progress. This includes the places visited by the user and the date when the user visited it.

The user can evaluate a visited place by rating it through the application. The rating will be taken into consideration when offering recommendation for a specific type of locations.

**2. Used technologies**

**2.1  Web API**

Asp.Net Web API is a framework for building HTTP services that can be consumed by a broad range of clients including browsers, mobiles, iphone and tablets. It is very similar to ASP.NET MVC since it contains the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection. But it is not a part of the MVC Framework. It is a part of the core ASP.NET platform and can be used with MVC and other types of Web applications like Asp.Net WebForms. It can also be used as an stand-alone Web services application.2

In order to create the specification of the API we used RAML because it offers a simple and intuitive way to describe the resources, methods, parameters and responses. The API will be accessed using the following methods:

* /location
* /locations
* /locations/{id}
* /locations/restaurants
* /locations/hotels
* /locations/museums
* /locations/hospitals
* /locations/shops
* /locations/(restaurants/hotels/museums/hospitals)/{id}

**2.2  BrightstarDB**

BrightstarDB is a native .NET NoSQL semantic web database. It can be used as an embedded database or run as a service.

BrightstarDB supports the [W3C RDF](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/) and SPARQL 1.1 [Query](http://www.w3.org/TR/sparql11-query/) and [Update](http://www.w3.org/TR/sparql11-update/). standards, the data model stored is triples with a graph context (often this is called a quad store). 3

The RDF Client API provides a simple set of methods for creating and deleting stores, executing transactions and running queries. It should be used when the application needs to deal directly with RDF data. An RDF Client can connect to an embedded store or remotely to a running BrightstarDB instance.4

BrightstarDB is used to keep users preferences and the travel progress. When a recommendation is desired, BrightstarDB will be interogated via the Sparql endpoint in order to extract the current user’s preferences. Visited places with low rates  will be excluded from the list of recommended places.

**2.3  Google Maps API**

Google Maps API is used to geographically represent the places searched by the user and also to extract his current position.

**2.4  Data sources**

The data sources that we will use are listed below. They can provide information about major points of interest for the user by their SPARQL endpoints. The namaspace “geo” enables us to search by location latitude and longitude. This fulfills a major need of the application, since all the searches are based on geographical coordinates or based on the id of a resource.

* DBPedia
* El Viajero
* TourPedia
* LinkedGeoData

Query examples using dbpedia enpoint for retrieving different points of interest:

*Restaurants with Seafood cuisine*

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

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

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?name (STR(?cuisine) AS ?cuisine) ?address ?website

WHERE {

?f rdf:type dbo:Restaurant .

?f dbo:cuisine ?cuisine .

?f foaf:homepage ?website .

?f rdfs:label ?name .

?f dbo:address ?address

FILTER (langMatches(lang(?name), "EN") && (STR(?cuisine) AS ?cuisine)="Seafood") .

}

LIMIT 10



*Museums filtered by latitude and longitude*

PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>

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

PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?museum\_name ?lat ?long  WHERE {

?s a dbo:Place .

?s geo:lat ?lat .

?s geo:long ?long .

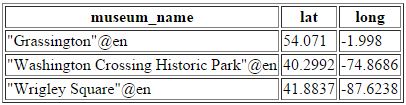
?s rdfs:label ?museum\_name .

?s dbp:museum ?museum .

FILTER ( ?long > 2.3508 - 100 && ?long < 2.3508 + 100 && ?lat > 48.8567 - 100 && ?lat < 48.8567 + 100 && langMatches(lang(?museum\_name ), "en"))

}

LIMIT 10



*Get hotels from linkedgeodata*

Prefix lgdo:<http://linkedgeodata.org/ontology/>

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

PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>

Select ?name, ?lat, ?long  where {

?f rdfs:label ?name .

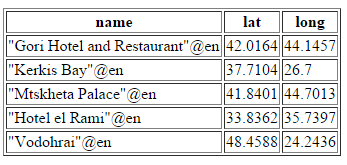
?f rdf:type lgdo:Hotel .

?f geo:lat ?lat .

?f geo:long ?long .

FILTER (langMatches(lang(?name), "EN"))

} Limit 20



**3.      Bibliography**

1. http://profs.info.uaic.ro/~busaco/teach/courses/wade/projects/

2.<http://www.dotnet-tricks.com/Tutorial/webapi/VG9K040413-What-is-Web-API-and-why-to-use-it-?.html>

3. <https://brightstardb.readthedocs.org/en/latest/Concepts/>

4. https://brightstardb.readthedocs.org/en/latest/RDF\_Client\_API/#rdf-client-api