עבודה 3.2

id	method name	http method	parameters	JSON returns	explanation
1	POI/getInfoPoint	GET	PointID	[{“id”,”name”, ”imagePath”, ”category”, ”views”, ”description”, “rating”}] [{“pointID”,”review”, “date”}]	The information related to the given point name. Each time the method is called the view field of that point is raised. Using GET because there is no information to deliver to the server.
2	Users/getUserQuestion	GET	Username	questions[]	The user question are stored in the DB, we need to access it from the server. Using GET because there is no information to change in the server-side.
	Users/getAllCategories	GET		Categories[]	In order to present the registered user all the categories he can choose from we need to access it from the server. Using GET because there is no information to change in the server-side.
3	Users/getAllQuestions	GET		Questions[]	In order to present the registered user all the questions he can answer we need to access it from the server. Using GET because there is no information to change in the server-side.
4	Users/getAllCountries	GET		Countries[{“id”,”name”}]	In order to present the registered user all the countries he can choose from we need to access it from the server. Using GET because there is no information to change in the server-side.
5	Users/passwordRetrival	POST	Username Answers[]	password	The user’s details are stored in the DB, so

					only the server can get it. Cannot pass in GET (not safe).
6	POI/getPopularPoints	GET	minimalRank	[{“id”,“name”, ”image_path”, ”category”}]	In the server side we filter the relevant Point of interest by the given rank (all that match the rank and above). Using GET because there is no information to change in the server-side.
7	Users/Login	Post	Username, Password	Token	Login cannot pass in GET (not safe)
8	UsersPOI /getSavedPoints	GET	Token, numOfPoints	[{“id”,”dateSaved“, ”sortingOrder”, “name”,”image_path”, ”category”}]	The saved point of a user are stored in the DB, so we need the server to access it, we use numOfPoints to allow flexibility. Using GET because there is no information to change in the server-side.
9	UsersPOI /getPopularPoints	GET	Token, numOfPoints	[{“id”,“name”,”image_path”, ”category”}]	We need to calculate in the server side the most popular point of interest for each category the user choose in the registration, the numOfPoint for flexibility. Using GET because there is no information to change in the server-side.
10	Users/addUser	POST	First name, last name, city, country, email, questions[], categories[], answers[], username, password		Use of POST since we insert to DB the given parameters
11	POI/getAllPoints	GET		[{“id”,“name”,”image_path”, ”category”}]	Points of interest are stored in the DB, the server need to retrieve

					them. Using GET because there is no information to change in the server-side.
12	UsersPOI /addSavedPoint	POST	Token, pointID		Using POST since we need to deliver the point of interest to the server for adding it to the DB.
13	UsersPOI /removeSavedPoint	DELETE	Token, pointID		Using DELETE because we are removing the point of interest from the user saved points in the DB.
14	UsersPOI /saveFavoriteOrder	PATCH	Token, [{“pointID”, “sortingOrder”}]		Use of POST since we insert to DB the given parameters to remember the order for future use (outside of the session).
15	UsersPOI/rankPoint	POST	Token pointID, rankNum, comment (optional)		We need to perform calculation of the new point rank and if we get a comment to add it to DB, all of this are actions that the server do, use of POST.
16	POI/getPopularPointsByCategory	GET	category	[{“id”,“name”,”image_path”, ”category”}]	Points of interest are stored in the DB, the server need to retrieve them. Using GET because there is no information to change in the server-side.