Part 1-

To allow other type of medical professionals, I need to add another parameter in the REST API which specifies location as well as profession. So, an example of such API will be -

http://localhost:8080/outpatient/physicaltherapy/default?location=<location name>&profession=<profession type>

Both these parameters will be mandatory parameters and it will return an error message, if any one of them is not entered while invoking the API.

Part 2 –

For both the questions in part 1, if I need to add a database in the backend, I will use a document store database such as MongoDB, or Elasticsearch.

1) For question 1 in part 1, the data will be stored in JSON format with the **location** being used as a key in JSON object to identify a entry along with four different key value pairs such as numberofpoints (int), pointswithratings (int), averagerating (float), reviewcount (int), location (varchar) will be present. The key **location** will be used to index each entry in the document to improve query retrieval speed.

Sample data –

{

"data": [{

"location": "atlanta",

"numberofpoints": 91.0,

"pointswithrating": 91.0,

"averagerating": 4.159340659340659,

"totalreviews": 366.0

}, {

"location": "seattle",

"numberofpoints": 91.0,

"pointswithrating": 91.0,

"averagerating": 4.159340659340659,

"totalreviews": 366.0

}]

}

2) For the second part also, the database will be indexed with the **location** parameter to improve query performance and the entire array object for a location will be stored under a node named ‘data’ for that location. Below is a sample database schema -

[{

"location": "seattle",

"data": [{

"name": "Leone's Chiropractic Accident & Injury Center",

"address": "7003 NW 11th PlGainesville, FL 32606",

"reviewCount": 8,

"rating": 5

}, {

"name": "ReQuest Physical Therapy",

"address": "4820 W Newberry RdGainesville, FL 32607",

"reviewCount": 1,

"rating": 1

}]

}, {

"location": "atlanta",

"data": [{

"name": "Leone's Chiropractic Accident & Injury Center",

"address": "7003 NW 11th PlGainesville, FL 32606",

"reviewCount": 8,

"rating": 5

}, {

"name": "ReQuest Physical Therapy",

"address": "4820 W Newberry RdGainesville, FL 32607",

"reviewCount": 1,

"rating": 1

}]

}

]

2) If we are integrating both the systems as a single unit, then the previous schema can be tweaked a little so that each item will have the location node and review node in it. Below will be the database schema –

{

"data": [{

"location": "seattle",

"name": "Leone's Chiropractic Accident & Injury Center",

"address": "7003 NW 11th PlGainesville, FL 32606",

"reviewCount": 8,

"rating": 5,

"review": "some review"

}, {

"location": "seattle",

"name": "ReQuest Physical Therapy",

"address": "4820 W Newberry RdGainesville, FL 32607",

"reviewCount": 1,

"rating": 1,

"review": "some review"

}, {

"location": "atlanta",

"name": "Leone's Chiropractic Accident & Injury Center",

"address": "7003 NW 11th PlGainesville, FL 32606",

"reviewCount": 8,

"rating": 5,

"review": "some review"

}, {

"location": "atlanta",

"name": "ReQuest Physical Therapy",

"address": "4820 W Newberry RdGainesville, FL 32607",

"reviewCount": 1,

"rating": 1,

"review": "some review"

}]

}