**Java Categories/Sets**

**NOTE: Please follow good software engineering practices.  In addition to a working program, unit tests should be produced. Treat this as if it was production code being delivered to a client. It should be properly packaged and delivered with sufficient documentation and instructions also provided. You must also describe how to deploy and monitor your service, as well as any changes that would be required to load balance your service.**

**Develop the following exercise in Java/Springboot.**

As part of the system architecture at a data processing company, you need to design a Service to clean data produced by another service (the client).

The data is a list of category sub-category pairs. For example, one set of data might be:

| **Category** | **Subcategory** |
| --- | --- |
| PERSON | Bob Jones |
| PLACE | Washington |
| PERSON | Mary |
| COMPUTER | Mac |
| PERSON | Bob Jones |
| OTHER | Tree |
| ANIMAL | Dog |
| PLACE | Texas |
| FOOD | Steak |
| ANIMAL | Cat |
| PERSON | Mac |

There is a list of valid categories managed by your service. By default, the valid categories are:

| **Category** |
| --- |
| PERSON |
| PLACE |
| ANIMAL |
| COMPUTER |
| OTHER |

When your service receives data from a client, it must process the data, removing duplicate (category, subcategory) pairs and invalid categories. The order of entries in the input data must be preserved, with the duplicates and invalid categories removed. The output must also include the count of entries for each valid category, sorted by the number of valid, unique entries.

Sample output for the sample input:

| **Category** | **Subcategory** |
| --- | --- |
| PERSON | Bob Jones |
| ANIMAL | Cat |
| ANIMAL | Dog |
| COMPUTER | Mac |
| PERSON | Mac |
| PERSON | Mary |
| PLACE | Texas |
| OTHER | Tree |
| PLACE | Washington |

| **Category** | **Count** |
| --- | --- |
| PLACE | 2 |
| PERSON | 3 |
| OTHER | 1 |
| COMPUTER | 1 |
| ANIMAL | 2 |

In addition to processing input data, your service must also provide the ability to add to, delete from, and list the valid categories. Once the category list is modified, subsequent processing requests will apply the new category list to the input data. While a real-world system would use a permanent datastore for the category information (so the current list of categories would be preserved if the service needs to be restarted), it is sufficient for this exercise to maintain the category list in memory.

Please implement a REST Service that provides the above functionality. It is up to you to define the input and output data formats, as well as the REST endpoints that are used for data processing and category management. You must also describe how to deploy and monitor your service, as well as any changes that would be required to load balance your service.