A. How to REQUEST data from the microservice

The StringValidator microservice closely follows RabbitMQ's tutorial on Remote Procedure Call (RPC) structure. For this program, that means a few things (most of which will carry over to the following section):

- RabbitMQ must be running locally
- Calling program must "import pika, json, uuid"
- The requesting program must connect to the same queue as the microservice ('StringValidation')
- The requesting program should contain a class that acts as the Client or Connector to the microservice, which should have a 'call' function that sends the proper data to the microservice
- This "proper data" should be:
    - A list (we'll call it *string_list*) containing [the user's string, an int representing the lower bound of string acceptability, an int representing the upper bound of string acceptability]. An example would look like ["Do homework today", 0, 40]. If user's string is <= 0 or >= 40, it will be deemed invalid.
    - Converted into JSON via json.dumps(*string_list*).
- An example call would thus look something like this:

```
stringvalidator_rpc = StringValidatorClient()
test_body = ["Do homework today", 0, 40]
send_body = json.dumps(test_body)
response = stringvalidator_rpc.call(send_body)
```

B. How to RECEIVE data from the microservice

The StringValidator microservice will receive this JSONified list, convert it, and then run a counter for every character in the user's string. If the counter is less than or equal to the lower bound, the microservice will return the string "Too Small". If the counter is greater than or equal to the upper bound, it will return "Too Big". Otherwise, perhaps predictably, the microservice will return "Just Right".

The calling program will need to use json.loads on the response to properly decode it, but then it's possible to create a testing loop where, if user's string comes back as anything other than "Just Right", user will be prompted to input a new string that meets the given character limits.

An example, following the sample call above, would look like:

```
json_resp = json.loads(response)
if json_resp == "Just Right" …
else: …
```

C. UML sequence diagram