Functional Features:

Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called Functional Specification

Functional Requirements of a system should include the following things:

twosum.py > ...

```python
def two_sum(nums,target):
    #check that the list is not empty
    if not nums:
        return None
    else:
        for i in range(len(nums)):
            for j in range(i+1,len(nums)):
                if nums[i] + nums[j] == target:
                    return i,j

print(two_sum([1,7,8,2,5],12))

```

Details of operations conducted in every screen

- Data handling logic should be entered into the system
- It should have descriptions of system reports or other outputs
- Complete information about the workflows performed by the system

It should clearly define who will be allowed to create/modify/delete the data in the system

How the system will fulfill applicable regulatory and compliance needs should be captured in the functional feature.

```
string01 = "MyStRing"
string02 = "MyStRing Is RepEatIng"


def normalized(input_string):
    input_string = input_string.replace(" ", "")
    return input_string.lower()

# Checking the occurrence in the alphabet
def alphabet_unique(input_string):
    alphabet = "abcdefghijklmnopqrstuvwxyz"
    for i in input_string:
        if i in alphabet:
            alphabet = alphabet.replace(i, "")
        else:
            return False
    return True


unique_string = normalized(string01)
non_unique_string = normalized(string02)

# Printing the normalized string
print(unique_string)
print(non_unique_string)

# Printing the result - True or False
print(alphabet_unique(unique_string))
print(alphabet_unique(non_unique_string))
```

For the first question, the possible recruiter asks us to “Implement an Algorithm to determine if a string has all unique characters” and suggests that we create at least one solution with no data structures usage.

Here, we’re going to use three different approaches to solve the problem: Dictionaries, a Set function and in the final solution we’ll look through the alphabet to figure out whether a character was already used or not in our string.