**What is a functional component?**

It’s a part of a system which realizes functional criteria. When designing functional components, be sure to not go too deep into details. You don’t want to program the functions, just design them.

**What is the responsibility of the function component?**

From what I can tell, it seems to be responsible for linking frontend to backend. In other words, making it so that when a user does something, the backend changes accordingly, which then should cause the frontend to give feedback to the user about the change/update.

**Which considerations are needed for the implementation of:**

* **Update functions**
  + They should only ever be run when the model is in a state where the given update function makes sense to be run. Take an ATM machine for example, if you are at the state where the user has to enter their pin code and you get a call to the update function for money being withdrawn, then something funky is going on that shouldn’t be going on.   
      
    As a side note, you need to think about how the function should react in cases where it affects multiple objects, but some of them refuse the function while others accept it.
* **Read functions**
  + You need to take into consideration how other functions, especially update functions, can be impacted by decisions for this type of function. For example, storing some attribute in all objects of the same class, each with a different value, will be easy to read as you just have to access that one object’s value to get it, while updating it becomes hell because you then need to find specific objects in order to change their value. On the other hand, if all changes were just stored in some sort of event log, updates would be easy because it would just have to save that some interaction occurred between object x and object y. Reading data from that would be utter hell though, as you would have to search through all the changes for specific objects and compute them in order to read them.  
      
    Basically what I’m saying is, you can either have a sweet or a sour cake. Trying to mix them would be redundant, and having only either of them will not leave everyone happy.  
      
    You should also consider if a read function needs input to change the way it reads data. Just a heads-up, famalam.
* **Computing functions**
  + Consider input, objects, their attributes, their relations to other objects, and the types of outputs they should be giving.
* **Signal functions**
  + Consider the rules for when a signal function should be called, as well as whether or not it should get input to affect the sort of output signal it gives. Also, consider the objects it affects, and the relations those objects have to other objects. You also need to consider the effects the signal output can have on the state of your system.