**Peer-graded Assignment: Capstone Assignment 2.3 – Identify and Fix Code Smells**

**Part 1**

**On sharingapp:**

1. On Class Dimensions:
   1. Data class anti-pattern: contain only data and no real functionality. Some more functions could be delegated to this class.
2. On Class ContactController:
   1. Data class anti-pattern: contain only data and no real functionality. Some more functions could be delegated to this class.
   2. Speculative generality code smell: -> addObserver and removeObserver: code that is not needed at the time.
3. On Class ContactListController:
   1. Data class anti-pattern: contain only data and no real functionality. Some more functions could be delegated to this class.
   2. Speculative generality code smell: -> getSize and getContactByUsername: code that is not needed at the time.
4. On Class EditItemActivity:
   1. Long method anti-pattern: on method saveItem. Should implement a method to validate the input.
5. On Class ItemAdapter:
   1. Comments code smell: on method getView -> getItem(): There is used he same method name than one used by a standard Java functions, change the name of the method to one different.

**Part 2**

In this problem fixing a Long Method anti-pattern, by Implement a validateInput() method in: AddItemActivity, EditItemActivity, AddContactActivity, EditContactActivity. We are also introducing a new anti-patterns or code smells the “Duplicated code” by breaking the D.R.Y. principle.

As a solution, could be the creation of a class for control input, like ItemInputValitate and ContactInputValitate whit the necessary functions inside.