

## **Apsynergy Exercise : Documentation Technique**

When it comes to the architecture of my project, After Analysis and conception phase, I come up with the following classes. The all architecture is relying on Uncle Bob principles like:

- Do not repeat yourself.
- Testability and Maintainability
- Avoid Large Functions
- Meaningful Names

### **The schema of the project**

#### **1.Trigger:**

**Trigger:** trigger OppoClose on Opportunity (After Update).

**OpportunityTrigger\_Handler:** A class that contains **OpporunityTriggerLogic()** method which responsible of the logic needed.

#### **2.AssetCreation Helper Class (Main Logic):**

**AssetCreationHelper:** A class that helps to create new asset from the concerned opportunity.

**GetOpportunitiesClosedWon():** A method from AssetCreationHelper serves to get the opportunity having the statut Closed Won.

**SetAssetFromOppo():**A method from AssetCreationHelper which serves to assign all the required information to the new created Asset.

#### **3.PageHelper Class:**

**PageHelper:** A class which is responsible to generate the External\_\_Id of the new Asset.

**TestCounterLastIndex():** A method of the class PageHelper. It is designed in order to find the number of occurrences of an already existed External\_\_Id.

**GenerateExternalId():** A method that uses the previous TestCounterLastIndex() to ultimately generate the new Asset External\_\_Id.

## **What can I do more?**

- I can work on the loose coupling by using dependency injection.
- I can implement Factory design pattern to generate my future object instances according to my business logic.

**NB : Je vous remercie infiniment pour votre temps. Nabil BOUHASSOUN.**