**Create the following architecture:**

* Category
  + category\_name
  + category\_description
    - def \_\_str\_\_(self)
* Product
  + product\_category
  + product\_name
  + product\_unit
  + product\_price
  + product\_cost
  + product\_date
  + product\_description
    - def \_\_str\_\_(self)
* Person
  + name
  + surname
  + address
  + city
  + phone
* Courier(Person)
  + Id
    - def \_\_str\_\_(self)
* Supplier(Person)
  + Id
  + company
    - def \_\_str\_\_(self)
* ProductManagement
  + cash
  + login
  + Password
  + search
    - Get input 1-product ; 2-category; 3- supplier; 4- courier
    - Get search keyword
    - Search the keyword in selected array
  + add\_remove\_supplier
    - Get input 1- add supplier 2- remove supplier
    - Add new supplier or remove existing one
  + add\_remove\_courier
    - Get input 1- add courier 2- remove courier
    - Add new courier or remove existing one
  + del\_product
    - Receive product name
    - Check if exists
    - Delete product from array
  + show\_array
    - Get input 1-product ; 2-category; 3- supplier; 4- courier
    - Must show the elements in selected array
  + check\_cash
    - print current cash
  + add\_product
    - Add new product instance to existing productArray
    - Decrease cash (cash -= quantity \* cost)
  + add\_category
    - Add new category instance to existing categoryArray
  + sell\_product
    - Check if the input product name exists
    - Input quantity & check if there is that much left in stock
    - Increase cash (cash += quantity \* price)
    - Remove from product quantity the sold units
* ManagementInterface
  + authorization
    - Get login and password and check if they are correct
    - If they are true:
      * Suggest to choose from existing actions (example: add product -1 , add category -2 and etc.)
      * Perform the action by calling the correct function

Final: Run the program and check if all the functionalities work correctly