

# **APPLICATION LOGIC SECTOR**

MAIN PROJECT  
REV 03 - 11 DIC 2020

*Victor G. De Matteis*

# UML FOR GUI PURPOSES

ColataElementManager
<b>For creation:</b> + CreateInDB (PrototypeColata) : Colata
<b>For order change:</b> + moveElementUp (index) + moveElementDown (index)
<b>For reading:</b> + get (index) : Colata + getAll ( ) : ArrayList
<b>For removal:</b> + deleteFromDB (index)
<b>For update:</b> + updateInDB(indexOfElement, newElement) + updateInDB(oldElement, newElement)
<b>For re-sync with DB (should not be necessary)</b> + fetchFromDB ( )

AcciaioElementManager
<b>For creation:</b> + CreateInDB (PrototypeAcciaio) : Acciaio
<b>For reading:</b> + get (index) : Acciaio + getAll ( ) : ArrayList
<b>For removal:</b> + deleteFromDB (index)
<b>For update:</b> + updateInDB(indexOfElement, newElement) + updateInDB(oldElement, newElement)
<b>For re-sync with DB (should not be necessary)</b> + fetchFromDB ( )

**ACCIAIO IS NOT REMOVABLE IF USED IN AT LEAST ONE COLATA!**

Deleted acciaio already used will just not be usable for new colata, but it will still be readable.

All this is handled automatically.

ProductElementManager
<b>For creation:</b> + CreateInDB (PrototypeElement) : Product
<b>For reading:</b> + get (index) : Prodotto + getAll ( ) : ArrayList + getByColata (colata: Colata) : Prodotto
<b>For removal:</b> + deleteFromDB (index)
<b>For update:</b> + updateInDB(indexOfElement, newElement) + updateInDB(oldElement, newElement)
<b>For re-sync with DB (should not be necessary)</b> + fetchFromDB ( )

Abstract Element <T>
Attributes: + code + deleted + checkDeleted ( ) : function called in every Getter.

Abstract Prototype
No functions or attributes.

ElementColata extends Element <ElementColata>
+ positionInList + status <b>No products array!</b>  + usableWeight + endTime + startTime + acciaio  Getter for all fields.

PrototypeColata extends Prototype
+ usableWeight + acciaio  No functions.

ElementAcciao extends Element <ElementAcciao>
+ name + description + notUsable (deleted but still readable, get functions will work)  Getter for all fields.

PrototypeAcciao extends Prototype
+ name + description  No functions.

ElementProdotto extends Element <ElementProdotto>
+ startTime + endTime + length + weight + sectionArea + colata  Getter for all fields.

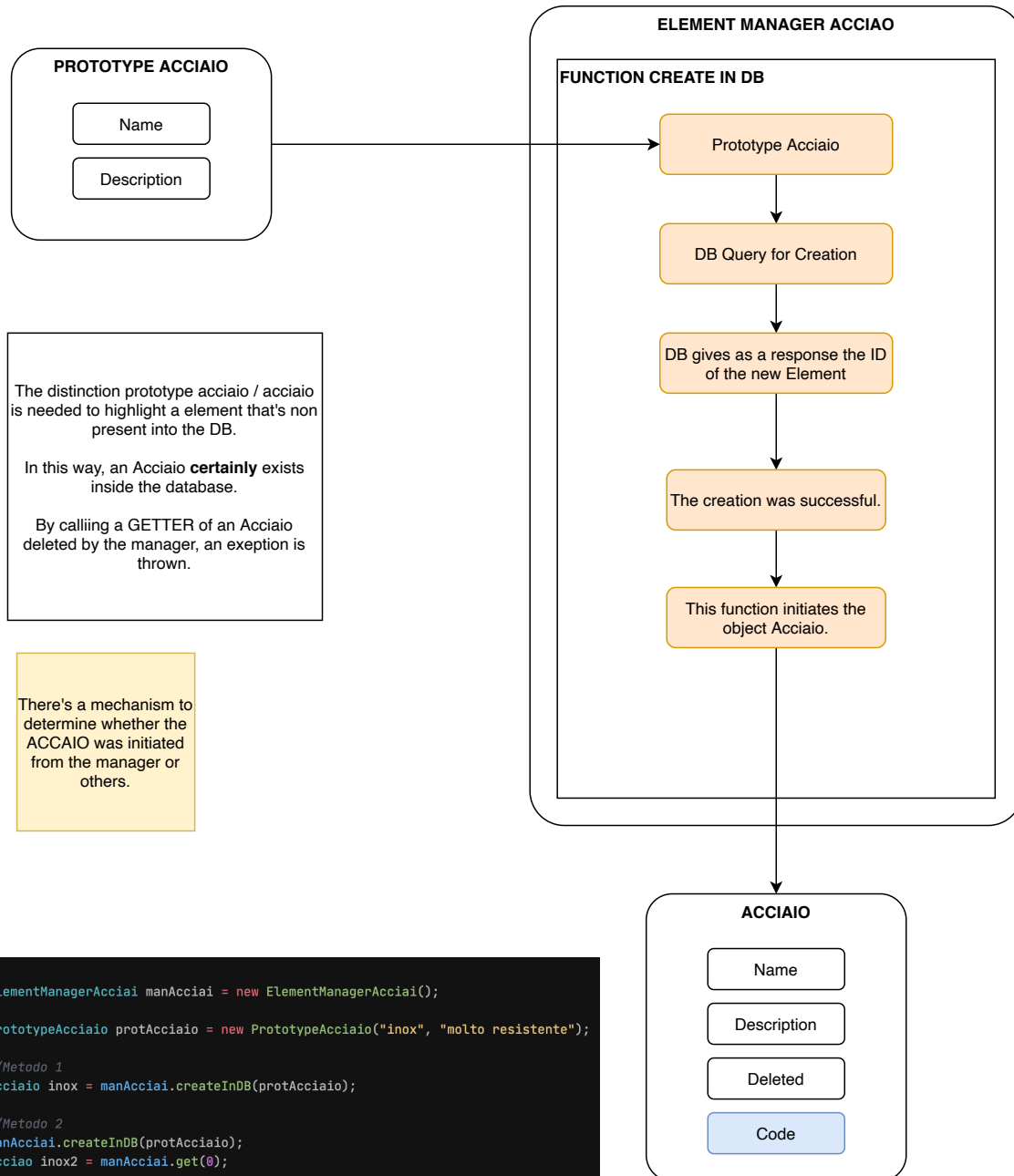
PrototypeProdotto extends Prototype
+ length + weight + sectionArea + colata  No functions.

Exeption PrototypeSystemAvoidance
The element has not been initialized using the prototype.

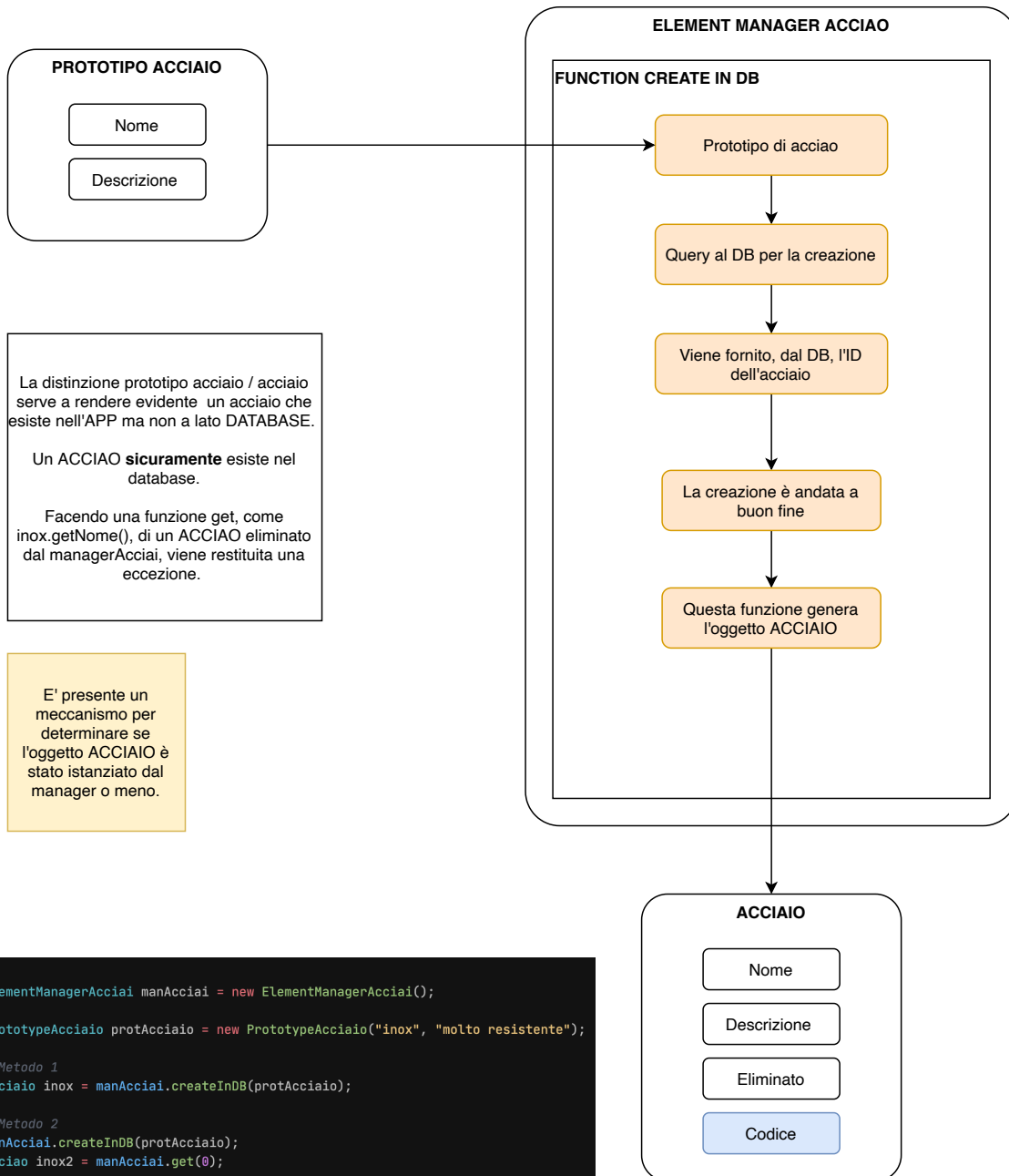
Exeption DeletedFromDB
The element has been deleted from the DB. The instance of that element is saved somewhere, outside the manager, where, after being deleted, is still usable.

<b>Other classes for GET and POST requests will have static methods, that will be called by managers only.</b>
--

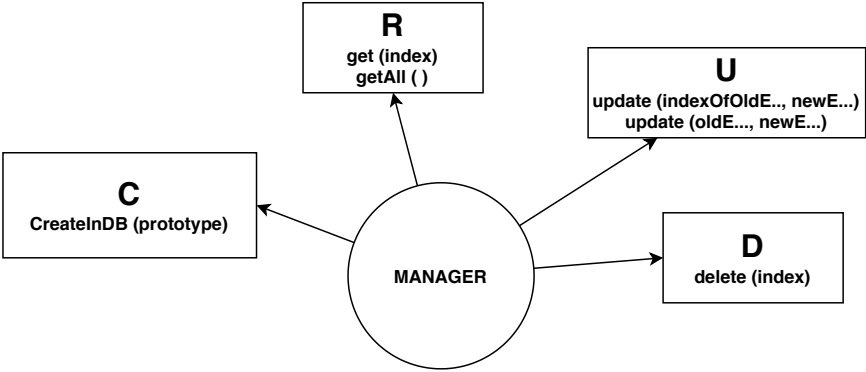
# CREATION



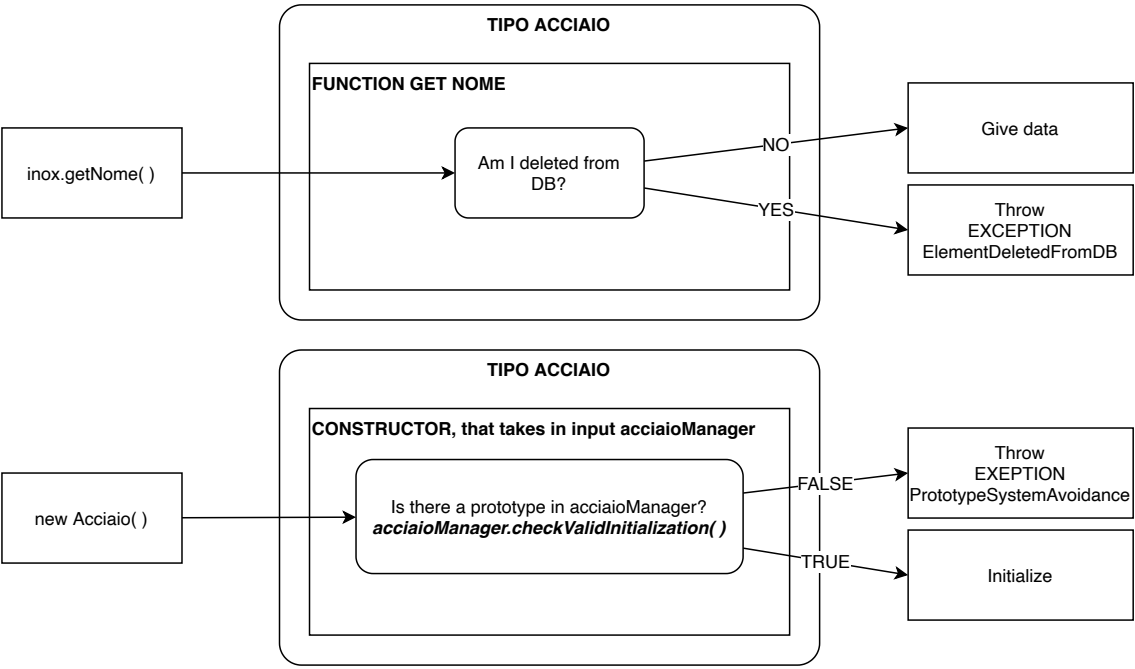
# CREATION (ita)



# MANAGER



# COHERENCE KEEPER



# INTEGRAL UML

ABSTRACT Element Manager <T>
- elements: ArrayList <T>, - prototypes: ArrayList <T>
<b>Public</b>  + get (index) : T + getAll ( ) : ArrayList + deleteFromDB (index) + updateInDB(indexOfElement, newElement) + updateInDB(oldElement, newElement) + fetchFromDB ( )  <b>Package Protected</b>  - CreateInDB (element : T) : T <b>abstract</b> - ElaborateJSONToObject ( ) <b>abstract</b> + checkValidInitialization ( ) <b>abstract</b> + checkDeleted ( )  <b>Constructor</b> Executes fetching

Element ManagerColata extends ElementManager <Colata>
<b>Public</b>  + CreateInDB (PrototypeColata) : Colata + moveElementUp (index) + moveElementDown (index)  <b>Package Private</b>  <b>Constructor</b> Execute fetching + elementManager Acciaio + elementManager Prodotto

## Element not initiated from prototype prevention mechanism

In the moment when the function **add()** is called, the prototype is added into an arrayList.

When the element is initiated, it asks the ElementManager if it's valid by calling the function **checkValidInitialization**.

This functions searches if there are any prototypes with the same data as the element.

If the initialization is not valid, an Exeption is thrown.

Element ManagerAcciaio extends ElementManager <Acciaio>
<b>Public</b>  + CreateInDB (PrototypeAcciaio) : Acciaio <b>Override</b> deleteFromDB (index) If "usedOnce" == true, do not delete. Just set "notSelectable" = true.  <b>Package Private</b>  <b>Constructor</b> Execute fetching

Element ManagerProdotto extends ElementManager <Prodotto>
<b>Public</b>  + CreateInDB (PrototypeProdotto) : Prodotto + getByColata ( colata )  <b>Package Private</b>  <b>Constructor</b> Execute fetching

**ACCIAIO IS NOT  
REMOVABLE IF  
USED IN AT LEAST  
ONE COLATA!**

Deleted acciaio  
already used will just  
not be usable for new  
colate, but it will still  
be readable.

Abstract Element <T>
<p>Attributes:</p> <ul style="list-style-type: none"> <li>+ Code</li> <li>+ deleted</li> <li>+ checkDeleted ( ) : operazione fatta ad ogni GET ( ).</li> </ul> <p>Particular functions:</p> <p><b>constructor</b> from JSON (Prototype &lt;T&gt;)</p> <p><b>constructor</b> ElementManager ( )</p> <p>generateJSON ( )</p>

Abstract Prototype

ElementColata extends Element <ElementColata>
<ul style="list-style-type: none"> <li>+ positionInList</li> <li>+ status ["DA_PRODURRE"; "PRODOTTA"]</li> </ul> <p><b>No products array!</b></p> <ul style="list-style-type: none"> <li>+ usableWeight</li> <li>+ endTime</li> <li>+ startTime</li> <li>+ acciaio</li> </ul>

PrototypeColata extends Prototype
<ul style="list-style-type: none"> <li>+ usableWeight</li> <li>+ acciaio (check that acciaio is usable, if false, set usedOnce == true)</li> </ul>

ElementAcciao extends Element <ElementAcciao>
<ul style="list-style-type: none"> <li>+ name</li> <li>+ description</li> <li>+ notUsable (deleted but still readable, get functions will work)</li> <li>+ usedOnce</li> </ul>

PrototypeAcciao extends Prototype
<ul style="list-style-type: none"> <li>+ name</li> <li>+ description</li> </ul>

ElementProdotto extends Element <ElementProdotto>
<ul style="list-style-type: none"> <li>+ startTime</li> <li>+ endTime</li> <li>+ length</li> <li>+ weight</li> <li>+ sectionArea</li> <li>+ colata</li> </ul>

PrototypeProdotto extends Prototype
<ul style="list-style-type: none"> <li>+ length</li> <li>+ weight</li> <li>+ sectionArea</li> <li>+ colata</li> </ul>

Exeption PrototypeSystemAvoidance
<p>The element has not been initialized using the prototype.</p>

Exeption DeletedFromDB
<p>The element has been deleted from the DB.</p> <p>The instance of that element is saved somewhere, outside the manager, where, after being deleted, is still usable.</p>

<p><b>Other classes for GET and POST requests will have static methods, that will be called by managers only.</b></p>
---