

## Assignment # 4 hand analysis

Users:

1. Administrator
    - a. Adding, removing objects to the list
  2. Client
    - a. Adding, removing objects to the list
- 

System: listing using an array

Use case: Add new element to the end of the list

Actor: Administrator

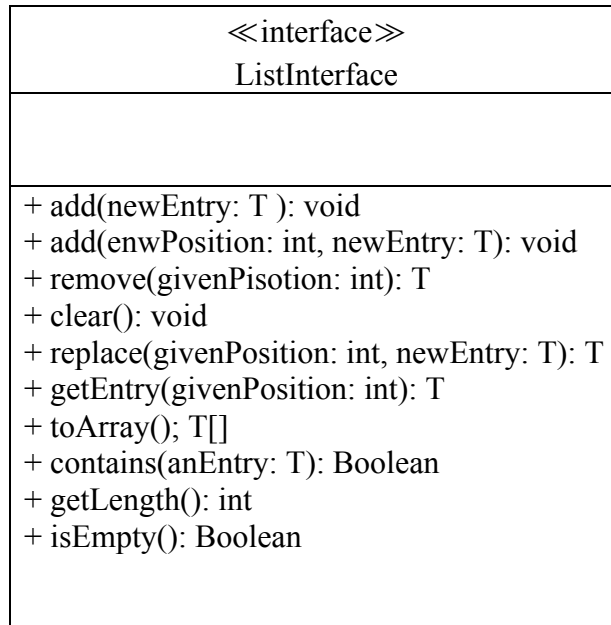
Steps:

1. Administrator enters identifying data
  2. System confirms eligibility to enter
  3. Administrator choses a new element to be added to the without a position number
  4. System checks for empty space at the end of the list
    - a. If there is space, it will add the element to the end of the list
    - b. If no space is left, it will throw an exception
  5. System confirms the addition
  6. System displays the added element
- 

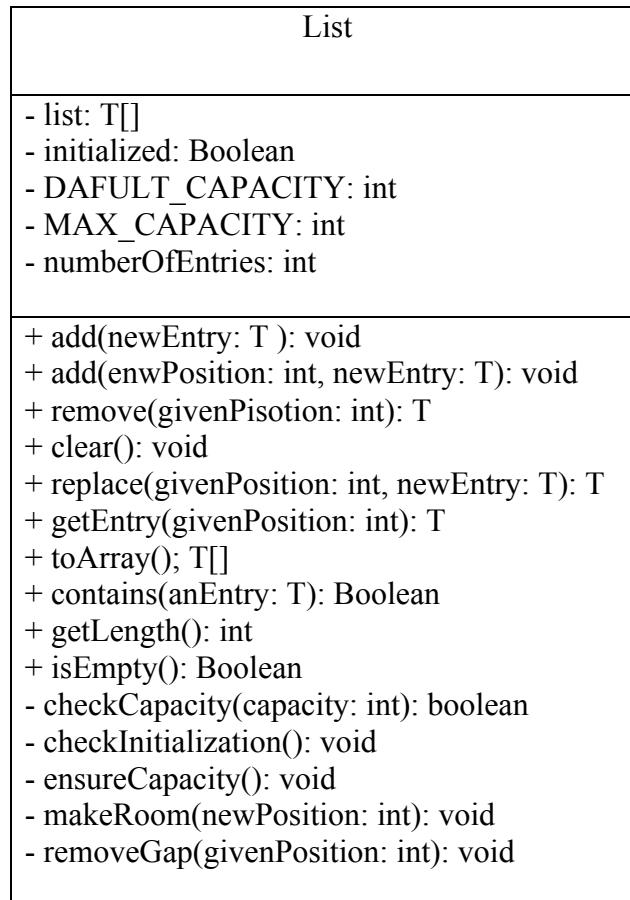
### CRC card for List array class

List using array
<b>Responsibilities</b>
<ul style="list-style-type: none"><li>• Add an element to the end of the list</li></ul>
<ul style="list-style-type: none"><li>• Add an element depending on the desired position</li></ul>
<ul style="list-style-type: none"><li>• Remove an element according the desired position</li></ul>
<ul style="list-style-type: none"><li>• Retrieve a desired entry of a position</li></ul>
<ul style="list-style-type: none"><li>• Empty the list</li></ul>
<ul style="list-style-type: none"><li>• If the list array is empty</li></ul>
<ul style="list-style-type: none"><li>• Returning the size of the list</li></ul>
<ul style="list-style-type: none"><li>• Replace the entry of a given position with a new entry</li></ul>
<ul style="list-style-type: none"><li>• Verifies whether the list contains an element</li></ul>
<b>Collaborations</b>
<ul style="list-style-type: none"><li>• Administrator</li></ul>
<ul style="list-style-type: none"><li>• Element</li></ul>

### UML diagram for the interface ListInterface



### UML diagram for List class



Pseudo code for adding an element to the list with a given position:

1. Checks to make sure that a list has already been created and initialized to true.
2. Sets a Boolean value of results to false.
3. Makes a call to the vectorList add method with the newPosition and the newEntry.
4. Verifies the capacity for making sure there is enough space for the next entry to be added by calling method ensureCapacity.
5. Sets the result to true.
6. Returns the result.