Classroom.cpp Documentation

- **Method Name:**
  - AddStudent()
- **Return**:
  - Bool
- **Pre-Condition**:
  - There are > 0 students, UID of new student != UID of current students.
- **Post-Condition**:
  - Returns true if new student was added; return false if student already present.

- **Method Name**:
  - RemoveStudent()
- **Return**:
  - Bool
- **Pre-Condition**:
  - There are > 0 students in gradebook, UID of student to be removed == UID of existing student.
- **Post-Condition**:
  - A value of true is returned if the student identified by their UID is removed; otherwise return false.

- **Method Name**:
  - AddProject()
- **Return**:
  - None
- **Pre-Condition**:
  - Current projects > 0
- **Post-Condition**:
  - New project is added to student's project list

- **Method Name**:
  - AddExams()
- **Return**:
  - None
- **Pre-Condition**:
  - Exams in list > 0, added exam grade > 0.
- **Post-Condition**:
  - Exam grade is added to student's exam grade list

- **Method Name**:
  - SetFinalExams()
- **Return**:
  - None
- **Pre-Condition**:
  - There are > 0 exam grades in list, final exam > 0.
- **Post-Condition**:
  - Sets final exam grade of student.

List_impl.hpp Documentation

- **Method Name**:
  - List()
- **Return**:
  - None
- **Pre-Condition**:
  - None
- **Post-Condition**:
  - Nodes are initialized correctly.

- **Method Name**:
  - ~List()
- **Return**:
  - None
- **Pre-Condition**:
  - None
- **Post-Condition**:
  - Data from list is deallocated.

- **Method Name**:
  - AppendItem()
- **Return**:
  - None
- **Pre-Condition**:
  - Counter != Max_Count
- **Post-Condition**:
  - Item added to list, counter is incremented

- **Method Name**:
  - DeleteItem ()
- **Return**:
  - bool
- **Pre-Condition**:
  - Item to be deleted == item in list
  - Counter > 0
- **Post-Condition**:
  - Value of true returned if item removed
    - Counter decremented
  - Value of false if not removed
    - Counter unchanged

- **Method Name**:
  - Count ()
- **Return**:
  - int
- **Pre-Condition**:
  - None
- **Post-Condition**:
  - Number of elements in list is returned

- **Method Name**:
  - Front ()
- **Return**:
  - Reference pointer
- **Pre-Condition**:
  - Number of items in list > 0
- **Post-Condition**:
  - Memory location of first item returned.

- **Method Name**:
  - Front ()
- **Return**:
  - Type
- **Pre-Condition**:
  - Number of items in list > 0
- **Post-Condition**:
  - Returns a copy of the first item in the list

- **Method Name**:
  - Back ()
- **Return**:
  - Memory location
- **Pre-Condition**:
  - Number of items in list > 0
- **Post-Condition**:
  - Memory of last item in list is returned.

- **Method Name**:
  - Back ()
- **Return**:
  - Type
- **Pre-Condition**:
  - Number of items in list > 0
- **Post-Condition**:
  - A copy of last item in list is returned.

- **Method Name**:
  - IterateItems ()
- **Return**:
  - pointer
- **Pre-Condition**:
  - Number of items in list > 0
  - _iterator pointer not at last list item
- **Post-Condition**:
  - Returns value at _iterator pointer
  - _iterator pointer now pointing at next list item

- **Method Name**:
  - AtEnd ()
- **Return**:
  - bool
- **Pre-Condition**:
  - Number of items in list > 0
- **Post-Condition**:
  - Value of true returned if iterator pointer is at list end
  - Value of false returned if not at end of list

- **Method Name**:
  - List()
- **Return**:
  -
- **Pre-Condition**:
  -
- **Post-Condition**: