55/100

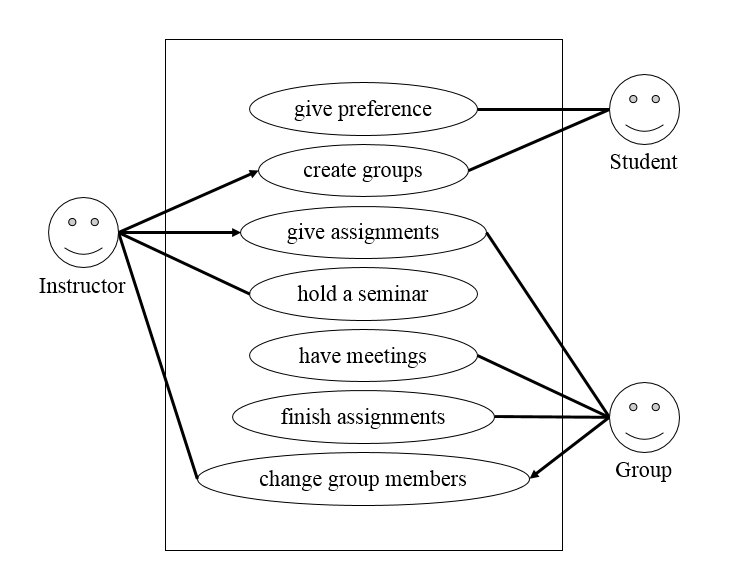
Identify and prioritize features

1. The instructor will specify how many students participate in a group.
2. Students can indicate a preference (both positive and negative) for their partners.
3. The instructor will have the power to force some students to work together or to prevent particular students from collaborating.
4. Groups have time to meet.
5. Groups have approximate ability.
6. Groups have meetings.
7. The instructor can give assignments to groups.
8. Groups submit their assignments to instructor.
9. The instructor holds a seminar for groups.
10. Groups can change their group members.

Be careful features 4 to 10 were not specified by the client and should not be part of the system.

What are the fundamental features you need to implement the basic functionality of the system?

Use case diagram



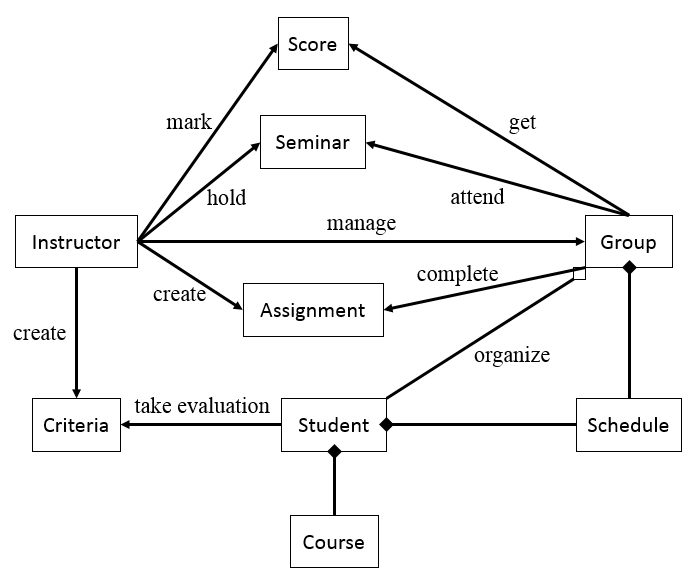
Same here. Several of the use cases are not required and were not defined by the client as a must of the system. Where do they come from?

Traceability matrix

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 |
| U1 |  | ✓ |  |  |  |  |  |  |  |  |
| U2 | ✓ | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |
| U3 |  |  |  |  |  |  | ✓ |  |  |  |
| U4 |  |  |  |  |  |  |  |  | ✓ |  |
| U5 |  |  |  | ✓ |  | ✓ |  |  |  |  |
| U6 |  |  |  |  |  |  | ✓ | ✓ |  |  |
| U7 |  |  |  |  |  |  |  |  |  | ✓ |

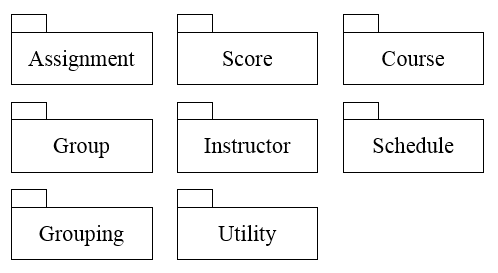
The use case number depends on the order of Use case diagram, from top to bottom.

Domain model



The basic attributes of the components need to be considered in the diagram. Again, it includes extra information that was not part of the original project. Did you discuss this with the client?

Modules



Assignment: Contain assignment information, such as content and seminar.

Score: Handle the feedback of instructor.

Course: Contain course information, such as what the requirement is to take this course.

Group: Contain students and group meetings.

Instructor: Just instructor, it is different from group.

Schedule: Just schedule.

Grouping: Handle how to organize groups.

Utility: Classes that provide generic behavior used by the other packages go here.

How did you control the risk of implementing something different to what the client was expecting of the system? What architecture did you build to implement your system? What facts guided your decisions on the UML modification and inclusion of extra features?