Niall Delaney Anton Stefansson

20113113 201452901

**Group Creator**

**Prioritized Feature List**

1. The system stores an editable private schedule, editable preference lists, and possibly a self-evaluation for each student.
2. The system stores all the criteria provided by the instructor.
3. The system creates and stores a list of groups based on criteria specified by the instructor.
4. The system is able to access the university systems for student grades and class schedules.
5. The system stores groups of students that must or must not work together.
6. The system can send the final list of groups to the students.
7. The system does not allow student access after a deadline.

Why the deadline is less important than the grouping preferences if the grouping preferences are part of an alternate path?

**Key Features**

1. Stored private schedule, preference lists, and self-evaluation for each student – essence of the system
2. Stored criteria provided by the instructor – essence of the system
3. Creating and storing a list of groups based on criteria provided by the instructor – essence of the system, and how is it done?

The key features should be part of the previously defined features. You defined a new list.

Are the student preferences part of the key path? Can you implement the basic grouping functionality without considering any other criteria excluding the group size and a list of students name?

**Justifications:**

* The stored information about each student is essential to the system. The system needs students from which to create the groups.
* Stored criteria from the instructor is essential to the system. The groups are created based on criteria the instructor decides, so this criteria is essential to the main function of the system.
* Creating and storing groups is essential to the system, because this is the main function of the system. This also gets the question “how is this done?” because the instructor must be consulted on the way the group creation algorithm will work.
* Having these key features helps reduce risk for building the system. They are the most important functionality where there could be problems. All features of lower priority are not part of the core functions, so it would not reduce risk if those were worked on first.
* The rest of the features are not key features. They add functionality to the system, but the system can perform basic functions without them.
* Fetching grades and class schedules, as well as storing lists of students that must or must not work together is a major part of group creation. But the group creation can still function without it.
* The system being able to send the list of groups to the students is not important to the core functionality of the system. The system can still use information to create the groups without this feature being implemented.
* Students not being able to access the system after a deadline does not affect the functionality of the system, so it is low in the prioritization.

If the student access the system after the deadline and the professor already created the groups, How is the student input going to be handled?