We have effectively merged the architecture each of our former two groups had produced into the following feature-based plan for our prototype, to be delivered on December 3rd.

Features our prototype will implement:

* Accept list of students from a course
* Accept group size from professor
* Accept mandatory and forbidden groupings from professor
* Accept criteria for self-evaluation from professor
* Accept self-evaluation from students
* Make groups according to professor's rules
* Balance skills and GPA among groups
* Display final list of groups to professor
* Allow professor to edit final groups
* Allow professor to view all students' information

This will constitute the essential functionality of the system and will put in place a framework into which more sophisticated considerations can be added. We defer to a later development iteration:

* Student preferences
* Availability considerations (class and private schedules)
* Consultation of grades for specific courses
* Deadline/automatic group generation
* Automated notifications to students
* Rank/weighting of parameters

We have prioritized the features in this way because we believe our prototype will offer the features that are the most immediately useful to and desired by the client:

* Ability to enforce or prohibit groups
* Gather input from students (self-evaluations)
* Create groups with balanced academic strength

Now that we have identified the essential, we will reduce risk by addressing the most difficult to implement: the algorithm that will generate the groups.

Questions for client:

- Do you consider it very important to be able to specify the relative importance of the grouping parameters?

- How is GPA to be balanced among groups? (minimum average? Equalized?)

- What controls over the parameters are essential to you?