Each assignment is graded over a series of categories. You will be judged on a scale of 1-4 for each criterion, where a 1 corresponds to a 60%, a 2 corresponds to 75%, a 3 corresponds to 90%, and a 4 corresponds to 100%. If there is no work for a criterion or it is clear that even a minimal amount of effort was not put in, you will receive a 0% for that section of the assignment.

The following is a tentative grading rubric for Assignment 1. This may change before final grading, but gives criteria to aim for with your submission.

Organization (25%):

4	Have a good organization including a logical layout, requirements grouped by similarity, all sections present, requirements formatted to be easily understood, uses good grammar, and has a single voice. No irrelevant data (i.e., made up "satisfaction numbers").
3	Most sections present, layout mostly logical, and requirements are easily understood. Lacks single voice and has some grammar issues.
2	Missing some sections, illogical layout, and requirements are hard to understand. Lacks a single voice, many grammar issues
1	Missing major sections, layout illogical, and requirements are not readable. Hard to read.

Use Cases (35%):

4	Captures core usage scenarios of GRADS system. Present and well formatted diagram. Descriptions are clear. System boundary and actors are clear and correct both in diagram and document.
3	Some mistakes in UC diagram or descriptions. Missing system boundary descriptions or actors incorrect.
2	UC is unclear and incorrect in several areas.
1	UC mostly incorrect - for example, specified a GUI without underlying data processing system.

Requirements (40%):

4	All major system functionality captured. Accounts for error cases. Requirements sufficiently complete and detailed enough to implement. Requirements are not contradictory.
3	Most system functionality captured, or error cases are not accounted for, or behavior related to some of the major graduation rules missing. Lacking in detail.
2	Missing some major functionality including, but not limited to, missing error cases, graduation rule checks, and basic form processing. Requirements barely detailed, are ambiguous, or are contradictory.
1	Missing most functionality. Generally unable to determine what system is supposed to do. Lack of detail sufficient to be unable to implement software.