Goals and motivation for the assignment: The purpose of this assignment is to give the student firsthand experience with the challenges entailed in creating a requirements specification, and being sure that the requirements specified are suitable as the basis for constructing a testing and evaluation plan.

What is the assignment?:
In assignment #1 you informally described a complex software system of interest to you. In assignment #2 you have demonstrated that you understand the power and difficulties of being precise in specifying software systems and their features. In this assignment you will go further and use some of the precise software product notations with which you are familiar as the basis for defining a precise specification of some aspects of the requirements for the complex system you have previous described informally. You will also use notations and formalisms that seem to you to be appropriate in order to specify elements of a test plan aimed at supporting demonstration that a running software system satisfies some representative parts of your requirement specification.

SPECIFICALLY:
1. Develop a specification of the requirements for three (or more) representative aspects of the system you described in assignment #1. One convenient way to do this might be by having each of these aspects of the system specified through the use of Use Cases patterned after the Use Cases in the Tokeneer requirements specification. This will require augmenting the Tokeneer-style Use Cases with graph representations.

For 520 Students: Your requirements specifications should address at least four of the following types of requirements. If you represent them through Use Cases you should develop at least two different Use Cases.
   Functional
   Environmental
   Robustness
   Accuracy
   Timing
   Security
   Safety
For 620 Students: Your requirements specification should address all of the following types of requirements. If you use Tokeneer-style Use Cases to do this, then you should develop at least two different Use Cases.

- Introduction
- Functional
- Environmental
- Robustness
- Accuracy
- Timing
- Security
- Safety

But, in case you feel that any of the above types of requirements are not applicable or relevant to your system, be sure to explain why.

For each type of requirement you should select a formalism or notation that seems to you to be appropriate for the clear and precise exposition of the requirements. Some examples of how to use DFGs, CFGs, Predicate logic, FSMs and other other formalisms, have been provided in the lecture material.

2. Develop a test plan that corresponds to the requirements you have specified. For this part of the assignment assume that there now exists an implementation of the system that is purported to meet the requirements specified as described above. For this part of the assignment, specify how you would go about testing that system to determine its adherence to the specified requirements.

As noted in the lecture material, a test plan should have an overall structure, and a decomposition down to the level of specific sets of test cases. So your test plan should indicate the structure of your proposed test cases in a way that makes the structure clear. It may be both expedient and effective for this structure to build upon the structure you have selected for your requirements specification. Your test plan should consists of sets of test cases grouped together, and labeled to identify the requirements element(s) that are to be evaluated by the group. This identification is essentially a relation, and so you must indicate the relation that connects the requirements element(s) and the test case group. Finally, each test case group will consist of individual test cases. It is not necessary that you include all test cases that you deem necessary, but you should provide some example test cases, and indicate informally the nature of the other test cases that you think would be needed to make a complete test case group. Each test case should specify inputs to be provided, outputs expected, timings, resource requirements, responses to failures, setup procedures, and any other information that you think is necessary to support definitive testing.

Anticipated length:
For CS 520 Students: 8-12 pages
For CS 620 Students: 15-25 pages
BUT NOTE: If you are confident that you can provide good answers to the questions in fewer pages, then do so. Under no circumstances should you put in extra time and effort writing more if you don’t feel that doing so is helping you to learn and understand more.