CSCE 247 - Project 3 - Draft Design

Due Date: Sunday, April 7, 11:59 PM (As PDF document(s), via Moodle)

Overview

In this deliverable, you are required to produce a detailed design of the GRADS system. The design must be an object-oriented (OO) design and refined down to detailed descriptions of each operation and attribute, with documentation to explain and justify your design. A design template will be available on the course web page to assist in your task.

The Design Process

The design of systems is still a bit of an art. The goal in the design process is a detailed design document that is complete enough to be a reference from which any competent software engineer can produce code and craft executable test cases.

A first step is typically a preliminary design. This can be considered a draft of the full design document with a lot of the details missing. As you go through the design process, everyone in the group should be involved and read the draft class diagrams and descriptions (see discussion below) and other parts of the document to make sure they understand the overall design and can provide design feedback. The next step is to enter a cycle of refinement until you have prepared the final design document. Most of the work will be done by individuals filling in the final details of their modules, but you must review the document as a whole to make sure all team members are using the same interfaces (the objects communicate in the right way) and to make sure that everything has been covered. As you refine your design, be sure to document your rationale - the hallmark of a good design is one that can be justified.

Details on what we expect for the design are included below.

The Document

The design document shall contain the following (at a minimum):

1. An introduction providing an overview of the system. This includes a system overview and design objectives.
2. A design overview describing the system architecture, description of all interfaces with the environment (users and external systems), and a description of all constraints and assumptions used in the design.
3. A static model (UML class diagrams and class descriptions). This section must contain a complete class model (possibly broken down into subdiagrams to simplify the figures), a high-level English description of the model, and detailed description of each class (class description, attribute descriptions, and method descriptions).

4. A written explanation of and justification for the design you chose, including specific rationale for the decisions made in the design (for example, why your design may be better than another or why you chose to implement a specific design pattern in one of your subsystems).

The document should be clear and well organized, and all diagrams must adhere to UML standards.

Resources

Several resources will be made available:

1. A template will be available from the web page.
2. Lists of graduation rules, data fields, and the top level interface of GRADS will be made available. Your system MUST implement this interface in a class named GRADS.
    a. To clarify - a Java Interface is a template that describes a set of methods that all concrete implementation of the interface will provide. The interface, called GRADSIntf, lists methods that will be implemented by your implementation, a class called GRADS. Your class diagram must show both the interface (GRADSIntf) and the implementation (GRADS)
    b. More information on interfaces is at https://www.tutorialspoint.com/java/java_interfaces.htm
3. Sample data that matches the specified data formats.

You may, of course, ask me questions about your design. However, please **do not** send me a document and ask if it’s good enough. If you want feedback, **ask specific questions**. I cannot critique your design unless you explain and justify your design and the decisions you arrived at in coming up with your design.