

CSCE 740 - Project 2 - Final Requirements and Requirement-based Tests

Due Date: Tuesday, October 10, 11:59 PM (In PDF format, via Moodle)

Overview

This assignment involves two tasks.

1. Revise and complete your requirements document from Assignment 1.
2. Develop a collection of requirements-based tests to
 - a. Improve the requirements in your requirements document
 - b. Form the basis for the testing effort during implementation.

Based on the feedback from the professor and additional topics covered in class, you are to revise, refine, and complete your requirements document. This document will form the basis for your design and implementation efforts in the latter half of the class.

In addition to completing your requirements document, you are required to produce a collection of requirements-based tests for the BILL system. You are required to document the test cases and, when the code becomes available later this semester, you will use these test cases to develop concrete tests that can be executed against the program.

Your Assignment

Refine and complete the requirements document. Note that, as discussed in-class, the development of the requirements-based tests can be a powerful tool to help you improve your requirements. Requirements development and test development should be viewed as an iterative process; refine the requirements and develop additional tests iteratively.

A template for test cases and the test document is available from the web page. As before, the document may be reorganized to suit your purposes, as long as all required information is present. Please include only relevant information when handing in your document.

You are required to turn in the following:

1. Your revised requirements document.
2. Test cases that cover the requirements, including detailed inputs and expected outputs.
 - a. Note 1: It is worth noting that one test case might cover several requirements and that thoroughly testing one requirement typically require several test cases.
 - b. Note 2: Make sure each requirement is tested by at least one test case.
 - c. Note 3: Remember that test cases need to cover both the expected behavior of BILL functions (the “happy path”) and error cases.

- d. Note 4: Inputs should be concrete (“A valid student profile” is not acceptable, you need to actually create sample profiles). Outputs should reflect the concrete input (“The remaining balance is \$1234”). Input and output should be accompanied with explanations to document why that output is expected for that input.
3. A traceability matrix linking test cases to requirements.
4. Peer evaluation form (see the instructions in the peer evaluation form available on the Dropbox course page – you will turn in one form per person for this deliverable).