IST 412 Lab 2 – Build and Test a Java Application – please read these instructions *very* carefully.

Intro: Now that you know how to write build/test scripts using Ant, you will now design and build a system, and part of that design and implementation will include testing using white and black box testing ideas.

The System You Build:

1. Your system must have a graphic user interface (GUI) with the ability to securely login and password-verify multiple users, including a way to securely manage and store those usernames and encrypted passwords. There will be rules for password alphabet and length which the system must enforce.
2. Your system must perform some non-trivial function that involves logged-in users. It doesn't need to be overly complex, but needs to be something that requires testing itself. In addition, you need to build in testing of the login/security/password-management system.
3. You need to write your own code for the GUI and login and password-management components of the system, but may use some external API library packages or classes for the system functionality. But you need to make sure that you can adequately test that part of the system also – the main point of the exercise is testing, not system functionality.

Testing: Testing must be part of the design and implementation – it must not be put off until the end. You must develop your testing rationale while you design, and should be applied to modules at the lowest levels possible – you'll find that maximizing modularity and encapsulation will make testing easier and more effective. You must also be testing as you build your system – again, at the lowest, most modular level possible.

Deliverable:

Design Rationale, Testing Rationale, and Test Results: Submit a design/test notebook where you document your design and testing rationale process and your test results. You should start this notebook when you begin the lab – treat it like a designer's or artist's notebook that tracks your progress. Do not edit this – we want to see your thought process. This means you need to write neatly and clearly from the beginning.

System Implementation: You will submit a zip file of just your .java source files, Ant build and test scripts, a build.properties file, any library files you need to implement your system, and a readme.txt that describes how to configure and run your system using the Ant scripts. There should be no specific paths in your build/test scripts, but put that information in a build.properties file as in Lab 1. You are free to develop on whatever IDE you like, but it must run using your Ant build/test scripts from the command line.

Important: you don't need to turn in your build.properties file. The only things you should change in that file are your system-specific paths. Do not change the names (like binDirectory, mainClass, and so on) used in the build-script. The TAs and I will run your build-script using the build.properties file given to you with only the system-specific paths changed for whatever system we're using.

Grading Rubric (out of 50 points):

Will discuss as we move along.