

## CSE 403 – Winter 2011 Assignment 5 – Beta Release

**Due date: Tuesday February 22, 2011 before 11pm**

**Demo date: Thurs and Fri Feb 24 and 25 in class**

### Overview

We, your executive customer, are excited to see the progress you're making on the product. This assignment is a milestone delivery to us, in exchange for which we'll give you a milestone payment (points towards your GPA, in fact!). While we don't expect a feature full product, we do want to see that all the major pieces of the product are in place and integrated at a very basic level. We also would like to see that your team is using good software engineering practices for your work, assuring us that the project will be completed on time, with good quality, and in a good position to evolve with customer needs.

### Deliverables

1. A beta release of your software. A "beta" release should show basic functionality in place, integrated, and working, for the major components of the system. It should be possible to perform multiple operations that start at the client, reach through to the server, and back to the client.

It is permitted for some functionality of the final product to be missing in the beta release, but the system should already be useful. A significant number of use cases should minimally work. The documentation should reflect which commands or features are working and which are in progress.

Your product need not necessarily be 100% bug-free. All known bugs should be documented in the bug database, and a user testing the system should not encounter a non-trivial number of bugs that are unlisted. Your system should be robust: errors should be gracefully handled as much as possible.

A "release" includes several elements packaged in two distributions.

- Binary (installable) distribution. [10 pts] The elements needed for an administrator to install your product on a server and run it remotely. For your web or android application, this may be a server package (zip file) and a url to invoke or android app to download. You do not need to package the database as part of the server for this assignment.

The release should include clearly identified Release Notes in the package that describe (1) how to install and run the software, (2) what commands are working, and (3) a list of any known issues. Please deploy your beta on a separate http port from the port that you are using for your development efforts. This way, your stable beta will persist, independently of your ongoing work.

- Source distribution. [14 pts] The elements needed by someone who is going to pick up the project at this stage and do further development. A separate package for each host target is often a clean way to organize this. (Again, you do not need to describe the underlying database installation for this assignment.)

The source distribution should include clearly identified Release Notes in the package that describe how to build the product from the original sources. In grading, we will be looking for the ease at which a correct build can be made.

The release notes should state any assumptions, such as a Visual Studio or Eclipse or Android emulator installation.

The release notes should identify the source code repository and how to look at file change logs. A new developer may be looking at the change logs to understand why code evolved a certain way, and we expect good developer documentation to be in place.

The release notes should identify the bug database being used with the project. We expect to see good usage of the bug tracking database, especially for bugs that span development groups, by your team members.

Lastly, your software should be supported by tests, and a new developer will need instructions on how to run at least the following:

- unit tests for one major class/component in your system
- integration/system tests for the functionality represented by one use case.

Don't forget that tests need documentation, just like other code.

2. Updated System Requirements Specification (SRS) and System Design Specification and Plan (SDS) docs, with change tracking on. [4 pts] Focus on the content changes, not the polish, for this deliverable. If you have made any changes to your feature list delivery schedule, add a note discussing the tradeoffs that were involved.
3. A description of how you've applied one design pattern or principle (other than the iterator pattern) that we've discussed in class, in your software design. [2 pts] Identify the pattern/principle and the code (file/lines, source location(s)) to which it applies. In grading, we will be looking for a meaningful (vs. trivial) application.

## **Turnin**

One of your team members should turn in all the deliverable material together so that there is one coordinated input for the team. Use the Turn-In link on the class web page. Put the team name in the filename of all components submitted, i.e., cool\_server.tar, cool\_client.tar, cool\_SRS.doc , cool\_SDS.doc.