Iteration 3

At the end of each iteration, every team needs to have a tagged version of their code up to that point (Iteration1, Iteration2, etc.). The course staff will only grade the tagged version. How to tag depends on your version control, so that is something to discuss with your TA.

Meetings for this iteration should be scheduled from March 2 to March 7 (Wed-Monday).

Deliverables

- You need to convince us that you are following the process that your team agreed on. Remember that in cs428 we grade the process not only the product.
- You should have addressed any issues that were discussed during the previous milestone meeting. This includes making sure that your wiki page has been formatted properly, ensuring that your user stories are well described, having automated tests, and any other issues that were brought up during the meeting.
- The wiki should be updated with your progress for the current iteration:
  1. You should fill in the actual time you’ve spent of the user stories in this iteration. Remember to document any bumping of (partial) user stories from one iteration to another.
  2. You should revise the user stories and estimates for the next iterations.
  3. Considering your new estimates, you should divide the user stories among iterations (which user story is going to be done in which iteration).
  4. You should reevaluate project risks, etc.
- You should have made significant progress with your application, shown by working and well tested user stories. We will ask to to show how each user story is implemented and tested.
- Your application functionality should match the user stories which you marked as completed.
- You should have several new commits in your repository of new code that your team has written – not just code from the web or from the examples/sample folder of your framework. Please select a portion of the code that you find most interesting to present during the meeting. We are not looking for a perfect design but you should be able to convince the TA that you are making progress in your project. Also, we will be checking to see that your code adheres to your team’s coding standards.
- You should tag your code.

Iteration Meetings

- It is expected that teams keep notes about their meetings (especially, but not limited to, the iteration meetings). What did the team accomplish? What are the issues causing trouble for the team? What needs to be done? So for each meeting, someone in the team will be the scribe. The notes can be in the form of minutes or anything else you agree on with your TA. Look to your peers to see which method works for them and for you. The bottom line is to create a transparent project that the TA can follow and more importantly that your teammates can follow to ensure excellent team communication. Leverage the wiki and the tools you have chosen to their fullest extent. These notes need not be excessively long (longer is not always better), but they should contain enough detail so that your team members and the TA know what is going on.
- There should be a different leader and scribe for each iteration meeting. Everyone should be able to and is expected to contribute to the discussion.
- All team members should attend all meetings. Any absences should be arranged ahead of time with the team and with your TA.

Documentation

As part of your final project submission you will be expected to compile a documentation document for your project and the software you developed. We will provide more information about the different types of information that will be expected in the near future. It should not be a surprise, however, that this documentation will include information on the requirements and design of your system, as this is an important aspect of the course. As you watch the lectures, read the required readings, and do the homeworks on requirements and design, reflect on how this material should be applied to your project. For instance, you should expect the requirements and design portion of your project documentation to include use cases, class diagrams, and sequence diagrams as discussed in the lectures.