Individual Homework. Turn in the following:

1. Write a short description of the work you did for the project over the last week. Focus your write up on: (1) clearly describing what you set out to do for the week; (2) clearly describing what you accomplished; and (3) reflecting on what went well for you during the week (in terms of progress), what didn’t, and what you will try to do to improve your productivity or teamwork within your team.

Project Homework. This part should be done with your team. Work on the following as a group and hand in all work items asked for as a single copy on the due date. Your team should be starting your seventh and final “sprint” of the semester. At the end of the week (next Tuesday), your team must:

1. Turn in a completed task sheet from this week and an initially filled out task sheet for next week.
2. Continue to use GitHub to store all of your project documents, source code, etc.
3. Write a short description of the results of your sprint planning meeting. Your write up should include the stories you are working on, and what you will have completed (e.g., to demo) by the end of the sprint.
4. Come to class on December 5th prepared to present your demo to other teams to get their feedback. Be sure (i) you can run your application demo in class, and (ii) you have practiced presenting your demo. There are often two ways to present a demo: the first is feature-based where each feature is described in isolation; and the second is task-based where the demo is organized into the sequences of features needed for users to perform their desired tasks. A common problem with a purely feature-based presentation is that the audience can get lost in how the features go together to perform useful tasks (i.e., how a user would actually use the application to get their work done). A common issue with a purely task-based presentation is that it can be difficult to fully understand the breadth of features that have been implemented. A good demo presentation should: (a) have a good pace (i.e., not rush through the tasks or features); (b) leave the audience with a good sense of what features have and have not been completed; and (c) leave the audience with a good sense of what tasks are supported by the application (i.e., what work a user can accomplish and how they would go about using the application to accomplish the work). In addition, it is important to give the audience a sense of where the application is in terms of usability, e.g., whether the “look and feel” and overall “layout” of the UI is complete or whether more work needs to be done, whether the design of the UI is complete or not, and so on.