Aaron Fine  
Michael Andrade  
Matthew Lister  
Abe Gunther

CS 3450  
3/12/18  
  
the flabbergasters  
Sprint 1 Retrospective

**What went well?**

The group’s goal for this sprint was to have a thin prototype up and running. This was accomplished. Our developers all have their own development environments that are functionally identical to the production server. We have git integrated into your development process and the basic views of our app implemented. A brief review of specific user stories completed during this sprint are as follows:

* As a developer I need a framework to support the menu view
* As a developer I need a framework to support the Admin view
* As a developer I need a framework to support the Cook view
* As a developer I need a framework to support the inventory view
* As a developer I need a framework to support the customer review upload page
* As a developer I need an environment to support developing in a two-tier Client/Server architecture. I.e. will not use proxy server

This sprint has helped the development team gain a better understanding of how the high-level design will be implemented in the finished project.

**What did not go well?**

**Working with a group is never easy, and for a group of inexperienced programmers some unique challenges did crop up. What was immediately apparent was a wide and varied skills gap between our developers. While we share a common training background, there was a significant difference in aptitude and technique between team members. We managed to mitigate this by assigning “programming buddies”: having two developers work side by side to help each other.**

**In general, our communication as a team was very good, but the project takes a long time to understand. Talking ended up being a significant portion of our overall time investment, and will likely remain a significant expense. If we compare this project to writing a group paper, the reason for so many long winded explanations becomes apparent. When someone adds a page or two to a paper the others can read over the changes and digest the impact. This is so very different from when a developer works out a few lines of code and adds them to the git. One, the changes may be in any of a dozen files, and two, the developer may be the only one on the team that understands how the code works. This led to our team spending a lot of time on show and tell type explanations.**

**How should you improve?**

**The most obvious place our team could improve is with commenting code. Especially with Django which self generates some configuration files, leaving a comment that includes who, what, when, where, and why every time someone changes the code would be very useful.**

**Improving our workload balance will be a focus of our upcoming sprint. We will reduce the scope of our sprint to better meet the time requirements. We also have more experience making time estimates which will further improve our planning stage.**