Marcos Pertierra

6.170 Fall 2015

November 24, 2015

GrubSplit MVP Reflections

**Peer Review**

We tried to divide the work equally. However, I had less work and no tests last week and this week, so I coded a more on days when teammates were busier with other classes. I still think we all collaborated very well. Matt did most of the work with integrating with delivery.com’s API and authentication. Jorrie helped with the SubGrub model and created the SubGrub menu and cart views. Amanda worked on the Grub and SubGrub model and routes. I worked on the User model and routes, helped with the Grub and SubGrub models, routes, and views.

**Evaluation**

We made good progress on the project in our MVP. Users can sign-up/login to our web app, and are also authenticated through delivery.com. Although users can not currently search for nearby restaurants (we hardcoded Café 472 to be the only search result), we have a good idea of how to accomplish this. Users can view the Café 472 menu (without price/option groups), create a group order (Grub), select items for the order (SubGrub), and share the link to the Grub so other users can create their own SubGrubs for that Grub as well. Currently, the users can not complete a Grub.

**Lessons Learned**

Menus can be very complicated. Each menu entity retrieved using delivery.com’s API is composed of submenus, which is composed of more submenus or menu items. Each menu item can contain any number of price groups or option groups, which can contain more options or more option groups. Some option groups only allow one option to be chosen; other option groups allow any number of option groups to be chosen. It is surprisingly difficult to parse through a menu and display it in a user-friendly way. I learned not to underestimate the difficulty of any component of a project.