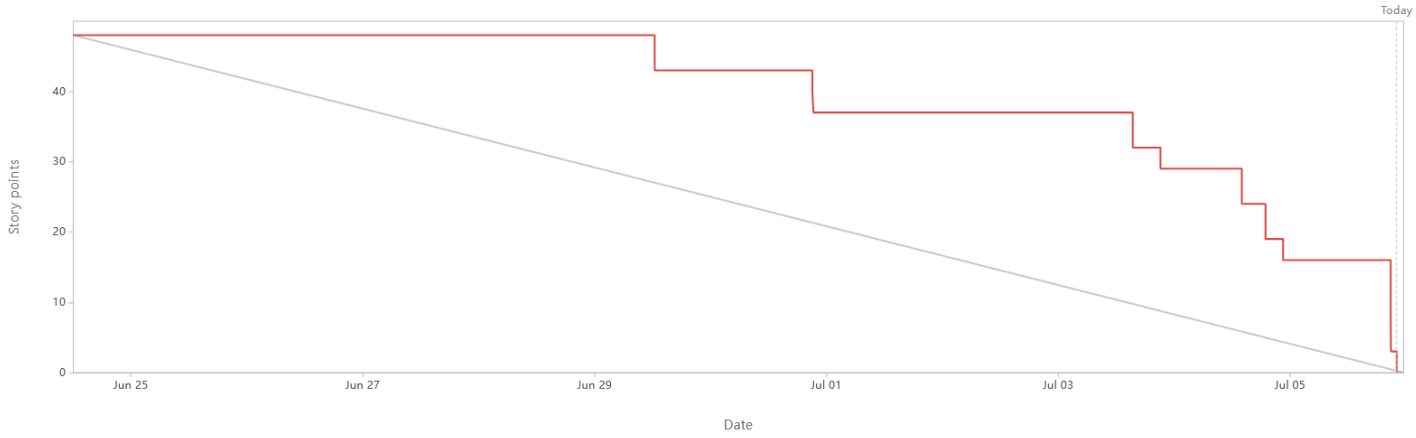
Burndown Chart



Analysis

Based on story points completed, our actual curve on the burndown chart differed greatly from the recommended guideline. We have consistently remained above the guideline, and there was a plateau of several days where no story points were completed, with the first completion of story points during the weekend before July 1st. We believe this is primarily because of two reasons.

1. Many of our members were busy with other courses, so there was limited time that team members could dedicate towards implementing their features and performing code review. Combined with the fact that we aimed to have the majority of our features ready for review by the end of the June 29th weekend, substantial progress only began to occur during this weekend as members raced to try meeting this soft deadline. Our use of this suggested time frame, which was originally made to avoid having to deal with merges last minute, may have given the false impression that we could delay pull requests.
2. The individual user stories this sprint were more complex. It required a few days for members to familiarize themselves with the required technologies, which meant that we could not immediately finish certain tickets early in the sprint. For example, this was the first sprint where we had to work on features pushing real-time information to the user, to alert them of changes in the status of food deliveries. This required some members to perform some initial research on possible technologies to use, which delayed the start of feature implementation.
3. There were more blockers in the sprint. This was the first sprint where we explicitly added dependencies between user stories on Jira, and compared to Sprint 1 each story required more coordination with other members to ensure that we were being consistent in our coding conventions and in the database schema. For example, we finished a major refactoring task for frontend request handling, which also required members to update their existing in-progress pull requests to accommodate these new changes. This delayed

the creation and approval of many pull requests which relied on API calls to the backend server.

Some of the issues could be addressed by breaking down user stories into smaller tasks, which can be completed in an hour or two. This allows for team members to accomplish more story points even if they have a busy schedule, resulting in a more gradual curve. This also means that story points can be checked off earlier in the sprint, since simpler tasks would be faster for team members to transition into at the beginning of the sprint. More frequent meetings or communication early in the sprint could also help us more quickly align on updates of the database schema and code architecture.

Velocity Comparison

Compared to the previous sprint, we tackled an almost identical amount of story points (49 for Sprint 1, 48 for Sprint 2) across the same amount of time (roughly 2 weeks), so our velocity did not change significantly. This was not surprising, since during the sprint planning meetings we actively tried to divide the user stories evenly across the sprints and between team members. We have also been successful in completing our sprint backlog for each sprint thus far. So the total velocity between sprints is not expected to differ greatly going forward, unless there are unforeseen issues or changes in scope that would contribute a significant number of story points.