Team 91 - Final Project Peer Evaluation

Shuai Shao, Tianyuan Ding, Xing Chen, Huiwen Zhang

We think our group set a reasonable amount of features to implement (as a four-person team) and we made a compact schedule for deadlines of implementing those features. We had a good start. However, due to the poor time management, we completed some features in either a much less complexity or less robust than we had expected before. For example, we planned to use Ajax to polish most part of our project and minimized the number of page refreshes(such as preference settings and posting comments) but ended up in not having enough time to implement all the Ajax. This is mainly because we didn't leave much time for continuous integrajaxation at the end of each sprint, and as a result, on the last minute before the project was due, we spent most of the time simply on fixing the bugs that occurred in our final version instead of adding complexity, which is really bad.

The another thing to note as a result of poor time management is our last-minute nightmare on deployment. On the day before the final presentation, we pulled on all-nighter simply on deploying our application on AWS and didn't have much time on fixing the bugs of accessing some static contents that we moved to S3. As a result, our final application still cannot appropriately display some static content like the background image of virtual restaurant. Therefore, we conclude that we didn't complete the final deployment very well because the application we deployed on public should be at least the same as what we have on our local Git repo.

Overall, we failed to communicate our time constraints well. In particular, we underestimated the amount of time other course work was taking, with each team member occupied with projects and homeworks from other courses. As a result, we've planned an overwhelming amount of expectations for our work with little consideration on time constraints of each team member. We kind of realized the issue after the first sprint since we failed to bring out our plan well. In the following sprints, the revised plans still didn't work out well because of the remaining unimplemented or raw work accumulated from previous sprints due to rush of time.

Considering communication of skill sets, on the other hand, we were able to divide work in a good manner. In our case, one team member Tianyuan has taken Machine Learning course and is familiar with search result suggestions. Therefore, we assigned the very part of our application to him. Also, as all team members were new to Javascript and frontend in web applications, we assigned most of the front-end work to one team member Xing, who is a fast learner and an all-star of our team. Other work was divided further according to each team member's interest, which may have promoted the team's efficiency.

We had regular weekly project meetings to discuss task divisions and set individual deadlines. Since we knew each other well, there wasn't much unnecessary concerns and mental burdens such as hurting each others' feelings. Therefore we think we had effective communications. Product leaders were trying to contact each member to find out if he or she has difficulties finishing his designated task. In addition everyone was able to show up on time since we always met shortly after class about 10 minutes and pointed out challenges we are facing, and had coding sessions together on Friday afternoon when everyone was available.

While it was unavoidable for us to encounter team issues, we were able to communicate efficiently so that those issues could be finally resolved. For example, when we did peer evaluations after sprint 2, Shuai found a bug in the code (recipe edit) that was written by another team member Tianyuan. Since we just presented our sprint, we felt very regretful about it. However, knowing that bugs were inevitable, Shuai just sat down and discussed the issue in the group meeting instead of blaming the work that the other person did. Doing in this way, we were able to fix this problem in no time. Reflecting back, if that member was very angry towards the lack of quality in other's code and criticized that person, it would be impossible for the issue to be resolved.

Overall, we divide our project into several sub-parts and we all agree on fractions that other team members have accomplished. In specific:

Xing Chen:

- Front-end html pages design/ Javascript
- Website structure
- Finding sample data to populate database
- Virtual Restaurant
- Menu planner
- Recipe view page

Shuai Shao:

- Database/model design
- Recipe manipulating functions(add/edit/rank/comment/delete/favorite)
- Monthly/Seasonal recipes recommendation
- Recipe suggestion by ingredient(Use leftovers)
- Third-party shares

- Preference settings

Huiwen Zhang:

- User register/login/confirmation emails
- Password reset
- Third party login
- User level feature
- Advanced searching (auto-complete and categorized search, and also the compact display of searching bars)
- Trial feature
- Profile picture/recipe picture cropping
- User profile editing

Tianyuan Ding:

- Recipe manipulating functions(add/edit/rank/comment/delete/favorite)
- Recipe recommendations based on taste preference and search history
- Populating ingredient samples

Every team member contributed to project deployment onto Amazon AWS and performed QA by manual testing.

You may refer back to our site by routing to:
ec2-54-69-24-140.us-west-2.compute.amazonaws.com