Antoine Saliba

Laker Books

Post Mortem

**What went well:**

The actual development of the application went very well. I was already familiar with how to use play framework after having taken Web Services. I remembered how to get information from my forms and pass it to my java backend application. I also had experience connecting a MySQL database to my java application so that was relatively easy. The database took about a day to design and the fact that I am currently taking the database course that is offered really helped me figure out how to structure it. All of the backend features were either implemented early or on time based on my Gantt Chart. I did a lot of research to find an API that had everything I needed and ended up choosing Direct Textbook. Even though they changed a bit of their API lately, it was really easy to use and had all of the information that I needed as well as pictures for most of the textbooks. Implementing bootstrap took a while but after learning how to use it, I was able to smoothly incorporate it into my application. It went very well and was able to easily change some of my html pages to make them responsible and look great on mobile devices. I also used a few other JavaScript libraries to do things like autocomplete the currency when a seller entered a price. It took a while to find one that did what I wanted and that worked but it was easy to implement.

The actual design of the webpages went very well as well. At the very beginning of the semester, I had created some mockups of what I wanted my application to look like. I wanted to emphasize the simplicity of it by having only the needed things on the screen and leaving a lot of free space. I had previously made webpages but had never really done much with making them look good. The process went very well though and, in my opinion, my webpages look extremely clean and simple and even better than I thought they would be. The actual designing of how everything looks did take some time to accomplish but I had put adequate time to do so in my Gantt Chart.

**What did not go well:**

I did not have any major setbacks while creating this application. However, the user interface and the webpages did take longer to implement than I had originally thought. One of the problems I faced was not having the webpage redo the action every time the page was refreshed. For example, if a user posts a book online, he is then directed back to the main page of the site. The problem was, if the user clicked refresh five times, it would add post five identical copies to the database and show up as him having posted five things. Another thing that took some unexpected time was the visual queues that were needed. Every page needed to have a way of indicating information that the user had either not entered or not entered correctly. This involved making the textboxes red and JavaScript was used to have temporary messages pop up at the top of the page when the user did something incorrectly or forgot to fill a textbox. This was a longer process than expected and was not thought of when originally doing the Gantt Chart. It was not very difficult to implement but was time consuming.

Another part that took more time than already planned in the Gantt Chart was the authentication. I was confident that I would be able to get a either parse the Gmail page with everyone’s information or get an up-to-date copy from SUNY Oswego. However, I realizing that going through SUNY Oswego to get access to their database with every user would take too long. I did, however, find a directory that everyone with an Oswego email has under Gmail contacts, which had the email addresses of all the students and staff for SUNY Oswego. I tried parsing this but I was not able to download the html pages, but rather the Gmail login page. Once I realized neither option would work I decided to just manually add each of the 180 pages of contacts to my Gmail account and then download it as an excel spreadsheet. The actual code needed to use the spreadsheet to authenticate users did not take long as expected but actually getting my hands on the spreadsheet with every SUNY Oswego email took longer than expected and was tedious.

**What I would do again:**

The thing I thought really simplified this application was using Play Framework. It was a great decision to use it since I already had experience with it and it made the communication between my webpages and my backend Java application simple and straightforward. The actually coding of the application was very enjoyable and the communication between my application and the webpages, database and API was easy and happened without any major problems. The error messages that the Play Framework gives back also were easy to understand and I was able to quickly fix any mistakes that were done. Even the use of JavaScript, Bootstrap, jQuery and other JavaScript libraries was very easy and Play Framework did an excellent job of keeping everything organized in separate folders.

Testing was done very well for this project and I would do it again. Instead of waiting till the very end to do a lot of the testing, I tested as I was developing. Once I got to the end of a phase in my development, I would test that functionality as well as the prior stuff I had done. So when I finished by emailing process, I made sure that works and also that everything else worked as well to make sure nothing else was broken. The same was done when I made my final changes to my frontend, I made sure to test all of my components constantly. This, in the end, made my development a lot smoother because once I was done with one part and it was all tested, I knew that problems that I had were most likely on the new stuff I was developing rather than the old stuff I had already tested.

Another thing I would do again is starting with some planning, Gantt Chart and mockups. These steps were critical while doing this project. The Gantt Chart really helped keep me on track and doing it forced me to really think of every single thing I would need to complete this project. Of course there were a few things that were not thought of that needed to be done such as the visual queues, reporting a user, currency auto-complete and “How It Works” section, but those were small parts that did not take very long. The previous project classes I have taken really prepared me in this aspect because I knew there would be a few things I would forget and knowing that allowed me to extend a few of the major parts such as the backend and frontend when I was doing my Gantt Chart.

**What I would not do again:**

There really was not anything major that I would not do again. However, I would switch up how I did the frontend development. What I did for this project was have the user interface complete for a computer and then later on implement bootstrap to make it dynamic and look good on devices of all sizes. Instead of doing it this way, I realized it would have been easier to simply start off by including bootstrap because, not only did it allow my website to look different on mobile platforms, but it also helps with the moving and resizing of components when the computer screen size was changed. Some of the things I had already done to make the website look good using percentages were later replaced by parts of bootstraps that were simpler to use.

**Conclusion:**

Overall, this was a really great experience and I loved working alone on an entire project for the first time with so much freedom. I was able to make very realistic deadlines and meet most of them. When I fell a bit behind in an area, I always had sufficient time to finish because I would keep working even if I was ahead of schedule. This allowed me to get some parts done early and spend more time working on a few things that either took a bit longer than expected or that I forgot to include in my Gantt Chart. I really enjoyed working on all parts of this project and it helped me improve my skills as a well-rounded programmer. This project forced me to take all the responsibility but I also did not have to wait for anyone else to finish something. The planning done at the beginning kept me on track throughout the entire semester and took into account that I would fall a bit behind in a few parts, allowing me to take the needed extra time without falling behind because I had planned sufficient time for each part.