Reflective Essay  
Alex Chan

In this Reflective Essay, I will reflect on the design project and course lectures. I begin each new topic by repeating the question, and then elaborating on the subject. In terms of the Design Project, I felt the course met my needs in showing me how a full stack application is carried out. Before this course, I never knew how languages were tied together, and the design project really helped reveal how. I also felt the Course topics were useful, because I was individually forced to contemplate how I was going to structure the code way before the code was even written

In the course, I got to experience both the server and client side of coding. Leading up to Demo 1, I worked on the server side, which was written in Python under the Flask framework and backed by the PostgreSQL database. Thus, I had to learn python and SQL and then use it to make several “get” functions and the joinLeague() function. In terms of the Design project, I learned about restful API endpoints, HTTP verbs, SQL queries, how to use the heroku database, how to create a web pages with HTML/CSS, and how to use Github efficiently. In terms of Course topics, there is too much to list; however, some notable topics that I specifically implemented in the Stock Fantasy League documentation included user stories, functional requirements, traceability matrices, Gantt chart, project management, etc.

The most difficult aspect of this course was meeting the standards of the Design project. Many of the students had only taken basic coding courses. In my opinion, I feel this course could be a senior design project as an accumulation of all previous coding classes. Students that had personal previous coding projects were able to keep up, but I know many students fell behind and weren’t able to follow all the new coding languages they had to implement.

The most difficult aspect of team projects was breaking up the code structure equally. There are nine people in a group, each with different coding backgrounds, and with different rates of learning new languages. It was extremely difficult in saying which team member would work on what part of the code, because a lot of times the code structures relied on each other.

On the group design project, there were several issues that my project confronted. These issues include conceptual problems such as differentiating between player IDs (pids) and user IDs (uids), software problems such as inserting ids into a column of arrays, and breakdown or responsibilities problems such as assigning tasks equally. If I were to approach the teamwork if I had another semester to work on the project, I would have half the team work on the front end and half the team work on the back end.

Technical challenges I encountered in the development of my software product included considering all the functions a user needed and syntax errors which naturally comes with learning a new language. Topics I feel that should be covered in a pre-requisite course for this course include Software Methodology and Intro to Data Management. This class should be a senior design project for computer engineering/computer science majors.

The software engineering techniques I learned in this course that helped me address the challenges listed above include user stories and user requirements. Techniques I found the most useful to my group include functional requirements and interaction diagrams. These two techniques were especially useful because my group was able to decide what functions we needed and how the modules related to each other. Thus, when my group was coding, we always had a model to reference and stay consistent with.

Techniques I found the least useful were the Gantt chart, because it was difficult to predict how long something would take to implement before it’s been implemented. For my group, this was all of our first full stack application, so implementing some things that seemed easy took longer than expected and vice versa.

Based on my experience, I think it is better to work on an already defined project, because it is easy to point out areas of improvement and know that the project is feasible for college students. If something has been done to some degree in the past, there is confidence that it can be done again but better. Also, everything can be improved.

Challenges I individually experienced by working on a software project as part of a team was working on an assignment that didn’t rely too heavily on other parts. Separating the functions by HTTP verbs was an easy way to do this though. Benefits I individually experienced by working on a software project as part of a team include having different people to ask questions to based on who is better at something. Because we all came into this project with varied coding backgrounds, it was nice to have someone with more experience in a topic as guidance if there was ever trouble.

Knowledge I feel might have helped me with the project development include knowing python and SQL before coming into this class. I think having a fundamental knowledge on how all the languages relate to each other would be very useful as well.

Overall, the course met my needs in exposing me to the languages in processes of creating a full stack application; however, I feel that was more group related and not course related. The course did show me how to document properly and think everything through before coding though and that was extremely useful skill in any project. Topics covered that helped in advancing my knowledge of software development include the lectures covered in class.