**Software Engineer and Team Management**

When the objective of the project was announced in the class of Software Engineering, our team of 8 people had only the bare-boned idea of the mission that was supposed to be accomplished over the course of 16 weeks. While the professor took his time to carefully elaborate the goals of this project, it took multiple emails and after class conversations to gain access to the true big picture of the software our group was supposed to engineer. Essentially, we were building an interface for a system that tracked points of the red team and the green team for a game called Laser Tag. Many groups were confused about the objective and failed to raise their confusion with helpful advisors, and this setback led to recipe of a disaster for them in later sprints because they had the task of trying to understand the project after the project had already started. We understood early on that open communication was a critical step in progression of a team. Therefore, we ensured that every individual in our team was active on multiple group chats to raise concerns about any challenges that needed to be solved. We had an expert scrum master in our team that had actual industry experience and was responsible for setting up group chats and team meetings for sprint planning. After the project objectives were met, the whole group understood that team management with open communication and roles dedicated to ensuring that was the single most important element that justified our group’s success.

There were tools that enhanced our scope of the workflow, such as Trello and Slack. Trello was a great improvement to our productivity and helped us avoid redundancy in handling tasks in a segregated yet connected manner. We used slack to manage status updates for each individual and for raising issues to professors and teaching assistants. Slack is one of the most popular tools used for team communication and is like discord, except it’s used exclusively to talk about work. One of the biggest challenges was choosing a web framework to work with, which I consider a tool to get the job done. Choices involved NodeJS, Spring Boot, Python Django, C# Asp.Net, GoLang, etc. We chose our backend web framework as NodeJS and ReactJS for our front-end framework for multiple reasons. Web frameworks like Spring Boot has a steep learning curve that would eventually have been a time drain for getting our minimum viable product deployed. The rest of the frameworks had documentation that was hard to find. NodeJS was the all-around winner than had JavaScript used in backend, same as front end. This made the learning portion of the project more possible and realistic for the time allotted. As always, learning is a painful process that yields benefits in the end, but also has its drawbacks in terms of time consumed and efforts involved by the team. We were looking for minimum output that yielded maximum output and our team organization followed that principle quite closely. The team members followed the same resources to learn the coding part of the project. This allowed group to have similar thinking pattern and minimize conflicts when it came to attacking a problem constructively in a particular method that the group agreed to.

There were time conflicts between team members, and this was one of the biggest challenges since there was rarely a meetup time where each of the eight members showed up. Instead, each team member discussed the times they were free, and we made a table to find the similarities in these times to setup meetings. This worked out well in the end and weekends seemed to be the only similarities between each group member due to job shifts or other college work during the rest of the week. We later decided that it was best to just have more chat sessions, as that gave everyone the freedom to reply on their terms instead of finding spare time. The next challenge that the group faced was the longing for a leader naturally. Without a leader, the group fails to strive in a direction far enough to succeed. Initially, the group was going in randomized chaos manner and that led to our crippling grade in sprint 1. I created a Slack workspace, following the instructions provided by the professor on day 1 of the introduction to the project. My mistake was misinterpreting the words presented and not openly communicating with the group and asking the professor or teaching assistants for more information. This led to a grade of 57% on Sprint 1 and caused a havoc that could have certainly been averted. A call to leader was necessary and I felt that I was responsible for this.

Every team needs a vision to fall back on when times get tough and a leader to guide them towards the vision. Going in a directionless manner leads to futile results. I was making the most commits on GitHub and that was evident to the team members, so they naturally felt that I had the most contribution towards the needs of this project and called me the “tech lead”. It was a great feeling to be responsible for something as credible as this Laser Tag project. The benefits to this title could not be ignored because the group listened to everything I had to say and decide. One of the responsibilities was to remind members of the status reports to see exactly how each member was progressing in their endeavors and to help them with any troubles they were incurring. I had to make sure that every individual was having a work life balance and that no one over exhausted themselves over the needs of this project. I had to keep the learning process to a minimum, so I went with the task-oriented approach. The task-oriented approach works by dividing the tasks into microtasks and letting the group members taking their pick of the litter. For each feature in the system interface, there were short yet helpful YouTube videos that saved plenty of time and made life easier. This is better than learning the whole tech stack and later applying the learned knowledge. Learning ReactJS takes 6 months and learning NodeJS takes 6 months, which is a major problem since no one in our group knew the simple things properly such as vanilla JavaScript. Therefore, we only learned what was necessary to get the job done. Whether it was implementing sockets or creating a React component, everything was learned on a as needed basis.

There are a few things I would have done differently had I known from the experiences before. Procrastination is a problem that many humans suffer from and it’s usually the anxiety of deadlines that is the push unfortunately. This project was a team effort and if one person had not made a commit before the deadline and the project’s functionality was at stake, this could hurt the whole group’s grade. Therefore, we made sure that everyone was upfront about their progress on the commit. If someone was at a risk of not reaching 100% of their progress, I would setup meetings with that person and try to understand the reasoning behind that. Life happens and some people can’t reach their end point, and this is when I step in and delegate the tasks in a flexible way.

Disobedience among team members to the project requirements can be a major problem, but our team was very committed to the same goal and vision. If hypothetically, I was in a group that had a member who wasn’t willing to put in any effort towards the codebase, I would try to reason with them and take their viewpoint and see if we can reach an agreement. In rare cases will an individual be totally ignorant towards the project, but if this was to be the case, then there would be no choice left but to escalate to professors and teaching assistants. Fortunately, this didn’t need to be the case in our group.

I was impressed by the ability of different tools to connect to create a usable end product for the users across the globe. The ability to share a link to anyone across the world to showcase your project is the motivation for me. The skills utilized in this project are used in industry for several years and I want to be part of this as well. In 2013, ReactJS was released by Facebook and that changed the way that the user interface operated in that now, everything is a separate service and a component. Thinking of each feature as a separate part of the page reminds me of human anatomy because each part of the body has a dedicated functionality, similar to our website, which has separate containerized features. Rest API is also an important tool that helps connect the backend and frontend by regulating http requests and sending information back and forth.

Overall, our group managed to successfully finish the tasks at hand successfully following the agile methodologies and sprints. We learned a lot about tech stacks and the advantages and limitations of certain tools when given a choice. Working in this kind of environment showed us a brief overview of the real enterprise world and prepared us for what is to come. I definitely feel more confident in my ability to engineer a software that can fulfill the requirements of consumers in their daily lives. I’m inspired to create my own software as a service (SaaS) on the side when I get time. I learned about team management and what it means to be a leader. Coding was a fulfilling part of this project and was an exercise for my brain. I learned strategies to optimize backend and frontend for best performance on the webapp. Heroku had limitations of 30 minutes before it became inactive until the user entered the webapp. I made an adjustment and sent an http request every 29 minutes to keep the site up and running 24/7 and this optimization helped avoid cold starts at the beginning of webapp which stole 10 seconds from the user’s time. Our team’s improvements led us to hundred percent on all the sprints except sprint 1. Our strong work ethic and continuous involvement to complete work on each sprint and reporting in sprint meetings led to successful team progression. I’m thankful and honored to be a part of this project. My career has been streamlined because of this course and I know what to make of my life after I graduate computer science. Building real world solutions to real world problems is one of the most satisfying feeling an individual can feel.