Senior Design Self-Assessment
Brendan Link

I had numerous individual contributions to the Safe Assistant project. One of the key strengths that I identified during my initial assessment in the fall was my ability to write great user stories and acceptance criteria. Working as a Product Manager during three of my COOP rotations, I learned firsthand how to properly write acceptance criteria and generate user stories for real-world projects. This knowledge is important, as writing user stories in practice can be much more difficult than what we learn about in theory during school courses. I was able to use this knowledge and expertise of mine in order to lead the group on our design phase writing user stories with their acceptance criteria. I was also able to use the knowledge that I gained during the Python programming course. Since our entire project was written in Python for the Raspberry Pi, I used this knowledge during the implementation phase of the project. During that course, my final project made use of third-party libraries and integrated those into an application. The Safe Assistant similarly used many third-party libraries for offline voice recognition and voice synthetization. Finally, I applied much of the knowledge that I learned when I did a COOP rotation as a software developer. During that COOP, I collaborated with a large team of engineers and used version control tools such as git in order to manage our code and combine it together. I was able to use these skills when working with Ethan and Alex on the Safe Assistant, since we used git and GitHub in order to manage our code.

During the implementation phase of the project in the spring, I was able to successfully complete numerous parts of the Safe Assistant. One of the major components that I was able to complete by myself was intent processing. This is a crucial functionality for a voice assistant; if the device can understand the words spoken to it but can’t determine what the user means, it will not function properly. In the Safe Assistant, one of our major design focus points was that our device would be highly configurable and customizable. Users are easily able to develop their own applications and can easily install them by simply adding them to the applications folder. Because there are such a wide range of built-in functions and many more that can be added by users in the future, it was important that the intent processing functionality is able to handle a dynamic array of user intents. In order to solve for this obstacle that I faced, I used knowledge that I learned in the Requirements Engineering course that I took in the Fall semester. During the course, one of the major projects that we did was writing a tool that automatically matched similar requirements to one another using a Jaccard index on the words of the requirements. I saw that the Jaccard index could be used in the Safe Assistant in a similar way. We could make a set out of the words that the user spoke, and a set of words that can be used to trigger each of the commands on the Safe Assistant. Which command most closely matched the set of words that the user spoke would be run. Another obstacle that I faced here was that if the user gave the Safe Assistant a command that it didn’t have, it would always call some command on the system that it most closely matched. In order to solve for this, I added a cutoff threshold if the user’s command was not very similar to any of the actions on the system. This would inform the user that no such command existed on the Safe Assistant. I then integrated this functionality in the server code that Alex developed. The other major component of the Safe Assistant that I was responsible for was voice synthetization. This process was straightforward since we used the espeak voice synthesizer available for Linux. I was able to integrate this successfully with our client code.

Throughout both semesters, our group was able to accomplish a lot. During the design phase in the fall, our group was able to collaborate on a design of a large software system. This was an important learning point for all of us. Since in our COOP experiences as software developers we usually were responsible for only implementing a design that others created, it was a great learning experience for us to collaborate on a software design. Our design was robust and was able to guide us through the implementation of the Safe Assistant. Also, it was important for us to be flexible when designing our software. We understood that when developing a system, obstacles can come in the way. Because of this, we wanted to ensure that our design would be flexible enough for us to adapt it during the implementation phase. Our group was successful in accomplishing this. During the implementation phase, our group was successful since we were able to build a fully functional voice assistant. The final product of the Safe Assistant that we delivered closely matched the specifications that we outlined during our design phase. The Safe Assistant is successfully able to listen for a hotword, listen for commands, determine which intent the user has, process this intent, and speak the results back to the user. All of this functions entirely offline.

While working through both the fall and spring semesters, we learned a few key lessons about how to work properly as a team. The first key lesson that we learned is that in order to stay accountable, we needed to meet on a regular basis. We found a time on Monday and Wednesday of every week where we met for one hour. During these meetings, we would begin by discussing course updates and our future schedule, and then we would each provide updates on our own progress that we made since the last meeting. Our team found these meetings to be very successful, as they held all members accountable, and information traveled quickly between team members. Another important lesson that our team learned during the project was having trust in others to finish a task. While it can be exciting to work on each part of a project like the Safe Assistant, we could not all work on each component; rather, we had to split the work and trust in one another to deliver the component completed to project specifications. Each of us was responsible for the engineering decisions in the components that we implemented, and the other group members had to trust the implementation. Our group found that by trusting one another to do our own tasks, we were able to accomplish much more than just one person could have by themselves. Each member on our team took responsibility to get work done, and each team member put in an equal amount of effort toward finishing the project. I would especially like to call out the great work that Ethan accomplished at the end of the project. During the last week of implementation, Ethan was able to get over a large obstacle he had faced for weeks and was able to successfully implement hotword detection in time for us to include it in our expo demo of the Safe Assistant. Overall, our team members each contributed greatly toward the completion of the Safe Assistant, and we all learned numerous lessons about teamwork throughout the year.