Capstone Assessment

Johnathan Lemon

The capstone is a way for all collective experience through the years of classes to come together. Every class teaches in a bubble and seldom branches into other topics outside of a small idea being related in one class. Some classes do build of each other, but others often run in parallel to each other. The capstone is a way to show how all can be utilized in tandem with each other. It is a form of a final point where all ideas and concepts can come together and mesh to be used together. It shows that all the academic experience was put to good use and the knowledge was received and can be used and built with other ideas learned. Overall, the capstone is a way to showcase abilities and progress as a computer scientist gained throughout the entirety of the academic career.

Many courses throughout my college career have helped prepare me for a project in developing software. One class that greatly shaped my coding ability was Data Structures (CS 2028C). It showed how to think and problem solve in the lens of computer science and coding. It was the first real coding class and opened the gateway to the world of writing code. The next big one is Software Engineering (EECE 3039C). It showed the flow of software development and taught teamwork and collaboration skills on large projects, very similar to what coops taught. Database Design and evaluation (CS 4092) also greatly prepared for the project. Databases are a powerful tool and knowing how to implement and utilize them efficiently is an important skill to have for projects like this. Overall, all the classes prepared for development of large projects, and these are just three of the main ones with larger influence. As mentioned, these courses taught technical skills: how to implement a linked list, how to utilize a tree, coding in multiple languages, making a schema, etc. However, they also taught skills like collaboration, communication, and problem solving. All these classes taught a skill whether technical or non-technical, and the capstone is now bringing all these skills together

Coop experience will also greatly guide this project. Coop not only provided real world experience and firsthand experience to work similar to that of a capstone. It taught how to work with large codebases for large projects being worked on by multiple people. It also taught many external skills like teamwork, deadlines, etc. Coop experience showed how to prioritize and ration work to a team when there are multiple people and multiple moving parts. It taught how to problem solve and debug as a team or how to work on something at the same time with someone. These are all skills crucial to not only the capstone project, but to software development as a whole. In general, coop provided hands on experience to work on large issues real time and a chance to not only apply what academia has been teaching, but also learn tricks and practices that academia may not have covered.

The approach to designing a solution is first brainstorming. Ideas need to be formed by the group and a common idea needs to be found. The group has proposed many ideas to many problems, and which one is the final one is still undecided, but they are all exciting ideas. All the ideas are interesting, and the motivation comes in the form of satisfaction that comes with solving a complex problem and presenting a solution. No matter what project is chosen, I look forward to working with the team and coming together to solve a problem. The approach used top solve this problem will be that of one solving a set of smaller problems. The larger problem will be broken into smaller problems which will also be turned into smaller problems once more. This method of breaking down problems make finding what needs to be done and doing it much easier. This is because they are small and manageable and will ultimately build back to the larger problem that is desired to be solved. These problems will come in the form of user stories and once all the expected problems / user stories are complete, the final product of an application will be done.

Once user stories are established to solve a larger problem, work can begin. Self-assessment can be performed via seeing how many stories are being done and if they’re meaningful. For example, if a small number of stories are being finished or only small easy ones, then it is necessary to contribute more and take on a larger workload. Stories also function as a good indication of quality of work as they point back to what you’ve been doing and if the areas in the stories are performing poorly, then most likely it is related to a change I worked on. As mentioned earlier, the final product will be an application. This means that we will know we are done when there is a functioning application that meets all criteria. To judge the quality of the work, we would simply look at the quality of the app and if it meets the desired needs in a quick, easy manner that does not break.