

Miranda Putnam

August, 2022

CS 470 Final Reflection

Presentation Link: <https://youtu.be/uZF6eH1yaXY>

### **Experiences and Strengths**

Throughout the Full-Stack Development course, I have learned many new skills as well as developed existing ones, which have helped me become a more marketable candidate in the Computer Science field. These include using a cloud service such as AWS, being able to articulate the intricacies of cloud computing, and building a complete full-stack application. Many companies are moving towards cloud computing and development as it continues to grow in popularity and offers many benefits versus traditional computing. Because of this, the skills I learned in this course make me a very valuable candidate to those companies, as I am now well versed in building applications and deploying them to the cloud.

My strengths as a software developer include paying acute attention to detail, understanding the needs and necessary components for a software, and knowing which tools are going to serve an application best. When developing software, it's very important to understand why you are building it and how it is going to serve users. Throughout my time at SNHU in the Computer Science program, I have learned how to understand user needs, build software that meets those needs, and pay close attention to detail when building and designing software.

Examples of roles I am prepared to assume in a new job include analysis, design, and quality assurance. I am a perfectionist, so I love analyzing the small details of things, ensuring that they meet their criteria and continue to serve their purpose going forward in the best way possible. I also enjoy collaborating with others to ensure every idea for a project has been considered so the best possible version of it can emerge.

### **Planning for Growth**

Throughout this course, we have focused on cloud development, and I have learned many of the necessary components and skills relating to it. Cloud computing can be very beneficial to a myriad of companies and applications in terms of management and scale. It can be used to manage scaling of an application, as a serverless model allows scaling to happen automatically as your application grows and changes. Cloud computing also offers a fully-managed experience through companies such as AWS (Amazon Web Services), so errors are easily handled and managed, as you have the support of the company behind you instead of having to manually manage and handle errors yourself. Cost is also an important factor when building an application. Cloud computing offers a solution to this in the form of a pay-as-you-use model. If you use a third-party service such as AWS, you pay the company for use of their servers, but you only pay for the amount of server space that you use. If your application grows and uses more server space, your cost automatically goes up. This is a great solution when compared to traditional development, as you'll never have to guess how much server space you will need or overspend on servers and other equipment. When comparing models such as containerization and serverless, I find serverless to be more cost predictable. This is because of the reasons mentioned before: It offers a fully-managed, scalable, pay-as-you-use experience. You will always only pay

for what you need and use, which eliminates the guessing game and over-paying for storage and equipment. Containerization can be convenient, but it can also come with soft costs due to the fact that it is user managed.

There are some pros and cons that would be deciding factors in plans for expansion when considering cloud development versus traditional development. For cloud computing, pros include easy scalability, the pay-as-you-use model, and a fully-managed experience. Cons would include being locked into a third-party service and facing a difficult transition if you were to eventually want to leave that service. For traditional development, pros include more user control during expansion and flexibility in the software used to build the application. Cons include less predictability when scaling and pricing out the cost of expansion, as well as manual management. Overall, if I were a brand new web application owner concerned with expansion over everything, I would choose a cloud development model because of the ease of use in scaling and cost.

Lastly, elasticity and pay-for-service play a large role in decision making for planned future growth. If you anticipate fluctuation or rapid growth for your application in the future, elasticity and the pay-for-service model are going to be very important. This is because elasticity will automatically scale the resources needed for your application, which will also help minimize infrastructure costs. Instead of trying to predict the amount of resources you will need for your future growth and over-paying for resources just to be safe, elasticity will automatically take care of everything for you. The same goes for the pay-for-service model, as your costs will automatically adjust based on the growth of your application and the amount of resources it is using.

