Thor Nielssen

CS499: Computer Science Capstone

Southern New Hampshire University

23 June 2024

**CS499 Professional Self-Assessment**

Throughout my coursework at SNHU, I have struggled, stumbled, and fallen short of my goals a few times – however, I have always gotten back up, worked with my instructors, advisors, and tutors, and persevered to complete a finished product that is worth showing off in my portfolio. My artifact selection from IT145 enabled me to truly demonstrate how far I have come as a computer science student, showcasing my earliest work in contrast with how much I have learned since. Contrasting this early work, other courses involving emerging trends in computer science and discrete mathematics have broadened my capabilities and enhanced how I approach and solution through a problem. These courses helped me understand algorithmic principles and address those needs accordingly within a program.

Thanks to entire courses focused on Scrum and Agile methodologies, I have also learned a lot more about what it means to be collaborative and communicative – including working within a team as well as providing valuable, non-technical advice to stakeholders and other vested parties. This coursework lent itself well to the development of my video code review and portfolio, because I have a much better handle on how to speak to a broader audience than I normally would. Thanks to this, I am certain that my published material can be easily read and understood by all sorts of people, and that it is free of syntax-related or grammatical errors.

Revisiting my early work also showed me just how much I have retained over the past three years. I have now proven that I possess the critical thinking and design skills to not only revisit and decipher my old code, but entirely re-design it from the ground up in a different programming language. Additionally, I have proven that I understand how databases function and interact in a code environment by implementing them accordingly, along with all of the necessary SQL syntax and operators required for the program to put the database to use. Finally, I overcame one of my biggest challenges – algorithm development. I must admit that I was not confident in my abilities in this category, but once I identified a logical need for algorithmic principles in my projects, I was able to take that need and come up with a code outline based on it. This led to a huge confidence boost and a successful outcome once I finally got the algorithms in the menus working.

All three of my enhancements center around one of my first projects at SNHU from IT145. The Animal Shelter (Rescue Animal) program was originally written in Java and used basic array lists held in memory to manipulate animal objects. These enhancements are slightly out-of-order compared to the rubric, because I revised my original algorithm enhancement – for this reason, enhancements two and three are switched in order.

My first and largest-scale enhancement was to completely re-design the original artifact from Java into Python. Python is likewise a decent language for object-oriented programming, and I knew that by rebuilding this project into Python, I would set myself up for better success in future enhancements thanks to the clarity and conciseness of the language. Throughout this process, I had to research and re-learn the syntax of Python and ensure that I was replacing the original Java code with something that worked at least as well, if not surpassing the original efficiency of the code.

The next enhancement was to add a database to the program, replacing the temporary in-memory array lists with a permanent data source that can be used to manage the animals in the program. While not code-heavy, this enhancement was tremendously difficult because it required learning SQLite and the code required to implement SQL commands and syntax *within* a program written in Python. I was able to get it working after some struggle, and even fixed an issue with the main menu loop in the process.

My final enhancement was to introduce more complex algorithms to the same program, adding functionality that allows a user to sort, organize, and remove objects from the program’s database. This included learning how to use Python’s datetime libraries in conjunction with the proper SQL syntax to reorganize the database in a way that is logical and useful, and not just arbitrary for the purposes of completing an assignment.