Nicholas Wyrwas  
CS-499-10452-M01  
Mrs. Philomena Ogoh  
10/19/2025

Professional Self-Assessment

Each artifact in my academic portfolio as a Bachelor of Science in Computer Science with a concentration in Software Engineering from Southern New Hampshire University (SNHU) tells the story of how software engineering works. Starting with software design and development basics and ending with a capstone project in which all skills are integrated, each course and subsequent project described here was carefully designed to build upon and develop those in previous work. Early courses like Foundations in Application Development and Software Development Lifecycle introduced me to many of the skills used in later work, such as logical branching, decomposition, command-line usage, version control, and modularity. While small, simple programs were written, the progression of lab and project exercises showed me the many iterations and refinements needed before one can consider an application polished or ready for production use. The use of Agile workflows became standard practice in later courses. CS-260: Data Structures and Algorithms added another dimension to the skills covered in the previous two courses by introducing trees, lists, stacks, and hash tables. The emphasis on choosing the appropriate data structure in this and other courses, such as CS-335: Database Development, has helped to make me a more efficient programmer. Additional lab and project work also required the calculation of time and space complexity which also can be applied to later course work and projects.

Client-Server Development and Emerging Systems Architectures and Technologies, CS-340 and CS-350, were two courses that particularly stood out for me as they introduced the concepts of distributed and embedded systems respectively. In CS-340, for instance, we wrote a client-server application in Python Flask using a MongoDB data layer. It introduced me to writing functions to interface with the database (CRUD), API Design and testing, RESTful interfaces, as well as user input validation. In CS-350, one of the projects I completed involved building out some code that had been completed earlier in the class on a Raspberry Pi computer. This involved not just building out the project with specific output displays, but also writing a handler on the client side and a serial reader on the server-side to allow for interactivity. Some of the key principles in these two courses that will be utilized in the capstone artifact are the data flow of data between multiple applications, use of database, as well as performance considerations in such a deployment.

CS-405: Secure Coding in C was all about building secure code, identifying and fixing vulnerabilities, as well as generally being able to code with a security lens. There was a focus on security by design, using the SEI CERT coding guidelines. For me, some key takeaways from this course were being able to identify issues with, and subsequently prevent, buffer overflows, SQL injection, and integer overflows. The artifacts I built in this course involved a lot of input validation to prevent security threats. Additionally, having a modular design with CRUD operations was also helpful as it allowed me to use one code for authentication and CRUD operations. Also, the data flow as well as error handling is limited or compartmentalized in each module, and general application context is not shared among them to improve security and to make the components more loosely coupled. Secure coding principles that are demonstrated in the capstone include some user-input sanitization, data validation and access control, as well as separating out authentication into a separate module from the rest of the application code. CS-410 Software Reverse Engineering was an interesting course for me because it involved a lot of analysis, specifically assembly code analysis to make a higher-level equivalent, in this case, C++. The artifacts from this course all involved breaking down existing code, either a machine code (assembly) or a high-level version (Python). This was beneficial to my later work because it required a deep understanding of both the low-level execution and an analysis of these low levels of computing to then build a higher-level construct. The knowledge gained about system internals here was especially important in my capstone, which also involves Python and developing a web application with Flask. In a professional environment, one would be able to use this understanding to more safely debug application code, as well as design better by building from a lower level in mind to improve the robustness of the application. This may also prove to be useful in debugging and reverse engineering any existing software being integrated.

All of the skills learned in these different classes were synthesized into one final capstone project for the CS-490: Software Engineering Project. It is an interactive Animal Shelter Dashboard written in Python and Dash which allows users to search through, visualize, and map an animal dataset. This application allows for real-time user input to update charts and maps based on user selections. Dash callbacks are also used for data filtering, matplotlib is used for charting, and leaflet for creating map visuals. The final result is an interactive map that is publicly available on GitHub pages. Additionally, it was an opportunity for me to focus on UI and UX design skills and deliver a professional presentation of my work. To this end, the application uses some dark-mode styling and a responsive grid layout for the dashboard. A preview of the finished dashboard is shown in Fig. 1.

For the most part, my work was largely individual. However, several courses, included peer code reviews, artifact documentation, and narrated presentations. These skills are important in software engineering and this was an opportunity for me to practice soft skills like effective technical communication with other professionals and the ability to concisely explain or present complex technical solutions to those outside of the technical team. For me, communication also includes being able to effectively negotiate trade-offs and be able to review the work of others to make course corrections or alternative recommendations.

CS-490 Capstone Artifact Screenshot