CS 499 – Computer Science Capstone

Professional Self-Assessment

Malgorzata Debska

Southern New Hampshire University

Prof. Wasim Alim

August 12, 2025

Professional Self-Assessment

Over the past three years in the Computer Science program, I have experienced a truly transformative journey one that has challenged me to grow as a problem solver, a collaborator, and a professional in the field of technology. What began as an interest in understanding how software works has evolved into a deep, multifaceted skill set spanning software development, algorithms, databases, security, and effective communication with diverse stakeholders.

Throughout my coursework, I have worked on projects that reflect the real-world complexity of computer science. In collaborative environments, I learned the importance of version control, clear communication, and the ability to merge different coding styles and perspectives into a cohesive final product. These experiences sharpened my ability to operate effectively in team settings, balancing leadership and cooperation depending on the needs of the project.

Equally important has been my development in communicating with stakeholders translating technical challenges into clear, actionable language. I have engaged in requirements gathering, status reporting, and solution presentations, always keeping the end user in mind. This skill has proven essential not only for ensuring that technical solutions meet business needs but also for establishing trust and clarity between technical and non-technical audiences.

Growth Through Coursework and ePortfolio Development

From my first introductory programming assignments to the complex, multi-component projects in my capstone, I have consistently challenged myself to take on work that demonstrates both depth and breadth of skill. Building my ePortfolio required me to revisit and enhance prior work, which reinforced my understanding of software engineering principles and gave me the opportunity to implement industry best practices.

One of the most valuable aspects of this process has been seeing my progression from writing functional code to designing maintainable, modular, and scalable solutions. The act of improving my artifacts was not just about cleaner code; it was about developing the mindset of a professional who anticipates future requirements, considers the user's needs, and applies systematic approaches to problem-solving.

Collaborating in a Team Environment

Team-based projects in my coursework taught me the importance of collaboration, version control, and role adaptability. I worked in environments where tasks were divided across team members, requiring consistent integration of different coding styles and ideas. Using tools like Git, I learned how to merge contributions smoothly, resolve conflicts, and maintain code integrity. These experiences also helped me understand that successful collaboration depends on clear communication, mutual respect, and an openness to feedback, skills I will carry into any professional setting.

Communicating with Stakeholders

My projects often involved translating technical concepts for non-technical audiences, such as explaining database structures, system limitations, or algorithm efficiency to stakeholders with varying levels of technical expertise. This taught me to use precise but accessible language, highlight the "why" behind technical decisions, and ensure that project deliverables aligned with stakeholder needs. In real-world development, this skill is just as critical as writing quality code, because even the most robust system is ineffective if its value is not understood by those it serves.

Data Structures and Algorithms

Throughout the program, I gained a deep appreciation for the importance of algorithmic efficiency and proper data structure selection. I learned how to weigh trade-offs in performance, scalability, and memory usage. These skills were particularly evident in my capstone work, where I implemented optimized database queries and filtering algorithms to improve application responsiveness. Understanding these concepts allows me to design systems that not only work but work efficiently under real-world constraints.

Software Engineering and Database Skills

My coursework in software engineering emphasized the value of modular design, code reusability, and systematic testing. I applied these principles in both front-end and back-end contexts, ensuring that my applications were both functional and maintainable. In database related projects, I gained experience with both SQL and NoSQL systems, focusing on schema design, query optimization, and data integrity. Integrating databases into larger software solutions allowed me to bridge the gap between persistent data storage and dynamic application behavior.

Security Awareness

Security considerations were embedded in many of my projects, reinforcing that security is not a final step but an ongoing requirement in every stage of development. From managing authentication to preventing SQL injection and securing sensitive data in transit and at rest, I learned to adopt a proactive security mindset. This approach is critical for protecting both the organization’s assets and the trust of its users.

How My Artifacts Fit Together

The artifacts in my ePortfolio collectively showcase the range of my skills:

Algorithms and Data Structures: Demonstrating database integration with optimized querying and interactive data visualization.

Software Engineering and Design: Showing modular, user-focused application development with robust input validation and error handling.

Databases: Illustrating secure, efficient, and scalable database interaction within software systems.

Together, these artifacts form a cohesive narrative of my abilities: I can design and implement full-stack solutions that are efficient, maintainable, secure, and user oriented. Each artifact builds on the same professional values, clarity, adaptability, and quality, that I will bring to my future work.