Anna Padgett

Southern New Hampshire University

CS-499 Computer Science Capstone 2026 C-1

Prof. Akhil Gudivada

1/23/2026

Software Design and Engineering Enhancement Narrative

The artifact selected for the Software Design and Engineering category is a Mortgage Calculator application. The project was originally developed as a console-based C# application designed to calculate a user's estimated monthly mortgage payment using inputs such as income, purchase price, interest rate, loan term, and related costs. This original version was created in late 2025 as part of coursework in the Microsoft Software and Systems Academy (MSSA).

For this capstone enhancement, the application was refactored into an ASP.NET Core MVC web application to improve structure, usability, and alignment with professional software development practices.

I selected the Mortgage Calculator for my ePortfolio because it clearly demonstrates my ability to evaluate existing code, identify architectural weaknesses, and refactor an application into a more maintainable and professional solution. Rather than starting from scratch, this artifact highlights my ability to work with legacy-style code and improve it using modern design principles.

Several components of this artifact showcase my software development skills. The refactored application applies layered architecture and separation of concerns by isolating data models, calculation logic, and decision-making into dedicated classes and services. This design improves testability, readability, and long-term maintainability.

Additionally, I implemented model-level input validation using data annotations, which demonstrates a security-aware mindset by ensuring that invalid or malformed input is handled early in the application lifecycle. Migration from a console interface to a web-based MVC structure also required clear routing, controller logic, and user-focused presentation, reflecting real-world application development practices.

Through these enhancements, the application evolved from a single-use academic exercise into a reusable, user-friendly tool that more closely resembles production software.

This enhancement met the course outcomes I planned to address in Module One. Most directly, it demonstrates Outcome 4, which focuses on using well-founded and innovative techniques, skills, and tools to implement computing solutions that deliver value. The refactor to ASP.NET Core MVC and the introduction of service-based architecture align strongly with this outcome.

I also addressed Outcome 5 by applying a security-focused mindset through early validation, controlled data flow, and preparation for future database integration. Finally, by documenting design decisions and clearly structuring the application, this enhancement supports Outcome 2, professional-quality technical communication.

At this stage, no changes are needed to my outcome-coverage plan for this category.

Enhancing the Mortgage Calculator significantly changed how I think about software design. One of the most important lessons I learned was the value of planning architecture before implementation. The original version of the application placed all responsibilities into a single class, which technically worked but made the code difficult to test, extend, or reason about.

Through the refactoring process, I gained a deeper understanding of how layered design improves clarity and maintainability. Separating responsibilities into focused classes made the application easier to understand and revealed opportunities for reuse and future enhancement.

The most challenging part of this enhancement was learning ASP.NET Core MVC while simultaneously refactoring the application. Prior to this project, my experience with the framework was limited. Understanding how controllers manage application flow and how models and views interact required experimentation and iteration. However, working through these challenges improved my confidence with unfamiliar frameworks and reinforced my ability to learn new technologies independently.

Overall, this enhancement marked a shift from writing code that simply works to designing software that is structured, maintainable, and aligned with professional expectations.