

Software Design and Engineering Enhancement

Kevin Crandall

Southern New Hampshire University

CS 499: Computer Science Capstone

July 19, 2026

Software Design and Engineering Enhancement

The artifact selected for my software design and engineering enhancement is the Travlr Getaways full-stack travel management application originally developed during CS-465: Full Stack Development I. The application consists of a customer-facing website built with Express and Handlebars, an Angular administrative single-page application (SPA), a RESTful API developed with Express.js, and a MongoDB database used to store trip information and user accounts. The original project demonstrated the integration of multiple technologies within the MEAN stack while providing functionality for viewing, creating, and updating travel packages through an administrative interface.

Although the original artifact met the requirements of the course in which it was created, I identified several opportunities to improve the overall quality of the application. Areas such as error handling, request validation, authentication, configuration management, responsiveness, and maintainability provided an opportunity to demonstrate software engineering practices that extend beyond simply creating a working application. These improvements aligned well with the goals of my CS499 capstone by allowing me to revisit an existing project, critically evaluate its architecture, and enhance it using industry-standard development practices.

Justification for Inclusion in the ePortfolio

I selected this artifact because it best demonstrates my ability to design, develop, and improve a complete software solution rather than a single algorithm or isolated programming assignment. The project includes frontend development, backend API development, database integration, authentication, and software architecture, making it representative of the broad range of technical skills developed throughout the computer science program.

Rather than rebuilding the application from scratch, I focused on improving the quality of the existing implementation. One of the most significant enhancements was improving the application's maintainability. I introduced centralized API error handling, reusable validation middleware, asynchronous controller wrappers, and a custom application error class to reduce duplicated logic and provide consistent responses throughout the backend. Environment-specific configuration values, including the MongoDB connection string and JWT secret, were moved into environment variables to improve security and deployment flexibility.

The authentication workflow was also redesigned to provide better user experience while improving security. The original application combined registration and login functionality into a single workflow. During the enhancement process, I separated registration and authentication into dedicated pages, added duplicate account detection, implemented guest route protection, improved JWT validation, and refined Bearer token processing for protected administrative operations. These changes produced a clearer workflow while demonstrating a stronger security mindset.

Significant improvements were also made to the Angular administrative interface. The user interface now provides responsive layouts, client-side validation, loading indicators, success and error messaging, disabled submit buttons during processing, and a responsive navigation menu for smaller devices. These changes improve usability while creating a more polished and professional application.

An additional architectural improvement involved consolidating the customer-facing website and administrative application to consume the same REST API. The original artifact contained both a static travel page and a dynamic Express implementation. By removing the obsolete static implementation and routing customer traffic through the API-backed travel page,

duplicate functionality was eliminated while ensuring both applications display the same data. This change improved maintainability by creating a single source of truth for trip information.

Collectively, these enhancements demonstrate software design principles that emphasize maintainability, modularity, security, usability, and long-term sustainability rather than simply adding new features.

Course Outcomes

The enhancement met the objectives identified in my Module One enhancement plan and demonstrated meaningful progress toward several Computer Science program outcomes.

Most notably, this enhancement demonstrates the ability to use well-founded software engineering techniques to implement computing solutions that provide practical value. The project applies industry practices such as layered application architecture, reusable middleware, centralized error handling, environment-based configuration management, modular component design, and RESTful API development. These improvements resulted in a cleaner and more maintainable application while preserving its existing functionality.

The enhancements also demonstrate progress toward developing a security mindset. Authentication was strengthened through improved JWT handling, guest route protection, duplicate account validation, and the removal of sensitive configuration values from the application source code. These changes reduce potential security risks while following common software development practices.

Finally, the project demonstrates progress toward professional communication by documenting the enhancement process through a comprehensive README and maintaining a structured Git commit history that clearly reflects the implementation process. Producing clear

documentation is an important aspect of professional software engineering because it allows other developers to understand, maintain, and extend an application over time.

At this point, I do not anticipate making significant changes to my planned outcome coverage. The remaining milestones will focus on demonstrating additional competencies through algorithm, data structure, and database enhancements while continuing to strengthen the overall ePortfolio.

Reflection

Enhancing this artifact reinforced the idea that professional software engineering extends well beyond writing functional code. While the original application met its functional requirements, revisiting the project from the perspective of maintainability, security, and software quality highlighted numerous opportunities for improvement. Many of the most valuable enhancements involved refactoring existing code rather than creating entirely new functionality.

One of the most important lessons learned during this process was the value of designing software with reuse and maintainability in mind. Creating reusable middleware for validation and centralized error handling significantly reduced duplicated code while making the backend easier to extend and maintain. Likewise, separating authentication responsibilities and consolidating both user interfaces to consume the same REST API reduced architectural complexity and improved consistency across the application.

Another valuable lesson involved balancing improvements with project scope. Throughout the enhancement process, I evaluated several possible features, including role-based authorization and additional administrative capabilities. Although these ideas could improve the

application, I chose to remain aligned with the original enhancement plan and focus on software engineering improvements rather than introducing unnecessary complexity. This reinforced the importance of managing project scope while ensuring that planned objectives are completed thoroughly.

The enhancement process also emphasized the importance of testing throughout development. As new functionality was introduced, I repeatedly tested authentication workflows, protected routes, CRUD operations, responsive layouts, API validation, and customer-facing functionality to ensure that architectural improvements did not introduce regressions elsewhere in the application. This iterative approach mirrors professional software development practices, where continuous testing is essential for maintaining application quality.

Overall, this enhancement transformed the original course artifact into a more maintainable, secure, and professional application. More importantly, it demonstrated how thoughtful software engineering practices can substantially improve an existing solution without fundamentally changing its core functionality. I believe this artifact effectively represents my growth throughout the Computer Science program and serves as a strong example of my abilities in software design and engineering.