**3-2 Milestone Two: Enhancement One: Software Design and Engineering**

Caleb Leavell

caleb.leavell@snhu.edu

Southern New Hampshire University

**Milestone Two Narrative: Enhancement One – Software Design and Engineering**

**Artifact Description**

The artifact I selected for this enhancement is a Java-based application titled **Appointment Service**, originally developed in a prior course as a simple object-oriented program. The application allowed the user to add and manage appointments, storing data in memory for the session. It provided basic functionality such as creating and retrieving appointments and demonstrated foundational object-oriented programming (OOP) skills like encapsulation and modular class design. The project was created to reinforce basic design and development practices in Java.

**Justification for Inclusion in the ePortfolio**

I chose this artifact because it serves as a strong baseline for demonstrating growth in software design and engineering. While the original artifact met the course requirements, it lacked features that would make it viable in a real-world scenario, such as persistent storage and a user-friendly interface. By enhancing this artifact, I had the opportunity to demonstrate industry-relevant skills and align my work with the outcomes of the Computer Science program.

Key enhancements I made include:

* **Persistent Data Storage**: Integrated file-based I/O to allow appointments to persist between sessions, simulating how real-world applications save state.
* **Search and Filter Capability**: Implemented logic for searching appointments by date, keyword, or description, reinforcing algorithmic thinking.
* **Update Functionality with Validation**: Added an update feature with strong input validation, improving data integrity and user experience.
* **JavaFX GUI**: Designed a basic graphical user interface using JavaFX to allow users to create, view, delete, and update appointments.

These improvements allowed me to showcase technical skills in file handling, validation logic, and user-centered software design. They also strengthened the software’s usability and real-world applicability, elevating a basic Java application into a more complete and professional solution.

**Alignment with Course Outcomes**

This enhancement strongly aligns with the following Computer Science program outcomes:

* **Outcome 3**: *Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.*
  + I demonstrated this by implementing searching and filtering logic, weighing the trade-offs between different data structures (lists vs. maps), and managing performance as the dataset grew.
* **Outcome 4**: *Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.*
  + I used JavaFX to create a GUI, structured the application using the DAO pattern to separate logic from data access, and added persistent file storage to simulate real-world needs.

The planned enhancements from Module One were fully implemented, and I did not need to change the outcome-coverage plan. In fact, the implementation reinforced the scope and relevance of those original goals.

**Reflection on Enhancement Process**

Enhancing this artifact gave me practical experience in designing software components that interact smoothly while remaining modular and maintainable. The most challenging aspect was integrating the GUI with backend logic in a clean, decoupled way. JavaFX had a learning curve, especially in handling event listeners and updating UI components dynamically in response to changes in the underlying data.

Another challenge was ensuring input validation remained intuitive without cluttering the code. I overcame this by creating utility methods for validation and error handling, which kept the logic clean and reusable.

Through this process, I deepened my understanding of the **Model-View-Controller (MVC)** pattern and the importance of separating concerns in a growing codebase. It also gave me experience with serialization and file I/O operations, which are crucial for software that needs to retain user data across sessions.

Overall, this enhancement helped me bridge the gap between academic projects and professional software development. It reinforced my ability to create scalable, user-friendly, and persistent applications that can be extended in future iterations—either by incorporating a database like SQLite or expanding the GUI with more advanced features.