Anthony Nguyen

CS 499

Milestone 2

Enhancement One

Software Design/Engineering

**A. Briefly describe the artifact. What is it? When was it created?**

The artifact I chose includes three Java classes: **Contact.java**, **ContactService.java**, and **ContactServiceTest.java**. Submitted by: CS 320 course (Software Testing, Automation, and Quality Assurance). The project description is to develop a contact management system with all CRUD operations (create, read, update and delete). Contact.java, A java class that defines a contact object, with fields like the contact ID, First name, Last name, Phone Number and Address. The **ContactService.java** Service layer manages creation, update and deletion of contacts java class. Finally, **ContactServiceTest.java** Service, a method of having unit tests on java to check the correct behavior of the service methods.

These artifacts were originally created during the academic year, where they served as a foundational exercise in object-oriented programming, encapsulation, and the use of service layers to manage business logic. The project demonstrated skills in software development, particularly in Java, unit testing, and service layer architecture.

**B. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

I selected this artifact for inclusion in my ePortfolio because it showcases my proficiency in **Software Testing, Automation, and Quality Assurance** and my ability to create structured, maintainable code following best practices in software engineering. The **Contact.java** class demonstrates my knowledge of encapsulation, constructor usage, and getter/setter methods. The **ContactService.java** class highlights my ability to implement business logic in a service layer, which is an essential skill in software engineering. The accompanying **ContactServiceTest.java** file further showcases my understanding of unit testing and test-driven development (TDD), ensuring that my code is reliable, and functions as expected.

In terms of enhancements, I improved the **ContactService.java** class by refactoring the methods to follow the **Single Responsibility Principle (SRP)**, ensuring that each method does only one thing. I also implemented additional validation to ensure that contact information, such as phone numbers and addresses, adheres to specific formats. Moreover, I enhanced the **ContactServiceTest.java** file by adding more comprehensive test cases to cover edge cases that were previously missing.

**C. Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

Yes, the enhancements I made to this artifact align with the course outcomes I had planned to meet in Module One. I focused on improving the code structure and readability by applying **SOLID design principles**, specifically the **Single Responsibility Principle (SRP)** and **Open-Closed Principle (OCP)**. These improvements allowed me to demonstrate my ability to design and implement software solutions that are maintainable and scalable.

In terms of updates, I added additional test cases in **ContactServiceTest.java** to increase code coverage and ensure that the enhancements function correctly under various conditions. This enhancement further supports my goal of demonstrating proficiency in **software testing** and **quality assurance**.

**D. Reflect on the process of enhancing and/or modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

Through the process of enhancing this artifact, I gained a deeper understanding of the importance of clean code and how following design principles such as **SRP** and **OCP** can significantly improve the maintainability and flexibility of a system. I learned how to better structure code to allow for future changes without introducing errors or requiring significant refactoring.

Some of the problems that I encountered working with JUnit test cases, it was hard to be sure that my code could be tested in the way which I expected. Originally, I had made use of a virtual desktop setup to do this assignment, then finally moving towards more user-friendly tools like Visual Studio. This is a great opportunity for me to re-visit the artifact and overcome the challenges that I faced then, further enhance my skills, and apply some advanced software engineering principles.

Additionally, I encountered difficulties when expanding the test coverage to include edge cases, as this required a thorough understanding of potential invalid inputs and exceptional scenarios that were not immediately obvious in the initial design.

By incorporating this enhanced artifact into my ePortfolio, I am demonstrating my proficiency in **software design and engineering** and my ability to produce professional-quality code that adheres to industry standards and best practices.