**Anthony Nguyen**

**CS 499  
Milestone 2  
Enhancement One  
Software Design/Engineering**

**Artifact Overview**

"This artifact is a Java-based contact management system that includes:

* Contact.java: A class that defines a contact object with fields like contact ID, first name, last name, phone number, and address.
* ContactService.java: This class manages the CRUD operations—Create, Read, Update, and Delete—for contacts.
* ContactServiceTest.java: A set of unit tests that verify the correct behavior of methods in the ContactService class.

These classes were originally part of an academic exercise in software engineering, focusing on object-oriented programming, unit testing, and quality assurance."

**Justification for Inclusion**

"I chose this artifact for my ePortfolio because it highlights key aspects of my software development skill set, particularly in testing and quality assurance.

* Contact.java demonstrates my understanding of object-oriented design principles, such as encapsulation and the use of constructors, getters, and setters.
* ContactService.java showcases my ability to implement business logic in the service layer, which is critical for building scalable and maintainable systems.
* ContactServiceTest.java reflects my proficiency in unit testing and the principles of test-driven development (TDD), ensuring that my code is reliable and functions as expected."

**Skills Demonstrated**

"This project allowed me to develop and demonstrate several important skills:

* Encapsulation and Object-Oriented Design: In Contact.java, I effectively used constructors, getter, and setter methods to ensure data encapsulation and proper object creation.
* Business Logic Implementation: ContactService.java focuses on applying business rules and ensuring that operations like updating or deleting contacts work correctly within the service layer.
* Unit Testing and TDD: In ContactServiceTest.java, I implemented unit tests to verify the correctness of the CRUD operations. This shows my grasp of software quality assurance practices and my ability to write maintainable test code.

Through this project, I also enhanced the code by refactoring it to follow the Single Responsibility Principle (SRP), which ensures that each method performs only one function. Additionally, I added validation to fields like phone numbers and addresses to ensure they follow specific formats."

**Process Reflection**

"During the enhancement process, I learned the importance of clean code and how SOLID principles like SRP and the Open-Closed Principle (OCP) can impact the maintainability and scalability of a system. Refactoring the methods in ContactService.java to adhere to SRP made the code easier to read, maintain, and extend.

One challenge I faced was with JUnit test cases. Initially, the tests did not fully cover all edge cases, which required me to expand the test coverage in ContactServiceTest.java. This made the system more robust by ensuring that invalid inputs, like malformed phone numbers, were properly handled.

Another difficulty involved my initial setup. I was using a virtual desktop to write and run the code, which proved inefficient. Eventually, I switched to Visual Studio, which provided a smoother workflow and allowed me to more effectively develop and test the artifact."

**Outcome Alignment**

"This artifact aligns with several course outcomes:

* Software Design and Architecture: By refactoring the code to follow SOLID design principles, I demonstrated my understanding of designing maintainable and scalable systems.
* Software Testing and Quality Assurance: I expanded test coverage in ContactServiceTest.java, incorporating edge cases to ensure the system could handle unexpected inputs.
* Encapsulation and Object-Oriented Programming: The Contact.java class illustrates my ability to design objects that encapsulate data properly, an essential principle in software engineering.

Through these enhancements, I showcased my ability to produce clean, maintainable, and testable code."

**Reflection on Enhancements**

"Enhancing this artifact taught me the importance of clean code and the value of design principles in making software easier to maintain and extend. By applying SRP and OCP, I was able to improve the system’s structure, ensuring that it could be easily modified in the future without breaking existing functionality.

Expanding the unit tests also emphasized the significance of covering edge cases, which is crucial in delivering high-quality, error-free software."

**Conclusion**

"In conclusion, this contact management system demonstrates my proficiency in Java programming, unit testing, and software quality assurance. By incorporating software engineering best practices and improving the test coverage, I ensured that the system was both maintainable and reliable.

Thank you for your time, and I would be happy to answer any questions you may have."