

Name: Steven Valdivieso Lemus

Class: CS-499-13167-M01

Date: April 21<sup>st</sup>, 2025

## Narrative: Enhancement One

The `ContactService.java` file is a Java class that provides basic functionality for managing a list of contacts, including operations to add, update, delete, and retrieve contact information. It was originally created in an earlier course in my Computer Science program and later enhanced during the Spring 2025 term as part of CS499. This artifact demonstrates essential object-oriented programming principles, including encapsulation, modularity, and method abstraction.

I selected this artifact for my ePortfolio because it reflects my growth in software design and engineering. While the original version of the class met functional requirements, it lacked robustness and proper input validation. Enhancing the artifact allowed me to demonstrate the application of core design principles such as single responsibility and clean code practices. By introducing helper methods like `isValidName`, `isValidPhoneNumber`, and `validateContact`, I was able to modularize the logic, reduce duplication, and improve maintainability. These changes showcase my ability to refactor existing code to meet higher quality standards, making the application more reliable and easier to extend in the future.

Through the enhancement process, I learned how critical thoughtful design decisions are when writing scalable and maintainable software. I focused on organizing the code more effectively, improving validation, and ensuring better fault tolerance through clear and concise error handling. I received feedback during the enhancement phase, and it confirmed that my design decisions aligned well with the expectations of the course. No significant changes were necessary, which reinforced my confidence in my approach to software design.

The artifact is significantly improved in terms of structure, readability, and resilience to invalid input. It now follows a more modular design, supports better error handling, and enforces constraints on the data being stored. This enhancement fully meets the Software Design and Engineering outcome of the course by demonstrating my ability to produce clean, modular, and logically sound code. Other outcomes such as Databases and Automation were not addressed in this particular enhancement, as the focus was strictly on improving the design and structure of the core application logic.