**Part 2: Narrative**

**Narrative: Algorithms & Data Structure Enhancement**  
**Name:** Mariana Zornes  
**Artifact:** Inventory Management Android Application

**1. Description of the Artifact**  
The artifact is an Android inventory management application built with Kotlin. It was originally created as a capstone project for a previous course. The app allows users to manage inventory items with full CRUD functionality and includes features like low-stock SMS alerts.

**2. Justification for Inclusion and Showcased Skills**  
I selected this artifact for my ePortfolio because it provides an excellent canvas to demonstrate practical algorithm and data structure implementation. The specific enhancements showcase several key skills:

The implementation of DiffUtil demonstrates advanced understanding of efficient data structure manipulation. Instead of refreshing the entire RecyclerView when any data changes (an O(n) operation), DiffUtil calculates the minimal set of changes (inserts, updates, removals) and performs only those operations (approaching O(n) but with much better performance in practice). This is a industry-standard optimization that shows I understand how to work with large datasets efficiently.

The **comprehensive input validation** algorithms demonstrate logical problem-solving skills. I implemented checks for empty strings, non-numeric values, and negative numbers, providing appropriate user feedback for each error case. This shows attention to detail and understanding of robust data handling.

The **user feedback system** using Toast messages demonstrates an understanding of user experience principles alongside technical implementation. These enhancements collectively show I can "Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution."

**3. Meeting Course Outcomes**  
This enhancement successfully addresses the planned outcome: *"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."*

The DiffUtil implementation represents a optimal design choice for managing the trade-off between computational complexity and user experience. The input validation algorithms demonstrate proper computer science practices for data integrity. My outcome coverage plan remains on track, with these enhancements providing strong evidence of algorithmic thinking.

**4. Reflection on the Process**  
Enhancing this artifact was particularly valuable for understanding real-world algorithm implementation. The main challenge was understanding the DiffUtil pattern and adapting the existing code to use ListAdapter instead of a simple RecyclerView.Adapter. This required rethinking how data updates were handled throughout the activity.

I learned that even seemingly simple user interactions (like updating a single item) can have significant performance implications when scaled. The process of implementing proper validation also reinforced the importance of defensive programming - anticipating every possible way users might interact with the interface and ensuring the system responds gracefully.

The most valuable lesson was understanding the practical application of algorithm efficiency. While DiffUtil adds some complexity to the codebase, the performance benefits for the user experience make it an essential tool for any professional Android developer. This experience has strengthened my ability to evaluate design trade-offs and implement optimal solutions.