CS 499

Professor Troy Hawk

Michael Thomas

1 June 2025

Milestone Three Narrative

1. Briefly describe the artifact. What is it? When was it created?
   * The Contact Service artifact is a contact management application originally created for *CS 320: Software Testing, Automation, and Quality Assurance*. It was first developed in Java in the summer of 2024 and allowed users to create, store, and delete contact records using a basic ArrayList. However, the original version had inefficiencies in search and duplicate prevention logic due to the linear structure. For my ePortfolio enhancement, I refactored the application to use a HashMap (dictionary) structure, implemented helper methods for internal access, improved readability and speed, and ensured logic was encapsulated and efficient.
2. 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?
   * This artifact highlights my capabilities in applying algorithmic principles and data structure optimization. The original implementation suffered from repetitive logic and inefficient linear searches. I selected this artifact specifically to demonstrate my understanding of how strategic data structure choices (in this case, replacing a list with a map) can vastly improve performance and code clarity. The refactored code introduces a getContactById() method, streamlines contact addition by checking map keys for duplicates, and simplifies update/delete logic using efficient constant-time lookups. These changes showcase practical improvements rooted in both design pattern recognition and real-world problem-solving approaches.
3. 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?
   * I planned to address course outcomes 3 and 4, and I met those objectives through this enhancement. For outcome 3, the replacement of a linear list with a HashMap aligns with algorithmic best practices for improving runtime efficiency and optimizing resource access patterns. The contact ID, now acting as a unique key, ensures quick lookups and duplicates are avoided without iterating over an entire list. Outcome 4 was addressed through my refactoring and modularization of logic, the introduction of helper functions, and the preservation of data validation and testability. These practices reflect the software engineering standards used in professional development environments. No changes are needed to my original outcome-coverage plan.
4. Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?
   * This enhancement was a great opportunity to apply real software engineering techniques to a previously rigid and inefficient program. One of the biggest takeaways was seeing the tangible impact that thoughtful structure and abstraction have on maintainability and performance. I learned how beneficial internal helper methods can be in reducing redundant code and improving the overall flow of logic. The most notable challenge was ensuring the contact update logic remained clean and used shared code paths to avoid inconsistency. I also had to refactor the test suite to confirm the program's behavior after switching from list-based to map-based logic. Overall, the experience reinforced the importance of scalability, readability, and maintainability in all aspects of development.