CS 499

Professor Gene Bryant

Michael Thomas

25 May 2025

Milestone Two Narrative

1. Briefly describe the artifact. What is it? When was it created?
   * The Appointment Service artifact is a task-tracking application originally developed during my coursework in *CS 320: Software Testing, Automation, and Quality Assurance*. While it was initially written in Java in the summer of 2024, the program allowed users to create, store, and delete appointment records with a basic list. It featured minimal error handling, rudimentary testing, and lacked scalability. As part of my ePortfolio enhancement, I refactored and rebuilt the application in Python, introducing exception handling, object-oriented best practices, and a comprehensive unit testing suite with Pytest.
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 demonstrates my proficiency in software engineering and design, especially the application of real-world practices like Object-Oriented Programming (OOP) – through maintainable classes (Appointment and AppointmentService), Test-Driven Development (TDD) – using Pytest for method-level testing, exception handling – for data validation and application robustness, and thorough commenting – helping future developers understand the code. This change demonstrates my ability to evaluate the weaknesses of an existing system and implement modern development standards to transform it into a more scalable, professional, and secure application.
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?
   * The course outcomes I projected to meet included outcomes 2, 3, and 4. Within outcome 2, I applied clear, professional communication through clean code organization and consistent commenting. The included Pytest suite serves as both validation and documentation of expected behavior, enhancing the overall understanding for technical reviewers. Outcome 3 is seen within my application of algorithmic and software testing principles, ensuring performance is correct. Functions like **update\_appointment** handles logic and constraints in a way that mirrors real-world systems. Implementing robust test coverage enforces confidence in the behavior under different edge cases. Finally, outcome 4 is represented in the refactor by demonstrations of innovative software engineering practices. Previously a tightly coupled Java implementation, the software has been converted to a Pythonic architecture, embracing best practices in code quality, reusability, and tooling (Pytest). No changes are needed so far to my outcome-coverage plan, as all my targeted goals were ultimately realized.
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?
   * Iterative development and careful planning were instrumental towards enhancing this artifact. Transitioning from Java to Python involved thoughtful translation of object-oriented principles (OOP) and validation logic. In addition, with the new update\_appointment feature, I was pushed to design extensible methods that maintained backward compatibility with original system behavior. Some of the challenges I faced included conversion and validation of datetime inputs without user input functions, ensuring proper handling of duplicate appointment IDs, and the use of Pytest unit tests that covered both valid and exceptional cases without the use of external tools or frameworks.