CS 499 Milestone Two Narrative

1. Briefly describe the artifact. What is it? When was it created?

The artifact for category one is a jukebox playlist application created in my CS-310 Collaboration and Team Project course taken from October 2024 to December 2024. Its purpose is to create a playlist of songs from different musical artists. In the class, The studentPlaylist() method creates a LinkedList of PlayableSong objects and populates it with a few songs from each band. The jukebox playlist application was a collaborative project using the BitBucket repository.

1. 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 is suitable for my ePortfolio because it demonstrates several key software development skills such as code refactoring and design, defensive programming, error handling, and adherence to coding standards. The transformation of the original code into a more structured, maintainable, and robust version showcases my ability to improve existing code. The introduction of the addSongsFromBand() helper method to remove repeated code highlights design skills. The addition of boundary checks and null/empty list handling demonstrates a proactive approach to preventing errors and writing robust code. The implementation of output messages for handling null or empty song lists demonstrates attention to providing feedback and improving the user experience. Lastly, consistent formatting and naming conventions demonstrate an understanding of and commitment to coding best practices.

Overall the artifact was improved through:

* Elimination of Redundant Code: The addSongsFromBand() method removed repeated code blocks.
* Prevention of Exceptions: Boundary checks and null/empty list handling prevent IndexOutOfBoundsException and NullPointerException.
* Improved Code Structure and Readability: Consistent formatting, naming, and the helper method enhance the overall code structure and readability.
* Enhanced Error Handling: The output messages provide feedback in case of errors.

1. 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?

Yes, I met the course outcomes I had planned to meet with this enhancement. The enhancements directly address course outcomes related to:

* Course Outcome #3: 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 refactoring and design improvements directly address this outcome.
* Course Outcome #4: Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.
  + The use of established coding practices and defensive programming techniques demonstrates this outcome.

1. 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?

From creating and improving the artifact in category one I learned how important code reviews can be. Before going through the code review checklist, I could not think of any improvements for my code. It was a challenge for me. The code review checklist showed me what to look for. This led me to understand how I would refactor my artifact. Refactoring the code to remove redundancy and improve structure came to me after looking at the code through a different lens. I also learned the value of defensive programming. Implementing boundary checks and null/empty list handling reinforced the importance of writing robust code that can handle unexpected input. Overall, the enhancement process for this artifact was a valuable learning experience that reinforced good coding practices and provided a practical application of software engineering principles.