**Conclusion**

Our assigned project involved the creation of a library system designed to enable library members to borrow books from that specific library. This project provided us with an opportunity to apply our computational skills while also demonstrating our critical thinking and reflective abilities, as evidenced in our work. During the project, I realized that integrating the code into a single file would be quite demanding and would impede my progress. To address this issue, I decided to divide the code into multiple files, allowing for better organization and a clearer overview of the entire project. I took this step because dealing with numerous code errors within a single file became a significant challenge. For instance, incorporating both the library system and the functionality to assign books to specific members presented considerable difficulties. As aforementioned, I split the files, this now allowed me to continue to develop my system/code with more ease, overcoming this challenge of multiple errors and misplacement in my project prior to dividing the files.

Another challenge I encountered was grasping the UML diagram, which took a considerable amount of time for me to fully understand how to apply to my code. This led to a time-consuming coding process as I meticulously followed the UML steps. However, as time passed, I began to see the value of the UML diagram in helping me understand the client's requirements, allowing me to plan the project more effectively to meet both the client's needs and the project's objectives. This systematic approach, step by step, resulted in a more robust and functional code. For instance, separating the classes from the main program file was a significant improvement. As mentioned earlier, I divided the code into multiple files as a solution to the problem of code errors that arose in the previous single-file structure. Consequently, I opted to separate the code into distinct files, with each file housing different classes, including "Library.h" and "Date.h," in addition to the main program file. The UML diagram played a crucial role in helping me identify my shortcomings and rectify them. Nevertheless, I still encountered difficulties when implementing the overdue book fine system.

**How would I change the approach of a similar task?**

A change I would make is to avoid hard coding the assignment. I believe that doing so would make the coding process more time efficient. For instance, having a concise code with no more than two files would be easier to understand for both myself and others reviewing it. Achieving this can be accomplished by focusing on the code's structure, including functions, lists, arrays, and so on. In my opinion, I would have preferred to create a more advanced system, one that challenges me to improve and explore multiple approaches for similar tasks. This could involve implementing security features for members and pushing the boundaries of my coding skills.