I spent around 10-15 hours physically writing code, 3-4 hours debugging and testing my code, and another hour cleaning up commenting and formatting for around 20-25 hours spent writing the actual project. Prior to beginning coding at any point, I spent around 3 hours on the planning process in addition to the time spent on homework 2, to make sure I understood exactly what I wanted out of the system. I technically began work on it when we started homework 2, but I began additionally planning about 3 or 4 days after homework 2 was due and began coding the following week. I lost my initial data at some point due to a bad save, and then had to recover it over the past two weeks and redo the entire project.

The project itself wasn’t terribly complex to code, but it required a lot of code because building a new system is always some intensive setup. I also wanted to build additional menus, confirmations, and safeguards, which was what took me so long. I ended up with around 1800 lines of code by the end of the project. The difficulty was mainly in both time management and the physical scale.

One thing I failed to understand that severely hampered my time management was that I can try to build my code at any time to help check for typos and compiler errors. The first build of my project was attempted when I had effectively finished it the first time. After writing nearly 1800 lines of code, I attempted my very first compile. This meant I spent probably an extra 2 hours just trying to figure out what my compiler errors were instead of making useful progress. This made my time management far poorer than it could’ve been. I need to solve problems in smaller steps and take more time away from my code to really understand what I’m doing with it. Starting earlier would’ve helped me immensely.

I plan to work on the next project earlier and build often instead of after writing all of my code. The most difficult part of this assignment ended up not being what I expected. Instead of file I/O, which I had to learn from scratch, I struggled more with correcting for potential user input errors than I realized I would. Currently, this is crudely solved with cin.fail() wrappers on a lot of code to safely exit the program and save the data instead of outright crashing, but this is not an ideal way to handle it. I feel like for what I have learned so far, my project is pretty good. There are definitely things I wish I knew how to do differently, but I’m fairly proud of my work. I take notes of what I want to do differently in each project and usually reattempt them later. One thing I look forward to trying is having a proper database attached to the system, card catalog services, and a functional GUI. I may not be able to do it well yet, but the possible ways to build the system out seem very fun. I enjoyed this project quite a bit.