Review and evaluation of the design

When approaching the design of the program, we made a conscious effort to avoid high coupling between the different modules. This is because it would make it unnecessarily difficult to maintain the software as a deep understanding of the whole system would be required just to make a small change. Designing modules to avoid high coupling proved difficult during the design phase and we decided that some coupling is worth the cost as it would both make the program more readable and increase reliability and correctness as it makes it easier to test each module. Searching is done through the separation of data into categories of related topics and this made it easier and much faster to obtain relevant information. If we were to improve on this design, we would multithread the process, so still divide up the data set into separate categories but search through a category with multiple threads. We would obviously see an increase in speed for the searching portion of the program. Also, when searching for authors one might notice that the first search takes a relatively long time (around 5 seconds depending on your computer). This is because from the first search, each author is mapped to every research paper they are involved with so that the next time the user searches by author, it would be done in constant time. We believe that this is the best approach rather than have the program take ~5 seconds every time the user wants to search by author, but it comes at a cost in terms of space. We believe that the trade off is worth it since users would likely want search hits faster than have a program that uses less space. To improve on this aspect, we would also multithread the first search of mapping each author to their respective research papers. We tried our best to increase separation of concerns by making sure every module does a distinct task. This decreases generality since the modules we created have methods that perform specific tasks but it makes it easier to maintain as testing the code is simpler.