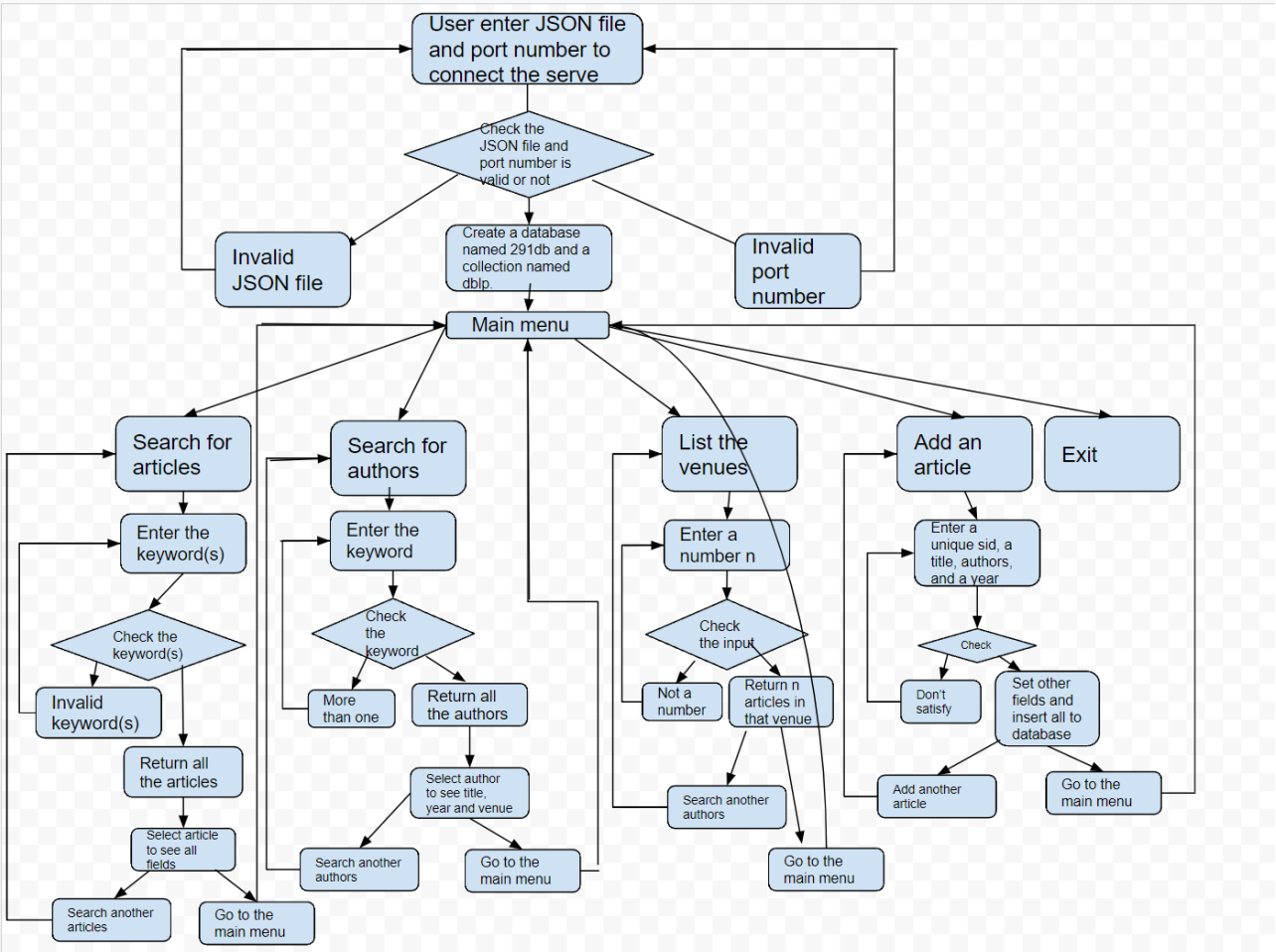
The General View

1. Users must enter the JSON file and the port number to connect with MongoDB.Then the program will create a database named 291db and a collection named dblp. (If the collection already existed, the program will drop it and create a new one). The program will also check whether the input of the JSON file and the port number is valid.
2. Users will go to the main menu page and the program asks users to enter 1(search for articles), 2(search for authors), 3(list the venues), 4(add an article), or 5(exit) for the next action. To search for articles, users need to provide one or more valid keywords to be matched in the database and the program will return all the articles that match the keyword(s). In addition, users can select the article to see all fields. To search for authors, the program will ask users to provide a keyword and then return all the authors whose name contains the keyword. The results will be sorted by year with more recent articles. Also, users can select the authors to check the title, year, and venue of all articles by that author. To list the venues, users could enter a number n to see a listing of the top n venues. The program will return the number of articles in that venue, and the number of articles that reference a paper for each venue. The results will be sorted by the number of papers that reference the venue with the top most cited venues. To add an article, users should provide a unique sid, a title, a list of authors, and a year. The program will set abstract and venue to null set the references to an empty array and set n_citation to zero. Finally, combine them and add the database.



Detailed Design

Phase 1 design:

1. User enter JSON file name
2. User enter port name
3. Construct server using MongoClient
4. Construct database
5. Create collection (using mongoimport)
6. Create text indexes and other indexes for phase 2

Phase 2 design:

1. Search for articles:
   Users enter some keywords using space to separate their keywords and then we use MongoDB text search to find the matching articles that match every keyword in either title, authors, abstract, venue or year fields. The results will not show all at a time but will show 5 at a time and ask the user to choose if they want to see more results. And once user select 'n' for 'no', the user could select an article and see the abstract and the authors in addition to the fields shown before. And the id, title, year of the article that reference this article will also be listed.

2. Search for authors:
   The user enters one keyword (no space allowed, has to be exactly one word) and system will finds the matching authors. The system automatically reads the total number of articles each matched author has participated in writing. The user can select a matched author by enter the name of the author and system will display all the articles he has participated in writing. The system sorts the articles by the data of published, and the articles with the early data will at the top.

3. List the venues:
   First, use $match to pick out the article with an empty venue, and then use look up to find the article's references containing this article. And we could get a list(the ids of articles that reference this article) for each article. We then use $unwind to seperate this list and so we could get multiple record for each article since the same article could be referenced by many articles. And we $group by value and $addToSet to get the number of the distinct number of article that references this venue. And then we sort by this distinct number and limit the number of venues(user input n)
   After we get the top n venue, we could use group to get the number of articles in that venue.

4. Add an article:
   Firstly, we have to get all the data which includes a unique sid, a title, a list of authors, and a year. To check whether the sid is unique, we create a query that uses “count” to check the number of occurrences of the input in the database. If the count number is greater than 0, it demonstrates the sid is not unique. Therefore, we will ask the user to check the sid and enter it again. For title and year, we use the “isdigit” function to make sure that a year is a number and use “len” function to confirm that the user will enter a valid title. In addition, we create a list named “author_list” to store the authors and the program runs a while loop to ask users if they want to continue to add authors (keep adding press 2, stop adding press 1); before the inputs are added to the list, the program will check if the input has already existed in the list or not. Secondly, we set abstract and venue to null set the references to an empty array, and set n_citation to zero. Then use the query “insert_one” to add the article information to the database. Finally, the program will move to a menu that asks users to enter 1 or 2 to decide whether to add another article or to return to the main menu.

Testing Strategy

Using small test first like dblp-ref-10.json and dblp-ref-1k.json and for listing venues, we crearte a sample test based on the class example we gave.
We test that we covered all the information required on eclass and then we test the accuracy. For example, when we check the results, we use “ctrk+f” to search the word and then see if the number are the same. And we also compare the results with other groups, like the listing venues. Since the results for listing venues should be the same if the input n is the same.
At last we test the time and update our function to make it more efficient and see if we could more mongodb query rather than process using python.