# CS 5500 – Managing Software Development - Homework 4

Author: Deepak Krishnan (krishnan.d@husky.neu.edu), NUID: 001681591

As a part of homework 4, I was assigned Team 21’s repository for testing. This report is a compilation of my experience with their product.

## Installation

The installation was straightforward since a executable JAR had already been compiled for testing purposes. There were no extra steps required to set up the database in my local machine since the team relied on an AWS instance to set up their backend.

From an end user perspective, the AWS backend was a welcome add on since it saved me a lot of time in the initial set up. It got me on my heels right on the word ‘go’. But as a tester, I would have liked access to their database in my local machine. This would have allowed me to run tests on the queries being generated by the application and also let me have an idea of the data available to the system.

## Code Review

As a traditional java programmer, I think the packages could have been named in a better way. Oracle recommends that the package names must start with the reverse internet domain name of a company and in our case, it would be edu.neu.\* or as a more relaxed version, com.neu.\*.

Certain classes are missing code formatting. This would create a readability issue for a new programmer who is trying to get to know the code. Another readability issue I noticed was the variable names in certain files. For example, instead of naming a variable as *list* (JournalMO.java), a better name would be *journalArticlesList*.

The methods are missing proper documentation. As per recommended standards, an ideal documentation for a method should describe its function and specify the arguments, return types and exceptions thrown by it. I think this is one place that requires some attention within the code.

The methods in certain classes have been left to their default scopes. I think this could be improved since no other class should ideally have access to the methods of a certain class unless absolutely intended by the programmer.

In its present form, the exceptions are not being logged by the program. It is being directly written to the console. I’d recommend for this to be updated such that the issues are logged into an independent text file for debugging purposes.

I found a few instances where the auto-generated TODO stubs were not removed after adding the method definitions (JournalMO.java). These need to be scrubbed off the final code. Also, there is a CommonUtils class which does not have any members. This must be removed to minimize unwanted clutter in the code.

I noticed that the database connection string was hardcoded inside the code. Ideally, I’d recommend that such things must be moved to a central properties file such that it could be easily edited when in need. One could also argue that specifying the username and password for the database in a properties file or in code is a security vulnerability.

Another improvement would be dependency maintenance. Being a traditional java project, it depends on the programmer for necessary libraries which may create issue down the line. For instance, different developers might end up using different versions of these libraries for their local development and the whole project might end up in chaos. So I would definitely recommend the team to use a build manager such as maven/gradle which will manage the dependencies and keep them consistent across workspaces.

## Testing & Debugging

From a testing perspective, I think the test cases could be made more exhaustive. At present, only certain case have been covered. I think the developers could add some error and exception scenarios in these tests and check if they are being handled properly by the application.

In my opinion, the application has a lot of scope to improve on the user experience side. Since the application is targeted towards program committee chairs to address, what I would call, a formal process of committee selection, I would have recommended a more professional look for the UI.

At present, the UI is adorned by a background image having a darker shade of blue and all other UI elements are being rendered over this image. Since the font colors are consistently black across the UI, there is a lack of contrast which may create readability problems for users with limited eyesight. As a designer, I would have gone for a high contrast interface such that there is proper distinction between the content and their wrapper UI controls.

Another prominent issue that I noticed was that the application failed to respond dynamically to changes in layout/screen resolution. Some of the elements kept overlapping each other and at certain instances the overlapping text failed the end user’s needs.

As a part of testing, I found the following issues with the application (Please note that the steps to reproduce these issues can be found in their respective GitHub URLs):

### Navigation issue inside the application

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/1>

*Issue:* As an end user, I found it difficult to navigate around the application. For instance, once I open the details of an author, there was no way for me to go back to the previous page. The only way I could do this was to click on the title which in turn took me to the application’s main page. Then I had to perform a search using the same keywords so that I could get the search results.

Another issue was that the team relies on the application title to provide the navigation back to the main page. This leaves the end users guessing on how to go back unless they take a wild guess to click on the title. For the first few times, I kept closing the application and re-opening it since I had no idea that the title was clickable and that it would take me back to the main page.

In my opinion, this was a huge usability issue. I think the navigation controls must be well defined and intuitive.

### Error message while searching for journals

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/2>

*Page:* Journal search

*Issue:* If I try searching for journals without entering any keywords, the application displays the following error message: "Searching for null".

This message is a bit confusing in my opinion. It could mean multiple things depending on the user's perspective. For example, it could mean that there are no journals in the database or it could also mean that the application doesn't allow empty fields.

This is a usability issue which might keep the end-users guessing.

### Search result inconsistency in journal search page

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/3>

*Page:* Journal Search

*Issue:* The search results seem to be inconsistent while searching with empty keywords inside journal search.

*Steps to reproduce:*

1. Open the application
2. Navigate to journal search
3. Click on the search button [Result: Error message saying that I am trying to search for null]
4. Now add a keyword. (I used 'acm')
5. Click on the search button [Result: This works as expected and shows me a list of ACM journals]
6. Now remove the keyword typed in the textbox.
7. Click the search button once again [Result: Now it shows a list of ACM journals]

If you notice, the results from Step 3 and Step 7 are inconsistent whereas a user would have expected them to be the same.

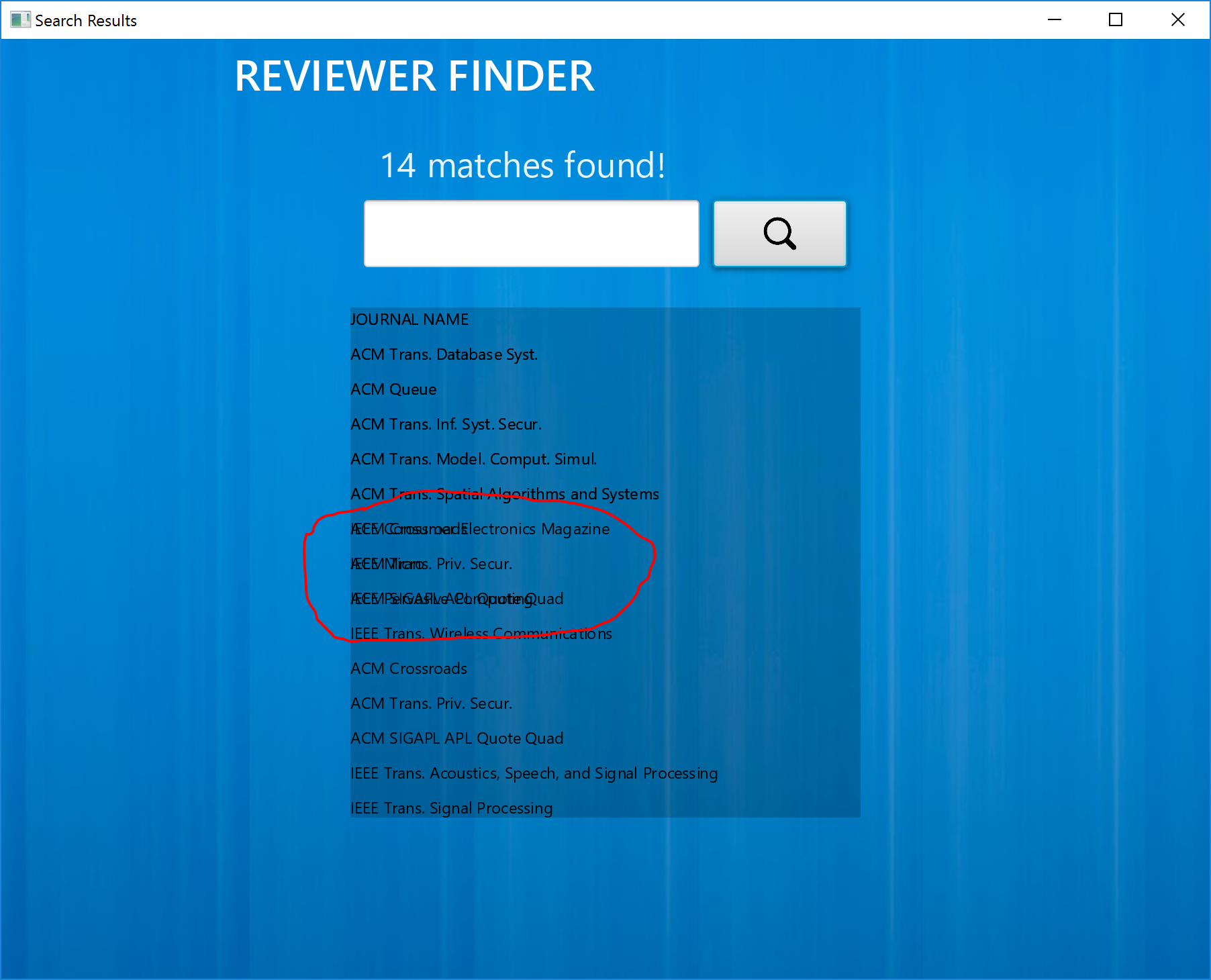
### UI rendering issue in journal page

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/4>

*Page:* Journal search

*Issue:* The search results seem to be rendering on top of each other in certain cases.

Please refer to the following image to see the rendering issue I am facing:

[](https://media.github.ccs.neu.edu/user/560/files/2dcadb0c-1ed2-11e7-98ea-d6a63801b6c3)

### Clickable links inside the application

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/5>

*Page:* Journal page

*Issue:*

This is a usability issue again. Since you have a lot of different flows inside your application, I think at present, it is hard for a user to comprehend which items are clickable and which are not.  
For instance, if I search for journals (say 'acm') and click on any item in the result list, it opens up a different page which shows the list of publications in that journal.   
But if you click on list of these publications inside a journal, nothing happens.

*Possible solution:* You could change the cursor type when the mouse pointer is over a clickable object.

### Unexpected results while filtering results by year

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/6>

*Page:* Journal page

*Issue:* If I filter the articles inside a journal by an author name, and then use filter by year, it displays articles by other authors as well.

### Search result inconsistency in conference search page

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/7>

*Page:* Conference search

*Issue:* The search results are inconsistent while searching with empty keywords.

### Longer cell values in tables are getting clipped

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/8>

*Pages:* I think this is an issue across all pages with tables

*Issue:* If the value in a table cell is long, it is getting clipped.

### Vertical scroll in tables

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/9>

*Page:* I think all pages with tables

*Issue:* None of the tables seem to be having vertical scrolls nor does it seem to have any sort of pagination.

As a user, I would have wanted to scroll down or switch pages to see more of the available authors from your database.

### Unimplemented functionality in Author search page

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/10>

*Page:* Author search

*Issue:* The filter options in the sidebar doesn't seem to be having any effect on the results shown in the page.

### Navigation between Author details page and Author search page

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/11>

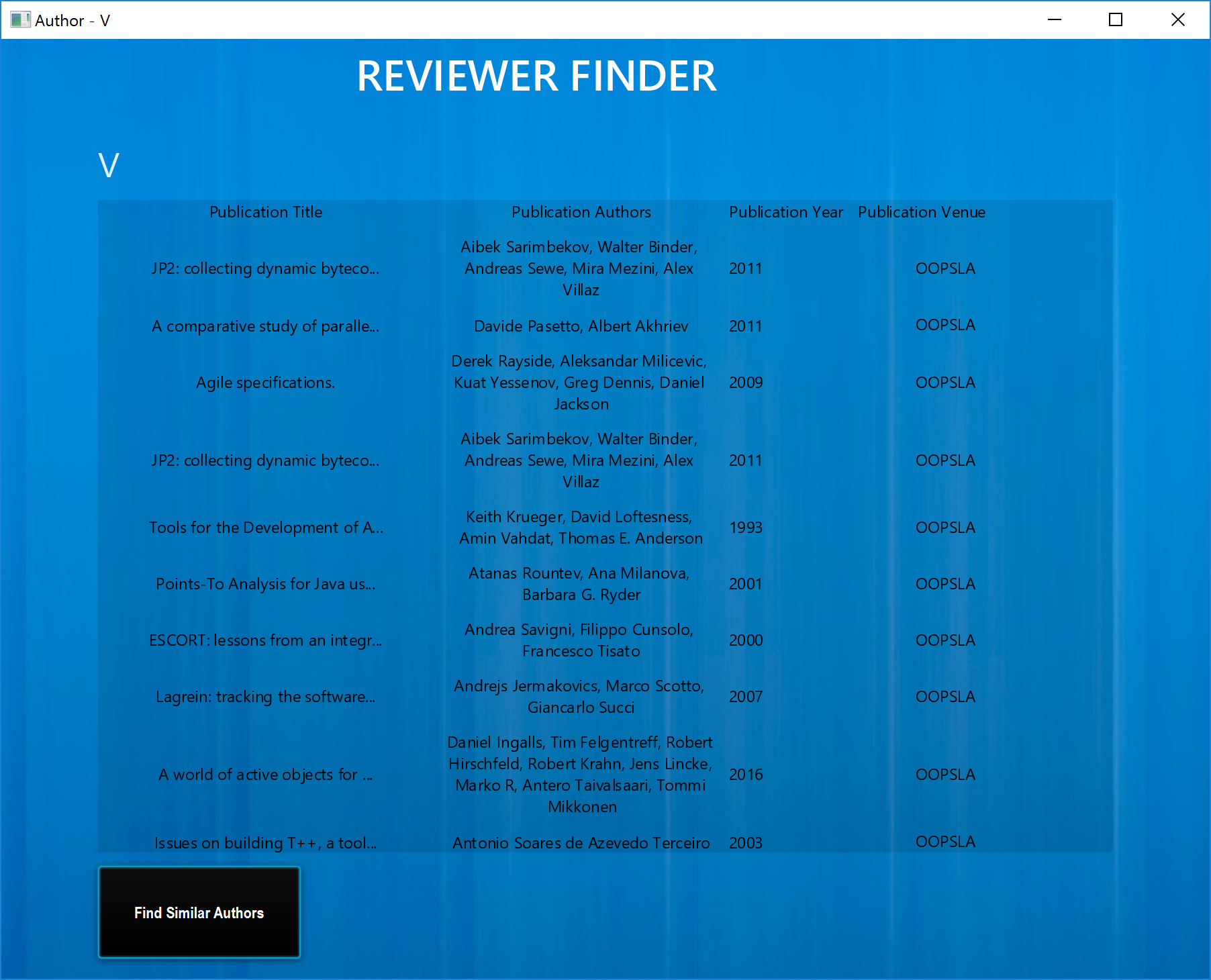
*Page:* Author details page

*Issue:* There is no option to go back to the author search page

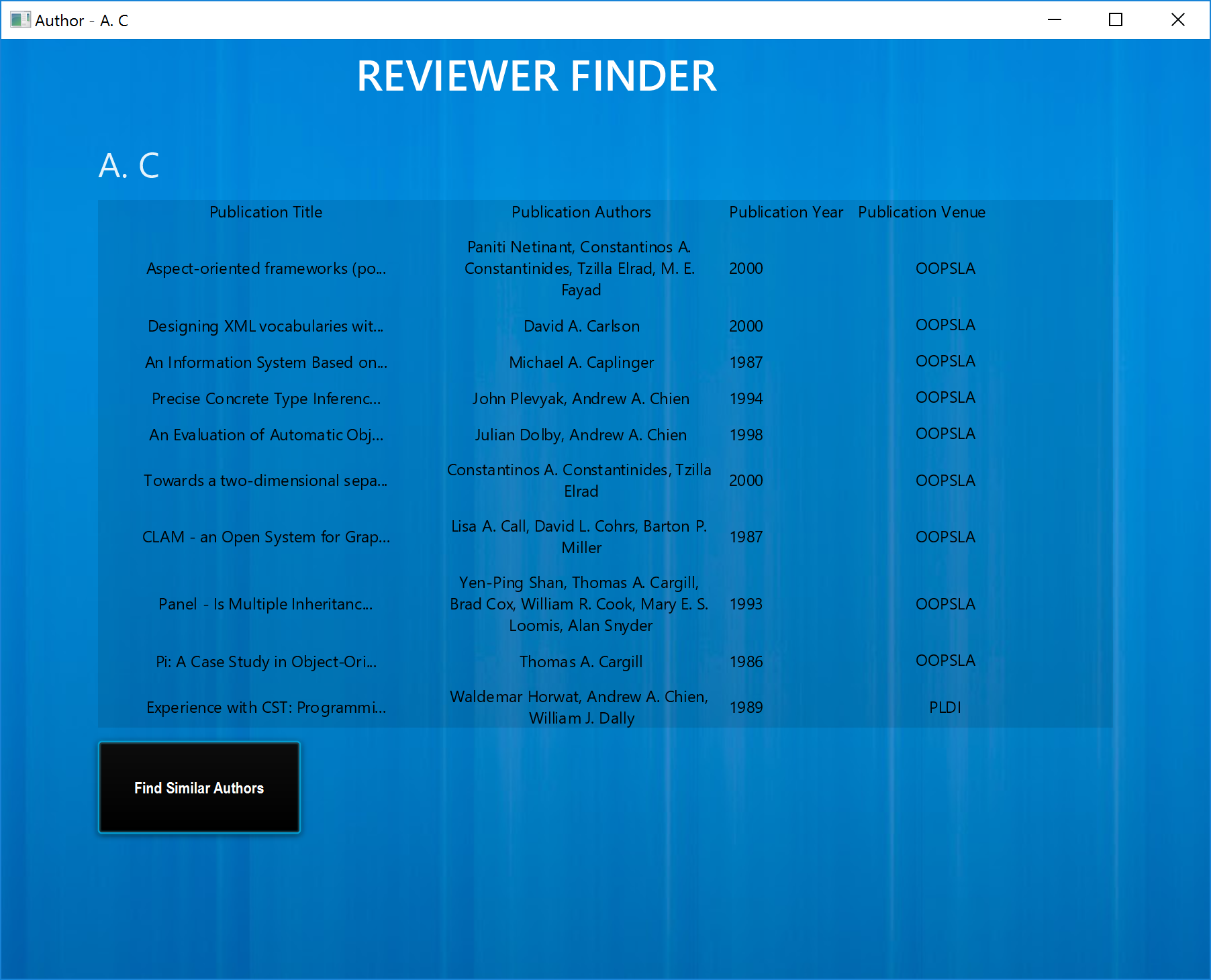
### Data Issue

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/12>

*Issue:* I found a few entries in Author page which whose names were just alphabets:

[](https://media.github.ccs.neu.edu/user/560/files/888ac164-1ee2-11e7-8f15-02b7011cef2f)

Just wanted to point this out in case you guys had missed it while cleansing the data. I think some of them are initials of the author like in the following case:

[](https://media.github.ccs.neu.edu/user/560/files/f270dd3e-1ee2-11e7-820d-42c94fa5cee1)

If you see the search results in above screenshot, you will notice that for 'A. C', the list contains articles written by multiple authors whose names contain these characters (Constantinos A. Constantinides, David A. Carlson, Michael A. Caplinger etc.) which are in fact different authors.

### Similar author search results getting clipped

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/13>

*Page:* Author details

*Issue:* If the similar author search retrieves a large list of authors, the height of the application window is clipping the results.

### Data input format / validation in filter text boxes

*GitHub link:* <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/14>

*Page:* Author page

*Issue:* The application doesn't show any validation error message if I enter invalid characters into textboxes.

I think an application must validate these inputs because a user could also argue that entering 11 for 2011 is valid since the application doesn't enforce any sort of input format for these fields.   
Similarly, the Committees filter doesn't have any explicit indication of what the expect input type is. A user would have to guess that it is expecting a year range as input.   
So I would say that this enhancement would be a good to have feature in this application.

### Overlap while rendering the results for keyword search

GitHub link: <https://github.ccs.neu.edu/CS5500-Spring2017/team21-test3/issues/15>

*Page:* Keyword search result page

*Issue:* The tables that display the authors and publications are overlapping each other.

### Inconsistent switching between screens

*Page:* Keyword search result page

*Issue:* In all other pages, when you click on any actionable object, the current screen used to close and the result used to open up in a new window. But in this page alone, when you click on an author name, the current page doesn't close.