




VERSION PHASE4

APRIL 17, 2017

MSD PROJECT REPORT

PRIYANKA HALESH
RUSHABH SHAH
SALONI SHAH
VIKAS JANARDHANAN



MSD PROJECT REPORT

OBJECTIVE

The objective of the project is to create a system to help find a set of suitable and available reviewers for judging conference or journal paper. The system leverages existing conference and journal repositories to propose appropriate candidates who can adequately review submissions to a Program Committee Chair or Editor-in-Chief.

PROJECT GOALS

The goal of this project is to design, implement, test, and evaluate an application that assists program committee chairs and associate editors with identifying suitable candidates for their committees. The system provides an interactive user-interface to execute queries against publication data from sources such as DBLP and Computer Science Rankings.

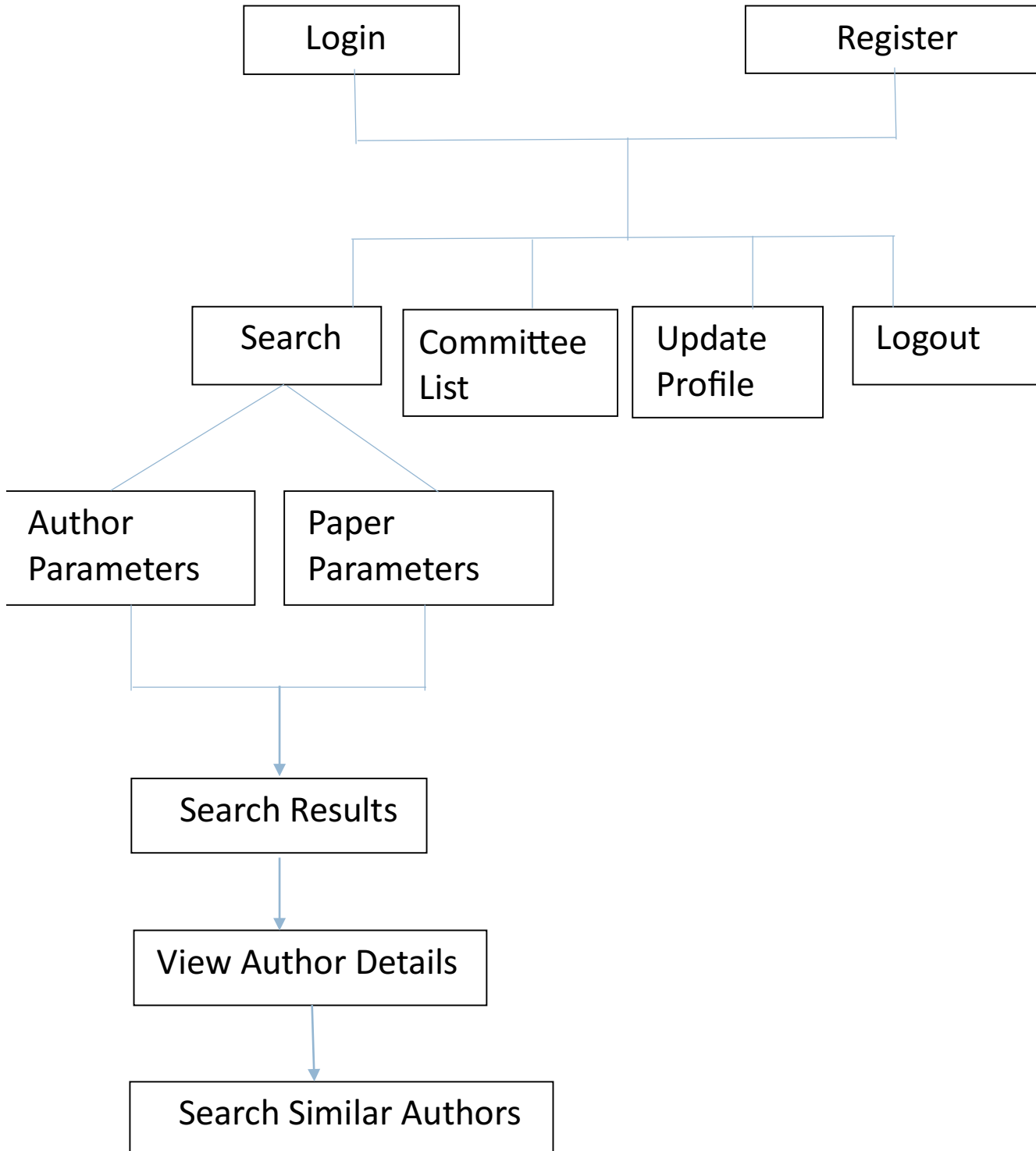
SYSTEM COMPONENTS

Front End: The front end is responsible for analyzing publication data. The system uses data from the DBLP publication site (<http://dblp.uni-trier.de/db/>) and Computer Science Rankings (<http://csrankings.org/>). The system relies on database uploaded on AWS.

Query Engine: The query engine is responsible for executing queries against the representation that the front end creates. It acts as a middle-tier to take input data from user interface and fetch appropriate result from the database. It is also responsible for displaying the result on the user interface.

User Interface: The user interface is responsible for providing end users facilities for entering and executing queries and then displaying the results. The end user can enter multiple parameters and can see the results displayed on the user interface. It provides as a powerful way of selecting authors based on parameters like name, school, thesis type, paper published and its timeframe, etc.

GENERAL FLOW OF SYSTEM



SYSTEM FUNCTIONALITY

Our system provides the below functionalities:

COMPOSITE SEARCH

The user is provided with various options to perform the desired search. Even though ours is a single composite search, to make the UI more clean and help user understand the different search parameters, we have labeled them as Author, Committee and Paper search. The results of a search will be list of author names matching the search criteria.

- **Author Parameters:** This provides the user to search based on information related to an author, such as author name, school name (where he/she published his/her thesis), type of thesis an author wrote (Masters or PhD), number of papers published (it can either be an exact or range search) and the university with which author is affiliated currently.
- **Committee Parameters:** Committee parameters provides the user to search for an author who might/might not have served in a committee before. The various parameters provided are whether or not the author has served in a committee before, the years (start and/or end range) when he/she was part of the committee and the conferences where he/she served as a committee member.
- **Paper Parameters:** This provides the user to search based on information related to a publication of an author, such as paper title, keywords, year the paper was published (start and/or end range), and conference and/or journals where the paper was published.

VIEW DETAILS OF AN AUTHOR

After a successful search, the user is given the option to view details of the author. This will provide the user with various publications of selected author. The details provided are publication name, type of paper (in-proceedings, journal or thesis), title of paper and the year it was published. Along with this, the user is provided with the information about whether or not the author has served in a committee before. If he/she has, they are given to view committee details which will show the user in which conference and which year had the author been on committee.

SEARCH SIMILAR AUTHORS

This functionality helps user find all those authors, who have similar profile to a selected author. We are providing the user to find similar authors based on:

- **Co-Authors:** Those authors who have co-authored a paper (in conference or journals) with the selected author
- **Based on Affiliated University:** Authors who are affiliated with the same university as selected author
- **Based on same Region:** Authors who are located in same region as selected author
- **Based on Research Area:** Authors who have same research area as selected author

If the selected author has no details of his/her region, affiliated university or research area, then respective options will be disabled.

FORM A COMMITTEE

This functionality provides the author with an option to shortlist a candidate (author) for the desired committee/panel formation. User can shortlist authors from 2 places, once after successful search and other after searching similar authors. Once the user shortlists the author, he/she can go to the Committee List where he/she can look at all the shortlisted authors, compare them by looking at their details (same details as in View Details of Author) or remove them from the list. This list is stored in the database.

CHANGES AFTER PHASE 3

ENHANCEMENTS

- In Phase 3, our DBLP dataset was limited to four paper conferences, two journals and four committee conferences. In Phase 4, we are providing the entire DBLP dataset. Along with this, we have extracted information about authors like affiliated university, region and research area from Computer Science Rankings, and incorporated into our database. This new data from Computer Science Rankings is used during searching of similar authors.
- **View Author Details:** We have added another functionality, where the user will be able to see all the publication and committee served details of a selected author. This will help the user when he/she wants to see those details and compare it to another author.
- **Search Results:** Previously, our search results used to show all the details (like paper published, conference/journal where it was published, etc.) of author inside the search results table only. In this phase, to reduce the redundancy in author names and make it cleaner, we are only displaying the author names in the search results.

ADDITIONAL FUNCTIONALITY

We have added 2 new functionalities in our system.

- **Compare Authors present in Shortlisted Committee:** Previously, after adding an author to shortlisted committee, to view that author's details, the user would have to search again for that particular author. So in this phase, we have provided the users to view Author Details (as explained in System Functionality) inside the Shortlisted Committee List itself, so that the user can compare authors.
- **Searching for Similar Authors:** Previously, our composite search itself was responsible for finding all the similar authors, but now, apart from that, when user selects an author, and views its details, he/she is provided with the option to search for Similar Authors (as explained above in System Functionality).

REFACTORING

- In Phase 3, we only considered subset of DBLP data which had specific conference records where the total record count was around 30k for all the publication records. For phase4, we are considering the entire DBLP dataset which has a total of around 3.5 million records. So to incorporate the same, we refactored the entire database design to denormalize the table structure so that queries come out faster.
- The Query Engine was also refactored to support the new database table structures and form a faster performing query.

- Both the Query Engine and Parser code was refactored to adhere to design patterns as explained below

DESIGN PATTERNS

- **Parser factory** : Parser code was refactored to use a parser factory that makes object of the respective parser object type required. In Phase 4, when we wrote parser for CSRanking region data and CSRanking affiliation data, the parser factory simply needed to write methods for creating objects of respective CSRanking parser. The newly created parser factory was used by the DBLPParser to create respective parser objects and depending upon the type of record currently being parsed, the respective parser object method was called to perform the processing of the record.
- **Query Builder Factory**: Query Builder component was also refactored with a query builder factory that makes the objects responsible for framing different parts of the query like the where clause, group by, query relevant to paper table filters etc. So in phase 4, when we implemented where clause for CSRankings university name, a new component to handle the where clause building for the same was added to the QueryBuilder factory.
- **SimilarSearch Adapter**: We also used adapter design pattern where the QueryBuilder createQuery interface was adapted to meet the needs of building query for similar search for coauthors, region and same research field. So SimilarSearchService acted as adapter here which makes use of the createQuery adaptee interface to serve the purpose of creating query to fetch similar author details for the specific similar search criteria.