PROJECT REPORT
TEAM 13

The goal of this project was to design, implement, test, and evaluate an application that assists program committee chairs(PC) and associate editors(AE) with identifying suitable candidates for their future committees.

Description:
Our system is a standalone application to be installed and run on system owned by PC or AE with admin credentials.

There are three main criterias on the basis of which we have divided our selection process

1. Search by Title & Citation Count -

Gives result of authors with their published paper, year, conference name, book title (book where article was printed) and citation count (number of papers where the author was cited)

2. Search by Article & Journal/Conference -

Gives result of authors with number of articles published and were part of specific conference.

3. Search by Committee

Gives result of authors who have been member of a specific committee.

4. Additional details of author -

Associated university, field of study, link to home page and region can be found by clicking on view button in the search results.

5. Apart from search, the users of the application can shortlist authors based on specific search criteria and export the selected authors to a .csv file.

New Features

1. The system has a an improved, consolidated, interactive and colorful user interface
2. New functionality of combined search for-
    - title and citation
    - article and conference
    - Committee and year

3. Hovering over the cells in the result table gives the full text of the table cell
4. Additional author data parsed from csrankings. In each search result, by clicking on view, details regarding author's - affiliated university, personal homepage, region and field of study can be viewed.
5. Apart from existing conferences, parsed and processed data for additional journals like tse, ieeesp, jacm and topsla. Committee data of conferences such as Ppopp, popl, issta, ismm, icse, esop, fse and ase is added to the existing data set.

Bug Fix and Refactoring

1. Fixed bugs raised after phase III on UI and QueryEngine front.
2. Refactored code to add design pattern -
    a. UI -

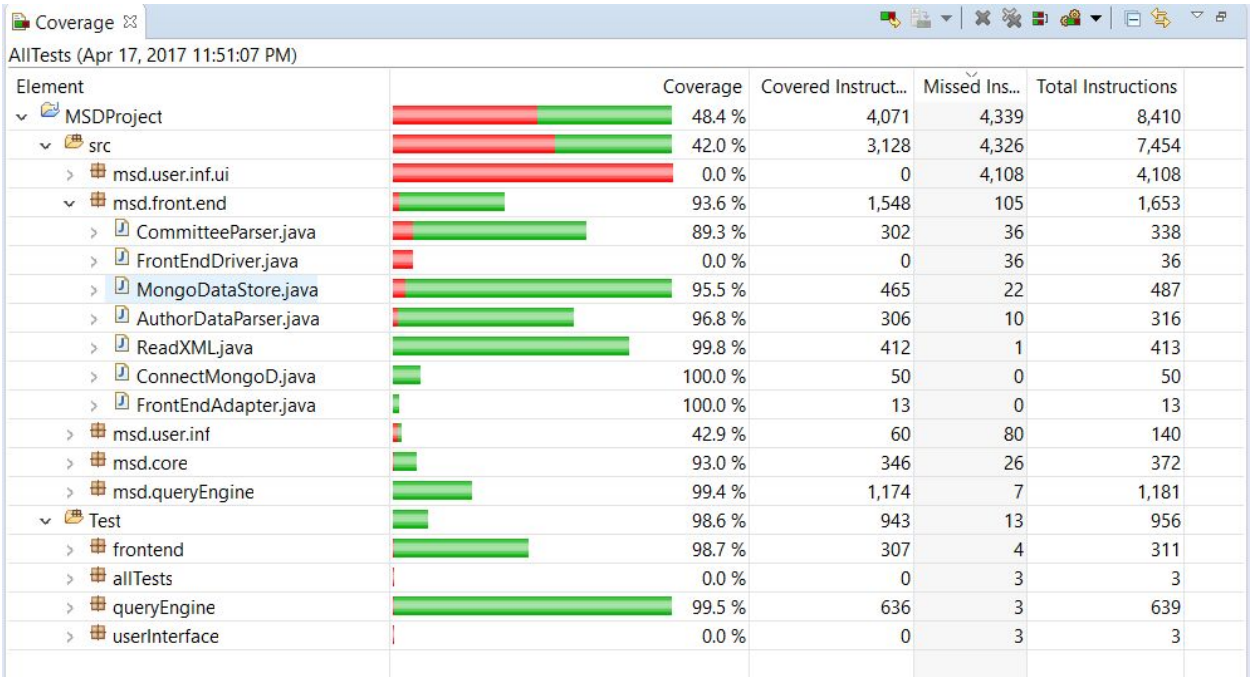
    i) Addition of author details on the user interface was done by using the “Adapter” design pattern. In order to send the query engine a new object, the AuthorQuery extends the previous UserQuery class and implements the InputCapture interface.

    ii) Created custom mouse listener using the “Adapter” design pattern by implementing the MouseListener interface

    b. Parser- DataReader is the common interface which is used by all the parser such as xml, committee , authorData and author. This interface is implemented by FrontEndAdapter and this adapter is extended by CommitteePaser and AuthorDataParser. This helped in eliminating the dead method which had to be implemented in these parsers, if this design pattern was not adopted.

    c. Query Engine - In phase III, display of data was according to use cases. It was unnecessary repetition of code. We cleaned the code to have one interface, multiple classes, with method call to each class through ‘factory class’ hence implementing the Abstract Factory Pattern.

Below is the test coverage screenshot



Element	Coverage	Covered Instruct...	Missed Ins...	Total Instructions
MSDProject	48.4 %	4,071	4,339	8,410
src	42.0 %	3,128	4,326	7,454
msd.user.inf.ui	0.0 %	0	4,108	4,108
msd.front.end	93.6 %	1,548	105	1,653
CommitteeParser.java	89.3 %	302	36	338
FrontEndDriver.java	0.0 %	0	36	36
MongoDataStore.java	95.5 %	465	22	487
AuthorDataParser.java	96.8 %	306	10	316
ReadXML.java	99.8 %	412	1	413
ConnectMongoD.java	100.0 %	50	0	50
FrontEndAdapter.java	100.0 %	13	0	13
msd.user.inf	42.9 %	60	80	140
msd.core	93.0 %	346	26	372
msd.queryEngine	99.4 %	1,174	7	1,181
Test	98.6 %	943	13	956
frontend	98.7 %	307	4	311
allTests	0.0 %	0	3	3
queryEngine	99.5 %	636	3	639
userInterface	0.0 %	0	3	3