

COMP4321 Final Documentation

*Team Members: Li Zhihang
Zhang Xiang
Wang Yiquan*

Overall Design

The project is a web based search engine consisting of 3 parts: user interface, search engine and database.

The database contains spider, indexer where the spider first crawl web pages and then pass the pages to indexer. The indexer will first do pre-process on each page's text content including stopword removal and stemming. Then each term will be stored in an inverted file which consists of (term ,posting-list) pairs.

Note that a posting list is a list documents which contains this term and the positions of this term is stored to support the phrase search function. Other information of the page like URL, last modification date will be stored in a forward file. After finish indexing all the pages, each page's PageRank score is calculated and will be used in search engine part.

The search engine receive query from user interface and process the query including stopword removal, stemming and phrase detection. Then it will use vector space model to find the most relevant pages in database. Other rank score like PageRank score, title matching is also considered when decide which pages should be retrieve.

The user interface is responsible for interact with users. User can input query through browser and submit the query to server and the interface will display all the relevant pages in the browser.

File Structure in Database

Inverted File: This is a class where contains Page Content Map and Page Title Map. In this class, it provides functions for other component to have access to these two Maps for functioning such as adding values in (addEntry) and Deletions (deleteEntry). Constructing this class is for over controlling two maps so that it simplifies control flow.

Forward File: This class is designed to store page specific information like title, URL, size and last modification date. It also stores some global statistic information like the tf max of a specific document. It as well contains all the terms a document has to support fast implementation of delete document operation. Relationships between pages are stored in this file since each forward file contains its child page's URLs so we can use the information to access its child pages.

URL and ID Map: The class called UrlToIdMap designed a map that stores URLs of a page with ID matched to each single of them uniquely. This enables us to have a faster access to ID of the page once we have an URL. In this class, method of addEntry is provided for other component of the project to have access to the map,

allowing them to add new pairs in. It also enables function that checks whether an URL exists.

Page Content Map: This map is realized in the Inverted File class. It is designed to make a direct connection between a word on a particular webpage and the word's information such as how many times it appears on this page and its position on the page. The information contained related to this word is stored using a linkedlist. We can obtain the detailed information as soon as a word is provided.

Page Title Map: This map has similar functionality with Page Content Map except it only processes page titles.

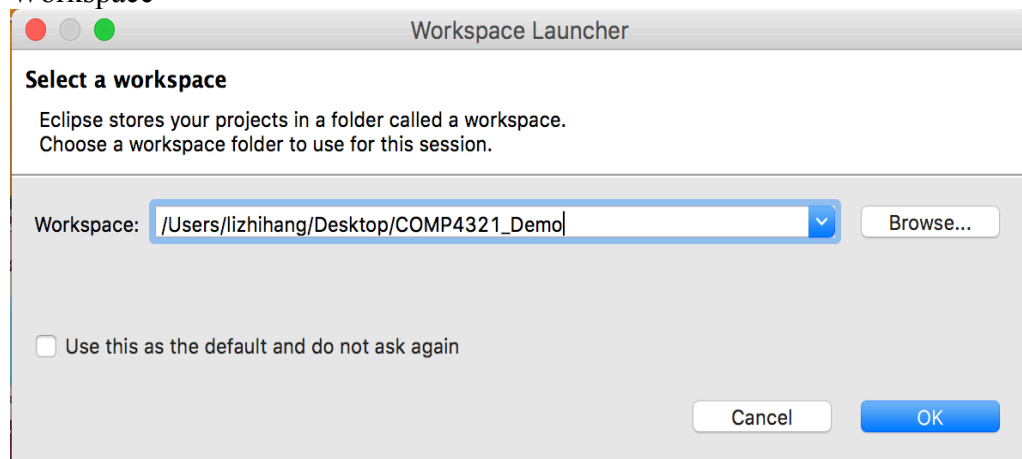
Algorithms Used

cosine similarity based on vector space plus page rank.

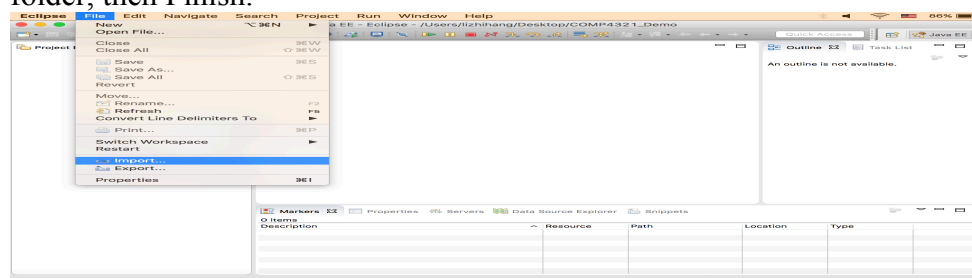
Details: Given a query, we preprocess it including stop word removal and stemming. For every word(phrase) in the query, we compute its weight as shown in the lecture slides and calculate cosine similarity. If there's match in the title, we increase the score dramatically by 0.3. Finally, we combine the cosine score with its page rank(summation) and return top 50 records or less if not so many documents are related to query).

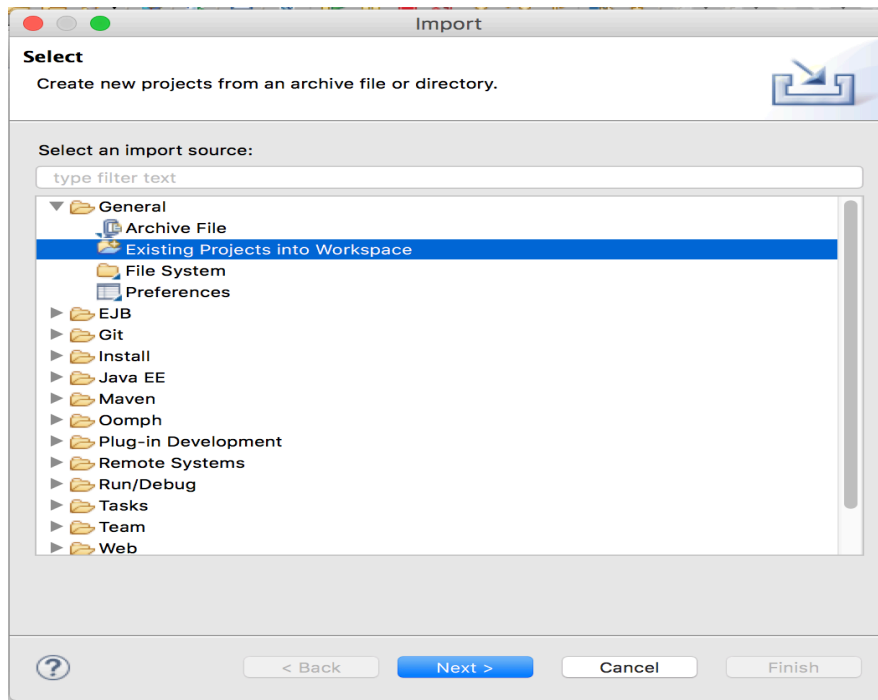
Installation procedure

1. Download zip file and unzip it.
2. Open eclipse and set the the folder where the unzipped file is located as Workspace

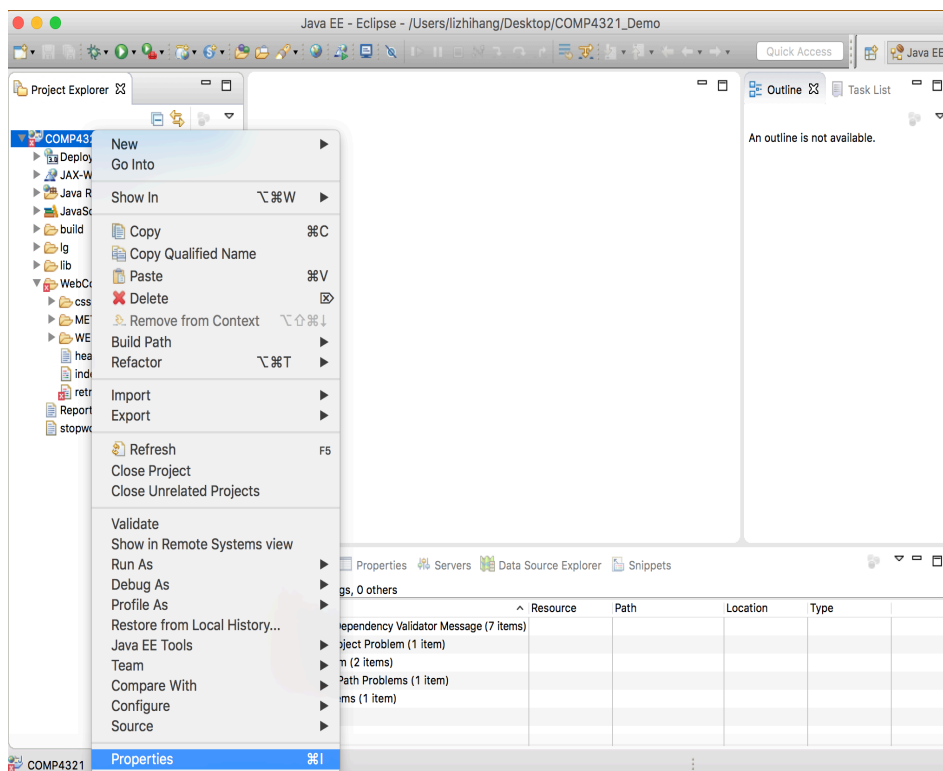


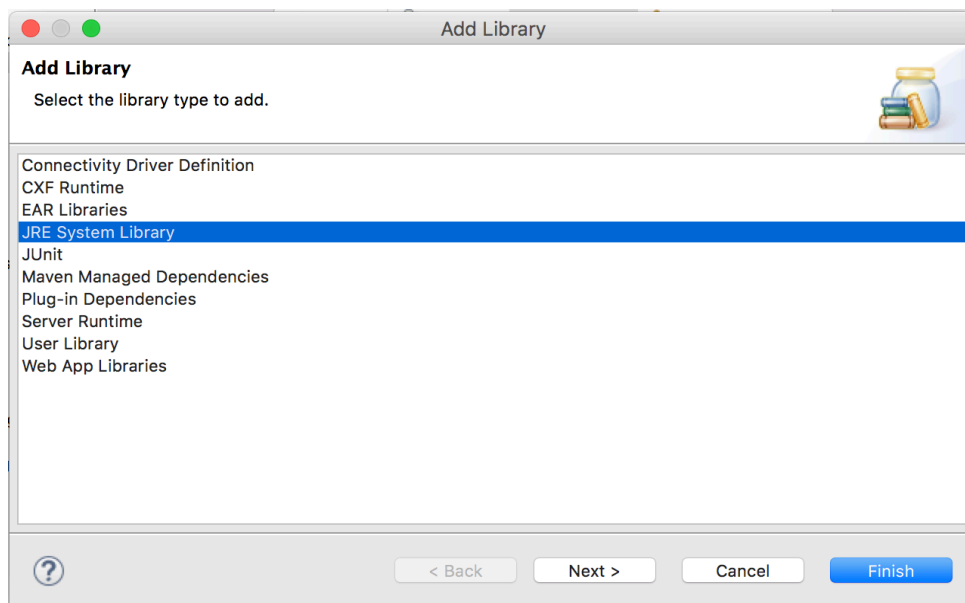
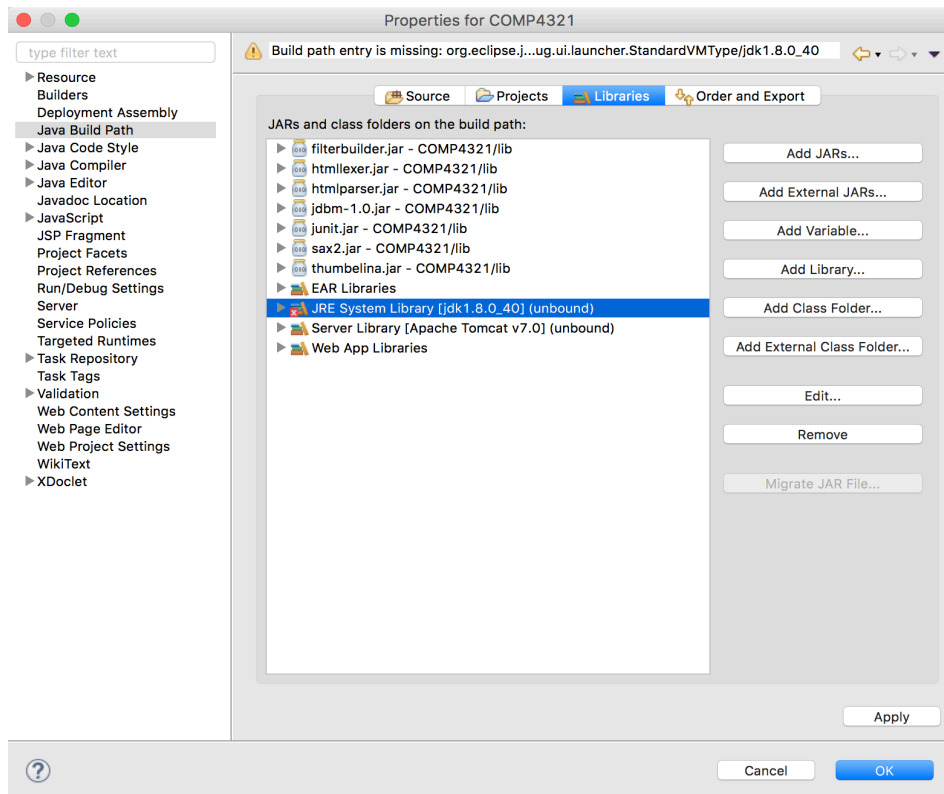
3. Click File -> Import..., then choosing "existing projects into workspace" under General, then click Next. After this, browse the unzipped file in this folder, then Finish.



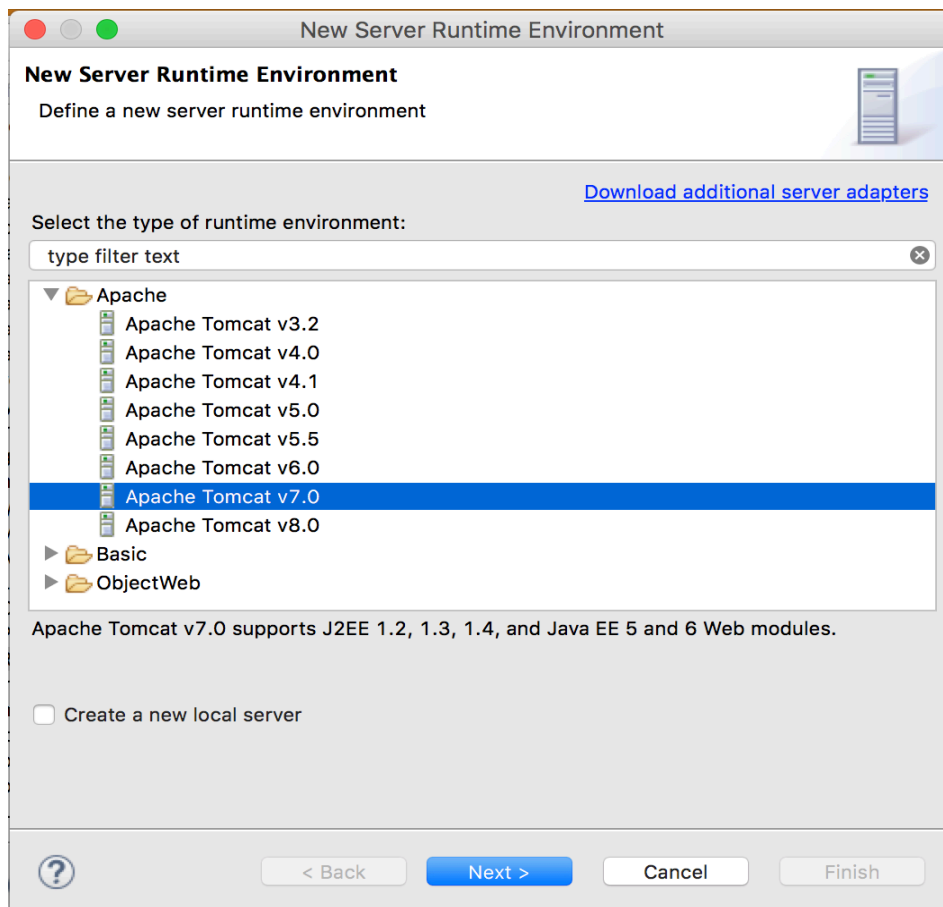
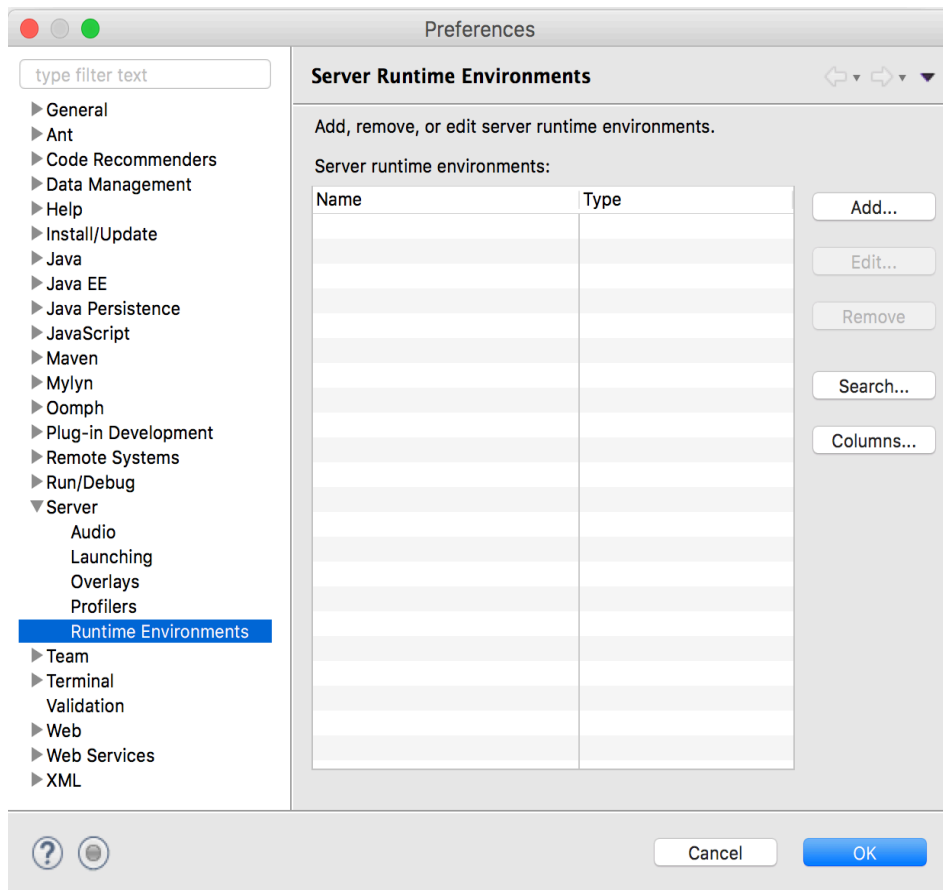


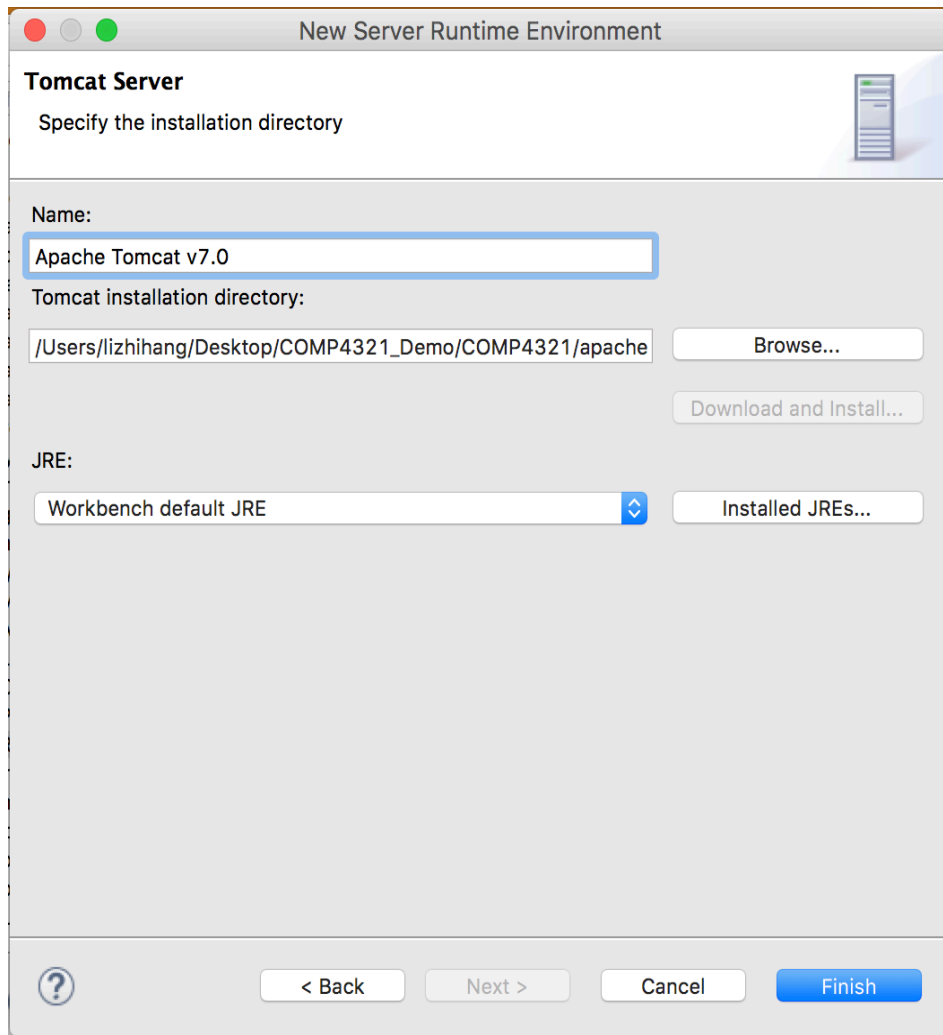
4. There is potential problem of existing error due to the fact that libraries version and path may be different in different computer. So a user may need to reset them on his or her own computer. You may do the following: right click on the project and select “Properties”, then go to “Java Build Path” and choose “Libraries”. Remove the one that contains error (most likely JRE system library). And then click on “Add library...”, select JRE System Library if that is the case, then click on Finish.





5. The next step is to add Apache Tomcat server, here we use version 7.0. Click preference of eclipse, choose “Runtime Environments” under “Server”. Then click on “Add”, select Apache Tomcat 7.0 and then click on Next. Browse the folder “apache-tomcat-7.0.69” located in the the project folder then Finish. The error now should be resolved and project is ready to run.

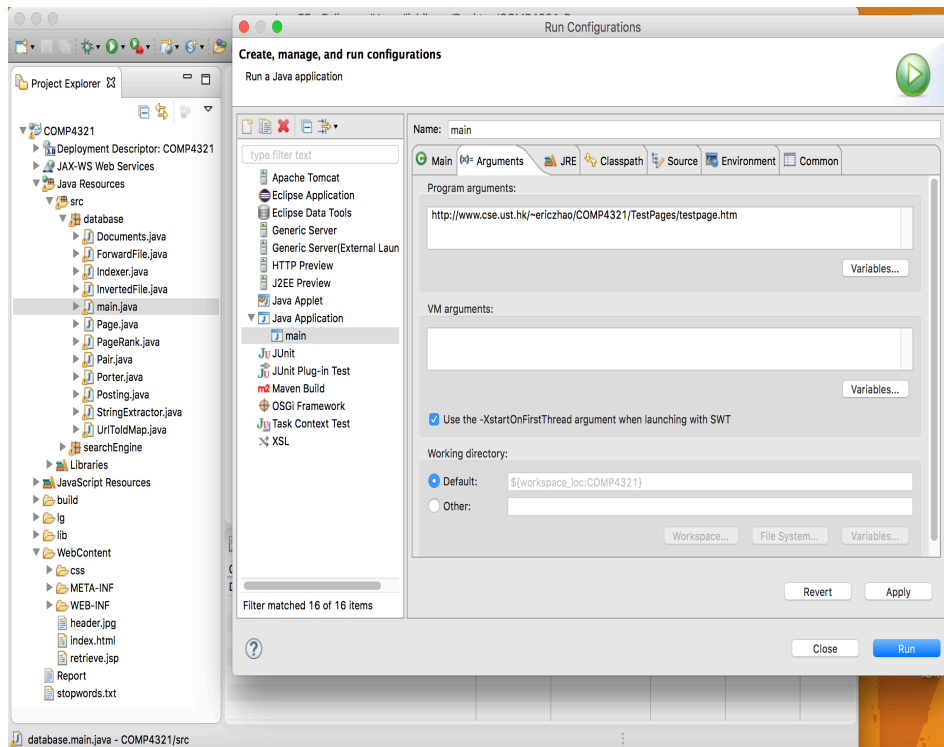




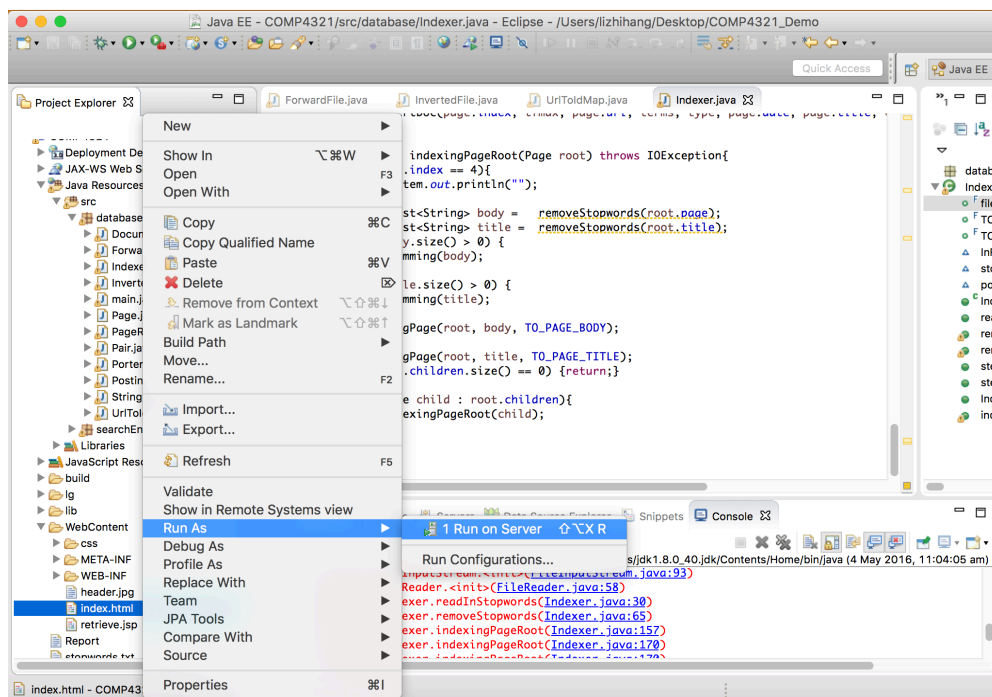
6. Note that Running the project should be under Java EE environment. Also, the user needs to change path that where files will be generating by this project in 4 Java files: Indexer.java, InvertedFile.java, ForwardFile.java and UrlToIdMap.java. The way to change it is to change the variable “filepath” into the location of the project where it is located. This variable is located at the very beginning of these 4 files.

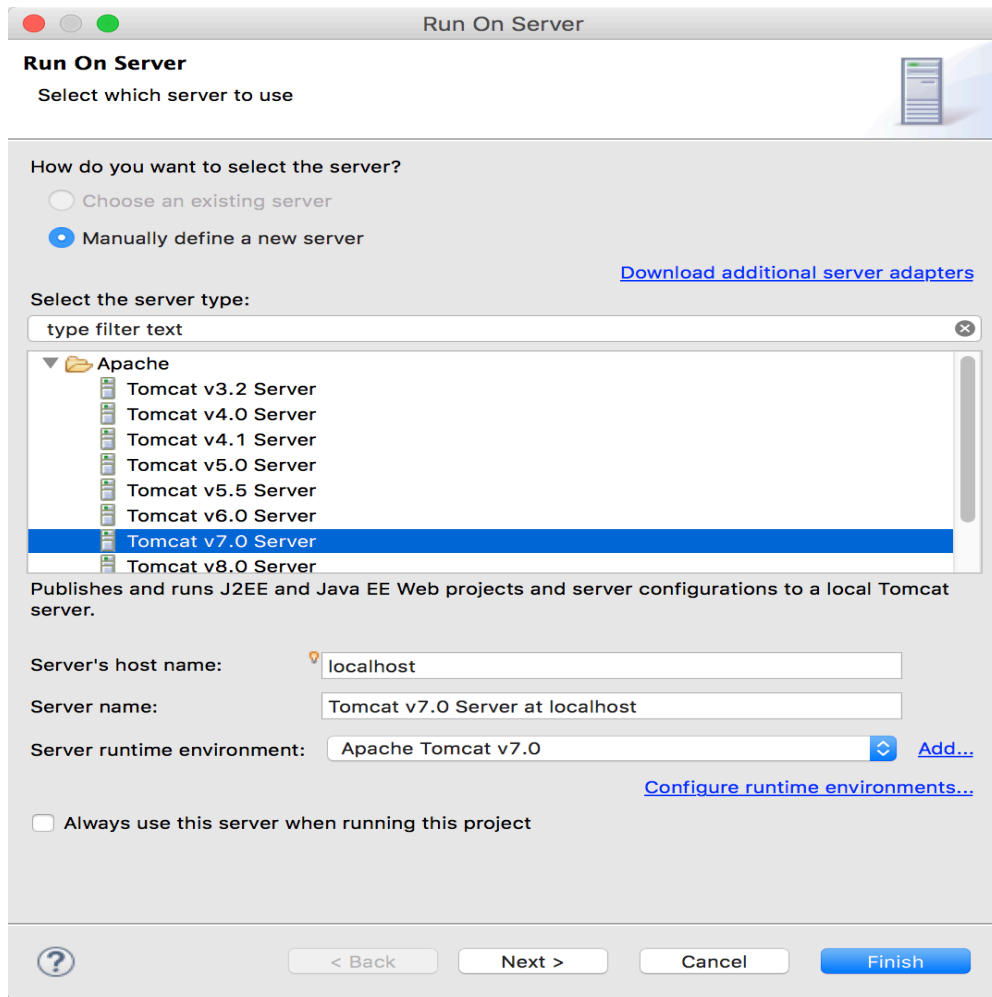
```
public final String filepath = "/Users/lizhihang/Desktop/COMP4321_Demo/COMP4321";
```

7. Right click on main.java under Java Resources -> database, select Run as... and then click on Run Configurations. Select main under Java Application. Paste the beginning site address into Program Arguments under Arguments, for example, the given test site “http://www.cse.ust.hk/~ericzhao/COMP4321/TestPages/testpage.htm”. Then click on Run.

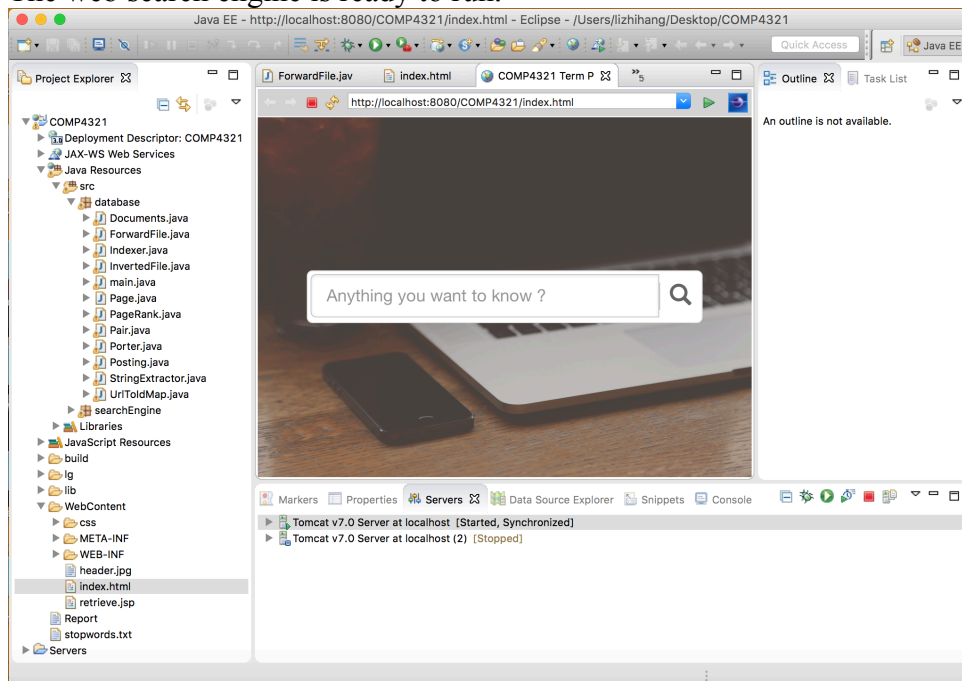


8. After finish running main.java, the search engine is ready to run. Right click on index.html under WEB-INF in WebContent, then click Run as... and run on server. Manually select Tomcat 7.0 Server then Finish.





9. The web search engine is ready to run.



Highlight of features

We implement a user-friendly interface using bootstrap and it can be visited both on PC and mobile phone.

We implement the PageRank algorithm and important pages (contains many useful links) will be favored in the retrieve process.

Our spider works at an exceedingly good speed.

Testing of the functions implemented.

We include a log file which record the test of database, search engine.

Conclusion

What are the strengths and weaknesses of your systems; what you would have done differently if you could re-implement the whole system; what would be the interesting features to add to your system,