

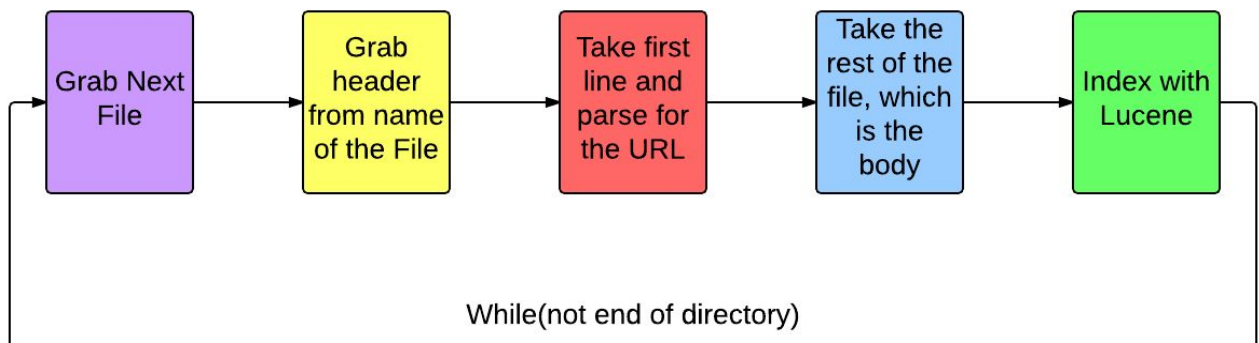
CS172: Introduction to Information Retrieval

Course Project: Web Crawler - Project 2

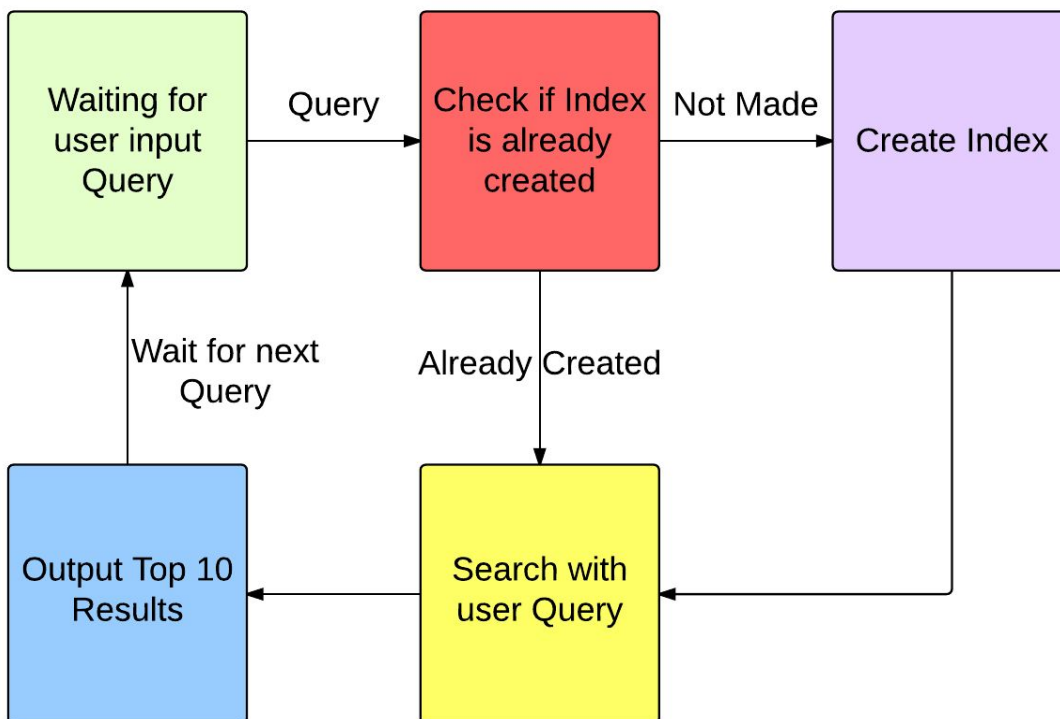
1. Overview of System:

a. System Architecture:

Design of the Index:



Design of overall system:



b. Index Structures:

- i. Now that the web application has been completed, the original public class indexer is not used in order to call the index and search_index, but instead it will be waiting for an user input inside of the web application. When the user successfully inputs a query word or query words, then it will call index and search.
- ii. `public void index(File folder, String outputDir):`
First thing to do inside of the index is to check whether or not the index has been created. If it has been created, it will return, as the index is already available and not need to re-create it. Inside of the index function the folder of crawled html results from part 1 of the program and will parse each html file by grabbing the head as the title, the first line as the URL and the whole contents of the pages as the Body, and using Lucene to add each individual html file into the index. Finally it is outputted the completed index into the outputDirectory.

c. Search Algorithm:

- i. A String queryString will be passed in that contains the query, and int called topk what will contain the number of results that is being returned. A list of WebDocuments is returned, which is a class that is defined in another file that contains the URL, Body, Title and the score. The search function will parse the query using QueryParser inside of Lucene. Then it will instantiate a new list of WebDocuments, in order to store and update the list of top documents as they are found.
- ii. For the search algorithm, the weight to find the query words inside of the title is 1.5 and finding the word inside of the body of the file is 1. The indexSearcher structure is used in order to search through the index. The search function is used to sort the indexes, and then store the result inside of a TopDocs variable. Then, the first 10 results are taken and then the

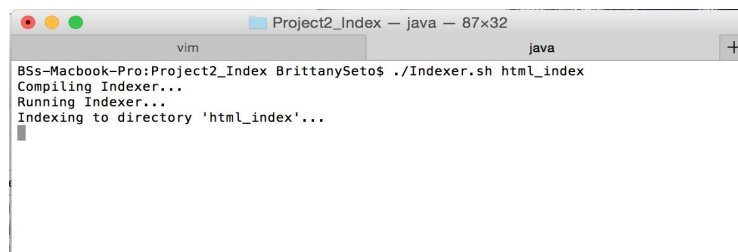
title, text, score, and the url are taken after. These four components are taken as input it into the list of WebDocument.

d. WebDocument class:

The specific class called WebDocument stores the URL, Title, Body, and Score all in a simple convenient place to access it after searching.

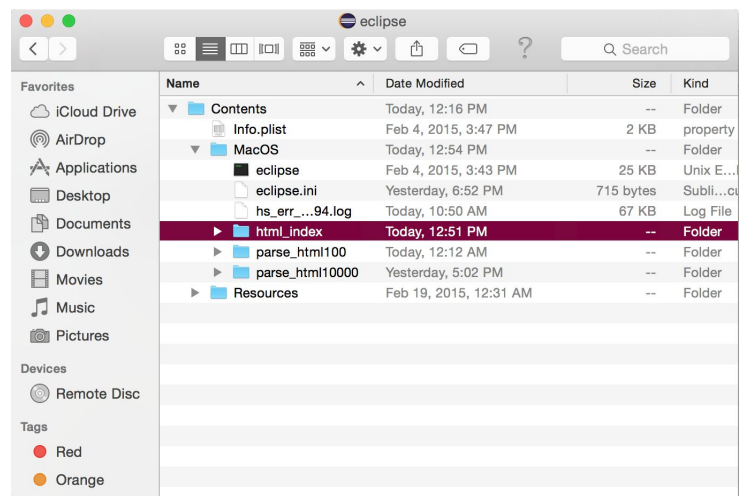
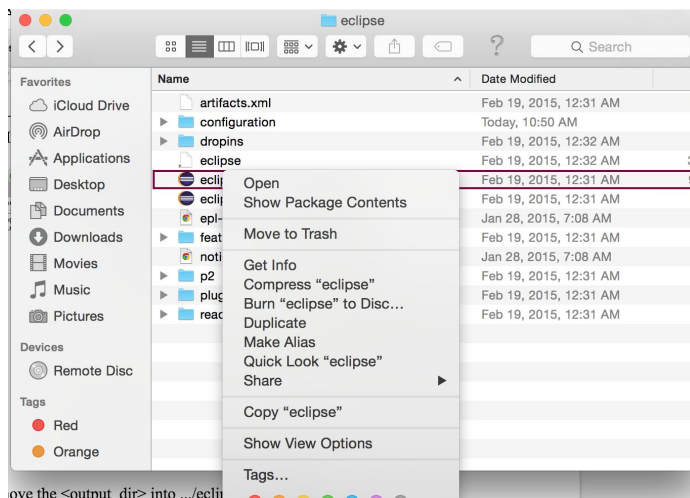
2. How to Run:

- Project2_Indexer
 - `./Indexer.sh <output_dir>`



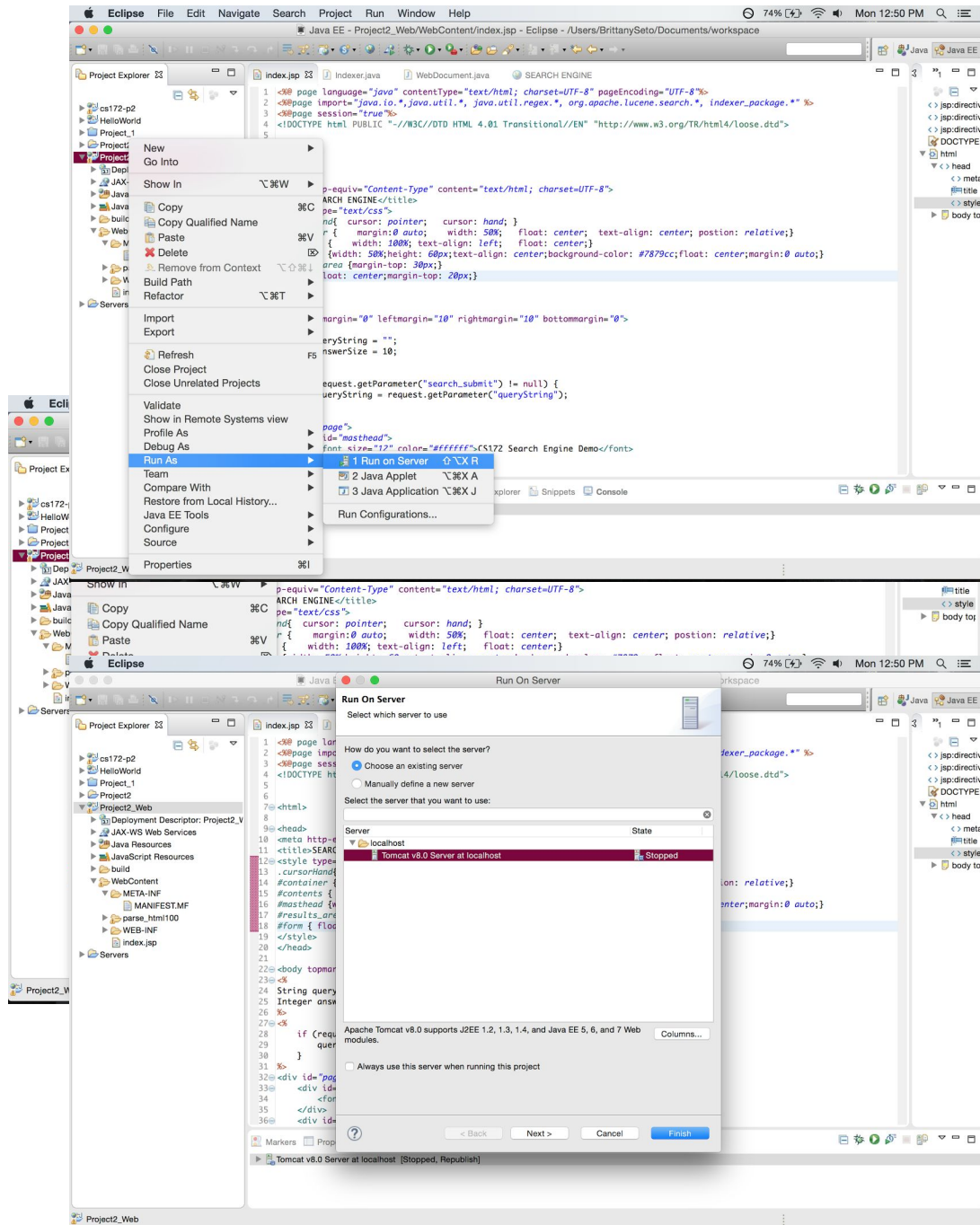
```
Project2_Index — java — 87x32
vim                                     java
BSs-Macbook-Pro:Project2_Index BrittanySeto$ ./Indexer.sh html_index
Compiling Indexer...
Running Indexer...
Indexing to directory 'html_index'...
```

- move the `<output_dir>` into `.../eclipse/contents/MacOS`



- Project2_Web
 - import Project2_Web into eclipse
 - right click on the project on the left sidebar

- go to "Run As"
- select "Run on server"



- choose desired server

3. Screenshots:

