**Comp4321 Project Documentation**

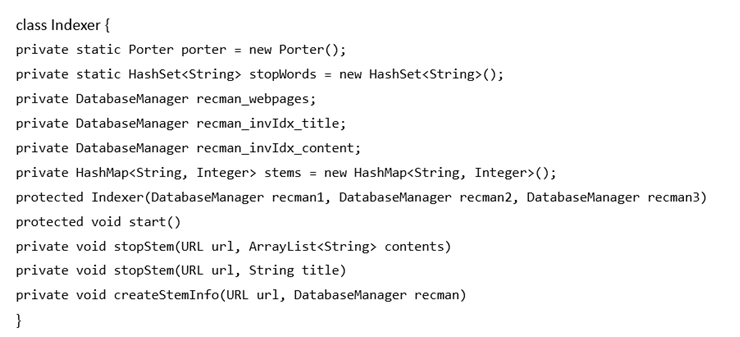
**Group No.: 15**

**Overall design of the system**

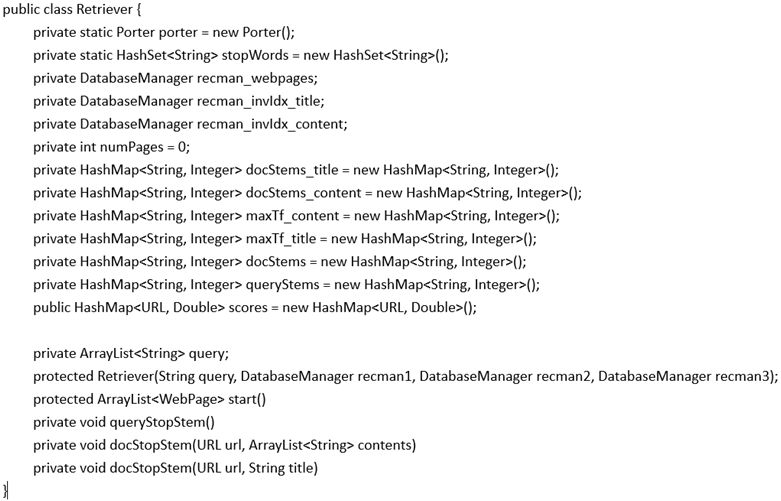
The system fundamentally consists of three main classes: Crawler, Indexer and Retriever, and responsible for different processes required for information retrieval. The Crawler is used for crawling the pages under the domain of [www.cse.ust.hk](http://www.cse.ust.hk).



The Indexer first filters out the stop words provided in the stop word list and implements the Porter’s algorithm to produce the stems for both the title and the body of the pages the crawler crawled. The stems are saved in two inverted file, one for stems in title and one for stems in page body.



The Retriever first process the input words by the user to filter out stop words and convert into stems. Then the input stems and each of the pages in the index database are compared, to produce a list of pages sorted with the similarity comparison, with a maximum number of results as 50. Finally, the interface provides a channel for the user to input the searching words for the processes of the Searching engine and to display the searching results.



**File structures in the index database**

In the index database, objects with type as “Webpage” are stalled. The class “Webpage” is implemented as follows:



A hashmap is used for saving the stems of webpages, which contains the stem itself as a String variable and the frequency of the stem as an Integer variable.

**Algorithm used**

The weighting formula of each stem term tf\*idf/max(tf). When the input stems are compared with the pages, the similarity measurement is based on cosine similarity. Since the matching in page titles should be given priority, the ranking of similar pages are first according to the degree of similarity between title and input, and according to that between page body and input when the previous measurement produces same results.

**Installation procedure**

Place the following files in the same directory:

Crawler.java

Test.java

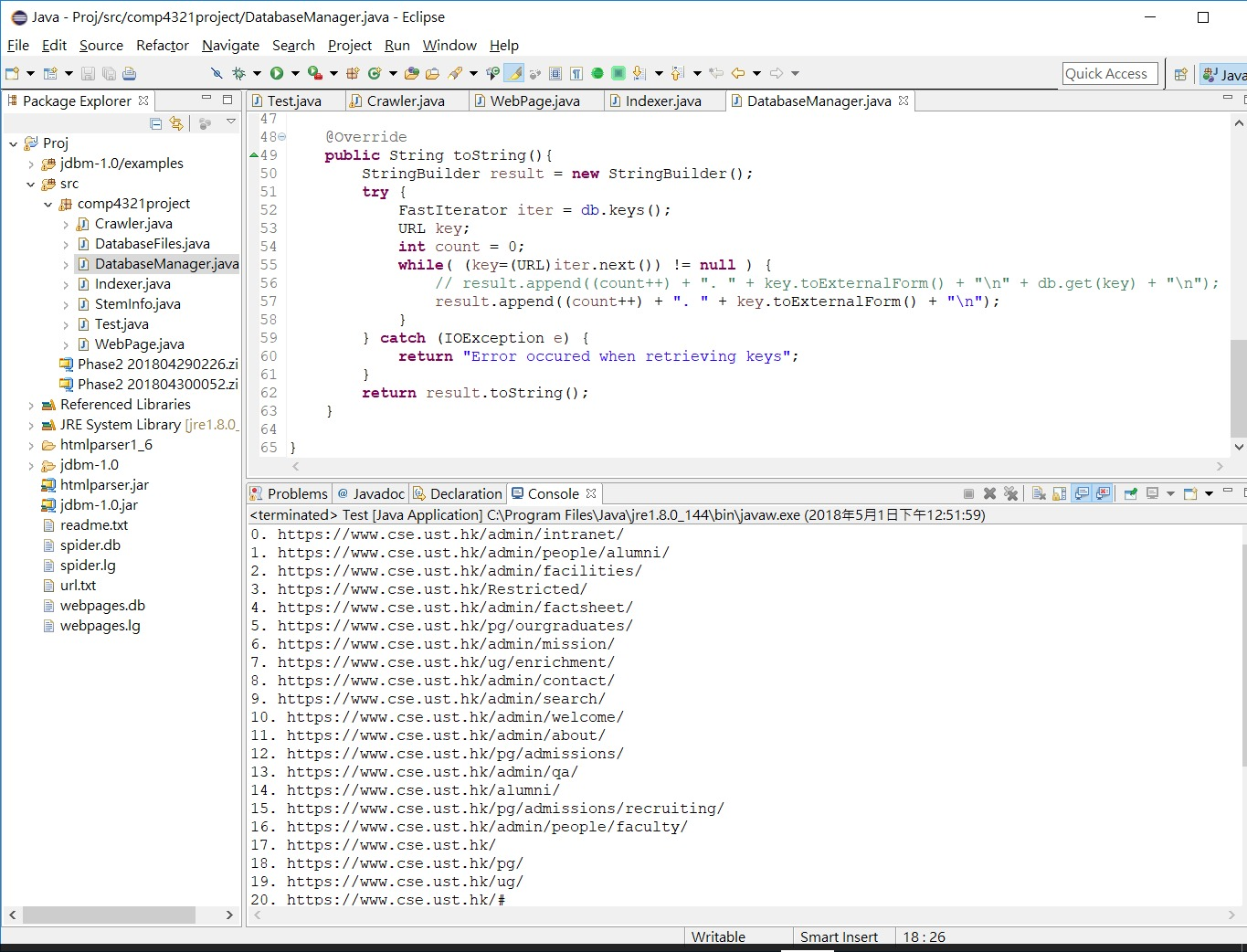
jdbm-1.0.jar

htmlparser.jar

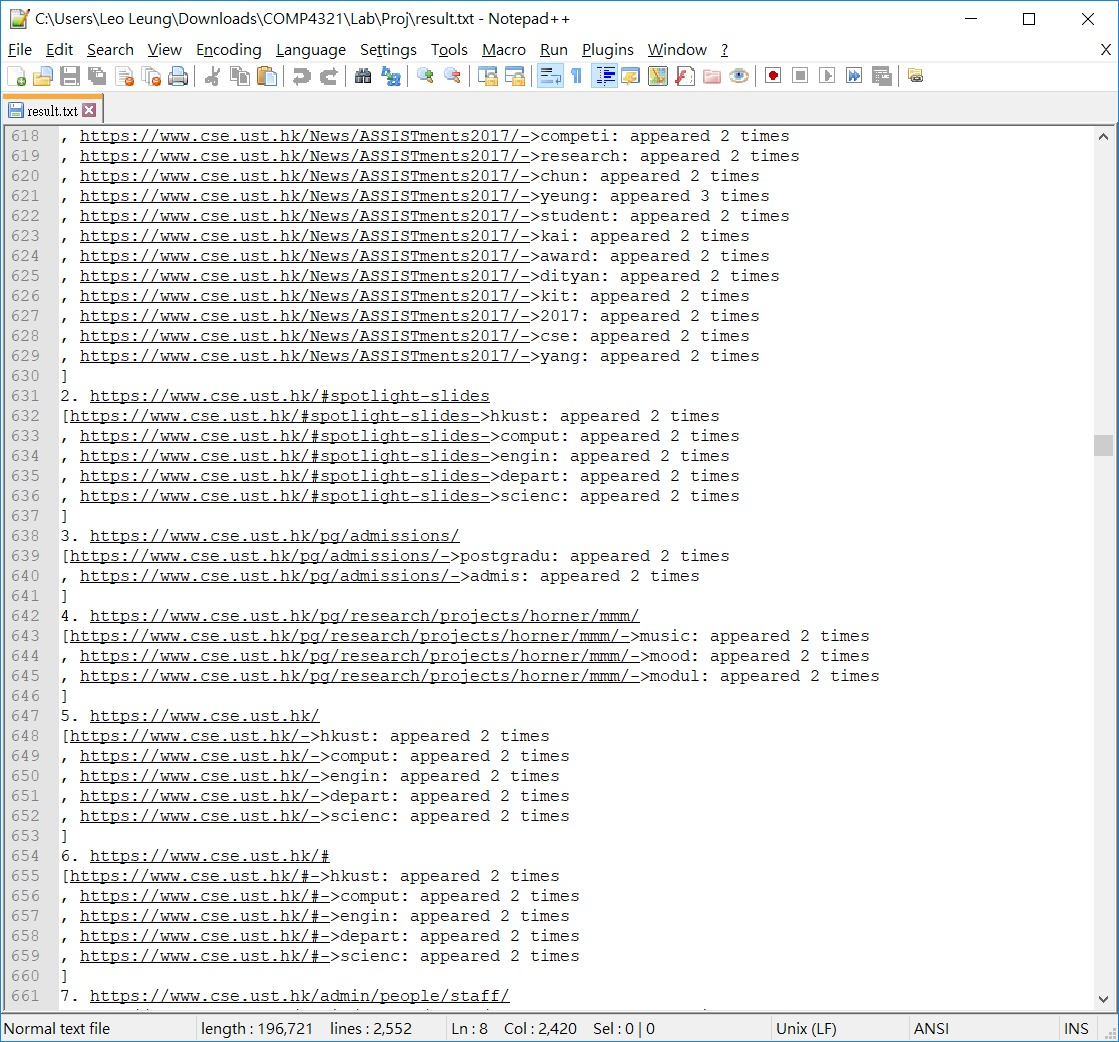
The crawler can be compiled by command "javac -cp \\* Crawler.java" and execute by "java -cp .:\\* Crawler"

Test by eclipse

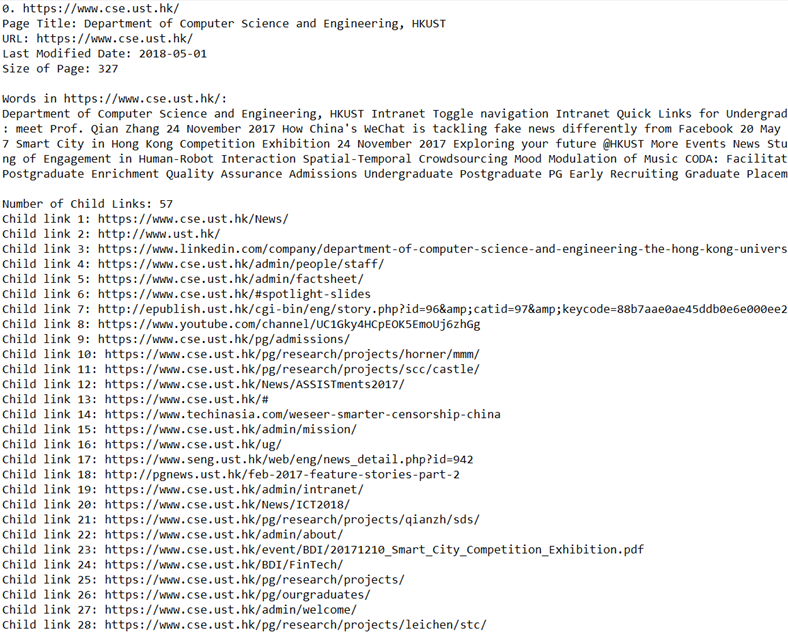
**Testing of functions implemented**

Testing of crawler function:

Testing of indexing function:



Testing for retrieval function



**Conclusion**

Our system is having a simple design for the interface, therefore it is user-friendly in terms of simple searching. However, it can only search by regarding each of the input words as the same weights. In reality, there always exists “so-called” keywords in each of the searching words, which should be regarded as “more important” for the searching, and it should carry higher weight. So, if we could re-implement the retrieval part, we would like to add functions for users to indicate the keywords of their searching words, or even functions for the system to investigate the searching words and find the keywords by the functions instead of indicated by the user, which makes it more user-friendly and “smart”. It will also be interesting to implement the system to be able to receive words for exclusion, which can make the searching result more “useful” for the user.