## Design of the Database Scheme

1. **Mapping Table (for URL)**

A HashTable is used for storing the PageID and URL of each web page. The key of it is PageID, which is used to retrieve URL of the corresponding page, and basically, it will be assigned in ascending order. Every web page, even the child link, will be added into the entry after checking that the URL have not been added before.

|  |  |
| --- | --- |
| PageID | URL |
| E.g. 1 | <http://www.cse.ust.hk> |
| 2 | http:// …. |
| … | … |

1. **Mapping Table (for Keywords)**

For the keywords mapping table, it is also built with a hash table structure. It contains all the keywords retrieved from every web page. However, in order to benefit the searching process afterward, the wordID is generated by using a hash function which is suggested by the Karp-Rabin algorithm. Also, stemming and stop word removal will be done for the keyword before adding the entry.

|  |  |
| --- | --- |
| Keyword | wordID |
| E.g. comput | (a long type variable) |
| … | … |

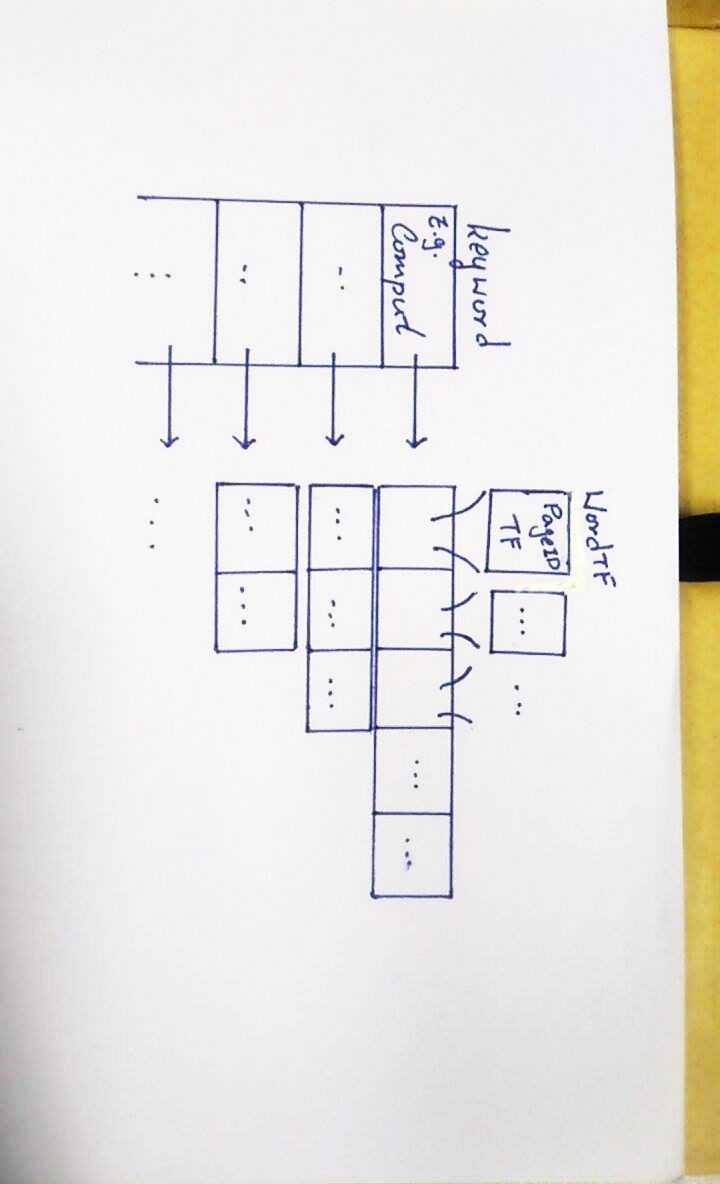
1. **Indexer**

We built a class called PageStruc. The title, last modification date, size and the child link information will all be initialized in this class. A pageID will be given to the constructor and use its URL which is retrieved by seeking the mapping table for URL to perform different functions. And finally, we will have 30 PageStruc objects according to the number of pages requirement. Below is the basic idea of the whole indexer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pageID | Title | Last modification Date | Size | Node  (child link) |
| E.g. 1 | Department | 2017-3-30 | (# of char) | (Vector String) |
| … | … | … | … | … |

1. **Inverted Index**

Other than hashtable, we used hashmap to construct the inverted index. The keyword used to initialize the key(directory) of the map. And for the mapped value (posting), we passed a list of object which is an own defined class called wordTF to it. There are two data members in class wordTF, they are PageID and TF. PageID has stored the id of the page that contains the corresponding keyword. And TF is the number of occurrence of the corresponding keyword in that page.



1. **Approximate percentage contribution**

Chan, See Ki 40%

Tong, Hau Kei 40%

XU, Zihan 20%