1. Crawler
2. Indexer
   1. Create Connection with two database of indexer & Crawler.
   2. Create and start Threads
   3. retrieve Documents from Crawler
   4. for each link of Documents check if its already in indexer database or not if yes remove it else do nothing
   5. get html document of link
   6. remove tags - stop words
   7. stem words & count number of header & title & other pf each word in document
   8. create inverted file using hashmap to calculate TF DF positions
      1. for each word
         1. if word is stored in hashmap
            1. increment tf & add position in array
         2. else
            1. store word in hashmap with tf = 1 & position
            2. add word to DF map with value 1
   9. create Documents that will be stored in database
   10. insert in database
   11. finally update Df for all words
3. Query Processor
   1. Wait for request for suggestion
   2. Wait for request for query search
4. Get query
5. Check if query phrase search or not
6. Get documents according to submitted queries
7. Store documents in hashmap
8. Send documents to ranker & phrase search if found
9. Get array of urls that will be sent to interface
10. Get title & description of 10 links
11. Send JSON Document contains 10 documents as response to interface
    1. Wait for request to pages
       1. Repeat step -> g to h in above algorithm
12. ranker
13. phrase search

