**Design Document**

**Description of architecture:  
  
Frontend:** The project was bootstrapped with [Create React App](https://github.com/facebook/create-react-app).

In the project directory, you can run:

### **npm start**

Runs the app in the development mode.

Open [http://localhost:3000](http://localhost:3000/) to view it in the browser.

The page will reload if you make edits.

You will also see any lint errors in the console.

### **npm test**

Launches the test runner in the interactive watch mode.

See the section about [running tests](https://facebook.github.io/create-react-app/docs/running-tests) for more information.

### **npm run build**

Builds the app for production to the build folder.

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.

Your app is ready to be deployed!

See the section about [deployment](https://facebook.github.io/create-react-app/docs/deployment) for more information.

### **npm run eject**

Note: this is a one-way operation. Once you eject, you can’t go back!

If you aren’t satisfied with the build tool and configuration choices, you can eject at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except eject will still work, but they will point to the copied scripts so you can tweak them. At this point you’re on your own.

You don’t have to ever use eject. The curated feature set is suitable for small and middle deployments, and you shouldn’t feel obligated to use this feature. However we understand that this tool wouldn’t be useful if you couldn’t customize it when you are ready for it.

**Backend:**Let’s discuss algorithms involved in each backend file.  
  
**Data Structures used**: B-Tree, Hash-map, lists, arrays.  
  
**Pre\_process.py:**  
  
This function is used to pre-process text that is given as input. This is used while precomputing tf-idf values during document precomputation and when the query is given by the user.  
  
**Algorithm:**

* The text is changed by turning the uppercase characters to lowercase characters.
* Tokenization: It basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language.
* Removing Stop words: We iterate through the text and look for stop words and remove them.
* Porter Stemming: The Porter stemming algorithm (or ‘Porter stemmer’) is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.

**Docs\_prcompute.py:**

Calculate the tf-idf values, no. of unique words, document vectors, df values of the text.

**Algorithm:**

* Gather the text in all of the documents.
* Pre-process the text.
* Set (B-Tree) data structure has been used to gather all of the words in the text.
* Iterate through all of the text and calculate the df value of each word using the map data structure. (Hash Map)
* Iterate through the words in the document and calculate the tf values of each word.
* Tf-idf = tf\*idf
* Append the document vector into a list.
* Save the tf-idf values, no. of unique words, df values of each word and document vectors and store the objects in persistent storage.

**Query.py:**

We can use the expanded, pre-processed text and return the most relevant queries.

**Algorithm:**

* Expand the query by appending similar words to the end of the list of words.
* Pre-process the text (the method we used earlier)
* Extract relevant objects from persistent storage.
* Calculate the n-dimensional vector of the query.
* Iterate through all document vectors and find the score, append the score in a list.
* Sort the list according to the decreasing value of it’s score.
* Return the top **K** results.