**DESIGN DOCUMENT**

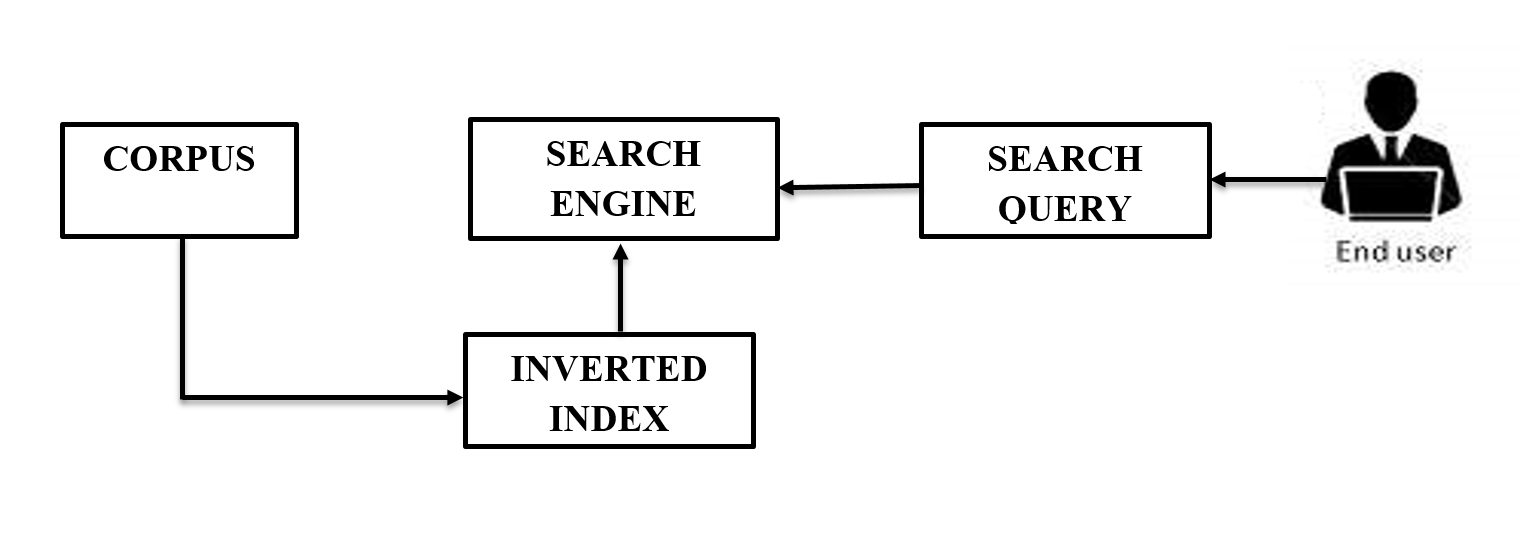
**Team**

* Rynaa Grover 2017A7PS0258H
* Gokul Kumar M 2017A7PS0236H
* Daivat Bhatt 2016B5A70952H
* Samarth Singh 2016B3A70609H

**Corpus Used**

Movie Plots - <https://www.kaggle.com/jrobischon/wikipedia-movie-plots>

**Project Architecture**

****

**Formulae Used**

* **TF(t)** = (Number of times term t appears in a document / Total number of terms in the document).
* **IDF(t)** = log\_e(Total number of documents / Number of documents with term t in it)

**Steps Involved in Preprocessing**

1. Removal of Punctuation in the corpus documents
2. Tokenization of Documents
3. Converting all words to Lowercase
4. Removal of stop words
5. Stemming (Used Porters Stemming)
6. Creation of Word Level Inverted Index

**Major Data Structures Used**

* **Dictionary** is used for making Inverted Index and this is chosen because of it’s O(1) access time.
* **List of Lists (2D list)** is being used so that the whole preprocessing becomes dynamic and new files can easily be added/modified and the whole preprocessing need not be done again.

**Advantages of Using Inverted Index**

* Inverted index is used to allow fast full text searches, at a cost of increased processing when a document is added to the database.
* It is easy to develop.

**Disadvantages of Using Inverted Index**

* Large storage overhead and high maintenance costs on update, delete and insert.

**Time Required:**

* 3.4299 seconds required for preprocessing 1000 documents
* 0.005 seconds required for searching a single word.