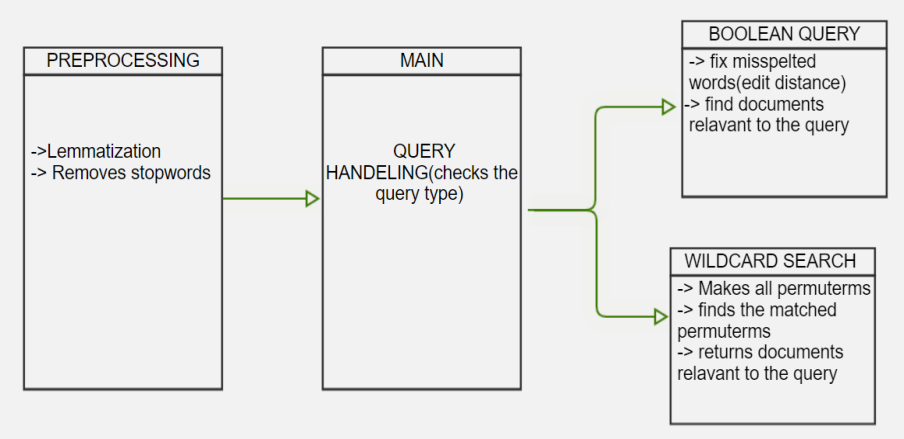
DESIGN DOCUMENT



The documents are first pre-processed by tokenization followed by removing all the stopwords and Porter stemming. A dictionary is used to create an inverted index and permuterm index for all the tokens.

When the code is executed, it takes input for either Boolean query or a wildcard query.

For a Boolean query, we call the function bool_query which first checks for the word(s) from the inverted index dictionary and returns the retrieved documents.

For a Wildcard query, we first process it based on its type (X*, *X, or X*Y) and then call the processQuery function which uses a prefix_match function to match the permuterm. We then print the retrieved document list based on the matched permuterm.

Also, edit distance method is used for spelling correction in Boolean queries.

The central data structures used are:

- Lists
- Dictionaries
- Stacks

RUNNING TIME

Pre processing time for our program is about 25-30 seconds

```
PS F:\3-2\IR\A1_2019A7PS0173H_2019B3A70375H_2018B5A70785H> python -u "f:\3-2\IR\A1_2019A7PS0173H_2019B3A70375H_2018B5A70785H\main.py"
The execution time is : 29.738434076309204
Enter Query :
```

For query handling running time for:

Boolean query is about 0.001 seconds

```
Enter Query : antony AND brutus
DOCUMENTS RETRIEVED
 ['antony-and-cleopatra_TXT_FolgerShakespeare.txt', 'henry-v_TXT_FolgerShakespeare.txt', 'julius-caesar_TXT_FolgerShakespeare.txt']
The execution time is : 0.0010242462158203125
```

Wildcard query is about 0.015 to 1.04 seconds

```
Enter Query : bum*
DOCUMENTS RETRIEVED
['a-midsummer-nights-dream_TXT_FolgerShakespeare.txt', 'measure-for-measure_TXT_FolgerShakespeare.txt', 'timon-of-athens_TXT_FolgerShakespeare.txt', 'twelfth-night_TXT_FolgerShakespeare.txt', 'romeo-and-juliet_TXT_FolgerShakespeare.txt']
The execution time is : 0.01800227165222168
```