Design Document (Phase 1)

Basic Stages:

- XML parsing (prefer SAX parser over DOM parser. If you use a DOM parser, you can't scale it up for the full Wikipedia dump later on)
- Tokenization
- Case folding
- Stop words removal
- Stemming/Lemmatization
- Posting List/Inverted Index Creation
- Optimize

Desirable Features:

- Support for Field Queries - Fields include Title, Infobox, Body, Category, Links, and References of a Wikipedia page. This helps when a user is interested in searching for the movie 'Up' where he would like to see the page containing the word 'Up' in the title and the word 'Pixar' in the Infobox. You can store field type along with the word when you index.
- Index size should be less than one-fourth of the dump size
- Index should be created in ~60 sec for C++/Java and ~150 sec for Python.

One important thing to note is the trade-off between index size and search time - a highly compressed index might increase the search time.

- The project directory should have an 'index.sh' file which would contain the necessary script to run your code.
- We will run your code like this:

```
$ bash index.sh <path_to_wiki_dump> <path_to_invertedindex_output> <invertedindex_stat.txt>
```

`invertedindex_stat.txt`: This file should contain two numbers on separate lines

- Total number of tokens (after converting to lowercase) encountered in the dump
- Total number of tokens in the inverted index

A sample of what `index.sh` might contain if you are using Python:

```
python wiki_indexer.py $1 $2 $3
```

A sample of what file `invertedindex_stat.txt` might contain:

```
42269602
1314519
```