Application Design Document

Index Page no

**Overview 2**

**System Design Overview 2**

**Detail System Design 4**

**Test Case & To run 5**

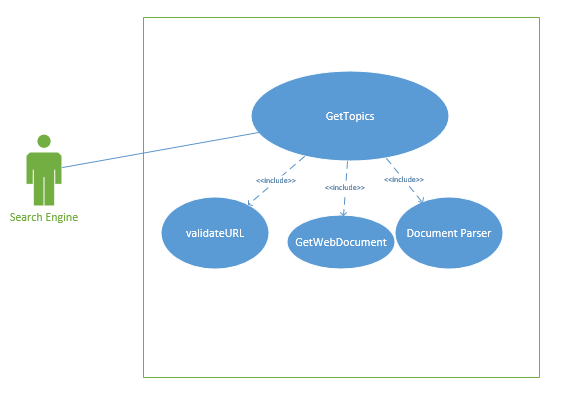
* Bijay Sharma

bijays.nitdgp@gmail.com

1. **Overview:**  To optimize a search engine, it’s important to increase the capability of search engine to identify what a webpage is about. In other word, we need to find the topic of the page or most frequency word used in that web page. This application will full fill the purpose. It will take an URL, read the web page, parse it and will return topics for the page.
2. **System design overview (uml diagram):** The application (WordDensityAnalysis) consists of two classes “Parser” and “WordParser” run by WordDensityDriver. The Parser class takes url string from “WordDensityDriver” and gets the document from the web for that url. It then uses “wordParser” class. The “wordParser” class then makes a Trie data structure consist of TrieNode (words) to find the most frequent words to find what the web page is all about.

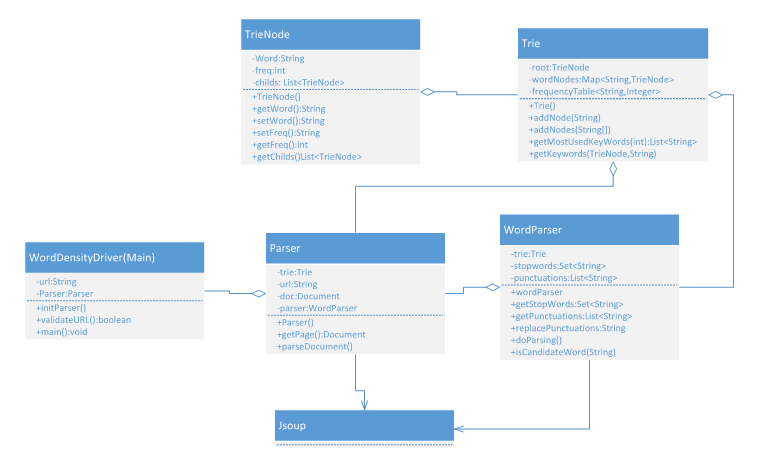
**In** this application, **composition has been preferred over inheritance.** Each class has reference to another class to use it.

Below diagram is use case diagram of the whole application.



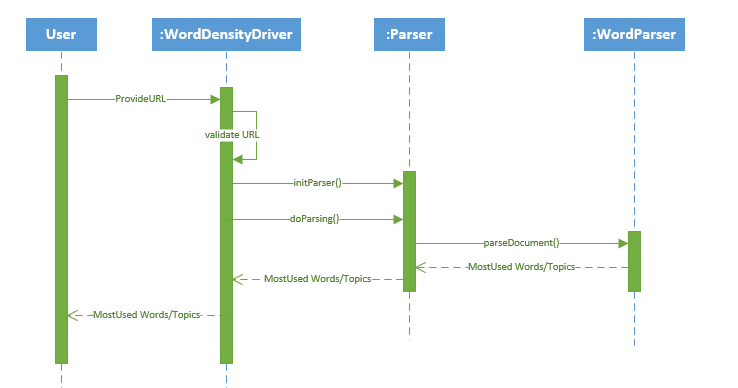
**Use case**

Each class has reference to another class to use it. Composition has been preferred over inheritance.



**Class Diagram**

Below diagram is system sequence diagram which shows how classes interact inside the application**.**



**System Interaction Diagram**

1. **Detail System Design (how the system works):**  The entire application consists of two functionalities.

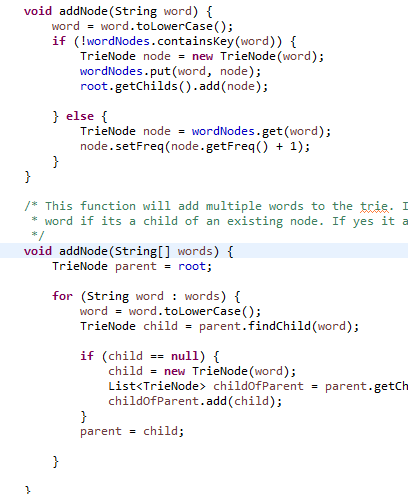
The first function of the system is to read the web page and parse it and store it in a document. This has been done using an external java library (Jsoup).

The second function of the system is to count the number of words in that document. Each word may be of length one (Cuisinart) and may also consists of more than two words (Compact toaster). These will be our candidate keys. However, we need to remove any stop word comes in the candidate keys such as ("?",",","…..). For simplicity, stopwords has been stored in stopwords.text.

**Data Structures:**

In this document, Trie data structure has been used. Each word has been used as node of this Trie. Each node has a value, frequency and list of its children.

Methods inside Trie:



The first node adds a single node to the Trie and second is capable of adding multiple nodes to the Trie.

**Example:**

Suppose we have word list [“stuart”, “stuart little” ,”sutart little mouse”]. After adding these words, the trie would be as follow.

[“stuart”, 3] 🡪 [“little”, 2] 🡪 [“mouse”,1]

Each word has mentioned with their frequency.

**Example of a sentence parsing:**

Sentence: how-to-introduce-your-indoorsy-friend-to-the-outdoors

“how” is a stop word? Yes 🡪 Ignore

“to” is a stop word? Yes 🡪 ignore

“introduce” is a stop word? No 🡪 Add as Node

“your” is a stop word? Yes 🡪 ignore

“indoorsy” is a stop word? No🡪 Add as node

“friend” is a stop word? No 🡪 Add as Node

“to” is a stop word? Yes🡪 Add “indoorsy friend” as a node

“the” is a stop word? Yes🡪 ignore

“outdoors” is a stop word? No 🡪 Add as a node.

[“introduce”, 1], [“indoorsy”,1] 🡪[“friend”, 1],[“friend”, 1],[“outdoors”,1]

1. **Test Cases Execution:** To run the application Use following command

**javac – jar Assignment.jar “URL”**

**Note: Please put the url in double quote.**

Input URL:

<http://www.amazon.com/Cuisinart-CPT-122-Compact-2-Slice-Toaster/dp/B009GQ034C/ref=sr_1_1?s=kitchen&ie=UTF8&qid=1431620315&sr=1-1&keywords=toaster>

Output:

KeyWord:[toaster]

KeyWord:[amazon]

KeyWord:[stars]

KeyWord:[cuisinart]

KeyWord:[prime]

KeyWord:[toast]

[**http://blog.rei.com/camp/how-to-introduce-your-indoorsy-friend-to-the-outdoors/**](http://blog.rei.com/camp/how-to-introduce-your-indoorsy-friend-to-the-outdoors/)

Output:

KeyWord:[friend]

KeyWord:[hike, time]

KeyWord:[rei, flat, great]

KeyWord:[outdoors, camp, introduce, short, camping, jennifer, travel, day, water, davis-flynn, activities]

KeyWord:[pace, easy, don’t, jazz, run, recreational, trail, conversation, films, she’s, indoorsy, overnight, stars, stewardship, climb, problem, kayaking, favorite, co-op, cycle, place, skiing, lead, equipment, friends, snow, paddle]

<http://www.cnn.com/2013/06/10/politics/edward-snowden-profile/>

Output:

KeyWord:[nsa]

KeyWord:[snowden]

KeyWord:[government]

KeyWord:[watched]

KeyWord:[videos, watch, replay]

KeyWord:[guardian, worked, privacy, hawaii, security]