**DOKUZ EYLUL UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CME2101 – PROJECT BASED LEARNING 3**

**A TEXT BASED SEARCH ENGINE**

**SOS**

**by**

**Abdulsamet İleri 2014510091**

**Özgür Hepsağ 2014510043**

**Seren Bolat 2014510013**

**November, 2016**

**İZMİR**

**Page**

**CONTENTS**

**INTRODUCTION 1**

**COMPLEMENT REPORT 2**

**CLASS DIAGRAMS 4**

**EXPLANATION OF CLASSES AND IMPLEMENTATION 5**

**CONCLUSION AND FUTURE WORK 6**

**REFERENCES 7**

INTRODUCTION

The main problem is to present most relevant documents for user queries in this way to save user’s time. SOS application is performing these functions quickly and effectively. These functions are introduced as File Operations, Indexing, Ranking and Sorting. Problems are solved using the main components as hashtable, data types, procedures and mathematical functions. In addition to these components, Object Oriented Programming is very helpful to develop SOS. Except programming problems, the project members problems are mentioned problems encountered chapter.

**Completion Report**

SOS project members are completely finished all problems. Milestones are completed with errorless.SOS project members solutions for

* FileOperations problems are solved using

BufferReader. All files are read only 35 seconds without mistake.

* Indexing problems are solved using hashtable

methods mention the following entries

* Chaining Method:

Hashtable-Size ==> 10.000

Total collusion ==> 311.014

Total elapsed time ==> 8 seconds.

* Rehashing are performed successfully using java.Math.BigInteger instance

to find the closest prime number to hashtable size.

* Open Addresing Methods with Rehashing:

*Linear Probing*:

Initial HashTable-Size ==> 10.000

HashTable-Size ==> 640.663

Total collusion ==> 920.171

Total Elapsed Time ==> 9 seconds.

*Quadratic Probing:*

Initial HashTable-Size ==> 10.000

HashTable-Size ==> 640.663

Total collusion ==> 803.669

Total Elapsed Time ==> 7 seconds.

*Double Hashing:*

Initial HashTable-Size ==> 10.000

HashTable-Size ==> 640.663

Total collusion ==> 770.059

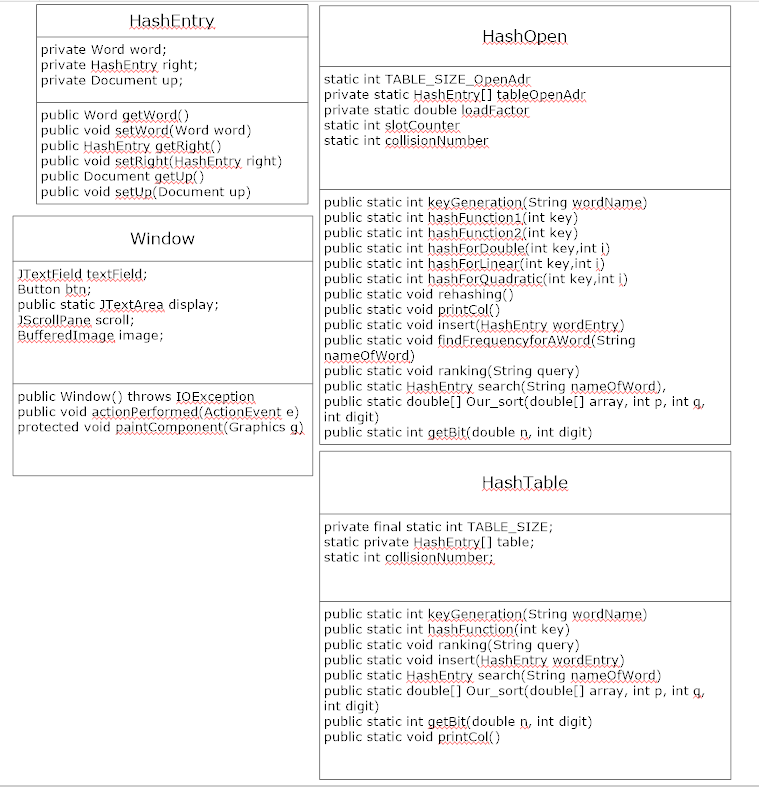
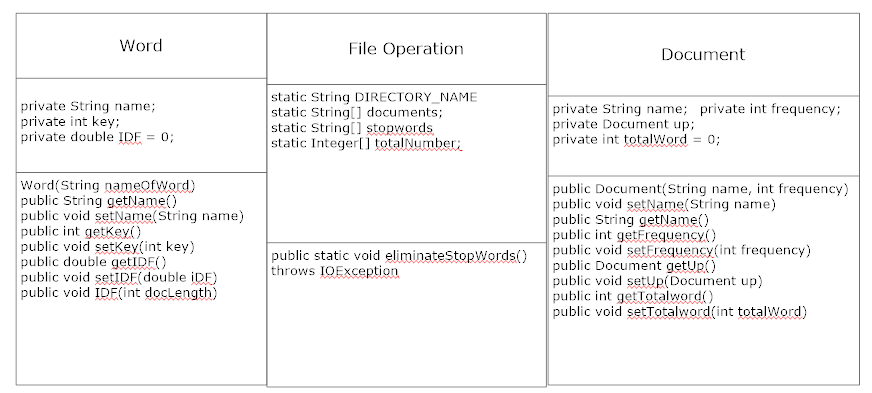
Total Elapsed Time ==> 6 seconds actually

* Ranking are solved using TF-IDF relation the given

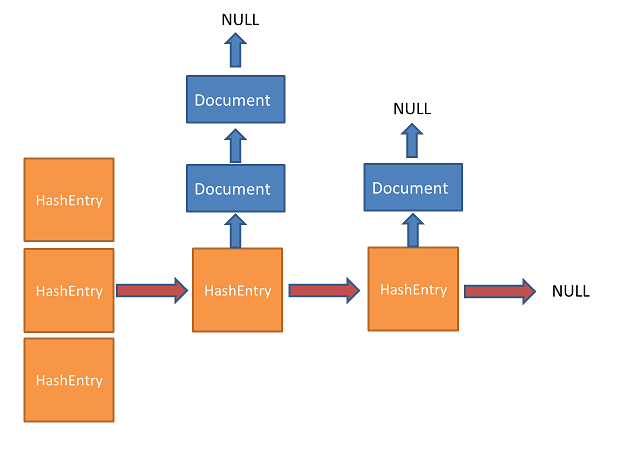
queries by user. Euclid relation is implemented on TF-IDF values and the results are made up of euclid relation and assigned user’s query terms.

* Sorting algorithm are implemented user’s query

term’s weight and sorted from large weight to small weight quickly and successfully, Thanks to sorting algorithm SOS application saves own user’s lifetime.



This is how SOS structure for indexing and storing.

****

****

**Conclusion and Future Work**

SOS will be obtained many documents for indexing and include useful tools like autosuggestion for queries.

References

[1] <https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/>

[2] <http://www.regular-expressions.info/quickstart.html>