

Software Architecture and Design

Project 1

KWIC Software Architecture for Web-based Search Engine

Document 1 Requirement Specification

**CS 6362 - Software Architecture
Dr. Lawrence Chung**

Team Tripod

Avinash Chiganmi
Phaneesh Kupphalli Lakshminarayana
Sameer Agrawal

Table of Contents

1. INTRODUCTION.....	3
2. REQUIREMENT SPECIFICATION	3
1.1 FUNCTIONAL REQUIREMENTS:.....	3
1.2 NON-FUNCTIONAL REQUIREMENTS:	3
3. CONCLUSION	4
REFERENCES.....	4

1. INTRODUCTION

The KWIC (Key Word In Context) index system provides an easy and efficient mechanism for searching based on the index by key words. For the project, the KWIC system shall accept an ordered set of lines, each line is an ordered set of words, each word is and ordered set of characters. Any line may be “circularly shifted” by repeatedly removing the first word and appending it to the end of the line. The system shall then output a listing of all circular shifts of all lines in alphabetical order. Later this system will be extended to be used for a web search engine.

The objective of this project is to design and build the KWIC software system to meet the functional and non-functional requirement specifications. This document describes the requirement specifications.

2. REQUIREMENT SPECIFICATION

1.1 Functional requirements:

- The KWIC (Key Word In Context) software system shall accept an ordered set of lines, where each line is an ordered set of words, and each word is an ordered set of characters.
- The user provides the input to the system through an input text box.
- Any line should be “circularly shifted” by repeatedly removing the first word and appending it at the end of the line.
- The KWIC index system shall output a list of all circular shifts of all lines in ascending alphabetical order.
- The output of the system will be shown to the user in an output text box, on clicking on a button.

1.2 Non-functional requirements:

- **Easily understandable** – the system should be easy to learn and understand
- **Portable** – the system should run on many platforms, browsers and operating systems
- **Enhanceable** – the system should allow for enhancement without major code rewrites or architectural changes

- **Reusable** – the components of the system should be reusable
- **Good performance** – the system should provide good performance on all platforms
- **User-friendly** – the system should be intuitive and easy to use
- **Responsive** – the system should respond to user actions quickly
- **Adaptive** – the system should be able to adapt to changes

3. CONCLUSION

This document describes the objective of the project and provides an overall view of the KWIC System. It also provides the functional specification that includes the functional and non-functional requirements for the system.

REFERENCES

- [1] Shaw, M, Garlan, D, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.
- [2] Technology For Reuse, “Case Study: Key Word In Context (KWIC) Product Line”, April 26, 2004, <http://fxvcl.sourceforge.net/kwicIndex.html>