

**CS6362.501 Software Architecture and Design**  
**Spring 2003**  
**Project 3: An Improved Web Search Engine**  
**Due: 2:00am Monday April 21**

**1. Project Summary**

As software architects, your team is to architect a web search engine, Microminer<sup>+</sup>, using KWIC<sup>++</sup> which is an extension to the KWIC<sup>+</sup> software system you have designed and implemented in Project 2. For this project, you will adopt a client-server architectural style, and build Java applications for both client and server.

**2. Microminer<sup>+</sup> - An improved web search engine**

**Functional requirements:** Microminer<sup>+</sup> shall contain two separate Java programs, Microclient and Microserver.

- Microserver shall use another software system, KWIC<sup>++</sup> - a KWIC (Key Word In Context) index system, in order to efficiently maintain a database of URLs and the corresponding descriptions. KWIC<sup>++</sup> shall accept an ordered set of lines, where each line consists of two parts: the descriptor part and the URL part:
  - The URL part, whose syntax is:  
URL ::= 'http:/' {identifier '.'}<sup>+</sup> ['edu' | 'com' | 'org' | 'net']  
identifier ::= {letter|digit}<sup>+</sup>  
letter ::= ['a' | 'b' | 'c' | ... | 'y' | 'z' | 'A' | 'B' | ... | 'Y' | 'Z']  
digit ::= ['1' | '2' | '3' | ... | '9' | '0']
  - The descriptor part, whose syntax is:  
{letter<sup>+</sup>}<sup>+</sup>.

The descriptor part of any line should be *circularly shifted* by repeatedly removing the first word and appending it at the end of the line. The KWIC<sup>++</sup> index system shall output a list of all circular shifts of all lines in ascending alphabetical order ' <a<A<b<B<...<y<Y<z<Z. No line in the output list shall contain any noise word such as "a", "an", "the", "and", "or", "of", "to", "be", "is", "in", "out", "by", "as", "at", "off". In addition, the output shall not have duplicate lines unless they are from different input lines.

- Microclient shall accept lines of keywords and return a list of descriptors and URLs whose descriptions contain any line of the given keywords in ascending alphabetical order, where ' <a<A<b<B<...<y<Y<z<Z. The order of the keywords in each line does not matter. No noise word shall be given as part of the list of keywords.

**Non-functional requirements:** The Microminer<sup>+</sup> system shall be easily understandable, portable, enhanceable and reusable with good performance. The Microminer<sup>+</sup> system must also be user-friendly, responsive, and adaptive.

**3. The Deliverable**

Your descriptions should be elegant and comprehensible. Your deliverable should be available as specifications:

- **The process architecture – management and organization of your team work:** describe how your team members were divided up in carrying out your own design tasks, and why each took the particular roles. In other words, describe the essential tasks as components of your own process architecture, their essential relationships as interactions, while taking into consideration other architectural concerns. In addition, describe how your team was organized and communicated with each other, e.g., hold meeting regularly, work closely together, or mostly independent, etc.
- **Architecture specification:**
  - a. Draw overall architecture diagrams and describe how the client-server architecture style incorporates with other styles you have used.
  - b. How the addition of functionalities may affect your architectures and designs? What are the options you have for each new function? Why do you choose the particular way to implement the function?
  - c. Try to use as many design patterns as possible in your systems. What design patterns have you used in your design? How are they used? Draw UML diagrams to show that. Why do you think they fit into your design?
- **Java program specification:** build Java programs and applets that should run with a web browser. Your program should be well documented and tested. You may choose to use sockets, Java RMI, or J2EE to implement your client-server architecture. (JSP has to be used with J2EE.)
- **User manual:** describe how the user can access and use the system. Your descriptions should include the addresses of your web sites where your applet (and all other deliverables) can be accessed. Also briefly describe the typical interactions between the user and the system, e.g., what are the steps the user has to follow in using the system. Use screenshots, if needed, to show how the system looks like initially as well as subsequent steps that the user takes to demonstrate the system behavior.

#### 4. Project Submission Policy

Projects must be submitted to WebCT before the deadline, which is 2:00AM on April 21. It is your responsibility to correctly submit the project to WebCT. A penalty of 30% will be deducted from your score for the first 24-hour period your project is late. A penalty of 70% will be deducted from your score for  $\geq$  24-hour period. No credit for  $\geq$  2 days. Weekend days will be counted.