

CS/SE 6362 – Advanced Software Architecture and Design Summer 2013

Project Phase I: KWIC Software Architecture

Due Date:- June 20 (Tuesday)[- (Thursday)] – Interim Project I submission & presentation
July 2 (Tuesday) – Final Project I submission

The important decisions in design are not what to put in but what to leave out. Attributed to Tony Hoare by Brinch Hansen

I. Summary

As system/software architects of a renowned company, your team is to architect a simple KWIC software system and implement it, which later will be used for a web search engine. For this project, you will use an Object-Oriented architectural style, and build a Java applet (or an equivalent), which should be accessible through your team's web page.

II. The KWIC System

II.1 Functional Requirements

The KWIC (Key Word in Context) index 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. Any line shall 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 listing of all circular shifts of all lines in ascending alphabetical order.

II.2 Non-Functional Requirements

The KWIC system shall be easily understandable, portable, enhanceable and reusable with good performance. The KWIC system shall also be user-friendly, responsive, and adaptable.

III. The Deliverable

Your description should be elegant and comprehensible. Your deliverable should be available as both on-line (one URL per team member) and offline specifications (submission of one copy per team). You can choose to use an (extended) IEEE-style format for the deliverable, in which the major sections typically include: Introduction, Main Body (items below, for this project), Glossary (Definitions and Acronyms) and References (See, for example, "Document Templates - general IEEE" on the Internet or course web site).

N.B. Your team's role play should be described in the updated Project Plan, among other things.

1. Requirements specification

The functional requirements specification is incomplete (e.g., where should the input come from, and the output go?). Describe any extensions, or clarifications, to the requirements specification. The non-functional requirements specification is also ambiguous. Clarify each non-functional term repeatedly as many times as you'd see necessary.

2. Architectural specification

Describe both pictorially and textually, the architectural style, components and. Your deliverable should also discuss the rationale in terms of the advantages and disadvantages of your architecture, in consideration of several scenarios whenever appropriate. Also describe all the constraints and patterns, if any. You should establish traceability between the requirements specification and the architectural design specification.

3. Specification of a Java applet or an equivalent, and A Prototype implementation

Your program specification, well documented and tested, and an implementation of a prototype system.

4. User Manual

A (preliminary) user manual should be developed, which should become more complete and consistent at the end of the 2nd phase of the project. Describe how the user can access and use the system, including the URL of each team's web site where your applet, or its equivalent (and all other deliverables), can be accessed. Also briefly describe essential scenarios --- 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 to show how the system looks like initially as well as to show subsequent steps that the user might take.