**SE4433/CMSC5433 Software Architecture and Design**

**KWIC Software Architecture for a Web-based Search Engine**

**Assignment 3 (Group)**

**1. Summary**

As software architects, your team is to architect a KWIC software system and implement it, which later will be used for a web search engine.

**2. The KWIC\* System**

**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 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, where “a<A<b<B<…<y<Y<z<Z”. The input of the system will be characters only, including uppercase and lowercase.

**Non-functional requirements:** The KWIC\* system shall be easily understandable, portable, scalable, and reusable with good performance. The KWIC\* system must also be user-friendly, responsive, and adaptive.

**3. The Deliverable**

Your descriptions should be elegant and comprehensible. The submission should include the following document and implementation.

• **Requirement specification:** the requirement specification is incomplete. Describe any extensions or clarifications to the requirement specification. Please use the UML **SEQUENCE** diagram to illustrate the requirements analysis.

• **Architecture specification:** You should use the pipe and filter architecture to implement the system.Please describe both pictorially (**COMPONENT** diagram) and textually the architectural style, including the components, connectors, any constrains and a discussion of advantages and disadvantages of the architecture.

• **Design specification:** Please use UML **CLASS** diagram to present your design. The design should be detailed enough so that unambiguous implementation can be achieved through the design.

• **Implementation specification:** your program should be well documented and tested.

• **User manual:** describe how the user can access and use the system. You should 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 for subsequent steps that the user takes.