**SE4433/CMSC5433 Software Architecture and Design**

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

**Assignment 2**

**1. Summary**

As software architects, your team is to architect a KWIC software system, which is an extension to the KWIC\* software system you have designed and implemented in Assignment 1. ***For this assignment, you will use the combination of shared data and OO architectural style***.

**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. ***No line in the output list shall start with any noise word such as “a”, “an”, “the”, “and”, “or”, “of”, “to”, “be”, “is”, “in”, “out”, “by”, “as”, “at”, “off”. You need to eliminate all these noise words in upper and lower cases.***

**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.

• ***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*.

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

• **Architecture specification:** You should use the combination of shared data and OO architecture style to implement the system.Please use UML ***component and deployment diagrams*** to describe both pictorially 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. The detailed design should be consistent with your architectural design.

• **Implementation specification:** your program should be well documented and tested. Since shared data architecture is used, data is not allowed to be duplicated.