# Cyber Miner KWIC Indexing System

**Final Phase I**

Yu Bai
Marco Mereu
Caroline Wigginton caw150730

https://www.utdallas.edu/~yxb121530/

# Table of Contents

# 1. Introduction

This section provides an overview of the information provided by this document by detailing the project's purpose, scope, and providing a list of definitions.

## 1.1 Purpose

The purpose of this document is to describe the requirements and architecture styles of a KWIC indexing system. This document also serves to further the development team's understanding of the problem domain and facilitate the development of said system.

## 1.2 Scope

Cyber Miner is a KWIC indexing system [….] Going to go into a lot of detail here

## 1.3 Definitions, Acronyms, and Abbreviations

- KWIC: Key Word in Context
- UML: Unified Modeling Language

## 1.4 Project Deliverables

- Phase I
  - Phase 1.1: Interim Project I
    - Deliverables: Preliminary Project Specification and Presentation
    - Due Date: June 14
    - Team Leader: Yu Bai
  - Phase 1.2: Final Part I
    - Deliverables: Finalized Project Specification and working indexing system
    - Due Date: June 25
    - Team Leader: Caroline Wigginton
- Phase II
  - Phase 2.1 Interim Project II
    - Deliverables: Project Specification and working search engine system
    - Due Date: July 12
    - Team Leader: Marco Mereu
  - Phase 2.2 Final Project
    - Deliverables: Presentation and system demo
    - Due Date: July 26
    - Team Leader: Caroline Wigginton

# 2. Process Architecture

This section describes the team composition and each team member's assigned roles. *Table 1* details the roles of each member.

| Team Member | Roles |
|---|---|
| Yu Bai | Architect<br>Requirements Engineer |
| Marco Mereu | Project Manager<br>Quality Assurance Engineer |
| Caroline Wigginton | Architect<br>Development Engineer |

*Table 1.* Team Roles.

- **Requirements Enginee**r: Responsible for gathering and defining the functional and nonfunctional requirements
- **Architect**: Responsible for delivering the system's conceptual architecture design for implementation
- **Development Engineer**: Responsible for software development
- **Quality Assurance Engineer**: Responsible for verifying the architectural design matches the requirements specification. This is a continuous process that ends only when the system is fully developed and shipped
- **Project Manager**: Responsible for keeping track of all the technical and team related development issues

## 3. Requirements Specification

This section describes the requirements and features of the Cyber Miner Indexing System to illustrate the system's processes and its actors to the reader.

### 3.1.1 Functional Requirements

- FR1.0 – The system shall provide a text file to accept an ordered set of lines.
- FR2.0 – The system shall accept an ordered set of lines.
    - FR2.1 – Each ordered set of lines shall be an ordered set of words.
    - FR2.2 – Each ordered set of words shall be an ordered set of characters.
- FR3.0 – The system shall circularly shift each line by repeatedly removing the first word and appending it to the end of the line.
- FR4.0 – The system shall output a list of all circular shifts of lines in ascending alphabetical order.

### 3.1.2 Non-Functional Requirements

- NFR1.0 – Modifiability: *consistently good in quality*
- NFR2.0 – Maintainability: *ease at which components can be added or replaced*
- NFR3.0 – Performance: *short response time*
- NFR4.0 – Reusability: *extent of possible component reuse*

- NFR5.0 – Intuitiveness: *easily understood*

# 4. Architectural Specification

4.1 Pipe and Filter Architectural Style

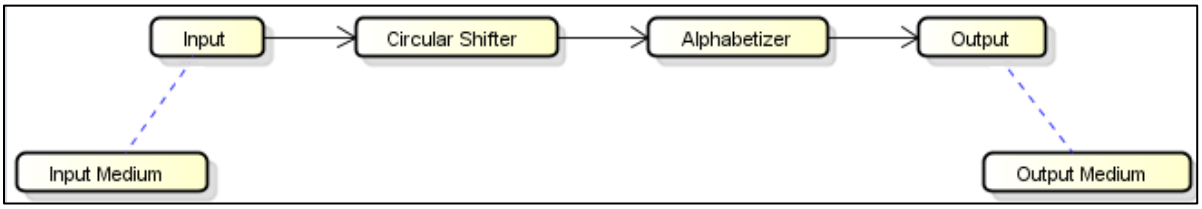


*Figure 1.* Pipe and Filter Architectural Style

4.2 Implicit Invocation Architectural Style

4.3 Shared Data Architectural Style

4.4 Abstract Data Type Architectural Style

### 4.5 Tradeoff Analysis

Four architectural styles in *Table 2* are rated from 1 to 5 based on five different nonfunctional requirements. Each nonfunctional requirement has a multiplier depending on the importance of that nonfunctional requirement to the system. The total score of each style varies between 5 (worst) and 75 (best).

| | **Shared Data** | **Abstract Data Type** | **Implicit Invocation** | **Pipe and Filter** |
|---|---|---|---|---|
| **Modifiability × 1** | 2 | 4 | 4 | 5 |
| **Maintainability × 3** | 3 | 4 | 5 | 3 |
| **Performance × 2** | 4 | 4 | 2 | 1 |
| **Reusability × 4** | 1 | 4 | 4 | 5 |
| **Intuitiveness × 5** | 2 | 4 | 3 | 4 |
| **Total** | **33** | **60** | **54** | **56** |

*Table 2.* Tradeoff Analysis.

# 5. Prototype Screenshots

This section shows sample input and output of the Cyber Miner KWIC Indexing System. *Figure 5* is a tested text file used as an input medium for the system. *Figure 6* is a list of the circularly shifted lines from the input in *Figure 5*. *Figure 7* is the circularly shifted lines from *Figure 6* alphabetized through the alphabetizer module.

```
hello$reading from the file$wow!$10/10$cool$sup dawg$A B C D$i like cats
```

*Figure 5.* Sample Input Text File.

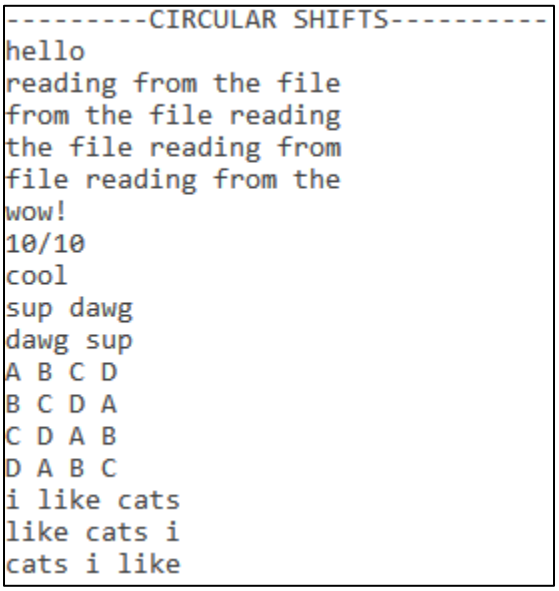


*Figure 6.* Circular Shift Output.

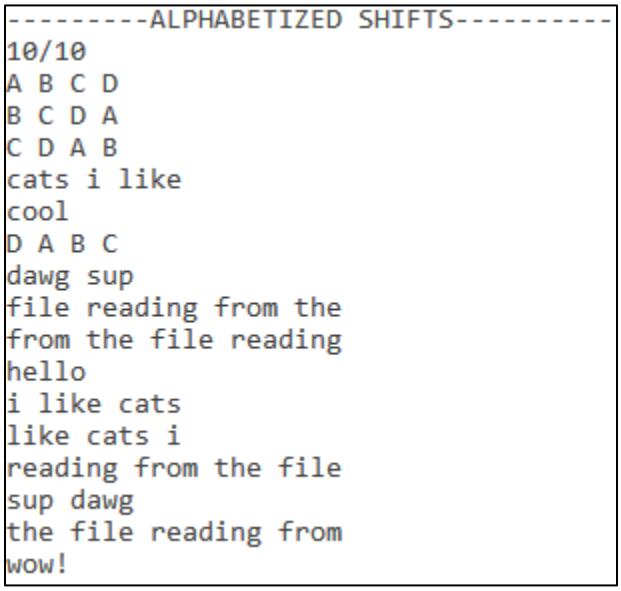


*Figure 7.* Alphabetizer Output.

## 6. User Manual

The source code for our KWIC Indexing System is available on our team website and can be run in Visual Studio with the .NET framework installed.

***Step 1.*** Open the text file and input a string with each line separated by a '$' delimiter as shown in *Figure 8.*

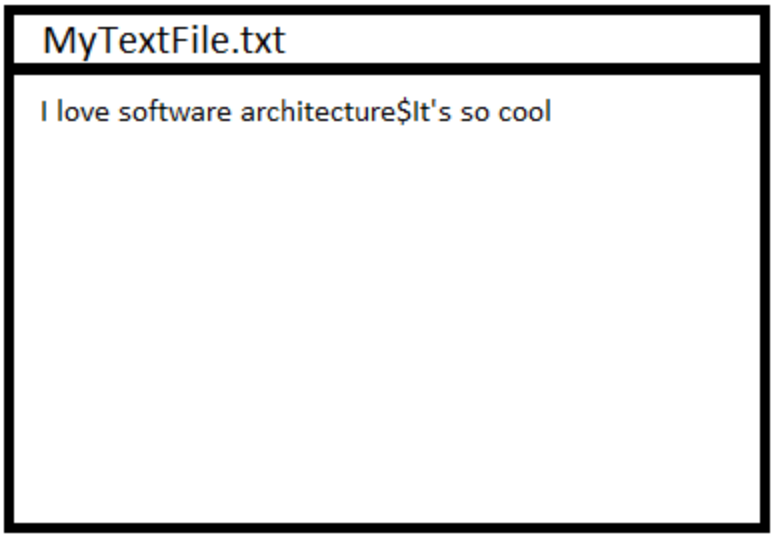


*Figure 8.* Text File Mock-up.

***Step 2.*** After entering the desired text, run the program by hitting F5 or pressing the Start button shown in *Figure 9.*


*Figure 9.* Visual Studio Start Option.

***Step 3.*** A console window will open displaying the circularly shifted and alphabetized lines as shown in *Figure 10*.

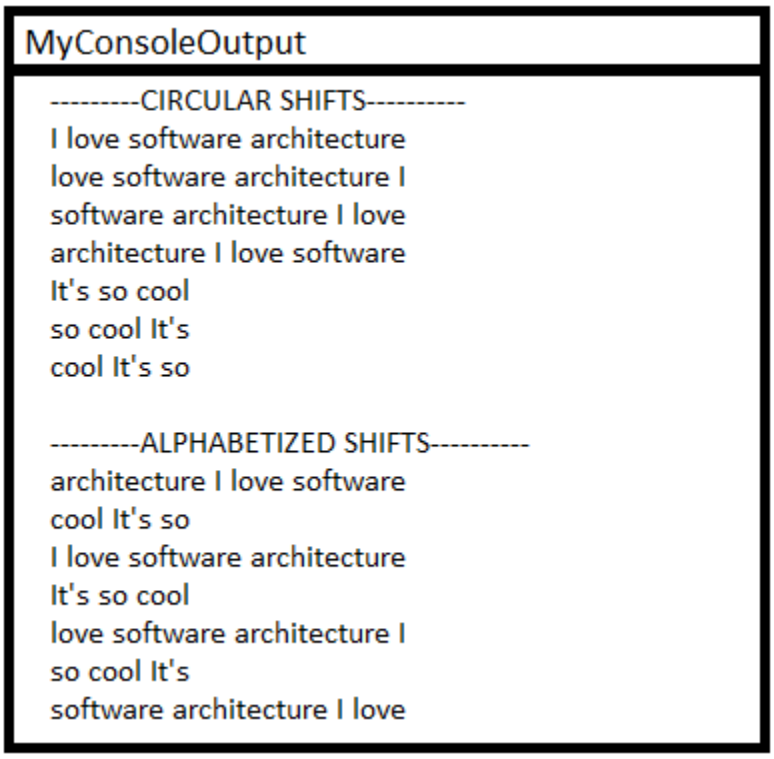

*Figure 10.* Console Output Mock-Up.

## 7. References

[1] Mary Shaw, David Garlan. “Software Architecture: Perspectives on an Emerging Discipline.” Upper Saddle River, New Jersey: Simon & Schuster, 1996.

[2] Course SE 4352 material from Professor Chung
https://www.utdallas.edu/~chung/CS4352/syllabus.htm