# Project Phase II: **CYBERMINER**

# Test Plan

*Versions 2.0*

CS/SE 6362 Advanced Software Architecture (Fall 2015)

**Submitted to:**

**Dr. Lawrence Chung,**

**Associate Professor,**

**Department of Computer Science,**

**The University of Texas at Dallas,**

**Richardson, TX -75080**

**Submitted By:**Team Name:  Quick Search

**Sruthi Chappidi,** [**sxc105920@utdallas.edu**](mailto:sxc105920@utdallas.edu)**,**

**Barbara Maweu,** [**bmk101020@utdallas.edu**](mailto:bmk101020@utdallas.edu)

**Maryellen Oltman,** [**mco130030@utdallas.edu**](mailto:mco130030@utdallas.edu)

**Twinkle Sharma,**[**txs151730@utdallas.edu**](mailto:txs151730@utdallas.edu)**.**

**Team Website:** [**www.utdallas.edu/~maryellen.oltman**](http://www.utdallas.edu/~maryellen.oltman)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| 11/24/2015 | 1.0 | Initial Draft | All |
| 11/26/2015 | 1.1 | Updates to Draft | Barbara Maweu |
| 11/30/2015 | 1.2 | Final Input | Barbara Maweu |
| 12/3/2015 | 2.0 | Final Submission | All |

**Table of Contents**

[**Project Phase II:** CYBERMINER](#h.gjdgxs)

[**TEST PLAN**](#h.30j0zll)

[1. Introduction](#h.3znysh7)

[2. Test Objectives](#h.2et92p0)

[3. Test Scope](#h.tyjcwt)

[4. Test Naming Scheme](#h.3dy6vkm)

[5. Features to be tested/not to be tested](#h.1t3h5sf)

[5.1. Features to be tested](#h.4d34og8)

[5.2. Features not to be tested](#h.2s8eyo1)

[6. Test Case Metrics](#h.17dp8vu)

[7. Test Cases](#h.3rdcrjn)

[7.1. FR1\_InputMethods](#h.26in1rg)

[7.2. FR1\_VariableInput](#h.lnxbz9)

[7.3. FR2\_IndexingComponent](#h.35nkun2)

[7.4. FR3\_InputValidation](#h.1ksv4uv)

[7.5. FR4\_ShiftEachLine](#h.44sinio)

[7.6. FR11\_AlphabetizedOutput](#h.2jxsxqh)

[7.7. NFR1\_EaseOfUnderstanding](#h.z337ya)

[7.8. NFR1\_EaseOfPortability](#h.3j2qqm3)

[8. References](#h.1y810tw)

## Introduction

This section describes the objectives and scope of the testing process that will be used by Team Quick Search. The goal of this document is to provide concise specification for performing tests on Cyberminer, the web search engine which utilizes the KWIC system to indexing search strings. This web-based system was developed in phases with the initial phase being the KWIC system and the second and final phase being Cyberminer.

This system will do the following:

* The system will provide the users with an input area for limited text input.
* The system will provide the users with a Search button to submit input text.
* The system will provide the users with a Report button to report broken URLs to the webmaster.
* The system will provide the users with an output area for displaying all possible results.

## Test Objectives

The primary focus of this plan is to ensure that the Cyberminer system satisfies the functional requirements and achieves the non-functional requirements documented in the Requirement specification document.

## Test Scope

The scope of the test cases is to validate the functionality of the system features and verification of the system’s quality attributes. To achieve the scope the category of test cases are detailed as follows:

Black box tests are derived from the functional requirements which were elicited during requirement analysis phase and are documented in the Requirement Specification document.

The system quality attribute tests cases are derived from the non-functional requirements are documented in the Requirement Specification document.

Integration tests are derived from Cyberminer documented in the Software Architecture document.

## Test Naming Scheme

Test case names are prefixed based on the domain that they represent. Reference the following prefix domain association:

* **NFR** - test cases derived from non-functional requirements
* **FR** - test cases derived from functional requirements

There will be an underscored number following the above prefixes if the test case is derived from a specific numbered requirement or design component otherwise a name identifier will be used. After the prefix and the number or name identifier, the test case name will be followed by a descriptive name representation.

## Features to be tested/not to be tested

The functional aspect of testing involves identifying all features and combinations of features to be tested. Features that are not included in this test plan will be documented and reasoning for not including them will be provided.

## Features to be tested

The following are the primary functionalities of Cyberminer system that are included in this test plan:

* The input area should be able to input phrases or text.
* A “Search” or similar functionality button should be provided to submit the input text.
* The results from the input textbox are displayed in the output area.
* The time required to search for the given input string is calculated and displayed and care is taken that it is not more than 5 seconds.

## Features not to be tested

Any third party software used to in the construction or required to use this system will not be included in this test plan. This includes the various browsers the users may use to access this application. The reason for not including them in the test plan is that they were evaluated for benefits and shortcoming prior to their choice.

## Test Case Metrics

The metrics to measure the success or failure of the test cases presented in this test plan are supplemented by a “pass” or “fail” notation. This general measure will be applied to the Cyberminer system as a measure of the following basic functionality

* Accept a set of lines which is then passed to the KWIC system for processing.
* KWIC (KeyWord in Context) index system accepts the ordered set of lines, where each line is an ordered set of words, and each word is an ordered set of characters.
* The KWIC system “circularly shifts" the ordered set of line by repeatedly removing the first word and appending it at the end of the line.
* The KWIC index system outputs a listing of all circular shifts of all lines in ascending alphabetical order.
* The results from KWIC system are passed back to Cyberminer system for processing of the search function based on a maintained index base.
* The Cyberminer system displays the matching searches results on the output display.

## Test Cases

This section, the core of the test plan, lists the test cases that are used during testing. Each test case will be described in detail.

## FR1\_InputMethods

* Test items: Input Textbox
* Input specifications: Use keyboard to enter text; Allow for input by copy/pasted.
* Output specifications: All circular-shifted lines.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None.
* Inter-case dependencies: None.
* Description: The user can input text into the search box via the keyboard and can also copy/paste stuff from/to the input area.

## FR1\_VariableInput

* Test items: Input text and phrases.
* Input specifications: Input text valid and invalid text.
* Output specifications: Output text should be filtered and presented line by line checking all the valid words are part of the output.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None.
* Inter-case dependencies: This test case depends on the dictionary of noise words and invalid characters.
* Example:
  + Input: This is the end
  + Output: The search results are based on a combination of valid text from the input text. viz. “This Is The End” movie

## FR2\_IndexingComponent

* Test items: Indices based on the input text and phrases.
* Input specifications: Input text valid and invalid words and phrases.
* Output specifications: An indexing system is referred to for the existing indices for the input text.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None.
* Inter-case dependencies: This test case depends on the KWIC system and the database of indices built on it.
* Description: The database that is built according to the indices processed from the KWIC system is referred for any search references to the valid text. Also, new indices are appropriately generated and maintained for new links input to the system and the same is an element of the result when a corresponding match for the index is found.

## FR3\_InputValidation

* Test items: Input text & phrases
* Input specifications: Input text containing valid and invalid words.
* Output specifications: Output text should be filtered and presented line by line without any noise words or invalid characters.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None.
* Inter-case dependencies: This test case depends on the dictionary of noise words and invalid characters.
* Description: The KWIC system shall accept an ordered set of lines, where each line consists of two parts:
  + The URL part, whose syntax is: URL ::= ‘http://’ identifier ‘.’ Identifier ‘.’ [‘edu’ | ‘com’ | ‘org’ | ‘net’]
  + identifier ::= {letter | digit}+
  + letter ::= [ ‘a’ | ‘b’ | … | ‘y’ | ‘z’ | ‘A’ | ‘B’ | … | ‘Y’ | ‘Z’]
  + digit ::= [‘1’ | ‘2’ | … | ’9’ | ‘0’]

The descriptor part, whose syntax is: identifier {‘ ‘ identifier}\*, of any line shall be “circularly shifted” by repeatedly removing the first word and appending it at the end of the line.

## FR4\_ShiftEachLine

* Test items: Input area and output area.
* Input specifications: Input more than one line with more than one word.
* Output specifications: Each line shifted should be output.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None.
* Inter-case dependencies: None.
* Example:
  + Input: This is the end
  + Output: This is the end

end This is the

the end this is

is the end This

## FR11\_AlphabetizedOutput

* Test items: Output area.
* Input specifications: Input text and phrases.
* Output specifications: Output all circular-shifted lines ordered by Alphabetizer.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None.
* Inter-case dependencies: None

## NFR1\_EaseOfUnderstanding

* Test items: Input area, circular shift area and output area.
* Input specifications: The input and output text font is readable; the system is presented by a proper color background.
* Output specifications: The text font is over 10; the system’s background color is different from the text color so that users can easily read what he/she typed and system output.
* Environmental needs: Web browser on client PC or Laptop.
* Special procedural requirements: None
* Inter-case dependencies: None

## NFR1\_EaseOfPortability

* Test items: Different OS and Web browsers.
* Input specifications: Input area, [Generate] button and output area.
* Output specifications: Readable and understandable input text and output text. Not have any CSS or JS errors.
* Environmental needs: PC or laptop with an OS installation of Windows, Linux or Mac and the following browsers: Chrome, Internet Explorer, Microsoft Edge or Mozilla
* Special procedural requirements: None.
* Inter-case dependencies: None.

## References

Dr. Lawrence Chung

Professor - CS/SE 6362 Advanced Software Architecture & Design – Fall 2015

Textbook

Mary Shaw & David Garlan. Software Architecture: Perspectives on an Emerging Discipline