

CS 6359 – Object-Oriented Analysis and Design Summer 2022

Project Phase I: Cyberminer – A web search engine –

MUST include: Use Case, Class, and Sequence Diagrams

Due Date:

- June 16 (Thursday) – **Interim Project I** submission (PPT) & Presentation
- June 30 (Thursday) – **Final Project I** submission

I. Summary

As system/software engineers of a renowned company, your team is to develop a simple web search engine, here called *Cyberminer*. For this project, you will use an Object-Oriented Analysis and Design, and build a prototype using an Object-Oriented Program, which should be accessible through your team's web page and/or run on your laptop.

Your 1st phase Cyberminer should evolve into your 2nd phase Cyberminer, which should be more complete than the 1st phase version.

II. The Cyberminer System

II.1 Functional Requirements

Cyberminer shall accept a list of keywords and return a list of URLs whose descriptions contain any of the given keywords.

Cyberminer shall allow for:

- *Case sensitive search*: The system shall store the input as given and retrieve the input also as such;
- *Hyperlink enforcement*: When the user clicks on the URL, which has been retrieved as the result of a query, the system shall take the user to the corresponding web site;
- *Specifying OR/AND/NOT Search*: A keyword-based search is usually an OR search, i.e., a search on any of the keywords given. The system shall allow the user to specify the mode of search, using "OR", "AND" or "NOT";
- *Multiple search engines*: to run concurrently;
- *Deletion of out-of-date URL*: and corresponding description from the database;
- *Listing of the query result in ascending alphabetical order, most frequently accessed order, or per payment*,

- *Setting the number of results to show per page, and navigation between pages;*
- *Autofill, while correcting typographical errors,*
- *Filtering out symbols that are not meaningful, according to the user configuration.*

II.2 Non-Functional Requirements

Cyberminer 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. Domain Modeling & Requirements specification

Your team should consider who the stakeholders of the system are, whether different kinds of stakeholders (e.g., organizational vs individual users, advertisers, and Cyberminer administrators) would need different kinds of features, etc.

The functional requirements specification is incomplete (e.g., where should the input come from, and the output go?). What and how inputs and outputs should be handled? 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) Design specification

Describe both pictorially and textually. Your team's deliverable should establish traceability between the requirements specification and the (architectural) design specification.

3. A Prototype implementation

Your program specification, well documented and tested, and an implementation of a prototype system. Your team should try to demonstrate that your Cyberminer works (*What would this mean?*)

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.