

Cyberminer – A web search engine

CS 6359.0U1 - Object-Oriented Analysis and Design - Su21

Preliminary Project Plan

Team Members Name:

Guo, Feifei/Wang, Qixiang/Wu, Ting-Wei/Zhang, Churong

(fxg180009/qxw200000/txw190006/cxz173430)

Leader: Wu, Ting-Wei

Team Document Website: <http://utdallas.edu/~qxw200000/OOAD/>

1. Introduction

This document will describe an overview of the project. It provides an overview of the project, project organization, and a preliminary solution to all 8 functional requirements.

1.1 Project overview

This project is developing a simple web search engine, here called Cyberminer. For this project, we will use an Object-Oriented Analysis and Design, and build an Object-Oriented Program, which would be accessible through our team's web page or run on our laptop.

Cyberminer, as a web search engine, shall accept a list of keywords and return a list of titles, descriptions and URLs, and the descriptions should contain some of the given keywords.

Cyberminer also shall allow for some other features such as: Case sensitive, 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has been retrieved as the result of a query, the system shall take the user to the corresponding web site; Deletion of out-of-date URL and corresponding description from the database.

It is also required that Cyberminer shall be easily understandable, portable, enhanceable and reusable with good performance. Cyberminer shall also be user-friendly, responsive, and adaptable. These form the non-functional requirements for the projects. The team is responsible for analyzing these requirements in order to detect any problems (including, but not limited to inconsistencies, ambiguities and incompleteness) that may exist with the requirements.

This document gives a preliminary plan for how to achieve the above stated aims. The first section gives an overview, describes project deliverables and itemizes the evolution of this document. Lastly the first section gives the meaning of acronyms that may be encountered in the rest of document and lists references from which guidelines have been drawn. In the second section, the organizational structure of the executing team is given. Technical Processes used are described in Section 4 while the fifth and final section details the work elements, meeting schedule for the project.

1.2 Project deliverables

The project is divided into two phases. Following is the project deliverable chart:

a. **PHASE 0:**

- Preliminary project Plan

b. **PHASE 1:**

- Requirement Specification
- Requirement Analysis
- Presentation
- Prototype

c. **PHASE 2:**

- Modified Implementation
- Modified Testing
- Presentation(if needed)

1.3 Evolution of this document

This project plan is a living document and will be evolved over time once a problem is refined and issues with requirement are resolved.

1.4 Definitions, acronyms, and abbreviations

LCS - longest common sequence

2. Requirements Analysis

2.1 Functional Requirements

1. Case sensitive search: The system shall store the input as given and retrieve the input also as such;

Store search history exactly as input.

2. 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;

Prepare some search results and links.

3. 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”;

After the user has entered the whole query, we need to check if there exists keyword like “OR/AND/NOT”, and split the query into 2 or more parts according to the number of keywords.

3.1 For ‘OR’ case: we need to search each part and combine the results.

3.2 For ‘AND’ case: we need to treat the query as a whole.

3.3 For ‘NOT’ case: we need to search each part, then use the result of ‘AND’/ ‘OR’ part subtract the ‘NOT’ part.

4. Multiple search engines: to run concurrently;

Let our search engine connect to the MySQL database, MySQL's inherent deadlock design will help us avoid concurrency issues.

5. Deletion of out-of-date URL: and corresponding description from the database;

Set up a timing mechanism to access the links in the database at regular intervals, like every 7 days, and remove the link if the return code of the link is not available.

6. Listing of the query result in ascending alphabetical order; most frequently accessed order, or per payment,

Do a simple sort and count.

7. Setting the number of results to show per page, and navigation between pages;

HTML related technology.

8. Autofill, while correcting typographical errors

8.1 First we need to an English word dictionary. We plan extract all the words from the Oxford Dictionary as our word dictionary.

8.2 Each time the user enters a space, the entered query will be split into words based on space. For each newly entered word that has not yet been entered a space, we use the LCS (longest common subsequence) algorithm for it, and we will find the word which is closest to what user has typed so far.

For example, user has already typed 'applr' in search box, for LCS, we will have 'appall' 66%, 'apple' 80%, and so on, we will display the autofill results in similarity order.

9. Filtering out symbols that are not meaningful, according to the user configuration.

This is similar to Google SafeSearch Filters, turning on SafeSearch helps hide some content you do not want to see. Here we allow users enter the set of words to be filtered out. And hide the results including those words.

2.2 Non-Functional Requirements

Easily understandable/User friendly

Issue 1: The interface should be easy to use.

Resolution:

We decided to use a graphical interface to make the user experience better

Portable

Issue 1: The Cyberminer system should be machine independent

Resolution:

We decided to use python as it is independent of any Operating system or platform.

Responsive

- The system should be interactive.
- Proper error messages should be displayed to enable the user understand the behavior of the system.
- In case of error, the system has to instruct the user to carry out corresponding steps to eliminate or detect the cause of error.

Issue 1: The input is not valid

Resolution: We decided to display warnings.

Reliable

- The system should behave the way the user expects.
- The behavior of the system should be consistent with the requirements specified.
- The Cyberminer will be released after thoroughly testing. Situations in which request and response run unexpectedly will be very minimal.

Performance

- The system should be implemented using an efficient algorithm, minimizing the time and space complexity.
- Response time of the system should be reasonably quick for any length and type of data.

2.3 Other Needs

To run the search engine, we need search data, thus we need to prepare a crawler to get some search input, result title, result description and result url. And the search input can be top 1000 search items of google.

3. PROJECT ORGANIZATION

3.1 Team Roles:

Our team is organized into 6 parts:

Project Leader

The project leader is responsible for scheduling meetings, overseeing meeting minutes, and keeping track of the progress of the deliverables. If the project is behind schedule, they are responsible for organizing emergency meetings and helping to complete the deliverables.

Software Architects

The diagram designer is in charge of designing the diagrams for the project documentation. This includes UML or diagrams for the prototype, and process diagrams for management plans.

Requirements Engineer

Requirements engineers help elicit requirements from the perspective of the implementation team. They also document the requirements in textual form.

Software Developers

Software developers will be involved with prototype creation and feasibility studies for requirements elicitation purposes.

Test Engineer

Testers will test and document the prototype's functionality.

End Users

Helps elicit requirements by approaching the situation from the perspective of a potential customer or user during requirements documentation group meetings.

3.2 Assumptions, dependencies, and constraints

Assumptions

Team members are assumed to have a working knowledge of basic Software Engineering concepts, such as UML and the incremental iterative evolutionary lifecycle model.

Dependencies

The prototype depends on the an improved understanding as well as refinement of the stated requirements, since refined requirements should exist before a prototype can be made.

Constraints

Team's available work times are constrained by individual student schedules, since this project is not taking place in a corporate work environment. The time for each iteration is constrained by the due dates assigned in class.

3.3 Risk management

No.	Risk	Monitoring and Controlling
1	Inappropriate version of the tools and components.	Select specific versions of tools and components to use and every member will adhere to the choice throughout the entire project.
2	Failure to meet deadlines for deliverable.	Setup milestones in advance of the final due date for each deliverable. If 2 or more such milestones are not met, a reassessment will be necessary in order to recover.
3	Unavailability of resources	Reassign resources to fill the gap, depending on availability and previous knowledge of what the currently unavailable resource was working on.
4	Accidental loss of valuable information	Most of the work will be done using online collaboration tools, Github, including version control software, and the resulting documents stored online to minimize the chances of loss of information. Copies of work not stored online will be kept by all team members.

3.4 Monitoring and controlling mechanisms

Project communication will be handled via Microsoft Teams. Weekly meetings along with meeting minutes are provided online for members to catch up on missed meetings or to review the project's progress.

4. Technical Process

4.1. Methods, tools, and techniques

Tools:

1. python
2. mysql
3. utd vpn
4. Microsoft Teams
5. Github

Techniques:

1. Web Crawler for search engine

4.2 Software documentation

UML diagrams will be provided to document the prototype's implementation of the requirements document.

The software user's manual shall be generated along with the functional requirements and be validated during the acceptance process.

4.3 Project support

Project communication will be handled via Microsoft Teams.

5.WORK ELEMENTS, MEETING SCHEDULE

5.1 Work Elements

Phase	Phase Deliverable	Start Date	End Date	Resources
Preliminaries	Preliminary Software Project Management Plan	05/29/2021	05/30/2021	Microsoft Word
Phase 1.1	Requirements & Architectural Specification	05/30/2021	06/13/2021	Microsoft Word Python
	Revised UML graphs Prototype Mid report	06/13/2021	06/17/2021	Microsoft Word PowerPoint Python
Phase 1.2	Revised Project Plan Revised UML graphs Complete more Functional Requirements	06/18/2021	06/22/2021	Microsoft Word Python
	Revised Cyberminer Requirements Architectural Specification Document Revised Cyberminer User Manual Report	06/23/2021	07/1/2021	Microsoft Word

Date	Meeting Task
05/29/2021	First discussion of project details
05/30/2021	Draft Preliminary Project Plan.
06/11/2021	Prepare for phase1.
06/13/2021	The crawler of data and the search data of the database is ready.
06/16/2021	Discuss the detail of autofill
06/22/2021	Update the UML graphs

06/25/2021	Update user manual and report
------------	-------------------------------