DeptOS

Department Official Website Search

Data Structure 112-1

第四組

111306079 徐偉哲

111306009 葉家妤

111306011 陳若庭

111306045 林俊聲

Table of Contents

1. Overview

We found that when using Google to search for academic departments, the official websites do not have an ideal priority in the search results. Therefore, we plan to address this issue by designing a new search engine.

This search engine is designed for users seeking the official websites of departments within Taiwanese universities. By modifying the keywords provided by the searcher, adjusting their respective weights, and implementing URL filtering, we aim to improve the prioritization of official websites.

Upon completion of this project, we are contemplating the inclusion of departments from foreign universities in our search results. Additionally, we plan to deploy this search engine to the public network in the future.

2. Background

2.1. Motivation

We found that when searching for specific academic departments without specifying a particular school, the results prioritize department introductions targeted at high school students. Similarly, when entering a school name without specifying a particular department, the official websites of departments do not appear in the top order.

We aim to address this issue through this project.

2.2. Project Objectives

Developing a search engine with a focus on prioritizing university department official websites. Include:

- search functionality
- search interface UI
- result interface UI

3. Project Scope

3.1. Features and Capabilities

- Local Search Engine:
  - Name: DeptOS Search
    - This name is a combination of the abbreviations for "Department" and "Official website." It signifies that this search engine is designed for querying the official websites of academic departments.
  - Function:
    - Input Keywords: Department Names, Academic Fields
    - Results: Relevant official websites of Taiwanese academic departments and programs

3.2. Exclusions

- At the same priority level, search results will not be arranged in a specific school order.
- Not supporting searching in English or other foreign languages
- Not including relevant departments from foreign universities.
- Not deploying the search engine to the public network.

3.3. Use case or Scenarios

- Target user:
  - People who want to find the official websites (OS) of departments
- Example: When entering department ‘a’
  output priority:
  1. OS of ‘a’
  2. OS of related credit programs
  3. OS of related departments

3.4. System Operation and Key Technology

- System operation:

  Operation based on Google Engine

I. User Input Keywords:
   - User-entered keywords are transmitted to the backend.

II. Backend Processing:
   - The system processes the input in the backend.
   - Adjusted keywords are input into the Google engine.

III. Capture Google's Returned Websites:
   - Capture the web pages returned by the Google engine.

IV. Ranking Websites:
   - The system calculates the keyword weights within the web pages.
   - The system filters out unnecessary URLs.

V. Send Calculated Results to the Frontend:
   - Transmit the calculated results to the frontend.

❖ Key Technology

➢ Front End: mainly written in HTML, CSS, javascript

■ UI design:
Output the CSS code after designing the interface style in Figma.

➢ Back End: mainly written in Java

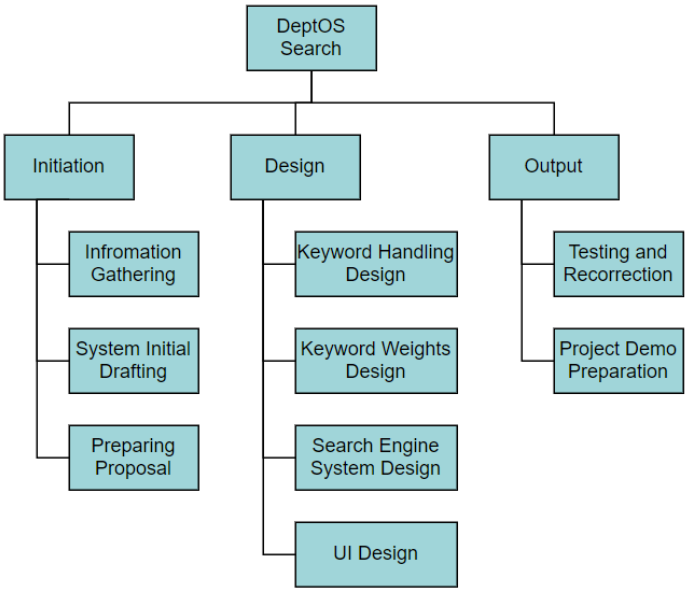
■ Ranking Websites:
Create a tree of the site and its subpages, and visit them to calculate the [number of keywords * weight], and finally add up to the site's rank.

■ Keyword weight adjusting:
Adjusting input keywords and other weights by method

■ Filtering URL:
Add points to the rank when the site has '.edu.tw' in the URL.

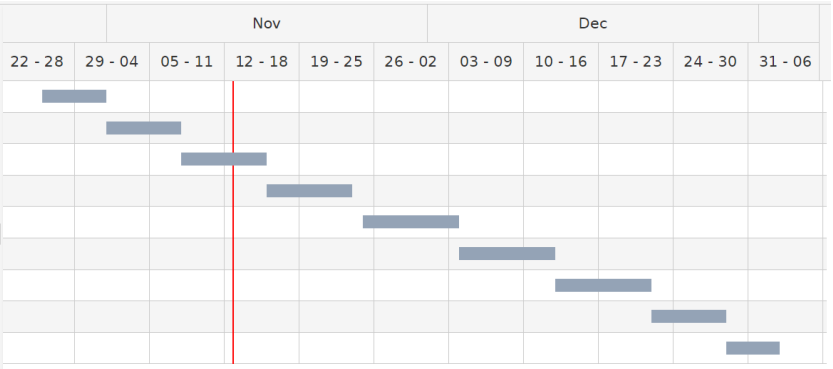
3.5. Development Schedule

- WBS



- Gantt Chart

ID	Title	Start Time	End Time
1	Information Gathering	10/26/2023	11/01/2023
2	System Initial Drafting	11/01/2023	11/08/2023
3	Preparing Proposal	11/08/2023	11/16/2023
4	Keyword Handling Design	11/16/2023	11/24/2023
5	Keyword Weights Design	11/25/2023	12/04/2023
6	Search Engine System Design	12/04/2023	12/13/2023
7	UI Design	12/13/2023	12/22/2023
8	Testing and Recorrection	12/22/2023	12/29/2023
9	Project Demo Preparation	12/29/2023	01/03/2024

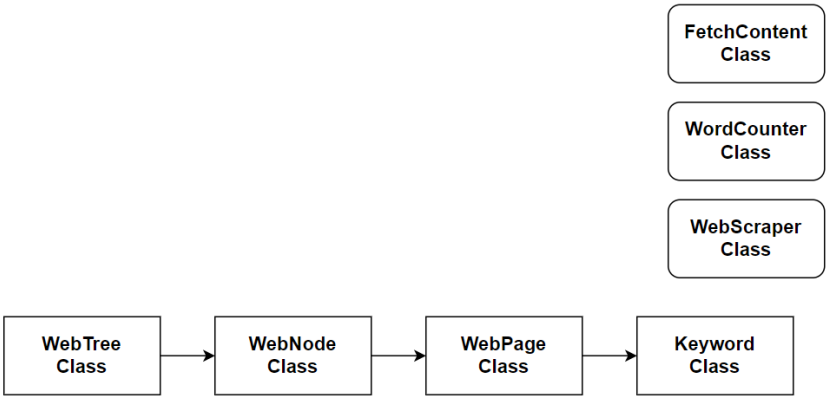


3.6. Future Development

- Support English-language searches
- Enhance search results with content from foreign universities
- Make the engine accessible on the public network

4. System design

4.1. Business Logic Layer and Object



4.2. Keyword

Keyword Class	
Parameters - name : String	//The name of the keyword. *private String name;*
- weight : double	//Record the weights of keywords. *private double weight;*
Methods - toString() : String	//Print out the results with specific formats. *public String toString(){}*

4.3. WebPage

WebPage Class	
Parameters - urlStr : String	//Record the website in String type. *private String urlString;*
- name : String	//The name of keywords. *private String name;*
- counter : WordCounter	//Record the number of times of keywords. *private WordCounter counter;*

- crawler : FetchContent - score : double	//Capture the content of websites. *private FetchContent crawler;* //Record number of times of a keyword. *private double score;*
Methods - setScore(ArrayList<Keyword> keywords) : void	//Record number of times of a keyword. *public void setScore(ArrayList<Keyword>keywords){}*

4.4. WebNode

WebNode Class	
Parameters - parent : WebNode - children : ArrayList<WebNode> - webPage : WebPage - nodeScore : double	//Record the parents of nodes. *private WebNode parent;* //Record the children of nodes. *private ArrayList<WebNode> children;* *private WebPage webPage;* //Record the score of the nodes. *private double nodeScore;*
Methods - setNodeScore(ArrayList<Keyword> keywords) : void - addChild(WebNode child) : void - isTheLastChild() : boolean	//Calculate the score of this node. *public void setNodeScore(ArrayList<Keyword> keywords)(){}* //Add the WebNode to its children list. *public void addChild(WebNode child){}* //Check whether it is the last web node.

WebNode Class	
	public boolean isTheLastChild(){}
- getDepth() : int	//Get the depth of the web node. *public int getDepth(){}*

4.5. WebTree

WebTree Class	
Parameters - root : WebNode	*private WebNode root;*
Methods - setPostOrderScore(ArrayList<Keyword> keywords) : void	//Calculate the score of every start node. *public void setPostOrderScore(ArrayList<Keyword> keywords){}*
- setPostOrderScore(WebNode startNode, ArrayList<Keyword> keywords) : void	*public void setPostOrderScore(WebNode startNode, ArrayList<Keyword> keywords)*
- rankNode() : void	//Put the nodes in order. *public void rankNode(){}*
- rankNode(WebNode startNode) : void	*public void rankNode(WebNode startNode){}*

4.6. FetchContent

FetchContent Class	
Parameters - urlStr : String	//Record the url of the website. *private String urlStr;*
- content : String	//Record the content of the website. *private String content;*
- url : URL	//Record the url of the website. *private URL url;*
Methods	

- getHtml() : String	//Get the content of the website; *public String getHtml(){}*

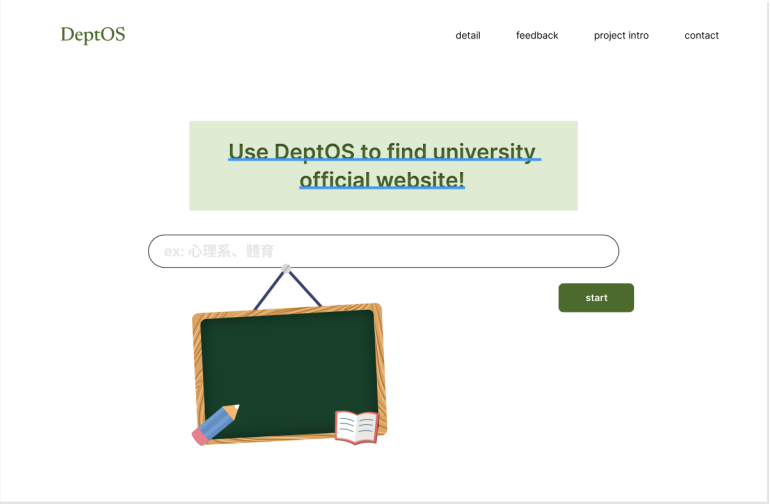
4.7. WordCounter

WordCounter Class	
Parameters - content : String	//The content of the website. *private String content.*
Methods - countKeyword(String keyword) : int	//Calculate the number of keywords in the website. *public int countKeyword(String keyword){}*

4.8. WebScraper

WebScraper Class	
Parameter - url : String	//Record the url of the website. *private String url;*
Methods - WebScraper(String url)	*public WebScraper(String url){}*

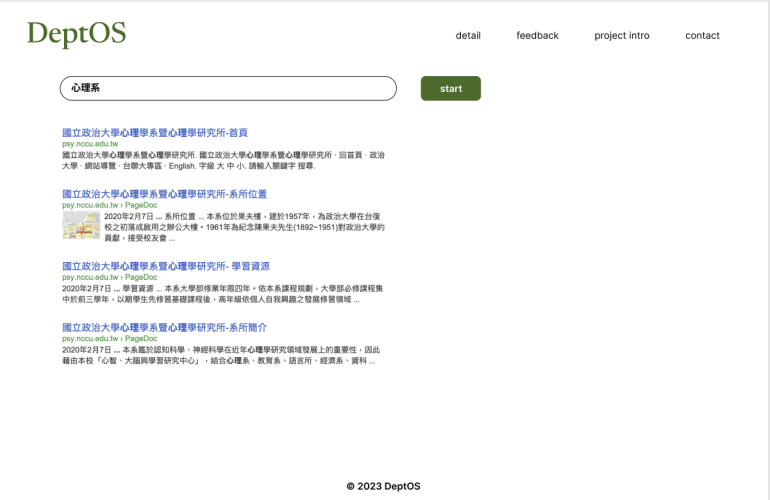
4.9. Results UI



Searching interface: The user's interface for searching.

```
...
"results": [
  {
    "cacheUrl": "http://www.google.com/search?q=cache:BWGyCdH003AJ:psy.nccu.edu.tw",
    "clicktrackUrl": "https://www.google.com/url?client=internal-element-
e&cx=d19c1ab4c3a914c6e&q=https://psy.nccu.edu.tw/PageStaffing%3Ffid%3D2874&sa=U&ved=2ahUKEwi6t6uopcCCAxVChlYBHVcUDGUQFnoECAAQAQ&usg=AOvVaw1pY1H5P4nFMm-
p0rDDba",
    "content": "師資介紹 &middot; 校內分機. 67394 &middot; 電子信箱. psyphh@nccu.edu.tw &middot; 學歷學位. 國立臺灣大學\u003cb\u003e心理\u003c/b\u003e學系暨研究
.",
    "contentNoFormatting": "師資介紹 · 校內分機. 67394 · 電子信箱. psyphh@nccu.edu.tw · 學歷學位. 國立臺灣大學心理學系暨研究所博士.",
    "title": "國立政治大學\u003cb\u003e心理\u003c/b\u003e學系暨\u003cb\u003e心理\u003c/b\u003e學研究所-師資介紹",
    "titleNoFormatting": "國立政治大學心理學系暨心理學研究所-師資介紹",
    "formattedUrl": "https://psy.nccu.edu.tw/PageStaffing?fid=2874",
    "unescapedUrl": "https://psy.nccu.edu.tw/PageStaffing?fid=2874",
    "url": "https://psy.nccu.edu.tw/PageStaffing%3Ffid%3D2874",
    "visibleUrl": "psy.nccu.edu.tw",
    "richSnippet": {
      "cseImage": {
        "src": "https://psy.nccu.edu.tw/upload/5/staffing/369/cache.54234_1ec1cdb3.jpg.w86_h120.jpg"
      },
      "metatags": {
        "viewport": "width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no",
        "ogTitle": "國立政治大學心理學系暨心理學研究所-師資介紹",
        "ogDescription": "  "
      },
      "cseThumbnail": {
        "src": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSa2RlAnVpPrwriQDasiChfHx9rZ7r4M9SyuwZO8fZ748zjBpqEcljEtQ",
        "width": "86",
        "height": "120"
      }
    },
    "breadcrumbUrl": {
      "host": "psy.nccu.edu.tw",
      "crumbs": [
        "PageStaffing"
      ]
    }
  },
  ...
```


Returned JSON File: The data in JSON returns after searching.



Result Interface: Showing the above results in the browser.