Final proposal

Group 20

Members：107206037 王姿予、108205003 林孜頤、

11135605 周曉林、110306021賴李仕翔、108305096 江彤恩

Contents

1.Introduction

Topic: Taiwan Mountain

Motivation:

As the pandemic changes the people's leisure preference, outdoor activities like hiking, camping, mountain climbing become progressively popular. Furthermore, the entry restrictions for foreigners to Taiwan in response to COVID-19 has been relaxed in recent months, and thus we could expect that inevitably increasingly mountaineers would like to visit Taiwan for the world-famous mountains. However, it would be a hard time for foreigners to get useful and correct information about Taiwan's mountains. For example, if we search esun (玉山) in Google, the first search result is esun bank. Therefore, we would like to devote ourselves to creating a reliable search engine for foreigners to collect accurate information about Taiwan's mountains.

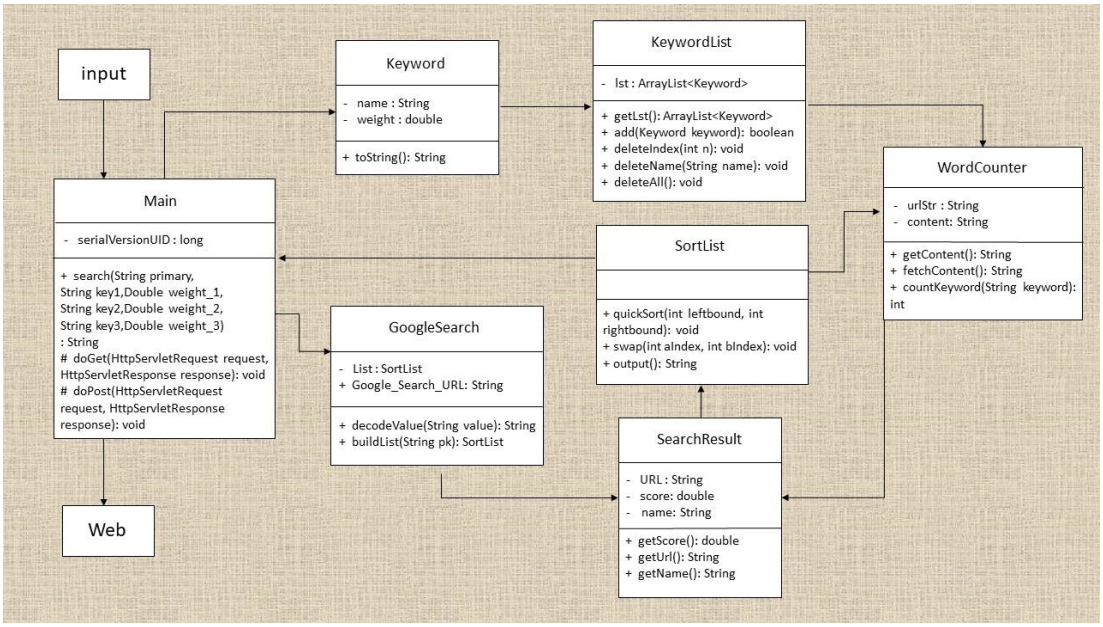
2.Search tricks(keywords)

segment	score	keywords
mountain-related	1	sunrise, kilometers, altitude, expedition, hiking, trails, trail, trekking, trek
level-related	3	100 peaks, difficulty level, peak, tallest mountain
prepared-related	5	permits, permit, alpinism, mountaineering, backpacking, stay

Since we assume that the information about preparation would be more valued to the user, we set keywords, like permits and stay, valuing 5 points. we expect that these 5-points keywords could return the web sharing the previous experience and living information. As for the others, we thought level-related and mountain-related keywords are different aspects of mountain climbing. Because of generalization of mountain-related keywords, we set its weight as 1 while level keywords may be more specific and thus set its weight higher, as 3.

3. System Design

(1)Class Diagram



In the beginning, Main class will read the input from the user. We expect our user to input 4 keywords, including primary keyword, important keyword, less important keyword and least important keyword. Next, the primary keyword will be used to run GoogleSearch. The result, which includes name and url, will be stored with a SearchResult object and added to SortList. After that, the WordCounter will calculate each URL score which was stored in the SortList with the given keywords. Finally, the system will store the result into SortList and execute sorting. After sorting, the result will be shown on our web.

(2)class

Google Search

Modifier and Type	Method(or Variable) and description
Instance Variable	
String	GOOGLE_SEARCH_URL be used to search keyword on Google.
SortList	List be used to store the result of Google search.
Instance Method	
String	decodeValue(String value) convert a string to object.
SortList	buildList(String pk) add the result which include name and URL to SortList.

Keyword

Modifier and Type	Method(or Variable) and description
Instance Variable	
String	name Keyword's name.
double	weight Keyword's weight.
Constructor	
Public KeywordList() Enable to initiate the object of Keyword with given name and weight.	
Instance Method	
String	toString() return a String in a specific format which includes name and weight.

KeywordList

Modifier and Type	Method(or Variable) and description
Instance Variable	
ArrayList<Keyword>	lst be used to store keywords' information.
Constructor	
Public KeywordList() Enable to initiate the object of KeywordList.	
Instance Method	
boolean	add(Keyword keyword) add keyword to lst, and return true.
void	deleteIndex(int n) remove the (n-1)th keyword in lst.
void	deleteName(String name) remove the specific keyword's name in lst.
void	deleteAll() remove all the keyword in lst.

SearchResult

Modifier and Type	Method(or Variable) and description
Instance Variable	
String	url the url of the webpage.
String	name the name of the webpage.
double	score the score of the webpage.
Constructor	
Public SearchResult(String url,String name,double score) Enable to initiate the object of SearchResult with given url, name and score.	
Public SearchResult(String url,String name) Enable to initiate the object of SearchResult with given url and name.	
Instance Method	
double	getScore() return score.
String	getUrl() return url..
String	gerName() return name.

SortList

Modifier and Type	Method(or Variable) and description
Instance Method	
void	sort() sort webs of the score by these webs.
void	quickSort(int leftbound, int rightbound) way to sort webs of the score.
void	swap(int aIndex, int bIndex) reset value to do quicksort.
String	output() print the result after sorting.

WordCounter

Modifier and Type	Method(or Variable) and description
Instance Variable	
String	urlstr the url of the webpage that we want to count keyword.
String	content the content of the webpage that we want to count keyword.
Constructor	
Public wordCounter(String urlStr) Enable to initiate the object of wordCounter with given url.	
Instance Method	
String	getContent() return webpage's content.
String	fetchContent() Fetch the content of the URL.
int	countKeyword(String keyword) using the content to count keyword in this URL.

4.Schedule

Task destribution		start	finish
1	project design	11/13	11/17
1.1	Content Elements	11/13	11/17
1.2	layout design	11/17	11/26
1.2.1	home page	11/17	11/20
1.2.2	searching page	11/20	11/23
1.2.3	result page	11/23	11/26
2	develope	11/17	12/6
2.1	back end	11/17	12/2
2.1.1	html handler	11/17	11/21
2.1.2	keyword	11/21	11/24
2.1.3	node	11/24	11/28
2.1.4	tree	11/28	12/2
2.2	front end	12/2	12/6
2.2.1	user interface	12/2	12/6
3	evaluation	01/03	01/03
3.1	project demo	01/05	01/05

5. Challenge

1) Frustration

–Less experience and knowledge

As Data Structure students, we don't have enough experience and knowledge about the project. Most of our members are not MIS majors nor have sufficient coding skills. As a result, it was difficult to assign specific tasks to team

members, wasting a lot of our time. Based on this unfamiliarity, we will also encounter many mistakes at the beginning. We spent a lot of time debugging, so the results we show are not the best. For example, we have encountered many problems in the development environment, but we often don't know where the problems are and how to deal with them.

–Coding ability
JavaEE is new to us, so it is more complicated to write codes and connect the front-end and the use of Jsoup, so it takes more time to try and understand how to work. Also, integrating previously written assignments is another problem because sometimes we don't really understand the methods.

–Time limited
We felt that time was too limited and the project started so late that everything that came out was a little underwhelming to perfectly represent our ideas. A few days before the presentation, the overall theme of the content changed, so the presentation didn't turn out as expected.

2) Exectation

–Interface
We wanted to pursue a more complete and understandable user interface, and we actually did pretty much everything we could. So we tried to correct some deficiencies. For example, our results show that the display of Chinese characters will appear garbled.

–Content
Our theme is mountaineering information, and the target audience is foreigners who come to Taiwan for mountaineering, but the search results may be different due to the order of keywords, which may lead to higher results than the desired information, so we are working hard Improve our systems and try to stop this from happening.

–Features
We would like to know if it is possible to use time as an argument. For example, when searching in winter, the search results will display the mountaineering information at that time to prevent people from entering the mountain by mistake during non-opening hours and causing danger.

6. Conclusion

First, we convert the method taught in class into code so that it can run. In this process, we need to understand the principle of each method so that we can sort it out. Second, we don't know much about JSP and Tomcat. We encountered the most problems in these two projects, and more than half of the team members did not know how to use and solve the problems. We also spent a lot of time asking classmates and teaching assistants, hoping to better understand the function of each piece of code, and hoping to solve the bugs we encountered. Lack of time was a big problem for us, we started too late so everything was rushed, luckily we ended up with results. In short, through this project and course, we have learned a lot about how search engines are formed and operated, as well as the knowledge and experience of doing projects. It is also very important to realize the allocation of time, division of labor, and use of everyone's strength.

7. Work distribution

周曉林	concept, front-end, back-end, github
林孜頤	concept, slide, presentation, logo
王姿予	concept, proposal
江彤恩	concept, back-end, proposal
賴李仕翔	concept, proposal