

## Find out the best café in Taipei City ○

尋找台北附近最棒的咖啡廳



Group20

Members: 108306093 戴寧、109306023 謝怡庭、109306019 劉承翰

### A. Introduction

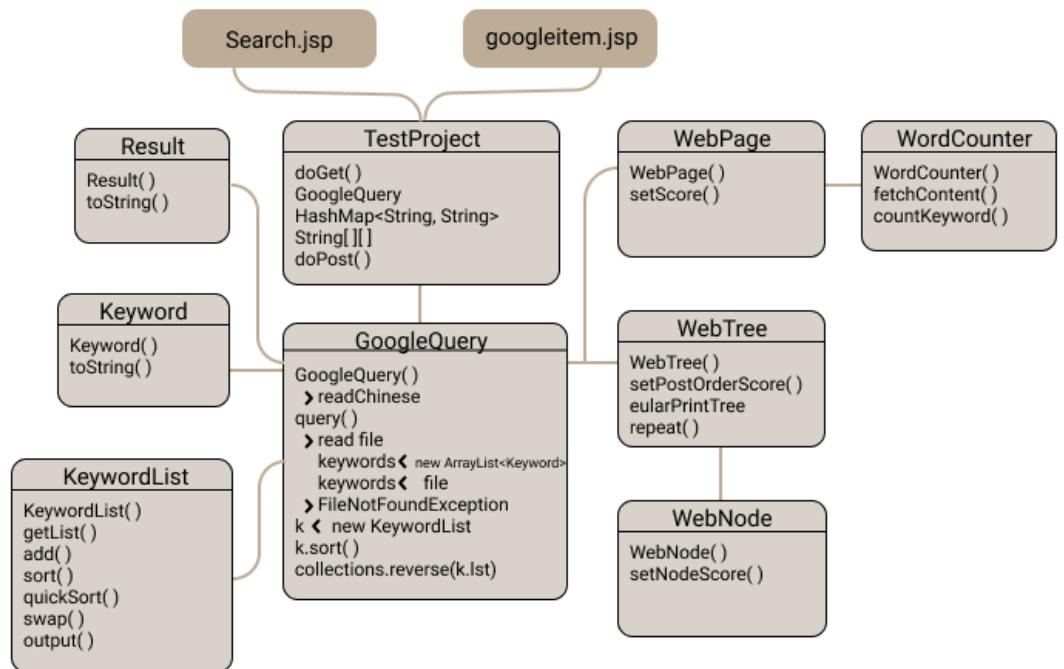
- a. Topic: Find out the best café in Taipei City
- b. TA: Students in Taipei city.
- c. Motivation: There is research that shows studying in a café could lead to an environment that makes you more concentrated and creative, so many students would choose studying in cafés rather than libraries. In addition to studying, many students go to cafés to relax, discuss their homework or reports. Since there are many kinds of cafés in Taipei city, students might go to the cafés that are not suitable for them, such as cafes that are really expensive, have limited time, or are too noisy. To solve this problem, our group wants to find out the most suitable cafe for students in Taipei city.

### B. Search tricks

keyw ord	咖啡 廳	台北	甜 點	不限 時	插 座	深夜咖 啡廳	網 美	安 靜	早午 餐	文山 區	蛋糕	戶外
weigh t	5	10	3.5	5	4	3.5	3.5	4	2	5	2	1

### C. System Specifications

- a. Users: Students in Taipei who want to go to the café studying or relaxing.
- b. How to use the system?  
Launch the search engine and enter the keyword relative with café and the best result will be represented on the result page.
- c. Class diagram



### TestProject Class

```

    < TestProject.java
      < TestProject
        serialVersionUID
        TestProject()
        doGet(HttpServletRequest, HttpServletResponse) : void
        doPost(HttpServletRequest, HttpServletResponse) : void
  
```

### GoogleQuery Class

```

    < GoogleQuery.java
      < GoogleQuery
        citeUrl2
        k
        {...}
        content
        searchKeyword
        title2
        url
        GoogleQuery()
        GoogleQuery(String)
        fetchContent() : String
        query() : HashMap<String, String>
  
```

### WebPage Class

```
▼ D WebPage.java
  ▼ C WebPage
    • counter
    • name
    • score
    • url
    • WebPage(String, String)
    • setScore(ArrayList<Keyword>) : void
```

#### WordCounter Class

```
▼ D WordCounter.java
  ▼ C WordCounter
    • content
    • urlStr
    • WordCounter(String)
    • countKeyword(String) : int
    • fetchContent() : String
```

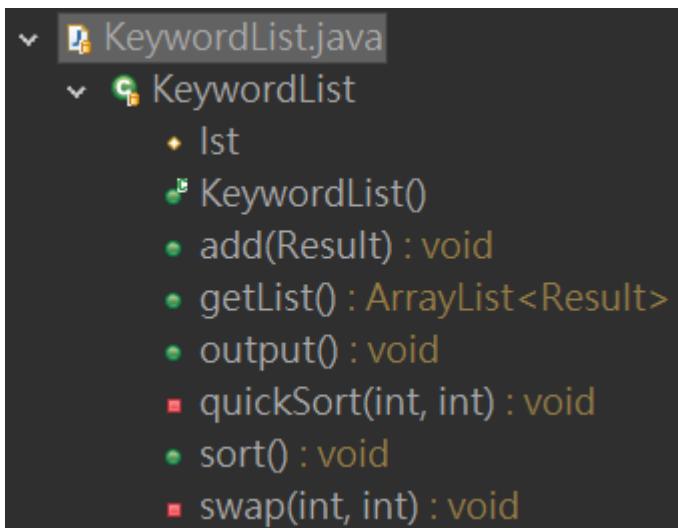
#### WebTree Class

```
▼ D > WebTree.java
  ▼ C > WebTree
    • result
    • root
    • WebTree(WebPage)
    • eularPrintTree() : void
    • eularPrintTree(WebNode) : void
    • repeat(String, int) : String
    • setPostOrderScore(ArrayList<Keyword>) : void
    • setPostOrderScore(WebNode, ArrayList<Keyword>) : void
```

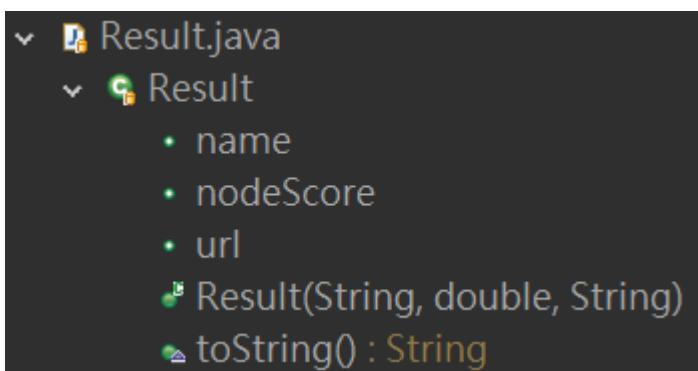
#### Keyword Class

```
▼ D Keyword.java
  ▼ C Keyword
    • name
    • weight
    • Keyword(String, double)
    • toString() : String
```

#### KeywordList Class



Result Class



Business Logic:

業務邏輯:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    // TODO Auto-generated method stub
    response.setCharacterEncoding("UTF-8");
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html");

    if(request.getParameter("keyword")== null) { // cat
        String requestUri = request.getRequestURI();
        request.setAttribute("requestUri", requestUri);
        request.getRequestDispatcher("Search.jsp").forward(request, response);
        return;
    }
```

1.

Launch the project from TestProject Class.

TestProject 開始進行doGet method

2.

Execute Search.jsp

執行Search.jsp

---



3.

Get the searching keyword from the inputBox.

從inputBox中獲得使用者輸入的關鍵字

---

```
public HashMap<String, String> query() throws IOException, MalformedURLException, FileNotFoundException
{
```

4.

Call the query method from GoogleQuery Class to implement querying and sorting functions.

呼叫GoogleQuery Class 的query method 進行排序

---

```
    citeUrl2 = li.select("a").get(0).attr("href").substring(7);
    title2 = li.select("a").get(0).select(".vvjwJb").text();
    if (title2.equals("")) {
        continue;
```

4-1.

Store the url and name respectively to citeUrl2 and title2.

爬蟲完搜尋到的網址及名稱分別存進citeUrl2, title2

---

```
    WebPage rootPage = new WebPage(citeUrl2, title2);
    WebTree tree = new WebTree(rootPage);
```

4-2.

Build the WebPage objects(citeUrl2, title2), Take these objects as the root of the WebTree.(These projects did not implement the searching subUrl function for the children of the tree. In the other words, each tree only exists a node — rootnode.

建立WebPage objects(citeUrl2, title2), 把此object當作WebTree中的root.(此專案未搜尋子網頁當作tree的childern, 即每個tree只有一個WebPage當作root.)

---

```

// fill in absolute path
File file = new File("C:/Users/User/git/DS_FinalProject/Webfinal/input.txt");
Scanner scanner = new Scanner(file);
//read 2 Yu 1.2 Fang 1.8
//Scanner scanner = new Scanner(System.in);
while(scanner.hasNextLine()){
    int numOfKeywords = scanner.nextInt(); //2

    // instantiate an arrayList to store Keyword objects(name, weight) scanned from input.txt.
    ArrayList<Keyword> keywords = new ArrayList<Keyword>();

    for(int i = 0; i < numOfKeywords; i++){
        String s = scanner.next(); //Yu

        // encoding for Chinese characters
        String name = new String(s.getBytes("GBK"), "UTF-8");

        double weight = scanner.nextDouble(); //1.2
        Keyword k = new Keyword(name, weight); //store key
        keywords.add(k);
    }
}

```

#### 4-3.

Use a Scanner object to scan keywords and weights into Keyword objects. Store Keyword objects into the arrayList("Keywords").

將input.txt中的keyword跟weight用scanner掃過存進Keyword objects, 再存進arrayList中。

```

tree.setPostOrderScore(keywords);
tree.eularPrintTree();

```

```

private void eularPrintTree(WebNode startNode){
    int nodeDepth = startNode.getDepth();

    if(nodeDepth > 1) System.out.print("\n" + repeat("\t", nodeDepth-1));

    //print "("
    System.out.print("(");
    //print "name", "score"
    System.out.print(startNode.nodeScore + ", " + startNode.webPage.name);

    // add node score to Result class in order to sort the scores
    WebTree.result = new Result(startNode.webPage.name, startNode.nodeScore, GoogleQuery.citeUrl2);
}

```

#### 4-4.

After calculating the website from each tree in the WebTree class, store the name, nodeScore and citeUrl2 parameters into the Result objects.

在WebTree中開始對每個tree裡的網址計算權重後, 在eularPrintTree method中把name, nodeScore, citeUrl2 存進Result的object中。

```

public class KeywordList {
    protected ArrayList<Result> lst;

    public static KeywordList finalSort;
}

finalSort.getList().add(WebTree.result);

```

#### 4-5.

Call the "lst" from KeywordList class to store Result objects.  
呼叫 KeywordList 的 lst 儲存 Result object.

---

```
finalSort.sort();
Collections.reverse(finalSort.lst);
    }
```

4-6.

Implement quicksort and reverse method to KeywordList in order to rerank the finalSort.lst

將 KeywordList 進行 quicksort, reverse 成權重由大到小排序的 list.

---

```
for(Result result:finalSort.lst) {
    retVal.put(result.name, result.url);
}
return retVal;
```

4-7.

Save the result's name and url into the hashMap, then return it to the query parameter from TestProject class.

將 KeywordList 排序好的 Result 的 name 跟 url 存進 HashMap 中並回傳給 TestProject 中的 query 參數.

---

```
GoogleQuery google = new GoogleQuery(request.getParameter("keyword")+"台北咖啡廳");
HashMap<String, String> query = google.query();

String[][] s = new String[query.size()][2];

request.setAttribute("query", s);
int num = 0;
for(Entry<String, String> entry : query.entrySet()) {
    String key = entry.getKey();
    String value = entry.getValue();
    s[num][0] = key;//title
    s[num][1] = value;//url
    num++;
}
// convert the KeywordList to double arrayList
KeywordList a = GoogleQuery.finalSort;
for(int i=0;i<a.lst.size();i++) {
    s[i][0] = a.lst.get(i).name;//title
    s[i][1] = a.lst.get(i).url;//url
}
request.getRequestDispatcher("googleitem.jsp")
    .forward(request, response);
```

5.

Store the value from the hashmap to the 2D arrayList.  
將 hashmap 中的 值 存進 2D arrayList.

---

【美食】推薦那些我去過的台北咖啡廳、早午餐、甜點店不限時適合 ...

\*2021台北咖啡廳整理\*氣氛好下午茶~不限時,適合久坐看書,有wifi,可 ...

大台北早午餐/下午茶/咖啡廳懶人包 (2021/12/28更新, 累計84間)

【台北咖啡廳】精選台北50間必去網美IG咖啡廳,下午茶

【台北】私藏必去的15間美好咖啡廳·偷渡超過100間wifi、不限時

精選台北25間必去美好咖啡廳、早午餐、下午茶網美餐廳、不限時

Top 10 台北最佳咖啡店 - TripAdvisor

文青老屋咖啡廳、下午茶甜點-台北

台北5間「異國風情咖啡廳」盤點！不出國也能享受在土耳其、古巴

6.

Demonstrate the result in the required sequence in googleItem.jsp.  
在googleItem.jsp中顯示排序好的結果

#### **D. System Developed Stage:**

☕ Stage 1. Using Keywords for page ranking

We provide relative keywords \* weights to rerank the website we searched. Input the .txt file with keywords \* weights and use the main class of HW6 to build an arraylist of these keywords.

☕ Stage 2. Read the sub URLs from the website and rank.

We have not implemented the function of reading the sub URLs from the website.

☕ Stage 3. Refine the rank of Google

Implementing the “GoogleQuery” class in HW10 to improve our searching engine. When calling the query() method, we create an array list of KeywordList to store the result, sorting and reversing the result to get the final sort sequence.

☕ Stage 4. Semantics Analysis

We skipped this stage for our searching engine.

☕ Stage 5. Publish the project online

Using tomcat8.5 to deploy our searching engine to the website.

#### **E. Schedule**

Task description	Start date	Finish date
Proposal	11/3	11/11
GUI Design	11/12	11/31

Class Design	11/12	11/31
Back-end	12/1	12/31
GUI	12/1	12/31
Project Demo	1/1	1/9

## F. Challenges

- Very few of the cafés have their own website, so we need to read many cafés' introductions through blog websites. It leads to the requirements of classification when doing the web crawler.
- How to present the website.