Data Structure Proposal

Group 5

Member : 109306005 楊鈺翎 109306024 陳姿穎 109306026 李慈琳

109306029 張曦勻 109306092 簡禎

Topic：Historical Figure

- Motivation

In the university, we have highly opportunity to select general courses related to history and must search for their relevant experience in lifetime to do a final project. However, we find out that the searching result generally is the restaurant, memorial hospitals, and various buildings nowadays named after these historical figures, isn't historical figures we expect. In order to tackle with the problem, we decide to improve it through learned knowledge we get in Data Structure course. Simultaneously, it make most students who would like to search for historical figures could find their literal needed information immediately as doing assignments.

- Example

Search field - 成吉思汗 / Expected - 成吉思汗(Yuan Dynasty Pioneer Emperor)

Unexpected - 成吉思汗健身俱樂部

Search field - Tesla / Expected - 尼古拉特斯拉(Nikola Tesla - Scientist)

Unexpected - 特斯拉科技(EV- Electric Vehicle)

Search field - 凱撒 / Expected - 凱撒大帝

Unexpected - 台北凱撒大飯店、凱撒衛浴

Search field - 馬偕 / Expected - 馬偕牧師

Unexpected - 馬偕紀念醫院

Search field - 忽必列 / Expected - 元朝軍事將領忽必烈

Unexpected - 忽必烈養生鍋物

Search field - 王陽明 / Expected - 王陽明 (Ming Dynasty Ideologist)

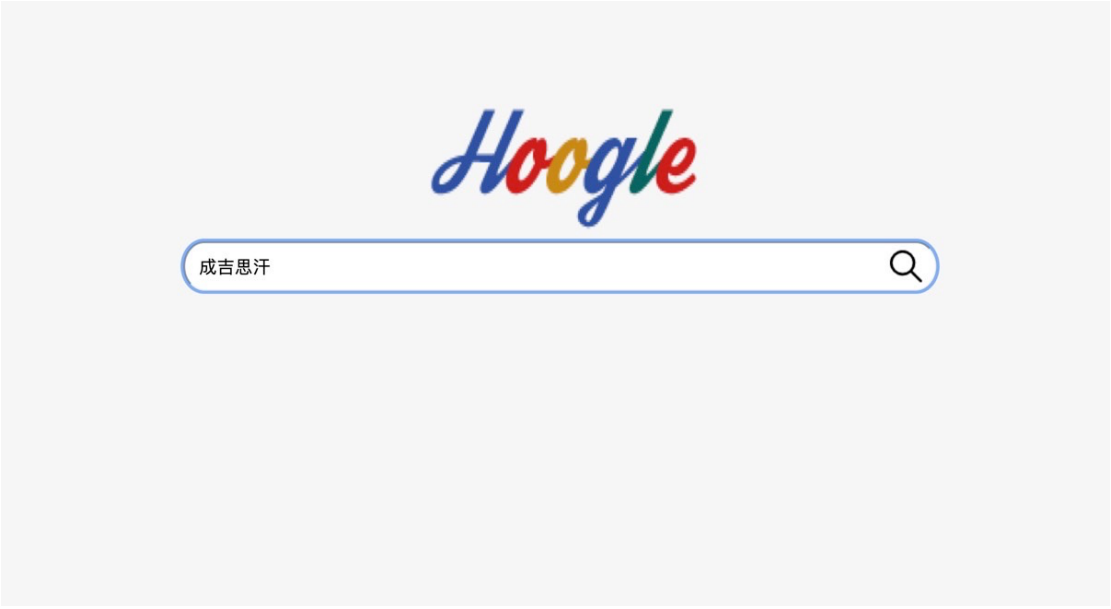
Unexpected - 王陽明 (an Entertainer)

Search field - Leonardo / Expected - 李奧納多達文西(Leonardo Da Vinci)

Unexpected - 李奧納多狄卡皮歐(Leonardo DiCaprio)

- Demo - Take「成吉思汗」as an instance

1. In「Search Field」, input the keyword「成吉思汗」



2. If we search for it on Google Search engine，the first one would be

「成吉思汗健身俱樂部」we aren’t expected.

3. If we search for it on our Hoogle Search Engine, the result would be expected 「成吉思汗」, who is pioneer emperor in Yuan dynasty.

- Search Website

1. Wikipedia

   https://zh.wikipedia.org/wiki/Wikipedia:%E9%A6%96%E9%A1%B5

2. Encyclopedia

   https://www.easyatm.com.tw/

- Search Keyword

Ø Bonus：

Twenty points：西元、歷史、炸藥、發明家、偉人、和平獎、化學、帝國、皇帝、戰爭、蒙娜麗莎、發明、最後的晚餐、文藝復興

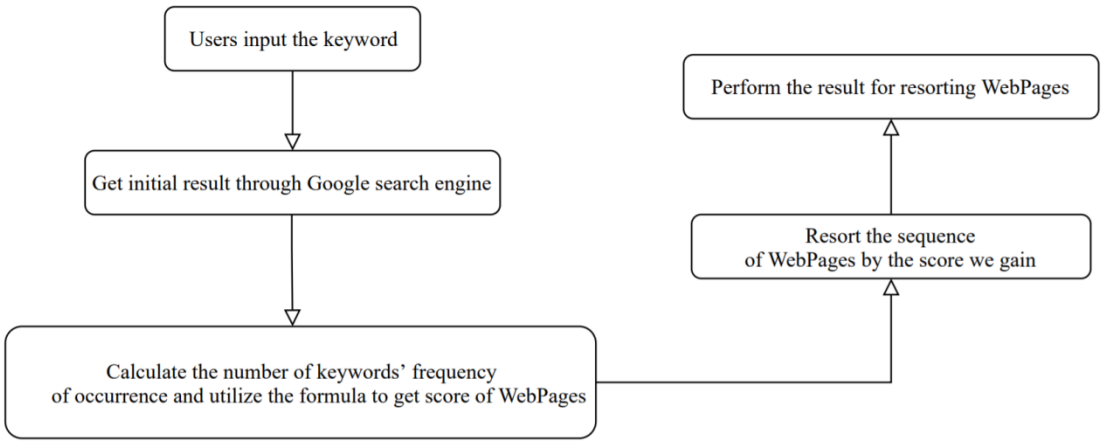
Ten points：出生、西元前、登基、統一、詩、詩人、貶謫、思想、朝

Ø Minus：

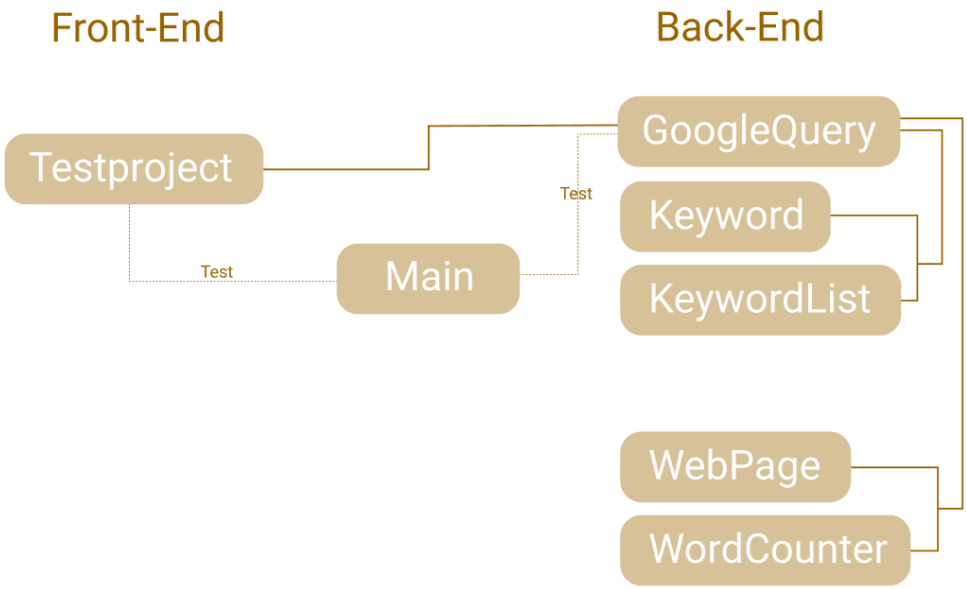
Ten points：藝人、演員、衛浴、飯店、球員、棒球、女友、緋聞、鐵達尼號、奧斯卡、奶凍捲、書局、眼科、電視劇、劇情、劇

formula：keyword count * weight

- System Design



- Class Diagram



- Class Explanation

Keyword Count	
Instance variable	
String	keyword (keyword)
int	count (keywords' frequency of occurrence)
double	weight (score of keyword)
Instance methods	
double	Score() Calculate the total score for a keyword， Score= count*weight

double	getScore() get score for a keyword

GoogleQuery	
Instance variable	
String	searchKeyword 使用者關鍵字
String	url
String	content 抓到的內容
HashMap <String,Integer>	scoreMap 還沒計算的分數 map
HashMap <String,Integer>	scoreSet 計算完成的分數 map
HashMap <String,String>	retVal 放 title 跟 url 的 map
Instance methods	
String	fetchGoogle() 取得 google 搜尋 content
HashMap <String,String>	query() 計算分數
HashMap <String,String>	getQuery() 可以 get 到在 query 列出的 retVal
HashMap <String,Integer>	scoremap() 可 getscoreSet 跟排列分數
List<Integer>	Sort(HashMap<Integer, String> score) 根據 Integer 排列出大小

Keyword	
Instance variable	
String	name 關鍵字
int	Count 權重
Instance methods	
String	toString

KeywordList	
Instance variable	
ArrayList<Keyword>	lst 儲存所有 keyword
Instance methods	
void	history() 來 add 內建所有關鍵字
ArrayList<Keyword>	getKeywordList()
void	add(Keyword k)
void	remove(Keyword k)

TestProject	
Instance variable	
Instance methods	
void	doGet (request, response)
void	doPost (request, response)

WordCounter	
Instance variable	
String	urlStr
String	content
Instance methods	
String	fetchContent() 去拿搜尋到的內容
void	countKeyword(String keyword) 算他有幾個 keyword

WebPage	
Instance variable	
String	url
String	name
WordCounter	counter
int	score
Instance methods	

void	setScore(ArrayList<Keyword> keywords)

- Schedule(WBS):

	Task Description	Start Date	Finish Date	Gantt Chart
1. Need Analysis 11/7 – 11/8	**1.1 Requirement Summary**	11/7	11/8	
	1.1.1 User Requirements	11/7	11/7	
	1.1.2 Reporting Requirements	11/8	11/8	
2. Design 11/9 – 11/25	**2.1 Front-End Layout**	11/9	11/25	
	2.1.1 Layout Design	11/9	11/11	
	2.2 Back-End Structure	11/12	11/25	
	2.2.1 Keyword Setup	11/12	11/18	
	2.2.2 Structure Design	11/19	11/25	
3. Development 11/26 – 12/29	**3.1 Layout Setup**	11/26	12/1	
	3.1.1 Searching Page	11/26	12/1	
	3.1.2 Result Page	11/26	12/1	
	3.2 Structure Construction	12/2	12/29	
	3.2.1 HTML Handler	12/2	12/8	
	3.2.2 Keyword	12/9	12/15	
	3.2.3 Google Query	12/16	12/29	
	3.2.4 Test Project	12/16	12/29	
4. Evaluation 12/30 – 1/13	**4.1 Program Testing**	12/30	1/4	
	4.2 Project Demonstration	1/5	1/11	
	4.3 Code and Report Uploading	1/12	1/13	

- Challenge

1. Character names entered in Chinese are the majority , however, the searching results are not the same as HW3.
2. Our theme is set as historical figures, regardless of any other countries, and the deeds of a certain character include life, illness, and death. As a result, we have to set more keywords to make the results more accurate and satisfy our needs.
3. I want to try to write a web page in a style that is compatible with all kinds of devices, so that you don't have to be tied to the computer device when querying.

- Expectation

Our primary target is to satisfy the need to obtain the literal needed information immediately as doing our assignments. After structuring our Search Engine “Hoogle”, we found out that “Hoogle” not only could be utilized on courses as

teaching tools by professors, but also be applied by students to do an effective research or survey through a precise search results.