

# BEAT GOOGLE - MOVIES



G14

108509315 徐嘉妤

109306032 林怡蕙

109306049 葉於芊

109306064 莊喻絜

1.

# MOTIVATION



Why does it start...

# MOTIVATION

 **Common interest :)**

 **Problem: results are not just about movies**

 **How to solve: create a prioritizing system**

2.

# KEYWORDS & WEIGHTS



The underlying components of the system

Keywords	Weights
電影	3
movie	3
film	3

Direct correlation: 3  
 High correlation: 2  
 Low correlation: 1

Keywords	Weights
影展	2
票房	2
爛番茄	2
大螢幕	2
cinema	2
theatre	2
film	2
IMDB	2
trailer	2

Keywords	Weights
好萊塢	1
奧斯卡	1
金馬獎	1
演	1
客串	1
劇情	1
soundtrack	1
director	1
stars	1

Words  
that means  
“movies”

Movie, film, drama

Trailer, blockbuster  
director, stars, cameo  
cinema, theater  
the big screen  
acting, soundtrack  
plot

Words that  
are related to  
“movies”

Awards,  
websites, etc.  
about movies

Hollywood  
Oscars,  
Golden Horse Awards  
IMDB,  
rotten tomatoes

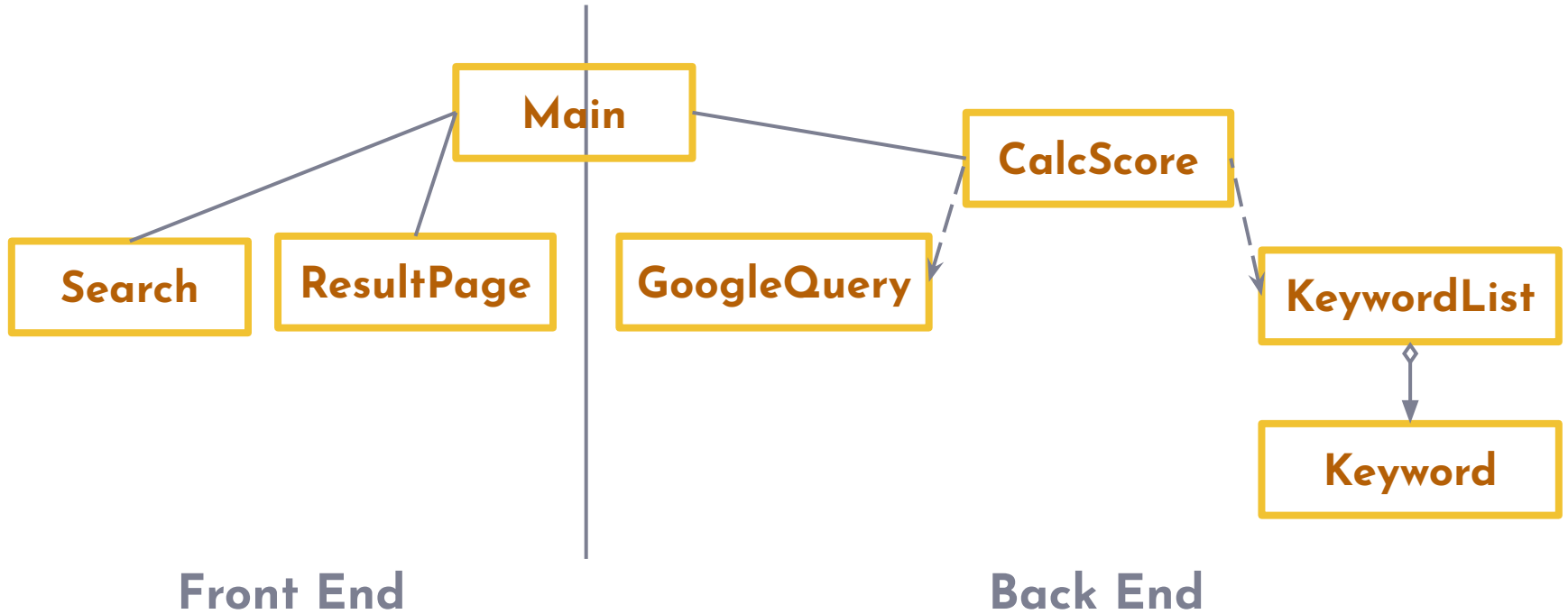
3.

# CLASS DIAGRAM



The system structure

# CLASS DIAGRAM





# CLASS DIAGRAM

## GoogleQuery

```
fetchContent(String): String  
fetchGoogle(String): String  
query(String): HashMap<String, String>
```

## CalcScore

```
countKeyword(String, String): int  
calcScore(): HashMap<Integer, ArrayList<String>>  
sort(HashMap<Integer, ArrayList<String>>): List<Integer>
```

## KeywordList

```
getKeywordList(): ArrayList<Keyword>  
filter(String): String  
movie(): void  
romance(); void  
fantasy(): void  
action(): void  
comedy(): void  
thriller(): void
```

4.

# TA & HIS/HER PURPOSE



who might be interested & why

# TA & PURPOSE

## TA



**Movie lovers**



**People who want to  
search for  
information related  
to movies**

## PURPOSE OF USING THE SYSTEM



**Easier to find the websites  
related to movies**

5.

# FUNCTIONS



what our system can do

& how to use it

# Categories



- 類別
- 愛情
- 奇幻
- 劇情
- 動作
- 喜劇
- 驚悚

請輸入關鍵字



# Categories

## Romance

Keywords	Weights	Keywords	Weights	Keywords	Weights
浪漫	15	娶	1	lover	1
愛	15	吻	1	couple	1
情	10	慾望	1	relationship	1
喜歡	10	romance	15	marry	1
戀人	1	romantic	15	marriage	1
兩性	1	affection	10	forever	1
婚	1	love	10	desire	1
嫁	1	sexual	1		

## Thriller

Keywords	Weights	Keywords	Weights	Keywords	Weights
懸疑	15	靈異	1	horror	15
驚悚	15	駭人	1	murder	1
恐怖	15	鬼片	1	detective	1
燒腦	1	顫慄	1	crime	1
推理	1	嚇	1	supernatural	1
謀殺	1	毛骨悚然	1	terrify	1
犯罪	1	thriller	15	horrify	1
殺人	1	suspense	15		

users only need to type what they want to search and  
click the button



# the system calculates the scores and shows the results on the result page

關於 [影評] 有 30 筆資料 (類型 **電影**)

影響影評網FilmCriticLog | 電影可以影響世界

【企劃】六十位台灣線上影評人的2020年度最佳電影

電影影評 - 闕小豪

柏C的電影雜記- 不專業的影評最直覺的觀點希望記錄下看完電影最 ...

你看過幾部？權威影評網站Indiewire公佈2018年25部必看神片！

網友滿意榜 - 奇摩電影

影評 - 奇摩電影

電影評論

關於 [影評] 有 30 筆資料 (類型 **懸疑/驚悚**)

影響影評網FilmCriticLog | 電影可以影響世界

如履的電影筆記- | 電影評論 | 推薦 | 情報 | 分析

《噩兆》影評：眼前的一片駭人陰森、逼你從頭到尾坐立難安 - 車庫娛樂

電影影評《真開心Happily》：一部令人捧腹的懸疑黑色喜劇

牠：簡單的故事架構，細緻的鋪陳 影評

「電影影評」現正上映電影與國內外熱門電影作品評論 - 電影神搜

首頁- 多多看電影-最新、最豐富的影視評論和新聞！

電影影評 - 闕小豪



6.

# TEST PLAN



Methods used for testing  
throughout the process

# TEST PLAN

- ✓ Walk-through
- ✓ Desk checking
- ✓ Unit testing
- ✓ Integration testing
- ✓ Performance test
- ✓ Alpha test



# UnitTest

```
Main.java  UnitTest.java x
1 import java.io.IOException;
2
3
4
5
6 public class UnitTest {
7     public static void main(String[] args) throws IOException {
8         System.out.println("Start testing.");
9
10
11         // searchKeywords: "電影", "adventure", "柯南", "林美秀", "2021", "資料結構", "郁力", "賽車", "Oscar", "tom
12         String[] inputs = { "影評" };
13
14         KeywordList lst = new KeywordList();
15         // filter options: romance, fantasy, action, comedy, thriller (default: "")
16         lst.filter("horror");
17
18         for (int i = 0; i < inputs.length; i++) {
19             String input = inputs[i];
20             GoogleQuery google = new GoogleQuery();
21             HashMap<String, String> query = google.query(input);
22             int query_before = query.size();
23
24             System.out.printf("\nCalculating score...\n");
25             CalcScore calc = new CalcScore(query, lst);
26             HashMap<Integer, ArrayList<String>> score = calc.calcScore();
27             List<Integer> order = calc.sort(score);
28
29             for (Integer integer : order) {
30                 System.out.printf("%s = %d\n", score.get(integer), integer);
31             }
32             System.out.printf("\nQuery_before size: %d\n", query_before);
33             System.out.println("Query_after size: " + query.size());
34             System.out.println("filter: horror");
35             System.out.println();
36         }
37         System.out.println("End of testing.");
38     }
39 }
```

1.

# SCHEDULE



How the system was gradually formed...  
and the progress of the whole project

# SCHEDULE

	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18
Project Initiation	10/18												
Structure Design		10/29			11/15								
Back End Coding										12/25	1/1		
Front End Coding											1/2		1/10
Testing & Debugging											1/6		1/10
Report												1/6	1/11
Presentation Preparation													1/12~ 1/13
Demo													1/13

8.

# WORK DISTRIBUTION



how we cooperate :)

徐嘉妤

Page rank  
Site rank

莊喻絜

Web Page

林怡蕙

Call Google  
Related keywords

葉於芊

PPT  
Final Report

9.

# PROBLEMS ENCOUNTERED & SOLUTIONS



what we had trouble dealing with  
&  
how we solved them



# PROBLEMS ENCOUNTERED



**Connect front end with back end**



**Convert final result into a webpage**



**Results not necessarily ranked according to categories**

# PROBLEMS ENCOUNTERED



## Servlet response 400/403/404 bad request

### HTTP Status 400 – Bad Request

**Type** Exception Report

**Message** Request header is too large

**Description** The server cannot or will not process the request due to something that is perceived to be a client error

**Exception**

### HTTP Status 403 – Forbidden

**Type** Status Report

**Message** Could not verify the provided CSRF token because your session was not found.

**Description** The server understood the request but refuses to **authorize** it.

Apache Tomcat (TomEE)/8.5.41 (7.0.6)

# PROBLEMS ENCOUNTERED & SOLUTIONS



## Servlet response 400 & 403 bad request

```
// search google
String url = "http://www.google.com/search?q=" + searchKeyword + "&oe=utf8&num=30";
System.out.println("url: " + url);

String content = fetchGoogle(url);
HashMap<String, String> retVal = new HashMap<String, String>();

Document doc = Jsoup.parse(content);
Elements lis = doc.select("div");
lis = lis.select(".kCrVT");

for (Element li : lis) {
    try {
        // parse down URL link
        String citeUrl = li.select("a").attr("href"); // System.out.println("origin: " + citeUrl);
        if (citeUrl.startsWith("/url?q=")) {
            citeUrl = citeUrl.replace("/url?q=", "");
        }
        String[] splittedString = citeUrl.split("&sa=");
        if (splittedString.length > 1) {
            citeUrl = splittedString[0];
        }

        // url decoding from UTF-8
        citeUrl = java.net.URLDecoder.decode(citeUrl, StandardCharsets.UTF_8);
        citeUrl.replaceAll(" ", "%20");
    }
}
```

```
public String fetchContent(String url) throws IOException {
    String retVal = "";

    try {
        URL u = new URL(url);

        HttpURLConnection.setFollowRedirects(false);
        HttpURLConnection conn = (HttpURLConnection) u.openConnection();
        conn.setRequestProperty("Accept", "*/*");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("User-Agent",
            "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Chrome/96.0.4664.110 Safari/15.2");
        CookieHandler.setDefault(new CookieManager());

        if (conn.getResponseCode() == 403 || conn.getResponseCode() == 400 || conn.getResponseCode() == 404) {
            retVal = url;
            System.out.printf("Error %d: %s\n", conn.getResponseCode(), retVal);
        } else {
            InputStream in = conn.getInputStream();
            retVal = new String(in.readAllBytes(), StandardCharsets.UTF_8); // will fail with large amounts of data
        }
    } catch (MalformedURLException e) {
        e.printStackTrace();
        retVal = url;
    } catch (IOException e) {
        e.printStackTrace();
        retVal = url;
    }
}
```

# PROBLEMS ENCOUNTERED & SOLUTIONS



## Java.IO.Exception Premature EOF

```
Exception in thread "main" java.security.InvalidKeyException: IOException: Detect premature EOF
content={"code":"10000","msg":"Success","out_trade_no":"20191009681975330","qr_code":"https://qr.alipay.com/bax08876xtkegkpv91xg50d5"}, charset=utf-8, publicKey=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsH3BzT1SOGRniX92H0w1JmhGBaYb7k/cqakd6mk6pf919zoKLPb9uh91m7eccP/t9YWzxF3Bwvsc1XEux/bXrRCSVoHZPKonHvQwwynzAPoRjpBvZP8/1200gm/yCB9ytkKDydwxCkc9dHAFuVcsA19v9iGqyS4E28P9cXjWn08Lh4BzP26a2bY0M1GC1iHPgNTDz1nSdH6eTGIKFJKyI9rPxQefio/nK7H1dlccHcmOuU8LeYDqXBWAwt54JOMQ75Q2LnYV41+XEGxTlwjhb/nhd16oufqblCwko5dcmg74rXYCGZQWyLRvADg
```

```
if (conn.getResponseCode() == 403 || conn.getResponseCode() == 400 || conn.getResponseCode() == 404) {
    retVal = url;
    System.out.printf("Error %d: %s\n", conn.getResponseCode(), retVal);
} else {
    InputStream in = conn.getInputStream();
    retVal = new String(in.readAllBytes(), StandardCharsets.UTF_8); // will fail with large amounts of data
}
} catch (MalformedURLException e) {
    e.printStackTrace();
    retVal = url;
} catch (IOException e) {
    e.printStackTrace();
    retVal = url;
}
```

# PROBLEMS ENCOUNTERED & SOLUTIONS



## Server redirected too many times (20)

```
at https://www.google.com/
Exception in thread "main" java.net.ProtocolException: Server redirected too many times (20)
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
    at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(Unknown Source)
    at java.net.URL.openStream(Unknown Source)
    at URLReaderWithOptions.main(URLReaderWithOptions.java:58)
```

```
public String fetchContent(String url) throws IOException {
    String retVal = "";

    try {
        URL u = new URL(url);

        HttpURLConnection.setFollowRedirects(false);
        HttpURLConnection conn = (HttpURLConnection) u.openConnection();
        conn.setRequestProperty("Accept", "*/*");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("User-Agent",
            "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Chrome/96.0.4664.110 Safari/15.2");
        CookieHandler.setDefault(new CookieManager());
    }
}
```

10.

# FUTURE EXPECTATIONS



What we hope to achieve

# FUTURE EXPECTATIONS



**More keywords** (to be more accurate)



**More categories**



**Other types besides movies, e.g. TV series**



**Other languages**

THANK YOU  
FOR  
LISTENING  
:)

