

Introduction to Database Systems

Term Project: Report

0716214 江岳勳

0716222 黃偉傑

1 Data

1.1 Introduction

我們的資料以公司資訊、職缺資訊、地域、職缺類別、使用者資訊和最愛組成。其中公司資訊、職缺資訊、地域和職缺類別，是我們從 [1111 人力銀行](#) 網站爬蟲下來所得。而使用者資訊和最愛，是在 application 運行時，由使用者執行相對應的操作之後，執行 insert 指令所得。

1.2 Description of the tables

首先以一個表格來呈現資料之於 table 的構成。

Table	Columns	Primary Key	Description
jobinfo	job_id , job_name, degree, low_salary, high_salary, exp_year, job_type, worktime, is_night, needed_num	job_id	職缺資訊
company	com_id , com_name, capital, emp_number, addr	com_id	公司資訊
position	pos_id , pos_field, pos_name	pos_id	職缺的職務與領域類別
localarea	area_id , area_cc_name, area_td_name, area_cctd_name	area_id	臺灣各行政區
job	job_id , com_id, pos_id, area_id	job_id	職缺與公司、職缺類別、地域之間的關係
user	username , passwd	username	使用者資訊
favorite	username, job_id	N/A	使用者加入最愛

以下每個表格分別代表著一個 table，包含了 column 在 MySQL 內儲存的資訊與功能描述。

✓ **jobinfo**

Field	Type	Null	Key	Default	Description
job_id	int	NO	PRI	NULL	職缺 id
job_name	varchar(100)	NO		NULL	職缺名稱
degree	tinyint	NO		NULL	最低學歷要求，見下方說明
low_salary	int	YES		NULL	薪資下限，面議時為 4 萬
high_salary	int	YES		NULL	薪資上限，面議時為 NULL
exp_year	tinyint	NO		NULL	最低年資要求
job_type	tinyint(1)	NO		NULL	工作型態，見下方說明
worktime	float	NO		NULL	工作時間（單位為小時）
is_night	tinyint	NO		NULL	日／夜班，見下方說明
needed_num	tinyint	YES		NULL	職缺需求人數

*說明：

Field	Explanation			
degree	0（不拘）	1（國中以下）	2（高中／職）	3（專科）
	4（大學）	5（碩士）	6（博士）	
job_type	0（正職）		1（兼職）	
is_night	1（日班）		2（夜班）	3（皆有）

✓ **company**

Field	Type	Null	Key	Default	Description
com_id	int	NO	PRI	NULL	公司 id
com_name	varchar(100)	NO		NULL	公司名稱
capital	varchar(10)	YES		NULL	資本額
emp_number	int	YES		NULL	員工人數
addr	varchar(100)	YES		NULL	公司地址

✓ **position**

Field	Type	Null	Key	Default	Description
pos_id	int	NO	PRI	NULL	職務 id
pos_field	varchar(8)	NO		NULL	職務領域
pos_name	varchar(20)	NO		NULL	職務名稱

*說明：此處的資訊是求職網站為職缺所立的類別，例如「業務／業務助理」或「光電半導體／半導體工程師」等。

✓ **localarea**

Field	Type	Null	Key	Default	Description
area_id	smallint	NO	PRI	NULL	地域 id
area_cc_name	varchar(10)	NO		NULL	縣市名稱
area_td_name	varchar(10)	NO		NULL	鄉鎮市區名稱
area_cctd_name	varchar(10)	NO		NULL	前兩者之連接

✓ **job**

Field	Type	Null	Key	Default	Description
job_id	int	NO	PRI	NULL	職缺 id
com_id	int	NO		NULL	公司 id
pos_id	int	NO		NULL	職務 id
area_id	smallint	NO		NULL	地域 id

✓ **user**

Field	Type	Null	Key	Default	Description
username	varchar(15)	NO	PRI	NULL	使用者名稱
passwd	char(64)	NO		NULL	雜湊過的使用者密碼

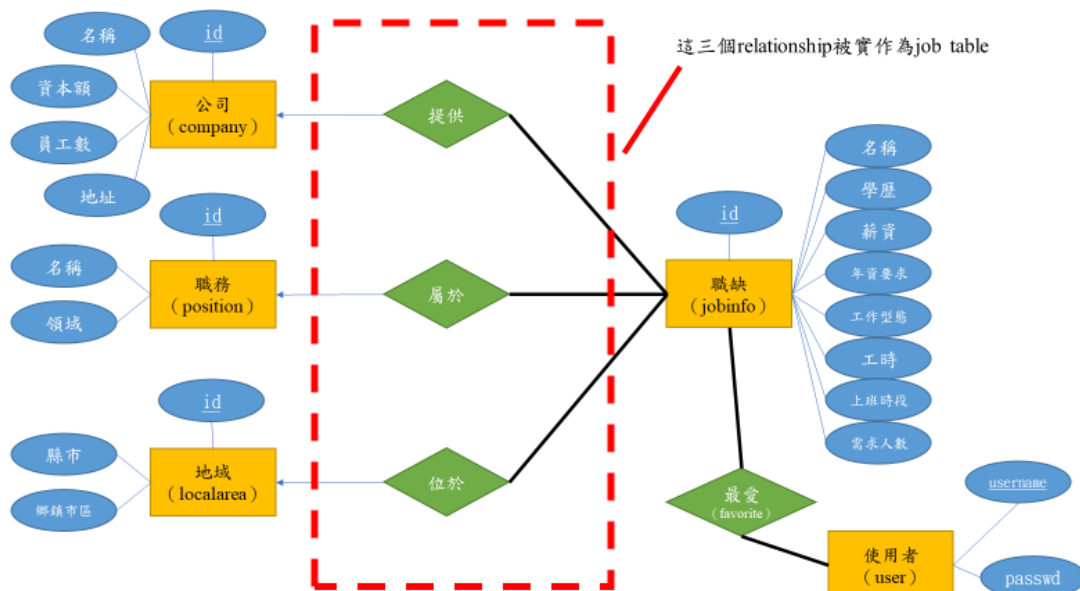
✓ **favorite**

Field	Type	Null	Key	Default	Description
username	varchar(15)	NO		NULL	使用者名稱
job_id	int	NO		NULL	職缺 id

*說明：此功能為使用者將職缺加入最愛，我們也會在功能說明的部分描述這項功能。

1.3 ER Models

下圖為我們的 ER Model。其中，職缺 (jobinfo) 和公司 (company)、職務 (position)、地域 (localarea) 三者之間的關係是多對一，而職缺 (jobinfo) 和使用者 (user) 的「最愛 (favorite)」關係是多對多。



2 Database

2.1 Introduction / Maintenance

我們使用的資料庫是 MySQL。

當我們要更新資料庫時，例如加入新的職缺資訊、公司資訊（陸續爬蟲下來），我們會利用 update.sql（/src/sql/update.sql）來更新，其中的步驟包含：

1. 將 job, jobinfo, company 三個 table truncate 掉。
2. 從爬蟲程式的輸出資料夾(/src/fetch/1111/result/<timestamp>)載入新取得的資料。
3. 檢查 favorite table，若其中包含著不存在於 job 的 job_id，代表這次的更新移除了部分使用者儲存過的職缺，這時會將這些資料從 favorite table 中移除。

除此之外，在運行階段時也會有動態的資料被加進資料庫中/從資料庫移除。這主要分為兩類：

1. 有新使用者註冊

當有新使用者要註冊時，若使用者資料皆符合要求(皆為英數字元，且 username 不存在於 user table 中)，則會在 user table 上新增一筆 (username, passwd) 的資料，其中 passwd 為經過 SHA-256 加密過的 64 位的十六進位字串。

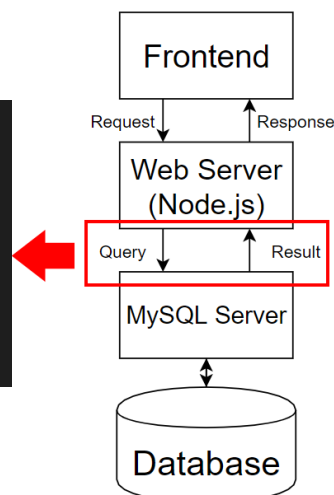
2. 使用者新增/移除了一筆最愛職缺

當使用者新增一筆最愛職缺時，會向 favorite table 增加一筆 (username, job_id) 的資料；反之當使用者移除某筆最愛職缺時，符合 (username, job_id) 的資料列則會從 favorite table 被移除。

2.2 Connection / Interaction

我們連接資料庫的方式是透過 Node.js 架設的後台在啟動時會 require npm 中的 mysql 模組，建立一個資料庫的 connection 實例，之後的 query 便都經由此實例來與資料庫互動。

```
const mysql = require('mysql')
const db = mysql.createConnection({
  host: 'localhost',
  user: 'db_final_user',
  password: 'cs2020db',
  database: 'db_final'
})
db.connect()
```



在前端呼叫後端存取資料庫時，前端會經由 HTTP POST 和 Web Socket 發送一條 serialized 成字串的 JSON，包含著必要的資料(通常是 job_id 等資訊)，後端收到前端的資料後會視情況(像是存取使用者最愛列表時)加入使用者目前 session 的資料，將資料渲染成相對應的 SQL 指令，並向 MySQL 的 connection 發出 query 請求，在結果回傳後會再將結果渲染成相應的 HTML 或 JSON，並回傳給前端顯示。

至於 Database exception handling 的部分，當資料庫的 query 出現錯誤時，後端會停止這個頁面的渲染，在 HTTP response 中寫入 error.html 的內容並回傳給前端。

前端有很多操作會需要連結資料庫，以下會詳細說明 (為了方便說明，從此以下到此小節結束前的檔案路徑都表示為斜體底線，並且檔案中的\${}皆是執行時段會被替換成正確資料的標籤)：

➤ 註冊

註冊時前端會將帳號及密碼 POST 到後端，後端會檢查資料庫中的 user table 有沒有包含 username 相同的資料列([user/userExist.sql](#))，若沒有則向 user table 添加一條(username, passwd)的資料列([user/newUser.sql](#))。

➤ 登入

登入時會向資料庫中檢查 user table 有沒有存在一條(username, passwd)皆符合的資料([user/validateUserPassword.sql](#))，若有則向前端返回登入成功，並將 username 寫入 session 中。

➤ 首頁最愛/首頁推薦

首頁最愛：使用者載入首頁時，後端會將 session 中的 username 渲染進([favorite/getUserFavInfo.sql](#))中，讀取出所有在 favorite table 中 username 相同的資料列。

首頁推薦：和最愛一樣，載入首頁時，後端會將 session 中的 username 渲染進([favorite/suggestionMain.sql](#))中，將 user 儲存在 favorite table 的資料讀出來做出推薦列表。

➤ 簡易搜尋/進階搜尋

網頁會將搜尋內容(若是進階搜尋則會包含其他限制條件)透過 Web Socket 傳給後端，後端會將資料渲染進([search/simpleSearch.sql](#) / [search/advanceSearch.sql](#))中來對資料庫做搜尋。

➤ 職缺頁面/公司頁面

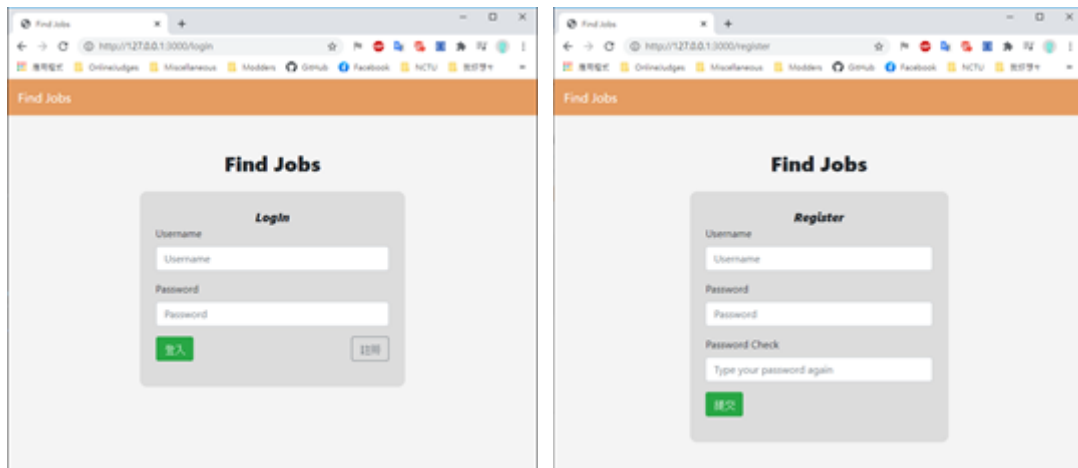
頁面的 job_id/com_id 會被替換進([job/singleJob.sql](#) / [company/complInfo.sql](#))中，取得資料庫中相對應的資料。

3 Application

3.1 Interface

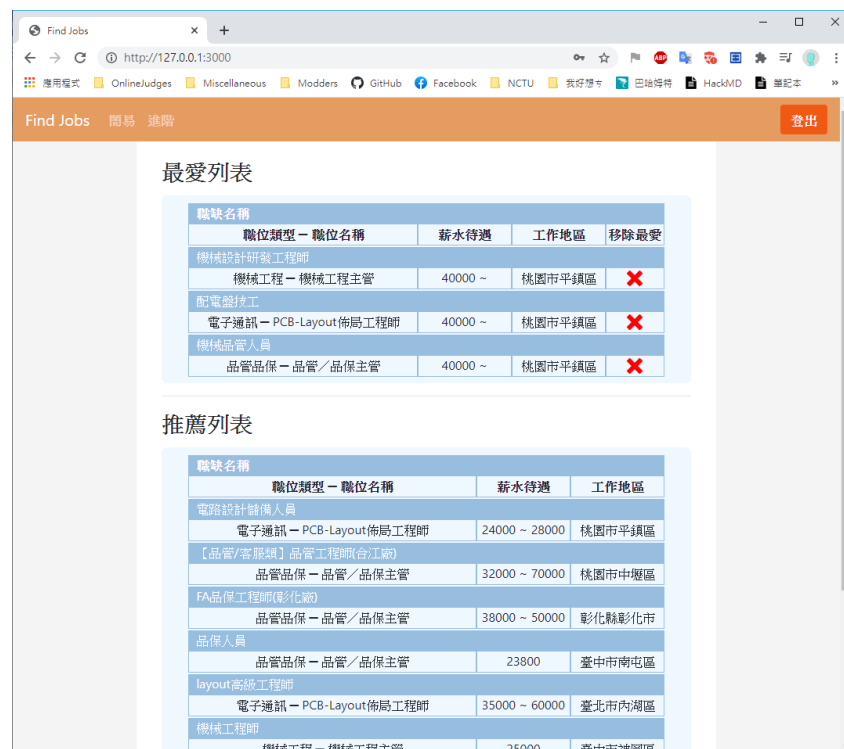
我們的專題使用 Web Server 架設了一個網站來呈現，我們的網站主要有幾個頁面：

➤ 登入/註冊頁面



上左圖為登入畫面，右圖為註冊畫面，使用時需要先登入帳號。

➤ 首頁



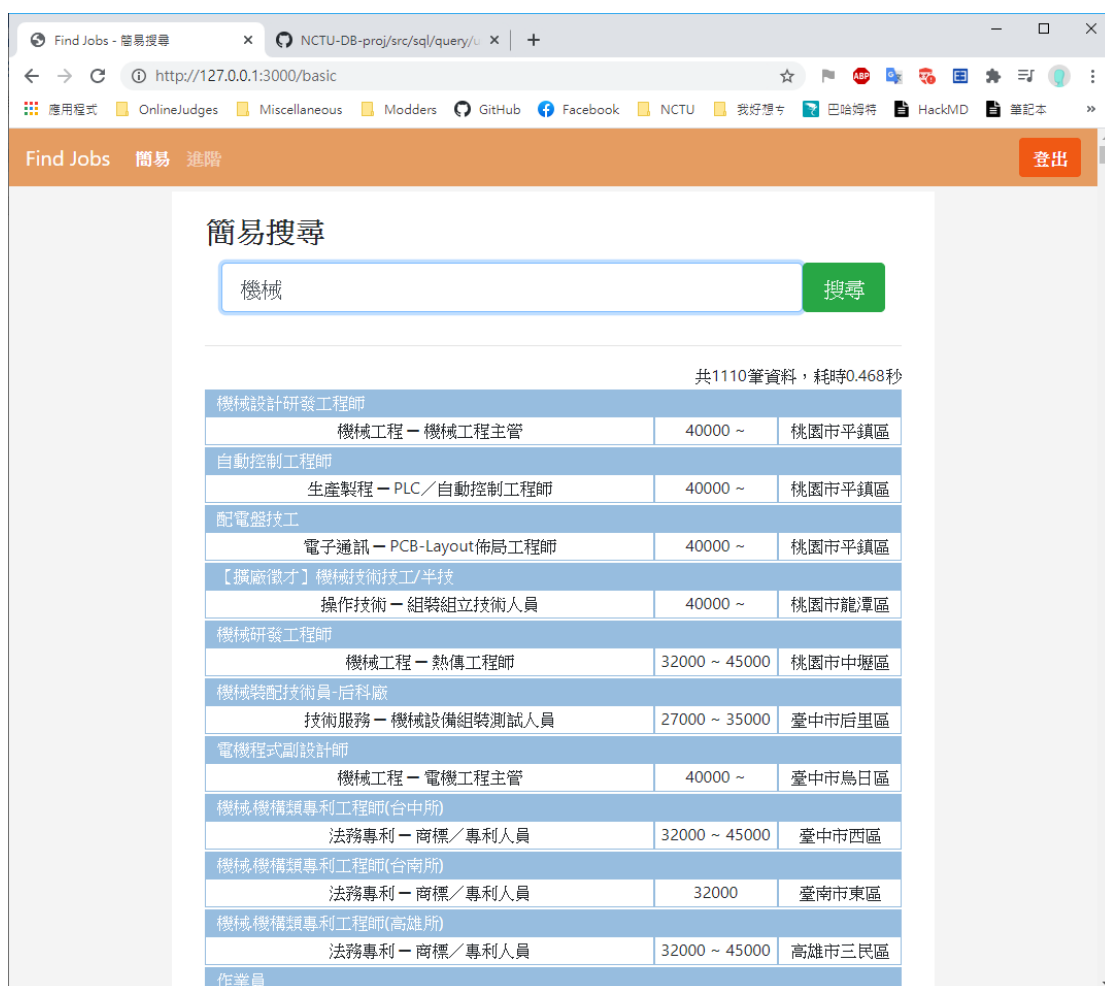
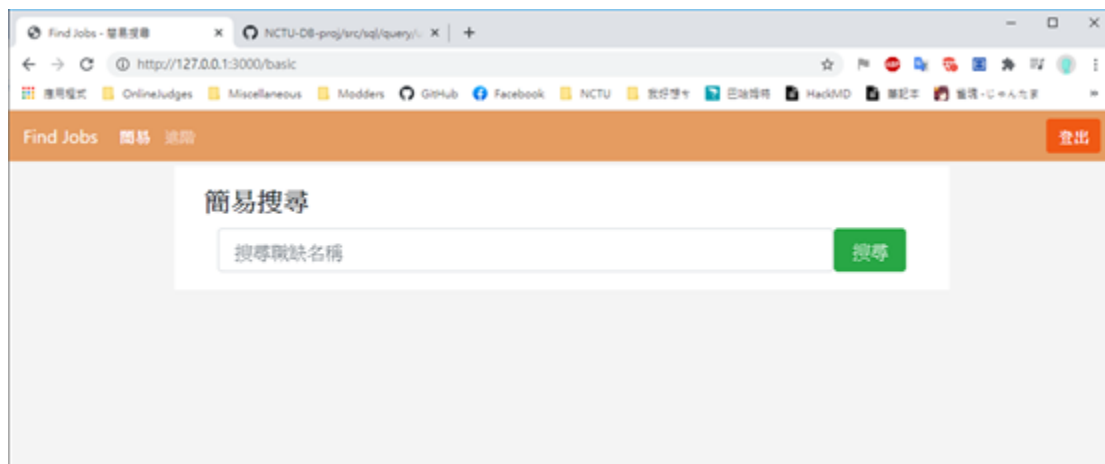
職位名稱	職位類型 - 職位名稱	薪水待遇	工作地區	移除最愛
機械設計研發工程師	機械工程 - 機械工程主管	40000 ~	桃園市平鎮區	✖
配電盤技工	電子通訊 - PCB-Layout佈局工程師	40000 ~	桃園市平鎮區	✖
機械品管人員	品管品保 - 品管/品保主管	40000 ~	桃園市平鎮區	✖

職位名稱	職位類型 - 職位名稱	薪水待遇	工作地區
電路設計儲備人員	電子通訊 - PCB-Layout佈局工程師	24000 ~ 28000	桃園市平鎮區
【品管/客訴類】品管工程師(含工廠)	品管品保 - 品管/品保主管	32000 ~ 70000	桃園市中壢區
FA品保工程師(彰化廠)	品管品保 - 品管/品保主管	38000 ~ 50000	彰化縣彰化市
品保人員	品管品保 - 品管/品保主管	23800	臺中市南屯區
layout高級工程師	電子通訊 - PCB-Layout佈局工程師	35000 ~ 60000	臺北市內湖區
機械工程師	機械工程 - 機械工程主管	25000	臺中市神岡區

首頁會顯示使用者加入過最愛清單中的職缺，並會依據使用者的最愛清單在下方顯示一個推薦列表，推薦使用者也可能會感興趣的職缺。

➤ 簡易搜尋

在這個頁面中使用者可以透過搜尋關鍵字找到符合的職缺，對應的職缺資訊列表會在使用者按下搜尋後被顯示在網頁搜尋欄的下方。



➤ 進階搜尋

在這個畫面中使用者可以設定除了關鍵字以外的限制，利用這些限制進一步的篩選符合自己條件及需求的工作

Find Jobs - 進階搜尋

進階搜尋

搜尋職缺名稱

您的學歷 大學 您的年資 2 年 工作時間 10 小時

職位類別 軟體工程 職位名稱 不拘

要求薪資 月薪 (全職) 地區 新竹市 東區

檢視統計資料 共25筆資料，耗時0.069秒

MIS 軟體工程師	軟體工程 ← MES工程師	40000 ~	新竹市東區
NEW((中層知名上櫃公司)) EAP_CIM工程師	軟體工程 ← MES工程師	40000 ~	新竹市東區
NEW((知名自動化企業)) EAP_CIM工程師	軟體工程 ← MES工程師	40000 ~	新竹市東區
NEW((徵優秀人才)) EAP_CIM工程師	軟體工程 ← MES工程師	40000 ~	新竹市東區
NEW資深自動化工程師 ((NO.1龍頭公司-環瑞(德))	軟體工程 ← MES工程師	40000 ~	新竹市東區
NEW中層上櫃公司 > 資深自動化工程師	軟體工程 ← MES工程師	40000 ~	新竹市東區
NEW資深自動化工程師 > 薪優 > 福利佳	軟體工程 ← MES工程師	40000 ~	新竹市東區
LI2001-MES系統開發工程師-新竹區力行廠	軟體工程 ← 軟體工程師	30000 ~ 70000	新竹市東區

使用者也可以點選上圖搜尋結果左上方的「顯示統計資料」按鈕，會跳出一個小畫面顯示符合使用者設定條件的所有職缺的統計資料，如下圖所示。

Find Jobs - 進階搜尋

進階搜尋 - 統計資料

平均薪資 42340.0000元

最高薪資 57500.0000元

最低薪資 32500.0000元

平均所需年資 0.3600年

平均工時 8.7800小時

最多符合條件工作的地區

縣市: 新竹市

鄉鎮區: 新竹市東區

關閉

➤ 職缺頁面

點選任何職缺資料的標題會開啟對應職缺的詳細資料畫面，七株包含了其工作時段即要求年資、學歷等資料，在下方還會推薦同個公司所徵求的職缺、同個职位的職缺及同地區的職缺讓使用者參考。

MIS 軟體工程師

群登電子股份有限公司			需求 1 名
月薪 40000 ~ 元	全職	日班	工時：9.5 小時
軟體工程 – MES 工程師	新竹市東區	大學	不拘 加入最愛

你可能感興趣...

同公司職缺

設備工程師(IC測試分類)		
光電半導體 – 半導體設備工程師	30000 ~ 38000	新竹市東區

軟體工程 – MES 工程師 職缺

機器人-數位化智慧應用處-MES資深工程師		
軟體工程 – MES 工程師	40000 ~	桃園市龜山區
【中科】數據應用工程師(機器學習&深度學習)		
軟體工程 – MES 工程師	32000 ~ 60000	臺中市大雅區
MES/WMS工程師-台南		
軟體工程 – MES 工程師	33000 ~ 48000	臺南市安南區
資訊工程師		
軟體工程 – MES 工程師	33000 ~ 50000	臺中市大甲區
MES 資訊工程師【戰略數據中心】地點：屏東長治鄉		
軟體工程 – MES 工程師	50000 ~ 70000	屏東縣長治鄉

同地區職缺

【竹科/半導體製造業】測試工程師 HCS.1167		
光電半導體 – IC封裝/測試工程師	38000 ~ 55000	新竹市東區
化妝品調製研發		
化學實驗 – 化學工程師	23800 ~ 35000	新竹市東區

➤ 公司頁面

點選職缺頁面中的公司名稱則會開啟該公司的資訊頁面，除了顯示該公司的詳細資訊，還會顯示該公司所徵求中的所有職缺。

群登電子股份有限公司

地址

新竹市東區明湖路476號7樓

員工人數

100 人

連絡電話

(03) 520-6089

該公司職缺

設備工程師(IC測試分類)		
光電半導體 – 半導體設備工程師	30000 ~ 38000	新竹市東區
MIS 軟體工程師		
軟體工程 – MES 工程師	40000 ~	新竹市東區

3.2 Functions & Implementations

我們的 application 的主要功能是查詢職缺及對職缺作統計分析，同時結合會員功能，使用者可以將中意的職缺加入自己的最愛中。以下我們將會條列各項功能，並解釋其運作方式及所用 query 的設計。我們使用的 query，是將 SQL statement 直接寫在後台中作為字串使用(見此)，使用時其中的標籤("\${...}")符號皆會被替換為對應的資料。

為了解釋用途，我們將用到的 SQL 儲存在/src/sql/query 資料夾底下，但並不會使用它們（而且因為其中包含著等待被替換的標籤，該 SQL 無法被直接執行）。

3.2.1. 登入和註冊

在使用任何功能之前，使用者必須先擁有一個帳號並且登入。我們的網頁伺服器會在使用者成功登入後在使用者的 request session 中寫入該使用者的 username，以便之後 query 資料時會用到。在連線到除了登入和註冊以外的任何頁面時，使用者 session 中的 username 都不應為 undefined，若是 undefined，代表此連線尚未登入，則將其重導向登入頁面。

我們的登入功能會在使用者點擊登入按鈕時向後台 POST 表單的內容，後台會先檢查資料是否過長或包含非法字元(username 長度限制 15 字元，username 和 passwd 皆只能有英數字元)，若通過則會將 passwd 經過 SHA-256 加密後執行([user/validateUserPassword](#))，若回傳長度為 1 代表有對應的使用者，此時將 username 寫入 session 中並導向首頁，若長度為 0 則回傳包含錯誤訊息的登入畫面，要求重試。

註冊功能則是在使用者發送註冊資料時先檢查字元合法性(同上述)，檢查使用者名稱是否存在([user/userExist](#))，若不存在再使用雜湊過的密碼執行([user/newUser](#))，將使用者名稱和加密過的密碼存入 user table 中，再將 username 寫入 session 中並導回首頁。

使用的 SQL：

[user/validateUserPassword](#)：SELECT 出 user table 中 username 和 passwd 皆符合條件的資料列，若列數大於 0 代表有相對應的使用者。

使用標籤：\${username}：使用者名稱, \${passwd}：加密過的密碼

[user/userExist](#)：SELECT 出 user table 中 username 跟標籤相等的資料列，若列數大於 0 代表使用者名稱已被使用。

使用標籤：\${username}：使用者名稱

[user/newUser](#)：將一系列新的(username, passwd) INSERT INTO user table。

使用標籤：\${username}：使用者名稱, \${passwd}：加密過的密碼

```
getHashedPasswd: function (passwd) {  
  return crypto.createHash('sha256').update(passwd).digest('hex')  
},
```

3.2.2. 首頁最愛/推薦

使用者在首頁可以看到他們曾經加入過最愛的工作列表，這個功能是透過在使用者載入首頁時，伺服器會使用 session 中儲存的 username 執行 ([favorite/getUserFavInfo](#))，取得使用者的最愛列表，並對每一筆最愛生成一個 HTML 元素顯示在首頁的最愛列表位置中。

首頁的推薦列表則是透過執行 ([favorite/suggestionMain](#))，統計使用者最愛清單中的前三多職位，並從所有工作中選出職位在前三名中，且本身不在最愛清單中，隨機排序後選出前十個，在如上述生成 HTML 元素插入相對應位置。

在最愛清單中的工作右下角會有一個刪除按鈕，按下後會生成一個包含該 job_id 的 HTTP POST 至 /delfav，伺服器收到請求後會執行 ([favorite/favDel](#)) 將 favorite table 中相對應的資料刪除。

使用的 SQL：

[favorite/getUserFavInfo](#)：SELECT 出 favorite table 中 username 和標籤相符的資料列，這些便是此使用者加入最愛的工作。

使用標籤：\${username}：使用者名稱

[favorite/suggestionMain](#)：先 SELECT 出 favorite table 中 username 和標籤相符的資料列，和 job table join 起來後選擇 favorite.job_id=job.job_id 的資料列，並 GROUP BY job.pos_id，用 COUNT 反向排序即可得到前三名的職位。再從所有 job 中 SELECT 出 pos_id 在前述 query 中且 job_id 不在該使用者最愛中的工作，隨機排序後取前 10 列。和 localarea, position, jobinfo 等 table join 後可瞿德其他資訊。

使用標籤：\${username}：使用者名稱

[favorite/favDel](#)：將 favorite table 中 username 和 job_id 皆符合的資料列刪除。

使用標籤：\${username}：使用者名稱, \${job_id}：欲刪除的工作 id

3.2.3. 簡易查詢功能

我們的查詢功能可以區分成簡易查詢與進階查詢。簡易查詢為了追求方便性，只有關鍵字搜尋，在使用者輸入關鍵字之後，網頁會透過 Web Socket 發送搜尋資料給後端，後端會用關鍵字執行 ([search/simpleSearch](#))，再對每一列搜尋結果生成相對應的 HTML，全部 concat 在一起後透過 Web Socket 送回前端。前端會在將搜尋結果放在正確的位置上，並顯示資料比數和消耗時間。

使用的 SQL：

[search/simpleSearch](#)：分別從 jobinfo 和 company 中 SELECT 出 job_name 和 com_name LIKE “%\${search}%”的資料，跟 job join 在一起後再和其他的 table join 取得詳細資料。

使用標籤：\${search}：搜尋關鍵字

3.2.4. 進階查詢功能

在進階查詢下，使用者可以針對職缺資訊，例如前述的關鍵字、學歷、年資、工作型態、工時、職位類別、薪資、地區做設定。

在按下搜尋時，前端會將所有的設定值透過 Web Socket 傳送給後端，後端則會用這些值來執行([search/advanceSearch](#))，再對每一列搜尋結果生成相對應的 HTML，全部 concat 在一起後透過 Web Socket 送回前端。前端會在將搜尋結果放在正確的位置上，並顯示資料比數和消耗時間。

在使用者未給條件賦值的情況下，我們也會給定合理的預設值。以下我們會針對各項資訊提供說明。

條件名稱	預設值	輸入	描述	條件標籤
關鍵字	不拘	文字	包含職缺或公司名稱之子字串	search
學歷	國中	選單	學歷需高於或等於職缺所需學歷	degree
年資	0	數字	年資需高於或等於職缺所需年資	exp_year
日／夜班	日班	選單	輸入需包含職缺工作時段	is_night
工時	0	數字	職缺工時須小於或等於輸入工時	worktime
職位類別	不拘	選單	輸入需與職缺職位類別相符	pos_field, pos_name
工作型態	全職	選單	輸入需與職缺工作型態相符	job_type
薪資	0	數字	薪資上限高於或等於預期薪資	salary
地區	不拘	選單	輸入需與職缺所在地區相符	area_cc_name, area_cctd_name

上面的 query 會先被 buffer 起來，統計分析會使用到這條 query。

在進階搜尋的清單回傳給前端後，後端會繼續執行統計資料的分析，前面 buffer 起來的 query string 會被當作統計 query 的 subquery 來使用。在統計分析完成後，會發送一個標記著統計資料類別的 json 給前端，其中包含著統計出的數值。前端會再將這些數值顯示在統計畫面上。

使用的 SQL：

[search/advanceSearch](#)：分別從 jobinfo 和 company 中 SELECT 出 job_name 和 com_name LIKE “%\${search}%”的資料，跟 job join 在一起後再和其他的 table join 取得詳細資料，join 時會在加上所設定的其他 constraint，做到進階搜尋的效果。

使用標籤：上列表格所列的所有條件標籤

[statistics/statisticAreaCC](#)：將搜尋結果 GROUP BY area_cc_name 後排序選出現最多的縣市。

使用標籤：\${base}：進階搜尋時 buffer 起來的 query string

[statistics/statisticAreaCCTD](#)：將搜尋結果 GROUP BY area_cctd_name 後排序選出現最多的縣市鄉鎮區。

使用標籤：\${base}：進階搜尋時 buffer 起來的 query string

[statistics/statisticSalary](#)：從搜尋結果中算出 MAX(_salary), MIN(_salary), AVG(_salary)。

使用標籤：\${base}：進階搜尋時 buffer 起來的 query string

[statistics/statisticsExpyear](#)：從搜尋結果中算出 AVG(exp_year)。

使用標籤：\${base}：進階搜尋時 buffer 起來的 query string

3.2.5. 職缺頁面

在使用者載入一個職缺頁面(/job/job_id)時，伺服器會將 job_id 放進 ([job/singleJob](#)) 中向資料庫 query 該職缺的詳細資料，在得到資料後會將結果放進 job.html 中對應的欄位裡。職缺資訊的右下角會根據用戶有沒有將這筆職缺加入最愛來決定要顯示「加入最愛」還是「移除最愛」，這個是透過執行 ([favorite/isInUserFav](#)) 來檢查的。

當使用者按下加入最愛時，會生成一個包含該 job_id 的 HTTP POST 至 /addfav，伺服器收到請求後會執行 ([favorite/favAdd](#)) 將 (username, job_id) 加入 favorite table 中。按下移除最愛時的行為和上面介紹首頁時按下最愛刪除的行為一樣。

在職缺頁面的下方則會顯示一些和此職缺有關的其他職缺，包含：

- 同公司的其他職缺
執行 ([job/suggestionComp](#))，用回傳結果生成 HTML 元素插入網頁中。
- 同職位的其他職缺
執行 ([job/suggestionPos](#))，用回傳結果生成 HTML 元素插入網頁中。
- 同地區的其他職缺
執行 ([job/suggestionArea](#))，用回傳結果生成 HTML 元素插入網頁中。

使用的 SQL：

[job/singleJob](#)：從 job 中 SELECT 出對應的資料列，再跟 jobinfo, company, localarea, position join 後取得其他詳細資料。

使用標籤：\${job_id}：職缺 id

[favorite/isInUserFav](#)：SELECT favorite table 中 username 和 job_id 皆等於標籤的資料列，若不為空代表此職缺已被此使用者加入最愛。

使用標籤：\${username}：使用者名稱, \${job_id}：職缺 id

[favorite/favAdd](#)：將 (username, job_id) INSERT INTO favorite table。

使用標籤：\${username}：使用者名稱, \${job_id}：欲新增的職缺 id

[favorite/favDel](#)：將 favorite table 中 username 和 job_id 皆符合的資料列刪除。

使用標籤：\${username}：使用者名稱, \${job_id}：欲刪除的職缺 id

[job/suggestionComp](#)：從 job 中 SELECT 出 com_id 等於標籤的資料，再和其他 table join 後取得其他資訊。

使用標籤：\${com_id}：公司 id

[job/suggestionPos](#)：從 job 中 SELECT 出 pos_id 等於標籤的資料，再和其他 table join 後取得其他資訊。

使用標籤：\${pos_id}：職位 id

[job/suggestionArea](#)：從 job 中 SELECT 出 area_id 等於標籤的資料，再和其他 table join 後取得其他資訊。

使用標籤：\${area_id}：地區 id

3.2.6. 公司頁面

在使用者載入一個職缺頁面(/comp/com_id)時，伺服器會將 com_id 放進([company/compInfo](#))中向資料庫 query 該公司的詳細資料，在得到資料後會將結果放進 comp.html 中對應的欄位裡。

在公司頁面的下方會透過執行([company/compJobs](#))取得這間公司徵求中的所有職缺並顯示在對應位置。

使用的 SQL：

[company/compInfo](#)：從 company 中 SELECT com_id 等於標籤的資料

使用標籤：\${com_id}：公司 id

[company/compJobs](#)：從 job 中 SELECT 出 com_id 等於標籤的資料，再和其他 table join 後取得其他資訊。

使用標籤：\${com_id}：公司 id

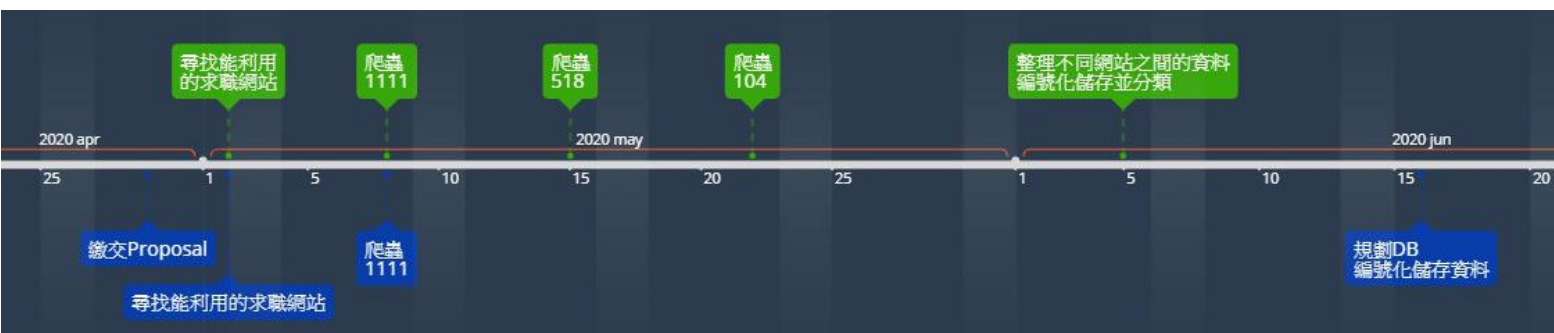
3.2.7. Exception Handling

在例外處理的方面，所有從前端傳回後端的資料都會經過檢查，確認資料合法性，像是數字欄位的內容必須要為數字(match “^[0-9]+\$”)，使用者名稱和密碼都只能為英數等。我們也有幫無輸入的欄位設立預設值(如前 3.2.4 節表格內容所示)，因此不會有 application 造成的例外發生。

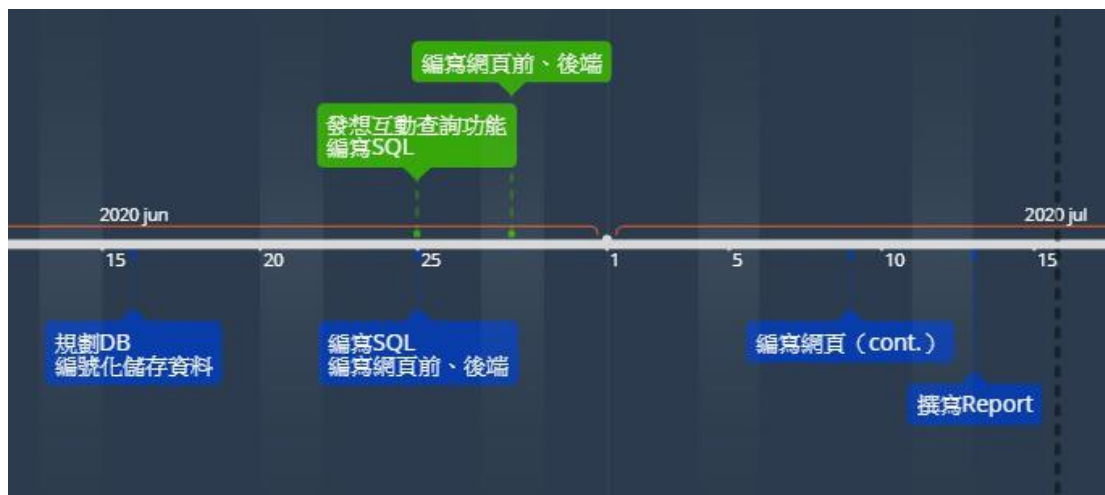
4 Others

4.1 Progress

上方為預期進度，下方為實際進度



(接續下頁)



4.2 Problems Encountered

- 爬蟲時發現目標網站會根據使用者行為來決定要更新什麼資訊，並不是一個靜態的網頁，所以我們改使用 selenium 套件來執行爬蟲，但用 selenium 就出現了執行速度慢的問題，因此我們不得不限縮資料的大小，我們拍攝 demo 影片時資料庫的資料量大約 13 萬筆，在執行爬蟲時便花了近 6 小時才將資料下載完成。未來使用時可能要使用分散式的爬蟲並讓爬蟲常駐在後台來避免更新耗時過久以及同一個 IP 位置向網站發出過多請求而被判定為惡意行為的問題。
- 從 4.1 節的進度圖表跟前面提到的爬蟲問題可以發現，我們在爬蟲上碰到的問題比當初想像的還要多很多，加上爬蟲耗時過久，發現問題到修正後重新執行並檢查問題是否解決，一個循環通常要花超過半天的時間，造成我們成功載下 1111 人力銀行的資料後發現進度比預期的落後非常多，我們判斷時間上並不允許我們再使用時間載下另一個求職網站的資料，必須趕快開始實作資料庫功能和使用介面才行，於是我們放棄了爬取其他求職網站上的資料，轉為只分析 1111 人力銀行的資料，並利用最後多出來的一點時間實作了原本計畫外的功能作為彌補（最愛功能、推薦職缺功能）。
- 我們在設計 SQL 指令時，因為在設計網頁時全部的連線都會共用同一個資料庫，所以不能有 temporary table 等暫存出現，不然在執行時會沒有辦法分別這個 temporary table 是哪一個連線在何時 CREATE 的。因此我們在設計進階搜尋時煩惱了許久，因為進階搜尋的統計功能需要重複的從一份職缺清單中做 query，考量到大部分的痛季都需要使用到 GROUP BY 等指令，會無可避免的遺失掉不少原本清單的資訊，所以沒辦法在一個 query 就算出所有的統計量。但若是一個一個算又會花太久時間，影響使用者瀏覽網頁的體驗，所以後來我們將其改成 query 出清單後會先回傳結果給前端，之後在使用原本的 query string 當作統計 query 的 subquery，算出統計量後分批回傳。這樣子唯一的影響只有在搜尋結果剛顯示時統計視窗內會顯示「統計中...」的字樣，大約過 1~2 後便會顯示出分批回傳的數值，不會影響到使用者操作。

4.3 Contributions

0716214 江岳勳	0716222 黃偉傑
爬蟲設計 架設伺服器、撰寫網頁 錄製 demo 影片 設計 SQL—登入註冊 設計 SQL—最愛列表 設計 SQL—職缺推薦 設計 SQL—職缺、公司資訊	專案進度管理 設計資料庫模型 為 demo 影片上字卡 設計 SQL—簡易搜尋 設計 SQL—進階搜尋 設計 SQL—統計分析

4.4 Link 2 Repo

<https://github.com/hm-ysjiang/NCTU-DB-proj>

4.5 Link 2 Discussion Channel

<https://trello.com/b/1isqFHkH/db-proj-discussion>