午餐系統及點餐資料分析

[文件副標題]

[壹、研究動機 3](#_Toc530432357)

[貳、研究目的 3](#_Toc530432358)

[參、實驗器材及設備 4](#_Toc530432359)

[肆、研究過程及方式 5](#_Toc530432360)

[一、資料庫 5](#_Toc530432361)

[(一)、資料結構 5](#_Toc530432362)

[(二)、資料表 6](#_Toc530432363)

[1. Order資料表 6](#_Toc530432364)

[2. Dish資料表 6](#_Toc530432365)

[3. User資料表 6](#_Toc530432366)

[4. 待擴充的資料表 6](#_Toc530432367)

[(三)、預存程序 6](#_Toc530432368)

[(四)、查詢點單 6](#_Toc530432369)

[1. 主查詢 6](#_Toc530432370)

[2. 擴充查詢 6](#_Toc530432371)

[3. 篩選條件 6](#_Toc530432372)

[(五)、資料庫死結 7](#_Toc530432373)

[二、後端 7](#_Toc530432374)

[(一)、系統架構 7](#_Toc530432375)

[(二)、處理流程 9](#_Toc530432376)

[(三)、session阻塞 9](#_Toc530432377)

[(四)、後端安全性 9](#_Toc530432378)

[(五)、最大負荷量 10](#_Toc530432379)

[(六)、架設環境 10](#_Toc530432380)

[1. 使用xampp的問題 10](#_Toc530432381)

[2. 使用wamp 10](#_Toc530432382)

[三、前端 10](#_Toc530432383)

[(一)、iOS前端 10](#_Toc530432384)

[1.呼叫後端 11](#_Toc530432385)

[2.Layout圖 11](#_Toc530432386)

[3.錯誤分析 11](#_Toc530432387)

[4.推播功能 11](#_Toc530432388)

[(二)、網頁前端 11](#_Toc530432389)

[1. ajax 11](#_Toc530432390)

[2. 分類樹 11](#_Toc530432391)

[3. 半耦合式後端 11](#_Toc530432392)

[(三)、輔助外掛 11](#_Toc530432393)

[(四)、前後端交換介面 12](#_Toc530432394)

[1. 點單資料結構 12](#_Toc530432395)

[2. 菜單資料結構 12](#_Toc530432396)

[3. 編碼 12](#_Toc530432397)

[四、預測模型 12](#_Toc530432398)

[(一)、擷取資料 12](#_Toc530432399)

[1. 點餐序列 12](#_Toc530432400)

[2. 點餐圖 13](#_Toc530432401)

[(二)、點餐矩陣 13](#_Toc530432402)

[1. 填入矩陣 13](#_Toc530432403)

[2. 矩陣特徵 13](#_Toc530432404)

[(三)、預測演算法 14](#_Toc530432405)

[壹、Dijkstra漫遊演算法 14](#_Toc530432406)

[一、目的 14](#_Toc530432407)

[二、實作 14](#_Toc530432408)

[貳、矩陣快速冪 15](#_Toc530432409)

[一、目的 15](#_Toc530432410)

[二、數學性質 15](#_Toc530432411)

[三、實作 15](#_Toc530432412)

[參、反矩陣解穩定狀態 16](#_Toc530432413)

[一、目的 16](#_Toc530432414)

[二、數學性質 16](#_Toc530432415)

[三、虛擬碼 16](#_Toc530432416)

[四、實作 16](#_Toc530432417)

[肆、線性回歸 17](#_Toc530432418)

[一、公式 17](#_Toc530432419)

[二、實作結果 17](#_Toc530432420)

[(四)、各演算法比較 18](#_Toc530432421)

[伍、目前研究結果 19](#_Toc530432422)

[陸、參考資料及其他 20](#_Toc530432423)

[篩選條件 20](#_Toc530432424)

# 壹、研究動機

建立一個可供廠商、消費者使用的平台

# 貳、研究目的

可追蹤特定目標族群

可預測明日之點餐數量

與產品經理討論並開發所需功能

# 參、實驗器材及設備

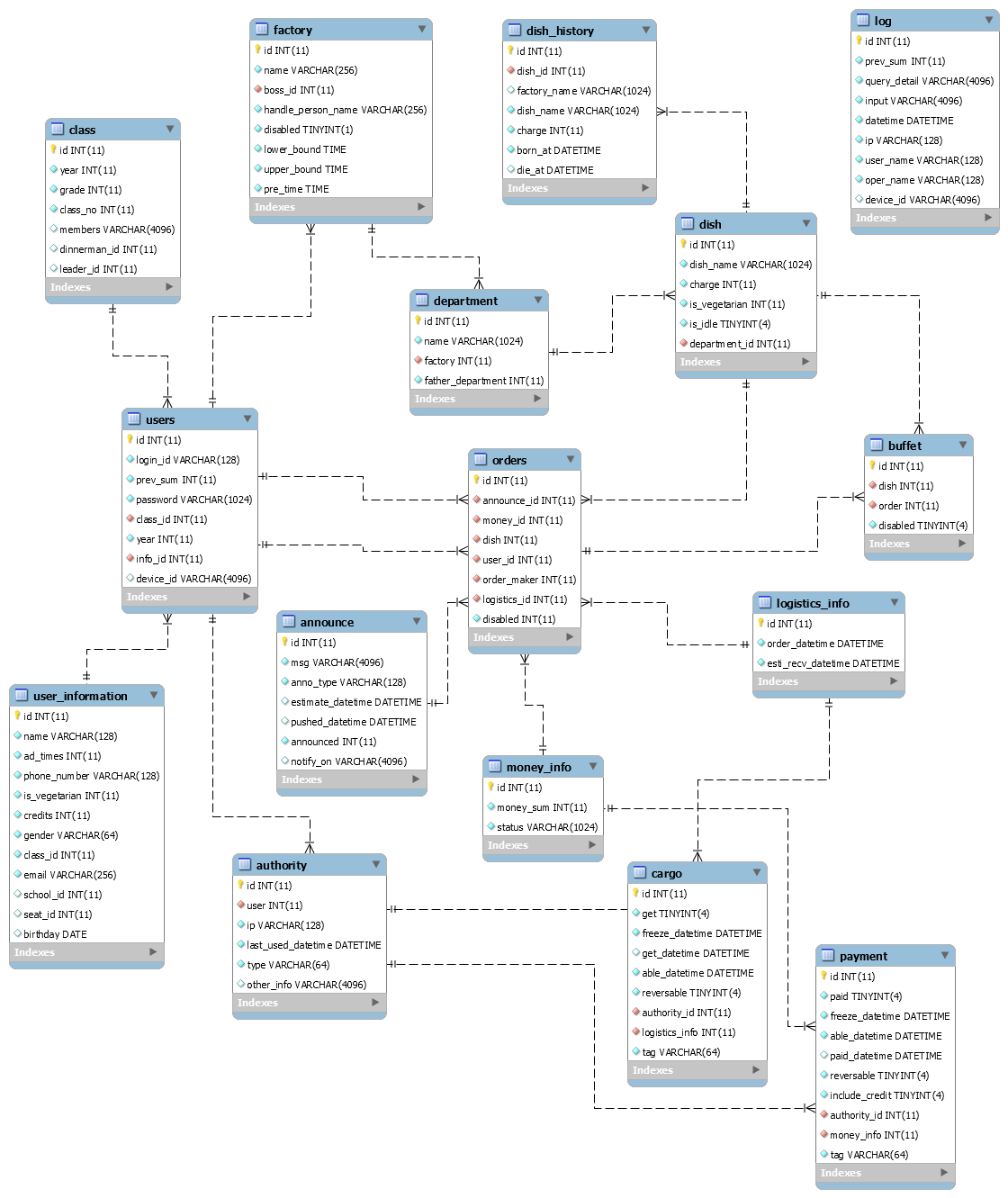
|  |  |  |
| --- | --- | --- |
| 電腦類 | 行政類 | 其他類 |
|  |  |  |
| Firebase | 校務會議支持 |  |
| Dynamic DNS |  |  |
| 伺服器電腦 |  |  |
|  |  |  |
|  |  |  |

# 肆、研究過程及方式

　　午餐系統後端係由php作為後台，mysql作為資料庫，並且輸出到前後端交換介面；凡是能夠操作前後端交換介面者，都可以稱呼為一個前台，包括網頁版前台、iOS前台、Android前台，都可稱為一個前台。

## 一、資料庫

　　下列為午餐系統的資料結構模型，每一條線代表一個關聯性連接。



#### 1. 點單查詢

由於點單的變動率高，直接從資料庫擷取資料較為合適，後台可以根據傳入參數，使用這六種篩選語句*(syntax)*組合成所需的篩選條件。

|  |  |
| --- | --- |
| 時間下界 | AND (? < LO.esti\_recv\_datetime) |
| 時間上界 | AND (? > LO.esti\_recv\_datetime) |
| 針對特定使用者搜尋 | AND (? = U.id) |
| 針對某間廠商進行搜尋 | AND (? = F.id) |
| 針對特定列查詢 | AND (? = O.id) |
| 針對班級查詢 | AND ((SELECT U.class\_id FROM users AS U WHERE U.id = ?) = U.class\_id) |

在下達sql指令時，可以使用代稱*(alias)*來方便撰寫sql語句，以下是篩選語句的解釋。

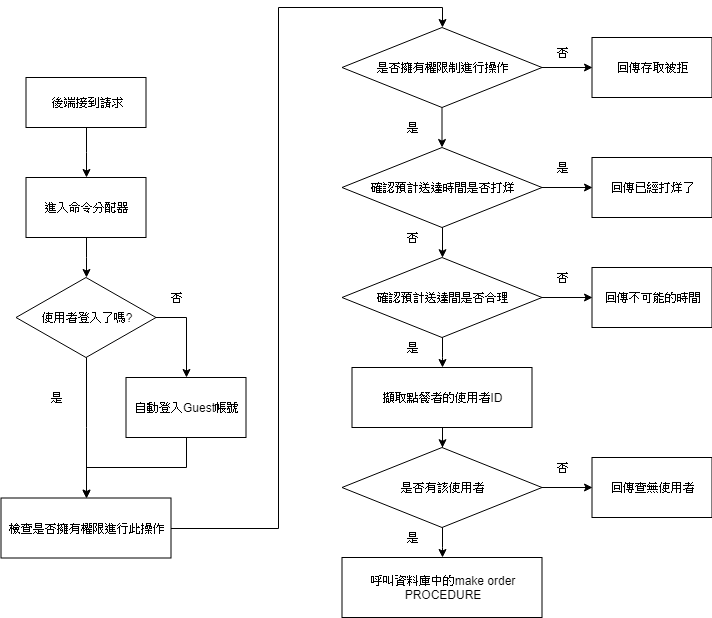
|  |  |  |
| --- | --- | --- |
| 中文名稱 | 英文全名 | 英文縮寫 |
| 使用者 | User | U |
| 物流資訊 | Logistic info | LO |
| 廠商 | Factory | F |
| 點單 | Order | O |
| 預計送達時間 | Estimate receive datetime | Esti recv datetime |

#### 2. Procedure優化

對於單語句(syntax)的sql操作，直接在php中呼叫資料庫即可；對於多語句(syntax)的sql操作，則包裝成一個Procedure，方便處理死結回溯，也不必與資料庫伺服器多次連線。

## 二、後端

### (一)、處理流程

****

　　以進行點餐操作為例，系統會先進行嚴密的輸入審查，確認是否無誤後，再將資料寫入資料庫。

### (三)、效能阻塞

#### 1. session 阻塞

　　php為了保證執行緒安全，同一個session同時只能給一個request使用。在每個request都會長期持有session的狀況下，同時送出大量request，就無法受益於平行處理；如果每個request都只會短期持有session，不必等待其他request繼續占用session，就可以同步進行操作。

　　session的資料中，大多都是常駐性資料，只有在登入時才會寫入，其他行為皆不會寫入session。為了受益於平行處理，系統經過命令分配器之後，就會決定是否已經可以釋放session了，使得其他request可以更快開始進行處理。

#### 2. sql效能低下

　　如果需要搜尋在「六月九號之後，八月七號之前」這個區間內的點單，若點單的時間有加上索引，我們可以使用二分搜尋法在*O(log N)*的時間內搜尋出資料；在沒有加上索引的情況下，系統沒辦法進行二分搜尋法，則需要*O(N)*搜尋整張資料表。

#### 3. 硬碟存取阻塞

　　硬碟的存取速度絕對低於記憶體的存取速度，如果每次使用常駐資料時，都向資料庫要求一次資料，則每次使用常駐資料都需要啟用硬碟一次。系統將資料先快取於session，需要使用常駐資料時直接從session調用資料，就不必再啟動硬碟一次了。

### (四)、安全性

針對後端安全，我們進行了以下幾種保護措施。

#### 1. 密碼安全性

對於所有工作人員的密碼，皆為六個字元以上的英數混和字串；並且將所有的登入失敗寫入紀錄檔中，若有異常登入資料，立刻通知該使用者。

#### 2. SQL注入

不以舊版的mysql模組操作資料庫，而以新版的mysqli操作資料庫；並且將所有的statement進行prepare後bind\_param，不直接在statement中寫入值，如下列虛擬碼所示。

|  |  |
| --- | --- |
| 具有sql注入風險 | 不具有sql注入風險 |
| $sql=”select id from orders where id = $uid”  $database->execute($sql) | $sql=”select id from orders where id = ?”  $database->prepare($sql)  $database->bind\_param(“1 or True”)  $database->execute() |

#### 3. XSS

在check\_valid 模組中，嚴格限制了每個輸入參數，凡是參數不符合規定，則立刻丟出例外狀況，並寫入記錄檔中。

#### 4. 行政安全性

向學生以及工作人員呼籲必須更改密碼，避免有他人惡意登入。我們選擇在學校的朝會、文字廣播系統上向學生呼籲必須更改密碼；因工作人員人數較少，我們一一協助工作人員更改自己的密碼。

#### 5. 資料庫死結

　　資料庫

## 三、前端

### (一)、iOS前端

#### 1.呼叫後端

#### 2.Layout圖

#### 3.錯誤分析

#### 4.推播功能

### (二)、網頁前端

#### 1. ajax

#### 2. 分類樹

#### 3. 半耦合式後端

### (三)、輔助外掛

#### 1. 當日會計報表

#### 2. 新生資料匯入軟體

### (四)、前後端交換介面

#### 1. 點單資料結構

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | |  |  | | --- | --- | | **id** | 22410 | | **user** | |  |  | | --- | --- | | **id** | 11184 | | **name** | 白翔云 | | **seat\_no** | 20904 | | | **dish** | |  |  | | --- | --- | |  | | | **dish\_name** | 卡拉雞 雞塊 三副菜 | | **dish\_id** | 1 | | **dish\_cost** | 55 | | | **recv\_date** | 2018-11-11 12:00:00 | |

#### 2. 編碼

## 四、預測模型

### (一)、資料結構

原始的點餐資料可以視為下列表格，「-」代表沒有點餐。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 甲生 | 乙生 | 丙生 | 丁生 | 戊生 |
| 08/07 | 韓式拌飯 | 烤肉飯 | - | 烤肉飯 | 素便當 |
| 08/08 | - | 烤肉飯 | 韓式拌飯 | - | 素便當 |
| 08/09 | 素便當 | 韓式拌飯 | 素便當 | 烤肉飯 | 素便當 |
| 08/10 | 烤肉飯 | 烤肉飯 | - | 韓式拌飯 | 素便當 |
| 08/11 | 韓式拌飯 | 烤肉飯 | 韓式拌飯 | 韓式拌飯 | 素便當 |

以下根據所需的原料進行分類，由上往下進行篩選，例如「韓式拌飯」符合第一行的Regex篩選，故屬於調味飯類；「烤肉飯」不符合第一行的Regex篩選，而符合第三行的Regex篩選，故屬於便當類。

|  |  |
| --- | --- |
| 調味飯類 | Regex: "(焗)|(拌飯)" |
| 麵類 | Regex: "(烏龍)|(麵)|(湯)" |
| 便當類 | Regex: "(副菜)|(飯)" |
| 小吃類 | Regex: "((餃)|(蔥抓餅)|(鍋貼)|(板條))" |
| 鍋類 | Regex: "(鍋)|(粥)" |
| 其他 | 不滿足以上篩選者 |

#### 1. 點餐序列

將甲生的點餐資料提取出來，並且加以分類，再填上是否有點餐，即為點餐序列。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 甲生 | 08/07 | 08/08 | 08/09 | 08/10 | 08/11 |
| 餐點類別 | 調味飯類 | - | 其他 | 便當類 | 調味飯類 |
| 是否點餐 | True | False | True | True | True |

#### 2. 點餐圖

根據原點餐序列，刪去沒有點餐的那一天，即如下表。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 甲生 | 08/07 | 08/09 | 08/10 | 08/11 |
| 餐點類別 | 調味飯類 | 其他 | 便當類 | 調味飯類 |
| 是否點餐 | True | True | True | True |

將每一天的轉移視為圖上的一個邊，每一個分類視為圖上的一個節點，則可得到一個有向圖，我們稱呼為點餐圖。

若將每一個人的點餐序列寫入同一張點餐圖，則可得到類似下面的有向圖。

#### 3. 點餐矩陣

### (一)、宏觀模型

|  |  |
| --- | --- |
| 電腦進行的操作 | 現實世界的意義 |
| 從點餐圖上的任意一個點出發 | 第一天先隨便點一道菜 |
| 經由任意一條邊 | 經過了一天 |
| 到達另外圖上的另外一個點 | 第二天點了另外一道餐 |
| 經過一個自環 | 經過了一天，決定要吃同一道菜 |
| 到了同一個點 | 第三天點了同一道菜 |

### (二)、微觀模型

方便細部追蹤一個群體

### (三)、模型應用

便當的原料

廠商只需要準備多少資料

### (二)、宏觀模型

# 伍、目前研究結果

## 一、系統特色

(一)、安全性

(二)、穩定性

(三)、效能

(四)、擴充性

## 二、分析實例

## 三、實際案例

# 陸、參考資料及其他

前後端資料交換介面

前後端命令界面

前後台原始程式碼 github

壓力測試數據

漫遊演算法虛擬碼

矩陣快速冪虛擬碼

資料表詳細描述

點餐序列

點餐圖

點餐矩陣

馬可夫轉移矩陣

方程式矩陣、反矩陣

平衡狀態矩陣

線性回歸方程式

伺服器環境參數

矩陣相乘演算法

並查集

有向圖、入度、出度、完全圖

反矩陣得解之證明

線性回歸公式之證明

系統權限等級

二分搜

合併排序、快速排序