

**國立台灣科技大學資訊工程系**  
**一百零七學年度第一學期專題研究**

總報告

**商品智慧訂閱系統**

研 究 組 員

廖晨竹      B10415031

謝明軒      B10415038

林崇恩      B10415046

陳重佑      B10415047

**指導老師：**邱舉明 教授

中            華            民            國            107 年    10 月    22 日

# 目錄

目錄	I
第一章 緒論	1
1.1 研究動機	1
1.2 研究目的	1
1.3 研究簡介	1
1.4 報告組織	2
第二章 相關研究	3
2.1 Python	3
2.2 PHP	3
2.3 CSS	4
2.4 HTML	4
2.5 SQLite	5
2.6 網路爬蟲	5
2.7 聊天機器人Chatbot	6
2.8 Selenium	7
2.9 BeautifulSoup	7
2.10 sqlite3	7
2.11 Requests	7
2.12 re	8
2.13 string	8
2.14 smtplib	8
2.15 thread	8
第三章 系統操作架構	8
3.1 執行流程	8
3.1.1系統主流程	8
3.1.2網頁執行流程	9
3.1.3 Line Chatbot執行流程	9
3.1.4 爬蟲執行流程	10
3.2程式架構	11
3.2.1 搜尋及查詢資料庫(演算法)	11
3.2.2 寄信與資料庫比對	12
3.2.3 網路爬蟲演算法	13
3.2.4 LINE Chatbot基礎架構	15
第四章 結果與討論	16
4.1前端畫面	16
4.1.1 首頁畫面	16
4.1.2 會員註冊畫面	16
4.1.3 登入畫面	17
4.1.4 商品搜尋	17
4.1.5 搜尋結果	18
4.2 Line ChatBot 介面	19
4.2.1 基本操作選單	19

4.2.2 指令	19
4.2.3 登入	20
4.2.4 登出	20
4.2.5 訂閱	21
4.2.6 查看	21
4.2.7 刪除	22
4.3 寄信範例	23
4.4 結論	23
參考文獻	24
附件A(後端)	30
附件B(前端)	53

# 第一章 緒論

## 1.1 研究動機

線上購物已成為現代人購物的主要方式之一，電子商務已然改變我們的生活方式，精打細算的客戶穿梭在不同的平台上比較價格，卻不一定可以找到符合其需求的商品。潛在客戶可能會因為沒有找到商品而流失，但也許隔幾天就有符合需求的商品出現，而且客戶的需求可能不全是突發性的，而是有一段需求的區間，例如購買生日禮物、機票等。

現有的購物網站，使用者需要每天固定去搜尋，他才能確認是否有他所需要的商品，而我們提供他們以需求的方式搜尋商品，使用者僅需要設定下商品、價格等相關資訊，我們即可為他提供定期搜尋，並且於需求符合時通知他。研究搜尋的執行時機、需求資訊與產品之間的比對方式、演算法的改進等，提供使用者更佳的購物體驗與方式，也是本計畫著眼的方向。

## 1.2 研究目的

開發出一個系統可以自動搜尋使用者所需的物品商品，現在人們能使用的線上購物網站太多，幾乎不可能將所有的購物網站都全部看過一遍來尋找自己預算中的商品。藉此這個系統來當個小助理幫忙這樣需求的用戶。只要輸入所需的物品、價格、規格等等。系統會自動搜尋直到一有物件便會立刻用電子郵件或其他通信軟體通知使用者，讓使用者能夠在最短最少的時間內也最精確的搜尋到他所要的物件。

- 時間：客戶為此商品設定一段期望購買到它的時間
- 預算價格：客戶的預算容許多少價格的商品，甚至不需要價格只需要輸入想要的商品，也能夠搜尋。
- 商品：客戶想購買的商品。例：手機iPhoneXS、筆電 ASUS S406UA 256G

## 1.3 研究簡介

結合使用者操作面、資料庫系統、背後的爬蟲程式來完成我們的專題。後端的爬蟲程式會往各大拍賣網站或是購物網站爬下所有商品資料進我們的大資料庫裡，而使用者在登入網站後即可對網站下訂單也就是搜尋功能。而使用者的訂單會儲存在資料庫系統裡，而資料庫就會開始比對開始檢索是否有對應資料(商品)。如果符合使用者需求，便會寄信給使用者通知已搜尋到。

- 使用者操作面：也就是前端網頁製作，需要把使用者的資料與資料庫結合，並且設計出使用者方便的介面。
- 資料爬蟲：也就是後端的資料來源，需要寫出爬蟲程式使其可以在各大拍賣網站或購物網站抓取資料，並且分類出名稱、樣式、價格、網址等等相關資料儲存進資料庫裡。
- 資料庫系統：結合了前端與後端的中樞橋梁，包括銜接程式運作與健全的資料庫。需要利用前端的使用者需求搜尋後端的爬蟲資料庫，進而把結果告知使用者，並且使用者能隨時修改需求時間及商品。資料庫要能隨時面對使用者不同的需求產生不同的結果。
- Line Chat Bot：鑒於現在智慧型手機普及，我們也加入了使用Line Chat Bot來做操作介面，讓使用者能直接在LINE上登入帳號並搜尋商品，而找到的商品也會直接用LINE來通知使用者，讓使用者又多出一個方便的管道。

## 1.4 報告組織

報告將針對相關研究、系統操作架構、結果與討論進行說明。第一章緒論，說明研究的動機目的與簡介此研究。第二章相關研究，簡述我們所採用的程式語言、框架。第三章系統操作架構部分，說明使用者之操作方式與程式內部執行之架構。第四章結果與討論，對於計畫最終所完成的結果進行說明與討論。

## 第二章 相關研究

### 2.1 Python

是一種廣泛使用的高階程式語言，屬於通用型程式語言，由Guido van Rossum 創造，第一版釋出於1991 年。作為一種直譯式語言，Python 的設計哲學強調代碼的可讀性和簡潔的語法（尤其是使用縮排劃分代碼塊，而非使用大括弧或者關鍵詞）。相比C++ 或Java，Python 讓開發者能夠用更少的代碼表達想法。不管是小型還是大型程式，該語言都試圖讓程式的結構清晰明了。

與Scheme、Ruby、Perl、Tcl 等動態型別程式語言一樣，Python 擁有動態型別系統和垃圾回收功能，能夠自動管理記憶體使用，並且支援多種編程範式，包括物件導向、命令式、函數式和程序式編程。其本身擁有一個巨大而廣泛的標準庫。

Python 虛擬機器本身幾乎可以在所有的作業系統中執行。Python 的官方直譯器CPython，用C語言編寫，是一個由社群驅動的自由軟體，目前由Python軟體基金會管理。

### 2.2 PHP

PHP-全名PHP:Hypertext Preprocessor (Personal Homepage Program)

PHP是一套讓你可以製作動態網頁的工具。內嵌PHP語法的HTML網頁文件仍會被當作一般的HTML網頁文件來處理，可以用同樣的標準HTML語法來編輯HTML部分。

內嵌的意思是，將PHP的程式寫在HTML裡面，而當伺服器執行網頁程式時，會先去直譯PHP的程式，並輸出在使用者的瀏覽器上，使用者所看到的是PHP程式執行的結果，而不是原始程式。

由於開放、免費、跨平台、易學及效率高等等的優點，全世界有許多的團體和組織提供一大堆的原始碼及函式庫，可協助快速而彈性地進行開發工作。所以說，PHP 可簡單的視為一種程式語言，可以用來設計：互動式介面、留言板、討論區、電子商務、多媒體應用、聊天室等等的動態網頁。

雖然PHP算是一個後端語言，但在此次專題中，前端負責人也有利用PHP加上MySQL來進行會員機制的管理以及資料呈現的撰寫。

## 2.3 CSS

現在要介紹的CSS再拿人來舉例的話，就好比人的膚色頭髮身上穿的衣服等等，對於HTML來說提供了裝飾、排版等功能。能夠使HTML所架構出來的網站進行更進一步的優化。

CSS是不能單純使用的，就好比衣服放在那邊終究只是一件衣服，重點還是要將其穿在人身上，才有它實質的功用，因此HTML跟CSS可以說是一體兩面，缺一不可的程式語言。

而CSS也如同PHP一樣有內嵌的特性，可以於HTML用style標籤所包圍起來，或是在任一HTML標籤當中引入style這個關鍵字，如此一來網頁在編譯時會自動將此區塊內的程式碼看做是CSS的程式，但大多數工程師都會將CSS撰寫於另外的檔案內，再於HTML用引入的方式來引入CSS，如此一來的好處是能使HTML的程式碼最為簡潔，若是要對CSS或HTML進行修改都能更為直覺以及方便。

## 2.4 HTML

HTML-全名(Hypertext Markup Language)，是一種超文字標示語言，利用其特有的元素(Element)以及標籤(Tag)，用來控制網頁上各式各樣的樣式呈現，包括文字大小、字體粗細、圖片或超連結的呈現等等，都能透過HTML來完成，HTML提供了一個網站最最基本的樣式呈現，就好比是人類的骨架，屬於網站基底的部分。

所謂「超文字」(Hypertext)和一般傳統文件最大的不同就是，傳統文件只能讓你循序地閱讀，而Hypertext藉由一些特殊標籤（例如連結）的協助，可以让你以樹狀甚至網狀（所以叫Web）來組織你的文件。標示(Markup)是透過在文章中插入標籤(Tags)來賦予文字一些特性，如標題，段落，連結等等。標籤乃是由「小於」(<)與「大於」(>)這兩個符號所括住的文字，如果想對某段文字加上特別的意義，我們只要在前後分別加上開始與結束的標籤即可，例如<title>我是標題</title>以上這段程式碼便能產生網頁最上層的網頁標題。

而當我們在瀏覽網頁時，可以隨時按下右鍵選擇(開發者工具/檢查網頁原始碼)，便能清楚的瀏覽當下你面前的網頁的內部結構，也能透過此工具修改此網站上的HTML進行暫時性的介面修改，CSS也能透過此工具進行暫時性的更動。

## 2.5 SQLite

SQLite 與其他一般資料庫差異不大，一般的SQL-92 語法都能夠使用，而且不需要建立一個資料庫系統，要使用的時候，只要在編譯程式的時候將SQLite 程式庫一起編入就可以使用。另外，SQLite 的資料庫（database）都是以單一檔案的形式存於磁碟中，不需要再安裝資料庫伺服器軟體，所以要把資料庫複製或建立在你的電腦上是相單簡單快速。

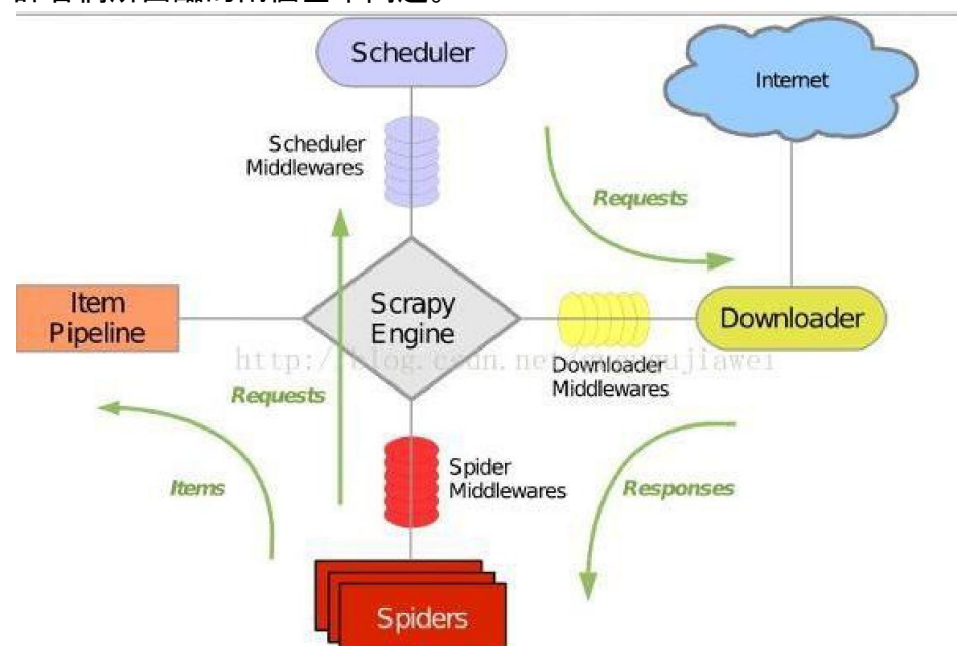
SQLite 是一個很小的C 語言程式庫，且本身就完全包含資料庫引擎的功能，而且可以嵌入至其他程式中，完全不用額外的設定。

## 2.6 網路爬蟲

網路爬蟲也叫網路蜘蛛（spider），是一種用來自動瀏覽全球資訊網的網路機器人。其目的為編纂網路索引。本專題採用基於python語言的scrapy。

網路爬蟲始於一張被稱作種子的統一資源位址（URL）列表。當網路爬蟲存取這些統一資源定位器時，它們會甄別出頁面上所有的超連結，並將它們寫入一張「待訪列表」，所謂爬行疆域（crawl frontier）。此疆域上的諸URL將被按照一套策略迴圈存取。

網際網路資源多如繁星，這意味著網路爬蟲在一定時間內只能下載有限數量的網頁，因此它需要最佳化它的下載方式。網際網路資源瞬息萬變，這也意味著網路爬蟲下載的網頁在使用前就已經被修改甚至是刪除了。這是網路爬蟲設計者們所面臨的兩個基本問題。





## 2.7 聊天機器人Chatbot

聊天機器人是透過文字、圖片等進行交談的電腦軟體，是近幾年來發展最熱門且應用在無論是商業或生活上最廣泛的技術之一，其實早在1966年麻省理工學院就有發展出當時世界上第一個聊天機器人(ELIZA)，隨著人工智慧、語意分析等技術的大幅進步，加上社加平台的全面普及化，聊天機器人又再一次活躍起來。目前聊天機器人的應用範圍包括客戶對答服務或資訊獲取應用等，有些聊天機器人會搭配自然語言處理系統，讓聊天機器人能自己找尋最合適的應答文字，讓使用者誤以為和真人在交談，這樣不僅能節省人力的成本，更能提升市場價值。

現今的聊天機器人廣泛運用在各大即時通訊消息平台，例如LINE、臉書Messenger、WeChat、Kik等，通常是以網路行銷、客戶服務和休閒娛樂為主。各大品牌廠商也運用聊天機器人來進行販售產品、回答問題、推廣品牌等；也能運用在教育上，聊天機器人能針對學生的問題進行回答，讓學習更加有效率且更方便。找尋聊天機器人最適合運作的場合，並加強與使用者溝通流暢性的行為，是目前聊天機器人重要發展的議題。

## 2.8 Selenium

是自動化測試工具。它支持各種瀏覽器，包括Chrome, Safari, Firefox 等主流界面式瀏覽器，如果你在這些瀏覽器裡面安裝一個Selenium 的插件，那麼可以方便地完成對Web界面的測試。換句話說叫Selenium 支援這些瀏覽器驅動。Selenium支持多種語言開發，比如Java, C, Ruby、Python。安裝時在cmd中輸入pip install selenium 即可。

## 2.9 BeautifulSoup

Beautiful Soup是一個Python包，功能包括解析HTML、XML文檔、修復含有未閉合標籤等錯誤的文檔（此種文檔常被稱為tag soup）。這個擴展包為待解析的頁面建立一棵樹，以便提取其中的數據，這在網絡數據採集時非常有用。

## 2.10 sqlite3

python對sqlite資料庫操作的library，主要使用包含connect、cursor、fetch、commit等函式。

connect()：即是資料庫的連接，可以利用它來切換連接到不同的資料庫。

cursor()：類似操作介面，是用來操作connect後的回傳值經cursor呼叫execute()即可使用sqlite的關鍵字。

fetch()：就是把execute()後的資料抓取出來，存成python列表的形式，有fetchone()、fetchall()兩種。

`commit()`：即許可，許可前面所有尚未被許可的操作但僅限於cursor所對應的資料庫。

## 2.11 Requests

是在Python上的一個HTTP library由Kenneth Reitz所撰寫，是用來做網路爬蟲非常簡單易懂的一個工具，它可以將一個網頁用僅僅幾行程式碼就將其抓下並用可讀的格式輸出處理，雖然requests並不是萬能的，可是對於大部分的網站卻是沒有問題的。

## 2.12 re

re就是所謂的正規化，正規化可以幫助我們更容易擷取的字串中我們所需要的部份。例如在爬蟲的時候，因為網頁都是用html寫的，所以可能會包含很多我們不想要的內容，這時候就可以用`re.search`，抓到我們要的，也可以利用`re.split`，切字串的方式，還有我們在做商品需求的字串比對的時候，為了預防可能有人亂輸入名稱，也許有特殊符號或是空格，所以我們會先用re模塊做一些字串上的整理。

## 2.13 string

string模組也是在對字串處理上有很大的幫助，跟re模塊可以相互幫助使用，像是punctuation的功能，可以將符號都刪除掉，`replace`則可以把字元替換成我們想要的。

## 2.14 smtplib

SMTP(Simple Mail Transfer Protocol)即簡單郵件傳輸協議，python對smtp協議進行簡單的封裝提供很方便的途徑發送email。

## 2.15 threading

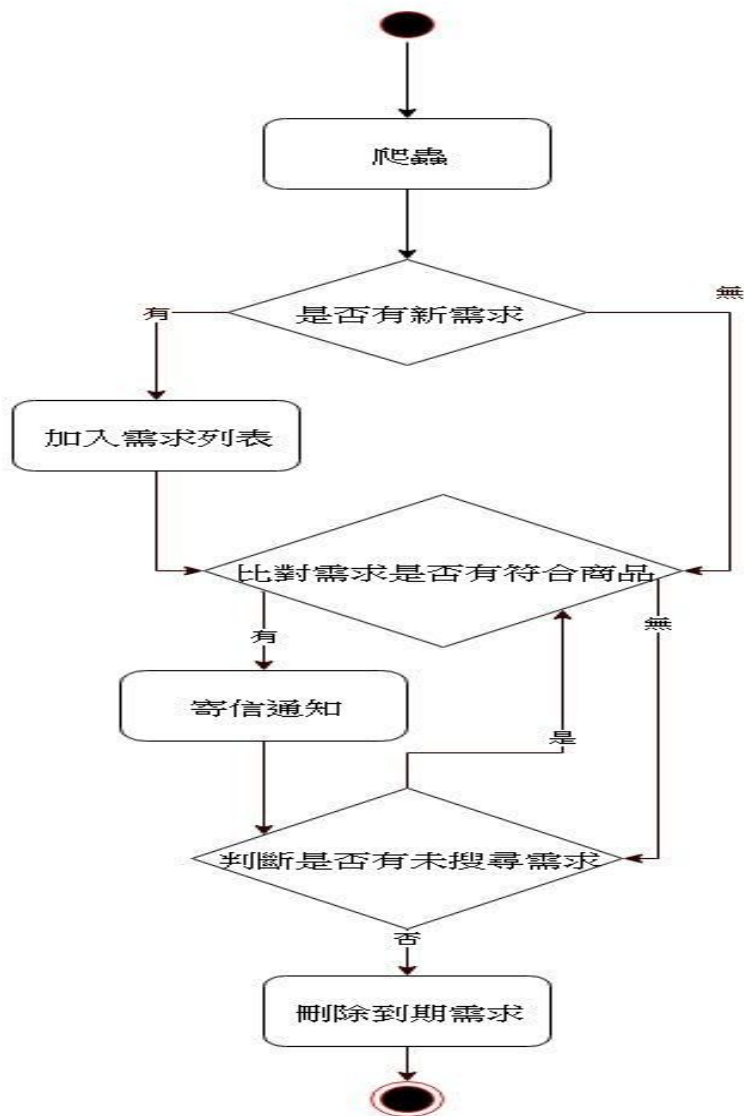
python的在執行緒這部分提供兩個套件，一個是thread、另一個是threading，thread提供較原始的線程以及鎖，而threading則是將一些功能封裝起來，讓我們可以直接使用，專題中我們為了提升爬蟲的效率，所以在這部分使用了Multi-thread，新增三個thread同時做三個爬蟲程式，還有再自動執行的部分，我們也是用threading裡面的Timer去做定時執行主程式的部分，例如我

們要定時去資料庫檢查是否有新的資料進來，我們就是採取每10秒鐘去執行檢查的程式，而爬蟲以及搜尋商品我們的想法是一天一次或兩次，不過為了方便讓我們看出程式的流程，我們會設定每幾分鐘就做一次。

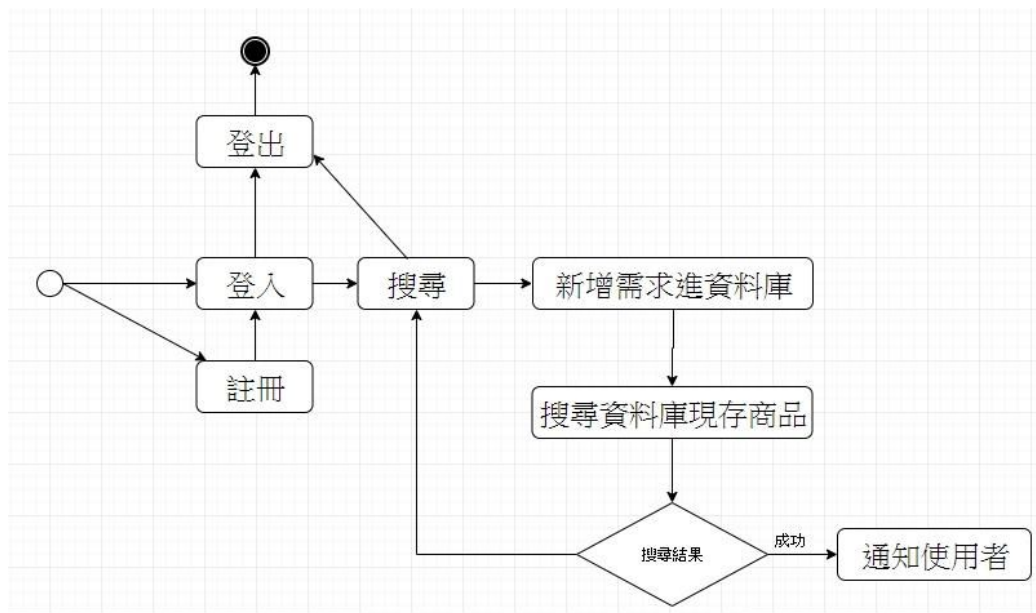
# 第三章 系統操作架構

## 3.1 執行流程

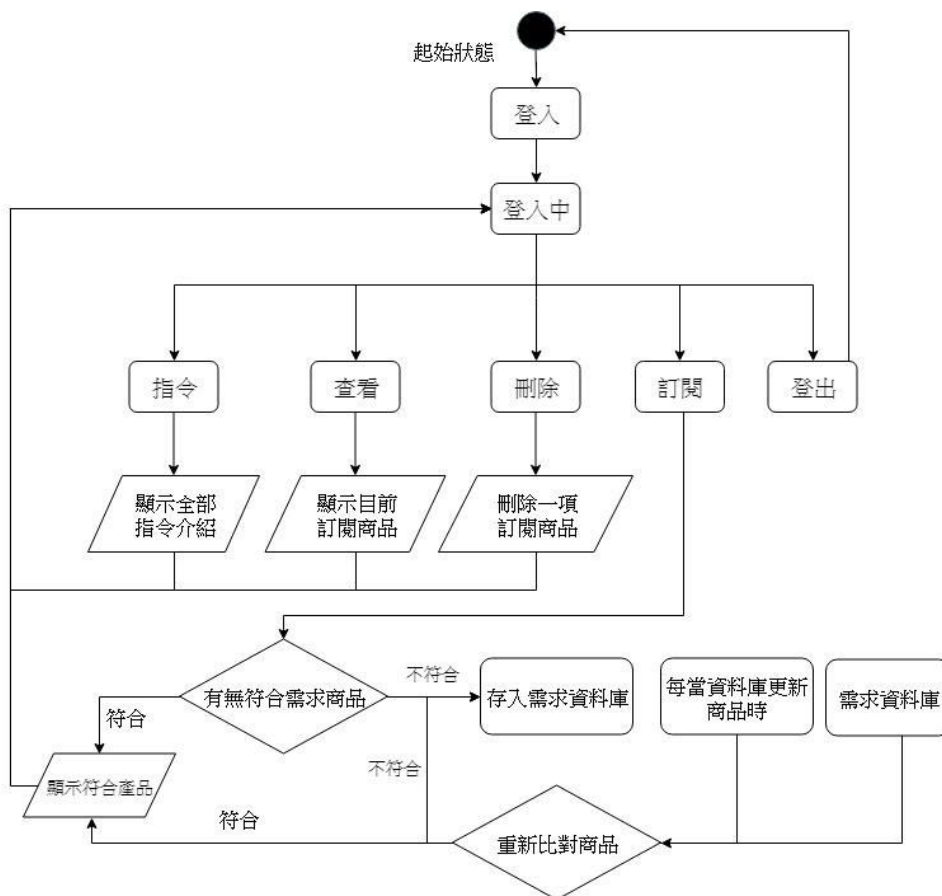
### 3.1.1系統主流程



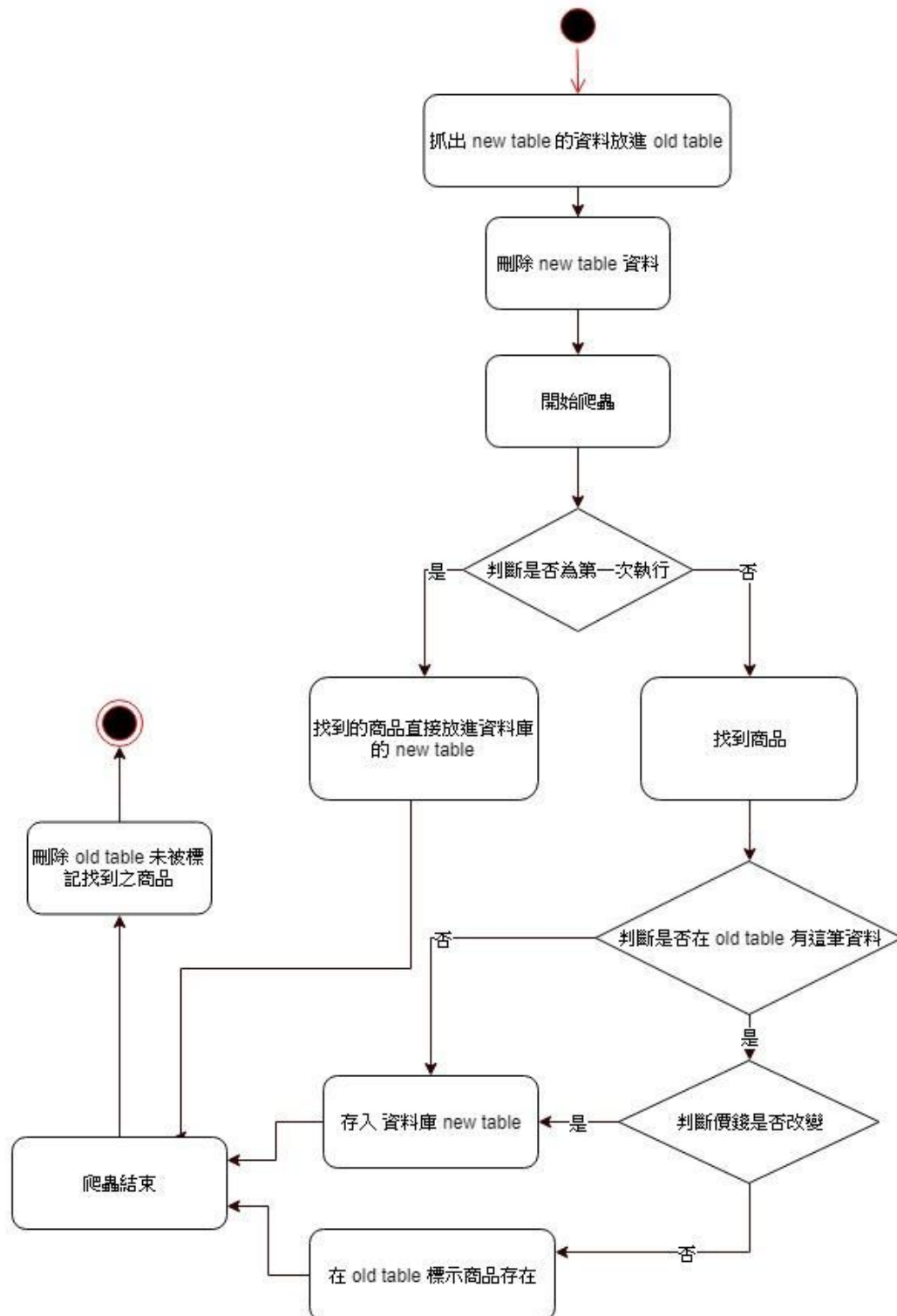
### 3.1.2 網頁執行流程



### 3.1.3 Line Chatbot執行流程



### 3.1.4 爬蟲執行流程



## 3.2 程式架構

### 3.2.1 搜尋及查詢資料庫(演算法)

#### a. 系統邏輯

1. 前端協作
2. 後端協作

main() 中執行的順序：

1. 呼叫爬蟲
2. 確認是否有新商品
3. 確認是否有新需求
4. 搜尋商品是否有符合需求的，有則寄信
5. 刪除到期需求(目前設定期限為兩周)

由於商品價格不會每天都改變，有的甚至一兩個月才會有改變，所以爬蟲這部分是採取 windows 工作排程器，定期呼叫程式去執行(目前設定一天一次)

#### b. 搜尋方法

##### 1. 字串分析(英文數字)

目前可搜尋商品分為三類：手機、筆電、平板，假設今天要搜尋的商品名稱為 iphone7 plus，在 iphone 跟 7 中間訂閱者輸入時並沒有空格，會造成後端在資料庫內搜尋商品時，會因為空格的關係，明明要比對到的產品，但因為字串比對不到，造成比對不精確，所以遇到數字的部份我們會進行切割，像是上面的例子會被切成 iphone、7 以及 plus 這三個字，在資料庫中商品字串要有符合這三個才會被比對到，英文的部分就是利用空格去做切割。

##### 2. 字串分析(中文)

中文的部分，因為要讓系統知道什麼樣單字、句字才會符合語意，所以我們利用「結巴分詞」(jieba)。

結巴分詞是目前大家認為做最好的 python 中文分詞套件，支援的模式有三種：「精確模式」、「全模式」以及「搜尋引擎模式」，精確模式顧名思義就是做最精確的切開，而全模式則是速度較快但不能解決歧義，搜尋引擎模式則是在精確的基礎上，對較長的詞語再次切分，提高召回率。

結巴有自己的字典(dict)，裡面有上萬的詞語，包含詞彙出現的次數和詞性，使用者也可以依照自己的需求，將裡面沒有的詞彙加入其中，或是改變出現次數等等，因為結巴分詞屬於概率語言模型，這個模型的任務就是在全切分所得的結果，求某個切分方案 S，使  $P(S)$  最大。

### 3.2.2 寄信與資料庫比對

#### 1.資料刪除

購物網站的商品如果下架，必須要在我們的資料庫商除這筆資料，避免訂閱者的需求搜尋到一筆不存在的資料。

#### 2.資料更新

商品價格如果有變動，我們並不會建立新的資料，而是更新資料庫舊有的資料。

#### 3.需求到期刪除

需求的搜尋期限為兩周，原本採用 (現在時間 - 發起需求的时间) 是否等於14，這樣的方式去做判斷，不過這樣要多做一步運算的步驟，雖然影響不是很大但多少還是會影響程式效率，所以最後就是在需求建立同時設定到期時間，只要比對到期時間是否等於現在時間。

#### 4.需求符合寄信

import python 原有套件 smtplib 來做寄信功能，符合的商品可能不只一件，所以會先把搜尋到的商品放入陣列，待全部比對完成後再做寄信，寄信內容為商品名稱、價錢以及網址，使用 gmail 來寄信。

#### 5.寄信通知:

當商品符合需求時，會以寄信方式來通知使用者。寄信的程式使用SMTP (Simple Mail Transfer Protocol) 的郵件傳輸協議，它是用一組源地址傳送到目的地址的傳送郵件規則。首先先建立SMTP連線，接著跟Gmail Sever進行溝通，並使用TLS的安全認證機制，然後登入Gmail傳輸資料到使用者的mail中，完成後關閉連線。



### 3.2.3 網路爬蟲演算法

#### a. 總流程及比對資料庫

1. 程式最開始會先初始化資料庫，如果是第一次執行則會建立資料庫以及所需要的 table(手機、筆電、平板)，裡面的欄位分別為：title, price, website, time, flag，前三個都是商品的資訊，time 則是爬到商品的時間，flag 是用來分辨是哪一個購物網站抓取下來的資料。
2. 接著執行 parse()，parse function 一開始會先執行 initFunc，這個 function 會把舊的資料抓出來，之後在爬蟲時會需要他們去比對是否已經存在，之後因為這邊使用 selenium，所以就會打開瀏覽器開始爬蟲。
3. 爬蟲找每一筆資料時，若是第一次執行則會無條件加入資料庫，否則會先和網址去比對是否有一樣的，如果沒有表示是新上架商品，直接存入資料庫的 new table，如果有再去比對價錢是否一樣，若一樣代表商品維持原樣且存在，若不一樣則代表價錢更改過，直接更新資料庫資料。
4. 判斷商品還在不在的方式是用一個變數 count，只要資料庫 old table 的資料有在這次爬蟲中被比對到，count 就設為 1，否則維持 0，在爬蟲結束之後會把 count = 0 的資料刪除，代表此商品已不再網頁上，這樣可以避免訂閱者需求找到已被刪除的頁面。

#### b. 爬蟲過程（露天）

1. parse() 會從 urls 陣列中取出欲爬取的網址，並用 for 迴圈去一一輸入瀏覽器。
2. 因為利用 selenium 模擬登入，所以會打開瀏覽器到接收到的網址，再使用 BeautifulSoup 套件去解析爬到的 html，方便我們去擷取需要的資訊。
3. 露天的瀏覽方式為換頁式，意即網址最後的數字為頁碼，所以只要將那個數字設定好範圍，利用迴圈下去跑，即可自己設定要爬的頁數。

#### c. 爬蟲過程（樂天）

1. parse() 會從 urls 陣列中取出欲爬取的網址，並用 for 迴圈去一一輸入瀏覽器。

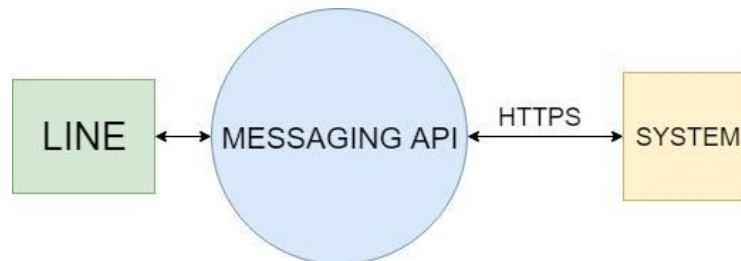
2. 因為利用 selenium 模擬登入，所以會打開瀏覽器到接收到的網址，再使用 BeautifulSoup 套件去解析爬到的 html，方便我們去擷取需要的資訊。
3. 樂天的瀏覽方式為換頁式，爬蟲資料爬到最後的頁數，因為頁數可能會增加或減少，所以有特別寫動態更換最後的頁數，防止錯誤發生。

d.爬蟲過程（PChome）

1. 使用 requests.get() 來抓下網頁的資料，再用json函數庫來將資料解析。
2. 將需要的數值從解析後的資料中一個個取出。
3. 雖然PChome24是使用滾輪式列表來顯示，但是網址卻是用頁碼來處理，所以只要將頁碼改成下一頁就能繼續重複抓取資料。

### 3.2.4 LINE Chatbot基礎架構

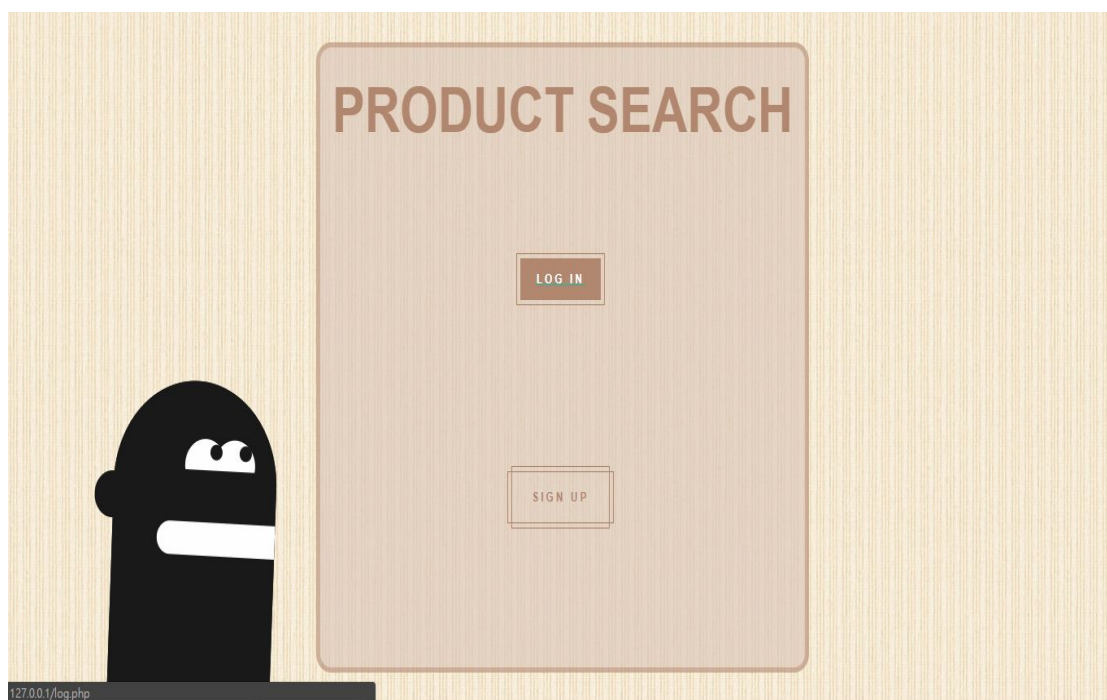
LINE Chatbot是LINE 提供的Messaging API（LINE公司的伺服器），它可以讓開發人員建立一個可接收並回覆訊息給好友的應用程式，也就是俗稱LINE BOT聊天機器人。LINE BOT的基本處理訊息的架構為下圖，Messaging API負責接收客戶端的訊息，並把接收的訊息傳給開發者的網站伺服器進行處理，伺服器端處理完後會回覆給Messaging API，並由它轉發相對應的訊息給客戶端。



# 第四章 結果與討論

## 4.1 前端畫面

### 4.1.1 首頁畫面



#### 4.1.2 會員註冊畫面



會員註冊

帳號:

密碼:

確認密碼:

信箱:

手機:

[回到首頁](#)

#### 4.1.3 登入畫面



登入

帳號

密碼

[回到首頁](#)

#### 4.1.4 商品搜尋

## LET'S SEARCH!

商品類型

平板

商品名稱

商品價格

搜尋

登出

#### 4.1.5 歷史結果

商品名稱	商品價錢	搜尋日期
asus	20000	2018-08-01 14:08:32
蘋果	12345	2018-08-13 20:42:32
華碩	2500	2018-08-17 12:01:54
聯想	21000	2018-08-24 09:24:36
華碩	132	2018-09-18 10:50:42
華碩	32000	2018-09-25 14:04:40



## 4.2 Line ChatBot 介面

### 4.2.1 基本操作選單:

進入Chatbot後底下點選「操作指令集」，目前有六個功能可以使用，分別是訂閱、查看、刪除、指令、登入和登出。



### 4.2.2 指令:

點選「指令」後，Chatbot會告知此程式的操作指令說明。



#### 4.2.3 登入:

點選「指令」後，必須填入網站註冊帳戶的帳號和密碼，才能使用Chatbot的功能。Chatbot設定為一個帳戶只能在一個裝置登入使用。



#### 4.2.4 登出:

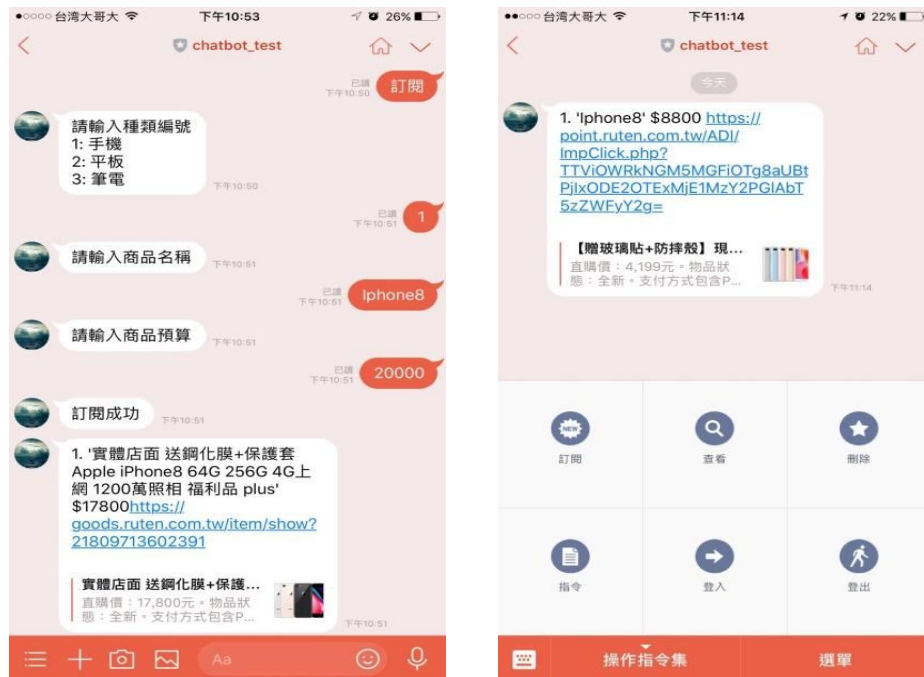
點選「登出」後，可以在別的裝置重新登入或登入另一個帳戶。





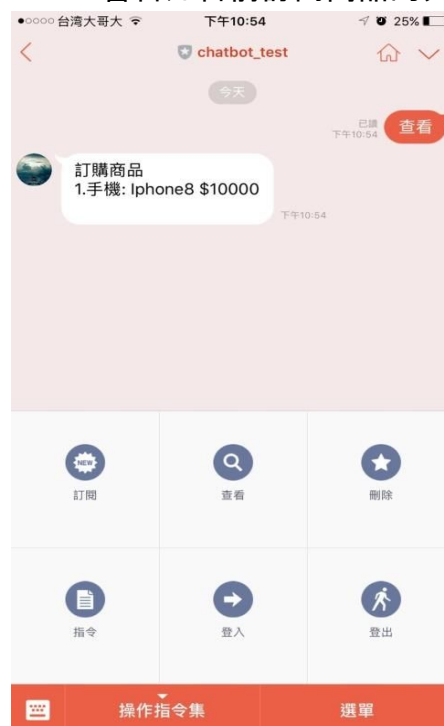
#### 4.2.5 訂閱:

點選「訂閱」後，輸入相對應的文字即可完成訂閱商品功能。當首次完成訂閱功能，如果當下有符合需求的商品，Chatbot會立即回覆；如果沒有的話，Chatbot會直到搜尋到符合商品才回覆訊息。



#### 4.2.6 查看:

點選「查看」後，Chatbot會告知目前訂閱商品的列表。



#### 4.2.7 刪除:

點選「刪除」後，輸入相對應的刪除編號即可完成刪除訂閱的商品。



### 4.3 寄信範例



### 4.4 結論

現今的購物環境已逐漸從實體店面轉移到網路購物平台，目前又有許多網路平台和電商的崛起，對於消費者來說是更方便來購買商品，但相對會對商品進行比價，以此來減輕荷包壓力。往往進行比價後的價格不是很滿意，並不符合當時消費者的預期價格，而消費者又要時時刻刻查詢他的需求商品是否能達到他的預期價格是不明智的行為。

因此本專題使用爬蟲加上排程工作器來進行監控商品價格的變化，透過演算法計算網站的商品價格是否符合使用者的預期價格，來幫助使用者找尋與需求最接近的商品。本系統除了有網頁介面之外，還有開發手機即時通訊介面——LINE來進行操作，讓系統更加普及大眾化，且節省很多時間讓使用者不用每天反覆去搜尋比對價格，系統會自動搜尋直到需求商品達到使用者的預期並透過信箱和LINE來通知使用者，不論是時間上或是金錢方面的幫助都是一大助益。

此系統除了可以應用在網路購物平台之外，未來也可以運用在智慧家庭上，讓使用者不必時時關注在產品身上，只要達到預期目標再通知使用者，為人類的生活節省了多餘時間，更專注在必要的行動上。

# 參考文獻

[1] 學習WEB開發:

[https://developer.mozilla.org/zh-CN/docs/Learn/CSS/Introduction\\_to\\_CSS](https://developer.mozilla.org/zh-CN/docs/Learn/CSS/Introduction_to_CSS)  
W3school : <https://www.w3schools.com/>

[2] PHP:

<https://ithelp.ithome.com.tw/articles/10191950>

[3] PHP教學:

<http://emn178.pixnet.net/blog/post/78014066-php%E6%95%99%E5%AD%B8---%E5%9F%BA%E6%9C%AC%E8%AA%9E%E6%B3%95%28syntax%29>

[4] 六角學院:

<https://www.hexschool.com/>

[5] Messaging API

<https://developers.line.me/en/docs/messaging-api/overview/>

[6] Line Message API with NodeJS

<http://yonaichin.logdown.com/posts/1256421-line-message-api-with-nodejs>

[7] LineBot+Python, 輕鬆建立聊天機器人

<https://yaoandy107.github.io/line-bot-tutorial/>

[8] 只要有心, 人人都可以做卡米狗

<https://ithelp.ithome.com.tw/users/20107309/ironman/1253>

[9] 網路爬蟲實戰教學

<https://www.youtube.com/watch?v=ceUhb2-gYOU&list=PLohb4k71XnPaQRTvKW4Uii1oq-JPGpwWF>

[10] re---Regular expression operation

<https://docs.python.org/3/library/re.html>

[11] Python 正則表達式

<http://www.runoob.com/python/python-reg-expressions.html>

[12] python smtp lib

[https://self.jx-tsai.info/2016/09/python\\_22.html](https://self.jx-tsai.info/2016/09/python_22.html)

[13] selenium 莫煩

<https://morvanzhou.github.io/tutorials/data-manipulation/scraping/5-01-selenium/>

[14] Mysql 遠端連線

<https://www.phpini.com/mysql/mysql-enable-remote-connect>

[15] Python pandas

<https://jerryneest.io/python-pandas-get-data/>

[16] Python sqlite

<http://www.runoob.com/sqlite/sqlite-python.html>

[17] python bs4

<https://blog.gtwang.org/programming/python-beautiful-soup-module-scrape-web-pages-tutorial/>

[18] requests網站

<http://docs.python-requests.org/en/master/>

[19] Python 爬蟲實作

<http://ivanjo39191.pixnet.net/blog/post/70548675-python-%E7%88%AC%E8%9F%>

B2%E7%B7%B4%E7%BF%92%E7%B4%80%E9%8C%842%28%E4%B8%80%29

[20] Python網路爬蟲

.<https://www.youtube.com/watch?v=YHc6KvLsv20>

[21] sqlite 語法教學

<http://work.oknow.org/2014/02/python-sqlite.html>

# 附件A

## 爬蟲(PChome)

```
import time
import datetime
import sqlite3
import requests
import json
from bs4 import BeautifulSoup

class database():
    #初始化DB,name是創建資料庫的名字
    def __init__(self, name):
        try:
            self.db_name = './Database/' + name + '.sqlite'
            self.conn = sqlite3.connect(self.db_name)
            print("%s database create success" % self.db_name)
            print("Database connected")
            self.cur = self.conn.cursor()
            self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
            self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
            self.conn.close()
        #
            print("database connect close")
        except sqlite3.Error as e:
            print("Database error: %s" % e)
        except Exception as e:
            print("Exception in _query: %s" % e)
    #插入資料,參數是二維list跟table名字
    def insert_items(self, items, table):
        self.cur.execute("select * from new where flag = '3';")
        data = self.cur.fetchall()
        #print(data)
        insert = "INSERT INTO old VALUES (?, ?, ?, ?, ?)"
        self.cur.executemany(insert, data)
        self.cur.execute("delete FROM new where flag = '3';")
        self.conn.commit()
        self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')

        self.check_repeat(items, 'old')
        insert = "INSERT INTO new VALUES (?, ?, ?, ?, ?)"
        self.cur.executemany(insert, items)
        self.conn.commit()
        print("Data insert success")
    #檢查新資料與舊資料有沒有重複,把重複的從新資料刪掉
    def check_repeat(self, items, table_name):
        #sql = 'select * from '+ table_name + ' order by price'
```

```

        sql = 'select * from ' + table_name + " where flag = '3';"
        self.cur.execute(sql)
        data = self.cur.fetchall()
        #print(self.cur.fetchall())
        for old_item in data:
            for new_item in items:
                if(old_item[2] == new_item[2]):
                    if(old_item[1] == new_item[1]):
                        self.conn.execute("UPDATE old set date = ?
where website = ?", (new_item[3], new_item[2]))
                        items.remove(new_item)
                    else:
                        self.conn.execute("DELETE from " + table_name
+" where website = '" + old_item[2] +"' ;")
            #print(items)
        def show_data(self, table_name):
            sql = 'select * from ' + table_name + ' order by price'
            self.cur.execute(sql)
            print(self.cur.fetchall())
        def delete (self):
            self.cur.execute('DROP TABLE new')
            self.cur.execute('DROP TABLE old')

```

```
# In[38]:
```

```

class crawler():
    def __init__(self):
        #網站網址
        self.main = ''
        #要搜尋的商品
        self.commodity = ''
        #物品二維LIST
        self.items = []
        self.count = 0
        #將商品爬下並放到items
        def search_items(self, commodity):
            main = 'http://24h.pchome.com.tw/prod/'
            max_page = 10
            mini_page = 1
            for page in range(mini_page, max_page):
                time.sleep(3)
                GGBB =
'https://ecshweb.pchome.com.tw/search/v3.3/all/results?q=' +
commodity + '&page=' + str(page) + '&sort=rnk/dc'
                #resp =
'https://ecshweb.pchome.com.tw/search/v3.3/all/results?q=%E7%AD%86%E9
%9B%BB&page=' + str(page) + '&sort=rnk/dc'
                res = requests.get(GGBB)

```

```

        soup = BeautifulSoup(res.text, "html.parser")
        #print(res.text)
        try:
            data = json.loads(res.text)
            now = datetime.datetime.now()
            for item in data['prods']:
                itemdata = []
                itemdata.append(item['name'])
                itemdata.append(item['price'])
                url = main+item['Id']
                itemdata.append(url)
                itemdata.append('%s/%s/%s %s:%s:%s' %
(now.year,now.month,now.day,now.hour, now.minute, now.second))
                itemdata.append('3')
                self.items.append(itemdata)
                self.count = self.count + 1
        except :
            pass
        print("%d items searched"%self.count)
    def show_items (self):
        print(self.items)

```



## 爬蟲(露天)

```
from bs4 import BeautifulSoup
import scrapy
from selenium import webdriver
import time
import lxml
import sys
from scrapy.selector import Selector
import requests
import shutil
import sqlite3 as lite
import os
import json
import re
import string
import datetime
import numpy as np

def strclear(text,newsign='') :
    signtext = string.punctuation + newsign
    signrepl = '@'*len(signtext)
    signtable = str.maketrans(signtext,signrepl)
    return text.translate(signtable).replace('@','')

class parseRuten(scrapy.Spider):
    name = 'parseRuten'
    domain = ['http://www.ruten.com.tw/']
    start_urls = [

        'http://class.ruten.com.tw/category/sub00.php?c=00210001&p=', #手機空
        機

        'http://class.ruten.com.tw/category/sub00.php?c=00110002&p=', #筆記電
        腦

        'http://class.ruten.com.tw/category/sub00.php?c=00110013&p='
        #平板
    ]
    sql = "insert into old(title,price,website,date,flag)
values(?,?,?,?,?)"
    sqlnew = "insert into new(title,price,website,date,flag)
values(?,?,?,?,?)"
    urlclass = "new"
    databaseclass = "old"
    urlclasscount = 0
    urlpage = 0
    db_title = []
    db_price = []
    db_web = []
    db_date = []
    db_flag = []
    db_intitle = []
    db_inprice = []
```

```

db_inweb = []
db_indate = []
db_inflag = []
db_incheck = []
flag = 0
count = 0
Header = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181
Safari/537.36"
}
def __init__(self):
    self.conn = lite.connect('./Database/cellphone.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
    self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
    self.conn.close()
    self.conn = lite.connect('./Database/notebook.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
    self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
    self.conn.close()
    self.conn = lite.connect('./Database/pad.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
    self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
    self.conn.close()
    print("init")
    self.urlclasscount = 0
    def initFunc(self):
        try:
            if(self.urlclasscount ==1):
                self.conn =
lite.connect('./Database/cellphone.sqlite')
                self.cur = self.conn.cursor()
                self.cur.execute("SELECT * FROM new where flag =
'1';")
                self.urlclasscount = 1
            elif(self.urlclasscount==2):
                self.conn =
lite.connect('./Database/notebook.sqlite')
                self.cur = self.conn.cursor()
                self.cur.execute("SELECT * FROM new where flag =
'1';")
                self.urlclasscount = 2

```

```

elif(self.urlclasscount==3):
    self.conn = lite.connect('./Database/pad.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute("SELECT * FROM new where flag =
'1';")

    self.urlclasscount = 3
results = []
results = self.cur.fetchall()
#print(results)
print("select item from new success")
if(len(results)!=0):
    print("exist")
    #print(self.urlclass[self.urlclasscount-1])
    self.cur.execute("delete FROM new where flag='1';")
#拿出之後把new裡面的商品都刪掉
    self.conn.commit()
    print("delete current new success")
    for i in range(len(results)):
        #將商品放進cellphone

self.cur.execute(self.sql, (results[i][0],results[i][1],results[i][2],
results[i][3], '1'))

    self.conn.commit()
    print("success insert data from new to cellphone")
    self.conn.close()
    if(self.urlclasscount ==1):
        self.conn =
lite.connect('./Database/cellphone.sqlite')
#檢查new裡面有
無商品，若有就拿出
        self.cur = self.conn.cursor()
        self.cur.execute("SELECT * FROM old where flag='1';")
        self.urlclasscount = 1
        #self.conn.close()
    elif(self.urlclasscount==2):
        self.conn =
lite.connect('./Database/notebook.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute("SELECT * FROM old where flag='1';")
        self.urlclasscount = 2
        #self.conn.close()
    elif(self.urlclasscount==3):
        self.conn = lite.connect('./Database/pad.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute("SELECT * FROM old where flag='1';")
        self.urlclasscount = 3
nowProd = self.cur.fetchall()
for product in nowProd:
    self.db_intitle.append(product[0])
    self.db_inprice.append(product[1])
    self.db_inweb.append(product[2])
    self.db_indate.append(product[3])

```

```

        self.db_incheck.append(0)
        #print(len(self.db_inweb))
#目前資料總數
    except:
        print("Error: can't take product from database")
        self.conn.close()
        print("database close")
    def parse(self):
        if(sys.argv[1] == "chrome"):
            chrome_path = "D:\chromedriver.exe"
            self.driver = webdriver.Chrome(chrome_path) #chromedriver
        elif(sys.argv[1] == "firefox"):
            self.driver = webdriver.Firefox()
        for start in self.start_urls:
            self.urlclasscount += 1
            #print(self.urlclasscount)
            parseRuten.initFunc(self)
            for page in range(0,1):#how many pages want spider
                url = start + str(page+1)
                self.driver.get(url)
                sp = BeautifulSoup(self.driver.page_source, "lxml")
                date =
datetime.datetime.now().strftime("%Y-%m-%d-%H-%M")
                for items in sp.select('.prod_name'):
                    self.db_web.append(items.select('a')[0]['href'])
                    self.db_title.append(items.select('a')[0])
                for items in sp.select('.price_box'):
                    self.db_price.append(items.select('.price'))
                for i in range(len(self.db_title)):
                    web_temp = self.db_web[i]
                    title_temp = self.db_title[i]
                    price_temp = self.db_price[i]
                    pric = "" + "','".join(map(str, price_temp)) +
'''
                    title = "" + "','".join(map(str, title_temp)) +
'''
                    #title
                    #website = "" + "','".join(map(str, web_temp)) +
'''
                    #website
                    website = web_temp
                    m = re.search('\<span
class="price">(.*?)</span>',pric)
                    price = m.group(1)
                    price = int(strclear(price,','))
#price
                    if(len(self.db_inweb)==0):
#第一次執行, cellphone裡面沒任何資料
                    if(self.urlclasscount==1):
                        self.conn =
lite.connect('./Database/cellphone.sqlite')
                        elif(self.urlclasscount==2):
                            self.conn =

```

[illegible]

```

self.cur.execute(self.sqlnew, (title, price, website, date, '1'))

        self.conn.commit()
        self.count = 0
    else:
        self.count = 0
        self.db_title.clear()
#初始化暫存
        self.db_price.clear()
        self.db_web.clear()
        self.db_date.clear()
        self.db_flag.clear()
        self.conn.close()
        print('db close')
        if (len(self.db_inweb) != 0):
            for i in range(len(self.db_inweb)):
#若資料庫資料完全沒被比對到，就刪除那筆資料
                if (self.db_incheck[i] == 0):
                    if (self.urlclasscount == 1):
                        self.conn =
lite.connect('./Database/cellphone.sqlite')
                        self.cur = self.conn.cursor()
                    elif (self.urlclasscount == 2):
                        self.conn =
lite.connect('./Database/notebook.sqlite')
                        self.cur = self.conn.cursor()
                    elif (self.urlclasscount == 3):
                        self.conn =
lite.connect('./Database/pad.sqlite')
                        self.cur = self.conn.cursor()

#print(self.databaseclass[self.urlclasscount-1])
        self.cur.execute("delete from old where
website = '%s' " % self.db_inweb[i])
        self.conn.commit()
        #print("the item %s was not exist" %
self.db_intitle[i])
        self.db_intitle.clear()
        self.db_inprice.clear()
        self.db_inweb.clear()
        self.db_indate.clear()
        self.db_inflag.clear()
        self.db_incheck.clear()
        self.conn.close()
        print("end of program")
        self.driver.close()

```

## 爬蟲(樂天)

```
from bs4 import BeautifulSoup
import scrapy
from selenium import webdriver
import time
import lxml
import sys
from scrapy.selector import Selector
import requests
import shutil
import sqlite3 as lite
import os
import json
import re
import string
import datetime
import numpy as np

def strclear(text, newsign = ''):
    signtext = string.punctuation + newsign
    signrepl = '@'*len(signtext)
    signtable = str.maketrans(signtext, signrepl)
    return text.translate(signtable).replace('@', '')

class parseRuten(scrapy.Spider):
    name = 'parseRuten'
    domain = ['http://www.ruten.com.tw/']
    start_urls = [

        'http://class.ruten.com.tw/category/sub00.php?c=00210001&p=', #手機空
        機

        'http://class.ruten.com.tw/category/sub00.php?c=00110002&p=', #筆記電
        腦

        'http://class.ruten.com.tw/category/sub00.php?c=00110013&p='
        #平板
    ]

    sql = "insert into old(title,price,website,date,flag)
values(?,?,?,?,?)"
    sqlnew = "insert into new(title,price,website,date,flag)
values(?,?,?,?,?)"
    urlclass = "new"
    databaseclass = "old"
    urlclasscount = 0
    urlpage = 0
    db_title = []
    db_price = []
    db_web = []
    db_date = []
    db_flag = []
    db_intitle = []
    db_inprice = []
```

```

db_inweb = []
db_indate = []
db_inflag = []
db_incheck = []
flag = 0
count = 0
Header = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181
Safari/537.36"
}
def __init__(self):
    self.conn = lite.connect('./Database/cellphone.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
    self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
    self.conn.close()
    self.conn = lite.connect('./Database/notebook.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
    self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
    self.conn.close()
    self.conn = lite.connect('./Database/pad.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute('create table if not exists old(title
text,price integer,website text,date timestamp,flag);')
    self.cur.execute('create table if not exists new(title
text,price integer,website text,date timestamp,flag);')
    self.conn.close()
    print("init")
    self.urlclasscount = 0
    def initFunc(self):
        try:
            if(self.urlclasscount ==1):
                self.conn =
lite.connect('./Database/cellphone.sqlite')
                self.cur = self.conn.cursor()
                self.cur.execute("SELECT * FROM new where flag =
'1';")
                self.urlclasscount = 1
            elif(self.urlclasscount==2):
                self.conn =
lite.connect('./Database/notebook.sqlite')
                self.cur = self.conn.cursor()
                self.cur.execute("SELECT * FROM new where flag =
'1';")
                self.urlclasscount = 2

```



```

elif(self.urlclasscount==3):
    self.conn = lite.connect('./Database/pad.sqlite')
    self.cur = self.conn.cursor()
    self.cur.execute("SELECT * FROM new where flag =
'1';")

    self.urlclasscount = 3
results = []
results = self.cur.fetchall()
#print(results)
print("select item from new success")
if(len(results)!=0):
    print("exist")
    #print(self.urlclass[self.urlclasscount-1])
    self.cur.execute("delete FROM new where flag='1';")
#拿出之後把new裡面的商品都刪掉
    self.conn.commit()
    print("delete current new success")
    for i in range(len(results)):
        #將商品放進cellphone

self.cur.execute(self.sql, (results[i][0],results[i][1],results[i][2],
results[i][3], '1'))

    self.conn.commit()
    print("success insert data from new to cellphone")
    self.conn.close()
    if(self.urlclasscount ==1):
        self.conn =
lite.connect('./Database/cellphone.sqlite')
#檢查new裡面有
無商品，若有就拿出
        self.cur = self.conn.cursor()
        self.cur.execute("SELECT * FROM old where flag='1';")
        self.urlclasscount = 1
        #self.conn.close()
    elif(self.urlclasscount==2):
        self.conn =
lite.connect('./Database/notebook.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute("SELECT * FROM old where flag='1';")
        self.urlclasscount = 2
        #self.conn.close()
    elif(self.urlclasscount==3):
        self.conn = lite.connect('./Database/pad.sqlite')
        self.cur = self.conn.cursor()
        self.cur.execute("SELECT * FROM old where flag='1';")
        self.urlclasscount = 3
nowProd = self.cur.fetchall()
for product in nowProd:
    self.db_intitle.append(product[0])
    self.db_inprice.append(product[1])
    self.db_inweb.append(product[2])
    self.db_indate.append(product[3])

```

```

        self.db_incheck.append(0)
        #print(len(self.db_inweb))
#目前資料總數
    except:
        print("Error: can't take product from database")
        self.conn.close()
        print("database close")
    def parse(self):
        if(sys.argv[1] == "chrome"):
            chrome_path = "D:\chromedriver.exe"
            self.driver = webdriver.Chrome(chrome_path) #chromedriver
        elif(sys.argv[1] == "firefox"):
            self.driver = webdriver.Firefox()
        for start in self.start_urls:
            self.urlclasscount += 1
            #print(self.urlclasscount)
            parseRuten.initFunc(self)
            for page in range(0,1):#how many pages want spider
                url = start + str(page+1)
                self.driver.get(url)
                sp = BeautifulSoup(self.driver.page_source, "lxml")
                date =
datetime.datetime.now().strftime("%Y-%m-%d-%H-%M")
                for items in sp.select('.prod_name'):
                    self.db_web.append(items.select('a')[0]['href'])
                    self.db_title.append(items.select('a')[0])
                for items in sp.select('.price_box'):
                    self.db_price.append(items.select('.price'))
                for i in range(len(self.db_title)):
                    web_temp = self.db_web[i]
                    title_temp = self.db_title[i]
                    price_temp = self.db_price[i]
                    pric = "" + "','".join(map(str, price_temp)) +
'''
                    title = "" + "','".join(map(str, title_temp)) +
'''
                    #title
                    #website = "" + "','".join(map(str, web_temp)) +
'''
                    #website
                    website = web_temp
                    m = re.search('\<span
class="price">(.*?)</span>',pric)
                    price = m.group(1)
                    price = int(strclear(price,','))
#price
                    if(len(self.db_inweb)==0):
#第一次執行, cellphone裡面沒任何資料
                    if(self.urlclasscount==1):
                        self.conn =
lite.connect('./Database/cellphone.sqlite')
                        elif(self.urlclasscount==2):
                            self.conn =

```

[illegible]

```

self.cur.execute(self.sqlnew, (title, price, website, date, '1'))

        self.conn.commit()
        self.count = 0
    else:
        self.count = 0
    self.db_title.clear()
#初始化暫存
    self.db_price.clear()
    self.db_web.clear()
    self.db_date.clear()
    self.db_flag.clear()
    self.conn.close()
    print('db close')
    if (len(self.db_inweb) != 0):
        for i in range(len(self.db_inweb)):
#若資料庫資料完全沒被比對到，就刪除那筆資料
            if (self.db_incheck[i] == 0):
                if (self.urlclasscount == 1):
                    self.conn =
lite.connect('./Database/cellphone.sqlite')
                    self.cur = self.conn.cursor()
                elif (self.urlclasscount == 2):
                    self.conn =
lite.connect('./Database/notebook.sqlite')
                    self.cur = self.conn.cursor()
                elif (self.urlclasscount == 3):
                    self.conn =
lite.connect('./Database/pad.sqlite')
                    self.cur = self.conn.cursor()

#print(self.databaseclass[self.urlclasscount-1])
        self.cur.execute("delete from old where
website = '%s' " % self.db_inweb[i])
        self.conn.commit()
        #print("the item %s was not exist" %
self.db_intitle[i])
        self.db_intitle.clear()
        self.db_inprice.clear()
        self.db_inweb.clear()
        self.db_indate.clear()
        self.db_inflag.clear()
        self.db_incheck.clear()
        self.conn.close()
        print("end of program")
    self.driver.close()

```

## 主程式

```
import re
import time
import sched
import string
import logging
import pymysql
import smtplib
import datetime
import threading
import CrawlFunc
import webbrowser
import sqlite3 as lite
from datetime import datetime
from selenium import webdriver
from line_bot import ConvertData
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from parseWeb import parseRute,parseRaku,parsePc

def main():
    def job(num):
        if num == 0:
            rute = parseRute.parseRuten()
            rute.parse()
        elif num == 1:
            raku = parseRaku.tablet()
            raku.start_requests()
            pass
        elif num == 2:
            cellphone = parsePc.crawler()
            cellphone.search_items('手機')
            time.sleep(5)
            notebook = parsePc.crawler()
            notebook.search_items('筆電')
            time.sleep(5)
            pad = parsePc.crawler()
            pad.search_items('平板')
            cell = parsePc.database('cellphone')
            cell.conn = lite.connect(cell.db_name)
            cell.cur = cell.conn.cursor()
            cell.insert_items(cellphone.items, 'new')
            cell.conn.close()
            note = parsePc.database('notebook')
            note.conn = lite.connect(note.db_name)
            note.cur = note.conn.cursor()
            note.insert_items(notebook.items, 'new')
            note.conn.close()
            paddd = parsePc.database('pad')
            paddd.conn = lite.connect(paddd.db_name)
```

```

        paddd.cur = paddd.conn.cursor()
        paddd.insert_items(pad.items, 'new')
        paddd.conn.close()

    def func():
        try:
            t1 = threading.Thread(target=job, args =
(0,) ,name='test1')
            t2 = threading.Thread(target=job, args =
(1,) ,name='test2')
            t3 = threading.Thread(target=job, args =
(2,) ,name='test3')
            #thread 沒被 kill, 所以用同一 thread 工作
            if t1.isAlive() is False:
                t1 = threading.Thread(target=job, args =
(0,) ,name='test1')
                t1.start()

            if t2.isAlive() is False:
                t2 = threading.Thread(target=job, args =
(1,) ,name='test2')
                t2.start()

            if t3.isAlive() is False:
                t3 = threading.Thread(target=job, args =
(2,) ,name='test3')
                t3.start()
            #join確定所有 thread 都執行完, 再繼續主執行緒
            t1.join()
            t2.join()
            t3.join()
        except:
            logging.error("Error: parse error")
        #-----連接 Database-----
        try:
            db =
pymysql.connect(host="140.118.155.126",user="test",passwd="123",db="u
ser_infor")

            cursor = db.cursor()
            cursor.execute("SELECT * FROM
user_infor.searchtable")
            results = cursor.fetchall()
            cursor.execute("delete From user_infor.searchtable
where dueTime = DATE(NOW());")
            db.commit()
            db.close()
            print("STEP 2")
        except:
            logging.error("Error: mysql connected error")
        #-----找request相符之商品並寄信-----
        try:
            t = CrawlFunc.check_request(results)

```

```

        if t ==0:
            print("no request today")
        print("STEP 3")
    except:
        logging.error("Error: check_request error")
        timer = threading.Timer(180, func) #每3分鐘執行一次
        timer.start()

def getMysql():
    conn = lite.connect('./Database/mysqlDB.sqlite')
    cur = conn.cursor()
    cur.execute("SELECT * FROM subDB")
    db =
pymysql.connect(host="140.118.155.126",user="test",passwd="123",db="u
ser_infor") #ip要改
    cursor = db.cursor()
    cursor.execute("SELECT * FROM user_infor.searchtable")
    prodData = cursor.fetchall()
    cursor.execute("SELECT * FROM
user_infor.userinformation")
    userData = cursor.fetchall()
    lineSub = cur.fetchall()
    #-----檢查 mysql, sqlite 有無更動-----
    global productC,userC, lineC
    if productC!=len(prodData) or userC!=len(userData) or
lineC!=len(lineSub):
        ConvertData.passDatabase()
    else:
        print("continue")
        userC = len(userData)
        lineC = len(lineSub)
        productC = len(prodData)
        print(userC)
        print(productC)
        print(lineC)
        db.close()
        conn.close()
        threading.Timer(10,getMysql).start() #每10秒執行一次
    conn = lite.connect('./Database/mysqlDB.sqlite')
    cur = conn.cursor()
    db =
pymysql.connect(host="140.118.155.126",user="test",passwd="123",db="u
ser_infor") #ip要改
    cursor = db.cursor()
    cur.execute("SELECT * FROM subDB")
    cursor.execute("SELECT * FROM user_infor.searchtable")
    cursor.execute("SELECT * FROM user_infor.userinformation")
    prodData = cursor.fetchall()
    userData = cursor.fetchall()
    lineSub = cur.fetchall()
    db.close()
    conn.close()

```

```
global productC, userC, lineC
productC = len(prodData)
userC = len(userData)
lineC = len(lineSub)
threads = []
#初始化 threads, target 分别是 job0, job1, job2
for i in range(3):
    threads.append(threading.Thread(target = job, args =
(i,)))
    func()
    getMysql()
if __name__ == "__main__":
    main()
```



## 字串比對、寄信、資料庫操作

```
import re
import time
import string
import logging
import pymysql
import smtplib
import datetime
import pandas as pd
import sqlite3 as lite
from datetime import datetime
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

def check_request(results):
    db =
pymysql.connect(host="140.118.155.126",user="test",passwd="123",db="u
ser_infor") #ip要改
    cursor = db.cursor()
    if(results):
        for r in results:
            if(r[9] != 1) :
                newwant = check_newtable(r)
                if r[7]==0:
                    oldwant = check_oldtable(r)
                    newwant = newwant.append(oldwant)
                if(len(newwant) == 0):
                    print("request no fit")
                else:
                    title,price,website =
checkProd(r[3],r[4],newwant)
                    if len(title)!=0:
                        context = ""
                        cursor.execute("DELETE FROM
user_infor.searchtable where id = '%s' " %int(r[0]))
                        for j in range(len(title)):
                            url = website[j]
                            ti = title[j]
                            context+=str(j+1) + ti + " \t\t (" + url
+ ") \n"
                        sendemail(r[2],"Dear " + r[1] + ", \n\n" +
"Your product is coming.\n" + context)
                    else:
                        print("not found")
            else:
                print("isline")
```

```

        cursor.execute("UPDATE user_infor.searchtable SET flag = 1")
        db.commit()
        db.close()
        return 1
    else:
        return 0
#比對newtable
def check_newtable(r):
    if r[6] == 1:
        DB_name = "./Database/pad.sqlite"
    elif r[6] == 2:
        DB_name = "./Database/cellphone.sqlite"
    elif r[6] == 3:
        DB_name = "./Database/notebook.sqlite"
    dfnew = []
    topPrice = r[4] + r[4]/5
    lowPrice = r[4] - r[4]/5
    conn = lite.connect(DB_name)
    dfnew = pd.read_sql_query("SELECT * FROM new WHERE price < %d and
price > %d" %(topPrice,lowPrice), conn)
    conn.close()
    return dfnew
#比對oldtable
def check_oldtable(r):
    if r[6] == 1:
        DB_name = "./Database/pad.sqlite"
    elif r[6] == 2:
        DB_name = "./Database/cellphone.sqlite"
    elif r[6] == 3:
        DB_name = "./Database/notebook.sqlite"
    dfold = []
    topPrice = r[4] + r[4]/5
    lowPrice = r[4] - r[4]/5
    conn = lite.connect(DB_name)
    dfold = pd.read_sql_query("SELECT * FROM old WHERE price < %d and
price > %d" %(topPrice,lowPrice), conn)
    conn.close()
    return dfold

def checkProd(searchWord,expPrice,df):
    title = []
    price = []
    website = []
    n = 0
    base = expPrice/5 #價錢基準化
    for char in string.punctuation: #去掉字串中符號
        searchWord = searchWord.replace(char, '')
    searchWord = searchWord.replace(" ", "") #刪掉字串空白
    b = re.split(r'([\d+])', searchWord,maxsplit=5)
    for i in range(len(df.values)): #查看資料庫所有
資料

```

```

        for k in range(len(b)): #字串切割後的
len
        if(b[k]!=""): #不看list中空
白部分
            if(re.search(b[k],df.values[i][0],re.IGNORECASE)): #比
對字串中符合之字串
                n = n + 1
            else:
                n = n + 1
            if(n==len(b)):
                if(df.values[i][1] >= expPrice-base and df.values[i][1]
<= expPrice+base): #字串符合比對價錢是否符合
                    title+=df.values[i][0]
                    price+=df.values[i][1]
                    website+=df.values[i][2]

n = 0
return title, price, website

```

```

def sendemail(sendperson,sendcontext): #sendperson要寄的email位置,
sendcontext寄信內容

```

```

    try:
        #gmail信箱的資訊
        host = "smtp.gmail.com"
        port = 587
        username = "SendingRobotkk@gmail.com"
        password = "52785278"
        from_email = username
        the_msg = MIMEText("alternative")
        to_list = sendperson

        the_msg['Subject'] = "親愛的用戶，找到商品囉"
        the_msg["From"] = from_email
        the_msg["To"] = sendperson

        plain_txt = sendcontext + " \nBest regards,\n" #內文
        context = MIMEText(plain_txt, 'plain', 'utf-8')

        the_msg.attach(context)
        # 建立SMTP連線
        email_conn = smtplib.SMTP(host,port)
        #跟Gmail Server溝通
        print(email_conn.ehlo())
        # TTLS安全認證機制
        email_conn.starttls()
        #登錄Gmail
        print(email_conn.login(username,password))
        #寄信
        email_conn.sendmail(from_email, to_list,
the_msg.as_string()) #寄信內容
        #關閉連線

```

```
        email_conn.quit()
    except:
        logging.error("Error: email sending error.")
```

# 附件B

## 網頁首頁

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Product search</title>
    <link rel="stylesheet" href="css/flat-ui.css">
    <link rel="stylesheet" href="css/all.css">
    <script type="text/javascript"
src="js/flat-ui.min.js"></script>
    <script src="js/jquery-3.3.1.min.js"></script>
    <script src="js/all.js"></script>
  </head>
</br>
<style>
.page
{
}
.page:after{
  content: '';/*必須*/
  position: absolute;
  top: 30px;/*四邊縮放距離*/
  right:380px;
  bottom: 30px;
  left: 380px;
  /*background-image: linear-gradient(180deg,
rgba(180,180,180,0.3) 40%,rgba(180,180,180,0));漸層語法 (方向,初始顏色,結
束顏色)*/
  opacity:0.6; /*透明度*/
  /*border: 5px solid #b2876f;*/
  background-color:#191919;
  pointer-events: none;
  z-index: -1;
  border-radius: 3%;
}
```

```
*,*::before,*::after {
  box-sizing:border-box
}
body {
  position:fixed;
  width:100%;
  height:100%;
  overflow:hidden
}
.tekst {
  color:#505050;
  margin:0;
  padding:0;
  font-family:'Lato',arial,sans-serif;
  text-transform:uppercase;
  font-size:14px;
  line-height:28px;
  font-weight:bold;
  display:block;
  letter-spacing:2px;
  max-width:100%;
  text-align:center;
  position:absolute;
  bottom:20px;
  left:30px;
  z-index:1
}
a {
  color:#fefefe;
  text-decoration:none;
  cursor:pointer
}
a:hover {
  text-decoration:underline
}
.pozycja {
  position:absolute;
  top:80%;
  left:15%;
  height:100%;
```

```

    opacity: 0.8;
    -webkit-transform: translate(-50%,-111px);
    -moz-transform: translate(-50%,-111px);
    -ms-transform: translate(-50%,-111px);
    -o-transform: translate(-50%,-111px);
    transform: translate(-50%,-111px)
}

.ludek {
    position: relative;
    width: 200px;
    height: 100%
}

.ludek .cialo, .skora {
    position: absolute;
    width: 100%;
    height: 105%;
    top: 0;
    left: 0;
    background: #191919;
    border-radius: 200px 200px 0 0;
    -webkit-transform: rotate(0deg);
    -moz-transform: rotate(0deg);
    -ms-transform: rotate(0deg);
    -o-transform: rotate(0deg);
    -webkit-transform-origin: bottom center;
    -moz-transform-origin: bottom center;
    -ms-transform-origin: bottom center;
    -o-transform-origin: bottom center;
    transform-origin: bottom center
}

.ludek .skora {
    overflow: hidden
}

.ludek .cialo .ucho {
    position: absolute;
    top: 80px;
    width: 45px;
    height: 45px;
    background: #191919;
    border-radius: 45px
}

```

```

}

.ludek .cialo .ucho.lewe {
  left:-10%
}

.ludek .cialo .ucho.prawe {
  right:-10%
}

.ludek .oko {
  width:50px;
  height:50px;
  position:absolute;
  top:85px;
  background:#fefefe;
  border-radius:50%;
  z-index:1
}

.ludek .oko.lewe {
  left:47px
}

.ludek .oko.prawe {
  left:103px
}

.ludek .oko .zrenica {
  width:15px;
  height:15px;
  position:absolute;
  top:18px;
  left:18px;
  background:#191919;
  border-radius:50%
}

.ludek .powieka_gorna {
  width:70%;
  height:30px;
  position:absolute;
  top:50px;
  left:15%;
  background:#191919;
  border-radius:30px 30px 0 0;
  z-index:3
}

```



```

}
.ludek .powieka_dolna {
    width:70%;
    height:30px;
    position:absolute;
    top:140px;
    left:15%;
    background:#191919;
    border-radius:0 0 30px 30px;
    z-index:3
}
.ludek .usta {
    width:80%;
    height:32px;
    position:absolute;
    top:145px;
    left:10%;
    background:#fefefe;
    border-radius:32px;
    z-index:4
}
</style>
<div class="pozycja">
    <div class="ludek">
        <div class="cialo">
            <div class="ucho lewe"></div>
            <div class="ucho prawe"></div>
            <div class="skora">
                <div class="oko lewe"><span class="zrenica"></span></div>
                <div class="oko prawe"><span class="zrenica"></span></div>
                <div class="powieka_gorna"></div>
                <div class="powieka_dolna"></div>
                <div class="usta"></div>
            </div>
        </div>
    </div>
</div>

<!-- <div class="page"> -->
<body style="background-size: cover;" background="img/ntust1.jpg" >

```

```

        <h1 class="frame-btn__text" style="position: absolute; top:
10px; text-align:left; font-size: 60px; color:white;
padding-left:30px;" >
        Product search
    </h1>
</br>

</div>

```

```

    <div style="right: 220px;top: -10px; position: absolute;">
        <a class="frame-btn" href="log.php">
            <span class="frame-btn__outline frame-btn__outline--tall">
                <span class="frame-btn__line frame-btn__line--tall"></span>
                <span class="frame-btn__line frame-btn__line--flat"></span>
            </span>
            <span class="frame-btn__outline frame-btn__outline--flat">
                <span class="frame-btn__line frame-btn__line--tall"></span>
                <span class="frame-btn__line frame-btn__line--flat"></span>
            </span>
            <span class="frame-btn__solid"></span>
            <span class="frame-btn__text" style="color:white;">Log
in</span>
        </a>
    </div>

```

```

    </br></br></br></br>
    <div style="right: 60px;top: -10px; position: absolute;">
        <a class="frame-btn" href="register.php">
            <span class="frame-btn__outline frame-btn__outline--tall">
                <span class="frame-btn__line frame-btn__line--tall"></span>
                <span class="frame-btn__line frame-btn__line--flat"></span>
            </span>
            <span class="frame-btn__outline frame-btn__outline--flat">
                <span class="frame-btn__line frame-btn__line--tall"></span>
                <span class="frame-btn__line frame-btn__line--flat"></span>
            </span>
            <span class="frame-btn__solid"></span>
            <span class="frame-btn__text" style="color:white;">Sign
up</span>

```

```

        </a>
    </div>
</br>
</div>
</div>
</body>
</div>
</html>
<script>
    $(function() {
        var tarcie = (1 / 6);
        var cialo_x = 0,
            cialoRuchX = 0,
            uszy_x = 0,
            uszy_y = 0,
            uszyRuchX = 0,
            uszyRuchY = 0,
            zrenica_x = 0,
            zrenica_y = 0,
            zrenicaRuchX = 0,
            zrenicaRuchY = 0,
            oko_x = 0,
            oko_y = 0,
            okoRuchX = 0,
            okoRuchY = 0,
            powieka_gorna_x = 0,
            powieka_gorna_y = 0,
            powiekaGornaRuchX = 0,
            powiekaGornaRuchY = 0,
            powieka_dolna_x = 0,
            powieka_dolna_y = 0,
            powiekaDolnaRuchX = 0,
            powiekaDolnaRuchY = 0,
            usta_x = 0,
            usta_y = 0,
            ustaRuchX = 0,
            ustaRuchY = 0;

        function animacja() {
            cialo_x += (cialoRuchX - cialo_x) * tarcie;

```

```

        $($('.cialo')).css({
            '-webkit-transform': 'rotate(' + -cialo_x + 'deg)',
            'transform': 'rotate(' + -cialo_x + 'deg)'
        });
        uszy_x += (uszyRuchX - uszy_x) * tarcie;
        uszy_y += (uszyRuchY - uszy_y) * tarcie;
        if (uszy_x < 0) {
            $($('.ucho.lewe')).css({
                '-webkit-transform': 'translate(' + -uszy_x * 7 +
                'px, ' + -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x * 7 + 'px, ' +
                -uszy_y + 'px)'
            });
            $($('.ucho.prawe')).css({
                '-webkit-transform': 'translate(' + -uszy_x + 'px, ' +
                -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x + 'px, ' +
                -uszy_y + 'px)'
            });
        } else {
            $($('.ucho.lewe')).css({
                '-webkit-transform': 'translate(' + -uszy_x + 'px, ' +
                -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x + 'px, ' +
                -uszy_y + 'px)'
            });
            $($('.ucho.prawe')).css({
                '-webkit-transform': 'translate(' + -uszy_x * 7 +
                'px, ' + -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x * 7 + 'px, ' +
                -uszy_y + 'px)'
            });
        }
        zrenica_x += (zrenicaRuchX - zrenica_x) * (tarcie * 3);
        zrenica_y += (zrenicaRuchY - zrenica_y) * (tarcie * 3);
        $($('.zrenica')).css({
            '-webkit-transform': 'translate(' + -zrenica_x + 'px, ' +
            -zrenica_y + 'px)',
            'transform': 'translate(' + -zrenica_x + 'px, ' +
            -zrenica_y + 'px)'
        });
    }
}

```

```

    });
    oko_x += (okoRuchX - oko_x) * tarcie;
    oko_y += (okoRuchY - oko_y) * tarcie;
    if (oko_x < 0) {
        $('.oko.lewe').css({
            'z-index': '2',
            '-webkit-transform': 'translate(' + -oko_x * 2 + 'px, ' + -oko_y + 'px)',
            'transform': 'translate(' + -oko_x * 2 + 'px, ' + -oko_y + 'px)'
        });
        $('.oko.prawe').css({
            'z-index': '1',
            '-webkit-transform': 'translate(' + -oko_x + 'px, ' + -oko_y + 'px)',
            'transform': 'translate(' + -oko_x + 'px, ' + -oko_y + 'px)'
        });
    } else {
        $('.oko.prawe').css({
            'z-index': '2',
            '-webkit-transform': 'translate(' + -oko_x * 2 + 'px, ' + -oko_y + 'px)',
            'transform': 'translate(' + -oko_x * 2 + 'px, ' + -oko_y + 'px)'
        });
        $('.oko.lewe').css({
            'z-index': '1',
            '-webkit-transform': 'translate(' + -oko_x + 'px, ' + -oko_y + 'px)',
            'transform': 'translate(' + -oko_x + 'px, ' + -oko_y + 'px)'
        });
    }
    powieka_gorna_x += (powiekaGornaRuchX - powieka_gorna_x) * tarcie;
    powieka_gorna_y += (powiekaGornaRuchY - powieka_gorna_y) * tarcie;
    $('.powieka_gorna').css({
        '-webkit-transform': 'translate(' + -powieka_gorna_x +

```

```

'px, ' + -powieka_gorna_y + 'px)',
    'transform': 'translate(' + -powieka_gorna_x + 'px, ' +
-powieka_gorna_y + 'px)'
    });
    powieka_dolna_x += (powiekaDolnaRuchX - powieka_dolna_x) *
tarcie;
    powieka_dolna_y += (powiekaDolnaRuchY - powieka_dolna_y) *
tarcie;
    $('.powieka_dolna').css({
        '-webkit-transform': 'translate(' + -powieka_dolna_x +
'px, ' + -powieka_dolna_y + 'px)',
        'transform': 'translate(' + -powieka_dolna_x + 'px, ' +
-powieka_dolna_y + 'px)'
    });
    usta_x += (ustaRuchX - usta_x) * tarcie;
    usta_y += (ustaRuchY - usta_y) * tarcie;
    $('.usta').css({
        '-webkit-transform': 'translate(' + -usta_x + 'px, ' +
-usta_y + 'px)',
        'transform': 'translate(' + -usta_x + 'px, ' + -usta_y +
'px)'
    });
    window.requestAnimationFrame(animacja);
}
$(window).on('mousemove click', function(e) {
    var lMouseX = Math.max(-100, Math.min(100, $(window).width()
/ 6 - e.clientX));
    var lMouseY = Math.max(-100, Math.min(100,
$(window).height() / 1.5 - e.clientY));
    cialoRuchX = (5 * lMouseX) / 200;
    uszyRuchX = (3 * -lMouseX) / 100;
    uszyRuchY = (10 * -lMouseY) / 100;
    zrenicaRuchX = (25 * lMouseX) / 200;
    zrenicaRuchY = (13 * lMouseY) / 100;
    okoRuchX = (20 * lMouseX) / 100;
    if (lMouseY <= 0) {
        okoRuchY = (10 * lMouseY) / 100;
        powiekaDolnaRuchY = (25 * lMouseY) / 100;
    } else {
        okoRuchY = (30 * lMouseY) / 100;

```

```
        powiekaDolnaRuchY = (55 * lMouseY) / 100;
    }
    powiekaGornaRuchX = (25 * lMouseX) / 100;
    powiekaGornaRuchY = (35 * lMouseY) / 100;
    powiekaDolnaRuchX = (25 * lMouseX) / 100;
    ustaRuchX = (35 * lMouseX) / 100;
    ustaRuchY = (10 * lMouseY) / 100;
    });
    animacja();
});
</script>
```

## 2. 會員註冊

```
<?php
session_start();

require_once 'db.php' ;//連結使用者資料庫
?>

<html>
  <head>
    <meta charset="utf-8">
    <title>會員登入</title>
    <link rel="stylesheet" href="css/flat-ui.css">

    <link rel="stylesheet" href="css/all.css">

    <script type="text/javascript"
src="js/flat-ui.min.js"></script>
  </head>

<!--消去連結底線 對齊用 其中"確認密碼"額外拉出來對-->
<style>
.size
{
  font-size: 30px;
  font-family: Microsoft JhengHei;
}
h1
{
  font-family: Microsoft JhengHei;
}
a
{
  text-decoration:none;
  font-family: Microsoft JhengHei;
}
.right
{
  margin-left: 80px;
}
```



```

.page
{

}

.page:after{
    content: '';/*必須*/
    position: absolute;
    top: 30px;/*四邊縮放距離*/
    right:420px;
    bottom: 40px;
    left: 420px;
    /*background-image: linear-gradient(180deg,
    rgba(180,180,180,0.3) 40%,rgba(180,180,180,0));漸層語法 (方向,初始顏色,結
    束顏色)*/
    opacity:0.4; /*透明度*/
    text-align: center;
    background-color:black;
    pointer-events: none;
    z-index: -1;
    border-radius: 3%;
}
</style>

<body style="background-size: cover; line-height: 0px;"
background="img/ntust1.jpg">
    <form action="dbSuccessful.php" method="POST"> <!-- 值的傳送
-->
        <h1 class="frame-btn__text" style="color:white;
padding-top: 20px; padding-bottom: 20px;">會員註冊</h1>
        <br>
        <div class="col page">
            <div class="form-group size" style="color:white;">
                帳號:<input type="text" name ="myaccount" autofocus
placeholder="輸入帳號..." class="form-control right" required />
            </div>
        </div>

        <div class="form-group size" style="color:white;">
            密碼:<input type="password" name="mypassword"
placeholder="輸入密碼..." class="form-control right"

```

```

onchange="mypassword2.pattern=(this.value);" required >
    </div>
    <div class="form-group size" style="color:white;">
        確認密碼:<input type="password" name="mypassword2"
placeholder="確認密碼..." pattern="" class="form-control" title="密碼不
相符" style="margin-left:20px" required >
    </div>
    <div class="form-group size"
style="color:white;">

        信箱:<input type="email" name="myEmail"
placeholder="輸入信箱..." class="form-control right" required >
    </div>

    <!--限制只能輸入數字-->
    <div class="form-group size" style="color:white;">
        手機:<input maxlength="10"
onkeyup="value=value.replace(/^[^d]/g, '')" type="text"
name="myphone" placeholder="輸入手機..." pattern='\d{10}' title="請輸入
十碼數字" class="form-control right" required >
    </div>
    </div>
    <p style="text-align: center" >
        <button type="submit" name="sub"
class="gradient-btn" style="color:white;font-family: Microsoft
JhengHei;">

        註冊

    </button>
    </p>
    <p class="frame-btn__text" style="font-size: 20px;
text-align: center;margin-top:20px;" ><a href="index.php">回到首頁
</a></p>
    </form>

</body>
</html>

```

### 3. 登入

```
<?php
session_start();

?>

<html>
  <head>
    <meta charset="utf-8">
    <title class="">Log in</title>
    <script src='https://www.google.com/recaptcha/api.js'></script>
    <script src="js/jquery-3.3.1.min.js"></script>
    <link rel="stylesheet" href="css/flat-ui.css">
    <link rel="stylesheet" href="css/all.css">
    <script type="text/javascript"
src="js/flat-ui.min.js"></script>
  </head>
  <body style="background-size: cover;"
background="img/ntust1.jpg">

<?php
//檢查是否已登入過 若有 則直接進入後台
if(isset($_SESSION['is_login']) && $_SESSION['is_login']==TRUE)
{
  header('Location: BackEnd.php');
}
?>
<style>
a,button,h1,p{
  font-family: Microsoft JhengHei;
}
.page
{
}
.page:after{
```

```
content: '';/*必須*/
position: absolute;
top: 30px;/*四邊縮放距離*/
right:480px;
bottom: 30px;
left: 480px;

/*background-image: linear-gradient(180deg,
rgba(180,180,180,0.3) 40%,rgba(180,180,180,0));漸層語法 (方向,初始顏色,結
束顏色)*/
opacity:0.4; /*透明度*/
text-align: center;
background-color:black;
pointer-events: none;
z-index: -1;
border-radius: 3%;
}
*,*::before,*::after {
box-sizing:border-box
}
body {
position:fixed;
width:100%;
height:100%;
overflow:hidden
}
.tekst {
color:#505050;
margin:0;
padding:0;
font-family: 'Lato', arial, sans-serif;
text-transform:uppercase;
font-size:14px;
line-height:28px;
font-weight:bold;
display:block;
letter-spacing:2px;
max-width:100%;
text-align:center;
position:absolute;
bottom:20px;
```

```
    left:30px;
    z-index:1
}
a {
    color:#fefefe;
    text-decoration:none;
    cursor:pointer
}
a:hover {
    text-decoration:underline
}
.pozycja {
    position:absolute;
    top:80%;
    left:85%;
    opacity: 0.8;
    height:80%;
    -webkit-transform:translate(-50%,-111px);
    -moz-transform:translate(-50%,-111px);
    -ms-transform:translate(-50%,-111px);
    -o-transform:translate(-50%,-111px);
    transform:translate(-50%,-111px)
}
.ludek {
    position:relative;
    width:200px;
    height:100%
}
.ludek .cialo,.skora {
    position:absolute;
    width:100%;
    height:105%;
    top:0;
    left:0;
    background:#191919;
    border-radius:200px 200px 0 0;
    -webkit-transform:rotate(0deg);
    -moz-transform:rotate(0deg);
    -ms-transform:rotate(0deg);
    -o-transform:rotate(0deg);
```

```
-webkit-transform-origin:bottom center;
-moz-transform-origin:bottom center;
-ms-transform-origin:bottom center;
-o-transform-origin:bottom center;
transform-origin:bottom center
}

.ludek .skora {
  overflow:hidden
}

.ludek .cialo .ucho {
  position:absolute;
  top:80px;
  width:45px;
  height:45px;
  background:#191919;
  border-radius:45px
}

.ludek .cialo .ucho.lewe {
  left:-10%
}

.ludek .cialo .ucho.prawe {
  right:-10%
}

.ludek .oko {
  width:50px;
  height:50px;
  position:absolute;
  top:85px;
  background:#fefefe;
  border-radius:50%;
  z-index:1
}

.ludek .oko.lewe {
  left:47px
}

.ludek .oko.prawe {
  left:103px
}

.ludek .oko .zrenica {
  width:15px;
```

```
height:15px;
position:absolute;
top:18px;
left:18px;
background:#191919;
border-radius:50%
}
.ludek .powieka_gorna {
width:70%;
height:30px;
position:absolute;
top:50px;
left:15%;
background:#191919;
border-radius:30px 30px 0 0;
z-index:3
}
.ludek .powieka_dolna {
width:70%;
height:30px;
position:absolute;
top:140px;
left:15%;
background:#191919;
border-radius:0 0 30px 30px;
z-index:3
}
.ludek .usta {
width:80%;
height:32px;
position:absolute;
top:145px;
left:10%;
background:#fefefe;
border-radius:32px;
z-index:4
}
</style>
</br> </br></br>
<!--傳送資料-->
```

```

<div class="pozycja">
  <div class="ludek">
    <div class="cialo">
      <div class="ucho lewe"></div>
      <div class="ucho prawe"></div>
      <div class="skora">
        <div class="oko lewe"><span class="zrenica"></span></div>
        <div class="oko prawe"><span class="zrenica"></span></div>
        <div class="powieka_gorna"></div>
        <div class="powieka_dolna"></div>
        <div class="usta"></div>
      </div>
    </div>
  </div>
</div>

```

```

<div class="page">
  </div>
  <form action="checkUser.php" method="POST">
    <h1 class="frame-btn__text" style="font-size:
30px;text-align:center; color:white;">登入</h1>
    <br>
    <div class="form-group">
      <p style="font-size: 20px;text-align: center; color:white;">
        <b>帳號 <input type="text" class="form-control" name="ACT"
autofocus required ></b>
      </p>
      <p style="font-size: 20px;text-align: center; color:white;">
        <b>密碼 <input type="password" class="form-control"
name="PAS" required >
      </p></b>
    </div>

```

```

    <!--我不是機器人<center> <div class="g-recaptcha"
data-sitekey="6LeD1mUUAAAAACVoiN475RJy-QBcAQbe_kRVBqNE
"></div></center>-->

```



```
        <p style="text-align: center;margin-top: 20px">
            <button type ="submit" class="gradient-btn"
style="color:white;">
                登入
            </button>
        </p>
        </br> </br>
        <p style="font-size: 20px; text-align: center"><a
href="index.php">回到首頁</a></p>
    </form>
</div>
</body>
```

```
</html>
```

```
<script>
```

```
    $(function() {
        var tarcie = (1 / 6);
        var cialo_x = 0,
            cialoRuchX = 0,
            uszy_x = 0,
            uszy_y = 0,
            uszyRuchX = 0,
            uszyRuchY = 0,
            zrenica_x = 0,
            zrenica_y = 0,
            zrenicaRuchX = 0,
            zrenicaRuchY = 0,
            oko_x = 0,
            oko_y = 0,
            okoRuchX = 0,
            okoRuchY = 0,
            powieka_gorna_x = 0,
            powieka_gorna_y = 0,
            powiekaGornaRuchX = 0,
            powiekaGornaRuchY = 0,
            powieka_dolna_x = 0,
            powieka_dolna_y = 0,
            powiekaDolnaRuchX = 0,
            powiekaDolnaRuchY = 0,
            usta_x = 0,
```

```

        usta_y = 0,
        ustaRuchX = 0,
        ustaRuchY = 0;

    function animacja() {
        cialo_x += (cialoRuchX - cialo_x) * tarcie;
        $('.cialo').css({
            '-webkit-transform': 'rotate(' + -cialo_x + 'deg)',
            'transform': 'rotate(' + -cialo_x + 'deg)'
        });
        uszy_x += (uszyRuchX - uszy_x) * tarcie;
        uszy_y += (uszyRuchY - uszy_y) * tarcie;
        if (uszy_x < 0) {
            $('.ucho.lewe').css({
                '-webkit-transform': 'translate(' + -uszy_x * 7 +
                'px, ' + -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x * 7 + 'px, ' +
                -uszy_y + 'px)'
            });
            $('.ucho.prawe').css({
                '-webkit-transform': 'translate(' + -uszy_x + 'px, '
                + -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x + 'px, ' +
                -uszy_y + 'px)'
            });
        } else {
            $('.ucho.lewe').css({
                '-webkit-transform': 'translate(' + -uszy_x + 'px, '
                + -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x + 'px, ' +
                -uszy_y + 'px)'
            });
            $('.ucho.prawe').css({
                '-webkit-transform': 'translate(' + -uszy_x * 7 +
                'px, ' + -uszy_y + 'px)',
                'transform': 'translate(' + -uszy_x * 7 + 'px, ' +
                -uszy_y + 'px)'
            });
        }
        zrenica_x += (zrenicaRuchX - zrenica_x) * (tarcie * 3);
    }

```

```

        zrenica_y += (zrenicaRuchY - zrenica_y) * (tarcie * 3);
        $($('.zrenica')).css({
            '-webkit-transform': 'translate(' + -zrenica_x + 'px, ' +
-zrenica_y + 'px)',
            'transform': 'translate(' + -zrenica_x + 'px, ' +
-zrenica_y + 'px)'
        });
        oko_x += (okoRuchX - oko_x) * tarcie;
        oko_y += (okoRuchY - oko_y) * tarcie;
        if (oko_x < 0) {
            $($('.oko.lewe')).css({
                'z-index': '2',
                '-webkit-transform': 'translate(' + -oko_x * 2 + 'px,
' + -oko_y + 'px)',
                'transform': 'translate(' + -oko_x * 2 + 'px, ' +
-oko_y + 'px)'
            });
            $($('.oko.prawe')).css({
                'z-index': '1',
                '-webkit-transform': 'translate(' + -oko_x + 'px, ' +
-oko_y + 'px)',
                'transform': 'translate(' + -oko_x + 'px, ' + -oko_y
+ 'px)'
            });
        } else {
            $($('.oko.prawe')).css({
                'z-index': '2',
                '-webkit-transform': 'translate(' + -oko_x * 2 + 'px,
' + -oko_y + 'px)',
                'transform': 'translate(' + -oko_x * 2 + 'px, ' +
-oko_y + 'px)'
            });
            $($('.oko.lewe')).css({
                'z-index': '1',
                '-webkit-transform': 'translate(' + -oko_x + 'px, ' +
-oko_y + 'px)',
                'transform': 'translate(' + -oko_x + 'px, ' + -oko_y
+ 'px)'
            });
        }
    }
}

```

```

        powieka_gorna_x += (powiekaGornaRuchX - powieka_gorna_x) *
tarcie;
        powieka_gorna_y += (powiekaGornaRuchY - powieka_gorna_y) *
tarcie;
        $('.powieka_gorna').css({
            '-webkit-transform': 'translate(' + -powieka_gorna_x +
'px, ' + -powieka_gorna_y + 'px)',
            'transform': 'translate(' + -powieka_gorna_x + 'px, ' +
-powieka_gorna_y + 'px)'
        });
        powieka_dolna_x += (powiekaDolnaRuchX - powieka_dolna_x) *
tarcie;
        powieka_dolna_y += (powiekaDolnaRuchY - powieka_dolna_y) *
tarcie;
        $('.powieka_dolna').css({
            '-webkit-transform': 'translate(' + -powieka_dolna_x +
'px, ' + -powieka_dolna_y + 'px)',
            'transform': 'translate(' + -powieka_dolna_x + 'px, ' +
-powieka_dolna_y + 'px)'
        });
        usta_x += (ustaRuchX - usta_x) * tarcie;
        usta_y += (ustaRuchY - usta_y) * tarcie;
        $('.usta').css({
            '-webkit-transform': 'translate(' + -usta_x + 'px, ' +
-usta_y + 'px)',
            'transform': 'translate(' + -usta_x + 'px, ' + -usta_y +
'px)'
        });
        window.requestAnimationFrame(animacja);
    }
    $(window).on('mousemove click', function(e) {
        var lMouseX = Math.max(-100, Math.min(100, $(window).width()
/1.1 - e.clientX));
        var lMouseY = Math.max(-100, Math.min(100,
$(window).height() / 1.5) - e.clientY));
        cialoRuchX = (5 * lMouseX) / 200;
        uszyRuchX = (3 * -lMouseX) / 100;
        uszyRuchY = (10 * -lMouseY) / 100;
        zrenicaRuchX = (25 * lMouseX) / 200;
        zrenicaRuchY = (13 * lMouseY) / 100;
    });

```

```
        okoRuchX = (20 * lMouseX) / 100;
        if (lMouseY <= 0) {
            okoRuchY = (10 * lMouseY) / 100;
            powiekaDolnaRuchY = (25 * lMouseY) / 100;
        } else {
            okoRuchY = (30 * lMouseY) / 100;
            powiekaDolnaRuchY = (55 * lMouseY) / 100;
        }
        powiekaGornaRuchX = (25 * lMouseX) / 100;
        powiekaGornaRuchY = (35 * lMouseY) / 100;
        powiekaDolnaRuchX = (25 * lMouseX) / 100;
        ustaRuchX = (35 * lMouseX) / 100;
        ustaRuchY = (10 * lMouseY) / 100;
    });
    animacja();
});
</script>
```

## 4. 搜尋

```
<?php
    session_start();
    //紀錄使用者帳號
    $idd=$_GET["id"];
    //檢查session狀態是否已登入
    if(isset($_SESSION['is_login']) && $_SESSION['is_login']==TRUE):

?>

<head>
    <meta charset="utf-8">
    <title>商品搜尋</title>
    <link rel="stylesheet" href="css/flat-ui.css">
    <link rel="stylesheet" href="css/all.css">
    <script type="text/javascript" src="js/flat-ui.min.js"></script>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.min.js"
>
    </script>
</head>

<style>/*隱藏所有超連結底線*/
a {
    text-decoration:none;

}
select {
    padding: 5px;
    font-size: 16px;
    border: 1px solid #a3a3a3;
    height: 34px;
    border-radius:8px;
    -webkit-border-radius:8px;
    -moz-border-radius :8px;
    box-shadow:inset -1px -1px 1px 1px rgba(10%,10%,10%,0.1);
}
.small
{
```

```
font-size: 30px;
}
.page
{

}
.page:after{
content: ' ';/*必須*/
position: absolute;
top: 30px;/*四邊縮放距離*/
right:420px;
bottom: 40px;
left: 420px;
/*background-image: linear-gradient(180deg,
rgba(180,180,180,0.3) 40%,rgba(180,180,180,0));漸層語法 (方向,初始顏色,結
束顏色)*/
opacity:0.4; /*透明度*/
text-align: center;
background-color:black;
pointer-events: none;
z-index: -1;
border-radius: 3%;
}
.temp1
{
}
.temp2
{
}
.temp3
{
}
.temp4
{
}
.search1
{

}
</style>
```

```
<body style="background-size: cover;" background="img/ntust1.jpg">
  <h1 class="frame-btn__text" style="color:white; padding-top:
20px;">Let's search!</h1>
```

```
  <h2 style="text-align: center;"></h2>
```

```
</br></br></br>
```

```
<form action="backEndSearch.php" method="POST">
```

```
  <!--把使用者名稱傳到backEndSearch.php 用php括起來
一定要加echo!!!!!!!!!!!!-->
```

```
    <input type="hidden" name="myid"
value="<?php echo $idd ?>">
    <input type="hidden" name="myEmail" value="<?php echo $idd
?>">
```

```
  <div class="page">
```

```
    <p class="frame-btn__text"
style="font-size: 20px;margin-right: 85px; color:white;">
      商品類型 <select name="type"
class="frame-btn__text" style="height: 40px;">
        <option value="computer">平板</option>
        <option value="phone">手機</option>
        <option value="laptop">筆記型電腦</option>
      </select>
    </p>
```

```
<div class="form-group">
```

```
  <p class="frame-btn__text"
style="font-size: 20px;text-align: center; color:white;"
data-effects="default">
    商品名稱 <input type="text" class="form-control"
name="productName" autofocus required list="team_list">
    <datalist id="team_list" class="frame-btn__text">
      <option class="temp1">Apple</option>
      <option class="temp2">華碩</option>
      <option class="temp3">聯想</option>
```



```

        <option class="temp4">Acer</option>
    </datalist>
</p>

    <p class="frame-btn__text" style="font-size:
20px;text-align: center ;color:white;">
        商品價格 <input type="" class="form-control"
name="productPrice" onkeyup="value=value.replace(/[\^\d]/g, '') "
required >
    </p>
</div>

</div>

    <button type ="submit" name="submit" value="add"
class="gradient-btn" style="color:white;">
        搜尋
    </button>

    <button style="margin-left: 100px; color:white;" type
="submit" name="submit" value="search" class="search1 gradient-btn">
        歷史紀錄
    </button>

</form>
<br>
<p style="text-align: center ; font-size: 20px;"><a
class="frame-btn__text" style="color:white;" href="LogOut.php">登出
</a></p>
</body>
<?php
else:
    header('Location: login.php');//若無則轉回登入畫面
endif;
?>
<script>
//若action為搜尋 則不受input required之限制
$(".search1").click(function(event) {
    $(".form-control").removeAttr('required');
});

$(".select").change(function(event) {

```

```
if ($("#select").val()==("computer")) {  
    $(".temp1").html("平板1");  
    $(".temp2").html("平板2");  
    $(".temp3").html("平板3");  
    $(".temp4").html("平板4");  
}  
  
else if ($("#select").val()==("phone")) {  
    $(".temp1").html("手機1");  
    $(".temp2").html("手機2");  
    $(".temp3").html("手機3");  
    $(".temp4").html("手機4");  
}  
  
else if ($("#select").val()==("laptop")) {  
    $(".temp1").html("筆電1");  
    $(".temp2").html("筆電2");  
    $(".temp3").html("筆電3");  
    $(".temp4").html("筆電4");  
}  
});  
  
</script>
```

## 5. 搜尋結果

```
<head>
  <link rel="stylesheet" href="css/all.css">
</head>

<style>
#customers
{
  margin-top: 30px;
  font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
  width:70%;
  border-collapse:collapse;
}
#customers td, #customers th
{
  font-size:2em;
  border:1px;
  padding:3px 7px 2px 7px;
  text-align: center;
}
#customers th
{
  font-size:1.9em;
  text-align:center;
  padding-top:5px;
  padding-bottom:4px;
  background-color:#b2876f;
  color:#white;
}
#customers tr.alt td
{
  color:#white;
  background-color:#EAF2D3;
}
.logout
{
  display:inline-block;
}
</style>
```

```

<?php
require_once 'db.php' ;
$myid=$_POST["myid"];
$pdName=$_POST["productName"];
$pdPrice=$_POST["productPrice"];
date_default_timezone_set('Asia/Taipei');//抓取目前時間
$datetime= date("Y/m/d H:i:s");
$futureDate=date("Y-m-d",strtotime("+14day"));//到期時間
$flagNum=0;
//判斷搜尋商品型態
$type=$_POST["type"];
$finaltype;
if($type=="computer"){
$finaltype=1;
}
else if($type=="phone"){
$finaltype=2;
}
else{
$finaltype=3;
}

// if(isset($myid))

    $sql = "SELECT * FROM `userinformation` WHERE `account` =
'$myid'";
    $result = mysqli_query($link,$sql);
    $row = @mysqli_fetch_row($result);
//若按下"搜尋"才INSERT新需求進資料庫
if($_POST["submit"] == "add")
{
    能有空格 會爆炸
    $sql="INSERT INTO `searchtable`
(`userName`,`email`,`product`,`price`,`type`,`flag`,`dueTime`) VALUE
('{$myid}','{$row[4]}','{$pdName}','{$pdPrice}','{$finaltype}','{$fla
gNum}','{$futureDate}')";

```

```

        mysqli_query($link,$sql);
    }

$sql = "SELECT * FROM `searchtable` "; //在searchtable資料表中選擇所有欄位

// $row = mysqli_fetch_row($result); //將陣列以數字排列索引
// $row = mysqli_fetch_assoc($result); //將陣列以欄位名索引
$result = mysqli_query($link,$sql); // 執行SQL查詢

$total_fields=mysqli_num_fields($result); // 取得欄位數6

$total_records=mysqli_num_rows($result); // 取得多少筆資料
?>

<body style="background-size: cover;" background="img/ntust1.jpg" >
<center>
<table id="customers">
<tr >
    <th >商品名稱</th>
    <th >商品價錢</th>
    <th >搜尋日期</th>
</tr>
<tr>
<td><?php // 只列出this.搜尋紀錄
    for($i=0;$i<$total_records;$i++){
        $row = mysqli_fetch_assoc($result);
        if($row["userName"]==$myid)
        {
            echo $row["product"];
            echo "<br>";
        }
    }
    }?></td>

    <?php mysqli_free_result($result); //釋放內容 重新連接?>

<td><?php
$result = mysqli_query($link,$sql);

```

```

        for($i=0;$i<$total_records;$i++){
            $row = mysqli_fetch_assoc($result);
            if($row["userName"]==$myid)
            {
                echo $row["price"];
                echo "<br>";
            }
        }
    }  mysqli_free_result($result);  ?></td>

<td><?php
    $result = mysqli_query($link,$sql);
    for($i=0;$i<$total_records;$i++){
        $row = mysqli_fetch_assoc($result);
        if($row["userName"]==$myid)
        {
            echo $row["time"];
            echo "<br>";
        }
    }  mysqli_free_result($result);  ?></td>
</tr>
</center>
</body>

<form action="BackEnd.php" method="GET">
<input type="hidden" name="id" value="<?php echo $myid ?>">
<button type="submit" name="sub" class="gradient-btn"
style="color:white; margin-right: 250px;
">繼續搜</button>
<a class="gradient-btn logOut" style="color:white;"
href="LogOut.php">登出</a>
<h3 style="margin-top: 30px;text-align: center; color:white;">以下為您
的搜尋紀錄 若有符合條件之商品 將立即通知您!! </h3>
</form>

```

## 6. 檢查會員資料是否重複

```
<head> <link rel="stylesheet" href="css/all.css"></head>

<link rel="stylesheet" href="css/flat-ui.css">
    <link rel="stylesheet" href="css/all.css">
    <script type="text/javascript" src="js/flat-ui.min.js"></script>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.min.js"
>
    </script>

<?php
require_once 'db.php' ;
$id = $_POST['myaccount'];
$pw = $_POST['mypassword'];
$mail=$_POST['myEmail'];
$phone=$_POST['myphone'];
$sql = "SELECT * FROM `userinformation` WHERE `account` = '$id'";
$result = mysqli_query($link,$sql);
$row = @mysqli_fetch_row($result);

$email_sql = "SELECT * FROM `userinformation` WHERE `Email` =
'$mail'";
$email_result = mysqli_query($link,$email_sql);
$email_row = @mysqli_fetch_row($email_result);

if ( $row[1] == $id)
    {
        echo "<script>alert('帳號重複囉!! 請重新輸入~~
'); location.href = 'register.php';</script>";
    }

else if ( $email_row[4] == $mail)
    {
        echo "<script>alert('此信箱已註冊!! 請重新輸入~~
'); location.href = 'register.php';</script>";
    }

else
```

```
{  
$sql="INSERT INTO `userinformation`  
(`account`,`password`,`Phone`,`Email`) VALUE  
('{$id}','{$pw}','{$phone}','{$mail}')";  
mysqli_query($link,$sql);  
}  
?>
```

```
<body style="background-size: cover;" background="img/ntust1.jpg">
```

```
</br></br></br></br></br></br>
```

```
<h1 class="frame-btn__text" style="color:white;" >註冊成功!! 請  
重新登入~~</h1>
```

```
</br></br>
```

```
<p class="gradient-btn" font-size: 30px"><a style="text-decoration:  
none; color:white;" href="index.php">回到首頁</a></p>
```

```
</body>
```