

112-1 Introduction to Database Systems -

Final team project: LineBot 新竹輕旅行精靈

Team 19: 111550060 劉千慈、111550074 陳映竹、111550114 趙培安

1. An introduction of your application, including why you want to develop the application and the main functions of your application.

a. 動機：

有感於在新竹市讀書一年半，學業中忙裡偷閒的時刻仍然會因為不知道可以去哪裡玩、吃什麼所以決定在宿舍的床上度過一整天，又或者第一千零一次抵達巨城。

另外，新竹市政府的LINE Bot沒有提供關於旅遊景點的資訊，僅僅只是聚焦在針對新竹市民的便民服務，市面上目前也沒有類似的檢索工具。對於遊客來說，獲取旅遊資訊不太容易。

我們希望建立一個渠道讓新竹市的訪客可以快速得知所在的行政區的旅遊資訊，讓大家不再因為繁雜的資訊搜索過程和冗長的旅遊計劃時間耽誤難得的假期。

b. 應用程式簡介：

為了方便大家使用，我們使用LINE Bot功能結合資料庫，使用者選擇所在區域(東區、北區或香山區)以及欲查詢項目(景點、美食或住宿)之後將列出位於該地可用的選項，提供使用者方便的查詢新竹市的旅遊資訊。

LINE bot的功能包含三種功能的查詢：景點、美食、住宿，三種功能皆可以使用不同的分類進行搜尋，三者的項目包含簡介、推薦美食等細節資料，將會使用進階查詢的方式呈現：

- 景點：列出景點名稱、地址，點選景點名稱可以展開景點介紹內容。
- 美食：列出商店名稱、連絡電話、地址，點選推薦菜色選項展開推薦內容。
- 住宿：列出名稱、聯絡方式、地址，點選查看更多展開房間總數、最低價位、最高價位、是否設有對外營業之餐飲。

c. LineBot 新竹輕旅行精靈：[LineBot](#)

d. Presentation and demo video：[影片](#)

2. Database design - describe the schema of all your tables in the database, including keys and index, if applicable (why you need the keys or why you think adding an index is or is not helpful).

a. 以下是在資料庫中建立三個table所使用的SQL：

```
CREATE TABLE Food (  
    Name VARCHAR(30) PRIMARY KEY,  
    Type INT NOT NULL,  
    Info VARCHAR(30),  
    Areacode INT NOT NULL,  
    Address VARCHAR(30),  
    Recommand VARCHAR(50),  
    Introduce VARCHAR(10000)  
);
```

```
CREATE TABLE Living (  
    Name VARCHAR(30) PRIMARY KEY,  
    Type INT NOT NULL,  
    Info VARCHAR(30),  
    Areacode INT NOT NULL,  
    Address VARCHAR(30),  
    Room INT,  
    Lowest_p INT,  
    Highest_p INT,  
    Restaurant INT  
);
```

```
CREATE TABLE Sightseeing (  
    Name VARCHAR(30) PRIMARY KEY,  
    Type INT NOT NULL,  
    Areacode INT NOT NULL,  
    Address VARCHAR(30),  
    Introduce VARCHAR(10000)  
);
```

b. 具體描述：

我們將三個搜尋項目分開建立表格，是考量搜尋時的成本，以及SQL指令的簡潔程度。使用者在使用LINE Bot時，會先選擇想要查詢的項目、選擇類別，最後是地點，所以依據要查詢的項目建立Table，依據使用者的需求在不同的Table中查找資料。

在使用SQL搜尋時使用的條件有：Type、Areacode，因此三個Table中的這兩個attribute都必須是not null，依據這些條件搜索出來的資料會列出名稱、電話、地址等等資訊。

沒有使用index的原因是因為每一個表格的內容不多，使用index反而增加成本，我們希望將儲存空間使用在提供更完整、詳細的內容，帶給使用者更好的體驗。

Key的部分只有使用PK來確保每一個項目是獨立的，因為使用者不會同時查詢不同的項目，且這三個項目之間在資料內容上沒有關連性所以沒有使用到FK。

3. Database design - describe the normal form of all your tables. If the tables are not in BCNF, please include the reason (performance trade-off, etc.).

a. Normal Form:

特色美食、景點、住宿，三個都是BCNF。其中推薦菜色雖然列出許多項不同的商品，但是因為獨立性不高，並且在使用者查看的時候我們希望可以一口氣呈現所有的推薦菜色，故將整個欄位視為字串的資料類型輸出而不分成許多個不同的欄位處理，因此沒有將其分開儲存。

4. From the data sources to the database - describe the data source and the original format.

a. 資料來源：

新竹市政府資料開放平台，[新竹市觀光景點清單](#)、[新竹市合法旅宿資料名冊](#)、[新竹市特色美食](#)。將資料開放平台上的csv檔下載之後，進一步的整理資料，刪去不必要的內容並且統一資料格式，以下將列出原始資料的內容以及我們所做的修改，並附上處理過後的資料以及從官網上下載的原始檔案：[原始資料及修改後資料](#)。

b. 資料描述：

- 景點：

- 原始資料：景點名稱、areacode、地址、景點介紹。

- 修改：

- 將地址格式統一為新竹市XX區，方便使用者進行導航、或確定店家的具體位置。

- 新增景點分類，將所有景點分為六個不同類別：歷史建築、自然景觀、宗教文化、博物館、公園與休閒區、市區景點。(Note：

LINE Bot功能限制最多只能有六個分類, 我們希望在允許的情況下進行最仔細的分類)

- 修改後資料: 名稱、分類、Areacode、地址、介紹。
- 美食:
 - 原始資料: 商店名稱(名稱)、網址、電話、手機、區域(行政區)、區域代碼(areacode)、地址、介紹。
 - 修改:
 - 刪除網址資料。許多店家沒有自己的網頁, 有網頁的店家有些已經許久沒有更新, 或者是網站提供的資訊貧乏, 參考價值不高。另外考量到當多仰賴google地圖提供的資訊以及網友評價, 所以將此項移除。
 - 將地址格式統一為新竹市XX區, 方便使用者進行導航、或確定店家的具體位置。
 - 將介紹中的推薦菜色和餐廳簡介分開, 使內容較為簡略、方便使用者獲得較精準的資訊。
 - 將行政區刪除, 併入店家地址。
 - 將電話和手機合併為聯絡方式項目。
 - 新增美食分類: 伴手禮、小吃、飲料與甜點、餐廳。
 - 修改後資料: 名稱、分類、聯絡方式、Areacode、地址、推薦菜色、簡介。
- 住宿:
 - 原始資料: 編號(序號)、統一登記證編碼、旅館名稱、電話、行動電話、郵遞區號(postcode)、縣市(countrycode)、區域代碼(areacode)、營業地址、緯度、經度、房間總數、最低價格、最高價格、設有對外營業之餐飲業、星級評鑑。
 - 修改:
 - 新增分類: 飯店、旅館、民宿。
 - 刪除序號、統一登記證編碼、郵遞區號、經度、緯度, 與旅遊資訊無關故不保留、
 - 將電話和行動電話綜合成聯絡方式欄位。
 - 將地址統一格式為新竹市XX區。
 - 刪除星級評鑑, 因為參與評鑑的旅館不多導致資料量太少,
 - 刪除縣市, 將內容併入地址。
 - 將設有對外營業之餐飲的資料改為以0和1表示。
 - 修改後資料: 名稱、分類、聯絡方式、Areacode、地址、房間總數、最低價位、最高價位、設有對外營業之餐飲。

5. From the data sources to the database - describe the methods of importing the original data to your database and strategies for updating the data, if you have one.

a. 資料匯入：

我們在 mySQL 上建立一個 Database 與 RDS Server 的連線，創建表格之後，我們原先要使用 import wizard 功能將 csv 檔中的資料匯入，但因為過程中會出現有中文變成亂碼的情況，查找資料後發現需要將字元轉換成合適的編碼，不過我們手邊並無可以使用的工具，因此我們最後是使用 INSERT 的方式，將各筆資料匯入至表格中。

b. 資料更新：

因為我們的資料來源並不會做實時的變化，而且我們也不希望使用者在對話過程中隨意更新資料內容，因此並沒有做資料更新的功能。

6. Application with database - explain why your application needs a database.

a. 使用資料庫原因：

在旅遊資訊LINE Bot中使用資料庫可以有效的減少使用者搜尋資料的成本。

我們觀察到網路上關於旅遊的資訊龐雜、格式不一。資料來源除了官方網站以外，許多旅客還會猜考私人部落格的遊記、景點推薦等資料，資料內容雖然十分詳盡，但是想要快速得知旅遊資訊會比較困難。特別對於初次造訪、需要有效率的規劃旅遊計劃的人來說，一個一個搜尋旅遊景點將耗費大量時間。

因此我們希望利用政府提供的資訊建立一個統整旅遊資料的平台，內容主要提供使用者快速、簡略的了解有哪些備選的景點、美食和住宿地點。比起使用搜尋引擎閱覽無數的推薦文章之後仍一無所獲，使用者可以透過我們建立的LINE Bot依據需求在資料庫中篩選出較有興趣的內容，再到網路上深入了解細節，旅遊資訊Bot就達到快速檢索資訊的作用。

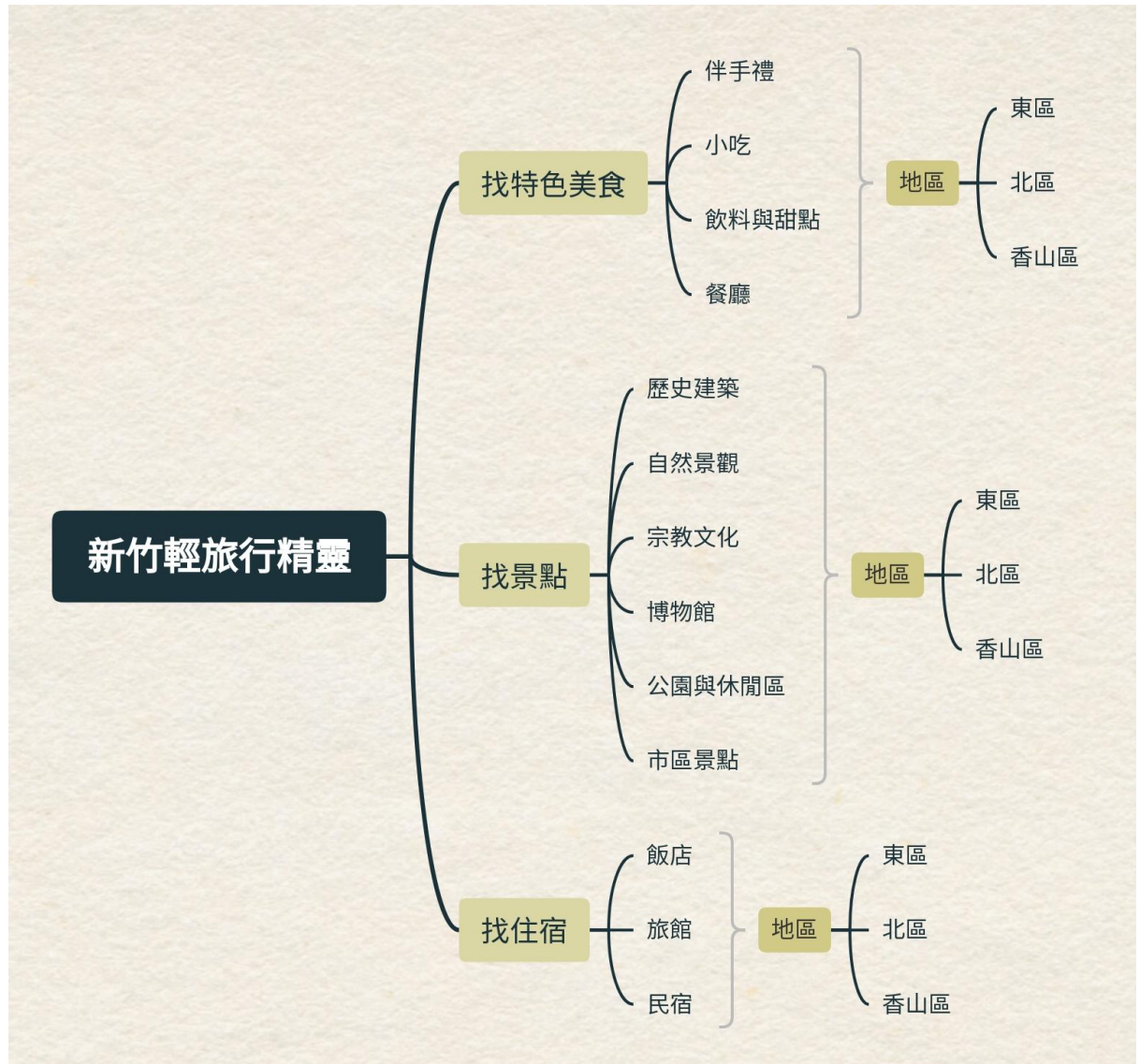
7. Application with database - includes the queries that are performed by your application, how your application performed these queries (connections between application and database), and what the cooperating functions for your application.

a. 應用程式執行的查詢：

我們的應用程式採用點選按鈕或圖片的方式提供使用者查詢不同的旅遊資料，搜尋結構如下方心智圖所示。

首先選擇想要查詢的項目(找美食/找景點/找住宿), 選擇不同的類別(因查詢的項目而異), 最後再選擇地區(10018010東區/10018020北區/10018030香山區)。點選完成後LINE Bot將列出所有符合的資料供使用者閱覽。

b. 應用程式查詢架構:



c. 呈現方式:

在LINE Bot呈現方面, 使用者首先可以在下方的選單中找到三張圖片, 點按圖片可以選擇想要搜尋的項目, 點選之後會回傳關鍵字, 並由此觸發選擇類別的選項。當回傳項目的關鍵字時, 例如: @找美食, LINE Bot會發送一張圖片, 該圖片分成幾個區塊, 代表不同的類別, 只要點選相應的圖片就會回傳關鍵字, 觸發選擇地區的選項。當回傳類別的關鍵字時, 例如: @小吃, LINE Bot將會回傳一個快速選單, 內容為「請選擇地區」。

在從資料庫中提取資料時, 我們使用Areacode和Type為條件找出需要的資料, 在python檔案中使用SQL提取資料後將資料變成dataframe的形式, 方便以轉盤的方式

在LINE Bot上呈現:其中在查詢時三個項目均有設定查看更多細節的功能,因此會將這些資料和基本的資料分開搜尋。以查詢美食為例,使用者點選完畢後會跳出符合條件的美食店家,其中在頁面上會直接呈現店家的名稱、聯絡方式和地址這三項基本資料,另外在這些資訊下面會出現兩個按鈕(推薦菜色、簡介),供使用者進一步了解。為了方便輸出資料,我們會將店家的基本資料、推薦菜色、簡介各用一個SQL搜索出來並建立成三個不同的dataframe,而非直接使用一個SQL、建立成一個dataframe。

另外,因為我們是根據不同項目、類別、地區做搜尋,因此可能會發生該地區沒有此類別的資料的情況(例如:香山區沒有飲料與甜點類的美食店家),此時會輸出「這個區域沒有您想要查詢的資料」來提醒使用者。而搜尋到的資料則會以轉盤的方式呈現。

- d. 以下是我們的完整程式碼:[程式碼](#)

8. All the other details of your application that you want us to know.

- a. 所有 LINE Bot 使用的圖片都是親自製作
- b. 使用 AWS Lambda 連接資料庫與 LineBot
- c. 雖然這是一個資料內容、schema結構相對簡單的作品,但是我們認為設計出一個好的應用程式最重要的是使用者是否能快速的獲取想要的資訊以及操作的簡便程度,因此在設計的時候我們以使用者的角度出發、考量搜尋成本和呈現的內容而產生了這個設計。透過這次的專題讓我們發現良好的設計必須與使用者站在一起,規劃的成品才會符合需求。
- 參考資料:
 - 書名:Python與LINE Bot機器人全面實戰特訓班 Flask最強應用(Creating LINE Bot with Python)
https://nycu.primo.exlibrisgroup.com/permalink/886UST_NYCU/1uq600n/alm_a99100359947740677
 - [\[Python+LINE Bot+PostgreSQL教學\]一篇搞懂LINE Bot讀取資料庫的方法\(learncodewithmike.com\)](#)
 - [AWS Lambda x Python LINE BOT微服務 I | by YM Chiu | Medium](#)
 - [使用 AWS Lambda 部署 LINE 聊天機器人\(上\) – eCloudture](#)