

Group J
Around Home

GitHub : https://github.com/moneychien19/db1101_midterm.git

Youtube : <https://youtu.be/CMWWeWo0-nA>

周敦翔 B07902050
錢紫翎 B07303129
李佳欣 B07303027
莊翔安 B08303028

一、背景與動機

有鑒於市面上租屋網站已收集房屋附近的交通、娛樂、教育（如鄰近中小學）等資訊，卻始終缺乏**環境指標、治安、環保**等與生活息息相關的指標和數據，對於在乎這類**生活品質與環保價值**的人而言，需要自行查詢這些資訊，著實是一個麻煩。

我們希望能創建一個平臺，讓使用者不必自行逐一查詢上述的資訊，只要輸入想查詢房屋的地址，就能從我們的資料庫中獲得**所有**資訊。我們以文字及地圖的形式展示，讓使用者能一目了然該地址的週遭情況。

二、網站介紹

我們的網站使用前後端分離的技術，分別啟動後端伺服器 and 前端應用程式後，即可在 **localhost:3000** 看到一個由 **React** 框架開發的前端網頁。

打開網頁後，會先看到**搜尋欄**及一張**地圖**，地圖預設顯示的是台大周圍的環境，使用者可以在網頁最上方的查詢區域輸入三個資訊：

1. 住家的地址或住家附近的地標。
2. 欲了解的資訊類別（有環境指標、環保、治安三個選項）。
3. 欲篩選的時間和距離範圍。

按下查詢後，查詢欄下方就會顯示使用者選擇的查詢類別及對應的資料。以下針對三個類別中的內容詳述。

1. 環境指標

此類別中包含空氣品質指標 **AQI**、紫外線指數及水庫水質卡爾森指數，這三個指標的數值均是取離住家最近的觀測站最近一次觀測到的值，指標欄位左側的燈號會隨觀測到的數值顯示綠燈、黃燈或紅燈，綠燈表示狀況最佳。

2. 環保

此類別中包含綠色友善餐廳、綠色商店、自備餐具優惠店家、垃圾車停靠點、舊衣回收箱，及公民營廢棄物清除機構之數量，前三項可以指引使用者進行更環保的消費，後三項則與日常生活機能相關。

我們預設一般民眾只想知道距離最近的垃圾車停靠點及舊衣回收箱，因此這二項資訊，我們僅抓取離使用者輸入的地址最近的五個點。此外，環保類別中顯示的資料，其位置與輸入地址的距離，皆會在使用者選擇的距離限制之內。

3. 治安

此類別包含六種竊盜案件（自行車竊盜、機車竊盜、汽車竊盜、住宅竊盜、強盜與搶奪）和交通事故的紀錄。上述資料位置與輸入地址的距離，皆在使用者選擇的距離範圍之內，發生時間範圍也在使用者選擇的限制之內。



圖 1：網站展示圖

除此之外，網頁最下方的地圖上，會標示使用者的房屋地址及環保和治安類別的資料位置，各類別皆有對應的圖針顏色，將滑鼠靠近圖針，會有**提示框**亮起，顯示此位置的詳細資訊。若想**選擇性顯示**類別中的某些資料，可以點選每個項目左側的小圖針以選擇隱藏或顯示。

三、資料收集流程

首先，我們搜尋網路上可供使用的公開資料，找到政府資料開放平臺、行政院環保署等資源豐富的網站；了解有什麼樣的資料能夠取得後，我們便開始確認需要納入我們資料庫中的資料；接著透過**爬蟲**及**直接下載**資料集的方式取得資料，輸入到資料庫中；最後，進行資料型態轉換、運算和篩選等步驟，讓資料更符合我們的需求。

以下是我們最後收集到的資料集，其中僅 **2.6** 的資料集是我們透過網頁爬蟲取得的，其他都是從網站上直接下載。

1. 環境指標

- (1) 空氣品質指標 **AQI**：行政院環保署空氣品質監測網
- (2) 紫外線指數：行政院環保署紫外線紫外線即時監測資料
- (3) 水庫水質卡爾森指數：行政院環保署全國環境水質監測資訊網

2. 環保

- (1) 垃圾車停靠點：政府資料開放平臺
- (2) 舊衣回收箱：政府資料開放平臺
- (3) 公民營廢棄物清除機構：行政院環保署
- (4) 綠色友善餐廳：政府資料開放平臺
- (5) 綠色商店：行政院環保署
- (6) 自備餐具優惠店家：行政院環保署的環保優惠店家查詢網

3. 治安

- (1) 竊盜案件（包含自行車竊盜、機車竊盜、汽車竊盜、住宅竊盜、強盜與搶奪，共六種竊盜類型）：政府資料開放平臺
- (2) 交通事故：政府資料開放平臺

四、資料庫架構

我們的資料庫共有**二十張**資料表，整體而言，是由數個環境指標、環保店家、生活機能相關的個體所組成。

設計資料庫的原則與正規化的方式都是來自於本學期的課程內容，以下進行詳細介紹。

（一）ER Diagram

我們的資料庫並不像常見的公司資料庫，是由最小單位（員工）組成較大的單位（部門）；我們核心的個體是房屋（地址），然而它並不會被儲存在資料庫中，而是每次使用者要查詢時，才會輸入。因此，我們的資料庫僅有多種地址以外的個體，而沒有這些個體和地址的關聯表。以下詳細介紹資料庫中的個體及關係。

1. 環境指標

- (1) 水庫觀測站（DAM_SITE）：記錄全台灣各個水庫的水質觀測站資訊，屬性包含水庫編號（Did）、名稱（Dname）及所屬縣市（Dcounty），其中編號及名稱為鍵值屬性。
- (2) 水庫觀測資料（DAM_OBSERVE）：記錄各個水庫在不同時間的水質觀測資料，屬性有水庫編號（Did）、觀測時間（Dtime）、透明度（Sd）、總磷（Tp）、葉綠素（Chl_a）、卡爾森指數（Ctsi）、藻類數（Algae）及水質狀態（Status）。
- (3) 水庫觀測站會觀測水庫水質並記錄觀測資料，兩者之間的關聯以 **OBSERVES** 表示。一個觀測站會觀測到多筆資料，因此是 **1：N** 關係。每個觀測站必有觀測紀錄，每筆紀錄也必由一個觀測站所觀測出來。
- (4) 空氣觀測站（AIR_SITE）：全台各空氣觀測站資訊。屬性包含編號（Sid）、名稱（Sname）、所屬縣市（Scounty）、經度（Slon）和緯度（Slat），其中編號及名稱為鍵值屬性。
- (5) 空氣觀測資料：屬性有觀測站編號（Site_id）、發布時間（Published_time）、空氣品質指標（Aqi）、空氣狀態（Status）及各項空氣組成成份的濃度。
- (6) 空氣觀測站會觀測空氣品質並記錄觀測資料，兩者之間的關聯以 **OBSERVES** 表示。一個觀測站會觀測到多筆資料，因此是 **1：N** 關係。每個觀測站必有觀測紀錄，每筆紀錄也必由一個觀測站所觀測出來。
- (7) 紫外線觀測站（UV_SITE）：全台各紫外線觀測站資訊。鍵值屬性為測站編號（Site_id）及名稱（Site_name），屬性有縣市（County）、經度（Ulon）和緯度（Ulat）。
- (8) 紫外線觀測資料（UV_OBSERVE）：屬性含測站編號（Uid）、觀測時間（Utime）與紫外線指數（Uvi）。
- (9) 紫外線觀測站會觀測紫外線並記錄其指數，兩者之間的關聯以 **OBSERVES** 表示。一個觀測站會觀測到多筆資料，因此是 **1：N** 關係。每個觀測站必有觀測紀錄，每筆紀錄也必由一個觀測站所觀測出來。

2. 治安

- (1) 交通事故 (ACCIDENT)：此個體為交通事故的案件記錄，鍵值屬性為案件編號 (Aid)，屬性包含事故日期 (Adate)、發生地點經度 (Alon)、緯度 (Alat)，以及該事故的受傷人數 (Injured) 和死亡人數 (Death)。
- (2) 竊案 (THEFT)：目前僅記錄台北市發生的竊案。鍵值屬性為編號 (Theft_id)，屬性有案發時間 (Theft_date)、案發地點經度 (Theft_lon) 和緯度 (Theft_lat)、竊案類型 (Theft_type)，其中竊案類型可分為搶奪、強盜、住宅竊盜、汽車竊盜、機車竊盜和自行車竊盜。

3. 環保

- (1) 舊衣回收箱 (CLOTHES)：記錄合法設立於台北市的舊衣回收箱的資訊。鍵值屬性為編號 (Cid)，並有經度 (Clon)、緯度 (Clat) 和設立團體編號 (Group_id) 屬性。
- (2) 慈善團體團體 (CGROUP)：屬性為編號 (Group_id) 及團體名稱 (Group_name)，兩者皆為鍵值屬性。
- (3) 慈善團體設立了舊衣回收箱，兩者間的關係以 **SETS_UP** 表示。一個慈善團體會設立多個舊衣回收箱，因此是 **1:N** 關係，且表中每個慈善團體必都有設立舊衣回收箱，舊衣回收箱也一定由某個團體所設立。
- (4) 綠色商店 (GREEN_STORE)：綠色商店為販售可回收、低汙染環保商品的商店。鍵值屬性為商店名稱 (Gname)，屬性有商店經度 (Gslon) 和緯度 (Gslat)、商店類型 (Store_type)，商店類型分為餐廳、飲料店和其他商店。
- (5) 環保獎勵商店 (REWARD_STORE)：環保獎勵商店是透過給予折扣以鼓勵消費者使用環保餐具、環保杯的商店。鍵值屬性為店家名稱 (Rname)，屬性含經度 (Rlon) 和緯度 (Rlat)、折扣條件 (Requirement)、折扣內容 (Discount) 和商店類型 (Store_type)。
- (6) 垃圾車停靠站點 (TRASH)：紀錄垃圾車停靠地點及停留時間。鍵值屬性為車牌號 (License_num)，屬性有停靠點經度 (Tlon) 和緯度 (Tlat)、抵達時間 (Arrival) 及離開時間 (Leaving)。
- (7) 廢棄物處置機構 (WASTE DISPOSAL)：紀錄各間廢棄物處理機構的資訊，機構名稱 (Wname) 是鍵值屬性，並有經度

（Wlon）和緯度（Wlat）、廢棄物處理方法（Wmethod）屬性。

下圖是我們的 ER Diagram 。

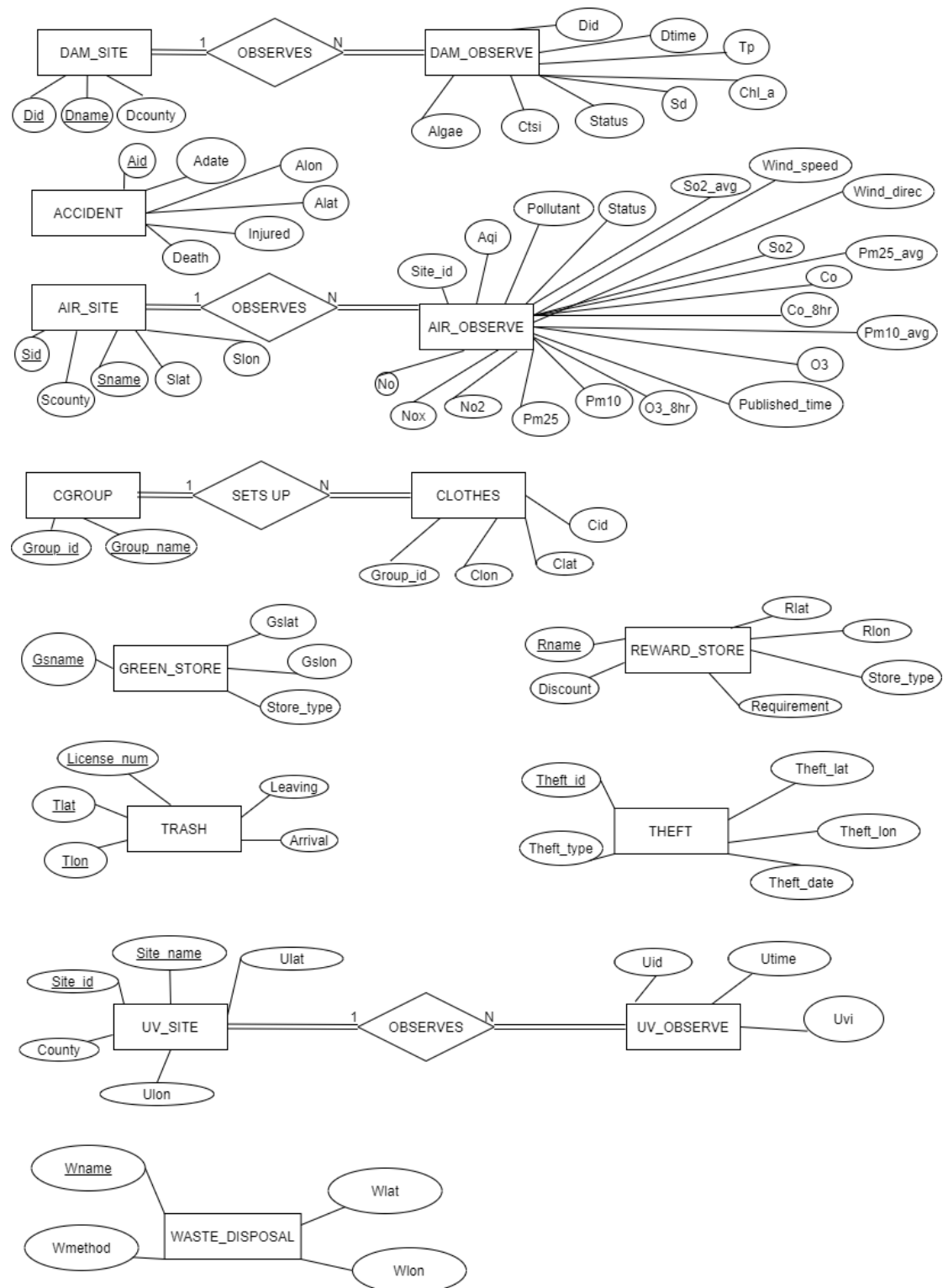


圖 2：ER Diagram

（二）DB Schema

資料表的欄位名稱及資料型別展示於下面兩張圖片中，由於前文已詳細介紹所有資料表的屬性，接下來僅說明我們進行的正規化。

1. 減少重複資料

在原本的垃圾車停留站點資料中，同樣的垃圾車駕照會出現在多筆資料裡，因此我們將車牌號碼獨立成一張表，並給予編號，將編號作為主鍵，而所有垃圾車會停靠的地點則成為另一張表，透過外鍵（車牌號碼編號）指向記錄車牌號碼的表。

竊案類型、商店類型和環保商店折扣也會不斷重複出現在表中，因此我們將這些資訊獨立成表，給予編號作為主鍵，而原本有這些資訊的表則用編號作為外鍵，指向我們新建立的表。

2. 2NF

由於原本的資料來源將水庫觀測站、空氣觀測站和紫外線觀測站資訊，與其觀測到的資料記錄在同一張資料表中，因此表中有大量的重複資訊（如：測站名稱、經緯度），且非主鍵的屬性會**完全功能相依**於主鍵（測站名稱及觀測時間）以外的屬性，例：測站經緯度功能相依於測站名稱，而非相依於測站名稱及觀測時間，因此我們將資料分為測站資訊及觀測紀錄兩張表，以符合 2NF 的要求。

測站表的主鍵是編號，指向其的外鍵則是觀測紀錄表的測站編號。

3. 3NF

本來的舊衣回收箱表中，舊衣回收箱編號會功能決定經緯度和設立團體名稱，而設立團體名稱又**功能決定**團體電話等團體資訊，不符合 3NF，因此我們將團體資訊獨立成表，接著刪除我們不需要的欄位，僅留下作為主鍵的編號和團體名稱。指向這主鍵的外鍵則是在原本表中新增的欄位（團體編號）。

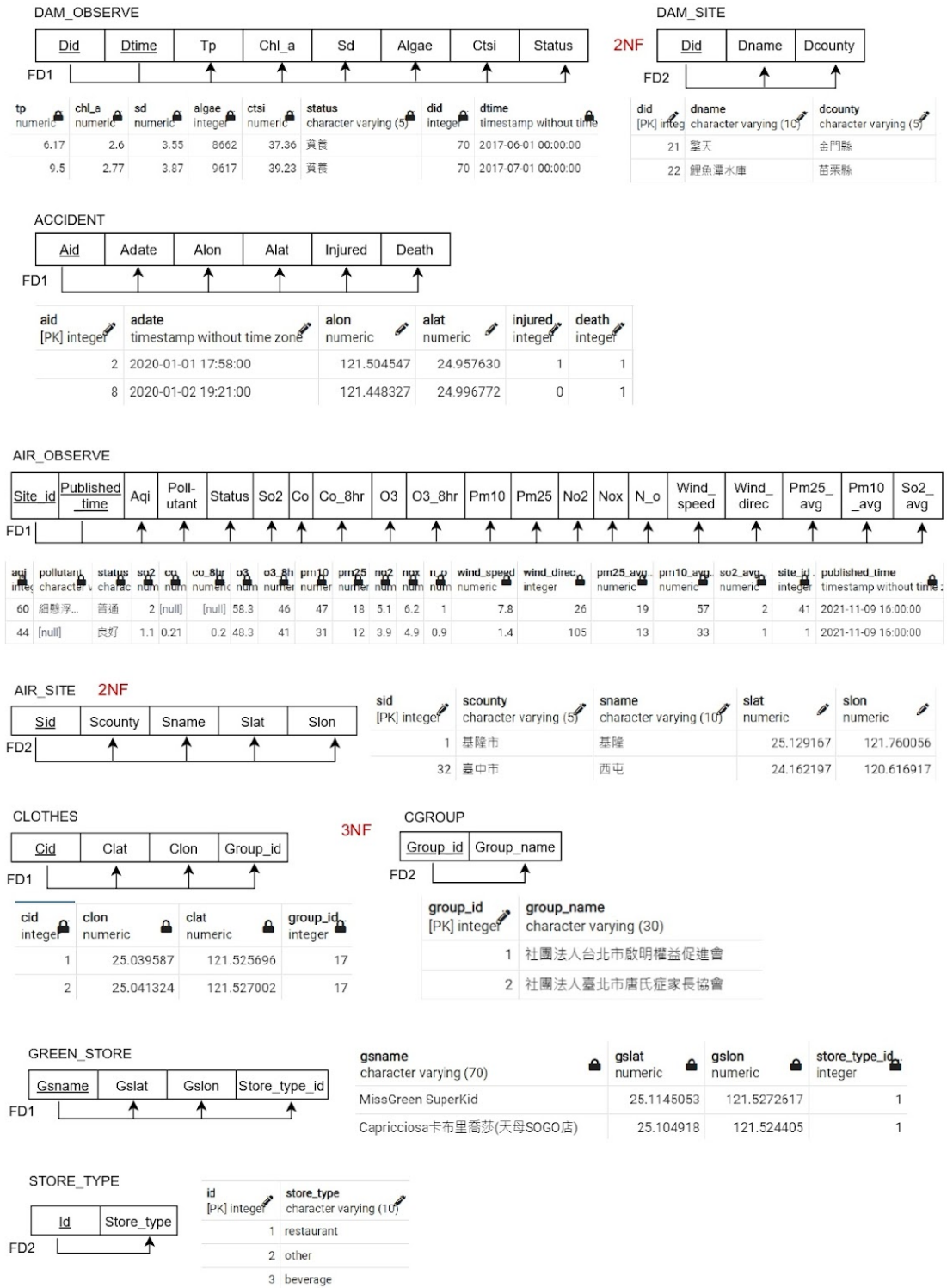


圖 3.1 : DB Schema

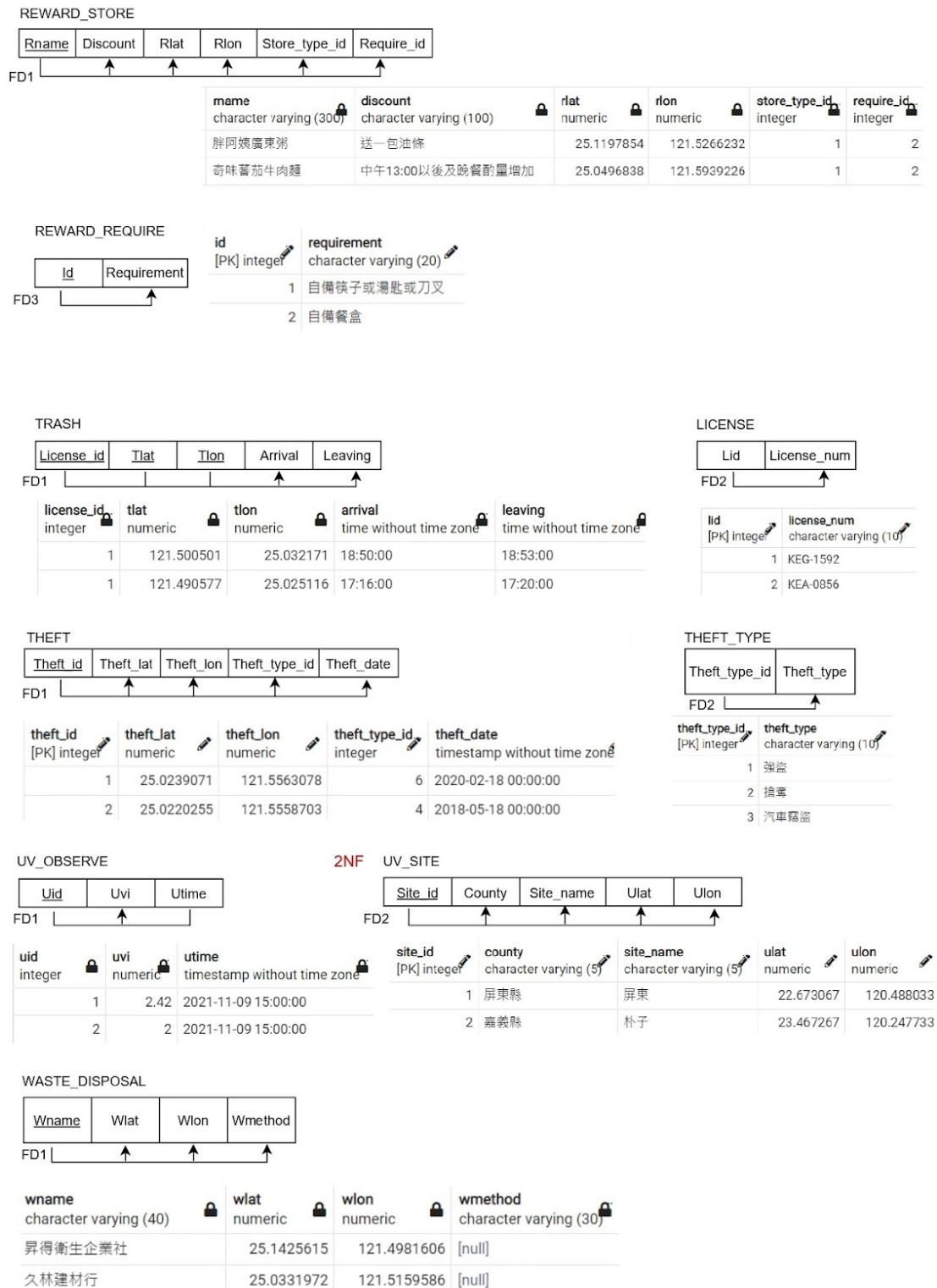


圖 3.2 : DB Schema (續)

五、網站架構

我們透過 REST API，實現網頁的前後端分離。首先，使用者在網頁輸入想要查詢的地址，並且設定時間與距離等參數。按下搜尋之後，前端會先利用 **Google Maps API** 提供的服務，將文字地址轉換成一組經緯度，再結合時間或者距離，呼叫後端提供的 API，從資料庫提取與使用者查詢地點相近的資料，標示在頁面上的地圖中。使用者可以從地圖上的資訊，更加具體的了解當地的生活環境。

（一）後端架構

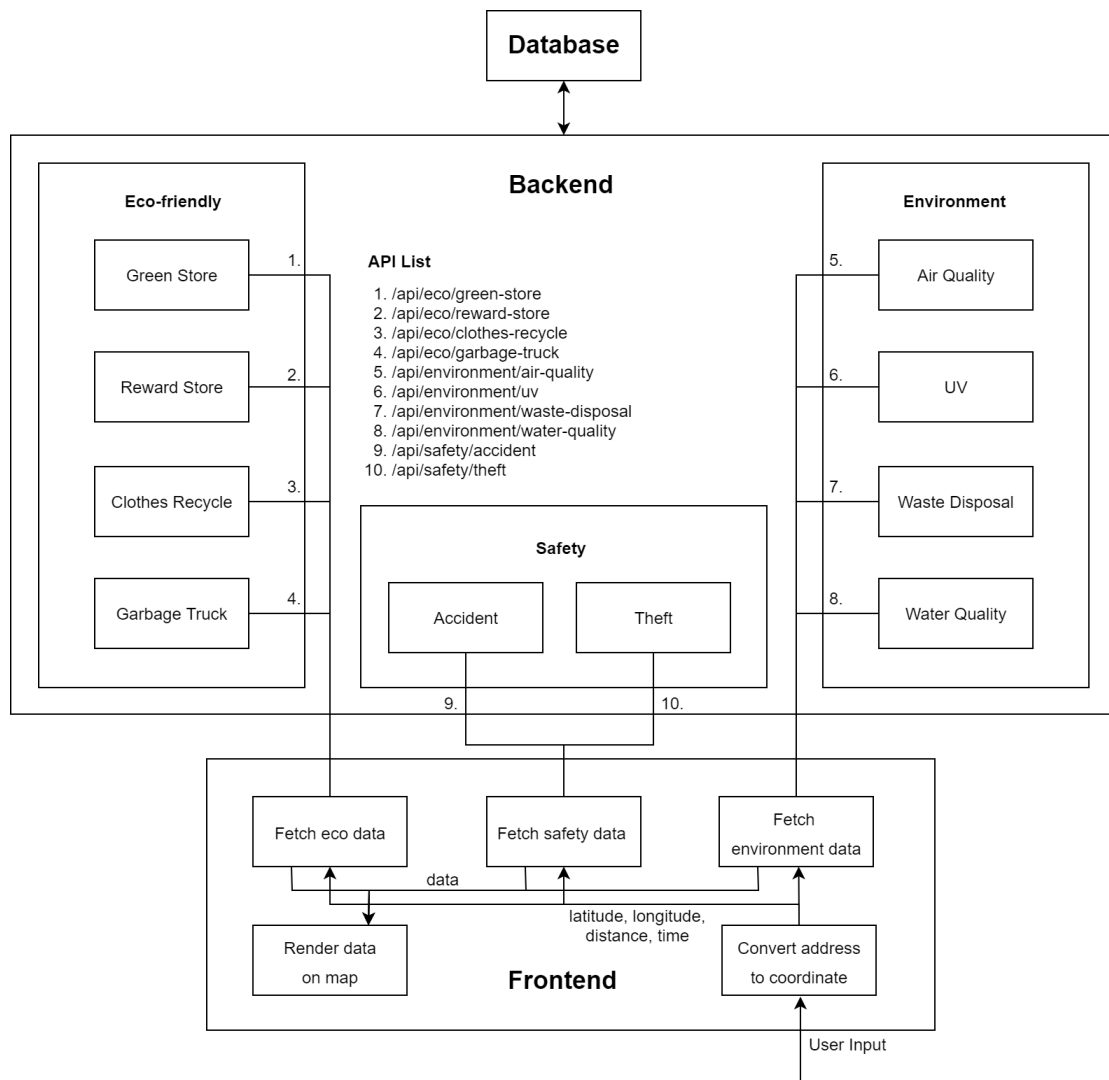


圖 4.1：網站架構圖

依據資料的屬性，後端的架構主要可以分為三個面向，第一個面向是環保（**Eco-friendly**），底下包含綠色商店、環保優惠商店（環保優惠例如：消費者自備餐具可以加料不加價）、舊衣回收箱位點、垃圾車停靠地點四個子項；第二個面向是環境（**Environment**），底下包含空氣品質、紫外線指數、廢棄物

處理廠、水質四個子項；第三個面向是治安，包含交通事故與竊盜事件兩個子項。每一個面向分別對應到 Django 框架中的一個 App。

（二）API 列表與說明

詳細資料範例請參考 [readme](#)。

1. 綠色商店

- 簡述：回傳查詢地點指定距離內的綠色商店
- 路徑：`/api/eco/green-store`
- 類別：POST
- 參數：經度、緯度、距離
- 回傳資料：陣列，陣列裡的每個物件包含商店名稱、商店地址、商店經緯度、商店類型

2. 環保優惠商店

- 簡述：回傳查詢地點指定距離內的環保優惠商店
- 路徑：`/api/eco/reward-store`
- 類別：POST
- 參數：經度、緯度、距離
- 回傳資料：陣列，陣列裡的每個物件包含商店名稱、折扣要求、折扣內容、商店經緯度、商店類型

3. 舊衣回收箱

- 簡述：回傳距離查詢地點最近的幾個舊衣回收箱
- 路徑：`/api/eco/clothes-recycle`
- 類別：POST
- 參數：經度、緯度、數量
- 回傳資料：陣列，陣列裡的每個物件包含回收箱編號、地址、經緯度、營運機構

4. 垃圾車停靠點

- 簡述：回傳距離查詢地點最近的幾個垃圾車停靠點
- 路徑：`/api/eco/garbage-truck`
- 類別：POST
- 參數：經度、緯度、數量
- 回傳資料：陣列，陣列裡的每個物件包含垃圾車編號、停靠地點、經緯度、抵達時間、離開時間

5. 空氣品質

- 簡述：回傳距離查詢地點最近觀測站的空氣品質觀測資料
- 路徑：`/api/environment/air-quality`

- 類別：POST
- 參數：經度、緯度
- 回傳資料：一個物件，包含 AQI 指標、空氣品質描述、觀測站所在位置、觀測時間

6. 紫外線指數

- 簡述：回傳距離查詢地點最近觀測站的紫外線品質觀測資料
- 路徑：/api/environment/uv
- 類別：POST
- 參數：經度、緯度
- 回傳資料：陣列，陣列裡的每個物件包含紫外線指標、觀測時間、觀測地點

7. 廢棄物處理廠

- 簡述：回傳查詢地點指定距離內的廢棄物處理廠
- 路徑：/api/environment/waste-disposal
- 類別：POST
- 參數：經度、緯度、距離
- 回傳資料：陣列，陣列裡的每個物件包含廢棄物處理廠編號、名稱、地點、處理方式、經緯度

8. 水質

- 簡述：回傳距離查詢地點最近水庫的水質檢測資料
- 路徑：/api/environment/water-quality
- 類別：POST
- 參數：經度、緯度、時間範圍
- 回傳資料：陣列，陣列裡的每個物件包含總磷濃度、葉綠素 a 濃度、透明度、卡爾森指數、水質狀態、水庫編號、檢測時間

9. 交通事故

- 簡述：回傳指定地點與時間內發生的交通事故
- 路徑：/api/safety/accident
- 類別：POST
- 參數：經度、緯度、距離、時間範圍
- 回傳資料：陣列，陣列裡的每個物件包含交通事故編號、地點、經緯度、傷亡人數、日期

10. 竊盜

- 簡述：回傳指定地點與時間內發生的竊案
- 路徑：/api/safety/theft
- 類別：POST

- 參數：經度、緯度、距離、時間費為
- 回傳資料：陣列，陣列裡的每個物件包含竊案編號、發生地址、經緯度、日期、竊案類別（可能是自行車竊盜、機車竊盜、汽車竊盜、住宅竊盜、強盜、搶奪）

（三）前端架構

我們在前端 **React App** 中定義了幾個元素以區分不同區塊的功能，資料會在各個元素間以 **hook** 的方式傳輸。

當 **Form** 接收到使用者輸入的資料後，會先利用 **Google Map API** 將地址轉為經緯度，接著將此經緯度當作參數，利用三個 **fetch** 檔案獲取三個類別的資料，再分別將這些資料傳進 **Environment**、**Eco**、**Safety**、**Map** 四個元素中，使資料內容能在個別元素中被顯示。

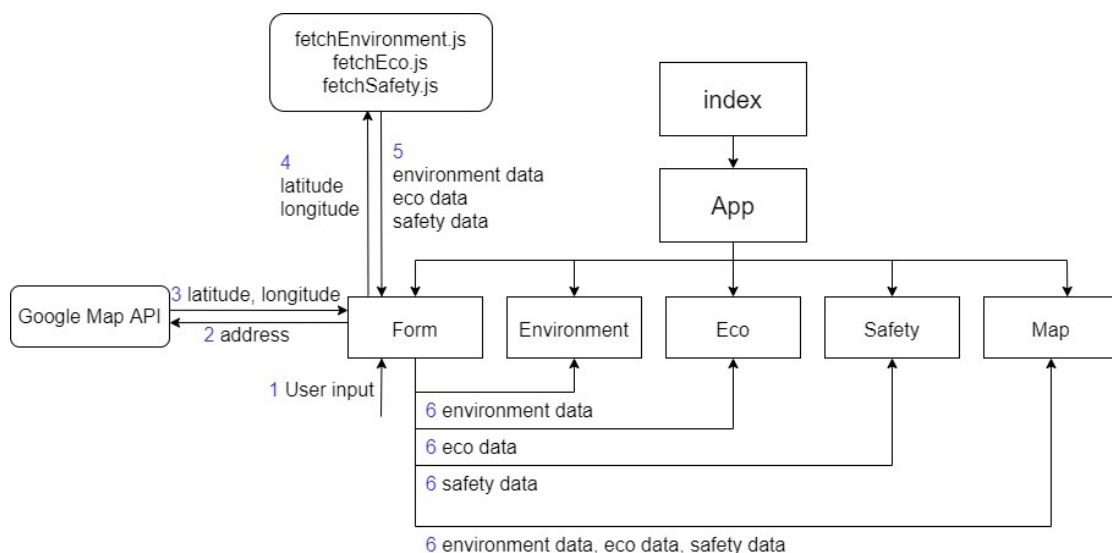


圖 4.2：前端 App 架構

六、個人心得

（一）周敦翔

在這個專案中，我主要是負責網頁後端的部分。之前修別的課的時候，也有做過網頁相關的專題，只是當時寫的是前端的部分，花了很多時間調整頁面上的每個元素的位置，但可能我經驗尚淺，所以它們常常不依照我的意思移動，搞到後來很崩潰，花最多時間的部分反而不是寫程式。這次有機會寫後端實在是一件開心的事。

我自己的感覺，這次專案最困難的部分，第一是決定架構，一開始我們這組沒有人有完整開發網頁的經驗，所以花了一點時間研究相關的技術和框架，後來才著手討論哪些工作由前端來做、哪些由後端負責。一旦決定好架構、了解前後端如何透過 **API** 互動之後，我認為後端剩下的部分算是進行得相當順利。雖然有些資料可能要找最近的幾個地點、有些要找指定範圍內的地點、有些還要篩選的時間，不過處理邏輯相似，只要寫好一個部份，剩下的只要類推便能完成。

第二則是資料，我們的資料表為數不少，而且每個主題的資料來源不同，有些要從台北市政府的網站下載，有些是環保署的開放資料集，還有一些要寫爬蟲從網頁上擷取。將這些資料儲存在電腦後，還要決定哪些欄位是必要的，哪些和我們有興趣的主題無關，不需要匯入資料庫。此外，許多資料表一開始是沒有經緯度的，如果等到使用者查詢，才及時的呼叫 **google map api**，將地址轉換成經緯度，勢必會造成查詢時間數倍的增長；同樣的地點會經歷數次重複的轉換，更是另外一種負面的消耗。所以，在整理資料的時候，必須額外查詢、匯入經緯度。再來，原始的資料有些不符合正規化的規範，轉換又需要另外一番工夫。

網頁架構和整理資料這些基礎的工作完成之後，後續的部分應該算是漸入佳境吧！

（二）李佳欣

我是負責設計資料庫和匯入資料的部分。從第一次團體作業一開始就發現設計資料庫沒想像中簡單，看似很簡單的資料，其實有很多冗餘重複或太多的功能相依，雖然我數度深感正規化很麻煩，但也在處理資料的過程中體認到 **SQL** 作為處理資料的語言，究竟有多麼方便——從簡單的匯入 **CSV** 檔案，到轉換資料型態、進行運算和彙總，**SQL** 可以輕易達成這些事情。

期待後續的課程能讓我們更加了解關於效率及安全性的議題，這樣能讓我們在進行資料正規化時更加心甘情願，以避免後續遭遇更大的麻煩。最後仍要不免俗地感謝組員，我們很幸運地有達到專業分工（有吧？），在溝通及分工的過程中也沒發生爭執，合作愉快。

（三）錢紫翎

在這次專案中，我負責前端網頁的設計和開發，因為後端的 **API** 寫得很好，因此在獲取資料的部分很快就做完了，我反而最苦惱網頁的呈現方式，為了想讓使用者更自由和直覺地去使用網站，我下了很多功夫在介面上，雖然最後成果

還是略顯陽春，但也有達成想讓使用者在我們的網站上可以得到所有他想得到的資訊這個目標，這樣我就心滿意足了！

這個專案給了我很好的實作 **React App** 的經驗，讓我對資料的傳輸更熟悉了，這是我很大的收穫，不過這次沒有碰到資料庫處理的部分還是有點可惜，希望期末專案能使用更多的資料庫，檢驗這學期學習的成果！

（四）莊翔安

我認為最困難的還是初期選擇題目和方向的時候，也是我覺得最迷惘的時候，因為總會想到一個可行的題目，卻因為沒有經驗不知道能不能負荷。這次選擇題目的時候最初在老電影收錄網站和租屋網資料中做選擇，選定以租屋網為目標後卻發現各大租屋網資料已經足夠齊全，我們可增加的功能寥寥可數，租屋網也有聲明不得透過爬蟲獲取他們的資料，至此是一個重大的危機，幸好最後轉了一個彎，改以環保和綠色愛地球的方式呈現。

因為沒有學過任何一個部分，讓我看到助教範例成品的時候十分茫然，根本不能了解到底個部分是如何交流，他們又處理了哪些部分。幸好組員們本身有點經驗，能有效率的分工，我完全是被他們拯救的小白呀.....

專業分工的情形下，我負責資料處理和資料爬蟲的部分，結果出乎大家意料，許多我們所需要的資料政府部門都有不斷在收集和更新，也讓我的工作量確實直直下降，爬蟲的過程也算順利，加了幾個假 **header** 就成功地獲取網頁上所需的資料。所以說這次專案是被其他組員凱瑞的真的不為過了，期末要好好貢獻更多自己的心力才行。

另外因為這門課，我陷入了環境設定的地獄，甚至為此格式化電腦打掉重來。也讓我明白之前在安裝和設定應用程式和套件時都沒有注意環境的問題，確實也是一個另類的收穫。