

組員

系級	學號	姓名
工海二	B10505059	陳琪予
工海二	B10505027	簡郁庭
工海二	B10505050	張鈞民
工海二	B10505022	鄧志彥

系統架構及功能簡介

專案題目：電動車租借平台

- 角色分為二：1.租車使用者 2.維修者人員
- 租車使用者的功能簡介：
 - # 會員登錄及註冊頁面：
 - 1.設定帳密、手機號碼、信用卡資訊 2.查看租換車歷史紀錄 3.查看優惠券
 - #查詢附近哪裡有車：
 - 1.可設定附近幾公里範圍內 2.壞掉的車不能查詢 3.資訊包含：車牌號碼、車位置、電量
 - #隨時隨地租車還車：
 - 1.還的點若太遠，電量不足就無法還車 2.還車付款時可選擇優惠券 3.金額用距離來算
 - #幫車充電：
 - 1.充電可回饋優惠券（不能當次使用） 2.充電的點若太遠，電量不足就無法充電
- 租車使用者的使用情境：
 - 1.租車：顯示經緯度、電量 2. 使用電動車時，會以 10 秒更新一次隨機位置、扣除電量、累積里程數 3.幫車充電，充完得到優惠券 4.還車顯示目前位置與實際使用里程數並顯示價格
- 維修人員的功能簡介：1.查詢故障資訊 2. 將故障車狀態調整為維修中或正常 3.幫車電量低於 20%的車充電，充完隨機放置

使用到的套件、技術

Android Studio：

- 開發 **Android** 應用程式的集成開發環境（**IDE**）。
- 提供了開發 **Android** 應用程式所需的工具、編輯器、除錯器等。
- 可以建立、編輯、編譯和執行 **Android** 應用程式。

JDBC：

- 是 **Java** 語言與資料庫之間的 **API**，用於連接和操作關聯式資料庫。
- 提供了一組類和方法，使 **Java** 應用程式能夠執行資料庫相關操作，如連接資料庫、執行 **SQL** 查詢、更新資料等。
- 在 **Android** 開發中，可以使用 **JDBC** 與支援 **JDBC** 驅動程式的資料庫進行連接和操作。

SQLite：

- 是一個輕量級的嵌入式關聯式資料庫管理系統。
- 適用於在本地存儲和操作資料，特別是在移動設備上。
- **Android** 平台內建支援 **SQLite**，並提供了相關 **API** 和類來創建、查詢和更新 **SQLite** 資料庫。

Google Map API：

- 是 **Google** 提供的地圖服務 **API**，用於在應用程式中顯示和操作地圖相關功能。
- 可以在應用程式中顯示地圖、標記位置、獲取地點資訊、導航等。
- 在 **Android** 開發中，您可以使用 **Google Map API** 來集成地圖功能到您的應用程式中。

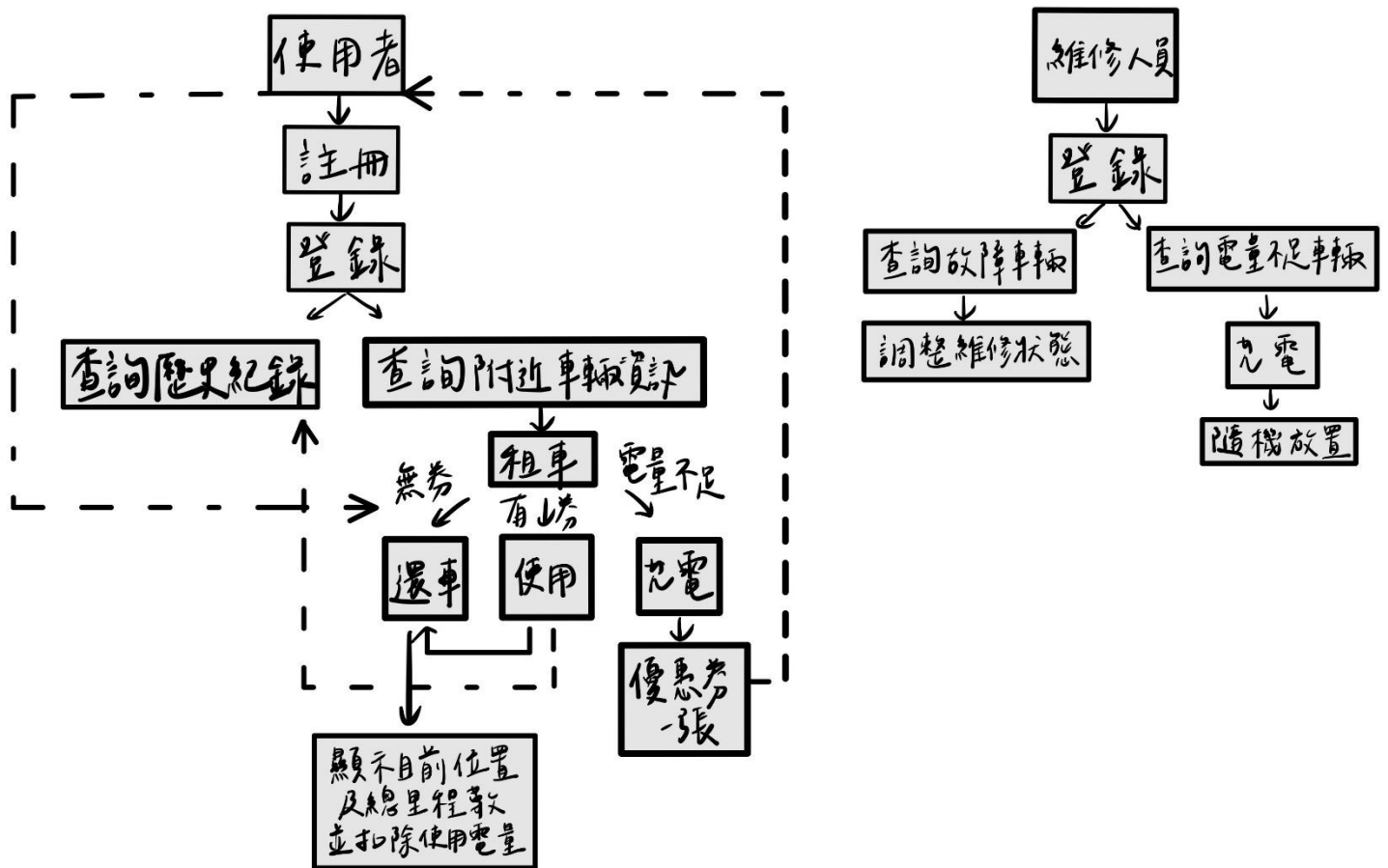
Chat GPT

- 輔助協助完成專案。

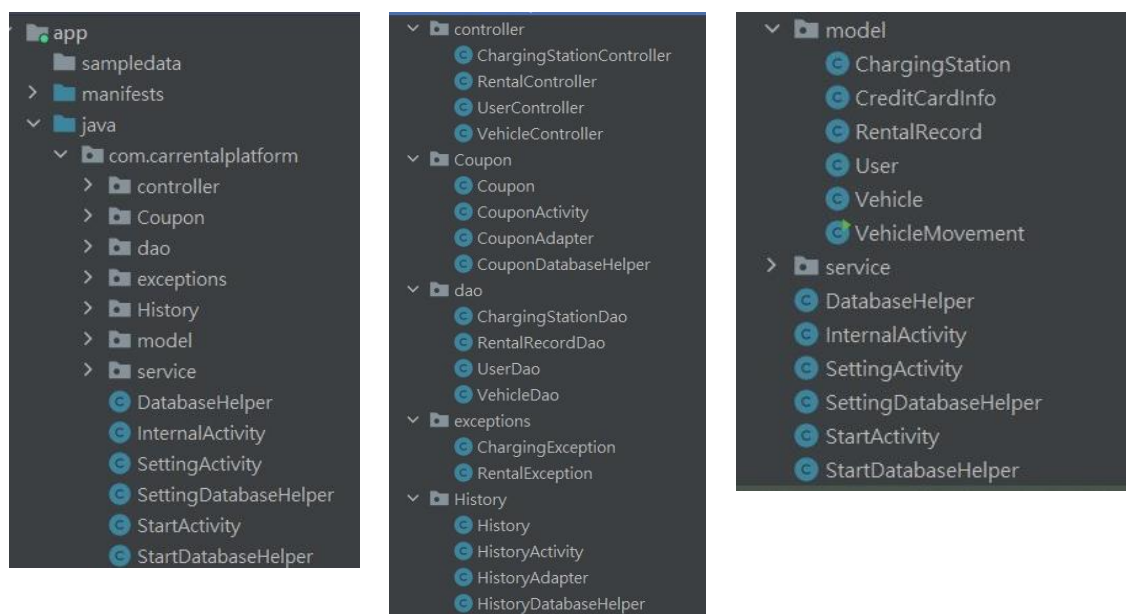
彼此的關聯：

Android Studio 開發 **Android** 應用程式，並使用 **Google Map API** 來在應用程式中顯示地圖，標記位置和實現相關地圖功能。在應用程式中使用 **SQLite** 作為本地資料庫，並使用 **JDBC** 來連接和操作 **SQLite** 資料庫。

流程圖



Package 規劃(Optional)



地圖 API (Optional)

1.建立 Google Cloud Platform (GCP) 帳戶：

- a.訪問 Google Cloud Platform 並創建一個帳戶。
- b.在 GCP 控制台中，創建一個新的專案，並啟用 Maps JavaScript API。

2.獲取 API 金鑰：

- a.在 GCP 控制台中，獲取 Maps JavaScript API 的 API 金鑰。
- b.記下此金鑰，稍後將在 Android Studio 中使用。

3.在 Android Studio 中設置 Google Play Services：

- a.在專案的 build.gradle 文件中，添加 Google Play Services 的依賴項。
- b.在 build.gradle 文件的 dependencies 區塊中添加以下代碼：

```
implementation 'com.google.android.gms:play-services-maps:17.0.0'
```

4. 在 AndroidManifest.xml 中設置權限和 API 金鑰：

添加了必要的權限、網絡訪問權限

```
<uses-permission android:name="android.permission.INTERNET" />
```

添加 Maps JavaScript API 的 API 金鑰

5.使用 MapFragment 通過程式碼動態添加到佈局中

6. 在 Activity 中設置地圖

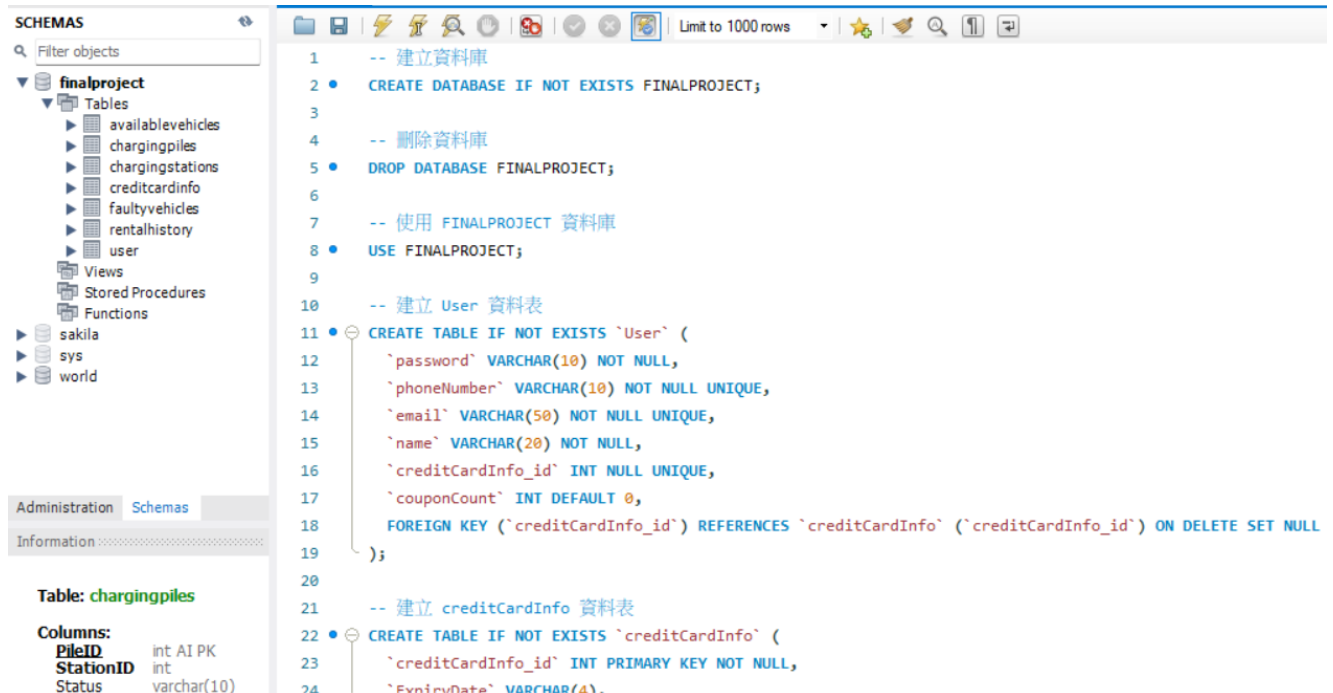
7. 在地圖上顯示車輛資訊

資料庫設計(Optional)

一開始我們組是用 MySQL 去與 Android Studio 做連接，因為：

- 1.MySQL 是一個強大的關聯式資料庫管理系統（RDBMS），被廣泛用於大型企業和網路應用。
- 2.它支援高並發性、可擴展性和安全性，適用於需要處理大量數據和複雜查詢的場景。
- 3.MySQL 提供了豐富的功能和工具，如存儲過程、觸發器、視圖等，使得資料庫管理和操作更加靈活和方便。

4.如應用需要多用戶訪問、大量資料存儲和複雜的數據操作，MySQL 可能是一個更好的選擇。



但後來我們發現 SQLite 其實更適合拿來開發 Android Studio，因為以下幾點原因：

1. 預安裝：SQLite 是 Android 平台的一部分，因此無需額外安裝或配置，開發人員可以直接使用。
2. 輕量級：SQLite 是一個輕量級的嵌入式資料庫引擎，專為嵌入式設備和移動應用而設計。它佔用的空間很小，且運行效能優秀，非常適合在移動設備上使用。
3. 簡單易用：Android 提供了一個方便的 API，使得在 Android 應用程式中使用 SQLite 非常簡單。開發人員可以輕鬆地建立、查詢和更新資料庫，並通過內置的 SQLiteOpenHelper 類來管理資料庫的創建和版本升級。

名稱	類型	架構
資料表 (9)		
AvailableVehicles		CREATE TABLE AvailableVehicles (no TEXT PRIMARY KEY,
no	TEXT	"no" TEXT
power	INTEGER	"power" INTEGER NOT NULL
lat	REAL	"lat" REAL NOT NULL
lng	REAL	"lng" REAL NOT NULL
ChargingPiles		CREATE TABLE ChargingPiles (PileID INTEGER PRIMARY KEY,
PileID	INTEGER	"PileID" INTEGER
StationID	INTEGER	"StationID" INTEGER
Status	TEXT	"Status" TEXT
ChargingStations		CREATE TABLE ChargingStations (no INTEGER PRIMARY KEY,
no	INTEGER	"no" INTEGER
lat	REAL	"lat" REAL NOT NULL
lng	REAL	"lng" REAL NOT NULL
Coupon		CREATE TABLE Coupon (id INTEGER PRIMARY KEY NOT NULL,
id	INTEGER	"id" INTEGER NOT NULL
isUsed	INTEGER	"isUsed" INTEGER DEFAULT 0
FaultyVehicles		CREATE TABLE FaultyVehicles (no TEXT PRIMARY KEY, lat
no	TEXT	"no" TEXT
lat	REAL	"lat" REAL NOT NULL
lng	REAL	"lng" REAL NOT NULL
Status	TEXT	"Status" TEXT
power	INTEGER	"power" INTEGER CHECK("power" >= 0 AND "power" <= 100)
RentalHistory		CREATE TABLE RentalHistory (Date TEXT, Tot
Date	TEXT	"Date" TEXT
TotalAmount	REAL	"TotalAmount" REAL
StartLongitude	REAL	"StartLongitude" REAL NOT NULL
StartLatitude	REAL	"StartLatitude" REAL NOT NULL
EndLongitude	REAL	"EndLongitude" REAL NOT NULL
EndLatitude	REAL	"EndLatitude" REAL NOT NULL
Distance	REAL	"Distance" REAL
StartTime	TEXT	"StartTime" TEXT
EndTime	TEXT	"EndTime" TEXT
TotalTime	TEXT	"TotalTime" TEXT
ChargingCount	INTEGER	"ChargingCount" INTEGER
CouponUsed	INTEGER	"CouponUsed" INTEGER DEFAULT 0
LicensePlate	TEXT	"LicensePlate" TEXT
user_email	TEXT	"user_email" TEXT
Staff		CREATE TABLE Staff (password TEXT NOT NU
password	TEXT	"password" TEXT NOT NULL
phoneNumber	TEXT	"phoneNumber" TEXT UNIQUE
email	TEXT	"email" TEXT NOT NULL UNIQUE
name	TEXT	"name" TEXT
User		CREATE TABLE User (password TEXT NOT NU
password	TEXT	"password" TEXT NOT NULL
phoneNumber	TEXT	"phoneNumber" TEXT NOT NULL UNIQUE
email	TEXT	"email" TEXT NOT NULL UNIQUE
name	TEXT	"name" TEXT NOT NULL
creditCardInfo_id	INTEGER	"creditCardInfo_id" INTEGER UNIQUE
couponCount	INTEGER	"couponCount" INTEGER DEFAULT 0
creditCardInfo		CREATE TABLE creditCardInfo (creditCardInfo
creditCardInfo_id	INTEGER	"creditCardInfo_id" INTEGER NOT NULL
ExpiryDate	TEXT	"ExpiryDate" TEXT
creditCardNumber	TEXT	"creditCardNumber" TEXT UNIQUE
Cvv	TEXT	"Cvv" TEXT

4. 本地資料存儲：許多 Android 應用程式需要在本地存儲和操作資料，例如使用者設定、離線資料等。SQLite 提供了可靠的本地資料存儲解決方案，可以輕鬆處理這些需求。

最後我們建了一個名叫 **FINALPROJECT** 的資料庫，並且在內創建 9 個資料表，分別如上圖。

但，事情總有個一波三折，我們最後發現 **Android Studio** 其實有內建資料庫，並用之前已經有的架構去完成。

最後 **Code** 如下圖：

```

public void onCreate(SQLiteDatabase db) {
    // 建立使用者資料表
    String createUserTable = "CREATE TABLE " + TABLE_USER + " ("
        + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + COLUMN_NAME + " TEXT NOT NULL,"
        + COLUMN_PHONE_NUMBER + " TEXT NOT NULL,"
        + COLUMN_EMAIL + " TEXT NOT NULL,"
        + COLUMN_PASSWORD + " TEXT NOT NULL,"
        + COLUMN_CREDIT_CARD_ID + " INTEGER," // 新增与信用卡信息关联的列
        + COLUMN_COUPON_ID + " INTEGER,"
        + COLUMN_History_ID + " INTEGER"
        + ")";
    db.execSQL(createUserTable);

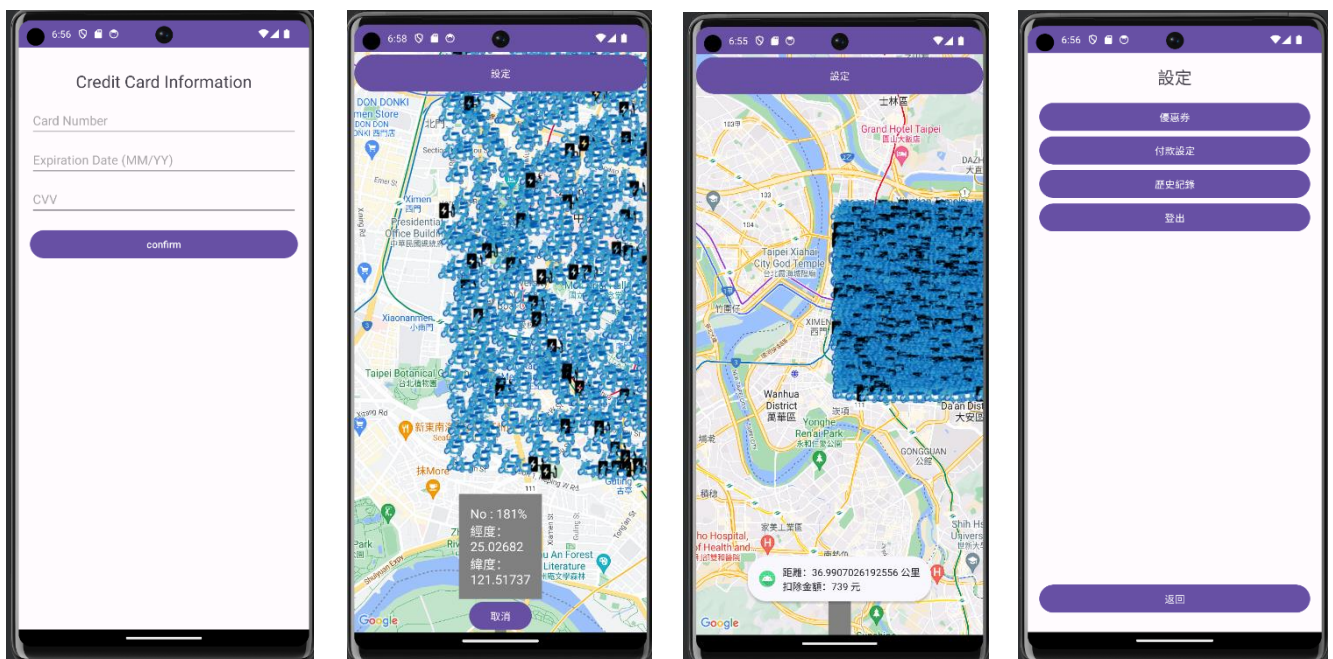
    // 建立信用卡資訊資料表
    String createCreditCardInfoTable = "CREATE TABLE " + TABLE_CREDIT_CARD + " ("
        + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + COLUMN_CREDIT_CARD_NUMBER + " TEXT NOT NULL,"
        + COLUMN_EXPIRY_DATE + " TEXT NOT NULL,"
        + COLUMN_CVV + " TEXT NOT NULL"
        + ")";
    db.execSQL(createCreditCardInfoTable);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // 在此處當作資料庫結構的更新邏輯，如果有需要的話
    // 例如，您可以使用 ALTER TABLE 或 CREATE TABLE IF NOT EXISTS 來修改資料表結構
}

public boolean saveCreditCardInfo(String creditCardNumber, String expiryDate, String cvv) {
    SQLiteDatabase db = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(COLUMN_CREDIT_CARD_NUMBER, creditCardNumber);
    values.put(COLUMN_EXPIRY_DATE, expiryDate);
    values.put(COLUMN_CVV, cvv);
    long result = db.insert(TABLE_CREDIT_CARD, null, values);
    return result != -1;
}

```

APP/GUI 介面截圖(Optional)



註冊及登錄介面、租車地圖介面、還車介面、設定介面

分工表

軟體架構	設計整體系統架構。	共同討論開發(100%)
資料庫	創建資料庫	張鈞民(50%)、簡郁庭(50%)
前端開發	用戶認證、資料庫連結、API	共同討論開發(100%)
後端開發	實現與前端連結	共同討論開發(100%)
介面設計	美觀及用戶體驗	共同討論開發(100%)
報告	PPT、report、demo	共編完成(100%)
