

國立雲林科技大學資訊管理系
資料庫管理系統期末報告企劃書

停車場管理系統
Intelligent Parking Management System

組員：A11223010 劉登豐

A11223032 林冠濤

A11223034 吳聲佑

A11223051 蘇典諺

中華民國 112 年 11 月

Nov, 2023

摘要

本專案旨在解決台灣嚴重的停車位不足問題，特別是在主要城市區域。該問題導致交通擁擠，浪費時間和成本，對環境產生不良影響。我們提出一個智慧停車場管理系統，以促進交通流動性並應對停車位不足的挑戰。

系統開發目標包括提升管理效率、增加收入、解決擁堵問題、安全管理和數據分析。停車場管理系統分析涵蓋使用者需求、可行性分析、增長預期、系統流程分析和系統功能分析。系統功能模組設計包括用戶管理、管理員管理、停車場管理、收費記錄管理和資料統計與報表。

系統架構含括三層，表現層（UI）、業務邏輯層（BLL）、資料層（DL），由表現層將 user 在前端系統功能中所輸入的 I/O 傳遞到業務邏輯層做資料的交互和處理，最終在傳遞到資料層做系統的儲存和管理，將以上的整體系統在 Amazon Web Services (AWS) 雲端平台進行伺服的運作與服務。

關鍵字： 停車、管理、資料庫、AWS、MySQL、MongoDB。

目錄

摘要.....	II
CHAPTER 1 緒論	1
1.1 系統開發的意義	1
1.2 系統開發的目標	1
CHAPTER 2 專案進度與治理	3
2.1 負責項目	3
2.1.1 劉登豐.....	3
2.1.2 林冠濤.....	3
2.1.3 吳聲佑.....	3
2.1.4 蘇典諺.....	3
CHAPTER 3 停車場管理系統分析.....	4
3.1 使用者需求	4
3.1.1 資料輸入需求：	4
3.1.2 資料處理需求：	4
3.1.3 資料輸出需求：	4
3.2 可行性分析	4
3.2.1 雲端平台	4
3.2.2 技術	4
3.3 增長預期.....	5
3.3.1 用戶數量增長：	5
3.3.2 停車場和停車位數據增加：	5
3.3.3 車輛數據變動：	5
3.3.4 收費資料增長：	5
3.4 系統流程分析	6

3.4.1	車輛新增流程	6
3.4.2	車輛收費流程	7
3.4.3	資料修改流程	8
3.4.4	刪除流程	8
3.5	系統功能分析	8
3.5.1	功能性分析	8
3.5.2	非功能性分析	8
CHAPTER 4	停車場管理系統總體設計	10
4.1	系統架構設計	10
4.2	擬定系統功能模組設計	11
4.2.1	整體功能模組設計	11
4.2.2	模組設計	11
4.3	資料庫設計	13
4.3.1	資料庫結構設計	13
CHAPTER 5	連結資料庫	15
5.1	AMAZON RDS - MySQL	15
5.2	EC2 - MONGODB	16
CHAPTER 6	員工管理	17
6.1	員工管理紀錄畫面	17
6.2	員工管理新增功能	17
6.3	員工管理刪除功能	18
6.4	員工管理更新功能	19
CHAPTER 7	車輛管理	20
7.1	車輛管理紀錄畫面	20
7.2	車輛管理新增功能	20

7.3	車輛管理刪除功能	21
7.4	車輛管理更新功能	22
7.5	車輛管理退場功能	23
CHAPTER 8	停車位管理	25
8.1	停車位管理顯示畫面	25
8.2	停車位管理編輯功能	26
CHAPTER 9	收費管理	27
9.1	收費管理顯示畫面	27
9.2	收費管理詳細資料功能	28

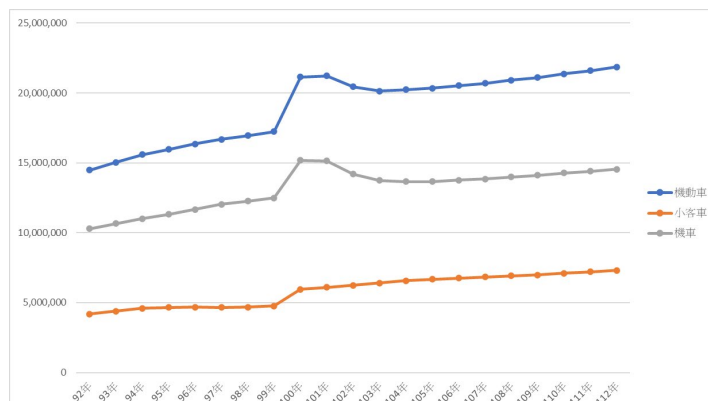
Chapter 1 緒論

1.1 系統開發的意義

台灣多年來，停車位嚴重不足已經是一個根深蒂固的問題，在一些主要城市的市中心區域，面臨著停車位供不應求，特別是商業中心、繁華區以及特定活動場所周邊，隨著城市發展未能有效應對汽車擁有率的快速增長，導致現有的停車設施已難以應對車輛的龐大需求。

根據二零二三年交通部公路局統計，台灣的「機動車輛登記總數」創下新高，高達兩千兩百八十四萬輛，其中小客車約有七百萬輛，機車約有一千四百萬輛，而過去二十年內，台灣人口成長約為一百萬人，但機動車卻增加了約七百三十萬輛，成長率高達五十個百分比。

因此停車位不足直接影響城市交通流動性，導致道路擁擠和交通阻塞，司機在尋找停車位的過程中，增加了行駛路段的徘徊時間，進而影響整個城市的交通效率，交通擁堵帶來的時間浪費和燃油消耗不僅增加了個人成本，還對環境產生不良影響，因此專案期望以「智慧系統」的設計概念，融入停車場的規劃項目，促進交通流動性，同時應對停車位不足的問題。



圖表 1. 機動車輛登記總數(2003-2023)

1.2 系統開發的目標

藉由智慧系統，允許汽車在不需要人類司機操作的情況下進行泊車動作，這項技術的引入旨在提高停車的便利性、安全性和效率，同時大幅減少停車場空間的浪費，

我們將此稱為停車倉儲。

智慧系統原理為在指定的停車位中，利用各種感測器，來捕捉周圍環境的資訊，通過這些感測器收集的數據，車輛可以確定車子的大小、形狀甚至是型號，基於這些數據，車輛的控制系統可以自動執行，將車輛安全地停放在停車倉儲中。

此專案可以帶來的便利包含「提升管理效率」— 實現一個自動化的停車場管理系統，以提升管理效率，這包括自動車輛進出記錄、停車位分配和即時監控等功能，減少人為操作和提高效率、「增加收入和營收」— 減少空間的浪費，並且轉換成可使用的倉儲空間，提高停車倉儲的容納量，以此提高盈利、「解決擁堵問題」— 通過提供即時的停車位信息，幫助車主快速找到可用停車位，減少在停車場內徘徊的時間，進而減緩城市交通擁堵問題、「安全管理」— 可以有效的阻絕車輛被盜或其他車輛有關財產偷竊事件、「數據分析和報告」— 透過數據收集和分析，提供報告和統計資訊，協助地方交通管理，若能將資訊引入「智能交通系統」技術，包括交通信號優化、即時路況信息等，提高交通系統的運作效率，將會是一個互利的策略。

Chapter 2 專案進度與治理

2.1 負責項目

2.1.1 劉登豐

- 期中報告發表人
- 資料模型製作
- 專案進度管控
- 書面報告製作

2.1.2 林冠濤

- 資料查詢及分析
- 書面報告書製作
- 雲端開發維運
- 期末報告發表人

2.1.3 吳聲佑

- 期中口頭報告提問人
- 期末口頭報告提問人

2.1.4 蘇典諺

- 前後端網頁開發
- 關聯式資料庫系統建立
- NoSQL 資料庫系統建立

Chapter 3 停車場管理系統分析

3.1 使用者需求

3.1.1 資料輸入需求：

用戶註冊時需要輸入個人資訊，包括身分證號、姓名、電子郵件地址、手機號碼、地址、生日等。

車輛註冊需要輸入車牌號、車輛品牌、車輛型號、車輛顏色等。

停車位管理中需要輸入停車位資訊，如停車位代號等。

3.1.2 資料處理需求：

系統應能驗證用戶註冊時提供的資訊的正確性，確保唯一性。

管理員能夠進行停車場、停車位、車輛等資料的新增、刪除、修改等操作。

處理車輛入場和離場的時間記錄，計算停車時長和相應的費用。

提供安全性措施，如加密用戶敏感資訊。

3.1.3 資料輸出需求：

生成停車場使用報告，包括各停車場的使用情況、收入等統計數據。

提供用戶的個人資訊、停車位、車輛等詳細信息的輸出。

顯示停車場地圖，以協助用戶找到空車位。

3.2 可行性分析

3.2.1 雲端平台

- Amazon Web Services (AWS)雲端平台

3.2.2 技術

- MySQL
- MongoDB
- PHP
- Apache
- Ubuntu
- AWS EC2

3.3 增長預期

3.3.1 用戶數量增長：

隨著系統的穩定運行，我們預期註冊用戶數量將持續增長。新用戶的加入將豐富系統的用戶群體，提升系統的互動性和活力。

3.3.2 停車場和停車位數據增加：

隨著合作停車場的增多，系統中的停車場和停車位數據將不斷擴展。這將為用戶提供更多選擇，同時提高系統的實用性。

3.3.3 車輛數據變動：

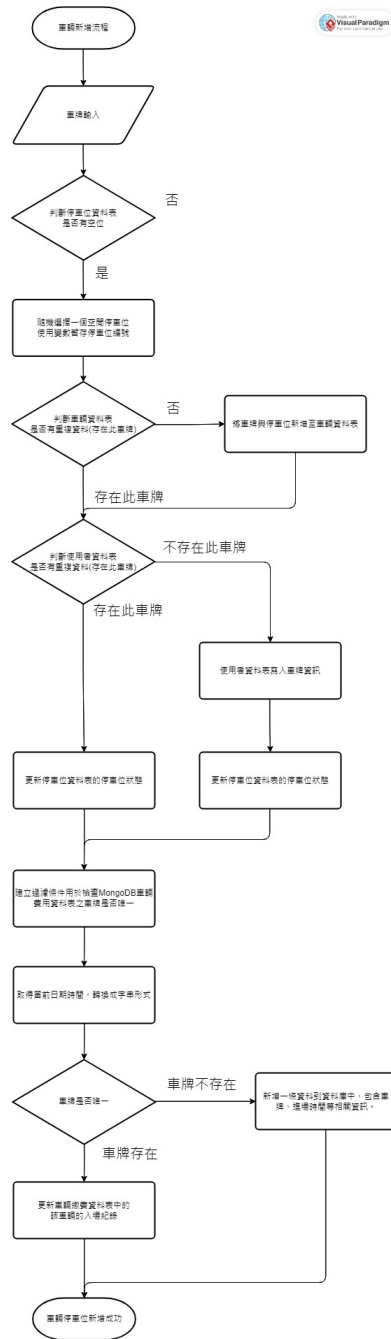
用戶車輛的註冊和離場操作將持續影響系統中的車輛數據。我們預期車輛信息的變動將與用戶活動和停車需求保持一致。

3.3.4 收費資料增長：

隨著停車場的使用量增加，系統中的收費資料將呈現穩健的增長趨勢。這將有助於系統生成更精準的收費報告和統計數據。

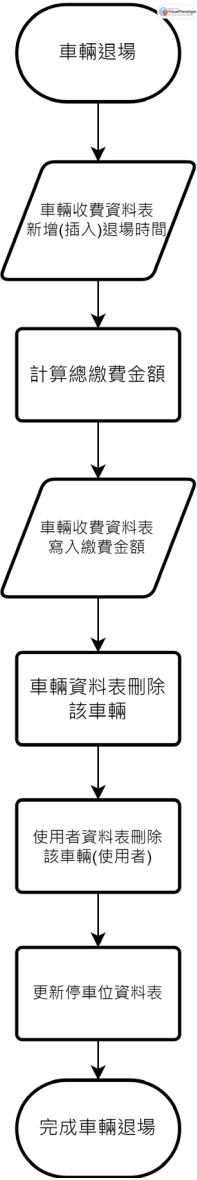
3.4 系統流程分析

3.4.1 車輛新增流程



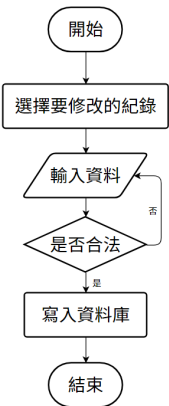
圖表 5. 車輛新增流程圖

3.4.2 車輛收費流程



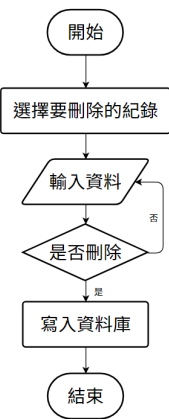
圖表 2. 車輛收費流程圖

3.4.3 資料修改流程



圖表 3. 數據修改流程圖

3.4.4 刪除流程



圖表 4. 數據刪除流程圖

3.5 系統功能分析

3.5.1 功能性分析

停車場管理系統的設計與實現是為了提供使用者更便捷的停車場管理體驗，使其能夠有效地查詢、預訂停車位，並簡化管理員的操作，提高停車場的效率。

3.5.2 非功能性分析

安 全 性	停車場管理系統的安全性是關鍵考量，主要集中在資料庫的安裝和運作。確保資料庫的建置符合相關規範，包括合乎標準的資料加密和使用者權限管理。密碼的設定應符合強度要求，確保用戶資料的安全性。
可	停車場管理系統必須能夠可靠地執行使用者指示，通過系統測試確保可靠

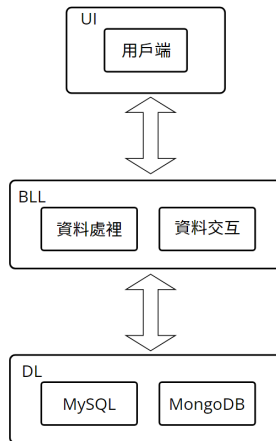
靠性	性達到 90%以上。系統在各種情況下應能正確處理用戶輸入，確保正常運作並預防系統故障。
效能	停車場管理系統的效能是其競爭優勢之一，需確保系統的運行速度和效能達到市場需求水平。優越的性能有助於提高用戶體驗，使系統更具吸引力。
可擴展性	為確保停車場管理系統的長期適應能力，系統應提供可擴展的結構。例如，資料庫應預留多個屬性以應對未來需求的擴大。系統介面的可擴展性也應得到確保，以容納新的功能和使用者需求。
易用性	停車場管理系統的易用性是提高使用者滿意度的重要因素。系統應提供直觀的使用者介面，用戶能夠輕鬆理解和操作系統。操作流程應簡單明瞭，讓用戶能夠迅速上手。
可維護性	停車場管理系統的開發應確保系統的可維護性。這包括易於進行系統更新、修復和升級，以應對未來的變化和需求。系統的程式碼和結構應具有清晰的註釋和文檔，使開發人員能夠方便地進行維護工作。

圖表 5. 非功能性分析

Chapter 4 停車場管理系統總體設計

4.1 系統架構設計

本系統從架構上分為三層：表現層（UI）、業務邏輯層（BLL）、資料層（DL）。



圖表 6. 停車場管理系統架構設計圖

表現層（UI）：停車場管理系統的 UI 層負責提供直觀且易於操作的界面，以使用戶能夠輕鬆使用系統。UI 應該支持不同設備和解析度，確保良好的兼容性。介面設計需要考慮停車場管理系統的不同功能，例如停車位查詢、預訂、支付等，並與業務邏輯層進行良好的整合。

業務邏輯層（BLL）：停車場管理系統的業務邏輯層負責處理用戶的操作和業務邏輯。這包括處理停車位的增加、刪除、修改，車輛的註冊和刪除，預訂和支付的處理，入場和出場的控制，報表和統計的生成等功能。業務邏輯層需與資料層進行交互，確保資料的一致性和正確性。

資料層（DL）：停車場管理系統的資料層主要涉及與資料庫的交互，包括停車場、停車位、車輛、收費等資料的儲存和管理。資料層應該與業務邏輯層進行良好的整合，確保資料的有效存儲和提取。在 MongoDB 中，涉及文檔的創建、查詢、更新和刪除。

4.2 擬定系統功能模組設計

4.2.1 整體功能模組設計

一、實用性強：設計一個符合用戶習慣、界面友好、易學易用的系統，滿足各層次用戶的需求，減少用戶操作的難度。

二、先進的程序層次結構：系統具有先進的程序層次結構，使用當前流行的編程語言和軟體，便於維護和管理。

三、安全可靠：系統具有高度的安全性和可靠性，並提供完善的維護功能。根據不同階段的使用情況，進行相應權限和操作的修改，實現系統的維護，確保系統的安全可靠。

四、模塊化的設計管理：系統採用模塊化的設計，具有良好的擴展性和維護性，以應對不同階段的發展需求。這有助於後續開發者分析和維護系統。

五、操作簡單，維護方便：每個子模塊都具有相對獨立的系統維護功能，使得系統操作簡單，維護方便。同時，系統能夠自行維護可變化的項目。

4.2.2 模組設計

根據停車場管理系統的資料庫表結構，結合實際使用情況，本系統功能模組設計分為如下幾個模組：

一、用戶管理：

1. 使用者通過用戶名和密碼登錄系統，進入停車場資訊展示頁面。
2. 查看個人停車記錄及詳細資訊。

二、管理員管理：

1. 管理員通過用戶名和密碼登錄系統，進入基本資訊頁面，可修改密碼。
2. 管理使用者資訊，包括添加、查看、修改、刪除。
3. 管理停車場/位資訊，包括添加、查看、修改、刪除。

4. 管理車輛資訊，包括添加、查看、修改、刪除。
5. 查看停車場的收費記錄，包括添加、修改、刪除。。

三、停車場管理：

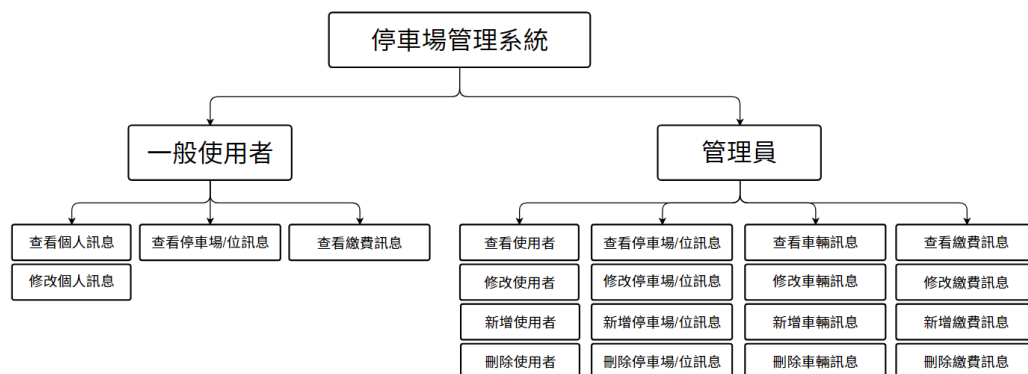
1. 停車場管理員登錄後，進入整體資訊管理頁面，可修改停車場基本資訊。
2. 管理使用者資訊，包括查看和修改使用者狀態。

四、收費記錄管理：

1. 收費員登錄後，進入收費記錄管理頁面，查看所有停車場的收費記錄。
2. 可通過查詢準則過濾符合條件的收費記錄，查看詳細資訊。

五、資料統計與報表：

1. 生成停車場使用報告，包括停車位利用率、收費統計等。
2. 提供管理員查看停車場使用情況的報表，支援運營管理和決策。



圖表 7. 停車場管理系統架構設計圖

4.3 資料庫設計

4.3.1 資料庫結構設計

名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
<u>id</u> 	int			否	無	用戶編號
<u>vehicle_id</u>	varchar(10)	utf8_unicode_ci		否	無	車輛編號
<u>name</u>	varchar(10)	utf8_unicode_ci	是		NULL	姓名
<u>age</u>	tinyint		是		NULL	年齡
<u>birthday</u>	date		是		NULL	生日
<u>email</u>	varchar(50)	utf8_unicode_ci	是		NULL	電子郵件
<u>phone_number</u>	varchar(10)	utf8_unicode_ci	是		NULL	手機號碼
<u>origin_home</u>	varchar(255)	utf8_unicode_ci	是		NULL	戶籍地址
<u>current_home</u>	varchar(255)	utf8_unicode_ci	是		NULL	目前住處
<u>user_type</u>	smallint		否		3	用戶類型(1管理員、2員工、3普通用戶)
<u>user_state</u>	smallint		否		1	帳戶狀態(1可用、2異常、3已凍結、4已登出)
<u>create_time</u>	datetime		否		CURRENT_TIMESTAMP	創建時間
<u>update_time</u>	datetime		否		CURRENT_TIMESTAMP	更新時間
<u>deleted</u>	tinyint(1)		否		0	是否刪除

圖表 8. 用戶資料表

名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
<u>id</u> 	int			否	無	車輛編號
<u>license_plate</u>	varchar(10)	utf8_unicode_ci	否		無	車牌號
<u>user_id</u>	int		否		無	用戶編號
<u>spot_id</u>	int		否		無	停車位編號
<u>vehicle_brand</u>	varchar(50)	utf8_unicode_ci	是		NULL	車輛品牌
<u>vehicle_type</u>	varchar(50)	utf8_unicode_ci	是		NULL	車輛型號
<u>vehicle_color</u>	varchar(50)	utf8_unicode_ci	是		NULL	車輛顏色
<u>vehicle_state</u>	smallint		否		1	當前狀態(1使用中、2維護中、3預訂中)
<u>special_requirements</u>	varchar(255)	utf8_unicode_ci	是		NULL	特殊需求
<u>notes</u>	varchar(255)	utf8_unicode_ci	是		NULL	備註
<u>create_time</u>	datetime		否		CURRENT_TIMESTAMP	創建時間
<u>update_time</u>	datetime		否		CURRENT_TIMESTAMP	更新時間
<u>deleted</u>	tinyint(1)		否		0	是否刪除

圖表 9. 車輛資料表

名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註
<u>id</u> 🗝️	int		否	無		停車位編號
<u>zone</u>	varchar(10)	utf8_unicode_ci	是	NULL		所屬區域
<u>type</u>	int		否	1		停車位類型(1普通、2殘疾人專用、3電動車充電位)
<u>state</u>	int		否	1		狀態(1空閒、2預訂、3使用中)
<u>cost_hour</u>	int		否	50		每小時的費用
<u>electric_charger</u>	tinyint(1)		否	0		是否有電動車充電設施
<u>create_time</u>	datetime		否	CURRENT_TIMESTAMP		創建時間
<u>update_time</u>	datetime		否	CURRENT_TIMESTAMP		最後修改時間
<u>deleted</u>	tinyint(1)		否	0		是否刪除

圖表 10. 停車位資料表圖表

下圖為我們的收費資料表的資料型態與說明

欄位名稱	資料型態	說明
id	int	車輛的 ID
license_plate	int	車牌號碼
in_time	array(date)	進場時間
out_time	array(date)	離場時間
fee	array(int)	停車費用
num	int	目前陣列的索引值
create_time	date	創建時間
update_time	date	更新時間

```

1  _id: ObjectId('6581e77b78859a466506ae5f')
2  id: 1188
3  license_plate: "AA2 ㄟ"
4  ▼ in_time: Array (2)
5      0: "2023-12-10 02:56:59am ㄟ"
6      1: "2023-12-20 02:59:05am ㄟ"
7  ▼ out_time: Array (2)
8      0: "2023-12-20 02:57:13am ㄟ"
9      1: "2023-12-20 02:59:24am ㄟ"
10 ▼ fee: Array (2)
11     0: 1000
12     1: 0
13 num: 1
14 create_time: "2023-12-20 02:56:59am ㄟ"
15 update_time: "2023-12-20 02:59:24am ㄟ"

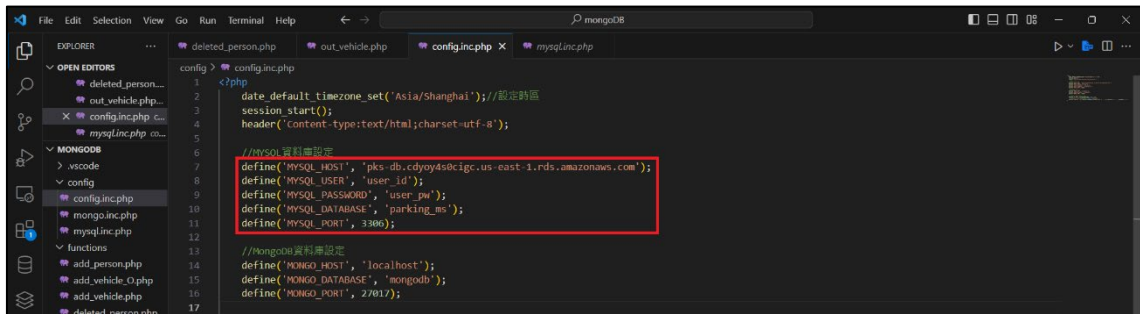
```

圖表 16. 收費資料表 (NoSQL) 圖表

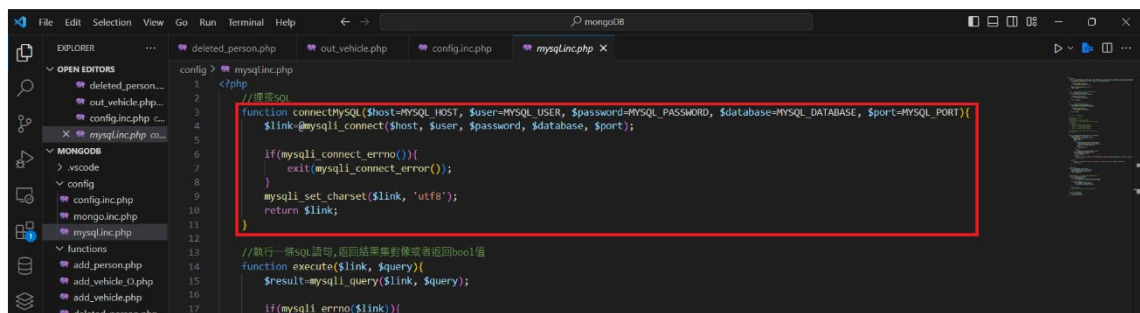
Chapter 5 連結資料庫

5.1 Amazon RDS - MySQL

定義 MYSQL 資料庫設定，並且將 Host 設定為 RDS 端點及埠號設定為 3306(預設)，名為 parking_ms 的資料庫將會以手動方式建立，其餘資料表將會依照程式執行內容自動生成。



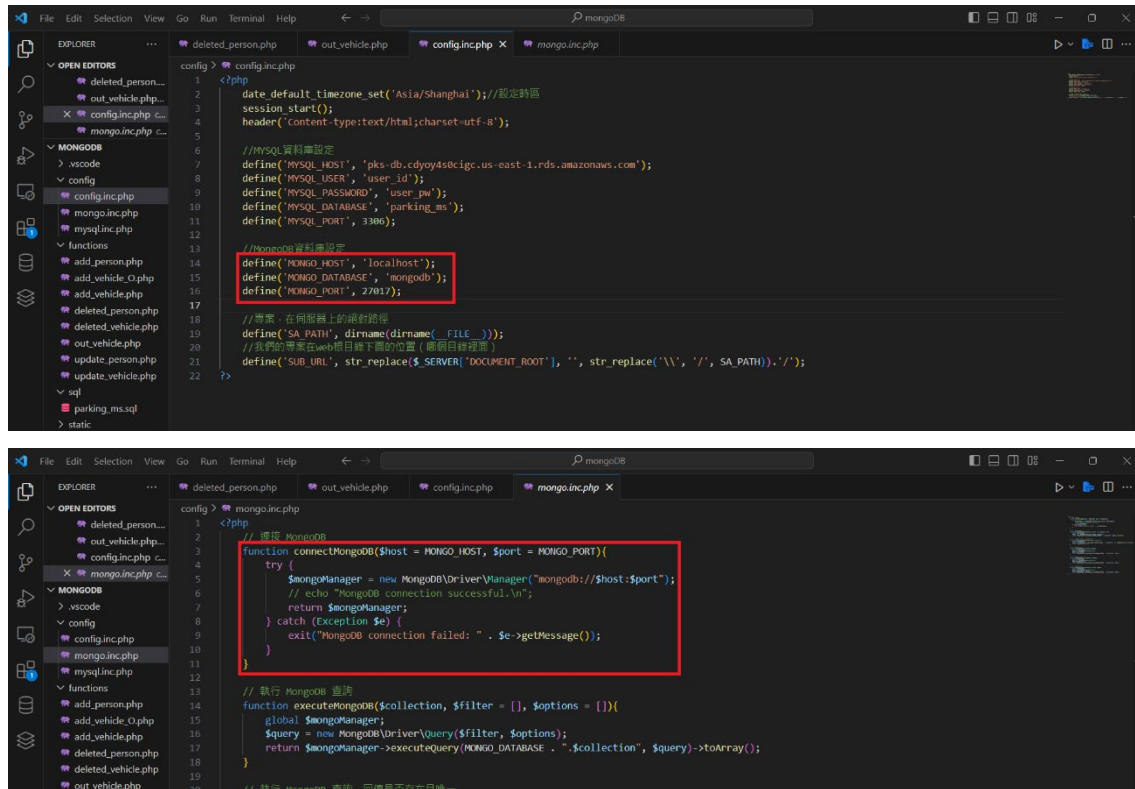
```
1 <?php
2 date_default_timezone_set('Asia/Shanghai');//設定時區
3 session_start();
4 header('Content-type:text/html;charset=utf-8');
5
6 //MySQL資料庫設定
7 define('MYSQL_HOST','pks-db.cdyoy4sdcicg.us-east-1.rds.amazonaws.com');
8 define('MYSQL_USER','user_id');
9 define('MYSQL_PASSWORD','user_pw');
10 define('MYSQL_DATABASE','parking_ms');
11 define('MYSQL_PORT',3306);
12
13 //MongoDB資料庫設定
14 define('MONGO_HOST','localhost');
15 define('MONGO_DATABASE','mongodb');
16 define('MONGO_PORT',27017);
17
```



```
1 //mysql
2
3 function connectMySQL($host=MYSQL_HOST,$user=MYSQL_USER,$password=MYSQL_PASSWORD,$database=MYSQL_DATABASE,$port=MYSQL_PORT){
4     $link=mysqli_connect($host,$user,$password,$database,$port);
5
6     if(mysqli_connect_errno()){
7         exit(mysqli_connect_error());
8     }
9     mysqli_set_charset($link,'utf8');
10    return $link;
11 }
12
13 //執行一條SQL語句，返回結果集對象或者返回bool值
14 function execute($link,$query){
15     $result=mysqli_query($link,$query);
16
17     if(mysqli_erro($link)){
```

5.2 EC2 - MongoDB

定義 MongoDB 資料庫設定，因 MongoDB 資料庫建立於 EC2 主機中，因此將 Host 設定於本機，埠號設定為 27017(預設)，MongoDB 不須建立資料庫，資料庫將會依照程式執行內容自動生成。



The image consists of two screenshots of a VS Code editor window. The top screenshot shows the 'config.inc.php' file with the following code:

```
<?php
1 date default_timezone_set('Asia/Shanghai');//設定時區
2 session_start();
3 header('content-type:text/html;charset=utf-8');
4
5 //MySQL資料庫設定
6 define('MYSQL_HOST', 'pks-db-cdyoy4s8cgc.us-east-1-rds.amazonaws.com');
7 define('MYSQL_USER', 'user_id');
8 define('MYSQL_PASSWORD', 'user_pw');
9 define('MYSQL_DATABASE', 'parking_ms');
10 define('MYSQL_PORT', 3306);
11
12 //MongoDB資料庫設定
13 define('MONGO_HOST', 'localhost');
14 define('MONGO_DATABASE', 'mongodb');
15 define('MONGO_PORT', 27017);
16
17 //需要，在伺服器上的絕對路徑
18 define('SA_PATH', dirname(dirname(__FILE__)));
19 //我們的專案在web根目錄下的位置 (即網頁目錄裏)
20 define('SUB_URL', str_replace($_SERVER['DOCUMENT_ROOT'], '', str_replace('\\', '/', SA_PATH)).'/');
21
22 ?>
```

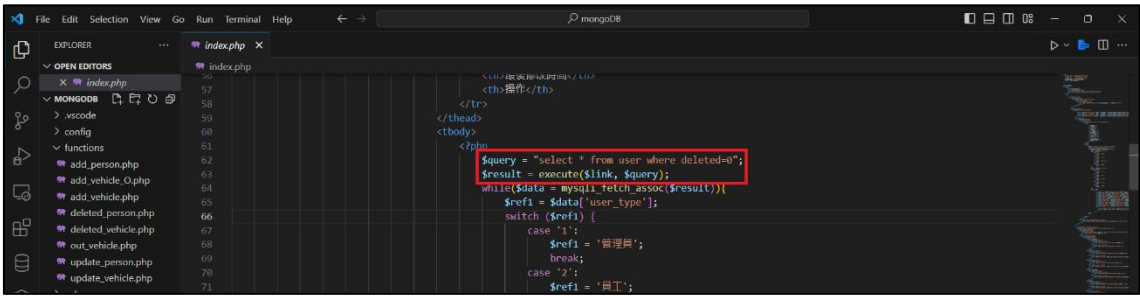
The bottom screenshot shows the 'mongo.inc.php' file with the following code:

```
<?php
1 // 連接 MongoDB
2 function connectMongoDB($host = MONGO_HOST, $port = MONGO_PORT){
3     try {
4         $mongoManager = new MongoClient("mongodb://$host:$port");
5         // echo "MongoDB connection successful.\n";
6         return $mongoManager;
7     } catch (Exception $e) {
8         exit("MongoDB connection failed: " . $e->getMessage());
9     }
10 }
11
12 // 執行 MongoDB 查詢
13 function executeMongoDB($collection, $filter = [], $options = []){
14     global $mongoManager;
15     $query = new MongoClient\Driver\Query($filter, $options);
16     return $mongoManager->executeQuery(MONGO_DATABASE . ".$collection", $query)->toArray();
17 }
18
19 // 返回 query 結果，即陣列形式返回
```

Chapter 6 員工管理

6.1 員工管理紀錄畫面

從資料庫裡找到名為 user 的 Table，並且符合 deleted 為 0。



前端網頁員工管理顯示畫面。

停車場管理系統 - 員工管理

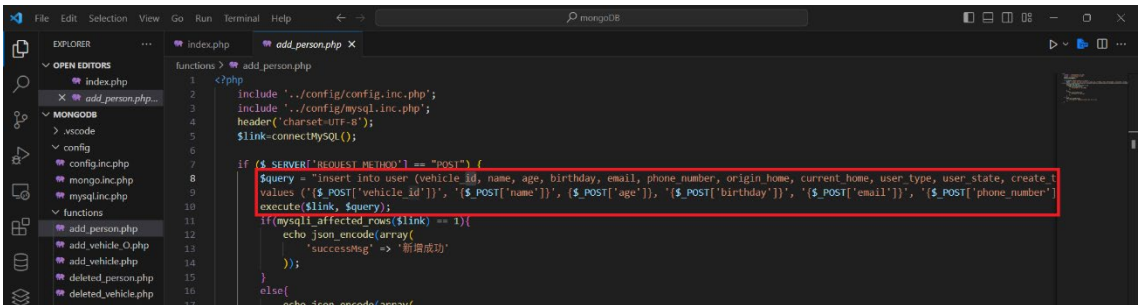
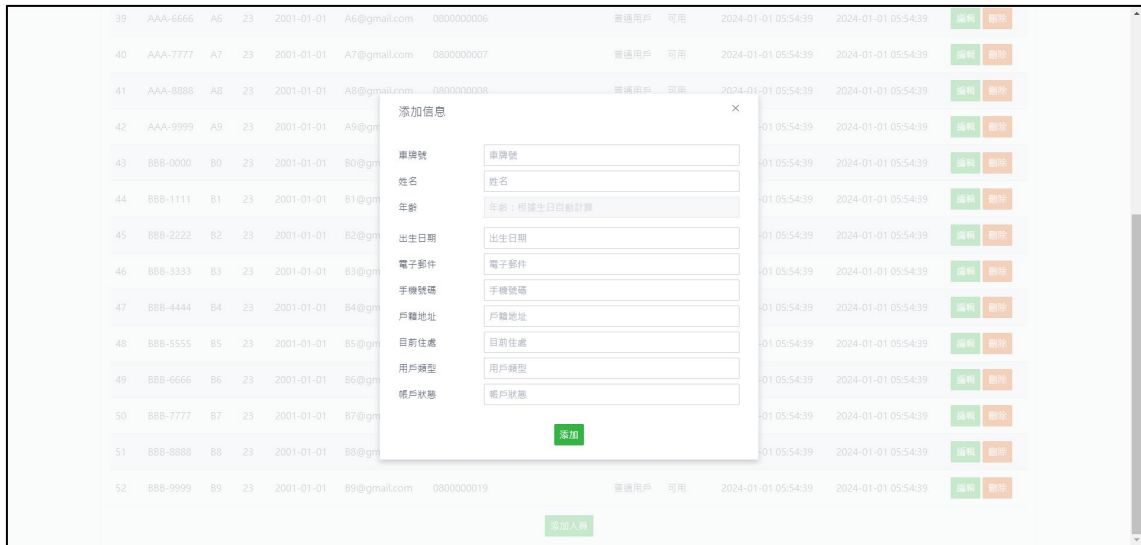
員工管理車輛管理收費管理停車場管理停車位管理

編號	車牌號	姓名	年齡	出生日期	電子郵件	手機號碼	戶籍地址	目前住處	用戶類型	帳戶狀態	註冊時間	最後修改時間	操作
34	AAA-1111	A1	23	2001-01-01	A1@gmail.com	0800000001			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
35	AAA-2222	A2	23	2001-01-01	A2@gmail.com	0800000002			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
36	AAA-3333	A3	23	2001-01-01	A3@gmail.com	0800000003			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
37	AAA-4444	A4	23	2001-01-01	A4@gmail.com	0800000004			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
38	AAA-5555	A5	23	2001-01-01	A5@gmail.com	0800000005			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
33	AAA-0000	A0	23	2001-01-01	A0@gmail.com	0800000000			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
39	AAA-6666	A6	23	2001-01-01	A6@gmail.com	0800000006			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
40	AAA-7777	A7	23	2001-01-01	A7@gmail.com	0800000007			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
41	AAA-8888	A8	23	2001-01-01	A8@gmail.com	0800000008			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
42	AAA-9999	A9	23	2001-01-01	A9@gmail.com	0800000009			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
43	BBB-0000	B0	23	2001-01-01	B0@gmail.com	0800000010			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除
44	BBB-1111	B1	23	2001-01-01	B1@gmail.com	0800000011			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯 刪除

6.2 員工管理新增功能

新增內容包含:

車牌號、姓名、年齡、出生(生日)、年齡(自動生成,依照出生的資料自動計算)、電子郵件、手機號碼、戶籍地址、目前住處、用戶類型(整數型態,1:員工管理、2:員工、3:普通用戶(default))、帳戶狀態(整數型態,1:可用、2:異常、3:已凍結、4:已登出(default))、註冊時間(自動生成)、最後修改時間(自動生成)。



6.3 員工管理刪除功能



前端刪除功能並非真實刪除資料，而是將前端顯示畫面的資料刪除，資料庫中的資料並不會刪除，當前端畫面刪除時，會將資料庫中有一個名為 deleted 欄位標頭更改為 1。


```
mysql.php
functions
add_person.php
add_vehicle.php
deleted_person.php
deleted_vehicle.php
out_vehicle.php
update_person.php
update_vehicle.php

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    @$deletedID = $_POST['id'];
    @$updateInfo = $_POST['updated_info'];

    if (!empty($updateInfo)) {
        $query = "update user set deleted={$updateInfo['deleted']}, update_time=now() where id={$deletedID}";
        // 刪除用戶
        // $query = "delete from user where id={$deletedID}";
        execute($link, $query);
        if(mysqli_affected_rows($link) == 1){
            // 刪除成功
        }
    }
}
```

id	vehicle_id	name	age	birthday	email	phone_number	origin_home	current_home	user_type	user_state	create_time	update_time	deleted
33	AAA-0000	A0	23	2001-01-01	A0@gmail.com	0800000000			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	1
34	AAA-1111	A1	23	2001-01-01	A1@gmail.com	0800000001			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	1
35	AAA-2222	A2	23	2001-01-01	A2@gmail.com	0800000002			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	0
36	AAA-3333	A3	23	2001-01-01	A3@gmail.com	0800000003			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	0
37	AAA-4444	A4	23	2001-01-01	A4@gmail.com	0800000004			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	0
38	AAA-5555	A5	23	2001-01-01	A5@gmail.com	0800000005			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	0
39	AAA-6666	A6	23	2001-01-01	A6@gmail.com	0800000006			3	1	2024-01-01 05:54:39	2024-01-01 05:54:39	0

6.4 員工管理更新功能

更新功能可將新增功能所含括的內容更新，其他在新增功能未含蓋的內容並不能更新，需要指定方式才能更新這些特殊內容。

停車場管理系統 - 員工管理

員工管理

車輛管理

收費管理

停車場管理

停車位管理

編號	車牌號	姓名	年齡	出生日期	電子郵件	手機號碼	戶籍地址	目前住處	用戶類型	帳戶狀態	註冊時間	最後修改時間	操作
35	AAA-2222	A2	23	2001-01-01	A2@gmail.com	0800000002			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	<div>編輯刪除</div>
36	AAA-3333	A3	23	2001-01-01	A3@gmail.com	0800000003			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	<div>編輯刪除</div>
37	AAA-4444	A4	23	2001-01-01	A4@gmail.com	0800000004			普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	<div>編輯刪除</div>

停車場管理系統 - 員工管理

員工管理

車輛管理

收費管理

停車場管理

停車位管理

編號	車牌號	姓名	年齡	出生日期	電子郵件	手機號碼	戶籍地址	目前住處	用戶類型	帳戶狀態	註冊時間	最後修改時間	操作
35	AAA-2222	A2	23	2001-01-01	A2@gmail.com	0800000002			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
36	AAA-3333	A3	23	2001-01-01	A3@gmail.com	0800000003			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
37	AAA-4444	A4	23	2001-01-01	A4@gmail.com	0800000004			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
38	AAA-5555	A5	23	2001-01-01	A5@gmail.com	0800000005			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
39	AAA-6666	A6	23	2001-01-01	A6@gmail.com	0800000006			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
40	AAA-7777	A7	23	2001-01-01	A7@gmail.com	0800000007			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
41	AAA-8888	A8	23	2001-01-01	A8@gmail.com	0800000008			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
42	AAA-9999	A9	23	2001-01-01	A9@gmail.com	0800000009			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
43	BBB-0000	B0	23	2001-01-01	B0@gmail.com	0800000010			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
44	BBB-1111	B1	23	2001-01-01	B1@gmail.com	0800000011			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>
45	BBB-2222	B2	23	2001-01-01	B2@gmail.com	0800000012			普通用戶	可用	2001-05-31 05:54:39	2024-01-01 05:54:39	<div>編輯</div> <div>刪除</div>

編輯資訊

編號

35

車牌號

AAA-2222

姓名

A2

年齡

23

出生日期

2001-01-01

電子郵件

A2@gmail.com

手機號碼

0800000002

戶籍地址

戶籍地址

目前住處

目前住處

用戶類型

普通用戶

帳戶狀態

可用

註冊時間

2024-01-01 05:54:39

最後修改時間

2024-01-01 05:54:39

更新

```
config.inc.php
mongo.inc.php
mysql.inc.php
functions
add_person.php
add_vehicle.php
deleted_person.php
deleted_vehicle.php
out_vehicle.php
update_person.php
update_vehicle.php

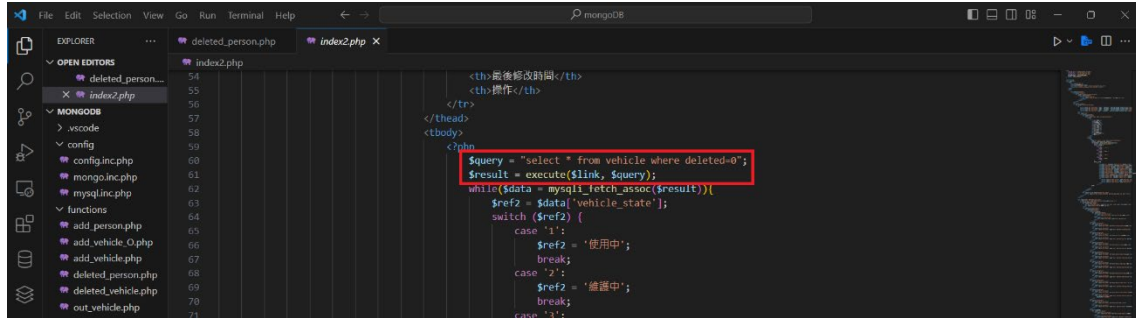
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    @$updateID = $_POST['id'];
    @$updateInfo = $_POST['updated_info'];

    if (!empty($updateInfo)) {
        $query = "update user set vehicle_id={$updateInfo['vehicle_id']}, name='{$updateInfo['name']}',
        age={$updateInfo['age']}, birthday='{$updateInfo['birthday']}', email='{$updateInfo['email']}',
        phone_number='{$updateInfo['phone_number']}', origin_home='{$updateInfo['origin_home']}', current_home='{$updateInfo['current_home']}',
        user_type={$updateInfo['user_type']}, user_state={$updateInfo['user_state']}, update_time=now() where id={$updateID}";
        execute($link, $query);
        if(mysqli_affected_rows($link) == 1){
            // 更新成功
        }
    }
}
```


Chapter 7 車輛管理

7.1 車輛管理紀錄畫面

從資料庫裡找到名為 vehicle 的 Table，並且符合 deleted 為 0。



```
<th>最後修改時間</th>
<th>操作</th>
</thead>
<tbody>
<?php
$query = "select * from vehicle where deleted=0;";
$result = execute($link, $query);
while($data = mysqli_fetch_assoc($result)){
    $ref2 = $data['vehicle.state'];
    switch ($ref2) {
        case '1':
            $ref2 = '使用中';
            break;
        case '2':
            $ref2 = '維護中';
            break;
        case '3':
            $ref2 = '預定中';
            break;
    }
}
```

前端網頁車輛管理顯示畫面。

停車場管理系統 - 車輛管理

員工管理

車輛管理

停車位管理

收費管理

車輛編號	車牌號	用戶編號	停車位編號	車輛品牌	車輛型號	車輛顏色	當前狀態	特殊需求	備註	註冊時間	最後修改時間	操作
21	CCC-0000	53	187	Benz	AMG GT43	white	使用中			2024-01-01 14:20:25	2024-01-01 14:40:25	<div>編輯刪除退場</div>
22	CCC-1111	54	191	Benz	C300	white	使用中			2024-01-01 14:37:50	2024-01-01 14:40:20	<div>編輯刪除退場</div>
23	CCC-2222	55	179	Ferrari	296 GTB	red	使用中			2024-01-01 14:39:58	2024-01-01 14:40:13	<div>編輯刪除退場</div>
24	CCC-3333	56	175	Ferrari	Portofino	red	使用中			2024-01-01 14:41:10	2024-01-01 14:41:30	<div>編輯刪除退場</div>

添加人員

7.2 車輛管理新增功能

新增內容包含:

車牌號、車輛品牌、車輛顏色、當前狀態(整數型態，1:使用中(default)、2:維護中、3:預定中)、特殊需求(字串型態，default:Null)、備註(字串型態，default:Null)、註冊時間、最後修改時間、最後修改時間(自動生成)。



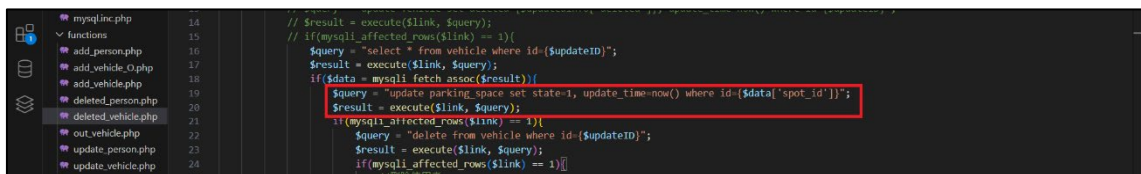
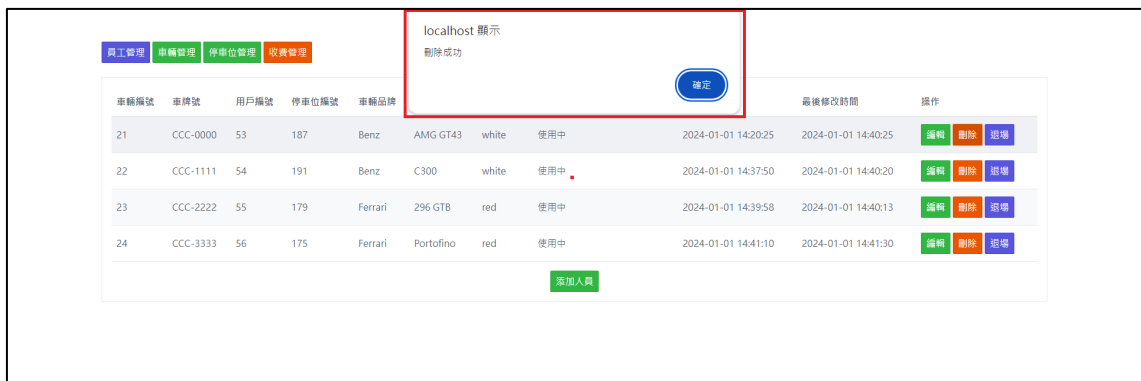
當車輛管理新增資料時，除了會將資料更新在資料庫中，還會同時將資料傳送到員工管理及停車位管理的資料表中，且更新員工管理及停車位管理。

50	BBB-7777	B7	23	2001-01-01	B7@gmail.com	0800000017	普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯	刪除
51	BBB-8888	B8	23	2001-01-01	B8@gmail.com	0800000018	普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯	刪除
52	BBB-9999	B9	23	2001-01-01	B9@gmail.com	0800000019	普通用戶	可用	2024-01-01 05:54:39	2024-01-01 05:54:39	編輯	刪除
53	CCC-0000						普通用戶	可用	2024-01-01 14:20:25	2024-01-01 14:20:25	編輯	刪除
54	CCC-1111						普通用戶	可用	2024-01-01 14:37:50	2024-01-01 14:37:50	編輯	刪除
55	CCC-2222						普通用戶	可用	2024-01-01 14:39:58	2024-01-01 14:39:58	編輯	刪除
56	CCC-3333						普通用戶	可用	2024-01-01 14:41:10	2024-01-01 14:41:10	編輯	刪除

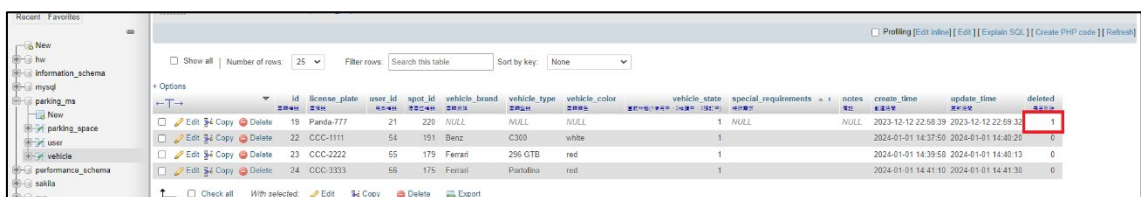
200	Zone1	電動車充電位	空閒	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	編輯	刪除
201	Zone3	電動車充電位	空閒	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	編輯	刪除
186	Zone1	殘疾人專用	空閒	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	編輯	刪除
187	Zone2	普通	使用中	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	編輯	刪除
188	Zone2	電動車充電位	空閒	50	是	2023-12-12 01:59:16	2023-12-12 22:40:14	編輯	刪除
189	Zone5	普通	空閒	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	編輯	刪除

7.3 車輛管理刪除功能

21	CCC-0000	53	187	Benz	AMG GT43	white	使用中	2024-01-01 14:20:25	2024-01-01 14:40:25	編輯	刪除	退還
22	CCC-1111	54	191	Benz	C300	white	使用中	2024-01-01 14:37:50	2024-01-01 14:40:20	編輯	刪除	退還
23	CCC-2222	55	179	Ferrari	296 GTB	red	使用中	2024-01-01 14:39:58	2024-01-01 14:40:13	編輯	刪除	退還
24	CCC-3333	56	175	Ferrari	Portofino	red	使用中	2024-01-01 14:41:10	2024-01-01 14:41:30	編輯	刪除	退還



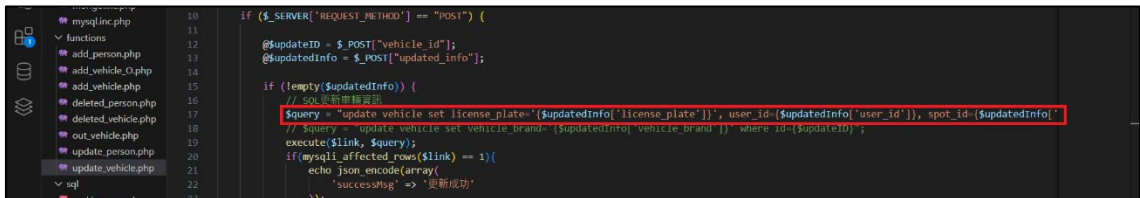
前端刪除功能並非真實刪除資料，而是將前端顯示畫面的資料刪除，資料庫中的資料並不會刪除，當前端畫面刪除時，會將資料庫中有一個名為 deleted 的欄位標頭更改為 1。



7.4 車輛管理更新功能

更新功能可將新增功能所舍括的內容更新，其他在新增功能未含蓋的內容並不能更新，需要指定方式才能更新這些特殊內容。





7.5 車輛管理退場功能

退場功能會依照時間將車輛做計費統計，並且將資料傳至 MongoDB。



因時間未滿半小時，因此繳費金額為 0，當超過半小時以 50 元開始計費。

停車場管理系統 - DDD-1111

員工管理

車輛管理

停車位管理

收費管理

車牌	進場時間	出場時間	繳費金額	建立時間	最後修改時間	操作
DDD-1111	2024-01-01 09:01:43pm	2024-01-01 09:05:08pm	0	2024-01-01 09:01:43pm	2024-01-01 09:05:08pm	<div>編輯刪除</div>

local

mongodb

foo

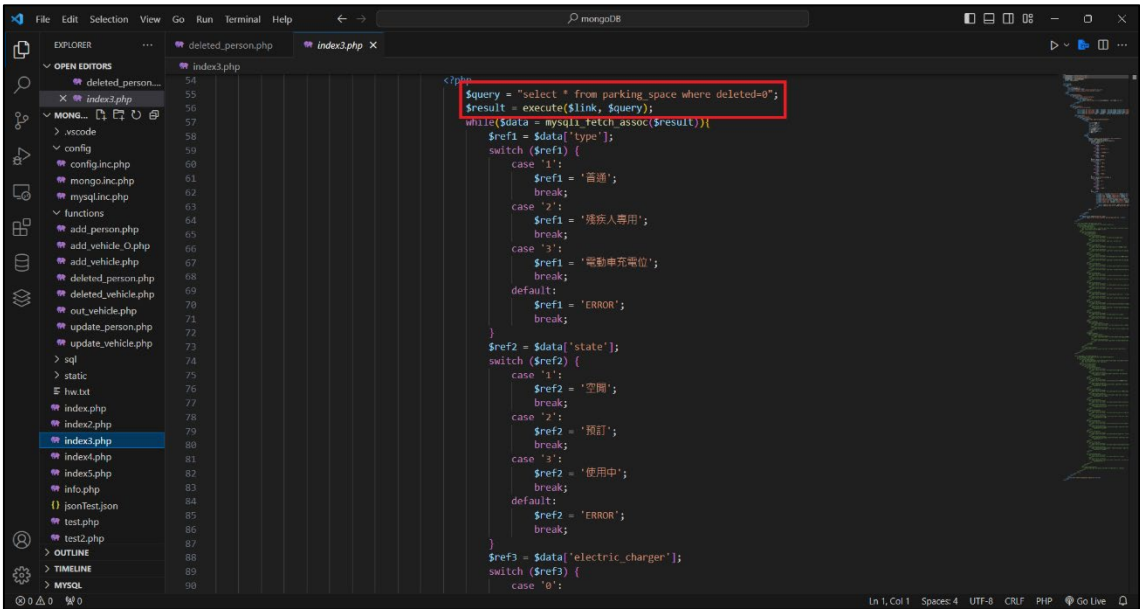
```
_id: ObjectId('6592b7a2e8c145085607aa92')
id: 1188
license_plate: "DDD-0000"
in_time: Array (1)
out_time: Array (empty)
fee: Array (empty)
num: 0
create_time: "2024-01-01 09:01:22pm"
update_time: "2024-01-01 09:01:22pm"

_id: ObjectId('6592b7b78c3b3a5afc9e0f83')
id: 1188
license_plate: "DDD-1111"
in_time: Array (1)
  0: "2024-01-01 09:01:43pm"
out_time: Array (1)
  0: "2024-01-01 09:05:08pm"
fee: Array (1)
  0: 0
num: 0
create_time: "2024-01-01 09:01:43pm"
update_time: "2024-01-01 09:05:08pm"
```

Chapter 8 停車位管理

8.1 停車位管理顯示畫面

從資料庫裡找到名為 parking_space 的 Table，並且符合 deleted 為 0



前端網頁停車位管理顯示畫面

停車場管理系統 - 停車位管理

員工管理

車輛管理

停車位管理

收費管理

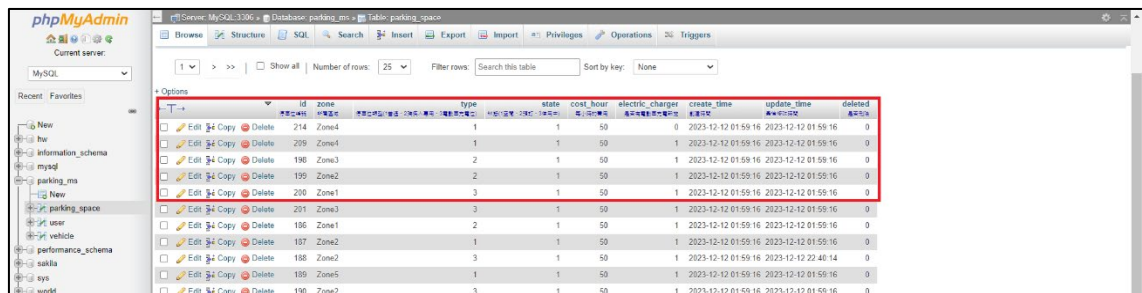
停車位編號	所屬區域	停車位類型	使用狀態	每小時的費用	電動車充電設施	創建時間	最後修改時間	操作
214	Zone4	普通	空間	50	否	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
209	Zone4	普通	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
198	Zone3	殘疾人專用	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
199	Zone2	殘疾人專用	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
200	Zone1	電動車充電位	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
201	Zone3	電動車充電位	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
186	Zone1	殘疾人專用	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
187	Zone2	普通	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
188	Zone2	電動車充電位	空間	50	是	2023-12-12 01:59:16	2023-12-12 22:40:14	<div>編輯刪除</div>
189	Zone5	普通	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
190	Zone2	電動車充電位	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>
191	Zone1	普通	空間	50	是	2023-12-12 01:59:16	2023-12-12 01:59:16	<div>編輯刪除</div>

8.2 停車位管理編輯功能

新增內容包含:

所屬區域、停車位類型(整數型態，1:普通、2:殘疾人專用、3:電動車充電位)、使用狀態(整數型態，1:空閒、2:預定、3:使用中)、每小時的費用、電動車充電設施(整數型態，0:無、1:有)、創建時間(自動生成)、最後修改時間(自動生成)。

因停車位目前為固定資料，因此在前端不開放新增操作，若需編輯操作，須至資料庫進行操作處。

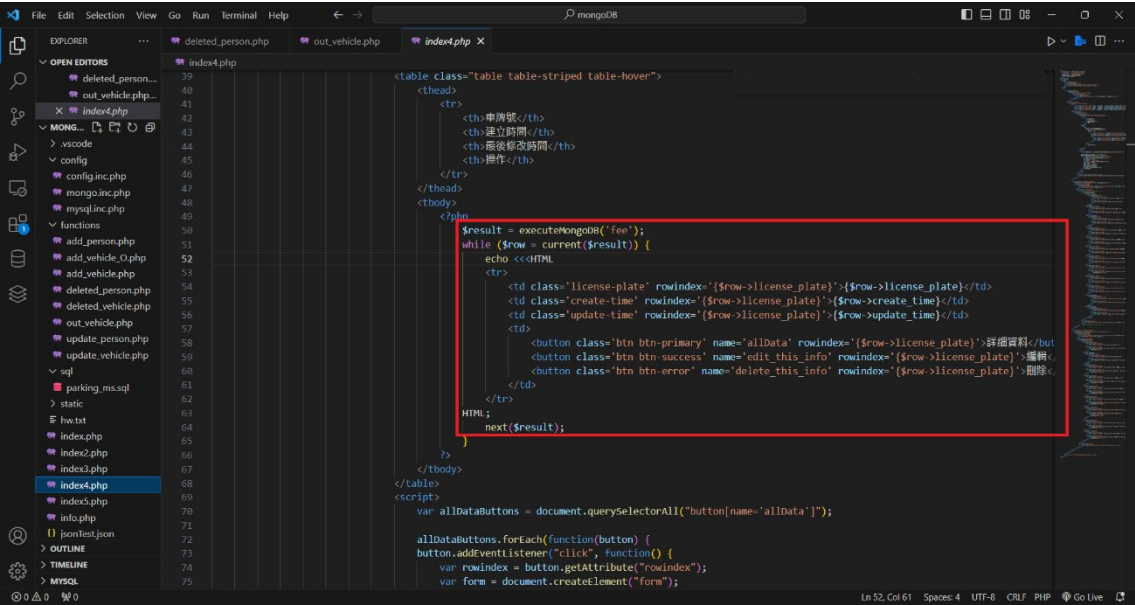


	id	zone	type	state	cost	hour	electric_charger	create_time	update_time	deleted
<input type="checkbox"/>	214	Zone4	1	1	50	0	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	209	Zone4	1	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	196	Zone3	2	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	195	Zone2	2	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	200	Zone1	3	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	201	Zone3	3	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	196	Zone1	2	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	137	Zone2	1	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	158	Zone2	3	1	50	1	0	2023-12-12 01:59:16	2023-12-12 22:40:14	0
<input type="checkbox"/>	109	Zone5	1	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0
<input type="checkbox"/>	190	Zone2	3	1	50	1	0	2023-12-12 01:59:16	2023-12-12 01:59:16	0

Chapter 9 收費管理

9.1 收費管理顯示畫面

從資料庫裡找到名為 fee 的 Collection。



```
File Edit Selection View Go Run Terminal Help
mongoDB
EXPLORER
  OPEN EDITORS
    deleted_person.php
    out_vehicle.php
    index4.php
  MONGO...
  vscode
  config
  config.inc.php
  mongo.inc.php
  mysql.inc.php
  functions
  add_person.php
  add_vehicle_O.php
  add_vehicle.php
  deleted_person.php
  deleted_vehicle.php
  out_vehicle.php
  update_person.php
  update_vehicle.php
  sql
  parking_ms.sql
  static
  hw.txt
  index.php
  index2.php
  index4.php
  index5.php
  info.php
  jsonTest.json
  OUTLINE
  TIMELINE
  MYSQL
  0 0 0 0
index4.php
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
<table class="table table-striped table-hover">
  <thead>
    <tr>
      <th>車牌號</th>
      <th>建立時間</th>
      <th>最後修改時間</th>
      <th>操作</th>
    </tr>
  </thead>
  <tbody>
    <?php
      $result = executeMongoDB('fee');
      while ($row = current($result)) {
        echo <<HTML
        <tr>
          <td class='license-plate' rowindex='{$row->license_plate}'>{$row->license_plate}</td>
          <td class='create-time' rowindex='{$row->license_plate}'>{$row->create_time}</td>
          <td class='update-time' rowindex='{$row->license_plate}'>{$row->update_time}</td>
          <td>
            <button class='btn btn-primary' name='allData' rowindex='{$row->license_plate}'>詳細資料</button>
            <button class='btn btn-success' name='edit_this_info' rowindex='{$row->license_plate}'>編輯</button>
            <button class='btn btn-error' name='delete_this_info' rowindex='{$row->license_plate}'>刪除</button>
          </td>
        </tr>
      }
    </tbody>
  </table>
  <script>
    var allDataButtons = document.querySelectorAll("button[name='allData']");
    allDataButtons.forEach(function(button) {
      button.addEventListener("click", function() {
        var rowindex = button.getAttribute("rowindex");
        var form = document.createElement("form");
      });
    });
  </script>
```

前端網頁收費管理顯示畫面。

停車場管理系統 - 收費管理				
<div>員工管理 車輛管理 停車位管理 收費管理</div>				
車牌號	建立時間	最後修改時間	操作	
DDD-0000	2024-01-01 09:01:22pm	2024-01-01 09:01:22pm	詳細資料	編輯刪除
DDD-1111	2024-01-01 09:01:43pm	2024-01-01 09:05:08pm	詳細資料	編輯刪除
DDD-2222	2024-01-01 09:02:08pm	2024-01-01 09:02:08pm	詳細資料	編輯刪除
DDD-3333	2024-01-01 09:03:38pm	2024-01-01 09:03:38pm	詳細資料	編輯刪除
添加人員				

9.2 收費管理詳細資料功能

詳細資料功能可檢視車輛以否出場，且計算繳費金額。

停車場管理系統 - AA2						
車牌	進場時間	出場時間	繳費金額	建立時間	最後修改時間	操作
AA2	2023-12-10 02:56:59am	2023-12-20 02:57:13am	1000	2023-12-20 02:56:59am	2024-01-01 11:41:59pm	編輯 刪除
AA2	2023-12-20 02:59:05am	2023-12-20 02:59:24am	0	2023-12-20 02:56:59am	2024-01-01 11:41:59pm	編輯 刪除
AA2	2024-01-01 11:41:59pm	尚未出場	尚未收費	2023-12-20 02:56:59am	2024-01-01 11:41:59pm	編輯 刪除

若已出場，則會顯示出場時間及繳費金額，在未出場時，顯示金額會顯示未收費。

停車場管理系統 - AA2						
車牌	進場時間	出場時間	繳費金額	建立時間	最後修改時間	操作
AA2	2023-12-10 02:56:59am	2023-12-20 02:57:13am	1000	2023-12-20 02:56:59am	2024-01-01 11:41:59pm	編輯 刪除
AA2	2023-12-20 02:59:05am	2023-12-20 02:59:24am	0	2023-12-20 02:56:59am	2024-01-01 11:41:59pm	編輯 刪除
AA2	2024-01-01 11:41:59pm	尚未出場	尚未收費	2023-12-20 02:56:59am	2024-01-01 11:41:59pm	編輯 刪除

```
> .vscode
  config
  config.inc.php
  mongo.inc.php
  mysql.inc.php
  functions
  add_person.php
  add_vehicle_O.php
  add_vehicle.php
  deleted_person.php
  deleted_vehicle.php
  out_vehicle.php
  update_person.php
  update_vehicle.php
  sql
  parking_ms.sql
  static
  hwa.txt
  index.php
  index2.php
  index3.php
  index4.php
  index5.php
  info.php

44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

<th>最後修改時間:</th>
</thead>
<tbody>
<tr>
<td>
<td class="license-plate" rowindex="{ $row->license_plate }">{$row->license_plate}</td>
<td class="create-time" rowindex="{ $row->license_plate }">{$row->create_time}</td>
<td class="update-time" rowindex="{ $row->license_plate }">{$row->update_time}</td>
<td>
<button class="btn btn-primary" name="allData" rowindex="{ $row->license_plate }">詳細資料</button>
<button class="btn btn-success" name="edit_this_info" rowindex="{ $row->license_plate }">編輯</button>
<button class="btn btn-error" name="delete_this_info" rowindex="{ $row->license_plate }">刪除</button>
</td>
</tr>
</tbody>
HTML;
next($result);
}
}
</table>
</script>
</script>
var allDataButtons = document.querySelectorAll("button[name='allData']");
```

金額計算方式：天數*100；若低於 1 天則低消 100 元

```
if(count($result[0]->out_time) === count($result[0]->in_time) - 1){
    $num = $result[0]->num;
    // 計算相差的秒數
    $diffInSeconds = strtotime($d) - strtotime($result[0]->in_time[$num]);
    // 將秒數轉換為天數
    $daysDifference = floor($diffInSeconds / (60 * 60 * 24));

    $updateFilter = $filter;
    $totalFee = $daysDifference * 100;
    if($totalFee < 100){
        $totalFee = 100;
    }
    $updateData = [
        '$set' => [
            'update_time' => $d
        ],
        '$push' => [['out_time' => $d, 'fee' => $totalFee]]
    ];
    $updateResult = updateDocument('fee', $updateFilter, $updateData);
    echo json_encode(array(
        'successMsg' => "以新增退場時間與費用"
    ));
}
```