

專題研究報告

- 自動旅遊規劃系統開發

畢業專題-自動旅遊規劃系統開發

專題摘要

出去玩的時候總是快樂，但是出去玩的事前準備也很重要，然而並非所有人都願意花時間在旅遊的事前規劃上，又或是已經有心儀的景點卻苦惱不知道該如何安排剩下的行程，這些都是要出去玩之前常遇見的情況，有鑒於此我們希望針對上述情況提供一套快捷的旅遊規劃解決方案。

為了貼近實際情況與利於成果展示，本專題之成果已實際建置，可由附件連結前往使用，程式原始碼附於第 9 章中。

關鍵字：旅遊規劃

專題成員

指導老師：陳同孝 教授

黃柏鈞

擔任組長職務，負責題目構想與架構設計、旅遊規劃邏輯設計與實現、前端設計及撰寫、API 開發

邱立晟

負責程式測試、協助旅遊景點資訊蒐集、協助前端畫面設計

吳靚珣

蒐集旅遊景點資訊、文案撰寫、協助程式測試

楊哲維

旅遊資訊蒐集平台開發、建立旅遊資料庫、協助 API 開發

目錄

1、動機與目的	3
2、開法與系統架構	3
2.1 開發架構	
2.2 系統架構	
3、主要應用技術與工具介紹	4
4、旅遊資訊蒐集平台	5
4.1 新增景點	
4.2 景點資訊提供	
4.3 API 功能	
5、旅遊規劃系統	6
5.1 系統模組	
5.2 旅遊規劃流程	
5.3 規劃方法概述	
5.4 規劃模式說明	
5.5 重點 API 介紹	
6、旅遊規劃系統優化	9
6.1 畫面優化	
6.2 效能優化	
6.3 體驗優化	
7、旅遊規劃與情境展示	11
7.1 情境 1 - 不熟悉該地區，但是想前往旅遊	
7.2 情境 2 - 只想去某幾個點，不想做其他旅遊規劃	
7.3 旅遊規劃結果	
8、結論與心得	13
9、專題程式碼	13
10、附件與連結	14
11、參考資料	14

1、動機與目的

當我們想要前往某一地區旅遊時總是需要做大量的事前功課，其中不外乎查閱大量部落客文章來尋找心儀的旅遊地點或是依靠別人推薦來決定要前往哪個地點，最後再考量交通、時間、預算等因素通過篩選整理後排入行程表。

然而並不是每個人都願意花時間在事前規劃上，或許你可能只想去某幾個地點，其他是如何規劃的並不會太在意；又或是你對於這個地區一無所知想來一場說走就走的旅遊，因此我們想透過開發一套系統來解決這類問題，並且可以透過其他使用者輸入的資訊來讓這套系統持續增長。

綜合上述本專題期望達成以下目標：

1. 提供自動旅遊規劃解決方案
2. 簡易操作與使用之使用者介面
3. 可在多平台使用
4. 建立旅遊資料庫藉使用者提供資料持續強化系統內容

2、系統架構

依據目的要求，本專題之系統架構如圖 2-1 所示，可以看見系統由兩個網站組成(紅框與藍框)，其中紅框部分(下稱旅遊資訊蒐集平台)由 Laravel 框架撰寫，負責旅遊資訊蒐集平台與 API 的部分；藍框部分(下稱旅遊規劃系統)主要使用 ReactJS 呈現規劃系統使用者介面與實現旅遊規劃邏輯，並搭配 Lumen 框架處理單頁應用程式、前端路由在部屬上的一些問題。

旅遊資訊蒐集平台將打造成類似景點維基百科，但其最重要的功能則為讓蒐集的景點資訊能夠應用於旅遊規劃系統上，在旅遊規劃系統上將會參考旅遊資訊蒐集平台資料庫之資料進行旅遊規劃，並且提供景點介紹。

而旅遊資訊蒐集平台在前端提供 RWD(Responsive web design)網頁方便手機瀏覽，旅遊規劃系統前端介面為了更加優化使用者體驗則進一步以 PWA(Progressive Web App)呈現。並且兩個網站都可以在常見的連網終端設備上執行。

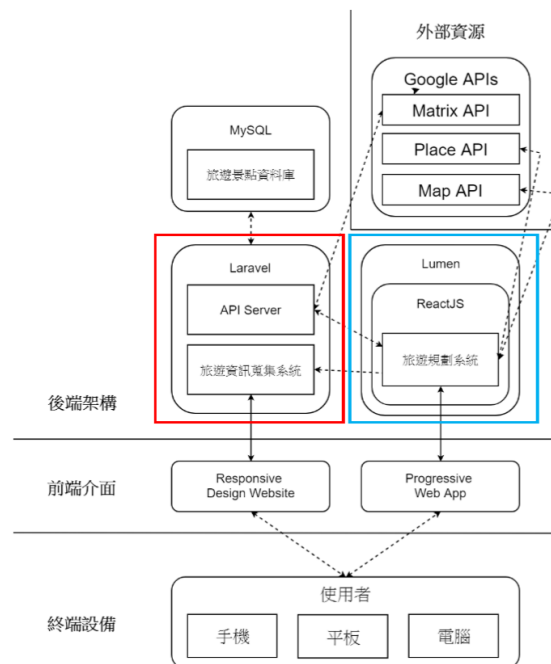


圖 2-1 系統架構

3、主要應用技術與工具介紹

本專題使用了多項技術整合而成，以下將簡單介紹專題中使用到的工具、框架與技術。

1. ReactJS

JavaScript 前端框架。作為實現前端的基石考量到跨平台性、高使用者互動要求、動態畫面、複雜的旅遊規劃邏輯均需要此實現，為求最佳使用者體驗遂決定採用生態系成熟且自己熟悉 ReactJS

2. Laravel

作為旅遊資訊蒐集系統與 API 開發的載具，為了能讓團隊可以擁有統一的開發規範並且節省開發時間、確保擴充性、可維護性、降低開發風險等因素決定使用大型 PHP 框架來開發。

3. Lumen

使用 ReactJS 製作的 SPA 仍然需要一個 Server 來處理一些簡單 Http 請求(例如 404 fallback)，因此使用小型 PHP 框架來當作後端服務器。

4. Progressive Web App

漸進式網頁應用程式指的是應用多種技術打造的網頁並且具有**可信任連線、原生應用體驗、離線可用**等特性，當網站滿足 PWA 條件之後瀏覽器將允許網站安裝至用戶的 Home Screen

5. Service Worker

在支持 Service Worker 技術的瀏覽器下，可以實現對網站的進階快取應用與策略;是實現 PWA 的重要關鍵技術。

6. Bootstrap

導入 CSS 框架減少美工及系統美化上的負擔，並且搭配 Bootstrap 提供的格線系統實現 RWD 排版，製作出能夠適應不同解析度的 UI

7. Git

導入版控系統成功讓團隊分工開發變得輕鬆與優雅，並且能夠適時追蹤專案進度或是追蹤問題

8. Google APIs

本專題使用了部分 Google APIs 來補助開發與確保效果呈現;Google map API 用以呈現路線規畫結果與地點顯示，Google matrix API 用來計算 A 點至 B 點之交通時間與距離，Google place API 則用來減少負責旅遊中”食”的資訊部分。

4、旅遊資訊蒐集平台

旅遊資訊蒐集平台為圖 2-1 紅框處，該平台除了負責收集景點資訊供旅遊規劃系統使用外也提供景點查詢、旅遊文章分享等功能。

詳細資料

地點 台灣台中市西屯區台灣大道
四段路思義教堂

分類 景

評價 尚未有資訊

季節 春夏秋冬

價錢 0-0

適合 中晚

預估
時間

120

店家

連結



是一座位於台灣台中市西屯區東海大學的基督新教禮拜堂，為著名台灣建築師陳其寬與美國華裔建築師貝聿銘之作。

圖 4-1 旅遊景點詳細資訊

4.1 新增景點

系統允許自行增加景點資訊，在此被新增的景點將會被規劃系統參考，並且因為網友持續的貢獻使得平台內容愈趨豐富、完整。

4.2 景點資訊提供

被新增之景點也允許用來查看景點資訊。以臺中路思義教堂為例(圖 4-1)，將可以看見地點詳細資料，包含了該地點評價、適合季節、適合時段、花費資訊、預估停留時間等資訊。

4.3 API 功能

此平台還有另一項重要功能便是提供旅遊規劃系統所需之 API，目前計有 13 條 API，雖目前多為專為旅遊規劃系統所開設之專用 API，但未來不排除開放可能。

5、旅遊規劃系統

旅遊規劃系統為圖 2-1 藍框部分，此章將介紹系統架構、特色與旅遊規劃方法。

5.1 系統模組

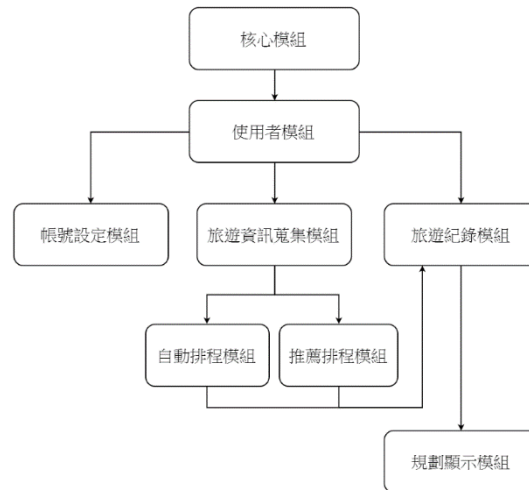


圖 5-1 系統模組架構圖

旅遊規劃系統在開發時即採用模組化開發方式，並且架構如圖 5-1 所示，得益於模組架構讓程式變得非常容易擴充與更改。其中核心模組負責了整個網頁應用的狀態(如:讀取狀態)。

5.2 旅遊規劃流程

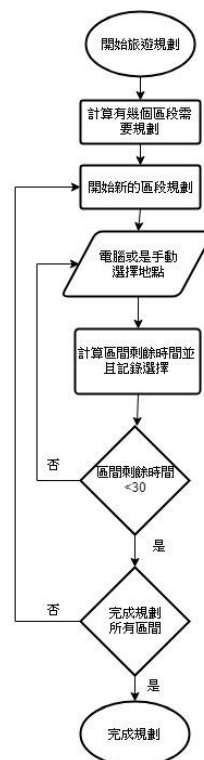


圖 5-2 旅遊邏輯規劃流程圖

旅遊規劃邏輯由一系列步驟組成，並且在滿足條件後完成旅遊規劃。旅遊規劃流程圖如下(圖 5-2)所示：

5.3 規劃方法概述

規劃方法分三階段，第一階段詢問必要資訊(圖 5-3)，第二階段選擇規劃模式進行規劃，第三階段顯示規劃結果。無論選擇何種模式都會將一天分為早上、下午、晚上，三個區段進行規劃，每一區段在進行地點選擇的時候都會考量地點與地點之交通時間與停留時間，當該區段時間用畢則切換下一區段或是下一天直到完成規劃。

5.4 規劃模式說明

旅遊規劃提供三種模式，應對不同使用情境，以下將敘述之：

1. 全自動模式

由系統完成全部的旅遊規劃；此模式將考量出發資訊與結束資訊，由出發點開始以目前規劃地點進行附近景點搜尋並且依權重考量候選地點之交通距離、景點評分等資訊選出最適地點，並且重複直到完成規劃。

圖 5-3 必要資訊詢問表單

圖 5-4 執行自動規劃行程

2. 指定地點模式

可指定想去的地點後由系統完成剩餘的旅遊規劃(此模式為全自動模式之變形)；此模式在開始的時候會先以目前規劃地點與最接近(時段)指定地點兩點為參考，建立一個搜尋範圍嘗試填補兩點間空白時段，並且依照交通距離、景點評分等資訊選出最適地點，如果指定地點均已完成完畢則依照全自動模式邏輯完成剩餘規劃。

圖 5-5 在自動規劃中指定地點

3. 地點推薦模式

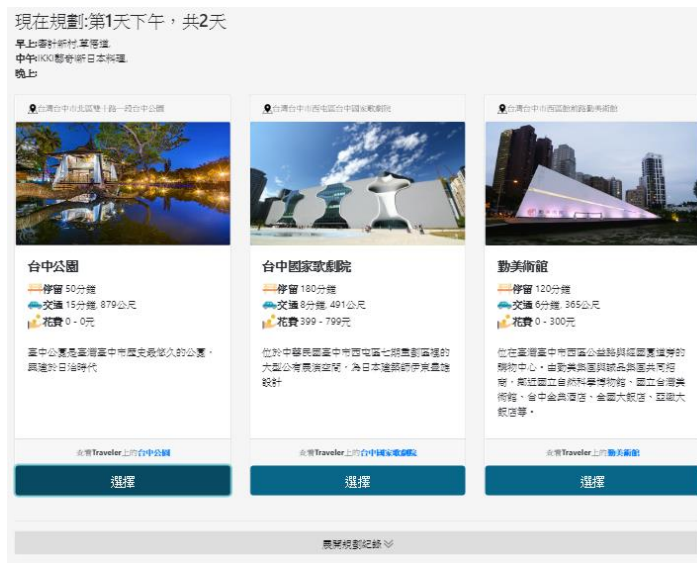


圖 5-6 推薦模式-電腦



圖 5-7 推薦模式-手機

由系統以推薦，使用者自行選擇的方式來完成旅遊規劃；地點推薦模式由系統依據資訊一次推薦三個地點給予使用者進行選擇，藉由一次一次的選擇填補區段，完成旅遊規劃。

5.5 重點 API 介紹

本節將介紹為實現上述規劃邏輯所實作之重點 API

1. 取得附近景點 Post: /get-nearby-place

以台中火車站周邊 3 公里進行搜尋示範，排除指定景點並且回傳三個候選選項(圖 4-8、圖 4-9)

POST https://traveler.thecodingday.com/api/get-nearby-place			
Authorization Headers Body Pre-request Script Tests			
<input checked="" type="radio"/> form-data <input type="radio"/> x-www-form-urlencoded <input type="radio"/> raw <input type="radio"/> binary			
	Key	Value	Description
<input checked="" type="checkbox"/>	token	\$2y\$10\$UGCxuGR8...	api token
<input checked="" type="checkbox"/>	lon	120.686789	台中火車站經度
<input checked="" type="checkbox"/>	lat	24.137326	台中火車站緯度
<input checked="" type="checkbox"/>	range	3000	優先搜尋3000公尺內景點
<input checked="" type="checkbox"/>	with_out_place_ids[1]	17	要排除之景點id
<input checked="" type="checkbox"/>	with_out_place_ids[2]	32	

圖 5-8 API 參數畫面

```

1 {
2   {
3     "id": 9,
4     "title": "自然科學博物館",
5     "address": "台灣台中市北區館前路國立自然科學博物館",
6     "longitude": "120.6660606",
7     "latitude": "24.1572335",
8     "recommendation": "本館區區由科學中心、太空劇場、生命
9     "rating": "0",
10    "price_start": 100,
11    "price_end": 200,
12    "stay_time": 120,
13    "place_img_url": "/image/15332816865d1XI-3-1.jpg",
14    "traffic_info_relative_last_select": {
15      "distance": {
16        "text": "4.4 公里",
17        "value": 4408
18      },
19      "duration": {
20        "text": "18 分",
21        "value": 1078
22      },
23      "status": "OK"
24    }
25  },
26  {
27    "id": 14,
28    "title": "台中公園",
29    "address": "台灣台中市北區雙十路一段台中公園",
30    "longitude": "120.6844612",
31    "latitude": "24.1448340",
32    "recommendation": "臺中公園是臺灣臺中市歷史最悠久的公
33    "rating": "0",
34    "price_start": 0,
35    "price_end": 0,
36    "stay_time": 50,
37    "place_img_url": "/image/1533441230VnHtp-3-1.jpg",
  
```

圖 5-9 API 返回資訊部分截圖

由於高彈性之設計上述規劃模式幾乎可以藉由此 API 完成旅遊規劃。以推薦模式來說直接將回傳結果顯示在網頁上供使用者候選即可，然後再藉由使用者選取之地點經緯度與地點 ID 進行下一次 API 請求；以指定地點模式舉例則取目前規畫地點與最接近指定地點經緯度連線之中點進行 API 請求。

6、旅遊規劃系統優化

考量到使用者習慣、環境與設備差異性，開發旅遊規劃系統時針對以下項目進行優化調整

6.1 畫面優化

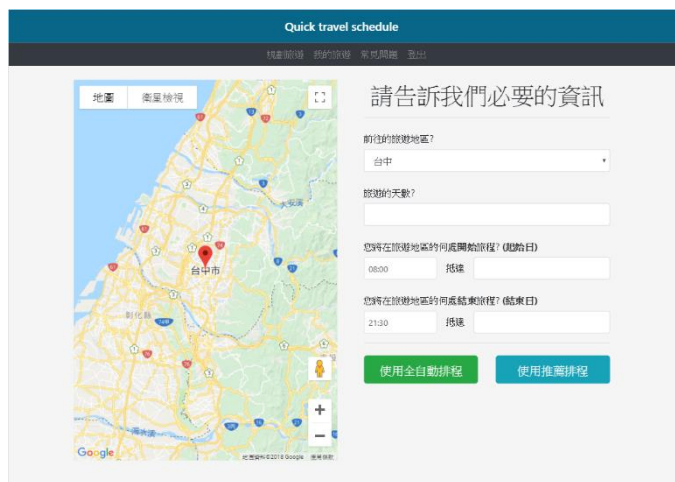


圖 6-1 RWD 呈現電腦版畫面



圖 6-2 RWD 呈現手機版畫面

藉由 RWD 技術，在不同解析度下使用者將會看見不同的畫面排版，甚至隱藏一些較不重要之資訊，增進使用體驗(圖 6-1、圖 6-2)。

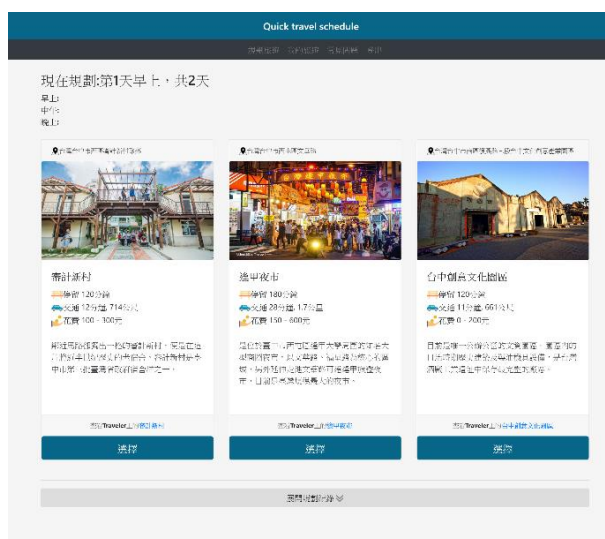


圖 6-3 正常電腦版畫面



圖 6-4 針對手機重新設計

然而並非所有情框均可以使用 RWD 技術解決，為了追求最佳使用者體驗，在某些畫面必須特別針對電腦版(圖 6-3)與手機版(圖 6-4)重新設計不同畫面。

6.2 效能優化

網頁應用的資源必須先從網路加載，在網路較差的時候會有非常糟糕的體驗，就算網路良好多少還是會受到一些影響，因此若能針對重要資源進行快取(如:應用程式本身及靜態資源)，日後使用時將有著媲美 Native App 的速度。

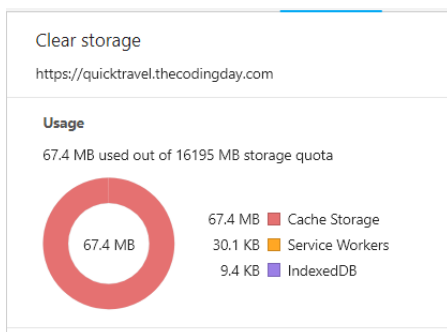


圖 6-5 Service Worker 快取使用量

Path	Content-Type	Content-Size	Time Cached
3eaa63453bf0e8960786c5610ab488c7.jpg	image/jpeg	394,055	2018/8/30 下午12:12:35
67bdb8fc56272f71034f4aacbb6ac98d.png	image/png	2,016	2018/8/30 下午12:12:48
6997a2d2914903cb8739ba0f81846e8e.png	image/png	1,463	2018/8/30 下午12:12:48
89a6c1597e241b49103e894f7d2ca4a.jpg	image/jpeg	263,961	2018/8/30 下午12:12:35
977e4f9b745cab186c9186b2d85dc0ed.png	image/png	1,861	2018/8/30 下午12:12:48
bbe1c9793aca6bbf937e7045dd87127f.png	image/png	14,774	2018/8/30 下午12:12:48
e36a734c1f63a6690aa549773172437d.jpg	image/jpeg	438,821	2018/8/30 下午12:12:34
e888863663be026fbc4718a7d61c1bcf.jpg	image/jpeg	2,063,444	2018/8/30 下午12:12:48
fd70598c48fee6ba51641e7b73bf17e.png	image/png	1,623	2018/8/30 下午12:12:48

圖 6-6 已快取圖片清單

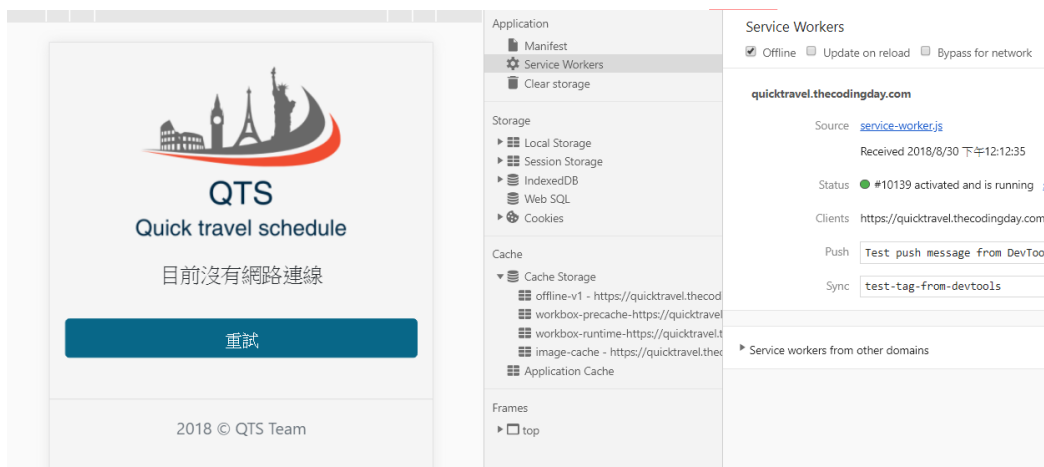


圖 6-7 Service Worker 實現網站離線能力

本專題使用 Service Worker 對圖片、JS File、CSS File 進行快取(圖 6-5、圖 6-6)，減少網路流量使用並且加快載入速度，並且具有網站離線運作能力(圖 6-7)。

6.3 體驗優化

本專題在一開始即有考慮到使用者體驗的部分，因此在評估旅遊規劃系統之初便因考量到會需要與使用者有著較多互動而使用 ReactJS 進行開發，在導入 PWA 標準後其特性更可以進一步提升使用者體驗。符合 PWA 應用標準的網頁應用可以直接進入系統桌面，與傳統 APP 相比更有著不用下載之優勢，相比傳統網頁應用更提高了使用者觸及率

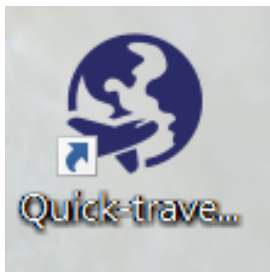


圖 6-8 進入 Windows 桌面

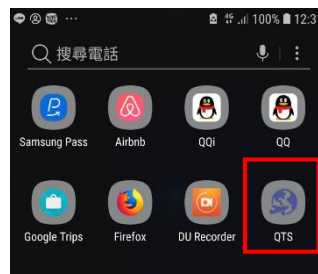


圖 6-9 進入安卓程式選單

7、旅遊規劃情境與展示

本章將設計多個情境舉例說明可能的狀況與進行專題展示。本章將旅遊規劃所需必要資訊設定如表 7-1 所示，並且搭配不同情境示範旅遊規劃。

表 7-1 旅遊規劃必要資訊

項目	參數
旅遊地區	臺中
旅遊天數	2 天
旅遊起始時間	早上 8:00
旅遊起始地點	臺中火車站
旅遊結束時間	晚上 9:30
旅遊結束地點(或回程點)	臺中高鐵站

7.1 情境 1 - 不熟悉該地區，但是想前往旅遊

此情境使用推薦模式，該模式以每次提供三個地點給於使用者進行選擇(圖 7-1)，並且提供地點介紹、交通、花費、停留時間等資訊予使用者。在使用者進行多輪選擇並且填滿行程後便完成旅遊規劃並且結果將如 7.3 節所示。

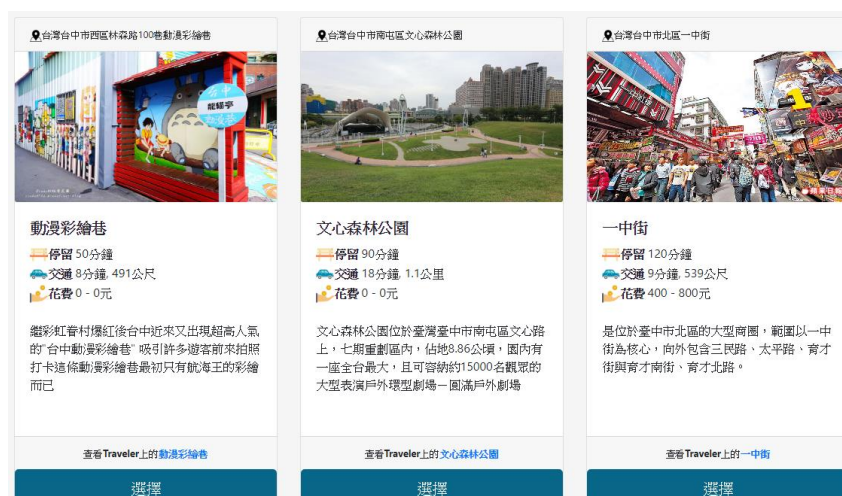


圖 7-1 推薦模式提供景點資訊供使用者選擇

7.2 情境 2－只想去某幾個點，不想做其他旅遊規劃

此情境使用全自動規劃搭配指定地點，讓使用者快速準備好旅遊前規劃，輕鬆享受整趟旅程，在本節將指定景點必須包含逢甲夜市、秋紅谷廣場(圖 7-2)，而剩下將由系統進行規劃。

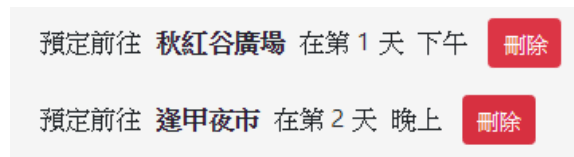


圖 7-2 指定旅遊地點

使用者指定完成地點後系統便開始規劃旅遊，等待系統規劃完成後結果將如 7.3 節所示。

7.3 旅遊規劃結果

本節將以 7.2 節之情境結果為主並進行說明，完整結果可參考附件 3 所附規劃結果分享網址。在旅遊規劃完畢後將顯示旅遊規劃之結果畫面(附件 3、圖 7-3、7-4)並且搭配 Google 地圖提供路線建議。

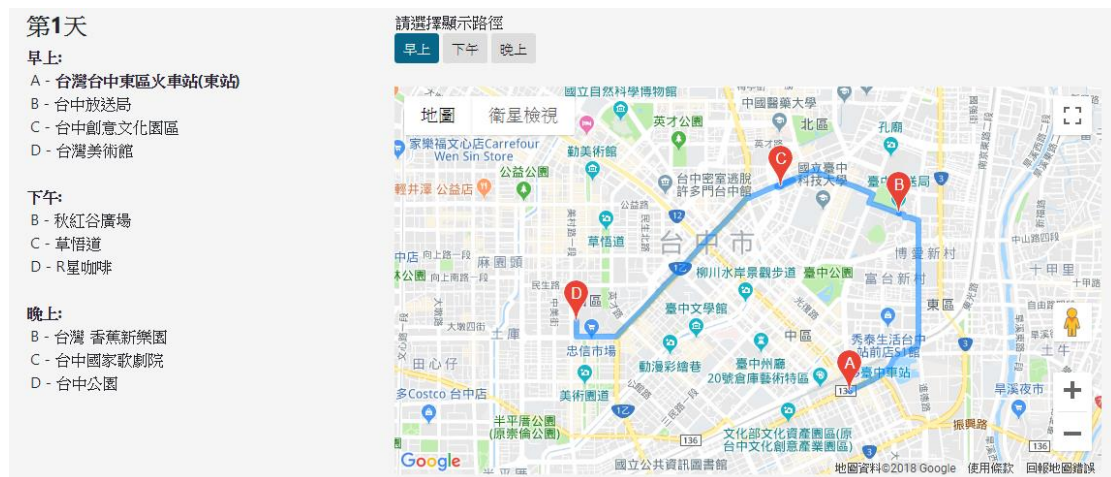


圖 7-3 規劃結果-第一天

在第一天的規劃結果(圖 7-5)可以看到成功的在第一天的下午將指定旅遊景點(秋紅谷廣場)加入旅遊規劃結果。

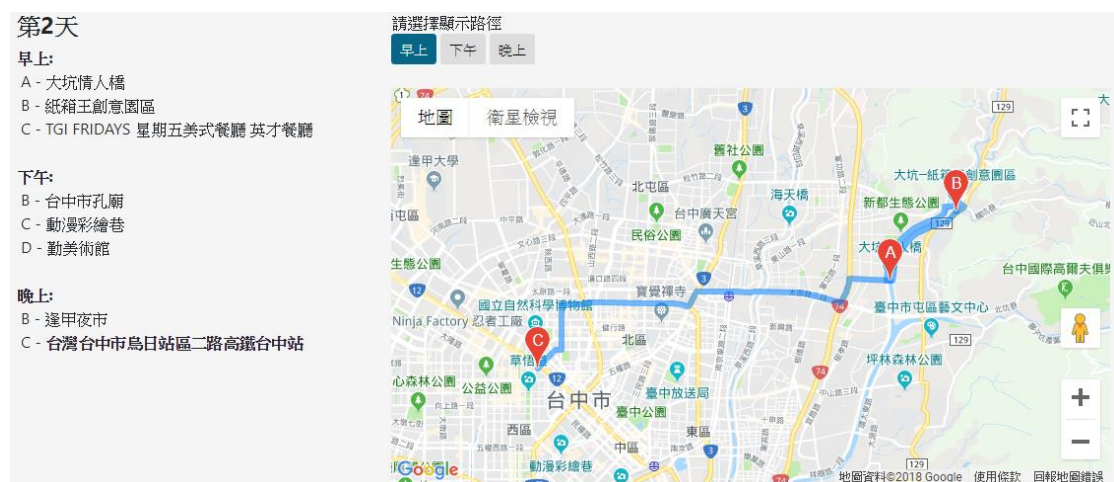


圖 7-4 規劃結果-第二天

第二天的規劃結果也可以看到成功的將第二個指定旅遊景點(逢甲夜市)加入旅遊規劃中，並且在最後前往台中高鐵結束旅遊。

7.4 分享旅遊



圖 7-5 分享按鈕

在結果畫面也提供多個不同社群之分享按鈕(圖 7-5)，方便與親朋好友分享本次旅遊規劃結果或是邀請一同參與旅遊。

8、結論與心得

本次專題是我目前做過最大規模的專案，也是最複雜、使用技術最多的一次，從題目構想、規劃一路到成品都花了不少的心力在上面，透過不斷與組員和指導老師討論、測試、重構、修正，最終才得以實現。

本次專案得以有今日成果我覺得可以歸功於良好的團隊合作成果與工具選擇；早在專題的初始階段本專題便導入 Git 進行版本控制與多人合作開發，而在決定開發工具時也盡量以框架作為優先選擇，使得開發風格得以盡量保持一致，結合上述在開發時若遇到問題有經驗的組員便可以快速地相互支援。

本專題成功的藉由兩個系統(旅遊資訊蒐集平台、旅遊規劃系統)達到當初設定之目標，大膽將想法轉化為現實，並且效果良好。此次專題也評估過市面上目前沒有相同之服務，也符合我的初衷：「想做一個創新且具有意義的專題」。

但是本專題仍有改善空間，如：經測試部分使用者在使用時覺得輸入資料的方式仍有改善空間，以及對於已經規劃完成的旅遊目前無法編輯節點或進行微調，而專題內部使用方法也較依賴 Google APIs 使得流量不斷增加的同時也會產生高額的資源使用費。

9、專題程式碼

本專題使用 GitLab 進程式碼託管、運用雲端特性進行多人開發，並且將項目切成多個 Repository 進行管理。

1. Philip / quick-travel-schedule

(1) 連結：<https://gitlab.com/p208p2002/quick-travel-schedule>

(2) 功能說明：本專題主 Repo，存放了專案前期企劃、邏輯設計、資料庫設計、API 設計等文件

2. Philip / quick-travel-schedule-laravel

- (1) 連結：<https://gitlab.com/p208p2002/quick-travel-schedule-laravel>
- (2) 功能說明：旅遊資訊平台程式碼 Repo，旅遊資訊平台程式碼與 API 程式碼均儲存
在此。

3. Philip / quick-travel-schedule-react

- (1) 連結：<https://gitlab.com/p208p2002/quick-travel-schedule-react>
- (2) 功能說明：旅遊規劃系統程式碼 Repo，為使用者介面、旅遊規劃邏輯等實現的地
方。

4. Philip / quick-travel-schedule-react-server

- (1) Repo 連結：<https://gitlab.com/p208p2002/quick-travel-schedule-react-server>
- (2) 功能說明：佈署旅遊規劃系統時所用到的後端，使用 Lumen 框架。

10、附件與連結

1. 旅遊規劃系統
<https://quicktravel.thecodingday.com/>
2. 旅遊資訊蒐集平台
<https://traveler.thecodingday.com/>
3. 旅遊規劃結果連結網址
<https://quicktravel.thecodingday.com/schedule-result?sid=94>
4. 自動旅遊規劃系統開發 Git Repository
<https://gitlab.com/p208p2002/quick-travel-schedule>

11、參考資料

1. 漸進式網頁應用
<https://developers.google.com/web/progressive-web-apps/>
2. Your first PWA
<https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp/>