

# 以元件為基礎軟體工程開發的地圖視覺化服務系統 — 以旅遊景點為例

## A Map Data Visualization Service System with Component-based Software Engineering Development – A Case Study of Travel Attractions

何承遠<sup>1,\*</sup>、洪立宇<sup>1</sup>、王家恩<sup>1</sup>、鐘岑哲<sup>1</sup>、楊永聖<sup>1</sup>、黃照宇<sup>2</sup>、張偉瑜<sup>1</sup>

<sup>1</sup> 亞洲大學 資訊工程學系

<sup>2</sup> 亞洲大學 創意商品設計系

Cheng-Yuan Ho<sup>1,\*</sup>, Li-Yu Hung<sup>1</sup>, Jia-En Wang<sup>1</sup>, Cen-Zhe Zhong<sup>1</sup>, Yong-Sheng Yang<sup>1</sup>,  
Chao-Yu Hwang<sup>2</sup>, Hsing-Yu Chang<sup>1</sup>

<sup>1</sup> Dept. of Computer Science and Information Engineering, Asia University

<sup>2</sup> Dept. of Creative Product Design, Asia University

Email: tommyho@asia.edu.tw\*

### 摘要

隨著資訊科技的發展，人們的需求也跟著不斷變動與增加，進而造成整體社會與商業環境，競爭越來越激烈，服務發展的越來越精進。為了要應付客戶不斷變動的需求和越來越嚴格的品質要求，服務提供者所提供的服務不僅要有足夠的彈性、擴充性和穩定性外，還要考量到更多樣的相關項目，例如：服務穩定、網路流量和資料安全與確認等。因此，如何快速開發一件產品/服務以及運用過去經驗開發新產品便是一個很重要的議題。另外，近年來地圖視覺化是一個很熱門的主題，相對於純文字、數字和地理資料的顯示，使用者可以較容易藉由視覺化後的圖表或地域特徵觀察、理解或解讀隱藏在文字、數字和地理資料的意義和趨勢。

基於上述原因，本研究採用以元件為基礎的軟體工程開發方式(或稱元件式)架構一個視覺化地圖的服務系統，前端負責在地圖上顯示經資料處理以及視覺化後的資訊，而後端專門用以收集、整理和處理資料、管理會員以及支援推薦系統等系統和服務，並以旅遊景點為案例說明本系統之應用。除此之外，當需要提供其他應用服務時，僅需要將本系統複製一份與修改相對應的元件和模組便可快速拓展出一個專門支援該應用的系統。

**關鍵字：**元件為基礎軟體工程、視覺化技術、地圖服務、旅遊景點

### 一、序論

資訊系統開發模式(Information System Development model)或稱為軟體流程模式(Software Process Model)自從資訊系統和軟體問世以來一直是一個很重要的課題。隨著技術的進步、軟硬體技術的精進、價格大幅地降低以及資訊系統的需求量和複雜度大幅地提升下，如何有效地開發資訊系統顯得越來越重要。除此之外，如何在服務與供需之間找到一個平衡點這個問題更是使得資訊系統開

發模式發展出各式各樣的特色。簡而言之，資訊系統開發模式可謂是資訊系統開發活動一系列的步驟及執行程序。當系統開發依循著系統化和邏輯化的步驟進行時，有利於標準、規範與政策之推行和建立，同時，開發的過程將會更有效率、更能確保品質，也更容易管理。

不同的資訊系統開發模式，適用於不同情況的系統開發，像是有的模式定義嚴謹的步驟與控制點，以處理明確需求；有的則是強調以快速的雛型開發，讓不確定的需求端可以依據此雛型逐漸完善其需求；有的則因資金或市場面的考量，使得主要目標在於控制風險或縮短時程而採用漸增方式。每個資訊系統開發模式都有其優缺點和合適性，並無絕對的好壞之分，至於每個模式的特徵與優缺點將於下個章節說明。

隨著社會經濟的發展、週休二日的普及，民眾的收入與可支配時間的增加，花費在休閒娛樂的事物也隨之提升。在國人的生活中，短時間的旅遊與美食成了一個較普遍的放鬆方法 [1-2]。近年來，隨著互聯網的發達，越來越多面向的旅遊體驗資訊、影音、影像與經驗分享在社群媒體網站中流動，進而產生日新月異的科技影響旅遊現象，同時促使旅遊分享行為成為在社群媒體間提升自己社交地位的一種方法 [3]。然而，要挑選一個景點可能需要事先耗費大量時間在瀏覽資訊，尤其現今這個資訊爆炸的時代、資料來源多元化後，人們反而花費大量時間在篩選自己真正需要的資料，進而造成多數人不想外出的原因之一，圖一示意出資訊量與人們花費時間。更進一步來說，以往的旅行景點選擇，多半是使用者透過閱覽大量旅遊資訊並篩選得到。而現有的旅行系統多著重於景點介紹以及規劃旅程，甚至是套用已經排定好的套裝行程。在介面呈現與景點選擇上忽略使用者真正的需求和其使用者經驗(user experience)。

有了在背後支援的資訊系統以及鎖定旅遊景點的應用服務後，如何讓使用者願意嘗試使用新軟體，這與使用者的第一印象有很大的關係。人們是美感的生物，因此資訊和資料的呈現具有極大的影

\* 本研究接受科技部編號：MOST 108-2221-E-468-010 - 研究計畫經費補助

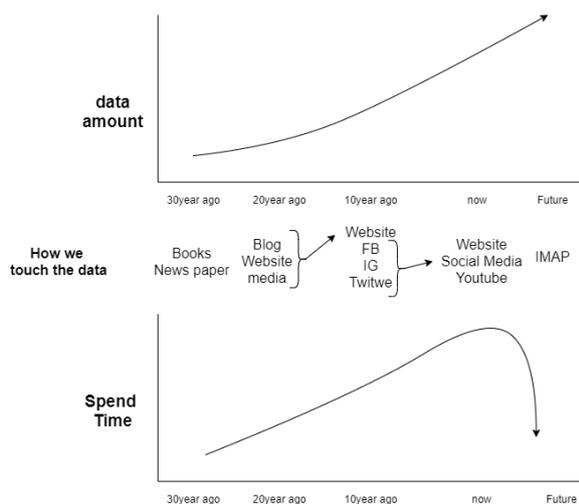


圖 1 不同時代資訊量與花費時間對比圖

響。現今與地理相關資訊較熱門的呈現方式為地圖視覺化，也就是將地理資料轉換成視覺化形態，通過具有地域特徵的資料或者資料分析結果以形象方式表現在地圖上，使得使用者可以更加容易理解資料規律和趨勢。

因此，本研究發展重心為一個以元件為基礎軟體工程開發的地圖視覺化服務系統，簡稱 iMap。顧名思義，本研究透過以元件為基礎軟體工程 (component-based software engineering, 或稱元件式) 為開發系統的開發模組與流程來制定相關的系統元件和功能，並以地圖視覺化的方式呈現針對使用者需求的資料分析結果，並以旅遊景點作為本系統的第一個應用。更進一步而言，本系統可根據不同的應用服務，調整需要微調的元件，便可快速拓展出一個專門支援該應用的系統。

本文章的結構如後。第二節介紹各種資訊系統開發模式的優缺點；iMap 的架構和各個元件以及最終呈現的畫面和使用流程將分別於第三節和第四節介紹與說明；並於最後一節敘述 iMap 的未來發展與工作。

## 二、資訊系統開發模式特徵

系統開發模式發展至今已有多數的理論及方法出現，而這些開發模式演化的主要目的在於降低系統開發成本、提高軟體品質以及縮短開發時程等。根據[4-5]，表一整理一些常見開發模式的特徵與其優缺點，從中可以看出有的開發模組偏向定義每次階段的工作，有的邊做邊修改邊前進，而有的開發模組偏向人與人之間的溝通等。

資訊科技發展與應用一日千里，造成很多運營商和企業對於軟體和系統的彈性和效率要求遠遠勝於過去，也因此軟體與系統提供商不僅需要滿足運營商和企業，更要滿足其客戶的需求。例如：軟體和系統要能支援分散在世界各地的經營管理，同時也要整合上下游廠商資訊系統，用以累積營運知識和經驗並能強化資訊的即時分析和處理。再加上，使用者的需求日益增多，造成系統模組不斷被開發

與不斷地快速膨脹，造成模組彼此間錯綜複雜的連結關係，使得後續維護管理工作極為困難，浪費龐大的系統開發資源。而以元件為基礎的軟體工程可以避免模組不斷被開發，盡可能重複使用相同的元件來完成相同的事情與工作。

## 三、iMap 系統架構與元件介紹

iMap 的系統架構如圖二所示，可分成前後兩端。前端負責與使用者互動與顯示資料分析跟統計結果，而後端主要負責資料所有處理，包含採集、清洗、預處理、統計和分析。更進一步而言，前端利用相同技術與元件開發促使使用者可以透過 APP 或 web 瀏覽器方式取得資料後得到相同顯示畫面以及使用者經驗。下面各段落將一一介紹 iMap 各個元件。

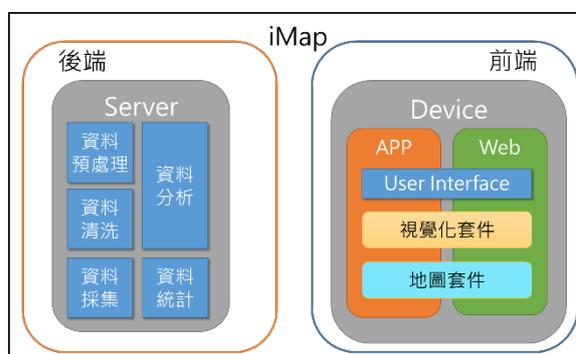


圖 2 iMap 系統架構

### (一) 前端 Device 部分

使用者可以透過 APP 或 Web 使用 iMap 的服務，為了要讓使用者獲得相同的使用者經驗與操作方式，iMap 在顯示的部分使用相同套件，分別如下所述。

#### (1) User Interface

iMap 目前是以網頁 Web 為基底，故採用市面上最常見的 HTML5、CSS3 和 JavaScript [6] 套件建立一個流暢且友善的使用者介面，APP 的部分考量目前開發成本和效能不要太差的要求下，暫時以 PhoneGap 跨平台框架 [7] 做轉換。待未來有能力再改版開發成原生性 APP 或跨平台 APP。

#### (2) 視覺化套件

以 JavaScript 與旗下 jQuery [8] 跟 D3.js [9] 函式庫為主，並用 Echarts [10] 和 finereport [11] 為輔助，目的在於讓使用者可以很容易從地圖上取得所要的資訊。例如：縮小地圖時看到的是在某一區符合使用者需求的地點個數，讓使用者有個大概念，放大地圖則會顯示每個符合條件的地點。此時點擊地點會出現詳細內容，如照片、地址、簡介、開放時間、票價等。

#### (3) 地圖套件

表 1 資訊系統開發模式特徵和優缺點

名稱	特徵	優點	缺點
瀑布式	<ul style="list-style-type: none"> <li>◇ 開發階段有清楚的定義，每階段均需考量完整的系統範圍，且各階段僅循環一次</li> <li>◇ 強調先有完整的設計與規劃，再進行編碼</li> <li>◇ 重視設計與規劃之文件</li> <li>◇ 一階段的完成需經驗證通過，才能進入下一階段</li> </ul>	<ul style="list-style-type: none"> <li>◇ 階層式開發</li> <li>◇ 任務明確</li> </ul>	<ul style="list-style-type: none"> <li>◇ 使用者需求不了解</li> <li>◇ 系統再用性低、風險高</li> </ul>
漸進式	<ul style="list-style-type: none"> <li>◇ 開發階段有清楚的定義，把整個系統範圍分解成若干個子系統，各子系統之開發可依序以瀑布模式進行，亦可平行進行再整合</li> <li>◇ 強調先有完整的設計與規劃，再進行編碼</li> <li>◇ 開發週期反覆的進行</li> </ul>	<ul style="list-style-type: none"> <li>◇ 快速產生系統雛型</li> <li>◇ 改良瀑布式於系統完成前任何產品可使用之缺點</li> </ul>	<ul style="list-style-type: none"> <li>◇ 系統必須具備高度彈性</li> <li>◇ 後續版本可能曲解系統目的</li> <li>◇ 系統版本功能不完善</li> </ul>
螺旋式	<ul style="list-style-type: none"> <li>◇ 系統開發階段無清楚之分野，且開發週期反覆地進行</li> <li>◇ 不強調先有完整的設計與規劃再進行編碼</li> <li>◇ 強調快速完成雛型且盡早使用，以作為雙方需求溝通與學習的工具</li> <li>◇ 強調各開發週期之規劃與風險評估</li> </ul>	<ul style="list-style-type: none"> <li>◇ 降低系統開發風險</li> <li>◇ 減少開發費用</li> </ul>	<ul style="list-style-type: none"> <li>◇ 開發人員必須具備風險評估知識與技術</li> <li>◇ 風險因素需完善考量</li> </ul>
結構化	<ul style="list-style-type: none"> <li>◇ 先進行系統的分析，以適當的資料結構加以描述，然後再建立相對應的執行流程，進而產生程式邏輯的架構</li> <li>◇ 採用資料流程圖及結構表來描述系統的功能</li> </ul>	<ul style="list-style-type: none"> <li>◇ 使用資料結構圖形描述系統</li> <li>◇ 系統架構、功能明確</li> </ul>	<ul style="list-style-type: none"> <li>◇ 結構太過嚴謹複雜</li> <li>◇ 使用者需求增加時，修改開發</li> <li>◇ 成本龐大</li> </ul>
物件導向	<ul style="list-style-type: none"> <li>◇ 物件導向模式主要運用物件(Object)的四項基本特質(封裝、繼承、抽象化、多型)來進行系統的開發</li> </ul>	<ul style="list-style-type: none"> <li>◇ 系統軟體再用、資訊封裝</li> <li>◇ 降低系統建置、維護複雜度</li> </ul>	<ul style="list-style-type: none"> <li>◇ 系統分析、設計較為困難</li> <li>◇ 系統開發前置時間較長</li> </ul>
元件式	<ul style="list-style-type: none"> <li>◇ 將具備特定功能且可獨立運作的程式單元，相互連結成為應用系統，使得不同程式設計師所開發的軟體元件能彼此呼叫使用</li> </ul>	<ul style="list-style-type: none"> <li>◇ 降低系統設計、管理複雜度</li> <li>◇ 降低維護修改成本</li> <li>◇ 平行開發、縮短時程</li> <li>◇ 系統更具彈性、降低整合複雜度</li> </ul>	<ul style="list-style-type: none"> <li>◇ 需使用標準或特定的技術規範</li> <li>◇ 對物件導向技術與系統功能需求充分了解</li> </ul>
敏捷式	<ul style="list-style-type: none"> <li>◇ 強調開發團隊與使用者間協同合作</li> <li>◇ 強調反覆與漸增的開發方式</li> <li>◇ 強調隨時因應變化</li> </ul>	<ul style="list-style-type: none"> <li>◇ 個體和互動勝過過程和工具</li> <li>◇ 可以工作的軟體勝過面面俱到的文件</li> <li>◇ 客戶合作勝過合同談判</li> <li>◇ 響應變化勝過遵循計劃</li> </ul>	<ul style="list-style-type: none"> <li>◇ 團隊成員的了解必須一致，工作能力最好也要差不多</li> <li>◇ 並不會真正加快速度，只是讓速度透明，讓大家專注重重要的事情</li> <li>◇ 不會解決工作內容的相依性</li> <li>◇ 注重人員溝通，忽略文件的重要性，若專案人員流動太大，會給維護帶來不少難度</li> </ul>

Google Map 為常用的地圖套件，除了安卓系統本身內建外，Google Map 在 iOS 系統上也很受使用者的喜愛，故開發初期直覺使用 Google Map API [12] 為 iMap 的地圖套件與應用程式介面(API)。利用 Google Map API 提供的功能可以做到地圖顯示、地點標記(簡稱標點)、標點內容詳細介紹、標點群、地圖區域範圍區塊等。然考量到 iMap 未來在地圖上的功

能與彈性，決定將地圖套件換成 Mapbox [13]。雖然目前 Mapbox 在地圖的準確度上還比不上 Google Map，但他在創造力、地圖特製、地勢、路線等顯示方式都可以由開發者自行決定和調整。

(二) 後端伺服器(Sever)部分

後端伺服器為了提供使用者需求的資訊與加值後的結果，具有五個元件，分別為資料採集、資料清洗、資料預處理、資料統計和資

料分析。這五個元件做的事項分別一一敘述如下。

#### (1) 資料採集

為了盡可能滿足使用者需求，iMap 在資料量上需要做到資料全體性、完整性，而不是抽樣。因此，iMap 的資料採集元件利用 Python [14]、Beautiful Soup [15]、Pandas [16]等撰寫了一隻爬蟲自動收集與該應用有關的所有可能資料。步驟為 a. 模擬使用者連結網址(url)、發送請求和下載網頁；b. 解析網頁內容；c. 結構化資料，d. 擷取所需資料；e. 反覆 a~d 四個步驟直到整個網頁擷取完畢；f. 反覆 a~e 五個步驟，直到整個網站擷取完畢；g. 將上述擷取資料儲存成檔案存入資料庫中。

#### (2) 資料清洗

資料擷取回來後並不一定能直接使用，因為有時可能會有空值(null)、字串全連在一起或是原本資料錯置等問題，故需要先清洗過。常用的資料清洗方法有處理空值、捨棄不要的行列、重置並捨棄索引、切割欄位、反向倒置和正規化等等方法。

#### (3) 資料預處理

清洗完後的資料仍不一定能馬上使用，還需要再經過一些處理，像是取得的地理資訊大多是經緯度，但使用者習慣看到的是路段巷弄這類地址，故需要再處理。以轉址為例，iMap 利用 geoCoder [17]來達成轉址、在地圖上標點、利用經緯度尋找地址、找尋所在區域，甚至可以利用 ip 位置找到所在的區域等需求。

#### (4) 資料統計和資料分析

利用 Pandas 做些簡單彙總，像是先算出排名前多少的值、常用的最大最小平均數等、與特定年齡或性別有關的值

等等，以利使用者一般需求，當使用者有特殊需求時再即時運算。

### 四、iMap 之旅遊景點應用情境與操作介面

上節介紹了 iMap 的系統架構和元件，這節將介紹 iMap 的介面設計和以旅遊景點為實際案例。

iMap 介面設計秉持了以下三個理念：

1. 簡潔：iMap 是一項融入使用者日常生活中的服務，因此不添加多餘天花亂墜的裝飾設計，而以簡潔與清楚的風格來呈現資料，以融入使用者之生活中。

2. 溫度：服務不能缺少的就是「溫度」，有溫度的服務才能更貼近使用者的生活，因此 iMap 所使用的是溫暖的紅色與樂觀的黃色所融合在一起之顏色「橘色」，讓使用者在使用本服務時，可以感受到溫暖的氛圍。

3. 直覺/藉由他方服務經驗連結到本服務：一項新穎的服務，若功能齊全但不易使用的話，仍是項扣分的服務，不會吸引太多客戶使用他。因此，為了能讓使用者憑「直覺」就能使用本服務，在設計上參考市面上很多產品和服務，以及透過問卷調查周遭親朋好友的習慣用法。

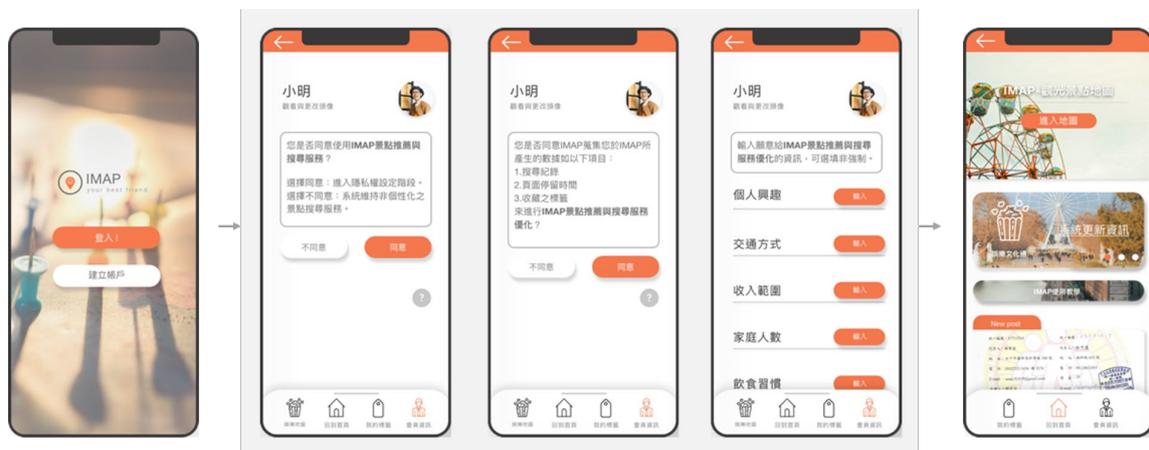
圖三到圖五顯示 iMap 之旅遊景點的使用流程，包含登入、設定、服務、地圖、景點簡介等頁面。

另外，本研究針對市面上的旅遊景點產品與地區所在做了四張比較表，以更深入了解他人產品的優劣勢。從表二到表五可以看出本產品還是有機會獲得使用者的信賴和關注。

### 五、結論

本研究以元件為基礎軟體工程開發一種地圖視覺化服務系統，前端負責在地圖上顯示經資料處理以及視覺化後的資訊，而後端專門用以收集、整理和處理資料並以旅遊景點為案例說明本系統之應用以及本旅遊景點產品跟市面上產品的比較。

本研究目前主要透過地圖視覺化方式實現協助使用者進行某項應用的功能，如旅遊。未來希望



1. 登入頁面

2. IMAP-景點推薦與搜尋服務隱私權設定頁面

3. 服務頁面

圖 3 登入、設定、服務頁面



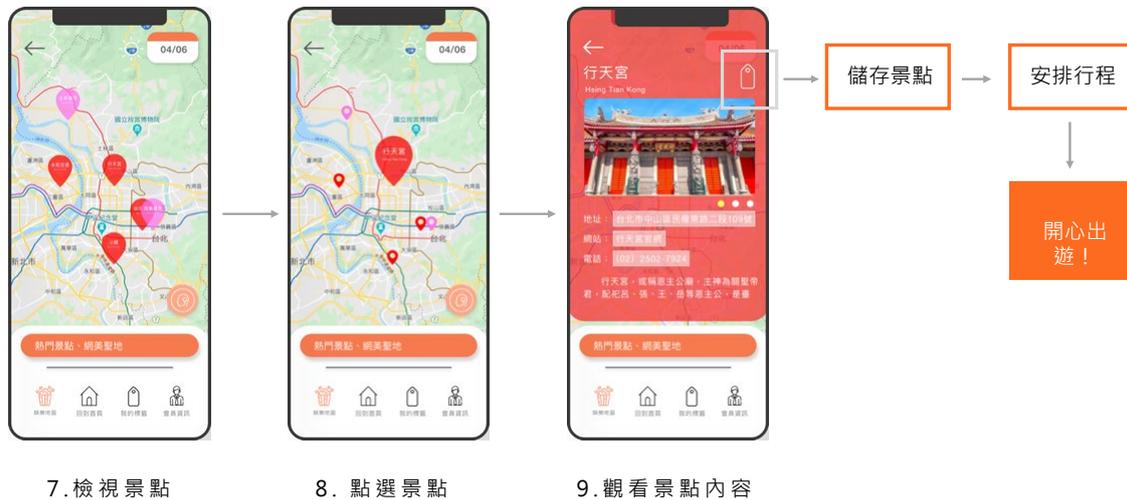
3. 服務頁面

5. 地圖頁面

6. AI景點推薦模式

7. 放大檢視推薦景點

圖 4 服務、地圖、景點頁面



7. 檢視景點

8. 點選景點

9. 觀看景點內容

圖 5 景點檢視頁面

表 2 台灣地區相關旅遊產品對於台灣旅遊景點的優劣勢

產品名稱	MAP8	TAIWAN -MAP	TGOS MAP	URMAP
優勢	地圖介面完整	功能多元, 有主題地圖	功能多元, 且有主題地圖	地圖介面完整
劣勢	1. 景點搜尋服務缺乏規劃, 沒有資料排序之規則 2. 缺乏景點推薦與探詢功能	1. 景點搜尋服務缺乏規劃, 使用者需要一一選擇景點種類後, 再按搜尋, 才能進行搜尋 2. 缺乏景點推薦與探詢功能 3. 搜尋結果不準確	1. 景點搜尋服務缺乏規劃 2. 缺乏景點推薦與探詢功能 3. 搜尋結果不準確	1. 景點搜尋服務缺乏規劃, 使用者需要一一選擇景點種類後, 再按搜尋, 才能進行搜尋 2. 缺乏景點推薦與探詢功能 3. 搜尋結果不準確

表 3 大陸地區相關旅遊產品對於台灣旅遊景點的優劣勢

產品名稱	高德地圖	百度地圖	騰訊地圖
優勢	1.地圖介面完整 2.景點搜尋與推薦功能完整	1.地圖介面完整 2.景點搜尋與推薦功能完整	1.地圖介面完整 2.景點搜尋與推薦功能完整
劣勢	1.台灣相關景點資料不齊全 2.不符合台灣人之使用習慣 3.缺乏個性化之推薦與搜尋服務	1.台灣相關景點資料不齊全 2.不符合台灣人之使用習慣 3.缺乏個性化之推薦與搜尋服務	1.台灣相關景點資料不齊全 2.不符合台灣人之使用習慣 3.缺乏個性化之推薦與搜尋服務

表 4 全球服務地區相關旅遊產品對於台灣旅遊景點的優劣勢

產品名稱	GOOGLE MAP	APPLE MAP	WINDOWS MAP
優勢	1.地圖介面完整 2.景點搜尋與推薦功能完整 3.知名度高	1.地圖介面完整 2.景點搜尋與推薦功能完整 3.知名度高	1.地圖介面完整 2.景點搜尋與推薦功能完整 3.知名度高
劣勢	缺乏個性化之推薦與搜尋服務	缺乏個性化之推薦與搜尋服務	缺乏個性化之推薦與搜尋服務

表 5 相關旅遊網站對於台灣旅遊景點的優劣勢

產品名稱	TRIPADVISOR	TRAVELKING	四方通行旅遊	交通部觀光局
優勢	1.服務介面完整 2.景點搜尋功能完整 3.知名度高	1.地圖介面完整 2.景點搜尋功能完整	1.地圖介面完整 2.景點搜尋功能完整	1.地圖介面完整 2.景點搜尋功能完整
劣勢	缺乏個性化之推薦與搜尋服務	缺乏個性化之推薦與搜尋服務	缺乏個性化之推薦與搜尋服務	缺乏個性化之推薦與搜尋服務

能在本系統加入管理會員、支援推薦系統、預覽行程、模擬跟提升行前/使用前體驗、回饋機制等功能、服務和系統。透過問卷、使用者反應和回饋，甚至是功能性錯誤回報等來改善本系統。除此之外，當本研究也會試著發展第二種應用服務來檢測本系統是否真的能快速拓展出一個專門支援該應用的系統。

#### 參考文獻

- [1] 楊芮瑄、葉道明, "根據使用者偏好以視覺化技術分析適配旅行景點之決策輔助系統研究", 2019 台灣軟體工程研討會 (TCSE), 2019.
- [2] 李世宏, "民宿旅遊地意象與旅遊意願之研究", 亞洲大學休閒與遊憩管理學系碩士論文, 2008.
- [3] 陳虹伶、唐玄輝, "旅遊資訊彙整平台的用戶體驗設計與研究", 工業設計, 138 期, 第 43-47 頁, 2018.
- [4] 陳柏辰, "以元件式設計建構晶圓代工廠 CRM 架構—實作換單管理系統", 交通大學工業工程與管理學系碩士論文, 2001.
- [5] 吳仁和, "物件導向系統分析與設計: 結合 MDA 與 UML", 元照出版, 第六版, 2020.

- [6] Javascript 官網, <https://www.javascript.com/>
- [7] PhoneGap 官網, <https://phonegap.com/>
- [8] jQuery 官網, <https://jquery.com/>
- [9] D3.js 官網, <https://d3js.org/>
- [10] Echarts 官網, <https://echarts.apache.org/en/index.html>
- [11] finereport 官網, <https://www.finereport.com/>
- [12] Google Map API 官網, <https://cloud.google.com/maps-platform/>
- [13] Mapbox 官網, <https://www.mapbox.com/>
- [14] Python 官網, <https://www.python.org/>
- [15] Beautiful Soup 官網, <https://www.crummy.com/software/BeautifulSoup/>
- [16] Pandas 官網, <https://pandas.pydata.org/>
- [17] geoCoder 開發文件, <https://developers.google.com/maps/documentation/javascript/reference/geocoder>