國立臺北商業大學

資 訊 管 理 系

108’資訊系統專案設計

**系統手冊**



**組 別：第108207組**

**題 目：Chavel**

**指導老師：葉明貴老師**

**組 長：10737011張祥智**

**組 員：10736010楊淯絨 10736012林宜萱**

**10736025顏毓陞 N1076716呂芷瑛**

**中華民國108年5月15日**

目錄

[**第一章 背景與動機** 1](#_Toc24994407)

[**1-1系統簡介** 1](#_Toc24994408)

[**1-2問題與機會** 1](#_Toc24994409)

[**1-3相關系統探討** 2](#_Toc24994410)

[**第二章 系統目標與預期成果** 3](#_Toc24994411)

[**2-1系統目標** 3](#_Toc24994412)

[**2-2預期成果** 3](#_Toc24994413)

[**第三章 系統規格** 4](#_Toc24994414)

[**3-1系統架構** 4](#_Toc24994415)

[**3-2系統使用環境** 4](#_Toc24994416)

[**3-3開發標準與使用工具** 5](#_Toc24994417)

[**第四章 專案時程與組織分工** 6](#_Toc24994418)

[**4-1專案時程** 6](#_Toc24994419)

[**4-2專案組織與分工** 7](#_Toc24994420)

[**第五章 需求模型** 8](#_Toc24994421)

[**5-1使用者需求** 8](#_Toc24994422)

[**5-2使用個案圖** 9](#_Toc24994423)

[**5-3活動圖** 9](#_Toc24994424)

[**5-4分析類別圖** 12](#_Toc24994425)

[**第六章設計模型** 13](#_Toc24994426)

[**6-1循序圖** 13](#_Toc24994427)

[**6-2設計類別圖** 15](#_Toc24994428)

[**第七章 實作模型** 16](#_Toc24994429)

[**7-1佈署圖** 16](#_Toc24994430)

[**7-2套件圖** 16](#_Toc24994431)

[**7-3元件圖** 17](#_Toc24994432)

[**7-4狀態機** 18](#_Toc24994433)

[**第八章 資料庫設計** 19](#_Toc24994434)

[**8-1資料庫關聯表** 19](#_Toc24994435)

[**8-2表格及其Meta data** 20](#_Toc24994436)

[**第九章 程式** 22](#_Toc24994437)

[**9-1元件清單及其規格描述** 22](#_Toc24994438)

[**9-2其他附屬之各種元件** 23](#_Toc24994439)

[**第十章 測試模型** 87](#_Toc24994440)

[**10-1測試計畫** 87](#_Toc24994441)

[**10-2測試個案與測試結果資料** 88](#_Toc24994442)

[**第十一章 操作手冊** 96](#_Toc24994443)

[**第十二章 使用手冊** 97](#_Toc24994444)

[**第十三章 感想** 103](#_Toc24994445)

[**第十四章 參考資料** 106](#_Toc24994446)

[**附件一 評審之建議修正情形** 107](#_Toc24994447)

**第一章 背景與動機**

**1-1系統簡介**

你曾和朋友一起出門遊玩過嗎?是不是在討論過程中倍感艱辛呢!也許是想去的地點太多不知如何篩選，也許是猶豫該如何排行程才能讓大家玩得盡興，又或者根本不想為了出去玩而特地下載APP來規劃旅遊，只能不停地在line群及網路上蒐集資料中徘迴。以上這些麻煩讓Chavel機器人幫助您解決。Chavel不只能夠輔助您排出順暢的行程，甚至不需要google瀏覽器就能在群組中提供你所需要的資訊供大家討論，那不是很方便嗎?在通訊軟體中，line是所有年齡層最能接受的，台灣人只要有手機，10人中有9人有用line，每天更有約800萬的台灣人使用LINE打電話，所以我們希望能開發一個幫忙排行程的line機器人，更快排出行程，而且規劃出最省時也最順路且令人難忘的旅遊。

**1-2問題與機會**

SWOT分析

▼表1-2-1 SWOT分析

|  |  |
| --- | --- |
| Strengths | Weaknesses |
| * 不需另載APP * 演算法排出推薦行程推薦 * 群組中提供服務，使用者方便討論 | * 所排出的推薦行程僅以地理為主 * 可能因資料過多而導致顯示資訊延遲 |
| Opportunities | Threats |
| * 台灣國內旅遊人數多 * Line為台灣主流社群媒體，使用門檻低 * 現階段台灣市場中並無以旅遊為主的linebot | * Linebot越來越普及且容易模仿 * 完善的旅遊網站、APP眾多紛紜 |

**1-3相關系統探討**

旅行蹤：結合google map、搜尋引擎、Blog來安排行程，安排完行程可分享到群組裡，同時也能查看航班、買機票，訂房，也能以最快速、最省錢的交通方式與適合的訂購交通票券，非常強大的一個旅行App。

Funliday：結合google map、搜尋引擎、Blog來安排行程，也有一些懶人行程範本參考，也能群組討論。

系統比較

▼表1-3-1 系統比較

|  |  |  |  |
| --- | --- | --- | --- |
| 名稱  功能 | Chavel | 旅行蹤 | Funliday |
| 行程規劃 | ⭕ | ⭕ | ⭕ |
| 附近景點 | ⭕ | ⭕ | ⭕ |
| 自動排行程 | ⭕ | ✖ | ✖ |
| 群組討論 | ⭕ | ⭕ | ⭕ |
| 購票安排 | ✖ | ⭕ | ✖ |

**第二章 系統目標與預期成果**

**2-1系統目標**

* 依據使用者訊息的關鍵字，顯示景點相關資訊
* 判斷所在位置，推薦附近景點
* 依據行程規劃推薦路線

**2-2預期成果**

* 朋友之間透過聊天方式，更加輕鬆安排行程
* 減少使用者安排行程以及搜尋資料的時間
* 同步旅程的資訊，減少不對稱性

**第三章 系統規格**

**3-1系統架構**

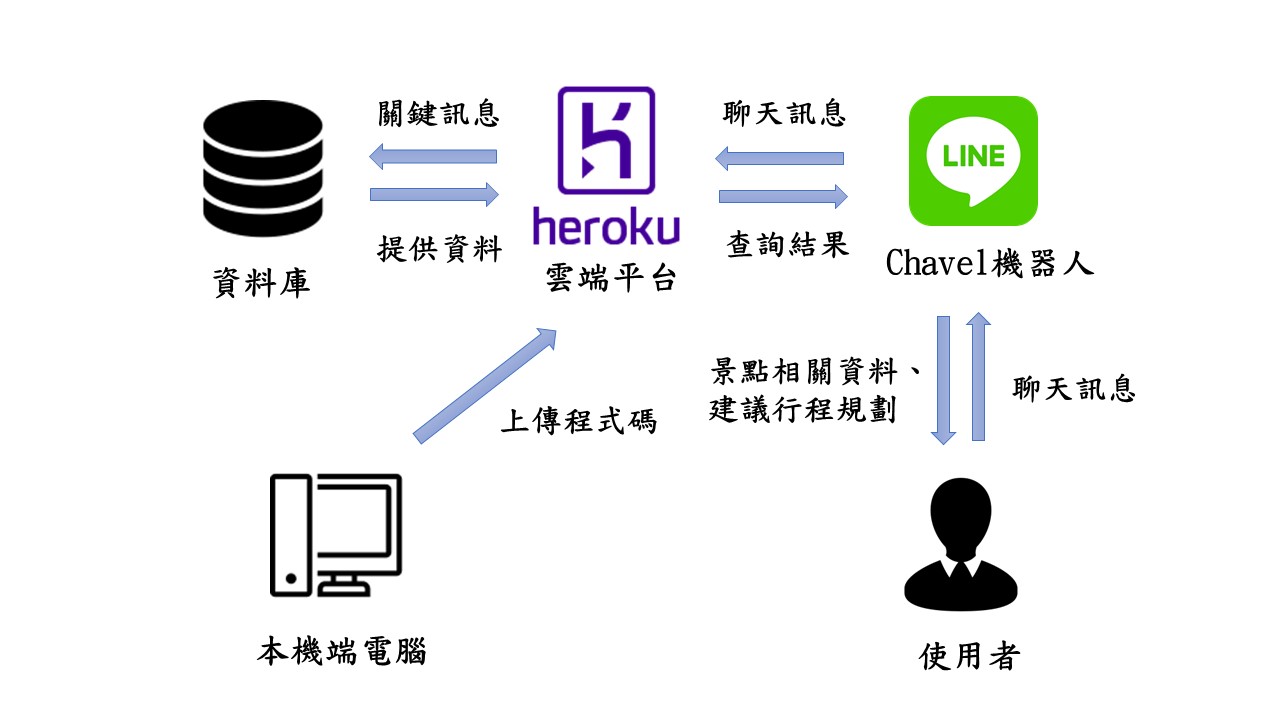


圖3-1-1 系統架構圖

**3-2系統使用環境**

▼表3-2-1 系統使用環境

|  |  |  |
| --- | --- | --- |
| 軟、硬體需求 | | |
| 作業系統 | LINE(Android、iOS、Windows、Mac) | |
| 版本 | Android 5.0以上 | |
| 行動裝置需求 | | |
| 版本 | | Android版本至少4.4(含)以上、iOS版本9以上 |
| 定位功能 | | GPS定位系統 |
| 網路需求 | | WI-FI/3G、4G行動網路 |

**3-3開發標準與使用工具**

▼表3-3-1 搜尋景點

|  |  |  |
| --- | --- | --- |
| 系統開發環境 | | |
| 作業系統 | | Windows7、Windows8、Windows10 |
| 資料庫 | | PostgreSQL |
| 資料庫管理介面 | | Navicat for PosgreSQL |
| 伺服器 | | Heroku Server |
| 程式開發技術 | | |
| LINE BOT | | Heroku Platform、JavaScript、GIT |
| 開發語言 | | Html5、CSS、JavaScript、JQuery、Node.js |
| 程式編輯器 | | Microsoft Visual Studio Code |
| 開發套件 | | Google SDK |
| 管理程式平台 | | |
| 專案管理 | github、google雲端 | |
| 設計工具 | | |
| 文件製作 | | Microsoft Word 2010、Microsoft Excel 2010、Google 文件 |
| 簡報製作 | | Microsoft PowerPoint 2010 |
| UML製作 | | Visual Paradigm for UML 7.2 Personal Edition |
| 設計圖樣 | | Adobe Illustrator CS6、Adobe Photoshop CS6 |

**第四章 專案時程與組織分工**

**4-1專案時程**

圖4-1-1 專案時程

**4-2專案組織與分工**

●主要負責 ○協助

▼表4-2-1 分工表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 成員  工作項目 | | 張祥智  10736011 | 林宜萱  10736012 | 楊淯絨  10736010 | 顏毓陞  10736025 | 呂芷瑛  N1076716 |
| 題目發想 | | ○ | ● | ○ | ○ | ○ |
| 程式設計 | | ● | ● | ● | ● | ● |
| 整體規劃 | | ● | ○ | ○ | ○ | ○ |
| 資料庫建  置與管理 | | ● |  |  | ○ |  |
| 設計 | 介面 |  | ○ | ● |  | ○ |
| LOGO |  |  |  |  | ● |
| 海報 |  | ● | ○ |  | ○ |
| 系統整合 | | ● | ○ | ○ | ○ | ○ |
| 文書 | 文件撰寫 | ● | ● | ● | ● | ● |
| 簡報製作 | ○ | ○ | ○ | ○ | ● |
| 學習技術 | | ● | ● | ● | ● | ● |

**第五章 需求模型**

**5-1使用者需求**

功能性需求

1. 使用者必須先加入好友才可以使用機器人的功使用者必須先加入好友才可以使用機器人功能
2. 使用者可以搜尋景點資訊
3. 使用者可以加入景點至行程，並且刪除景點(行程管理)
4. 使用者可以收藏景點和取消收藏(收藏管理)
5. 使用者可以查看附近景點
6. 使用者可以自己規劃行程或是機器人規劃推薦行程(行程規劃)

非功能性需求

1. 可以同時有300個群組加入機器人，機器人不會掛
2. 回覆速度不超過3秒
3. 加機器人好友及在line群上加入機器人就可使用功能

**5-2使用個案圖**

****

圖5-2 使用個案圖

**5-3活動圖**

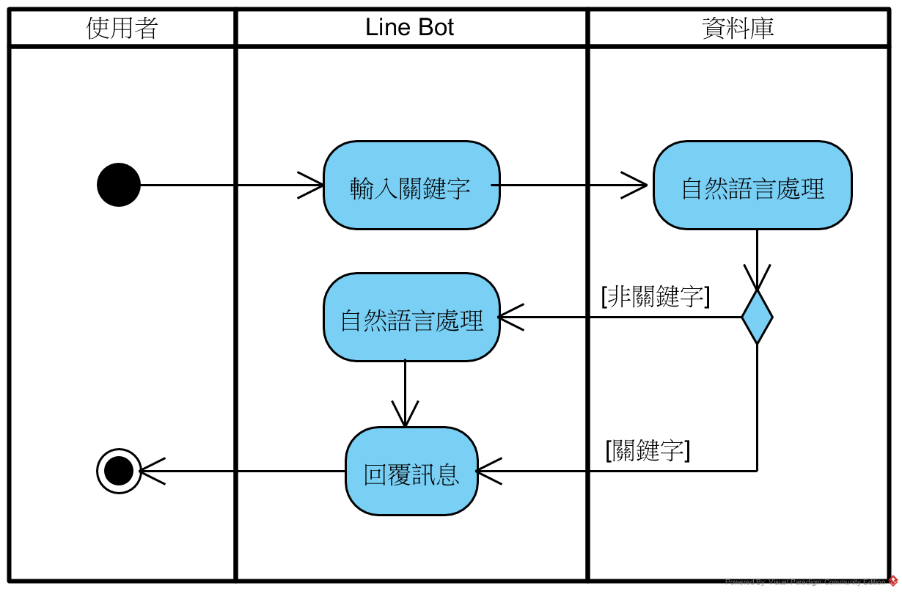
****

圖5-3-1 搜尋景點

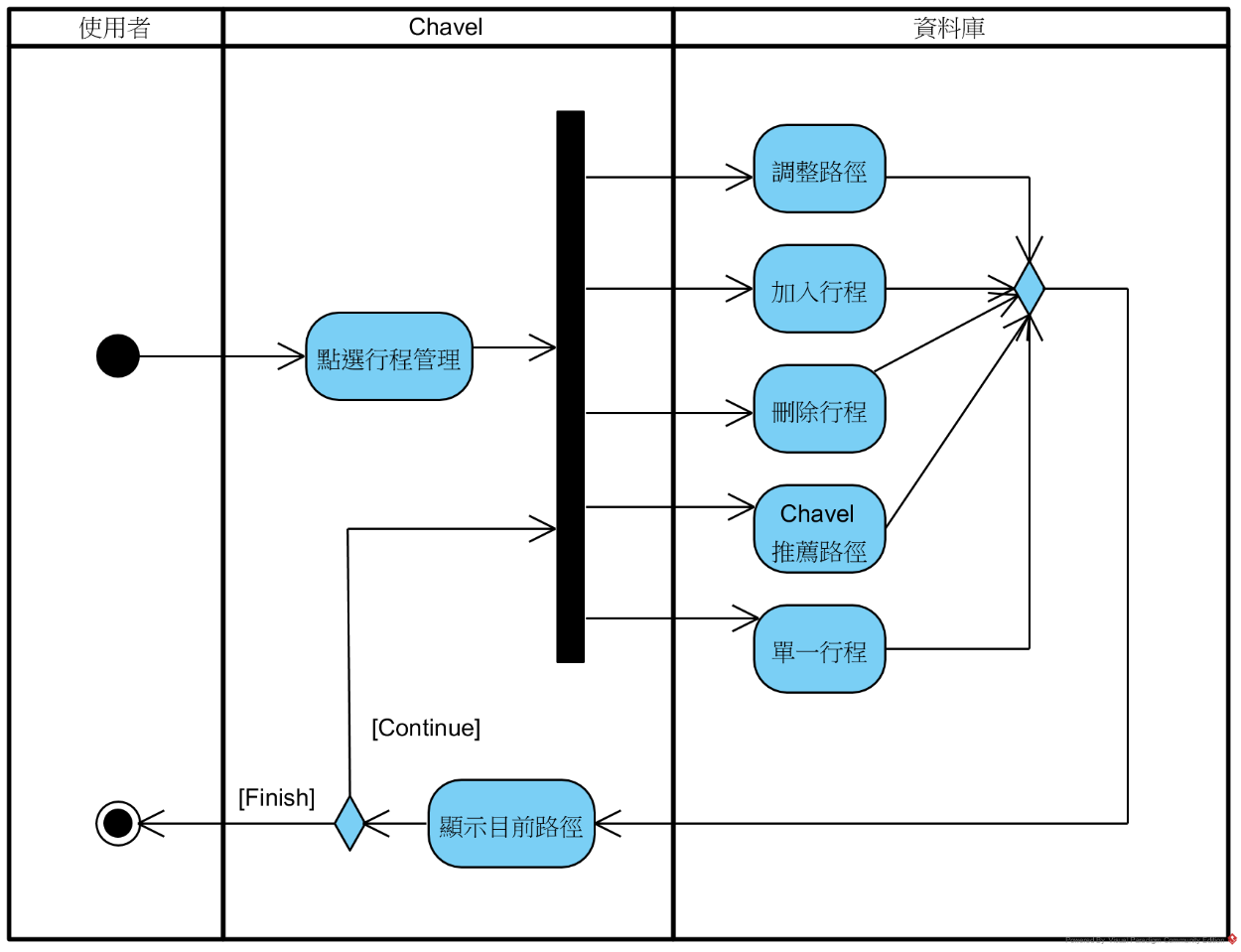
****

圖5-3-2 行程管理

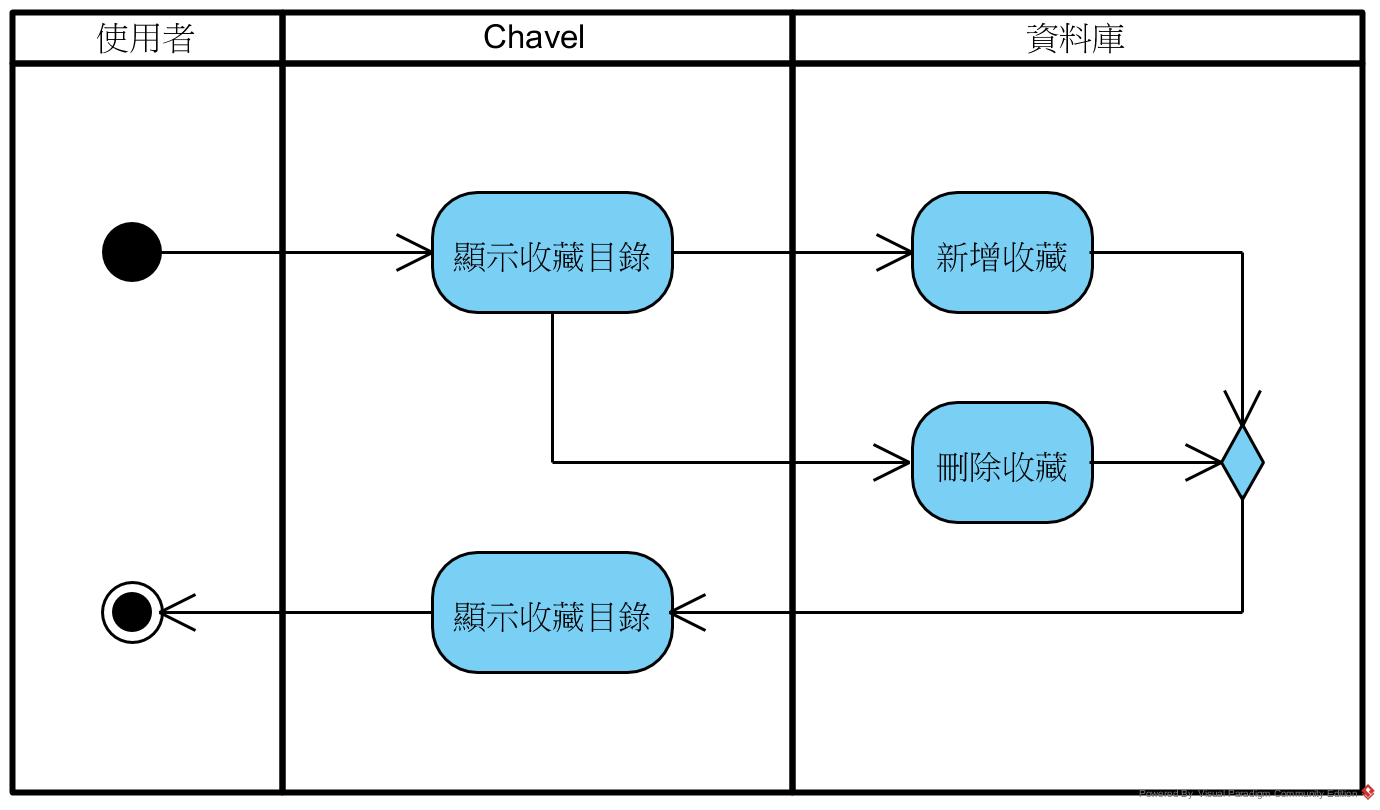
****

圖5-3-3 收藏管理

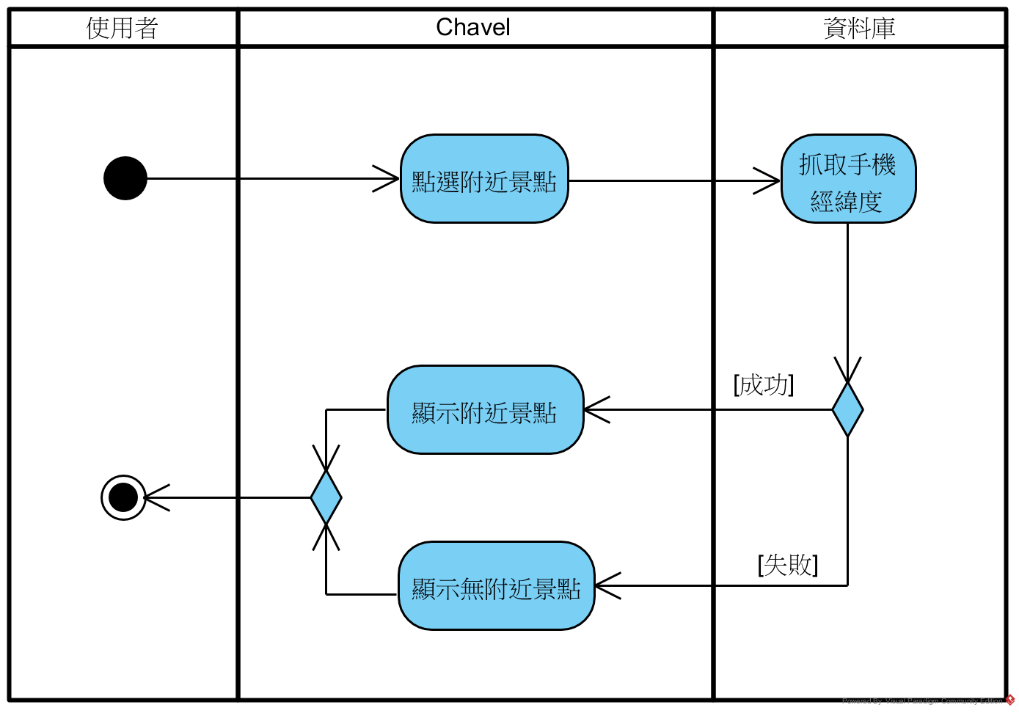
****

圖5-3-4 附近景點

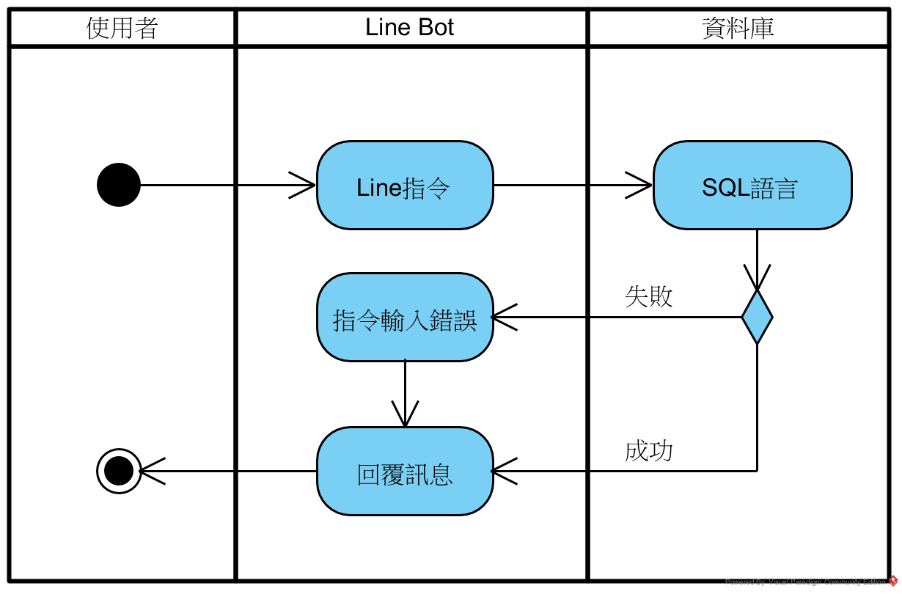
****

圖5-3-5 建立計畫表

**5-4分析類別圖**

****

圖5-4-1分析類別圖

**第六章設計模型**

**6-1循序圖**

****

圖6-1-1 搜尋景點

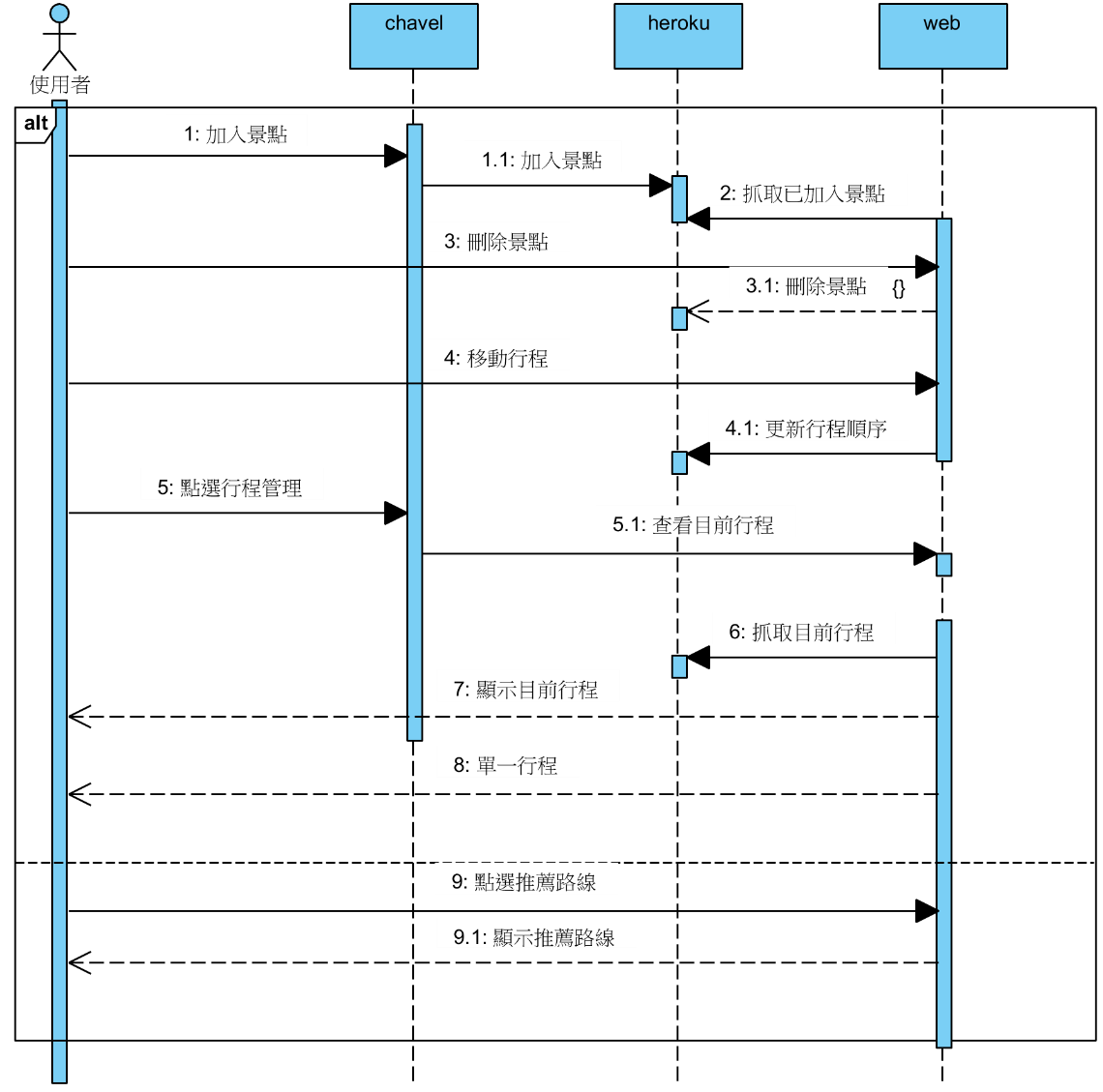


圖6-1-2 行程管理

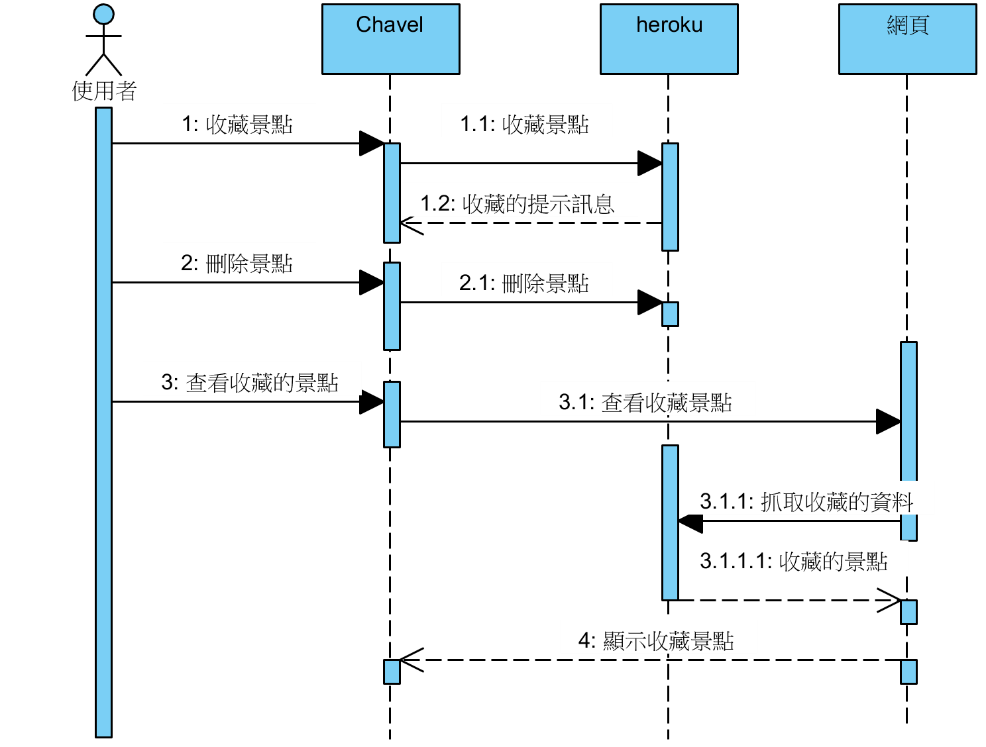
****

圖6-1-3 收藏景點

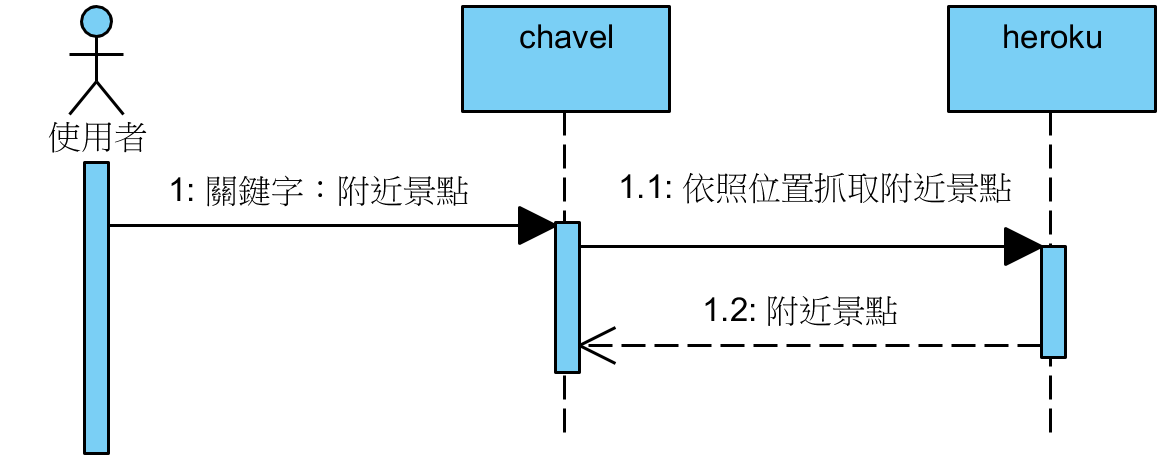
****

圖6-1-4 附近景點

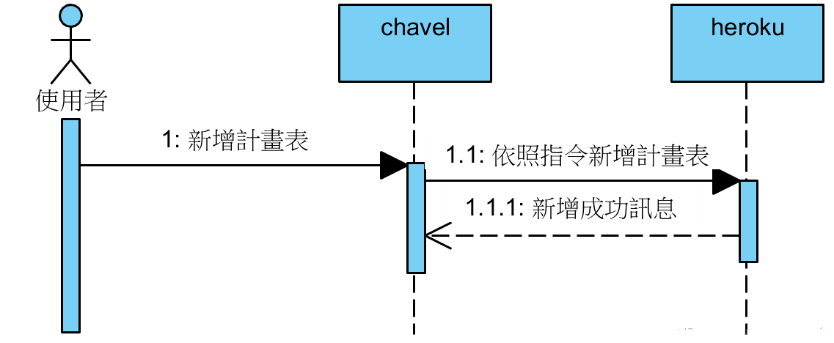
****

圖6-1-5 附近景點

**6-2設計類別圖**

****

圖6-2-1設計類別圖

**第七章 實作模型**

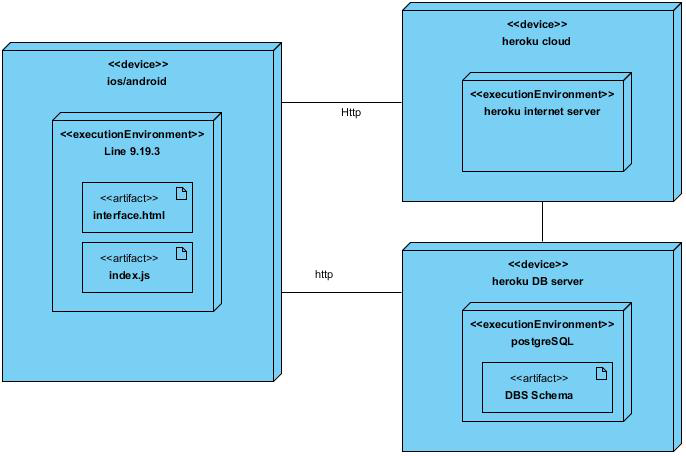
**7-1佈署圖**

圖7-1-1設計類別圖

**7-2套件圖**

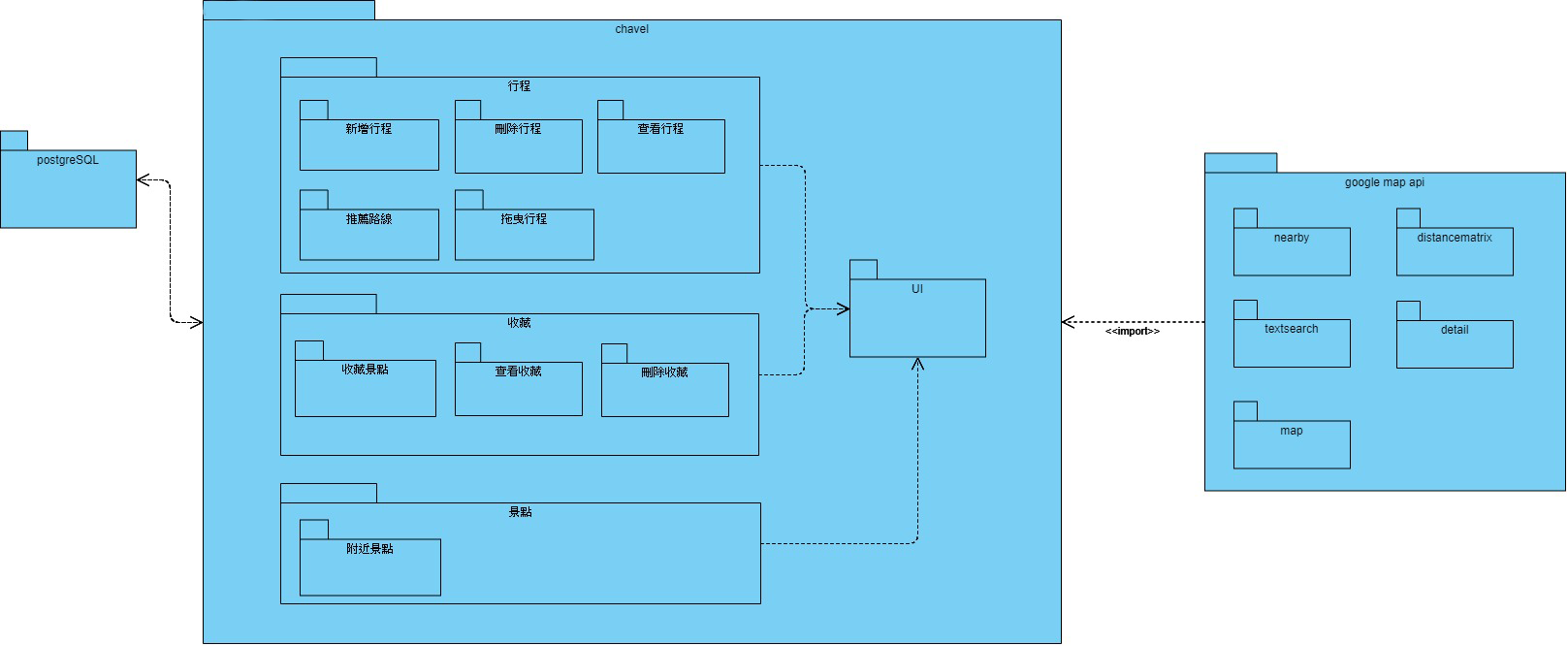
****

圖7-2-1套件圖

**7-3元件圖**

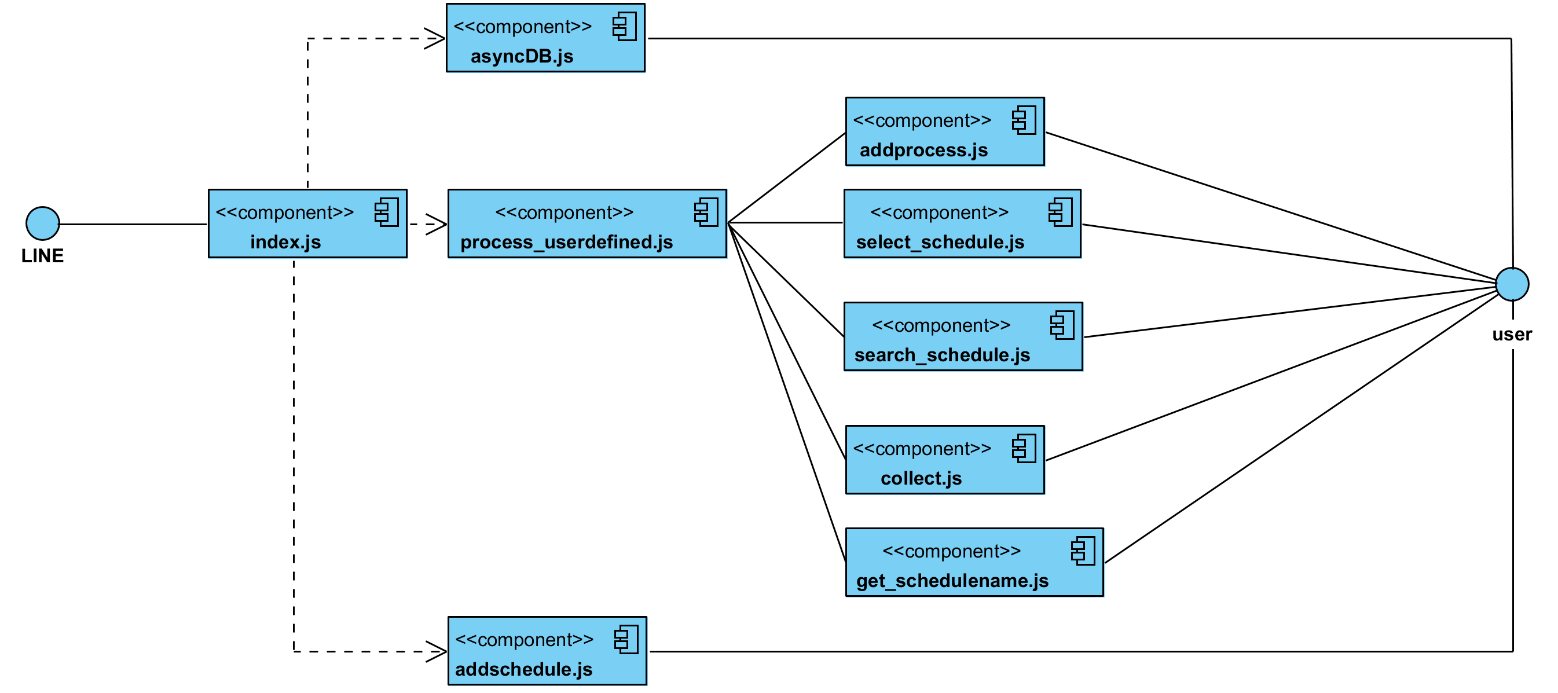
LINE端

圖7-3-1 LINE端元件圖

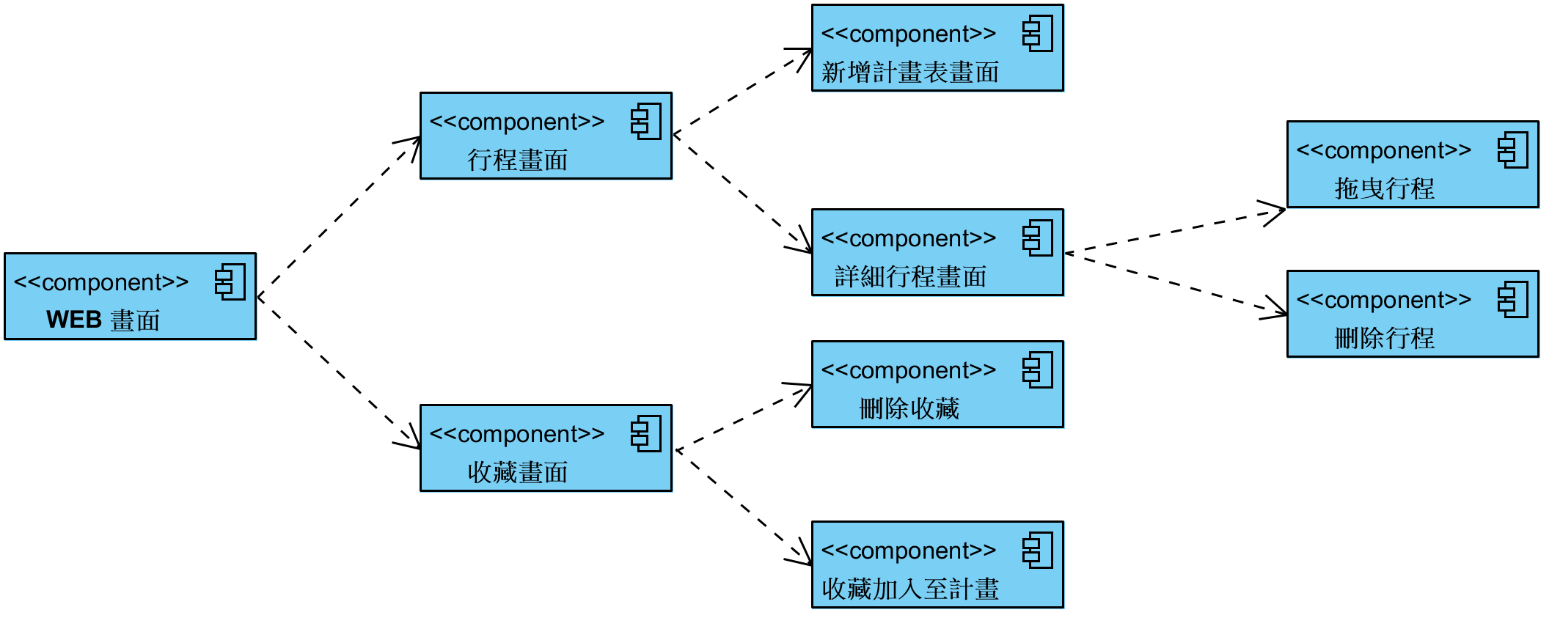
WEB端

圖7-3-1 WEB端元件圖

**7-4狀態機**

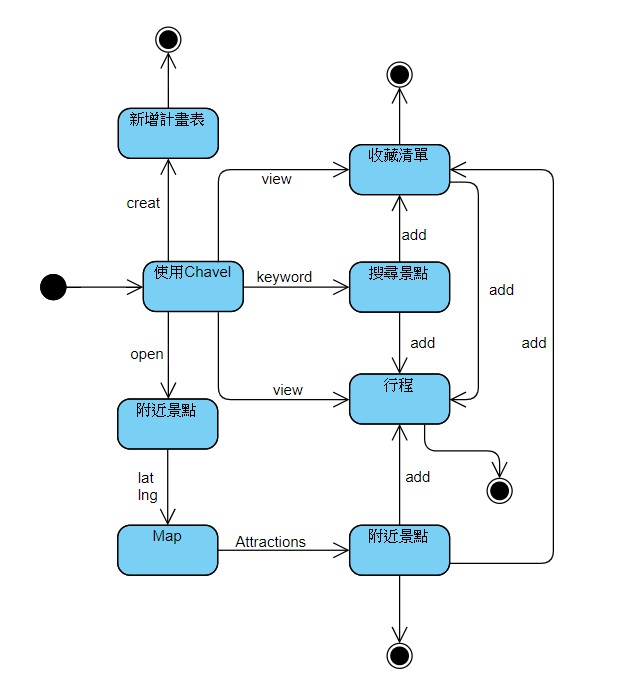
****

圖7-4-1 LINE端狀態機

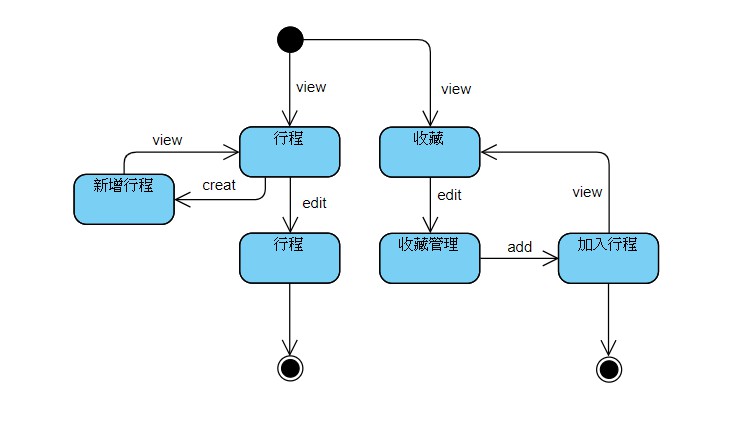
****

圖7-4-2 WEB端狀態機

**第八章 資料庫設計**

**8-1資料庫關聯表**

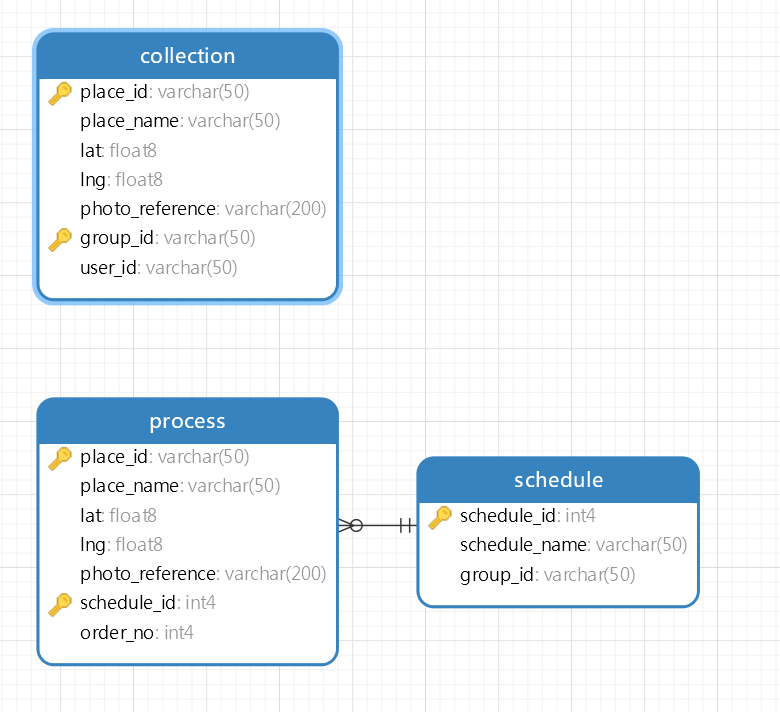
****

圖8-1-1 資料庫關聯圖

**8-2表格及其Meta data**

▼表 **錯誤! 所指定的樣式的文字不存在文件中。**‑1、資料表描述-T01計畫

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 計畫 | | 資料表編號 | | T01 | |
| 英文名稱 | schedule | | 主索引 | | schedule\_id | |
| 資料表陳述 | 儲存使用者的計畫表 | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 唯一性 | 允許空值 | 中文名稱 | 備註 |
| schedule\_id | int4 | 32 | 是 | 否 | 計畫編號 |  |
| schedule\_name | varchar | 50 | 否 | 是 | 計畫名稱 |  |
| group\_id | varchar | 50 | 否 | 是 | 群組編號 |  |
| user\_id | varchar | 50 | 否 | 是 | 使用者編號 |  |

▼表 **錯誤! 所指定的樣式的文字不存在文件中。**‑2、資料表描述-T02收藏

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 收藏 | | 資料表編號 | | T03 | |
| 英文名稱 | collection | | 主索引 | | place\_id  group\_id | |
| 資料表陳述 | 儲存使用者所收藏的景點資訊 | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 唯一性 | 允許空值 | 中文名稱 | 備註 |
| place\_id | varchar | 50 | 是 | 否 | 景點編號 |  |
| place \_name | varchar | 50 | 否 | 是 | 景點名稱 |  |
| lat | float8 | 53 | 否 | 是 | 經度 |  |
| lng | float8 | 53 | 否 | 是 | 緯度 |  |
| photo\_reference | varchar | 200 | 否 | 是 | 群組編號 |  |
| group\_id | varchar | 50 | 是 | 否 | 群組編號 |  |
| user\_id | varchar | 50 | 否 | 是 | 使用者編號 |  |

▼表 **錯誤! 所指定的樣式的文字不存在文件中。**‑3、資料表描述-T03行程

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 中文名稱 | 行程 | | 資料表編號 | | T02 | |
| 英文名稱 | process | | 主索引 | | place\_id  schedule\_id | |
| 資料表陳述 | 儲存使用者所規劃的行程 | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 唯一性 | 允許空值 | 中文名稱 | 備註 |
| place\_id | varchar | 50 | 是 | 否 | 景點編號 |  |
| place \_name | varchar | 50 | 否 | 是 | 景點名稱 |  |
| lat | float8 | 53 | 否 | 是 | 經度 |  |
| lng | float8 | 53 | 否 | 是 | 緯度 |  |
| photo\_reference | varchar | 50 | 否 | 是 | 群組編號 |  |
| schedule\_id | int4 | 32 | 是 | 否 | 計畫編號 |  |
| order\_no | int4 | 32 | 否 | 是 | 排序編號 |  |

**第九章 程式**

**9-1元件清單及其規格描述**

▼表9-1-1 LINE端程式清單

|  |  |  |
| --- | --- | --- |
| 編號 | 檔案名稱 | 功能說明 |
| 1 | index.js | 主程式(機器人接受回復、訊息處理) |
| 2 | addschedule.js | 新增計畫表 |
| 3 | async.js | 連結資料庫 |
| 4 | collect.js | 收藏景點 |
| 5 | get\_schedulename.js | 取得計畫表名稱 |
| 6 | process.js | 新增景點至計畫表 |
| 7 | search\_schedule.js | 取得一計畫表內所有景點資訊 |
| 8 | select\_max\_orderno.js | 取得計畫表內景點的最大排序編號 |
| 9 | select\_schedule.js | 取得群組內所有計畫表 |

▼表9-1-2 WEB端程式清單

|  |  |  |
| --- | --- | --- |
| 編號 | 檔案名稱 | 功能說明 |
| 1 | product.js | 取得相關頁面資料 |
| 2 | collection.js | 顯示收藏頁面 |
| 3 | collection\_add.js | 把相關參數帶到選擇加入至某個計畫表頁面 |
| 4 | product\_add.js | 執行新增計畫表 |
| 5 | addform.js | 顯示新增計畫表頁面 |
| 6 | reschedule.js | 執行推薦路徑 |
| 7 | schedule.js | 顯示所有計畫表 |
| 8 | schedule\_deleteprocess.js | 刪除行程 |
| 9 | schedule\_move.js | 拖曳行程並改變順序 |
| 10 | schedule\_query.js | 顯示詳細行程畫面 |
| 11 | select.js | 選擇將收藏的景點加入至某個計畫表裡 |
| 12 | main.js | 判斷各個頁面的button所需執行的function |

**9-2其他附屬之各種元件**

▼表9-2-1 LINE端程式規格描述

|  |  |
| --- | --- |
| 程式名稱 | index.js |
| 目的 | 主程式(機器人接受回復、訊息處理) |
| 程式碼 | |
| //----------------------------------------  // 載入必要的模組  //----------------------------------------  var linebot = require('linebot');  var express = require('express');  var request = require('request');  //增加引用函式  const process\_userdefined = require('./utility/process');  const addschedule = require('./utility/addschedule');  const collect = require('./utility/collect');  const select\_schedule= require('./utility/select\_schedule');  const search\_schedule= require('./utility/search\_schedule');  const get\_schedulename= require('./utility/get\_schedulename');  const select\_max\_orderno=require('./utility/select\_max\_orderno');  //----------------------------------------  // 填入自己在Line Developers的channel值  //----------------------------------------  var bot = linebot({      channelId: '1647290894',      channelSecret: '7585f07ee67ce3c9ca156ecb1677dcf3',      channelAccessToken: 'mZ/M5CZSHCyy1ogif29onWyvJuqhJaUuRX+eWAF4RW2T73sMMjFxMCQZ8cOvyCnskawvLbZbE3sAWNdIykAPKoHXbaSbBrYJH9LyaXnbgam9U8qX7u38VE50qD7m1jVrDsH/4QEEPFf8paPuHC+thgdB04t89/1O/w1cDnyilFU='  });  //========================================  // 機器人接受回覆的處理  //========================================  bot.on('postback', function(event) {    const getdata = event.postback.data;    const groupId = event.source.groupId;    const userId = event.source.userId;    const postback\_identifier=getdata.substr(getdata.length-1,1);    const real\_getdata=getdata.substr(0,getdata.length-1);    const detailurl="https://maps.googleapis.com/maps/api/place/details/json?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&placeid="+real\_getdata+"&language=zh-TW";    console.log('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'+getdata);    if(postback\_identifier=='P'){                  //如果是P就是加入行程       select\_schedule.selectSchedule(groupId).then(data=>{           if(data==-9){               console.log('執行錯誤');           }else if(data==-1){               console.log('無schedule');               event.reply('目前無計畫表，請先新建資料表!');           }else{                var flex={                    "type":"flex",                    "altText":"加入行程",                    "contents":{                        "type": "bubble",                        "body": {                            "type": "box",                            "layout": "horizontal",                            "contents": [                              {                                "type": "image",                                "url": "https://node-express-browser.herokuapp.com/imgs/add.png",                                "size": "xxs",                                "flex": 3,                                "align": "center",                                "offsetEnd": "10px"                              },                              {                                "type": "text",                                "text": "欲加入之計畫表",                                "flex":7,                                "weight": "bold",                                "size": "xl"                            }                            ]                        },                        "footer": {                            "type": "box",                            "layout": "vertical",                            "spacing": "sm",                            "contents": [                              ],                            "flex": 0                        }                    }                }                for(i=0;i<10 && i<data.rows.length;i++){                    var action=  {                        "type": "button",                        "style": "link",                        "height": "sm",                        "action": {                            "type": "postback",                            "label": "",                            "data": ""                        }                    }                    action.action.label='加入'+data.rows[i].schedule\_name;                    action.action.data=real\_getdata+data.rows[i].schedule\_id+'S';   //傳到POSTBACK讓事件以'S'識別要做選擇SCHEDULE加入景點                    flex.contents.footer.contents.push(action);                }                event.reply(flex);           }       })      }else if(postback\_identifier=='L'){          search\_schedule.searchSchedule(real\_getdata).then(data =>{          console.log(data);          if(data == -1){              event.reply('目前計畫表裡沒有行程');          }else{              var message=({              "type": "flex",              "altText": "查看詳細行程",              "contents":              {                "type": "bubble",                "body": {                  "type": "box",                  "layout": "vertical",                  "contents": [                    {                      "type": "image",                      "url": "https://images.pexels.com/photos/693490/pexels-photo-693490.jpeg?cs=srgb&dl=adventure-aircraft-balloons-693490.jpg&fm=jpg",                      "size": "full",                      "aspectMode": "cover",                      "aspectRatio": "1:1",                      "gravity": "center"                    },                    {                      "type": "image",                      "url": "https://scdn.line-apps.com/n/channel\_devcenter/img/flexsnapshot/clip/clip15.png",                      "position": "absolute",                      "aspectMode": "fit",                      "aspectRatio": "1:1",                      "offsetTop": "0px",                      "offsetBottom": "0px",                      "offsetStart": "0px",                      "offsetEnd": "0px",                      "size": "full"                    },                    {                      "type": "box",                      "layout": "horizontal",                      "contents": [                        {                          "type": "box",                          "layout": "vertical",                          "contents": [                            title={                              "type": "box",                              "layout": "vertical",                              "contents": [                                {                                  "type": "text",                                  "text": "台北一日遊",                                  "size": "xxl",                                  "margin": "none",                                  "weight": "bold",                                  "style": "normal",                                  "decoration": "none"                                }                              ],                              "position": "relative"                            }                          ]                        }                      ],                      "position": "relative",                      "offsetBottom": "0px",                      "offsetStart": "0px",                      "offsetEnd": "0px",                      "paddingAll": "15px"                    },                    {                      "type": "box",                      "layout": "vertical",                      "contents": [                        total\_address={                          "type": "box",                          "layout": "vertical",                          "contents": [                            /\*                            address={                              "type": "box",                              "layout": "baseline",                              "contents": [                                /\*                                icon={                                  "type": "icon",                                  "url": "https://node-express-browser.herokuapp.com/imgs/placeholder.png",                                  "margin": "xs",                                  "size": "xxl"                                },                                add={                                  "type": "text",                                  "text": "侯硐貓村",                                  "size": "xl",                                  "margin": "lg",                                  "style": "normal",                                  "weight": "bold"                                }                                ],                              "paddingAll": "3px"                            }                            \*/                          ]                        }                      ],                      "paddingStart": "18px"                    }                  ],                  "paddingAll": "0px",                  "backgroundColor": "#FFFFE8"                }                ,                "footer": {                  "type": "box",                  "layout": "vertical",                  "contents": [                    {                      "type": "spacer",                      "size": "xxl"                    },                    uri={                      "type": "button",                      "color": "#335219",                      "action": {                        "type": "uri",                        "label": "調整行程",                        "uri": "https://linecorp.com"                      },                      "gravity": "center",                      "style": "primary"                    }                  ],                  "backgroundColor": "#FFFFE8"                }                }            });              title.contents[0].text=data[0].schedule\_name;            console.log(data[0].schedule\_id);            const url="http://node-express-browser.herokuapp.com/schedule/query?schedule\_id="+data[0].schedule\_id;            uri.action.uri=url;            for(i=0;i<data.length;i++){                  address={                  "type": "box",                  "layout": "baseline",                  "contents": [],                  "paddingAll": "3px"                  }                    icon={                    "type": "icon",                    "url": "https://node-express-browser.herokuapp.com/imgs/placeholder.png",                    "margin": "xs",                    "size": "xxl"                  },                  add={                    "type": "text",                    "text": "侯硐貓村",                    "size": "lg",                    "margin": "md",                    "style": "normal",                    "weight": "bold"                  }              add.text=(i+1)+'.'+data[i].place\_name              total\_address.contents.push(address);              address.contents.push(icon);              address.contents.push(add);            }                event.reply(message);            //flex message              }        })      }else if(postback\_identifier=='C'){                 //如果是C就是加入收藏          //placedetail google api url          request(detailurl,function(error,response,body){               if(!error&&response.statusCode==200){                   var jsobject=JSON.parse(body);                   var subobject=jsobject.result;                   var name=subobject.name;                   var lat=subobject.geometry.location.lat;                   var lng=subobject.geometry.location.lng;                   var photo\_reference=subobject.photos[0].photo\_reference;                     collect.addCollection(real\_getdata, name, lat, lng, photo\_reference, groupId, userId).then(data=>{                      if(data == -9){                         console.log('執行錯誤');                         event.reply('已加入收藏');                      }else{                         console.log('已增加' + data + '筆記錄');                         event.reply('收藏 '+name+' 成功');                      }                   })               }          });      }else if(postback\_identifier=='N'){          //如果是N就顯示電話號碼             //placedetail google api url          request(detailurl,function(error,response,body){              if(!error&&response.statusCode==200){                  var jsobject=JSON.parse(body);                  var subobject=jsobject.result;                  var isPhoneElement=subobject.hasOwnProperty("formatted\_phone\_number");                  var phone=subobject.name+":\n";                  //判斷有無電話                  if(isPhoneElement){                      phone+=subobject.formatted\_phone\_number;                  }else{                      phone+='無';                  }                  event.reply(phone);              }           });      }else if(postback\_identifier=='O'){        request(detailurl,function(error,response,body){          if(!error&&response.statusCode==200){              var jsobject=JSON.parse(body);              var subobject=jsobject.result;              var name=subobject.name;              var isOpenElement=subobject.hasOwnProperty("opening\_hours");              var opentime=name+'的營業時間:\n';              if(isOpenElement){            //判斷有無營業時間                  for(i=0;i<7;i++){                     opentime+=subobject.opening\_hours.weekday\_text[i]+'\n';                  }              }else{                  opentime+="無營業時間";              }              event.reply(opentime);          }        });      }else if(postback\_identifier=='S'){ //指定加入到某個schedule        var schname;        var next\_order;        get\_schedulename.get\_schedulename(real\_getdata.substr(27,real\_getdata.length-27)).then(data=>{            if(data == -9){                console.log('執行錯誤');            }else{                schname=data.rows[0].schedule\_name;            }        })        select\_max\_orderno.select\_max\_orderno(real\_getdata.substr(27,real\_getdata.length-27)).then(data=>{          if(data==-9){              console.log('執行錯誤');          }else{              if(data.rows[0].max==null){                  next\_order=1;              }else{                  next\_order=data.rows[0].max+1;              }          }      })        const detailurlS="https://maps.googleapis.com/maps/api/place/details/json?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&placeid="+real\_getdata.substr(0,27)+"&language=zh-TW";        request(detailurlS,function(error,response,body){             if(!error&&response.statusCode==200){                   var jsobject=JSON.parse(body);                 var subobject=jsobject.result;                 var name=subobject.name;                 var lat=subobject.geometry.location.lat;                 var lng=subobject.geometry.location.lng;                 var photo\_reference=subobject.photos[0].photo\_reference;                   process\_userdefined.addProcess(real\_getdata.substr(0,27), name, lat, lng, photo\_reference, real\_getdata.substr(27,real\_getdata.length-27),next\_order).then(data=>{                    if(data == -9){                       console.log('執行錯誤');                       event.reply('計畫表已有此景點');                    }else{                       console.log('已增加' + data + '筆記錄');                       event.reply(name+' 成功加入 '+schname);                    }                 })             }        });      }  });  //========================================  //========================================  // 機器人接受訊息的處理  //========================================  bot.on('message', function(event) {            event.source.profile().then(                //呼叫API新增計畫表                function (profile) {                      //取得使用者資料                  const groupId=profile.groupId;                    const no = event.message.text; //使用者傳來的訊息                    if (event.message.type==('text')){                      const schedule=no.substr(0,3); //取得指令計畫表                      const schedule\_name=no.substr(4,no.length-1); //取得計畫表名稱                      const search=no.substr(0,1);   //取得指令搜尋                      const search\_name=no.substr(2,no.length-1);   //取得計畫表名稱                      console.log(search);                      console.log(search\_name);                        if(schedule == '計畫表'){  //建立計畫表                          addschedule.addschedule(schedule\_name,groupId).then(data => {                                if(data == -9){                                    event.reply('執行錯誤');                                }else {                                  event.reply([                                      {                                          "type": "text",                                          "text": '新增'+schedule\_name+'成功'                                      }                                  ]);                                  }                          })                      }else if(no == '建立計畫表'){ //計畫表指令教學                        event.reply([                          {                              "type": "text",                              "text": '建立計畫表教學'                          },                          {                              "type": "text",                              "text": '例如：計畫表 淡水一日遊'                          },                          {                              "type": "text",                              "text": '記得計畫表後要空一格^^'                          },                          {                              "type": "sticker",                              "packageId": "1",   //packageId可輸入1~4                              "stickerId": "2"                          }                      ]);                      }else if(no == '幫助'){                        event.reply([                          {                             "type":"text",                             "text": '功能說明：\n1. 建立計畫表：建立新的計畫表(指令教學)\n2. 搜尋景點：如何尋找景點(指令教學)\n3. 附近景點：傳送目前位置到LINE，機器人會顯示附近10筆的景點\n4. 單一行程：查看目前某個計畫表的詳細行程\n5. 收藏：查看目前收藏哪些景點\n6. 行程：查看目前有哪些計畫表，也可以在裡面改變行程順序。'                          }                        ])                      }else if(no == '搜尋景點'){                        event.reply([                          {                              "type": "text",                              "text": '搜尋景點教學'                          },                          {                              "type": "text",                              "text": '(1)景點名稱\n 例如：中正紀念堂、臺北101\n(2)指定區域內的店家\n例如：中和區牛肉麵\n(3)輸入一些平常用語\n例如：去淡水玩、去爬山'                          },                          {                            "type": "text",                            "text": 'P.S 若是找不到，可以直接打完整名稱'                        },                          {                              "type": "sticker",                              "packageId": "1",   //packageId可輸入1~4                              "stickerId": "2"                          }                        ]);                      }else if(no == '詳細行程'){                        select\_schedule.selectSchedule(groupId).then(data=>{                          if(data==-9){                              console.log('執行錯誤');                          }else if(data==-1){                              console.log('無schedule');                              event.reply('目前無計畫表，請先新建資料表!');                          }else{                               var flex={                                   "type":"flex",                                   "altText":"選擇計畫表",                                   "contents":{                                       "type": "bubble",                                       "body": {                                           "type": "box",                                           "layout": "horizontal",                                           "contents": [                                            {                                              "type": "image",                                              "url": "https://node-express-browser.herokuapp.com/imgs/search.png",                                              "size": "xxs",                                              "flex": 2,                                              "align": "start",                                              //"offsetEnd": "10px"                                            },                                           {                                               "type": "text",                                               "text": "欲查看之計畫表",                                               "flex":8,                                               "weight": "bold",                                               "size": "xl"                                           }                                           ]                                       },                                       "footer": {                                           "type": "box",                                           "layout": "vertical",                                           "spacing": "sm",                                           "contents": [                                             ],                                           "flex": 0                                       }                                   }                               }                               for(i=0;i<10 && i<data.rows.length;i++){                                   var action=  {                                       "type": "button",                                       "style": "link",                                       "height": "sm",                                       "action": {                                           "type": "postback",                                           "label": "",                                           "data": ""                                       }                                   }                                   action.action.label='查看'+data.rows[i].schedule\_name;                                   action.action.data=data.rows[i].schedule\_id+'L';   //傳到POSTBACK讓事件以'L'識別要做選擇SCHEDULE查看詳細行程                                   flex.contents.footer.contents.push(action);                               }                               event.reply(flex);                          }                        })                        }else if(no == '選單'){  //泡泡選單(quick reply)                          event.reply([                              {                              "type": "text",                              "text": "選擇你需要的功能",                              "quickReply": {                                "items": [                                  {                                    "type": "action", //建立計畫表                                    "imageUrl":"https://node-express-browser.herokuapp.com/imgs/schedule.png",                                    "action": {                                      "type": "message",                                      "label": "建立計畫表",                                      "text": "建立計畫表"                                    }                                  },                                  {                                    "type": "action", //搜尋景點                                    "imageUrl": "https://node-express-browser.herokuapp.com/imgs/location.png",                                    "action": {                                      "type": "message",                                      "label": "搜尋景點",                                      "text": "搜尋景點"                                    }                                  },                                  {                                    "type": "action", //附近景點                                    "imageUrl":"https://node-express-browser.herokuapp.com/imgs/attraction.png",                                    "action": {                                      "type": "location",                                      "label": "附近景點",                                      "text": "附近景點"                                    }                                  },                                  {                                    "type": "action", //詳細行程                                    "imageUrl":"https://node-express-browser.herokuapp.com/imgs/view\_detail\_big.png",                                    "action": {                                      "type": "message",                                      "label": "單一計畫表",                                      "text": "詳細行程"                                    }                                  },                                  {                                    "type": "action", //收藏                                    "imageUrl": "https://node-express-browser.herokuapp.com/imgs/collect.png",                                    "action": {                                      "type": "message",                                      "label": "收藏",                                      "text": "看收藏"                                    }                                  },                                  {                                    "type": "action", //行程                                    "imageUrl":"https://node-express-browser.herokuapp.com/imgs/process.png",                                    "action": {                                      "type": "message",                                      "label": "行程",                                      "text": "看行程"                                    }                                  },                                  {                                    "type": "action", //行程                                    "imageUrl":"https://node-express-browser.herokuapp.com/imgs/help.png",                                    "action": {                                      "type": "message",                                      "label": "幫助",                                      "text": "幫助"                                    }                                  }                                ]                              }                            }                          ])                      }else if(no == '看收藏'){  //開啟收藏網站                            const url ='http://node-express-browser.herokuapp.com/collection?group\_id='+groupId                          event.reply({                            "type": "flex",                              "altText": "查看收藏",                              "contents":                              {                                "type": "bubble",                                "body": {                                  "type": "box",                                  "layout": "horizontal",                                  "contents": [                                    {                                      "type": "image",                                      "url": "https://node-express-browser.herokuapp.com/imgs/collect.png",                                      "size": "xxs",                                      "flex": 2,                                      "align": "end",                                      "offsetEnd": "10px"                                    },                                    {                                      "type": "text",                                      "text": "查看我的收藏",                                      "weight": "bold",                                      "size": "xl",                                      "align": "start",                                      "flex": 5                                    }                                  ]                                },                                "footer": {                                  "type": "box",                                  "layout": "vertical",                                  "spacing": "sm",                                  "contents": [                                    {                                      "type": "button",                                      "style": "link",                                      "height": "sm",                                      "action": {                                        "type": "uri",                                        "label": "開啟收藏",                                        "uri": ""+url                                      }                                    },                                    {                                      "type": "spacer",                                      "size": "sm"                                    }                                  ],                                  "flex": 0                                }                              }                          });                      }else if(no == '看行程'){  //開啟行程網站                          const url ='http://node-express-browser.herokuapp.com/schedule?group\_id='+groupId                          event.reply({                            "type": "flex",                              "altText": "查看行程",                              "contents":                              {                                "type": "bubble",                                "body": {                                  "type": "box",                                  "layout": "horizontal",                                  "contents": [                                    {                                      "type": "image",                                      "url": "https://node-express-browser.herokuapp.com/imgs/planning.png",                                      "size": "xxs",                                      "flex": 3,                                      "align": "center",                                      "offsetEnd": "10px"                                    },                                    {                                      "type": "text",                                      "text": "查看我的行程",                                      "weight": "bold",                                      "size": "xl",                                      "align": "start",                                      "flex": 7                                    }                                  ]                                },                                "footer": {                                  "type": "box",                                  "layout": "vertical",                                  "spacing": "sm",                                  "contents": [                                    {                                      "type": "button",                                      "style": "link",                                      "height": "sm",                                      "action": {                                        "type": "uri",                                        "label": "開啟行程",                                        "uri": ""+url                                      }                                    },                                    {                                      "type": "spacer",                                      "size": "sm"                                    }                                  ],                                  "flex": 0                                }                              }                          });                      }else{                        const searchurl="https://maps.googleapis.com/maps/api/place/textsearch/json?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&query="+no+"&region=tw&language=zh-TW"                        //url內有中文所以要編碼                        const encodeurl=encodeURI(searchurl);                        request(encodeurl,function(error,response,body){                            if(!error && response.statusCode==200){                                var jsobject=JSON.parse(body);  //將接收到的data轉為json物件                                var flex={                                    "type":"flex",                                    "altText":"關鍵字搜尋景點",                                    "contents":{                                        "type":"carousel",                                        "contents":[]                                    }                                }                                var loopnum;                                if(jsobject.results.length<=10){                                    loopnum=jsobject.results.length                                }else{                                    loopnum=10;                                }                                for(i=0;i<loopnum;i++){                                 var subobject=jsobject.results[i];                                   var name=subobject.name;                                 var rating;                                 var formatted\_address=subobject.formatted\_address;                                 var isAddressElement=subobject.hasOwnProperty("formatted\_address");//資料有無地址                                 var photo\_reference;                                 var place\_id=subobject.place\_id;                                 var att\_lat=subobject.geometry.location.lat;                                 var att\_lng=subobject.geometry.location.lng;                                 var isOpenElement=subobject.hasOwnProperty("opening\_hours"); //資料有無opening\_hours元素                                 var isOpen;                                 var isPhotoElement=subobject.hasOwnProperty("photos");  //資料有無photos元素                                 var isRatingElement=subobject.hasOwnProperty("rating");  //資料有無rating元素                                      //判斷有無地址                                if(isAddressElement){                                  formatted\_address=subobject.formatted\_address;                                }else{                                  formatted\_address="-"                                }                                   //判斷是否有評分                                 if(isRatingElement){                                    rating=subobject.rating;                                 }else{                                     rating="無評分";                                 }                                 //判斷是否有照片                                 if(isPhotoElement){                                    photo\_reference=subobject.photos[0].photo\_reference;                                 }else{                                    photo\_reference=null;                                 }                                   //判斷是否營業的條件式                                 if(isOpenElement==false){                                    isOpen='-';  //沒有就 傳"-"                                 }else if(isOpenElement==true){                                     var open\_now=subobject.opening\_hours.open\_now;                                       if(open\_now==true){                                       isOpen='營業中';                                     }else if(open\_now==null){                                       isOpen='-'   //有opening\_hours元素 但其open\_now元素為空值                                     }else if(open\_now==false){                                       isOpen='休息中' ;                                     }                                 }                                     //google map api url                                 const search\_map="https://www.google.com/maps/search/?api=1&query="+att\_lat+","+att\_lng+"&query\_place\_id="+place\_id                                   var content={                                    "type": "bubble",                                    "hero": {                                        "type": "image",                                        "url": 'https://maps.googleapis.com/maps/api/place/photo?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&photoreference='+photo\_reference+'&maxwidth=1000&height=1000',                                        "size": "full",                                        "aspectRatio": "20:13",                                        "aspectMode": "cover",                                        "action": {                                        "type": "uri",                                        "uri": ""                                        }                                    },                                    "body": {                                        "type": "box",                                        "layout": "vertical",                                        "contents": [                                        {                                            "type": "text",                                            "text": "",                                            "weight": "bold",                                            "size": "xl"                                        },                                        {                                            "type": "box",                                            "layout": "baseline",                                            "margin": "md",                                            "contents": [                                              ]                                        },                                        {                                            "type": "box",                                            "layout": "vertical",                                            "margin": "lg",                                            "spacing": "sm",                                            "contents": [                                              ]                                        }                                        ]                                    },                                    "footer": {                                        "type": "box",                                        "layout": "vertical",                                        "spacing": "sm",                                        "contents": [                                          ],                                        "flex": 0                                    }                                }                                    var address={                                                "type": "box",                                                "layout": "baseline",                                                "spacing": "sm",                                                "contents": [                                                {                                                    "type": "text",                                                    "text": "地址",                                                    "color": "#aaaaaa",                                                    "size": "sm",                                                    "flex": 1                                                },                                                {                                                    "type": "text",                                                    "text": "testaddress",                                                    "wrap": true,                                                    "color": "#666666",                                                    "size": "sm",                                                    "flex": 5                                                }                                                ]                                 }                                 address.contents[1].text=''+formatted\_address;                                 content.body.contents[2].contents.push(address);                                   var open=             {                                                "type": "box",                                                "layout": "baseline",                                                "spacing": "sm",                                                "contents": [                                                {                                                    "type": "text",                                                    "text": "狀態",                                                    "color": "#aaaaaa",                                                    "size": "sm",                                                    "flex": 1                                                },                                                {                                                    "type": "text",                                                    "text": "testisopen",                                                    "wrap": true,                                                    "color": "#666666",                                                    "size": "sm",                                                    "flex": 5                                                }                                                ]                                     }                                  open.contents[1].text=isOpen;                                  content.body.contents[2].contents.push(open);                                      var goldstar={                                                "type": "icon",                                                "size": "sm",                                                "url": "https://scdn.line-apps.com/n/channel\_devcenter/img/fx/review\_gold\_star\_28.png"                                            }                                  var graystar={                                                "type": "icon",                                                "size": "sm",                                                "url": "https://scdn.line-apps.com/n/channel\_devcenter/img/fx/review\_gray\_star\_28.png"                                            }                                  var halfstar={                                                "type":"icon",                                                "size":"sm",                                                "url":"https://node-express-browser.herokuapp.com/imgs/half-star.png"                                            }                                              if(rating=="無評分"){                                              for(j=1;j<=5;j++){                                                content.body.contents[1].contents.push(graystar);                                              }                                            }else{                                              for(p=1;p<=rating;p++){                                                 content.body.contents[1].contents.push(goldstar);                                              }                                              if(rating\*10-Math.floor(rating)\*10>=5){                                                  content.body.contents[1].contents.push(halfstar);                                                  for(k=1;k<=(5-Math.floor(rating)-1);k++){                                                      content.body.contents[1].contents.push(graystar);                                                  }                                              }else{                                                for(k=1;k<=(5-Math.floor(rating));k++){                                                    content.body.contents[1].contents.push(graystar);                                                }                                                }                                            }                                      var ratingpoint= {                                                "type": "text",                                                "text": "",                                                "size": "sm",                                                "color": "#999999",                                                "margin": "md",                                                "flex": 0                                            }                                   ratingpoint.text=rating.toString();                                   content.body.contents[1].contents.push(ratingpoint);                                          var action\_process={                                            "type": "button",                                            "style": "link",                                            "height": "sm",                                            "action": {                                                 "type": "postback",                                                 "label": "加入行程",                                                 "data": ""                                            }                                    }                                  var action\_collect={                                            "type": "button",                                            "style": "link",                                            "height": "sm",                                            "action": {                                                 "type": "postback",                                                 "label": "收藏",                                                 "data": ""                                            }                                    }                                  var action\_phone={                                            "type": "button",                                            "style": "link",                                            "height": "sm",                                            "action": {                                                 "type": "postback",                                                 "label": "電話",                                                 "data": ""                                            }                                    }                                  var action\_opentime={                                    "type": "button",                                    "style": "link",                                    "height": "sm",                                    "action": {                                         "type": "postback",                                         "label": "顯示營業時間",                                         "data": ""                                  }                              }                                    action\_process.action.data=place\_id+'P';    //傳到POSTBACK讓事件以'P'識別要做http request加入行程                                  action\_collect.action.data=place\_id+'C';    //傳到POSTBACK讓事件以'C'識別要做http request加入收藏                                  action\_phone.action.data=place\_id+'N';      //傳到POSTBACK讓事件以'N'識別要做http request顯示電話                                  action\_opentime.action.data=place\_id+'O';    //傳到POSTBACK讓事件以'O'識別要做http request顯示電話                                  content.footer.contents.push(action\_process);                                  content.footer.contents.push(action\_collect);                                  content.footer.contents.push(action\_phone);                                  content.footer.contents.push(action\_opentime);                                  content.hero.action.uri=search\_map;                                  content.body.contents[0].text=name;                                  flex.contents.contents.push(content);                                  }                                  event.reply(flex);                          }                  })                      }                    }else if (event.message.type=="location"){                      var lat=event.message.latitude;    //使用者傳的經緯度                      var lnt=event.message.longitude;                      const address=event.message.address;  //使用者傳的位置                      //call place nearby google api 並將api key 經緯度等參數塞進去                      const nearbyurl='https://maps.googleapis.com/maps/api/place/nearbysearch/json?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&location='+lat+','+lnt+'&rankby=distance&keyword=attractions&language=zh-TW'                        request(nearbyurl,function(error,response,body){                          if(!error && response.statusCode==200){                              var jsobject=JSON.parse(body);  //將接收到的data轉為json物件                              var flex={                                  "type":"flex",                                  "altText":"關鍵字搜尋景點",                                  "contents":{                                      "type":"carousel",                                      "contents":[]                                  }                              }                              var loopnum;                              if(jsobject.results.length<=10){                                  loopnum=jsobject.results.length                              }else{                                  loopnum=10;                              }                              for(i=0;i<loopnum;i++){                               var subobject=jsobject.results[i];                                 var name=subobject.name;                               var rating;                               var formatted\_address=subobject.formatted\_address;                               var photo\_reference;                               var place\_id=subobject.place\_id;                               var vicinity=subobject.vicinity;                               var att\_lat=subobject.geometry.location.lat;                               var att\_lng=subobject.geometry.location.lng;                               var isOpenElement=subobject.hasOwnProperty("opening\_hours"); //資料有無opening\_hours元素                               var isOpen;                               var isPhotoElement=subobject.hasOwnProperty("photos");  //資料有無photos元素                               var isRatingElement=subobject.hasOwnProperty("rating");  //資料有無rating元素                                 //判斷是否有評分                               if(isRatingElement){                                  rating=subobject.rating;                               }else{                                   rating="無評分";                               }                               //判斷是否有照片                               if(isPhotoElement){                                  photo\_reference=subobject.photos[0].photo\_reference;                               }else{                                  photo\_reference=null;                               }                                 //判斷是否營業的條件式                               if(isOpenElement==false){                                  isOpen='-';  //沒有就 傳"-"                               }else if(isOpenElement==true){                                   var open\_now=subobject.opening\_hours.open\_now;                                     if(open\_now==true){                                     isOpen='營業中';                                   }else if(open\_now==null){                                     isOpen='-'   //有opening\_hours元素 但其open\_now元素為空值                                   }else if(open\_now==false){                                     isOpen='休息中' ;                                   }                               }                                   //google map api url                               const search\_map="https://www.google.com/maps/search/?api=1&query="+att\_lat+","+att\_lng+"&query\_place\_id="+place\_id                                 var content={                                  "type": "bubble",                                  "hero": {                                      "type": "image",                                      "url": 'https://maps.googleapis.com/maps/api/place/photo?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&photoreference='+photo\_reference+'&maxwidth=1000&height=1000',                                      "size": "full",                                      "aspectRatio": "20:13",                                      "aspectMode": "cover",                                      "action": {                                      "type": "uri",                                      "uri": ""                                      }                                  },                                  "body": {                                      "type": "box",                                      "layout": "vertical",                                      "contents": [                                      {                                          "type": "text",                                          "text": "",                                          "weight": "bold",                                          "size": "xl"                                      },                                      {                                          "type": "box",                                          "layout": "baseline",                                          "margin": "md",                                          "contents": [                                            ]                                      },                                      {                                          "type": "box",                                          "layout": "vertical",                                          "margin": "lg",                                          "spacing": "sm",                                          "contents": [                                            ]                                      }                                      ]                                  },                                  "footer": {                                      "type": "box",                                      "layout": "vertical",                                      "spacing": "sm",                                      "contents": [                                        ],                                      "flex": 0                                  }                              }                                  var address={                                              "type": "box",                                              "layout": "baseline",                                              "spacing": "sm",                                              "contents": [                                              {                                                  "type": "text",                                                  "text": "地址",                                                  "color": "#aaaaaa",                                                  "size": "sm",                                                  "flex": 1                                              },                                              {                                                  "type": "text",                                                  "text": "testaddress",                                                  "wrap": true,                                                  "color": "#666666",                                                  "size": "sm",                                                  "flex": 5                                              }                                              ]                               }                               address.contents[1].text=''+vicinity;                               content.body.contents[2].contents.push(address);                                 var open=             {                                              "type": "box",                                              "layout": "baseline",                                              "spacing": "sm",                                              "contents": [                                              {                                                  "type": "text",                                                  "text": "狀態",                                                  "color": "#aaaaaa",                                                  "size": "sm",                                                  "flex": 1                                              },                                              {                                                  "type": "text",                                                  "text": "testisopen",                                                  "wrap": true,                                                  "color": "#666666",                                                  "size": "sm",                                                  "flex": 5                                              }                                              ]                                   }                                open.contents[1].text=isOpen;                                content.body.contents[2].contents.push(open);                                    var goldstar={                                              "type": "icon",                                              "size": "sm",                                              "url": "https://scdn.line-apps.com/n/channel\_devcenter/img/fx/review\_gold\_star\_28.png"                                          }                                var graystar={                                              "type": "icon",                                              "size": "sm",                                              "url": "https://scdn.line-apps.com/n/channel\_devcenter/img/fx/review\_gray\_star\_28.png"                                          }                                var halfstar={                                              "type":"icon",                                              "size":"sm",                                              "url":"https://node-express-browser.herokuapp.com/imgs/half-star.png"                                          }                                            if(rating=="無評分"){                                            for(j=1;j<=5;j++){                                              content.body.contents[1].contents.push(graystar);                                            }                                          }else{                                            for(p=1;p<=rating;p++){                                               content.body.contents[1].contents.push(goldstar);                                            }                                            if(rating\*10-Math.floor(rating)\*10>=5){                                                content.body.contents[1].contents.push(halfstar);                                                for(k=1;k<=(5-Math.floor(rating)-1);k++){                                                    content.body.contents[1].contents.push(graystar);                                                }                                            }else{                                              for(k=1;k<=(5-Math.floor(rating));k++){                                                  content.body.contents[1].contents.push(graystar);                                              }                                              }                                          }                                    var ratingpoint= {                                              "type": "text",                                              "text": "",                                              "size": "sm",                                              "color": "#999999",                                              "margin": "md",                                              "flex": 0                                          }                                 ratingpoint.text=rating.toString();                                 content.body.contents[1].contents.push(ratingpoint);                                        var action\_process={                                          "type": "button",                                          "style": "link",                                          "height": "sm",                                          "action": {                                               "type": "postback",                                               "label": "加入行程",                                               "data": ""                                          }                                  }                                var action\_collect={                                          "type": "button",                                          "style": "link",                                          "height": "sm",                                          "action": {                                               "type": "postback",                                               "label": "收藏",                                               "data": ""                                          }                                  }                                  var action\_phone={                                          "type": "button",                                          "style": "link",                                          "height": "sm",                                          "action": {                                               "type": "postback",                                               "label": "電話",                                               "data": ""                                          }                                    }                                    var action\_opentime={                                    "type": "button",                                    "style": "link",                                    "height": "sm",                                    "action": {                                         "type": "postback",                                         "label": "顯示營業時間",                                         "data": ""                                    }                            }                                    action\_process.action.data=place\_id+'P';    //傳到POSTBACK讓事件以'P'識別要做http request加入行程                                action\_collect.action.data=place\_id+'C';    //傳到POSTBACK讓事件以'C'識別要做http request加入收藏                                action\_phone.action.data=place\_id+'N';      //傳到POSTBACK讓事件以'N'識別要做http request顯示電話                                action\_opentime.action.data=place\_id+'O';    //傳到POSTBACK讓事件以'O'識別要做http request顯示電話                                content.footer.contents.push(action\_process);                                content.footer.contents.push(action\_collect);                                content.footer.contents.push(action\_phone);                                content.footer.contents.push(action\_opentime);                                content.hero.action.uri=search\_map;                                content.body.contents[0].text=name;                                flex.contents.contents.push(content);                                }                                event.reply(flex);                                }                      })                    }                }          );  });  //========================================ㄡ  //----------------------------------------  // 建立一個網站應用程式app  // 如果連接根目錄, 交給機器人處理  //----------------------------------------  const app = express();  const linebotParser = bot.parser();  app.post('/', linebotParser);  //----------------------------------------  // 可直接取用檔案的資料夾  //----------------------------------------  app.use(express.static('public'));  //----------------------------------------  //處理Unhandle的問題  process.on('unhandledRejection', error => {    // Will print "unhandledRejection err is not defined"  });  new Promise((\_, reject) => reject(new Error('woops')))  .catch(new Function());  //----------------------------------------  //----------------------------------------  // 監聽3000埠號,  // 或是監聽Heroku設定的埠號  //----------------------------------------  var server = app.listen(process.env.PORT || 3000, function() {      const port = server.address().port;      console.log("正在監聽埠號:", port);  }); | |

▼表 9-2-2 addschedule.js

|  |  |  |
| --- | --- | --- |
| 程式名稱 | addschedule.js | |
| 目的 | 新增計畫表 | |
| 程式碼 | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 由ID查詢景點資料  //------------------------------------------  var addschedule = async function(schedule\_name,group\_id){      //存放結果      let result;      //讀取資料庫      await query('insert into schedule (schedule\_name,group\_id) values ($1, $2)',[schedule\_name,group\_id])          .then((data) => {              result = data.rowCount;  //新增資料數          }, (error) => {              result = -9;  //執行錯誤      });      //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {addschedule}; | | |
| 程式名稱 | | async.js |
| 目的 | | 連接資料庫 |
| 程式碼 | | |
| 'use strict';  //-----------------------  // 引用資料庫模組  //-----------------------  const {Client} = require('pg');  //-----------------------  // 自己的資料庫連結位址  //-----------------------  var pgConn = 'postgres://xbxbmmpqeyooks:a007d9568d9e2cf776bd60767287047863a70b21b7d1c53fa0cc945ce4221b3b@ec2-75-101-133-29.compute-1.amazonaws.com:5432/d5pkpvitslg453';  //產生可同步執行query物件的函式  function query(sql, value=null) {      return new Promise((resolve, reject) => {          //設定資料庫連線物件          var client = new Client({              connectionString: pgConn,              ssl: true          })          //連結資料庫          client.connect();          //回覆查詢結果          client.query(sql, value, (err, results) => {              if (err){                  reject(err);              }else{                  resolve(results);              }              //關閉連線              client.end();          });      });  }  //匯出  module.exports = query; | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 程式名稱 | | | | | | collect.js |
| 目的 | | | | | | 收藏景點 |
| 程式碼 | | | | | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var addCollection = async function(place\_id, place\_name, lat, lng, photo\_reference, group\_id, user\_id){      //存放結果      let result;      //讀取資料庫      await query('insert into collection (place\_id, place\_name, lat, lng, photo\_reference, group\_id, user\_id) values ($1, $2, $3, $4, $5, $6, $7)', [place\_id, place\_name, lat, lng, photo\_reference, group\_id, user\_id])          .then((data) => {              result = data.rowCount;  //新增資料數          }, (error) => {              result = -9;  //執行錯誤          });      //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {addCollection}; | | | | | | |
| 程式名稱 | | | | get\_schedulename.js | | |
| 目的 | | | | 取得計畫表名稱 | | |
| 程式碼 | | | | | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var get\_schedulename = async function(schedule\_id){      //存放結果      let result;      //讀取資料庫      await query('select schedule\_name from schedule where schedule\_id=$1 ', [schedule\_id])          .then((data) => {              if(data.rows.length>0){                  result=data;              }else{                  result=-1;//找不到資料              }          }, (error) => {              result = -9;  //執行錯誤          });      //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {get\_schedulename}; | | | | | | |
| 程式名稱 | | | | | process.js | |
| 目的 | | | | | 新增景點至計畫表 | |
| 程式碼 | | | | | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var addProcess = async function(place\_id, place\_name, lat, lng,photo\_reference, schedule\_id,order\_no){      //存放結果      let result;      //讀取資料庫      await query('insert into process (place\_id, place\_name, lat, lng,photo\_reference, schedule\_id,order\_no) values ($1,$2,$3,$4,$5,$6,$7)', [place\_id, place\_name, lat, lng,photo\_reference, schedule\_id,order\_no])          .then((data) => {              result = data.rowCount;  //新增資料數          }, (error) => {              result = -9;  //執行錯誤          });      //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {addProcess}; | | | | | | |
| 程式名稱 | | | search\_schedule.js | | | |
| 目的 | | | 取得某個計畫表內所有景點資訊 | | | |
| 程式碼 | | | | | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var searchSchedule = async function(schedule\_id){      //存放結果      let result;      //讀取資料庫      await query('select schedule\_name,place\_name,schedule.schedule\_id from process inner join schedule ON process.schedule\_id = schedule.schedule\_id where schedule.schedule\_id=$1 order by order\_no asc', [schedule\_id])          .then((data) => {              if(data.rows.length>0){                  result = data.rows;              }else{                  result=-1;//找不到資料              }          }, (error) => {              result = -9;  //執行錯誤          });        //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {searchSchedule}; | | | | | | |
| 程式名稱 | select\_max\_orderno.js | | | | | |
| 目的 | 取得計畫表內景點的最大排序編號 | | | | | |
| 程式碼 | | | | | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var select\_max\_orderno = async function(schedule\_id){      //存放結果      let result;      //讀取資料庫      await query('select max(order\_no) from process where schedule\_id=$1', [schedule\_id])          .then((data) => {              if(data.rows.length>0){                  result=data;              }else{                  result=-1;//找不到資料              }          }, (error) => {              result = -9;  //執行錯誤          });      //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {select\_max\_orderno}; | | | | | | |
| 程式名稱 | | select\_schedule.js | | | | |
| 目的 | | 取得群組內所有計畫表 | | | | |
| 程式碼 | | | | | | |
| 'use strict';  //引用操作資料庫的物件  const query = require('./asyncDB');  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var selectSchedule = async function(group\_id){      //存放結果      let result;      //讀取資料庫      await query('select \* from schedule where group\_id=$1', [group\_id])          .then((data) => {              if(data.rows.length>0){                  result=data;              }else{                  result=-1;//找不到資料              }          }, (error) => {              result = -9;  //執行錯誤          });      //回傳執行結果      return result;  }  //------------------------------------------  //匯出  module.exports = {selectSchedule}; | | | | | | |

▼表 9-2-2 WEB端程式規格描述

|  |  |
| --- | --- |
| 程式名稱 | product |
| 目的 | 取得相關頁面資料 |
| 程式碼 | |
| 'use strict';  //引用操作資料庫的物件  const sql = require('./asyncDB');  //------------------------------------------  //執行資料庫動作的函式-傳回所有產品資料  //------------------------------------------  var list = async function(group\_id){  var result=[];    await sql('SELECT \* FROM collection where group\_id=$1',[group\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-取出單一商品  //------------------------------------------  var one = async function(group\_id){  var result={};    await sql('SELECT \* FROM collection INNER JOIN attractions ON collection.att\_id = attractions.att\_id WHERE group\_id = $1', [group\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  //執行資料庫動作的函式-傳回所有產品資料  //------------------------------------------  var processlist = async function(){  var result=[];    await sql('SELECT \* FROM schedule AS a LEFT JOIN process AS b ON a.schedule\_id = b.schedule\_id LEFT JOIN attractions AS c ON b.att\_id = c.att\_id')  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  var schedulelist = async function(group\_id){  //var group\_id = 1;  //console.log('\*\*\*\*');  //console.log(group\_id);  var result=[];    await sql('SELECT \* FROM schedule where group\_id=$1',[group\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  //----------------------------------  // 新增計畫表  //----------------------------------  var addPlan = async function(newData){  var result;  await sql('INSERT INTO schedule (schedule\_name,group\_id) VALUES ($1,$2)', [newData.schedule\_name,newData.group\_id])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 新增景點  //----------------------------------  var add = async function(newData){  var result;  await sql('INSERT INTO schedule (schedule\_name) VALUES ($1,$2)', [newData.schedule\_name])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 0724查詢某行程的所有景點  //----------------------------------  var scheduledetail = async function(schedule\_id){  var result={};    await sql('select \* from schedule join process on schedule.schedule\_id= process.schedule\_id where schedule.schedule\_id= $1', [schedule\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  //----------------------------------  // 刪除商品  //----------------------------------  var remove = async function(place\_id){  var result;  await sql('DELETE FROM process WHERE place\_id = $1', [place\_id])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------  // 刪除收藏  //----------------------------------  var removecollection = async function(place\_id){  var result;  await sql('DELETE FROM collection WHERE place\_id = $1', [place\_id])  .then((data) => {  result = data.rowCount;  }, (error) => {  result = -1;  });    return result;  }  var scheduledetail = async function(schedule\_id){  var result={};    await sql('select place\_id , order\_no , schedule.schedule\_id , schedule.group\_id ,process.place\_name from schedule join process on schedule.schedule\_id= process.schedule\_id where schedule.schedule\_id= $1 order by order\_no', [schedule\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  //修改order\_no  var schedulemove = async function(place\_id,order\_no){  var result={};  await sql('update process set order\_no = $1 where place\_id= $2', [order\_no ,place\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  var deleteprocess = async function(place\_id){  var result={};    await sql('delete from process where place\_id = $1', [place\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  //------------------------------------------  //執行資料庫動作的函式-依照placeid去找他的GID  //------------------------------------------  var select = async function(group\_id){  var result=[];    await sql('SELECT \* FROM schedule where group\_id=$1',[group\_id])  .then((data) => {  result = data.rows;  }, (error) => {  result = null;  });    return result;  }  var collectiontoprocess = async function(newData){  var result;  await sql('INSERT INTO process (place\_id,schedule\_id,place\_name) VALUES ($1,$2,$3)', [newData.place\_id,newData.schedule\_id,newData.place\_name])  .then((data) => {  result = 0;  }, (error) => {  result = -1;  });    return result;  }  //----------------------------------------  // 新增收藏資料  //------------------------------------------  var selectProcess = async function(schedule\_id){  //存放結果  var result;  console.log(schedule\_id+"test")  //讀取資料庫  await sql('select place\_name from process where schedule\_id=$1 ', [schedule\_id])  .then((data) => {  if(data.rows.length>0){  result=data;  }else{  result=-1;//找不到資料  }  }, (error) => {  result = -9; //執行錯誤  });    //回傳執行結果  return result;  }  //------------------------------------------  // 新增收藏資料  //------------------------------------------  var setOrder = async function(order\_no,schedule\_id,place\_name){  //存放結果  var result;  //讀取資料庫  await sql('UPDATE process set order\_no= $1 where schedule\_id=$2 and place\_name=$3', [order\_no,schedule\_id,place\_name])  .then((data) => {  result = data.rowCount; //update資料數  }, (error) => {  result = -9; //執行錯誤  });  //回傳執行結果  return result;  }  //------------------------------------------  //處理Unhandle的問題  process.on('unhandledRejection', error => {  // Will print "unhandledRejection err is not defined"  });  new Promise((\_, reject) => reject(new Error('woops')))  .catch(new Function());    //----------------------------------------  //匯出  module.exports = {list,one,processlist,schedulelist,add,scheduledetail,remove,removecollection,schedulemove,deleteprocess,scheduledetail,select,collectiontoprocess,addPlan,setOrder,selectProcess}; | |

|  |  |  |
| --- | --- | --- |
| 程式名稱 | collection | |
| 目的 | 顯示收藏頁面 | |
| 程式碼 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const product = require('./utility/product');  //接收GET請求  router.get('/', function(req, res, next) {      var group\_id = req.query.group\_id;      console.log(req.query.group\_id);        product.list(group\_id).then(data => {          if(data==null){              console.log("error");              res.render('error');  //導向錯誤頁面          }else if(data.length > 0){              console.log("collection\_success");              res.render('collection', {items:data});  //將資料傳給顯示頁面          }else{              console.log("notfound");              res.render('notFound');  //導向找不到頁面          }      })  });  router.get('/collect/delete/:place\_id', function(req, res, next) {      var place\_id = req.params.place\_id;      product.removecollection(place\_id).then(d => {          if (d >= 0) {              return res.redirect('back');              console.log(place\_id);          } else {              res.render('addFail');     //導向錯誤頁面              console.log(fail);          }      })  });  module.exports = router; | | |
| 程式名稱 | collection\_add | |
| 目的 | 把相關參數帶到選擇加入至某個計畫表頁面 | |
| 程式碼 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const product = require('./utility/product');  //接收POST請求  router.post('/', function(req, res, next) {      var place\_id = req.body.place\_id;      var schedule\_id = req.body.schedule\_id; //取得產品編號      var place\_name = req.body.place\_name; //取得產品編號      console.log(place\_id);      console.log(schedule\_id);      console.log(place\_name);      // 建立一個新資料物件      var newData={          schedule\_id:schedule\_id,          place\_id:place\_id,          place\_name:place\_name      }        product.collectiontoprocess(newData).then(d => {          if (d==0){              res.render('addSuccess');  //傳至成功頁面              console.log('addSuccess '+place\_id);              console.log('addSuccess '+schedule\_id);              console.log('addSuccess '+place\_name);          }else{              res.render('addFail');     //導向錯誤頁面              console.log('addFail '+place\_id);              console.log('addFail '+schedule\_id);              console.log('addFail '+place\_name);          }      })  });  module.exports = router; | | |
| 程式名稱 | product\_add | |
| 目的 | 執行新增計畫表 | |
| 程式碼 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const product = require('./utility/product');  //接收POST請求  router.post('/', function(req, res, next) {      var schedule\_name = req.body.schedule\_name;                  //取得產品編號      var day\_number = req.body.day\_number;              //取得產品名稱      // 建立一個新資料物件      var newData={          schedule\_name:schedule\_name,          day\_number:day\_number      }        product.add(newData).then(d => {          if (d==0){              res.render('addSuccess');  //傳至成功頁面          }else{              res.render('addFail');     //導向錯誤頁面          }      })  });  module.exports = router; | | |
| 程式名稱 | | addform |
| 目的 | | 顯示新增計畫表頁面 |
| 程式碼 | | |
| var express = require('express');  var router = express.Router();  //增加引用函式  //var moment = require('moment');  const product = require('./utility/product');  //接收POST請求  router.post('/', function(req, res, next) {        //console.log('post begin');      var schedule\_name = req.body.schedule\_name;                  //      var group\_id = req.body.group\_id;        // 建立一個新資料物件      var newData={          schedule\_name:schedule\_name,          group\_id:group\_id      }        product.addPlan(newData).then(d => {          console.log('d = ' + d);          if (d == 0){              console.log('success');              res.render('product\_add\_form', {items:d});  //傳至成功頁面          }else if(d == -1){              console.log('notFound');              res.render('notFound');  //導向找不到頁面          }else{              console.log('error');              res.render('error');  //導向錯誤頁面          }      })    });  module.exports = router ; | | |
| 程式名稱 | | reschedule |
| 目的 | | 執行推薦路徑 |
| 程式碼 | | |
| var express = require('express');  var router = express.Router();  var request= require('request');  //增加引用函式  //var moment = require('moment');  const product = require('./utility/product');  //接收POST請求  router.post('', function(req, res, next) {      var schedule\_id = String(req.body.schedule\_id);      var lat = String(req.body.latitude);      var lng = String(req.body.longitude);          console.log("js"+schedule\_id);        product.selectProcess(schedule\_id).then(data=>{          if(data==null){              console.log("執行錯誤");              res.render('error');  //導向錯誤頁面          }else if(data==-1){              console.log("此計畫表目前無景點");              res.render('notFound');  //導向找不到頁面          }else{               var origin='21.903975, 120.854543'; //出發地經緯度              //var origin =lat +', '+ lng;//出發地經緯度              var destination=origin+'|';              var place\_array=new Array(data.rows.length);                for(i=0;i<data.rows.length;i++){                  if(i!=(data.rows.length-1)){                      destination=destination+data.rows[i].place\_name+'|';                      place\_array[i]=data.rows[i].place\_name;                  }else{                      destination=destination+data.rows[i].place\_name;                      place\_array[i]=data.rows[i].place\_name;                  }              }              console.log(place\_array);                const distance\_matrix\_url=encodeURI('https://maps.googleapis.com/maps/api/distancematrix/json?key=AIzaSyA2dazAHnsooxnM6akxlJcp\_vq1klEDScE&origins='+destination+'&destinations='+destination+'&region=tw&language=zh-TW');              request(distance\_matrix\_url,function(error,response,body){                    if(!error&&response.statusCode==200){                      console.log(body+"test2");                      var jsobject=JSON.parse(body);                    //建立二維陣列                      var distance\_matrix=new Array();                      for(i=0;i<=data.rows.length;i++){                          distance\_matrix[i]=new Array();                          var subobject= jsobject.rows[i];                          for(j=0;j<=data.rows.length;j++){                              distance\_matrix[i][j]=subobject.elements[j].duration.value;                          }                      }                        //找陣列中非零最小值                      Array.prototype.min=function(){                          var min=this[0];                          if(min==0){                              min=Infinity;                          }                          this.forEach(function(ele,index,arr){                              if(ele<min&&ele!=0){                                  min=ele;                              }                          })                          return min;                      }                      console.log(distance\_matrix);                      var index\_array=new Array();                      var index;                      var next;                      for(i=0;i<distance\_matrix.length-1;i++){                          if(i==0){                              index=0;                          }else{                              index=next;                          }                          next=distance\_matrix[index].indexOf(distance\_matrix[index].min());                            product.setOrder(i+1,schedule\_id,place\_array[next-1]).then(data=>{                              if(data==-9){                                  console.log("執行錯誤");                              }else{                                  console.log("update成功");                              }                          })                          console.log("第"+(i+1)+"個要去的景點:"+place\_array[next-1]);                            //將已選擇過的景點的該列該行元素皆設為無限大                          for(j=0;j<distance\_matrix.length;j++){                              distance\_matrix[index][j]=Infinity;                              distance\_matrix[j][index]=Infinity;                          }                        }                      console.log('hihi');                  }              })                console.log('reschedulesuccess');              res.render('schedule\_query', {items:data});          }      })  });    module.exports = router ; | | |

|  |  |
| --- | --- |
| 程式名稱 | schedule |
| 目的 | 顯示所有計畫表 |
| 程式碼 | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const product = require('./utility/product');  //接收GET請求  router.get('/', function(req, res, next) {       var group\_id = req.query.group\_id;       console.log(req.query.group\_id);      product.schedulelist(group\_id).then(data => {          if(data==null){              console.log("error");              res.render('error');  //導向錯誤頁面          }else if(data.length > 0){              console.log("schedule\_success");              res.render('schedule', {items:data});//將資料傳給顯示頁面          }else{              console.log("notfound");              res.render('notFound');  //導向找不到頁面          }      })  });  //處理Unhandle的問題  process.on('unhandledRejection', error => {      // Will print "unhandledRejection err is not defined"    });    new Promise((\_, reject) => reject(new Error('woops')))  .catch(new Function());      //----------------------------------------  module.exports = router; | |

|  |  |
| --- | --- |
| 程式名稱 | schedule\_deleteprocess |
| 目的 | 顯示收藏頁面 |
| 程式碼 | |
| var express = require('express');  var router = express.Router();  //增加引用函式  //var moment = require('moment');  const product = require('./utility/product');  //接收POST請求  router.post('/', function(req, res, next) {      // console.log('abc');      //var newData = req.body.count;      //var place\_id = String(req.body.id);      var place\_id = String(req.body.place\_id);        console.log(place\_id);      //product.addPlaceID(newData).then(d => {      product.deleteprocess(place\_id).then(d => {          if (d == null){              console.log('error');              res.render('error');  //導向錯誤頁面          }else if(d == -1){              console.log('notFound');              res.render('notFound');  //導向找不到頁面          }else{              console.log('deletesuccess');              res.render('schedule\_query', {items:d});  //傳至成功頁面          }      })    });  module.exports = router ; | |
| 程式名稱 | schedule\_move |
| 目的 | 拖曳行程並改變順序 |
| 程式碼 | |
| var express = require('express');  var router = express.Router();  //增加引用函式  const product = require('./utility/product');  //接收POST請求  router.post('/', function(req, res, next) {      var place\_id = String(req.body.place\_id);      var order\_no = req.body.order\_no;      product.schedulemove(place\_id,order\_no).then(d => {          if (d == null){              console.log('error');              res.render('error');  //導向錯誤頁面          }else if(d == -1){              console.log('notFound');              res.render('notFound');  //導向找不到頁面          }else{              console.log('movesuccess');              res.render('schedule\_query', {items:d});  //傳至成功頁面            }      })      });  module.exports = router ; | |
| 程式名稱 | schedule\_query |
| 目的 | 顯示詳細行程畫面 |
| 程式碼 | |
| var express = require('express');  var router = express.Router();    //增加引用函式  //var moment = require('moment');  const product = require('./utility/product');  //接收GET請求  router.get('/', function(req, res, next) {      var schedule\_id = req.query.schedule\_id;   //取出參數      product.scheduledetail(schedule\_id).then(data => {           if (data==null){              res.render('error');  //導向錯誤頁面          }else if(data==-1){              res.render('notFound');  //導向找不到頁面          }else{              console.log(schedule\_id);              res.render('schedule\_query', {items:data});  //將資料傳給顯示頁面          }      })  });  router.get('/place/delete/:place\_id', function(req, res, next) {      var place\_id = req.params.place\_id;      product.remove(place\_id).then(d => {          if (d >= 0) {              return res.redirect('back');              //return res.redirect('/schedule\_query');  //傳至成功頁面              console.log(place\_id);          } else {              res.render('addFail');     //導向錯誤頁面              console.log(fail);          }      })  })  module.exports = router; | |
| 程式名稱 | select |
| 目的 | 選擇將收藏的景點加入至某個計畫表裡 |
| 程式碼 | |
| var express = require('express');  var router = express.Router();  const product = require('./utility/product');  /\* GET home page. \*/  router.get('/', function(req, res, next) {      var place\_id = req.query.place\_id;      var group\_id = req.query.group\_id;      console.log('exe select');      console.log('place\_id  '+req.query.place\_id);      console.log('group\_id  '+req.query.group\_id);          product.select(group\_id).then(data => {          if(data==null){              console.log("error");              res.render('error');  //導向錯誤頁面          }else if(data.length > 0){              console.log("select\_success");              res.render('select', {items:data});//將資料傳給顯示頁面          }else{              console.log("notfound");              res.render('notFound');  //導向找不到頁面          }      })  });  module.exports = router; | |
| 程式名稱 | main |
| 目的 | 判斷各個頁面的button所需執行的function |
| 程式碼 | |
| $(function() {      $("#sortable").sortable(      {opacity: 0.7,          update :function(event,ui){              var count = 0;                $("#sortable").find(".card").each(function(index, element){                    count++;                  var id = $(this).attr("id");                    var data = {                  place\_id : id,                  order\_no : count                  };                  $.ajax({                      type:'post',                      url: '/schedule/move',                      data : data,                      dataType : JSON                  });                });                // put\_result(ui);          },          stop :function(event,ui){              var count = 1;                $("#sortable").find(".card").each(function(index, element){                  var id = $(this).attr("id");                  $("#"+id).find(".card-header").find("strong").html("景點");                  $("#"+id).find(".card-header").find("strong").append(String(count));                  // $("#header" + count).find("strong").html("景點");                  // $("#header" + count).append(String(count));                  count++;              });          }      });      $("button.btn-danger.float-right").click(function(){          if ( confirm("確定刪除嗎？") )  {              var id = $(this).attr("id");              id = id.replace("button","");              // alert(id);              $.ajax({                  type:'post',                  url: '/schedule/deleteprocess',                  data : {                      place\_id : id                  },                  dataType : JSON              });              setTimeout(function() {                  location.reload()              }, 500);                  // let params = (new URL(document.location)).searchParams;              // let schedule\_id = params.get("schedule\_id");                // $.ajax({              //     type:'get',              //     url: '/schedule/query',              //     data : {              //         schedule\_id : schedule\_id              //     },              //     dataType : JSON              // });           }           else{             };        });      $("#addPlanFunction").click(function(){            var group\_id = "";            var url = location.href;          //var url = req.param('group\_id');          var ary = url.split('?')[1].split('&');          //此時ary的內容為：          //ary[0] = 'id=U001'，ary[1] = 'name=GQSM'            //下迴圈去搜尋每個資料參數          for(i=0;i<=ary.length-1;i++)          {              //如果資料名稱為id的話那就把他取出來              if(ary[i].split('=')[0] == 'group\_id')              group\_id = ary[i].split('=')[1];          }            var planContent = document.getElementById("inputschedule").value;            if (planContent.length < 1)          {              alert('請輸入資料')          }          else          {              var data = {                  group\_id : group\_id,                  schedule\_name : planContent              };              // alert(group\_id);              // alert(schedule\_name);                $.ajax({                  type:'post',                  url: '/product/addform',                  data : data,                  dataType : JSON              });              alert('新增成功');              setTimeout(function() {                  location.reload()              }, 500);          }      });      $("#recommendPath").click(function(){          if ( confirm("確定推薦嗎？") )  {              //var id = $(this).attr("id");              var id = "";                var url = location.href;              //var url = req.param('group\_id');              var ary = url.split('?')[1].split('&');              //此時ary的內容為：              //ary[0] = 'id=U001'，ary[1] = 'name=GQSM'                //下迴圈去搜尋每個資料參數              for(i=0;i<=ary.length-1;i++)              {                  //如果資料名稱為id的話那就把他取出來                  if(ary[i].split('=')[0] == 'schedule\_id')                  id = ary[i].split('=')[1];              }              $.ajax({                  type:'post',                  url: '/reschedule',                  data : {                      schedule\_id : id,                      latitude  : lat,                      longitude : lng                  },                  dataType : JSON              });                $('#circle').show()                setTimeout(function() {                  location.reload()              }, 3000);              setTimeout(function() {                  location.reload()              }, 3000);           }           else{             };        });        //收藏加入景點      $("#addcollectionFunction").click(function(){          var val=$('input:radio[name="scheduleid"]:checked').val();          if(val==null){              alert("請選擇一項計畫表!");              return false;          }          else{              var url = location.href;              let tempUrl = new URL(url);              var place\_id = tempUrl.searchParams.get('place\_id');              var place\_name =  tempUrl.searchParams.get('place\_name');              //var url = req.param('group\_id');              var data = {                  schedule\_id : val,                  place\_id : place\_id,                  place\_name : place\_name              };              $.ajax({                  type:'post',                  url: '/collection/add',                  data : data,                  dataType : JSON              });              alert('新增成功');          }        });    });  function toAddPlan(){      var group\_id = "";      var url = location.href;      //var url = req.param('group\_id');      var ary = url.split('?')[1].split('&');      //此時ary的內容為：      //ary[0] = 'id=U001'，ary[1] = 'name=GQSM'      //下迴圈去搜尋每個資料參數      for(i=0;i<=ary.length-1;i++)      {          //如果資料名稱為id的話那就把他取出來          if(ary[i].split('=')[0] == 'group\_id')          group\_id = ary[i].split('=')[1];      }      //location.href('/product/add/form');      location.href = '/product/add/form?group\_id=' + group\_id ;  }; | |

**第十章 測試模型**

**10-1測試計畫**

我們以使用者的角度測試Chavel的各個功能是否能正常運作，LINE端測試結果如表10-2-1～10-2-10，網站測試結果如表10-2-11～10-2-15。

* LINE端

1. 建立計畫表：確認新增計畫表是否成功(LINE端)。
2. 選單功能：確認選單功能是否成功跳出。
3. 搜尋景點：確認是否可以找到景點。
4. 附近景點：確認是否可以跳出目前位置的附近景點。
5. 查看單一行程：確認查看單一計畫表是否可以顯示在LINE上。
6. 加入收藏：確認是否可以把景點加入至收藏頁面。
7. 加入行程：確認是否可以把景點加入至某一個行程裡。
8. 打開收藏頁面：確認是否可以跳轉到收藏畫面。
9. 打開行程頁面：確認是否可以跳轉到行程畫面。
10. 指令幫助：確認是否會跳出指令說明。

* 網站

1. 建立計畫表：確認新增計畫表是否成功(網站)。
2. 行程拖曳：確認是否能拖曳行程改變行程順序。
3. 刪除行程：確認是否能刪除指定的行程。
4. 推薦路徑：確認是否能按照演算法排出推薦順序。
5. 刪除收藏：確認是否可以刪除收藏
6. 收藏加入至行程：確認是否可以收藏的景點加入至某一個計劃表裡。

**10-2測試個案與測試結果資料**

▼表10-2-1 建立計畫表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 建立計畫表 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-2 選單功能

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 選單功能 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | | | |

▼表10-2-3 搜尋景點

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 搜尋景點 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-4 查看單一行程

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 查看單一行程 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-5 加入收藏

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 加入收藏 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-6 加入行程

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 加入行程 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-7 指令功能

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 指令功能 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | | | |

▼表10-2-8 打開收藏頁面

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 打開收藏頁面 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-9 打開行程頁面

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 打開行程頁面 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-10 附近景點

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 附近景點 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-11 行程拖曳

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 行程拖曳 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-12 刪除行程

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 刪除行程 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-13 刪除收藏

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 刪除收藏 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-14 推薦路徑

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 推薦路徑 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

▼表10-2-15 收藏加入至行程

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名稱 | 收藏加入至行程 | 測試結果是否成功 | 是 |
| 測試畫面 | | | |
|  | |  | |

**第十一章 操作手冊**

▼表11-1 Chavel操作手冊

|  |  |
| --- | --- |
| 步驟1. 開啟LINE進入加入好友畫面，輸入Chavel的ID | 步驟2.將Chavel加為好友。 |
|  |  |
| 步驟3. 將Chavel加進群組 | 步驟4. 開始使用Chavel |
|  | C:\Users\hwanacorn0806\Downloads\1573805868293.jpg |

**第十二章 使用手冊**

▼表12-1 Chavel使用手冊

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 建立計畫表 | | |
|  | |  |  |
| 叫出選單，點擊計畫表按鈕 | | Chavel顯示建立計畫表教學 | 顯示建立計畫表成功 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | 搜尋景點並顯示景點圖卡 | | |
|  | |  |  |
| 叫出選單，點擊搜尋景點按鈕 | | Chavel顯示搜尋景點教學 | 顯示景點 |
| 3 | 景點圖卡功能 | | |
|  | |  |  |
| 可以顯示景點電話及營業時間 | | 選擇景點欲加入之計畫表 | 顯示景點加入計畫表成功 |

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | 附近景點 | | |
|  | |  |  |
| 叫出選單，點擊附近景點按鈕 | | 取得手機位置 | 推薦手機附近景點 |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 查看單一計畫表 | | |
|  | |  |  |
| 叫出選單，點擊單一行程按鈕 | | 選擇查看的計畫表 | 點選調整行程 |
|  | |  |  |
| 可直接拖曳改變順序 | | 點擊推薦路徑 | 依據現有景點推薦路徑 |

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | 收藏功能 | | |
|  | |  |  |
| 叫出選單，點擊收藏按鈕 | | 顯示所有收藏，可將收藏新增至行程 | 選擇景點欲加入之行程 |
|  | |  |  |
| 顯示新增至行程成功 | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 建立計畫表 | | |
|  | |  |  |
| 叫出選單，點擊行程按鈕 | | 所有建立過的計畫表 | 顯示計畫表中之所有景點 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | 幫助使用者操作Chavel | | |
|  | |  |  |
| 叫出選單，點擊幫助按鈕 | | 顯示Chavel功能說明 |  |

**第十三章 感想**

10736010楊淯絨

經歷過五專三學期的專題，學到很多東西，而二技又與不同的同學做很不一樣的題目，第二次的專題已經不比五專時更慌張不知所措，雖然在一評的時候也被評審質疑後我們也灰心過，但我們還是堅持下來並做出一個大家都還滿意的作品；同樣幸運的是我還是與一群認真優秀的組員們共同打拼，在過程中大家一起討論學習，互相鼓勵互相幫忙，大家也能用溝通而不是爭吵的方式來解決問題，很謝謝我的組員在這一年的包容，能跟我的組員們一起經歷這一年真的很棒，雖然辛苦，但學到的東西真的是很寶貴。

10736011張祥智

經歷了兩個學期的專題,過程中學到了許多事情,一開始的討論題目,大家也都努力的想,過程中雖然有不同的意見,但是也都能達成共識。在寫程式的過程中,有時也會遇到許多瓶頸,例如:server的建置、資料庫與程式介接、排版和顏色配置、文件製作。尤其Linebot又是比較新的技術，學習的時間也花了不少，團隊合作也少不了的,當遇到比較困難的問題,大家也能夠互相幫忙。有時候問題卡住時,會不太想碰程式,會有挫折感,經過詢問指導老師和同學,解決問題後,會有種莫名的成就感,也有動力繼續寫下去。經過這次專題,讓我可以了解到寫程式的辛苦,從討論主題到動工寫程式,是需要花費相當大的心力與時間,也算是學習到寶貴的經驗。謝謝組員一起團結努力完成專題,也謝謝指導老師的教導,這一路走來也是難能可貴的經驗。

10736012林宜萱

面對第二次的專題，照理來說應該比較得心應手，雖然是有比較不那麼緊張，問題處理上也比較容易上手，不過這次專題方向往linebot發展，對於我們完全沒學過也無任何基礎，我覺得也是一大挑戰，又必須重頭開始摸索。第一次發表時，我們這組也是進度落後，看每組簡報時，幾乎都可以demo雛型，但我們的介面滿零碎，整體無法拼湊起來，那時候真的滿擔心，但隨著我們的努力，漸漸地有一點一點的雛型，心中還是挺感動。專題除了團隊溝通外，我覺得最重要就是學習面對困難，以及心態如何調整，遇到問題找出答案固然很重要，但更重要的是找出問題。

10736025顏毓陞

從一評時，對LineBot完全沒概念，到現在快最後一評了，不敢說學了很多，但至少學了不少，不管是在程式編碼，還是在小組間的團隊分工合作，或是在專案上的責任感。還記得剛開始撰寫程式時，我什麼都不懂，不知從何開始起步，過程中應用程式常常跑不出來或是跑出來不是預期的，到現在發現其實一路上就是不懂就去爬文，文爬到了結果出不來，就是不斷地debug，就這樣反反覆覆就好像無窮迴圈一樣地去執行，始終會有答案。在這個過程中我體驗到了做一個專案的方法，流程，態度等等，很感謝能有這樣的經驗。

N1076716呂芷瑛

已經是第二次面對資管的畢業專題，但這次所運用到的技術以及合作夥伴與五專時期全然不同。我認為在專題過程中很重要的一環是我們如何學習解決問題，比如當遇到需要的技術卻不會使用時，是該如何尋求資源讓現狀有所進展。非常幸運能夠和朋友組成一個團隊。在第一次評核結束時我們不免有些懷疑自己所做的是否沒有價值，但再和團隊討論後我們更加確定彼此共同的目標，就是一同完成一個完善的linebot機器人，即使我們所做不一定符合現階段社會的商業模式，但從中學習到的技術及能夠善用的資源卻可以讓我們有所進步，我相信專題的目的不僅僅是技術的學習與進步，更是在過程中與夥伴們的分工與合作。非常幸運能和朋友組成一個團隊!

**第十四章 參考資料**

<https://github.com/tomlinNTUB/NodeJS-Express>

<https://github.com/tomlinNTUB/Linebot-Heroku>

<https://github.com/tomlinNTUB/NodeJS-Express>

<https://getbootstrap.com/docs/4.3/getting-started/introduction/>

<https://developers.google.com/maps/documentation/javascript/tutorial>

<https://developers.line.biz/en/docs/messaging-api/using-flex-messages/>

<https://jqueryui.com/sortable/>

<https://www.w3school.com.cn/jquery/ajax_post.asp>

<https://www.itread01.com/p/1326057.html>

https://github.com/tomlinNTUB/Linebot-Heroku

https://developers.line.biz/en/services/messaging-api/

https://docs.postgresql.tw/the-sql

https://developers.google.com/

https://www.w3schools.com/

**附件一 評審之建議修正情形**

|  |  |
| --- | --- |
| **評審建議事項** | **修正情形** |
| 景點與行程與現實面差異太大，使系統的實用性太低。 | 有大幅改善整個系統操作流程。 |
| 比較難以實現的排程部分尚未看到比較可行的方法。 | 我們以距離矩陣演算法去計算推薦路徑。 |
| 簡報的主軸掌握要再加強。 | 最後一評我們會著重在DEMO跟系統特色，以凸顯該系統的獨特性。 |
| 系統進度需再注意 | 謝謝評審，我們有努力趕進度，也知道自己做得太少。 |