Weather Tour

設計文件

|  |  |
| --- | --- |
| 專案名稱 | 天氣旅遊(WT) |
| 撰寫日期 | 撰寫日期: 2017/5/16 |
| 發展者 | 朱恆慶、吳家豪、莊天耀 |

**版次變更記錄**

|  |  |  |
| --- | --- | --- |
| 版次 | 變更項目 | 變更日期 |
| 0.1 | 初版 | 2017/11/17 |
| 0.2 | 二版 | 2017/11/18 |
| 1.0 | 發佈版 | 2017/11/22 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[版次變更記錄…………………………………………………………………………………..1](#_Toc435575557)

[1. 系統模型與架構(System Model/System Architecture)…………………………………...1](#_Toc435575558)

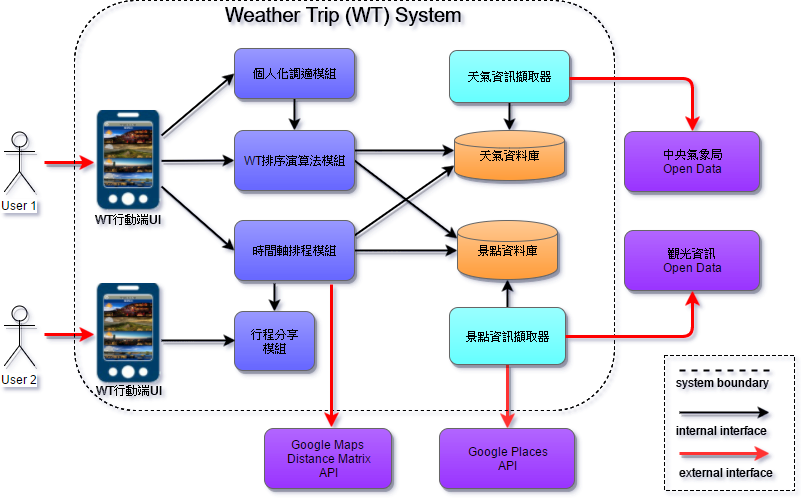
[2. 類別圖設計(Class Diagram).……………………………………………………………10](#_Toc435575563)

[3. 資料細部設計(Data Design).……………………………………………………………13](#_Toc435575565)

[4. 設計議題(Design Issue) …………………………………………………………………22](#_Toc435575566)

1. **系統模型與架構(System Model/System Architecture)**

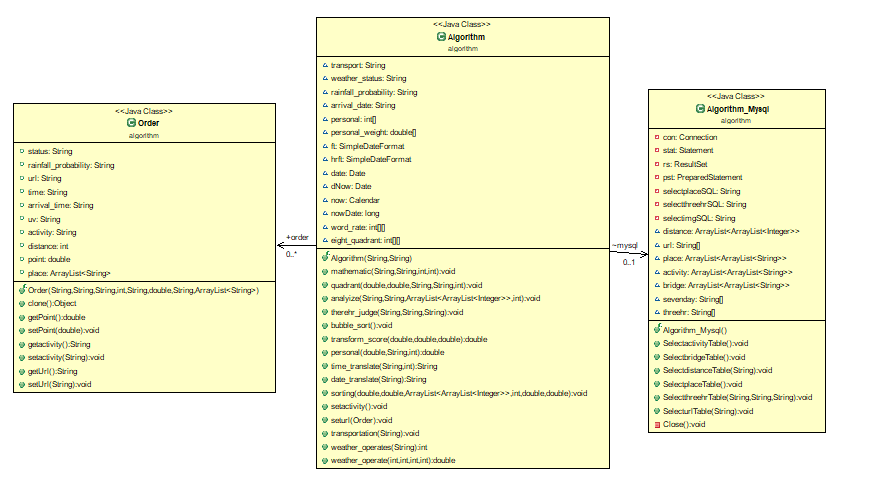
Weather Tour為旅遊app, 藉由該景點的天氣、景點距離兩者的演算法排序將當日或當時適合的遊玩的景點排序出來供使用者來選擇，讓使用者能只用Weather Tour就可排除不適合旅遊點(不適合指壞天氣、距離過遠或是塞車…等等因素)，讓使用者在出遊時不會碰上到了當地才發現傾盆大雨，因而一掃出遊興的情況，而也能進而知道該景點的細項天氣或是周邊景點、美食、活動等其他綜合資訊。而使用者也能透過偏好設定來選擇交通、日期、地區、類別、個人化比重等設定來製成景點選單，系統也提供了列表模式跟樂透模式等兩種模式供使用者挑選行程。在選定行程後可將行程放入 ”旅遊籃” 或是 ”我的行程” 內，而 ”旅遊籃” 為記錄所有使用者選擇的景點，以清單模式供使用者查看；”我的行程” 則是將使用者所選擇的景點排序為行程模式，可檢視行程時間軸、行程導航和景點周邊細項紀錄等，是以行程時間軸等方式供使用者查看，而將行程封裝後也可供使用者以”QR code” 或是 ”網址分享” 等方式讓使用者分享自己當天的行程給他人查看。



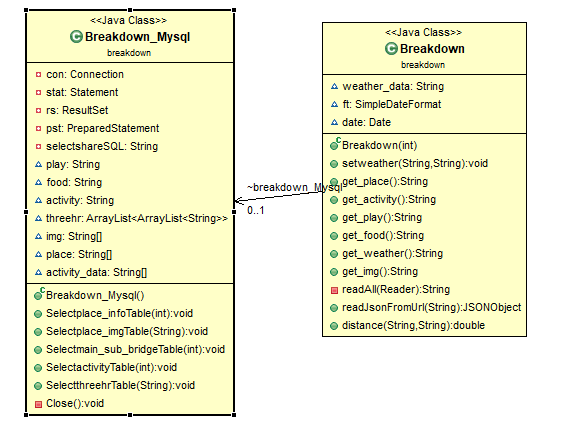
(模組架構圖)圖)

1. **類別圖設計(Class Diagram)**

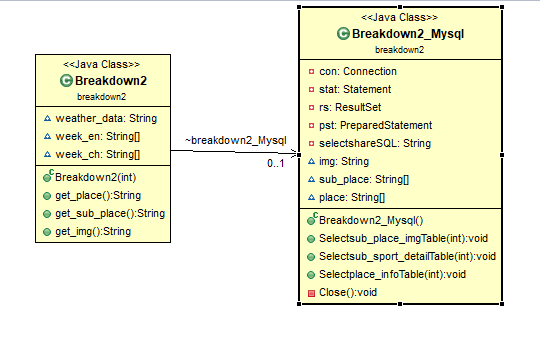
* 行程排序



* 主景點資訊



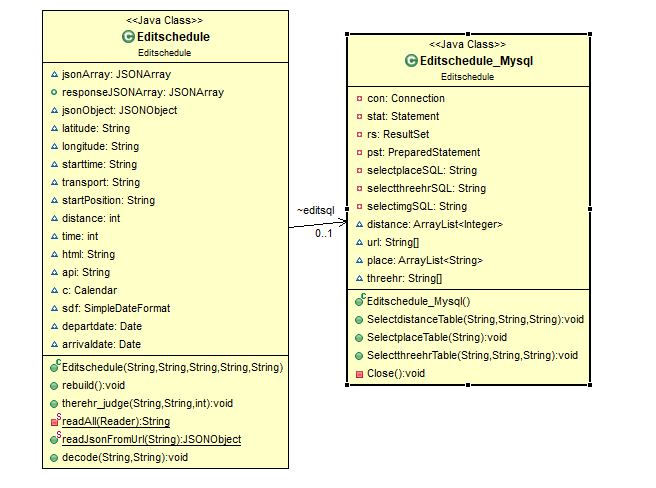
* 子景點資訊



* 修正景點時間



* 編輯行程



* 行程分享



|  |  |
| --- | --- |
| Class Name | WT. algorithm. Algorithm |
| Category | Java |
| Description | 運算使用者各項資訊，回傳行程推薦列表 |
| Attributes | 無 |
| Relationship with other classes | Algorithm\_Mysql，Order |

|  |  |
| --- | --- |
| Class Name | WT. breakdown. Breakdown |
| Category | Java |
| Description | 回傳主景點各項資訊 |
| Attributes | 無 |
| Relationship with other classes | Breakdown\_Mysql |

|  |  |
| --- | --- |
| Class Name | WT. breakdown2. Breakdown2 |
| Category | Java |
| Description | 回傳子景點各項資訊 |
| Attributes | 無 |
| Relationship with other classes | Breakdown2\_Mysql |

|  |  |
| --- | --- |
| Class Name | WT. correctTourbasket. CorrectTour |
| Category | Java |
| Description | 定時修正旅遊籃暫存資訊 |
| Attributes | 無 |
| Relationship with other classes | correct\_Mysql |

|  |  |
| --- | --- |
| Class Name | WT. editschedule. Editschedule |
| Category | Java |
| Description | 將編輯過後的行程資訊修正，且回傳 |
| Attributes | 無 |
| Relationship with other classes | Editschedule\_Mysql |

|  |  |
| --- | --- |
| Class Name | WT. share. Share |
| Category | Java |
| Description | 分享行程時外部網頁生成 |
| Attributes | 無 |
| Relationship with other classes | Share\_Mysql |

1. **資料細部設計(Data Design)**

在mySql上分別建立9個表格：activity、distance\_car、distance\_scooter、distance\_walk、main\_place、place\_img、share、sub\_viewpoint、sub\_viewpoint\_img

1. Activity:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **id** | int(5) |
| 2 | **name** | varchar(50) |
| 3 | **start** | varchar(20) |
| 4 | **end** | varchar(20) |
| 5 | **lat** | varchar(15) |
| 6 | **lng** | varchar(15) |
| 7 | **info** | varchar(1000) |
| 8 | **url** | varchar(1000) |
| 9 | **pic1** | varchar(1000) |
| 10 | **pic2** | varchar(1000) |
| 11 | **pic3** | varchar(1000) |

1. distance\_car:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **ID** | int(10) |
| 2 | **placeAID** | int(5) |
| 3 | **placeBID** | int(5) |
| 4 | **distance** | int(10) |
| 5 | **time** | int(10) |

1. distance\_scooter:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **ID** | int(10) |
| 2 | **placeAID** | int(5) |
| 3 | **placeBID** | int(5) |
| 4 | **distance** | int(10) |
| 5 | **time** | int(10) |

1. distance\_walk:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **ID** | int(10) |
| 2 | **placeAID** | int(5) |
| 3 | **placeBID** | int(5) |
| 4 | **distance** | int(10) |
| 5 | **time** | int(10) |

1. main\_place:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **id** | int(5) |
| 2 | **place\_id** | varchar(30) |
| 3 | **name** | varchar(50) |
| 4 | **lat** | varchar(15) |
| 5 | **lng** | varchar(15) |
| 6 | **postal\_code** | varchar(5) |
| 7 | **weekday\_text** | varchar(500) |
| 8 | **phone** | varchar(20) |
| 9 | **In\_out\_door** | varchar(10) |
| 10 | **tag** | varchar(10) |

1. place\_img:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **ID** | int(10) |
| 2 | **place\_id** | varchar(30) |
| 3 | **name** | varchar(50) |
| 4 | **url** | varchar(3000) |

1. Share:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **id** | int(5) |
| 2 | **encloseid** | varchar(20) |
| 3 | **enclosename** | varchar(20) |
| 4 | **startPosition** | varchar(30) |
| 5 | **transoprt** | varchar(10) |
| 6 | **viewpoint** | varchar(3000) |

1. sub\_viewpoint:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **id** | int(5) |
| 2 | **main\_id** | int(5) |
| 3 | **place\_id** | varchar(30) |
| 4 | **type** | varchar(5) |
| 5 | **name** | varchar(50) |
| 6 | **lat** | varchar(15) |
| 7 | **lng** | varchar(15) |
| 8 | **weekday\_text** | varchar(300) |
| 9 | **phone** | varchar(20) |
| 10 | **rating** | varchar(5) |

1. sub\_viewpoint\_img:

| **名稱** | **型態** | **編碼與排序** |
| --- | --- | --- |
| 1 | **ID** | int(10) |
| 2 | **place\_id** | varchar(30) |
| 3 | **name** | varchar(50) |
| 4 | **url** | varchar(3000) |

1. 設計議題(Design Issue)

議題1：使用者儲存問題，當使用者切換頁面暫存之資料都會遺失。

原因：因為沒有使用者登入，所以沒用後端紀錄使用者資訊與操作。

解決方案：以本地端儲存使用者資訊

1.一開始使用WebSql，空間雖然大但速度太慢，所以淘汰。



2.使用Localstorage較快速，且儲存空間充裕。

議題2：google所提供的圖片許多與景點不相符合。

原因：沒有優美及開放資料的圖片，故圖片為自行上網抓圖(版權問題)，可能有侵犯著作權問題。

解決方案：

1.在圖片邊緣放上圖片網址，但應不美觀，所以棄用。



2.由img 之alt 屬性來說明圖片來源。

議題3：天氣與路徑演算法計算繁雜，比例難以拿捏。

原因：因要平衡天氣與路程和路況等等因素，讓推薦的排序能有一個平衡的標準，符合以人出發點作為思考。

解決方案：

1.只使用距離(公尺)與天氣概況作為運算，運算結果不合宜，所以棄用。



2.使用距離比例+時間與天氣概況+降雨機率+個人化，使結果更加符合以人出發點作為思考。

議題4："我的行程"內刪除行程造成行程中斷。

原因：因"我的行程"內的行程皆可讓使用者自行調整和刪除，將可能造成整天行程有中斷的情形。

解決方案：

1.清空“我的行程”所有景點，重新開始排序。



2將編輯完形成串聯，送回後端更新資料，回傳正確資訊給使者。



議題5：從api抓取與景點距離&時間資訊緩慢。

原因：透過Google Distance Matrix API與每個景點作運算十分緩慢(每個約需1秒)，會造成後端運算過久，效能極為低落。

解決方案：將每個景點間距離儲存至資料庫

1.純粹以四象限儲存，找出各象限與目前經緯度最近者，透過Google Distance MatrixAPI，在加上由其景點至其他景點距離，得其餘景點距離，但資料有些不夠精準。



2.以八方位象限儲存，找出各象限與目前經緯度最近者，透過Google Distance Matrix API，在加上由其景點至其他景點距離，得其餘景點距離，資料誤差極小，且運算速度與四象限相當。