

TravelScheduler 系統

組長: 109703041 邵靖翔

109703051 吳姜維

109703007 薛兆宸

109703052 黃欒生

109703054 莊維軒

109703058 曹柏泓

108305077 余玖熹

1. 系統功能

應用動機:

我們發現當今人們出去遊玩必備的一項工具就是 google map，但在 google map 上要規劃一個旅程卻是一件很痛苦的事情，於是我們想利用網頁前後端+資料庫+google map api 來將規劃與 google map 合二為一。

應用目的:

使用 google map api, google place api 等 api 加上我們自己設計的流程及功來實作一個以 google map 為底的旅遊規劃應用程式(目前以 web 形式實作，日後完成時會將 web 打包成一個漸進式網頁應用程式)。

應用使用流程:

使用者透過 google oauth 登入後，可自行創建旅程，創建旅程時會使用 google map api 讓使用者在 google map 上選擇地點，並且將地點加入旅程中。另外同樣會透過 google map api 幫使用者計算到往返地點的時間並且呈現路線，支援多種交通方式的計算，而已經結束的旅程使用者也可以標示為完成或是標示為收藏。

當使用者創建好旅程後可隨時透過旅程列表頁面查看已創建的旅程，創建好的旅程資訊以及旅程中的地點將會被存放在資料庫中。為確保一個地點在資料庫中只有一筆資料，我們將使用地點的某個唯一的特徵(例如地址)去做 hash 來做為當筆資料的 id。當資料庫中存放有使用者的地點後即可呈現出使用者去過的地點、未來會去的地點以及收藏的地點，可查看該地點的詳細情形，例如營業時間、評價等，其中使用者可為每個地點貼上一個標籤 tag，標示該地點的

功能，例如：餐廳、停車場等。

另外還有一項功能是團體，我們允許使用者透過個人與團體的 `id` 交換來獲得其個人或團體所擁有的旅程，達成共同擁有旅程的功能，未來也有考慮實作 `websocket` 來實現使用者共同編輯旅程的功能

其中包含五個實體：使用者 `user`，旅程 `travel`，地點 `spot`，團體 `group` 以及標籤 `tag`，除了 `user` 不能夠被新增修改刪除外，其餘實體皆有 `CRUD` 的功能，`group` 還另外有新增成員及剔除成員的功能。

2. 系統架構

前端 + 後端 + 資料庫：

依照主流的 `mvc` 架構進行開發，其中前端和後端以 `REST api` 溝通，後端和資料庫在同一台主機的不同 `port` 之間溝通，未來不排除將前端、後端和資料庫個別封裝進 `docker` 中來學習 `microservice` 的架構。

3. 系統程式語言、工具、模組

前端：

使用 `reactjs` 配合 `redux` 做中心化的資料管理，並且使用 `material UI` 來簡化繁瑣的 `css` 和 `rwd`，另外使用 `firebase` 的 `authentication` 功能來達成 `google` 的 `oauth` 登入功能。

*其餘: `react-router`, `react-jss`, `react-icon` 等

後端：

使用 `nodejs` 並使用 `expressjs` 框架進行開發，配合許多適用於後端的模組建構 `api`。

*其餘: `dotenv`, `body-parser`, `cors`, `session`, `helmet` 等

DBMS：

使用 `mysql` 做為資料庫，同時透過 `connection pool` 的概念去降低後端連線到 `mysql` 的成本，同時 `session` 也記錄在 `mysql` 中，節省前端傳的參數。