

# Web-Based Recording and Visualization Framework for Moving Trajectories

Po-An Yang<sup>1</sup> LiJung Chi<sup>1</sup>

Seth Chen<sup>2</sup> Jonathan Tsai<sup>2</sup>

Kun-Ta Chuang<sup>1</sup>

Yung-Chung Ku<sup>2</sup>

<sup>1</sup>Dept. of Computer Science and Information

Engineering

National Cheng Kung University

Tainan, Taiwan, R.O.C.

ktchuang@mail.ncku.edu.tw

<sup>2</sup>Innovative DigiTech-Enabled Applications &

Services Institute, IDEAS

Institute for Information Industry

Taipei, Taiwan, R.O.C.

{seth, jonathan, piperku}@iii.org.tw

## Abstract

隨著智慧型手機的普及，個人軌跡資料的收集越來越方便，有關軌跡的應用也越來越多，像是 Foursquare [1]、Human [2] 等 app 已陸續成為最火紅的手機應用。不過這些應用通常都只有很簡單的呈現打卡地點或軌跡，讓資料以靜態的出現，並沒有進一步的將資料視覺化，即使有，通常也要等到官方自行整合後釋出資料。我們提出的 Web-based 的系統架構，不僅可以即時動態呈現軌跡，更能跨平台，不受限於裝置環境，使用者可以透過瀏覽器來記錄自己的軌跡。軌跡儲存後，我們也提供一個即時呈現大家軌跡的移動動態呈現機制，不僅可以看見自己的軌跡，也可以看見他人的軌跡。另外有關於個人軌跡紀錄，通常使用者會有個人隱私的擔憂，在我們的架構中，會給每個使用者 UUID，透過 UUID 模糊了使用者與個人軌跡之間的連結，只有使用者的 UUID 及軌跡會留在 Database 中，系統不會知道使用者的 identity 資訊。最後我們也實作一些資料視覺化的技術，讓軌跡得以動態方式呈現。

**Keywords: Trajectory, GPS, Data Visualization.**

## 1. Introduction

有關於地圖軌跡的應用，已經是現在人不可缺少的一環，但多數的系統都偏向做導航或打卡等，鮮少有系統像世界迷霧 [3] 一樣，讓使用者以地圖為日誌，記錄自己的軌跡。而即使有這些系統，通常這樣的應用程式都無法跨平台，包含 iOS 及不同的 android 等作業系統，均須特別撰寫不同的應用程式。在本文中，我們提出並實作了一個 Web-based 的軌跡擷取及呈現的系統架構，讓使用者可以簡單的透過瀏覽器或手機 app 記錄自己的

軌跡，並且以資料視覺化的技術將軌跡呈現。

我們實作的平台是 **Github Pages** [4]，透過 **Github Pages**，在此設計中，我們完全使用現在熱門的雲端服務，完全不用架設自己的 **server**，只要將 **code** 放到指定的 **branch**，使用者就能很方便的使用我們的服務，並且讓未來需要這個系統架構的開發者或研究員，可以很方便的 **fork** 我們的作品，增加他們需要的功能。

另外在記錄軌跡的部份，有些裝置沒有 **GPS** 定位的晶片，因此我們的系統支援 **HTML5** 的技術透過 **wifi** 或行動網路來定位；如果裝置有 **GPS**，我們的系統會用 **GPS** 來定位較精準的位置。有關軌跡資料的儲存，我們系統用 **Firebase** [5]當作資料庫。**Firebase** 是目前矽谷很熱門的雲端資料庫系統，**Firebase** 讓我們可以專注在前端的開發，透過 **Firebase** 簡便的 **API**，我們的系統可以很容易的存取使用者的軌跡資料。在軌跡呈現的部份，地圖是用 **Open Street Map** [6]，**OSM** 是一個 **Crowdsourcing** 平台，由於圖資較其他開放式地圖完整，所以我們系統選用 **Open Street Map** 當底層地圖；軌跡是用 **Leaflet** [7]，這是一套 **Open-source JavaScript Library**，除了有很方便的 **API** 可以在地圖上畫路徑，更可以讓開發者在他們的架構下，增加自己需要的套件，以疊圖層的方式，將資料呈現在地圖上，我們會有兩層的 **layer**，一層是靜態的路線，一層是動態的路線移動。

以下我們將分別介紹系統的整體架構以及實作的細節，最後也會探討一些未來的加值應用。

## 2. Framework Overview

在 **Framework Overview** 中會介紹我們系統架構的流程，如下圖(一)。我們全部的架構都是以 **Web** 為基礎，如此可以很容易的跨平台。

首先使用支援 **HTML5** 的瀏覽器或是手機 **app** 來取得使用者的位置，由於現今瀏覽器大多支援 **HTML5**，所以可以方便我們跨平台做定位。接下透過我們系統架構中的 **GPS-Logger**<sup>1</sup> 記錄使用者的軌跡，當使用者按下 **Save** 後，**GPS-Logger** 會把軌跡傳至 **Firebase**。將 **Firebase** 獨立出來，可以很清楚的切割我們的架構，透過 **Firebase** 連結前面的 **GPS-Logger** 收集軌跡資料，及後面的 **Trajectory Visualization** 軌跡呈現，**Firebase** 讓我們不需要任何伺服器編碼，就能將整套系統即時呈現。最後使用者進入 **Trajectory Visualization**，便能馬上看到自己動態的軌跡。

在 **Part3** 中，我們會介紹 **GPS-Logger** 如何運作，如何記錄使用者的軌跡。在 **Part4** 中，我

---

<sup>1</sup> **GPS-Logger** : <http://frankyang0529.github.io/gps-log/>

們會介紹資料儲存的格式為何，以及如何存入跟讀取軌跡。在 part5 中，我們會介紹 GPS-Path-View<sup>2</sup>如何以新穎的方式，將軌跡資料呈現。在 Part6 中，我們提出一些未來展望，會介紹幾個有潛力的研究方向，除了前端的 Data Visualization 還有後端的 Data Mining，讓 Trajectory 的應用更豐富。最後在 Part7 中，我們會對我們的系統下結論，並且提出我們對未來 LBS 應用趨勢的看法。



圖(一) Our Framework

### 3. Implementation of GPS-Logger

在 GPS-Logger 中，我們定位是使用 HTML5 Geolocation API [8]，這是由 World Wide Web Consortium(W3C) [9]所訂定的標準，由於這套 API 的支援度很廣泛，所以可以滿足我們跨平台的需求。定位前會先詢問使用者是否願意讓我們的系統存取位置資料，如果使用者同意，Geolocation API 會透過 IP address、Wi-Fi、Bluetooth MAC address、radio-frequency identification(RFID)、Global Position System(GPS)或 GSM/CDMA cell ID 去查詢使用者的位置，並且回傳精準度最高的位置資訊。目前多數瀏覽器都有支援這套 API，像是 Internet Explorer 9、Firefox 3.5、Chrome、Safari 5 及 Opera 10.6 以後的版本都有支援這套標準。Geolocation API 有很多細節可以操作，可以讓使用上更為靈活，像是有個參數是 enableHighAccuracy，讓開發者可以選擇是否取得較精準的位置，如果選擇 true，定位上會比較久，但位置資訊較為精準。

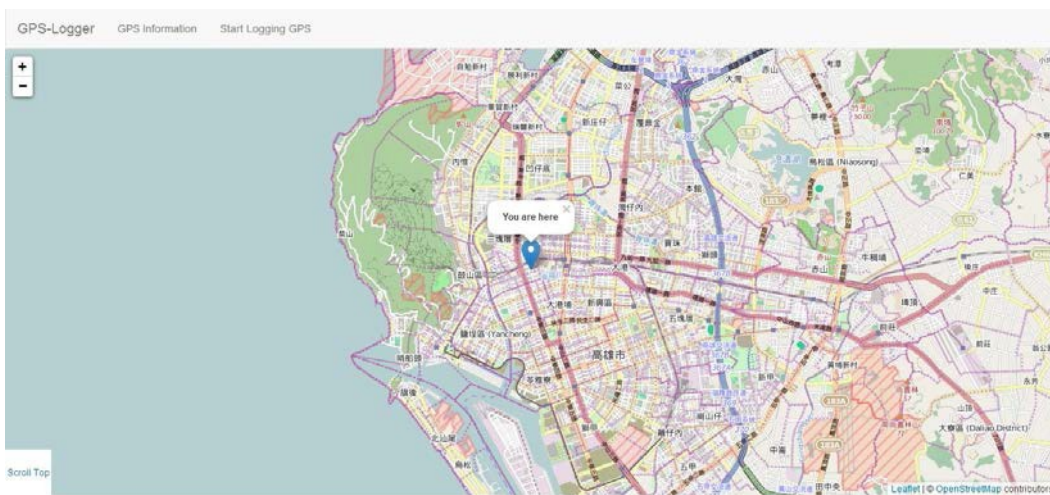
使用者進入我們系統架構的當下，系統就會透過 random 函數製造一個長度為 36 的 Universally Unique Identifier(UUID)，UUID 是通訊唯一識別碼，藉此來隱藏使用者的真實資訊，達到匿名的效果，當使用者重新開啟 GPS-Logger 時，都會重新分配 UUID，讓使用

---

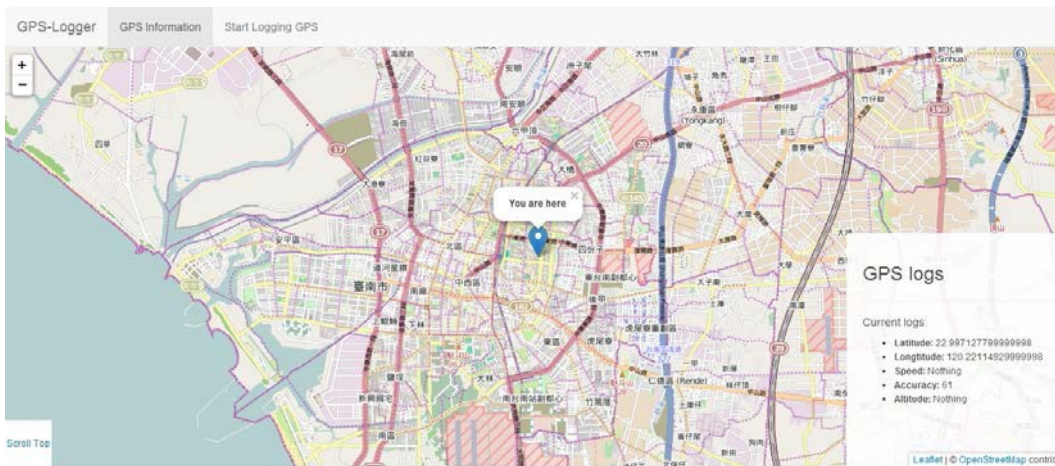
<sup>2</sup> GPS-Path-View : <http://chilijung.github.io/gps-path-view/>

者資訊更不容易被查詢。UUID 包含了 32 個十六進位數字，並且用 '-' 區分成 5 組，個數分別是 8, 4, 4, 4, 12，像是 523512e4-7b50-ee94-9475-f5dac6fe1cca，UUID 全部大約有  $3.4 \times 10^{38}$  個，由於數量級龐大，所以有心人士並不容易透過我們的系統找到使用者的真實身份，即使每奈秒( $10^{-9}$ 秒)生產一兆個 UUID，也要百億年才用的完。

當使用者取得 UUID 並且同意讓網站取得位置資料後，我們將會在地圖上標示出使用者的位置，如下圖(二)。如果點選了 GPS Information 則會顯示現在的經緯度、移動速度、位置誤差及高度資訊，其中經緯度的單位是 degree，位置誤差及高度是 meter，移動速度則是 meter/second。

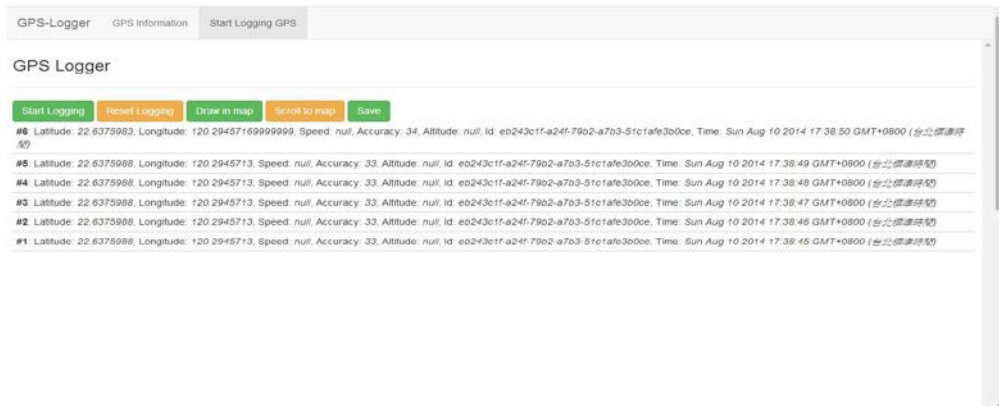


圖(二) GPS-Logger



圖(三) GPS-Information

當使用者按下 **Start Logging GPS**，會跳到另一頁面，如下圖四。按下 **Start Logging** 後，則會紀錄下每秒 GPS 的資訊，使用者也可以在頁面上看到自己的即時定位，此時資料都還沒上傳到 Database，所有記錄都還只是在本地端，當使用者按下 **Save** 後，資料才會上傳到我們的 Database；如果使用者不希望我們存取資料，也可以選擇 **Draw in map**，GPS-Logger 也會以動態的方式將資料呈現再本地端。

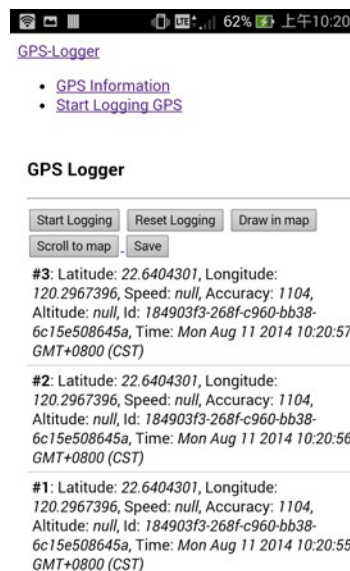


圖(四) Start Logging GPS

手機版的 app，我們是透過 phonegap [10]把 HTML、CSS 及 JavaScript 轉成 Android app，運作情況如下圖五及圖六。



圖(五) GPS-Logger app



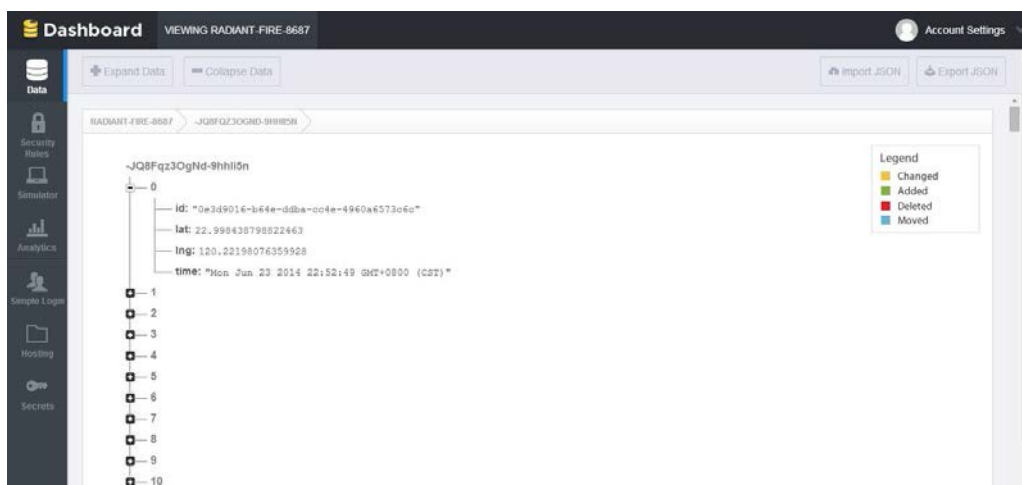
圖(六) GPS-Logger app Start Logging

## 4. Trajectory Database

Firestore 是目前矽谷最熱門雲端資料庫系統，有很多新興的即時(real-time)應用程式都以 Firestore 當資料庫，Firestore 讓開發者不用分心於 server 架設，就能同時容納數萬人上線，使開發人員可以專注於前端的開發，而不用擔心後端或是資料庫的狀況，即使是免費的方案，也能擁有 5GB 的流量及 100GB 的儲存空間，這項服務最好的地方就是，將資料存取和資料轉換在單一系統內合併，以前，資料傳送和存取原本是兩件分開的事，但現在兩者已結合成資料同步。

透過 Firestore，我們可以很方便的存取資料，GPS-Logger 每秒記錄一個點，當使用者按下 Save 後，我們會將本地端的資料都存入 Database，同時也將本地端的資料都清除，所以當使用者下次按下 Save 時，並不會存入一樣的資料，但是 UUID 是相同的，每次 Save 都會獨立成一條軌跡。

傳送到 Database 的資料是 array of json objects，json object 中包含了 id、lat、lng 及 time，id 是使用者的 UUID，lat 是緯度，lng 是經度，time 是時間資料，是由 javascript 的 Date object 轉成 string，當這些資料以 json object 儲存後，在往後的 Data Visualization 或是分析都可以很方便的擷取資料。當使用者每次儲存資料在 Firestore 時，Firestore 會亂數給一個編碼，像是-JQ8Fqz3OgNd-9hhli5n，而此編碼底下就是使用者存入的資料。在 Firestore 的網站，也可以很容易的察看資料庫中的資料，如下圖七。



圖(七) Firestore 資料庫查詢介面

## 5. Trajectory Visualization

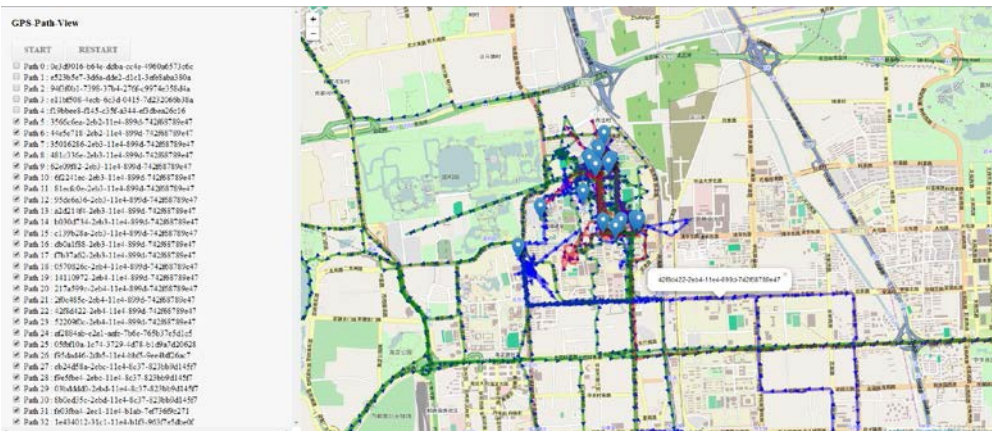
在 Data Visualization 中，我們使用的地圖是 Open Street Map，OSM 是目前圖資最豐富，

且更新最快的開放式地圖，顯示軌跡的圖層是 Leaflet，Leaflet 是很輕的 JavaScript Library，擁有豐富的畫圖 API，可以在地圖上畫 path、circle、polygon 等等，並且也有廣大的社群開發許多 Leaflet 套件，讓 Leaflet 功能更為廣泛，路徑上的箭頭與動畫即是 Leaflet 的插件，分別是 Leaflet.TextPath [11]及 Leaflet.AnimatedMarker [12]，點擊路徑時，會彈出一個對話框，告訴使用者路徑編號，而對話框也是 Leaflet 本身的 API。

在實作上，首先，我們會從 Firebase 取得全部資料，依序讀取 Firebase 隨機編號下的資料存入本地端的變數，再來將同一隨機編號底下的 Trajectory 集成成一條路徑，並加上包含 UUID 的對話框，最後會把路徑加上箭頭、做成動畫並產生 checkbox，讓使用者選擇是否把該路徑繪在自己的地圖上。Checkbox 的預設都是不勾選的，當使用者勾選某一條路徑時，系統會將地圖畫面移至該條路徑的起點。

在網頁的左上角有 START 跟 RESTART，按下 START，則地圖上的藍色標籤會開始沿著有勾選的路徑跑，當標籤在移動時，checkbox 會鎖起來，不能再新增或取消勾選，所以路徑無法從地圖上新增或移除；當使用者按下 RESTART，則會解鎖 checkbox，並將座標移回該條線的起點。

我們使用的 Trajectory 除了從 GPS-Logger 中收集的之外，還有從 Microsoft Research - GeoLife GPS Trajectories [13]取得軌跡，並且也能使用 My tracks<sup>3</sup>記錄軌跡，在透過我們的 kml 轉換器<sup>4</sup>，將軌跡上傳 Firebase。下圖八是 GPS-Path-View 的整個介面，下圖九跟十是部份 GeoLife GPS Trajectories 的 Data Visualization。



圖(八) GPS-Path-View with 170 Trajectories in Microsoft Research – GeoLife GPS Trajectories

<sup>3</sup> My Tracks: <https://play.google.com/store/apps/details?id=com.google.android.maps.mytracks>

<sup>4</sup> Kml 轉換器: <http://kml-parse.herokuapp.com/kmz/>



圖(九) GPS-Path-View Zoom out



圖(十) GPS-Path-View Zoom in

## 6. Future Work

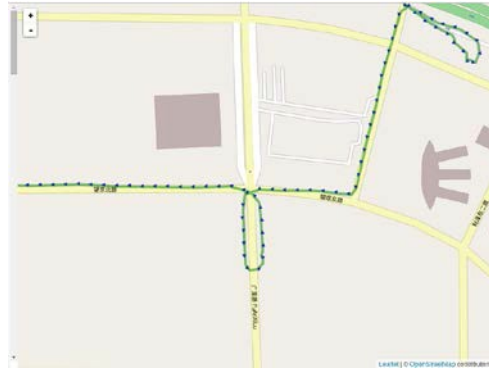
使用者軌跡記錄及分析近年來已經普遍用於運動、健康等應用中。例如 NIKE+ RUNNING [14]，即為一個很熱門的運動路徑記錄 APP。然而目前這類軟體均為 APP-based 的架構而非 Web-based，在使用上多少會因為作業系統版本間的差異，而有系統更新的困難。我們認為，我們提出利用 Web-based 架構能克服平台差異造成的問題，並且我們使用的服務平台 Github Pages 及 Firebase 都是免費的，可以減輕開發者或研究員的負擔。後續利用我們所提出的軌跡記錄架構，除了能做為相關運動健康應用，我們正著手於以下應用上的開發。

首先，我們將針對 Open Street Map 的推廣跟便利度這方面來著手，加大 Open Source 平台的編輯便利度。OSM 是一個 Crowdsourcing 的平台，由於地圖的資料均由熱心的圖客們 (Mapper) 貢獻完成，如何讓圖客們很容易的了解某個地方可能有新的道路更新，包含道路新增，是一個很重要的課題。舉例來說，下圖十一是還沒加上路徑的 Open Street Map，下圖十二是標上路徑以後的 Open Street Map，我們發現本來地圖上沒有的路，但是卻有車子走過，表示這個地方可能有路，細察之後，我們發現這是北京市望京北路及廣澤路的交叉路口，配合比對 Google Map (下圖十三)，則會發現該地區路況已經變了，廣澤路在 Open Street Map 上只有一條道路，但在 Google Map 上卻有兩條道路，由此現象可以觀察到，未來我們可以將路徑資料與地圖資料做比對，當有一定數量的車子從原本地圖上沒有路的地方走過時，表示該地區有可能道路已經變更，如此可以增進 Open Street Map 改善的速度，讓 Open Street Map 的圖資更為完整。





圖(十一) Open Street Map



圖(十二) OSM with Trajectory



圖(十三) Google Map

此外，軌跡探勘技術已經被視為下一個階段的重要研究課題。研究上通常會透過評量軌跡之間的相似度來作為下一步應用的先備知識，然而要完成這項評量是非常困難的，因為軌跡的長度及空間上的關聯十分難用傳統的尤拉距離計算來評斷。我們正進行將軌跡轉換為使用者行動的 POI 的方式來提供研究人員進行 semantic 上的分析。例如一個使用者的軌跡，我們能轉換為”孔廟停留 30 分鐘”->”安平古堡停留 30 分鐘”。如此我們將更能提供研究技術人員提出更容易使用者個人化行銷或推荐等應用。此兩方面將是我們接下來著手的挑戰。

## 7. Conclusion

在本文中，我們提出了一個 Web-based 的軌跡擷取及資料即時呈現的雲端架構。透過現有的雲端服務，使用者可以不受限於手持式裝置的作業系統，均可便利的記錄其移動的細

節。我們認為，軌跡加值應用將是未來十分重要的一個 LBS 應用趨勢，透過資料的呈現及分析，可以提供個人化加值服務。我們也提出了一些未來的實作細節，包含將使用者的停留資訊擷取出來的的方法，以便後續的 data mining 分析過程，可以用 semantic 的角度來進行分析動作，我們預期能顯著的提高應用服務的精準度。

## Acknowledgments

The authors especially thank the Taiwan Ministry of Economic Affairs and Institute for Information Industry for financially supporting this research : “Plan title : Fundamental Industrial Technology Development Program (2/4)”.

## Reference

- [1] “Taipei | Food, Nightlife, Entertainment.” Available: <https://foursquare.com/>.
- [2] “Human - Activity Tracker: track walking, running, and biking.” Available: <http://human.co/>.
- [3] “世界迷霧 | 去實現你環遊世界的夢想吧!” Available: <http://zh-hant.fogofworld.com/>.
- [4] “GitHub Pages,” 15-Aug-2014. Available: <https://pages.github.com/>.
- [5] “Firebase - Build Realtime Apps.” Available: <https://www.firebase.com/>.
- [6] “OpenStreetMap.” Available: <http://www.openstreetmap.org/#map=5/51.509/-0.088>.
- [7] “Leaflet — an open-source JavaScript library for interactive maps.” Available: <http://leafletjs.com/>.
- [8] “Geolocation API Specification.” Available: <http://dev.w3.org/geo/api/spec-source.html>.
- [9] “World Wide Web Consortium (W3C).” Available: <http://www.w3.org/>.
- [10] “PhoneGap | Home.” Available: <http://phonegap.com/>.
- [11] “makinacorp/Leaflet.TextPath,” *GitHub*. Available: <https://github.com/makinacorp/Leaflet.TextPath>.
- [12] “openplans/Leaflet.AnimatedMarker,” *GitHub*. Available: <https://github.com/openplans/Leaflet.AnimatedMarker>.
- [13] “GeoLife GPS Trajectories.” Available: <http://research.microsoft.com/apps/mobile/download.aspx?p=b16d359d-d164-469e-9fd4-daa38f2b2e13>.
- [14] “Nike Running,” *Nike.com*. Available: [http://www.nike.com/tw/zh\\_tw/c/running](http://www.nike.com/tw/zh_tw/c/running).