CMoney 菁英軟體工程師戰鬥營
期末專題成果報告書

專題名稱: 危機地圖 **DangerMap**

團隊成員:張文煥、洪詩喻、張品楷、陳靖旻

目錄

一、專案簡介

動機:

世風日下, 社會案件層出不窮, 民眾無法預知危機的發生, 且在資訊不對稱的情況下, 當周遭有危害發生時, 也無法第一時間知曉, 只能之後在社群平台上瀏覽零星的貼文, 因此本團隊想透過平台整合政府資料及民眾上傳案件, 將全台的犯罪事件及交通事故案件依照區域地理位置進行分類, 並利用數據視覺化的方式呈現在地圖上, 讓使用者可以清楚明瞭附近正在發生的事件, 並提高警覺來預防事件再度地發生。

問題:

為什麼使用者無法保障生命安全,為什麼使用者警覺性不足

問題的成因:

1. **不知道哪裡有危險**
2. **危機不定期發生**
3. **警消資源不足無法應付所有突發狀況**
4. **已經遇到危險, 無法求救**
5. **其他**

找出成因的解法:

1. **告知使用者哪裡有危險**
2. **即時告知危險發生**
3. **協助警消, 人人都可通報**
4. **安全觀念宣導**
5. **他人協助報案或自行求救**

執行:

1. **告知使用者哪裡有危險**
   - **利用地圖和事件點讓使用者知道哪裡有危險**
   - **連結政府資料**
2. **危機不定期發生**
   - **即時收到危險資訊**
   - **聽廣播**
   - **口耳相傳**
   - **親眼目睹**
3. **警消資源不足無法應付所有突發狀況**
   - **即時上傳當前資訊**
   - **即時報警**
4. **已經遇到危險, 無法求救**
   - **跑馬燈宣傳**
   - **危機處理 SOP**

- 路口死亡人數

二、開發環境

※硬體環境

規格	Zephyrus G14	MSI Prestige 15	MacBook Air	ACER SwiftX
處理器	AMD Ryzen 7-4800HS	Intel Core i7-1185G7	Apple M1	AMD R7 - 5800U
作業系統	Windows10	Windows 10	macOS Big Sur 11.6	Windows10
記憶體	16GB	8GB * 2 DDR4-3200	16GB	16GB
資料儲存	1TB SSD	1 TB SSD	256GB SSD	512GB SSD
顯示晶片	Nvidia GeForce GTX 1660Ti 6G	NVIDIA GTX 1650 Max-Q	Retina 顯示器	Nvidia GeForce RTX 3050Ti

※開發軟體

xcode
Visual studio
Android studio

※使用語言

Swift

c#

kotlin

三、需求分析

1. 事件地圖

   進入 App 時所見的頁面，其中又分為即時事件地圖以及累計事件地圖兩種顯示，即時事件地圖主要供使用者檢視當下附近範圍發生的事故事件。非即時的累計事件地圖主要用來顯示指定區域於過去發生指定事件多寡的等級，等級預計以顏色或是聊天式通知來區分。

   即時事件地圖：

   1. 顯示使用者當下地點
   2. 顯示當下附近事件（全部種類，每個種類以不同 icon 圖釘區分）
   3. 使用者可以在地圖上上傳事件
   4. 地圖上的每個事件都是一個留言版

   累計事件地圖：

   1. 依照標籤來區分事件種類，以區塊顏色來區分事件發生頻率等級
      或是像聊天室的形式跳通知
      - 所有事件
      - 各式犯罪事件
      - 累計數量
   2. 資料以一年為累計基礎

   頁面中同時可導向上傳事件、留言版、會員系統等功能。

   跑馬燈：安全宣導

2. 事件上傳

   使用者上傳目擊事件，填上相關資訊並上傳，上傳內容：

   - 類型
   - 標題
   - 內容：可自行決定是否上傳照片
   - 地點描述：地點補充描述
   - 目擊時間：預設當下時間,可再自行調整

   隨時間流逝消失，24小時內無人更新則消失

3. 推播

   偵測使用者所在地點，若靠近事件地點會自動推播通知
   - 附近事件告知
   - 累積資料接近告知

4. 會員系統

需要註冊成為會員才能使用上傳事件的功能, 系統包含:

- 註冊(串接 FB or Google or Twitter, Ask Stan)
- 名稱
- Email
- 密碼
- Email 認證
- 上傳權限開通
- 附議權限開通
- 上傳過的事件紀錄
- 選擇頭像: 初始三款
- 他人也可看不同會員的資料

5. 事件附議

   點擊事件 icon 後, 如同樣目擊該事件可按附議證明事件發生

6. 留言版

   使用者可點擊地圖上之物件參與討論, 使用者可以在上面討論、追蹤事件, 目擊者也可以在討論版更新事件最新狀況, 目擊者發言有標示該使用者為目擊者。

7. 求助功能

   自定義呼叫功能, 緊急撥號或是自定義呼救鈴聲

8. 募集目擊區

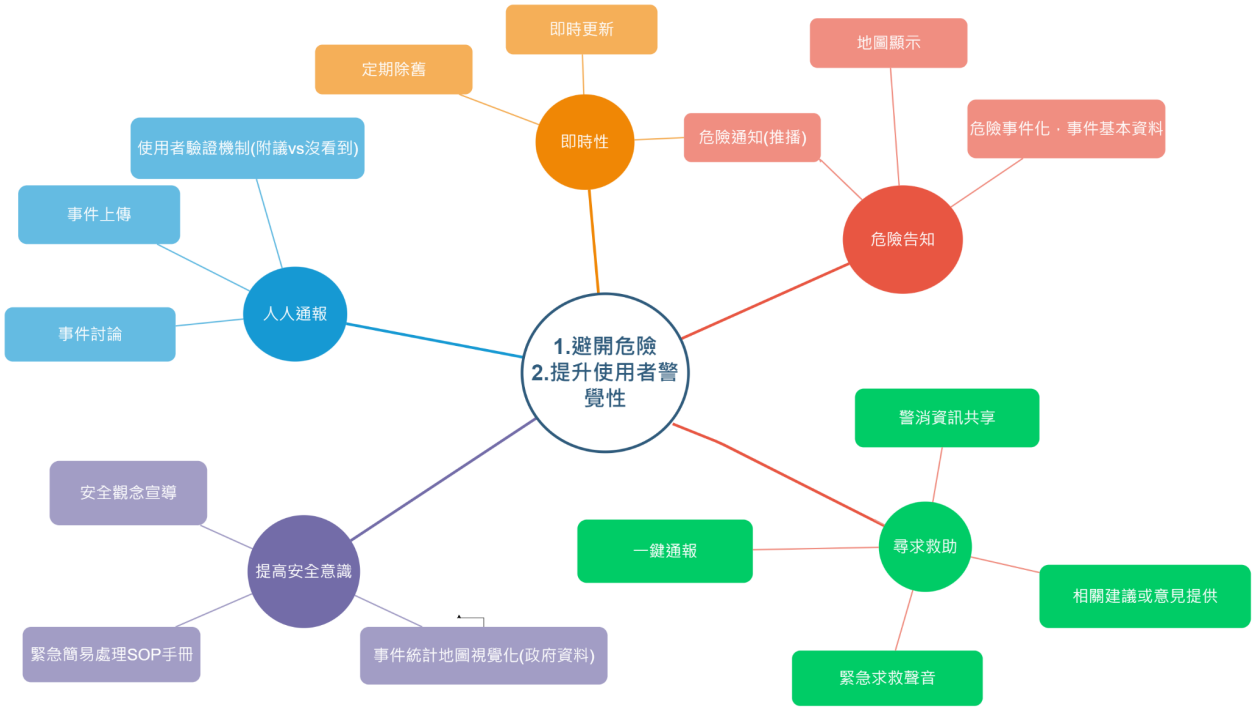
   招募特定地點之目擊資訊(ex行車紀錄器), 由使用者上傳相關資訊提供給有需求的使用者

9. 會員積分 **&** 會員頭貼 **&** 相框 **&** 成就

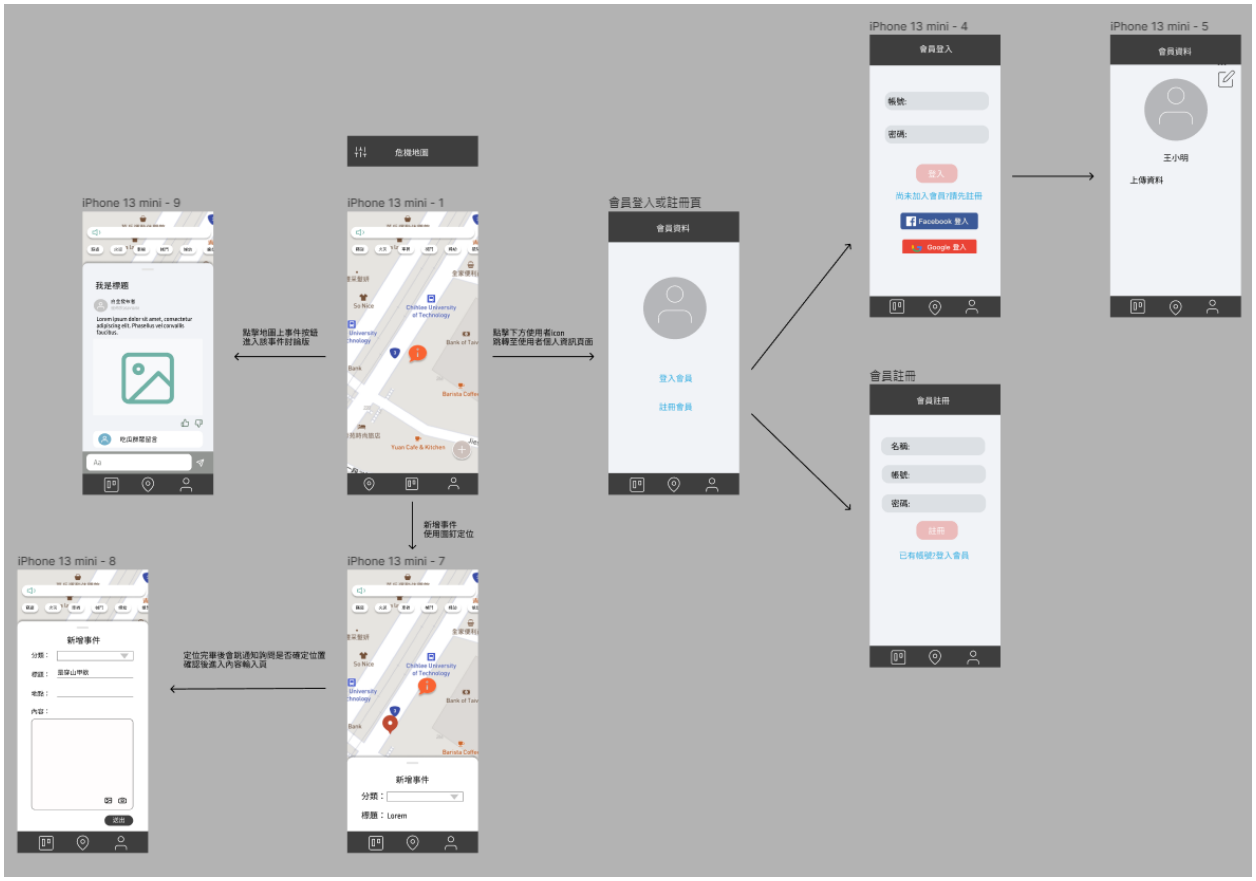
   鼓勵使用者上傳資訊並提供回饋

四、系統架構

- 功能架構圖


1.避開危險
2.提升使用者警覺性
即時性
即時更新
定期除舊
危險通知(推播)
危險告知
地圖顯示
危險事件化，事件基本資料
人人通報
使用者驗證機制(附議vs沒看到)
事件上傳
事件討論
提高安全意識
安全觀念宣導
緊急簡易處理SOP手冊
事件統計地圖視覺化(政府資料)
尋求救助
警消資訊共享
一鍵通報
相關建議或意見提供
緊急求救聲音

- 原型流程圖


危機地圖
iPhone 13 mini - 9
iPhone 13 mini - 1
會員登入或註冊頁
iPhone 13 mini - 4
iPhone 13 mini - 5
會員登入
會員資料
登入會員
註冊會員
會員註冊
點擊地圖上事件按鈕
進入故事件封鎖版
點擊下方使用者icon
跳轉至使用者個人資訊頁面
新增事件
使用圖釘定位
iPhone 13 mini - 8
iPhone 13 mini - 7
新增事件
定位完畢後會跳通知詢問是否確定位置
確認後進入內容輸入頁

五、團隊組成與分工

WBS	Task description	Start date	Finish date	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6
wbs	task	start	finish	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
1	**GitLab**	**12/13/2021**	**12/14/2021**																									
1.1	熟悉 GitLab	12/13/2021	12/14/2021																									
2	拉介面	12/13/2021	12/14/2021																									
3	非即時資料傳輸	**12/13/2021**	**12/15/2021**																									
3.1	資料需求確認	12/13/2021	12/13/2021																									
3.2	資料來源確認	12/13/2021	12/13/2021																									
3.3	資料格式確認	12/13/2021	12/13/2021																									
3.4	資料表建立	12/14/2021	12/14/2021																									
3.5	資料庫建立	12/14/2021	12/14/2021																									
3.6	資料表正規化	12/15/2021	12/15/2021																									
4	即時資料傳輸	**12/16/2021**	**12/17/2021**																									
4.1	資料表建立	12/16/2021	12/16/2021																									
4.2	資料表正規化	12/17/2021	12/17/2021																									
5	地圖串接(定位)	12/14/2021	12/17/2021																									
6	回傳上傳事件到後端(建立事件)	12/16/2021	12/17/2021																									
7	篩選地圖呈現	12/18/2021	12/19/2021																									
8	**API 建立**	**12/18/2021**	**12/24/2021**																									
8.1	API 需求確定	12/18/2021	12/20/2021																									
8.2	API-搜尋資料(GET)	12/21/2021	12/21/2021																									
8.3	API-上傳資料	12/22/	12/22/																									

	(POST)	2021	2021	
8.4	API-更改資料(PUT)	12/23/2021	12/23/2021	
8.5	API-刪除資料(DELETE)	12/24/2021	12/24/2021	
9	資料庫 **IO &** 快取	12/25/2021	12/25/2021	
10	串接api	12/25/2021	12/25/2021	
11	研究推播token	12/20/2021	12/21/2021	
12	研究非即時區塊顯示	12/20/2021	12/22/2021	
13	推播功能	**12/21/2021**	**12/26/2021**	
13.1	熟悉推播性質	12/21/2021	12/23/2021	
13.2	使用Token串接前段	12/24/2021	12/26/2021	
14	研究程式背景運作	12/22/2021	12/25/2021	
15	研究跑馬燈、點擊url	12/26/2021	12/28/2021	
16	跑馬燈功能	**12/26/2021**	**12/28/2021**	
16.1	跑馬燈資訊確定	12/26/2021	12/26/2021	
16.2	跑馬燈資料表建立	12/27/2021	12/27/2021	
16.3	跑馬燈資訊爬蟲	12/28/2021	12/28/2021	
17	即時留言板	**12/27/2021**	**12/28/2021**	
17.1	WebSocket 研究	12/27/2021	12/27/2021	
17.2	WebSocket 實現	12/28/2021	12/28/2021	
18	會員系統(頭像、會員資料、第三方註冊)到後上傳	12/29/2021	12/30/2021	
19	會員系統建置	**12/29/2021**	**12/31/2021**	
19.1	串接 FB Google 登入	12/29/2021	12/29/2021	

19.2	JWT登入驗證	12/29/2021	12/30/2021	
19.3	資料表建立	12/30/2021	12/30/2021	
19.4	email 認證	12/31/2021	12/31/2021	
20	留言板事件覆議處理	12/31/2021	1/1/2022	
21	雲端資料庫	**1/1/2022**	**1/2/2022**	
21.1	圖片資料庫	1/1/2022	1/1/2022	
21.2	影片資料庫	1/2/2022	1/2/2022	
22	測試Debug	1/2/2022	1/6/2022	
23	測試	**1/3/2022**	**1/6/2022**	
23.1	壓力測試	1/3/2022	1/3/2022	
23.2	使用者測試	1/4/2022	1/4/2022	
23.3	Debug	1/5/2022	1/6/2022	

六、實作方法

(一)會員驗證
使用JWT的方式, 讓有登入的會員得到一個Token, 並由前端傳遞此Token至後端來做不同API的驗證, 此外, 我們自行製作一個驗證機制API, 僅有到信箱收信的會員才算完成驗證。我們將會員分成三種等級, 第一個是完全沒有註冊的使用者, 儘可以觀看資料;第二個是有註冊但未驗證的會員, 可以留言但不能上傳事件、也不能對事件進行附議;最後則是有註冊且有驗證的會員, 則可以開通所有服務。另外, 針對JWT有失效可能, 我們也使用了Refresh Token的方式協助使用者保持登入狀態。

(二)推播系統
使用Firebase三方所提供的API進行推播, 推播可以分為兩大類, 第一種為即時性推播, 在使用者周遭有大於一定數量的事件、一定數量的車禍, 或是犯罪的高風險區時, 將會對使用者進行一個推播通知;此外, 我們也使用GCP的Cloud Scheduler來在晚上特定時間, 對使用者進行一個夜歸提醒的通知。比較可惜的是, 目前只有安卓系統提供推播功能, 因為IOS的推播必須要申請開發者帳號才得以進行。

(三)圖標聚落
由於我們的事件是透過座標分布在各個地圖上, 有時候較為密集的事件將會有些許重疊或是過於密集而不易觀察的情形, 為此, 我們使用了圖標聚落的功能, 將事件隨著畫面縮放, 顯示筆數, 如此一來, 就可以清楚知道該地區有多少起事件, 而將地圖放大也能正常顯示出各個事件的詳細分布。

(四)資料表正規化
對於每一個事件與其對應的發文者、發文內容、發文時間等等，若是讓每張表都有完整資訊, 其時容易造成過多的重複資訊, 因此我們透過關聯的方式, 讓每張表僅擁有最小必要資訊, 並且符合一個欄位一格資訊的方式來設計資料表, 直到API設計時, 再將這些表透過join的方式回傳整合資訊。

(五)輸入資料之驗證
關於使用者的任一輸入, 像是帳號、信箱、暱稱、大頭貼URL等等, 會有格式的需求、字數的長度、甚至是禁止出現的符號等等。實作方式除了前端會先做一層驗證(密碼則會使用MD5進行加密), 後端也會使用標籤的方式來做第二層格式的驗證。

七、專案成果


危機地圖
REGENT'S PARK
Euston Station
University College
Wallace Collection
National Portrait Gallery
Academy of Arts
ST. JAMES'S
Green Park
St. James's Palace
GREEN PARK
ST. JAMES'S
圖1:Logo

圖2:登入畫面

12:47

危機地圖

地圖　帳號

新增事件

事件標題: 標題

事件種類: 請選擇分類

地點描述: 描述事件地點

內文

送出

圖3:主畫面與新增事件畫面

圖4:事件與留言畫面

圖5:帳號資訊畫面


12:44
Back
CNR 晉豪汽車(保
底盤煞車冷氣維修
三民路
三民路
宏泰汽車鈑金烤漆
廠/保險理賠申請
103
家禾汽車大樓隔熱紙
三民公園
台灣中油蘆洲
三民路站(直營
蘆洲 戴師傅 修鞋達人
居家用品店
103
死亡人數：0
受傷人數：1
新北市政府消防
局鷺江消防分隊
仁愛街
永平街32巷
振宇五金 蘆洲店
五金行
103
宜加寵物生活館 長榮店
寵物用品
7-ELEVEN 新
便利商店
好食肌 · 蘆洲
G&T GoodTime
健康食品
長樂路237巷
漣福水果城
蔬果店
Google
地圖
帳號

圖6:交通事故畫面

12:44

Back

地圖

帳號

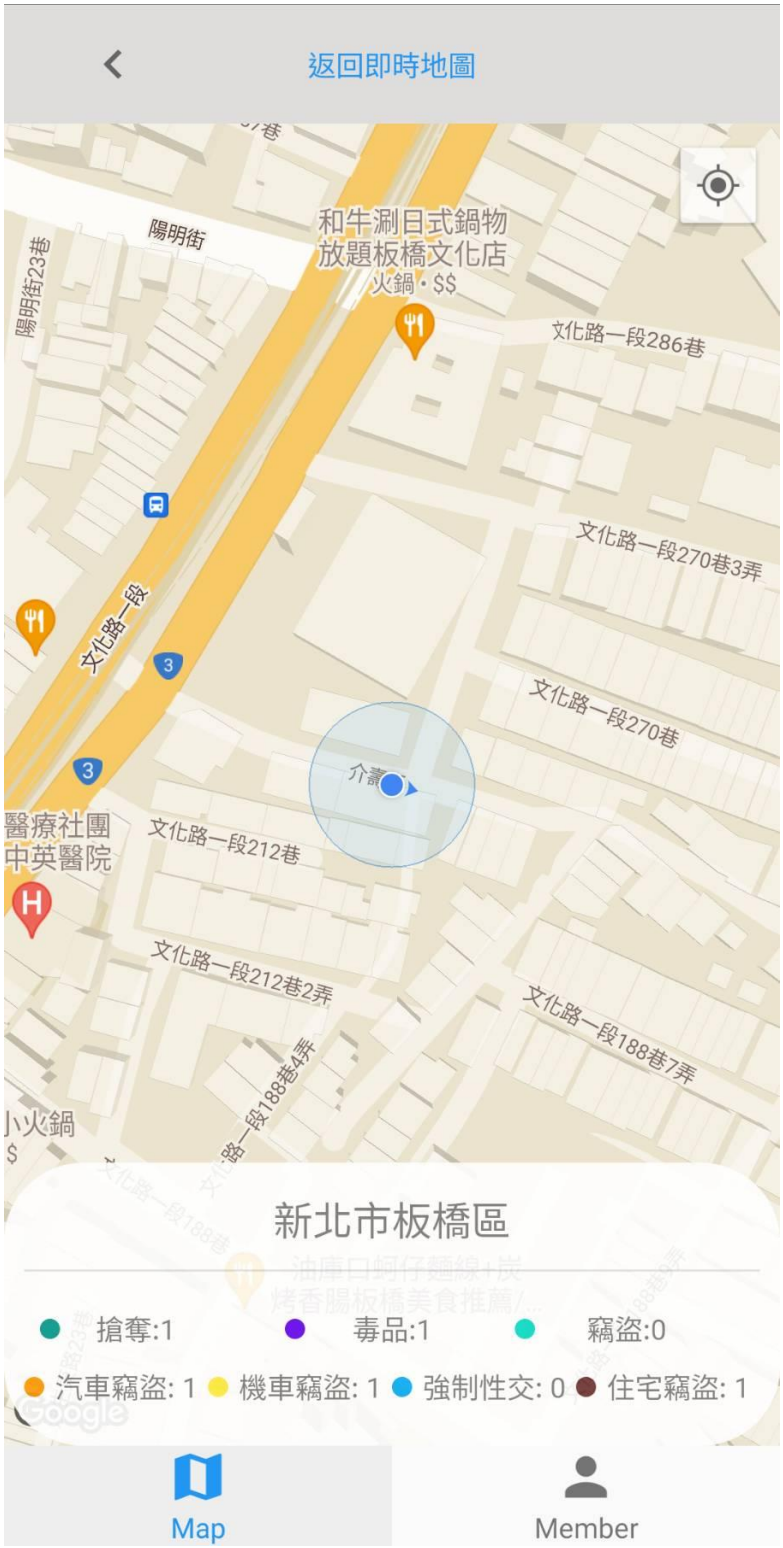


圖7:犯罪統計畫面ios與安卓


網際網路
專注模式
Mi Smart Ban··
無線基地台
Android 系統
已連接 USB 偵錯工具
輕觸即可關閉 USB 偵錯功能
危機地圖警報！ • 現在
你已抵達新北市板橋區，請注意安全
危機地圖警報！ • 現在
請小心，這裡是交通事故熱點
危機地圖警報！ • 現在
你身邊有15起事件正在發生
靜音
正在透過 USB 為這個裝置充電

圖8:推播畫面

八、討論與結論

根據製作過程與發表回饋, 我們歸結出以下三個問題。

一、資料傳遞即時性
目前我們在畫面更新上, 都是使用者必須操作特定動作, 像是留言、上傳事件或是移動畫面等等才會更新事件資料, 然而若是使用者未做任何動作, 將會導致資料不同步。目前想到的解決方法是透過輪詢的方式, 一定期間來更新資料, 然而即使如此, 也有其他效能上的疑慮, 的確是一個值得省思的問題。

二、前後端的溝通
即使有了API文件, 事實的溝通與討論仍然是前後端橋樑不可或缺的步驟, 以我們開發的過程為例, 前端使用API時, 對於回傳資料的型態有各種選擇, 後端習慣回傳Status code當作呼叫成功與否的依據, 然而前端卻未必理解每一個code的實際情形, 導致必須花費更多心力去解讀資料。因此, 我們認為開發時, 前後端除了討論API的需求, 對於傳入與回傳的資料型態與其意涵, 也都必須一併討論, 以求有一致的標準。

三、可以再更新之功能
我們整合各方意見, 最後匯集出以下作為未來擴充功能上可以著墨之處, (1)事件座標的代入, 點集Google上既有的圖示, 即可帶入該地之座標與地點名稱等等。(2)事件真偽判斷, 有些使用者認為還是必須要有照片, 較能防止隨意上傳的事件產生，換句話說，上傳事件時，照片是必須的而非選填之選項。(3)事件解決功能，當部分事件結束並且無異議時, 應該要能將事件清除掉, 以避免造成使用者誤會, 這部分有許多爭議, 一來我們不可能親自人工審核事件是否被解決, 二來或許有些使用者會想觀看過去發生的事件, 對此, 我們認為可以針對政府工作等類型的事件(如鋪路)來實踐事件解決之機制。(4)個人檔案功能, 針對各個使用者來記錄其發過哪些事件, 製成事件紀錄表作為統計並回饋使用者;另外也可以新增追蹤功能, 每當事件有新的進度時, 有追蹤之使用者便會收到通知。

最後, 對於本專案尚有許多可以更加精進之處, 但也很高興能夠透過這支APP理解各個使用者對於周遭事件的不同看法, 希望未來有機會能夠真正的實踐危機地圖, 讓所有民眾都能編織屬於大家的社會安全網。