

```
# 產品需求文件 (Product Requirements Document)
```

```
## 專案名稱：社區守護者 - 志工巡守追蹤系統
```

```
**版本**：1.0
```

```
**更新日期**：2026-01-15
```

```
**專案負責人**：Daniel Kai
```

```
**產品類型**：行動應用程式 (iOS / Android)
```

```
**技術棧**：React Native (Expo SDK 51) + Firebase + TypeScript
```

```
---
```

```
## 一、產品概述
```

```
### 1.1 產品願景
```

社區守護者是一個創新的志工巡守追蹤系統，旨在透過智慧型手機的藍牙技術與 GPS 定位，協助社區志工即時監測長者活動狀態，提供主動式的關懷服務。系統能夠自動偵測長者佩戴的藍牙信標 (Beacon)，並在特定情境下觸發警報通知，確保長者安全。

```
### 1.2 核心價值
```

- **主動關懷**：無需長者操作，被動式監測活動狀態
- **即時預警**：邊界點警報、不活躍警報、當日首次活動通知
- **志工賦能**：將志工手機轉化為移動式監測點
- **社區守護**：建立社區安全網，多點監測長者行蹤
- **智能分流**：根據設備類型自動選擇最佳處理流程

```
### 1.3 目標用戶
```

1. **主要用戶**：社區志工、里辦人員
2. **受益對象**：社區獨居長者、關懷戶
3. **管理人員**：社區管理員、社工人員
4. **相關方**：家屬、照護機構

```
## 二、產品功能需求
```

```
### 2.1 核心功能模組
```

```
#### 2.1.1 藍牙掃描系統
```

```
**功能描述**：
```

- 持續掃描周遭的 BLE Beacon 設備
- 支援 iBeacon 格式解析 (UUID、Major、Minor)
- 自動解析非 iBeacon 格式設備
- 計算訊號強度 (RSSI) 與估算距離

技術規格 :

~~~

- 掃描模式：持續掃描 / 背景掃描
- 支援格式：iBeacon (Apple 標準)
- 訊號過濾：允許重複掃描同一設備
- 距離計算：基於 RSSI 和 TX Power
- 接近程度：immediate (<1m) / near (1-3m) / far (>3m) / unknown

~~~

功能細節 :

1. **前景掃描** : App 開啟時持續掃描
2. **背景掃描** : App 進入背景後定期掃描 (每分鐘)
3. **鎖屏掃描** : 手機鎖屏狀態下仍可掃描
4. **UUID 過濾** : 可設定特定 UUID 進行過濾
5. **Debounce 機制** : 1 分鐘內同一 Beacon 只上傳一次

使用場景 :

- 志工攜帶手機在社區巡邏
- 手機自動掃描長者佩戴的 Beacon
- 偵測到訊號後觸發後續處理流程

2.1.2 GPS 定位系統

功能描述 :

- 高精度 GPS 定位
- 前景與背景定位追蹤
- 位置變化觸發更新

技術規格 :

~~~

- 精度：高精度模式 (bestForNavigation)
- 前景更新：即時獲取位置
- 背景更新：每 30 秒或移動 50 米

- 前景服務 : Android 顯示持續通知

~ ~ ~

\*\*資料欄位\*\* :

- 緯度 (latitude)
- 經度 (longitude)
- 海拔高度 (altitude)
- 精確度 (accuracy, 以米為單位)
- 時間戳記 (timestamp)

\*\*使用場景\*\* :

- 掃描到 Beacon 時立即獲取當前位置
- 背景持續追蹤志工移動軌跡
- 為每筆訊號記錄附加 GPS 座標

---

#### #### 2.1.3 移動接收點 (Mobile Gateway) 系統

\*\*功能描述\*\* :

將志工的智慧型手機註冊為移動式訊號接收點，等同於固定式硬體 Gateway 的功能，但具備移動性。

\*\*註冊流程\*\* :

1. 開啟 App, 點擊「註冊為接收點」
2. 選擇所屬社區（從社區列表選擇）
3. 填寫接收點資訊：
  - 接收點編號（例如：大愛社 001）
  - 位置名稱（例如：志工巡守）
  - 備註（選填）
4. 設定是否為邊界點（勾選框）
5. 系統自動生成序列號：`MOBILE-{型號}-{隨機碼}`

\*\*序列號格式範例\*\* :

~ ~ ~

MOBILE-IPHONE-A3K9F2

MOBILE-GALAXS-B7H4M1

MOBILE-PIXEL-C8K2N5

~ ~ ~

\*\*Gateway 類型\*\* :

- `general` : 一般接收點
- `boundary` : 邊界點（觸發警報）
- `mobile` : 移動接收點（手機）

**\*\*自動記錄資訊\*\* :**

- 設備品牌 (Apple、Samsung、Google 等)
- 設備型號 (iPhone 15、Galaxy S24 等)
- 作業系統版本
- App 版本
- 註冊時間與地點

**\*\*狀態管理\*\* :**

- `active` : 運作中
- `inactive` : 已停用
- 可隨時啟用/停用

**\*\*使用場景\*\* :**

- 志工開始巡邏前註冊為接收點
- 多位志工形成移動監測網
- 特定位置設為邊界點（如社區出入口）

#### #### 2.1.4 智能路由上傳系統

**\*\*功能描述\*\* :**

根據掃描到的設備類型，自動選擇最佳的資料處理流程。

**\*\*設備分類邏輯\*\* :**

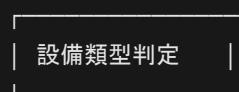
~ ~ ~

掃描到 Beacon (MAC Address)

↓

檢查 Firebase Collections

↓



長者設備

一般設備

未註冊設備

| (elders)                                                          | (devices)   | (無)         |
|-------------------------------------------------------------------|-------------|-------------|
| Cloud Function<br>觸發通知                                            | 直接上傳<br>僅記錄 | 跳過上傳<br>不處理 |
| ~ ~ ~                                                             |             |             |
| **處理流程對比** :                                                      |             |             |
| 設備類型   Collection   上傳方式   觸發通知   創建記錄                            |             |             |
| -----   -----   -----   -----   -----                             |             |             |
| 長者設備   elders   Cloud Function   ✓   logs + locationLogs + alerts |             |             |
| 一般設備   devices   直接上傳   ✗   logs                                  |             |             |
| 未註冊   -   跳過   ✗   ✗                                              |             |             |
| **長者設備處理流程** :                                                    |             |             |
| 1. 檢測到長者設備 (MAC 在 elders collection)                              |             |             |
| 2. 獲取當前 GPS 位置                                                    |             |             |
| 3. 調用 Cloud Function `receiveSignal` API                          |             |             |
| 4. 傳遞參數 :                                                         |             |             |
| - MAC 地址 (標準化格式)                                                  |             |             |
| - RSSI (訊號強度)                                                     |             |             |
| - Gateway 序列號                                                     |             |             |
| - GPS 座標 (緯度、經度、精度)                                               |             |             |
| - 時間戳記                                                            |             |             |
| - Metadata (距離、志工 ID、UUID、Major、Minor)                            |             |             |
| 5. Cloud Function 處理 :                                            |             |             |
| - 創建訊號記錄 (logs)                                                   |             |             |
| - 創建行蹤記錄 (locationLogs)                                           |             |             |
| - 更新長者最後出現時間                                                      |             |             |
| - 檢查通知規則                                                          |             |             |
| - 發送 LINE 通知 (如適用)                                                |             |             |
| - 創建警報記錄 (如適用)                                                    |             |             |
| **一般設備處理流程** :                                                    |             |             |
| 1. 檢測到一般設備 (MAC 在 devices collection)                             |             |             |
| 2. 獲取當前 GPS 位置                                                    |             |             |
| 3. 直接寫入 Firebase `logs` collection                                |             |             |
| 4. 不觸發任何通知                                                        |             |             |
| 5. 完成記錄                                                           |             |             |
| **回退機制** :                                                        |             |             |

- 如果 Cloud Function 調用失敗（網路問題、API 錯誤）
  - 自動回退到直接上傳模式
  - 確保資料不會遺失
  - 記錄錯誤日誌供後續分析
- 

#### #### 2.1.5 通知與警報系統

\*\*整合說明\*\*：

本系統整合現有的 Cloud Function 通知系統，所有通知邏輯統一由後端處理，確保手機 App 與硬體 Gateway 的一致性。

\*\*通知類型\*\*：

##### #### 1. 邊界點警報 🚨

\*\*觸發條件\*\*：

- Gateway 設定為邊界點 (`isBoundary = true`)
- 長者經過該 Gateway

\*\*處理流程\*\*：

1. Cloud Function 檢查 Gateway 是否為邊界點

2. 創建警報記錄：

```
```javascript
{
  alertType: 'inactivity',
  severity: 'medium',
  message: '王宗憲 經過邊界點「里長辦公室」',
  status: 'pending'
}
```
```

3. 發送 LINE 通知給相關人員

4. 前端 LIFF App 即時顯示警報

\*\*使用場景\*\*：

- 社區出入口設為邊界點
- 長者離開社區範圍時觸發
- 長者進入危險區域時觸發

##### #### 2. 當日首次活動通知 🎉

**\*\*觸發條件\*\* :**

- 長者當天第一次被掃描到

**\*\*處理流程\*\* :**

1. Cloud Function 查詢今日是否有先前記錄
2. 發送 LINE 通知：「王宗憲 今日首次活動：在『里長辦公室』」
3. 不創建警報記錄（僅通知，非警報）

**\*\*使用場景\*\* :**

- 家屬確認長者已起床活動
- 志工了解長者活動狀態
- 建立長者作息記錄

**##### 3. 不活躍警報 🕒**

**\*\*觸發條件\*\* :**

- 排程任務每小時檢查
- 長者超過設定時間未活動（預設 12 小時）

**\*\*處理流程\*\* :**

1. 背景排程任務自動執行
2. 創建警報記錄：

```
```javascript
{
  alertType: 'inactivity',
  severity: 'high',
  message: '王宗憲 已超過 12 小時未偵測到活動訊號',
  status: 'pending'
}
```

```
3. 發送 LINE 通知
4. LIFF App 顯示高優先級警報

**\*\*使用場景\*\* :**

- 長者可能發生意外
- 長者外出未歸
- 需要志工或家屬確認狀況

**##### 4. 緊急求救警報 SOS**

**\*\*觸發條件\*\* :**

- Beacon 發送緊急訊號 (`signalType: 'emergency'`)

**\*\*處理流程\*\* :**

1. Cloud Function 識別緊急訊號

2. 創建最高優先級警報 :

```
```javascript
{
  alertType: 'emergency',
  severity: 'critical',
  message: '王宗憲 觸發緊急求救按鈕！',
  status: 'pending'
}
```
```

```

3. 立即發送 LINE 通知給所有相關人員

4. LIFF App 顯示閃爍警報

**\*\*使用場景\*\* :**

- 長者按下 Beacon 緊急按鈕

- 需要立即救援

**\*\*備註\*\* :** 目前手機掃描模式僅支援 `normal` 訊號, `emergency` 訊號需要特殊硬體 Beacon 支援。

##### 5. 電量不足警報 🔋

**\*\*觸發條件\*\* :**

- Beacon 電量低於 20% 或 5%

**\*\*處理流程\*\* :**

1. Cloud Function 檢查電量資訊

2. 創建警報記錄 :

```
```javascript
{
  alertType: 'low_battery',
  severity: 'medium' / 'high',
  message: '王宗憲 的裝置電量不足 (15%)',
  status: 'pending'
}
```
```

```

3. 發送 LINE 通知
4. 提醒更換電池或充電

\*\*備註\*\*：手機掃描無法獲取 Beacon 電池電量，此功能僅適用於硬體 Gateway。

#### #### 2.1.6 權限管理系統

##### \*\*功能描述\*\*：

智慧請求並管理 iOS 和 Android 平台所需的各項權限。

##### \*\*權限清單\*\*：

##### #### Android 權限

```

- ACCESS\_FINE\_LOCATION (精確定位)
  - ACCESS\_COARSE\_LOCATION (粗略定位)
  - ACCESS\_BACKGROUND\_LOCATION (背景定位)
  - BLUETOOTH\_SCAN (藍牙掃描, Android 12+)
  - BLUETOOTH\_CONNECT (藍牙連接, Android 12+)
  - BLUETOOTH\_ADVERTISE (藍牙廣播, Android 12+)
  - FOREGROUND\_SERVICE (前景服務)
  - FOREGROUND\_SERVICE\_LOCATION (定位前景服務)
- ```

##### #### iOS 權限

```

- NSBluetoothAlwaysUsageDescription (背景藍牙)
  - NSLocationAlwaysAndWhenInUseUsageDescription (背景定位)
  - NSLocationWhenInUseUsageDescription (使用時定位)
- ```

##### \*\*權限請求流程\*\*：

##### 1. \*\*啟動檢查\*\*：

- App 啟動時檢查所有權限狀態
- 顯示權限狀態於主畫面

##### 2. \*\*分步驟請求\*\*：

```

第一步：藍牙權限

↓

第二步：前景定位權限

↓

第三步：背景定位權限 (ios 需分兩步)

~~~

3. **Android 12+ 特殊處理**：

- 新的藍牙權限模型
- 需要 BLUETOOTH_SCAN 和 BLUETOOTH_CONNECT
- 自動處理舊版本相容性

4. **iOS 背景定位引導**：

- 先請求「使用時允許」
- 再請求「始終允許」
- 提供友善的說明文字

權限說明文字（可自訂）：

- 藍牙：「允許應用程式在背景掃描藍牙設備以記錄巡守點」
- 定位：「允許應用程式在背景記錄您的位置以完成巡守任務」

錯誤處理：

- 權限被拒絕時顯示友善提示
- 引導用戶前往系統設定手動開啟
- 持續顯示權限狀態提醒

2.1.7 背景任務系統

功能描述：

確保 App 在背景、鎖屏狀態下仍能持續運作。

背景任務類型：

1. 背景定位追蹤

~~~

任務名稱：BACKGROUND\_LOCATION\_TASK

更新頻率：每 30 秒或移動 50 米

執行內容：

- 獲取當前 GPS 位置

- 記錄移動軌跡

- 附加到訊號記錄

~~~

2. 背景 Beacon 掃描

~~~

任務名稱：BACKGROUND\_BEACON\_SCAN\_TASK

執行間隔：每分鐘（iOS 最小 15 分鐘）

執行內容：

- 啟動 BLE 掃描 10 秒

- 偵測 Beacon 訊號

- 處理訊號上傳

- 停止掃描節省電力

~~~

3. Android 前景服務

~~~

顯示：持續通知

內容：「社區守護者正在背景運行」

功能：

- 防止系統強制關閉

- 提醒用戶 App 運作中

- 可點擊返回 App

~~~

電池優化處理：

- 提示用戶關閉電池優化

- 確保背景任務不被系統終止

- 平衡功能與電力消耗

2.1.8 資料儲存系統

Firebase Firestore Collections：

1. elders (長者資料)

```javascript

{

  id: "gozE1W5dDarUgya1rAbY",

  name: "王宗憲",

```
macAddress: "11:11:11:11:11:11",
tenantId: "ZzcoHNKb6qN1vF6Pvf4O",
deviceId: "ItUYUc6IDagvM1RO7Aco",
status: "active",
lastSeen: Timestamp,
lastSignalRssi: -50,
lastGatewayId: "MOBILE-IPHONE-A3K9F2",
inactivityThreshold: 12, // 小時
createdAt: Timestamp,
updatedAt: Timestamp
}
```
```
#####
2. devices (設備資料)

```javascript
{
  id: "ItUYUc6IDagvM1RO7Aco",
  deviceNumber: "大愛社001",
  macAddress: "11:11:11:11:11:11",
  assignedElderId: "gozE1W5dDarUgyalrAbY",
  status: "assigned", // available/assigned/maintenance
  batteryLevel: 85,
  lastMaintenance: Timestamp,
  createdAt: Timestamp
}
```
```
#####
3. gateways (接收點資料)

```javascript
{
 id: "V31K35RW68aMKUB15kAw",
 gatewayNumber: "大愛社001",
 serialNumber: "MOBILE-IPHONE-A3K9F2",
 type: "mobile", // general/boundary/mobile
 isBoundary: false,
 tenantId: "ZzcoHNKb6qN1vF6Pvf4O",
 location: "志工巡守",
 address: "手機移動接收點",
 status: "active",
 coordinates: {
 latitude: 25.028109,
 longitude: 121.525955
 },
}
```

```
DeviceInfo: {
 brand: "Apple",
 model: "iPhone 15",
 osVersion: "17.2",
 appVersion: "1.0.0"
},
lastHeartbeat: Timestamp,
createdAt: Timestamp,
updatedAt: Timestamp
}
```
#### 4. logs (訊號記錄)

```javascript
{
 id: "abc123xyz789",
 tenantId: "ZzcoHNKb6qN1vF6Pvf4O",
 elderId: "gozE1W5dDarUgyalrAbY", // 長者設備才有
 deviceId: "ItUYUc6IDagvM1R07Aco",
 macAddress: "11:11:11:11:11:11",
 rssi: -50,
 gatewayId: "MOBILE-IPHONE-A3K9F2",
 signalType: "normal", // normal/emergency
 timestamp: Timestamp,
 metadata: {
 latitude: 25.028109,
 longitude: 121.525955,
 accuracy: 10.5,
 distance: 2.5,
 volunteerId: "volunteer-001",
 uuid: "E2C56DB5-DFFB-48D2-B060-D0F5A71096E0",
 major: 1,
 minor: 2
 }
}
```
#### 5. locationLogs (行蹤記錄)

```javascript
{
 id: "xyz789abc123",
 tenantId: "ZzcoHNKb6qN1vF6Pvf4O",
 elderId: "gozE1W5dDarUgyalrAbY",
 timestamp: Timestamp
}
```

```

```
elderName: "王宗憲",
gatewayId: "V31K35RW68aMKUB15kAw",
gatewayNumber: "大愛社001",
location: "里長辦公室",
isBoundary: true,
rssi: -50,
timestamp: Timestamp,
latitude: 25.028109,
longitude: 121.525955
}
```
6. alerts (警報記錄)

```javascript
{
  id: "alert123456",
  tenantId: "ZzcoHNKb6qN1vF6Pvf4O",
  elderId: "gozE1W5dDarUgyalrAbY",
  elderName: "王宗憲",
  alertType: "inactivity", // emergency/low_battery/inactivity
  severity: "medium", // critical/high/medium/low
  message: "王宗憲 經過邊界點「里長辦公室」",
  status: "pending", // pending/acknowledged/resolved
  notificationSent: true,
  acknowledgedBy: "admin-user-id",
  acknowledgedAt: Timestamp,
  resolvedAt: Timestamp,
  createdAt: Timestamp
}
```
7. tenants (社區資料)

```javascript
{
  id: "ZzcoHNKb6qN1vF6Pvf4O",
  name: "大愛社區",
  address: "台北市信義區...",
  contactPerson: "李里長",
  contactPhone: "0912345678",
  lineNotifyToken: "YOUR_LINE_TOKEN",
  status: "active",
  createdAt: Timestamp
}
```

```  
\*\*本地儲存 (AsyncStorage) \*\* :

- Gateway 序列號
  - 志工 ID
  - App 設定
  - Debounce 記錄 (快取)
- 

### ### 2.2 用戶介面需求

#### #### 2.2.1 主畫面

\*\*佈局結構\*\* :

```



• 設備002 (x2)
距離: 0.5m RSSI: -42

⌚ 已發現的 Beacon (3)
• 王宗憲的設備
UUID: E2C56...
Major: 1 Minor: 2
RSSI: -50 距離: 2.5m
15秒前

• 設備002
非 iBeacon 設備
RSSI: -60
30秒前

~~~

#### \*\*互動元素\*\* :

1. \*\*請求權限按鈕\*\* : 未授予權限時顯示
2. \*\*開始掃描按鈕\*\* : 權限已授予且未掃描時顯示
3. \*\*停止掃描按鈕\*\* : 掃描中時顯示
4. \*\*註冊為接收點按鈕\*\* : 未註冊時顯示
5. \*\*啟用/停用接收點按鈕\*\* : 已註冊時顯示
6. \*\*近距離區域\*\* : 綠色邊框, 顯示 < 1m 且重複出現的 Beacon
7. \*\*Beacon 列表\*\* : 即時更新, 顯示時間倒數

----

#### #### 2.2.2 接收點註冊表單 (Modal)

##### \*\*表單欄位\*\* :

~~~

⌚ 註冊為接收點

所屬社區 *
[請選擇社區 ▼]
- 大愛社區
- 信義社區
- ...

接收點編號 *
[例如：大愛社001]

位置名稱 *
[例如：志工巡守]
● 手機為移動接收點， 無需填寫固定地址
備註（選填）
[]
<input type="checkbox"/> 設為邊界點
[取消] [註冊]

```

#### \*\*必填欄位\*\* :

- 所屬社區（下拉選單，自動載入 tenants collection）
- 接收點編號
- 位置名稱

#### \*\*選填欄位\*\* :

- 備註
- 設為邊界點（勾選框）

#### \*\*驗證規則\*\* :

- 所有必填欄位不可為空
- 接收點編號建議格式：社區名 + 編號
- 社區列表自動從 Firebase 載入

----

### # # # 2.2.3 Beacon 項目顯示

#### \*\*一般 Beacon\*\* :

```

設備名稱	15秒前
UUID: E2C56DB5...	
Major: 1 Minor: 2	
RSSI: -50 dBm 距離: 2.5m	

近距離重複 Beacon (綠色邊框) :

```
王宗憲的設備 ×3 10秒前
UUID: E2C56DB5...
Major: 1 | Minor: 2
RSSI: -45 dBm 距離: 0.8m
```

非 iBeacon 設備 :

```
設備 ID: AA:BB:CC... 20秒前
⚠ 非 iBeacon 設備
RSSI: -60 dBm
```

視覺標識 :

- 一般 Beacon : 藍色左邊框
- 近距離重複 : 綠色雙邊框 + 綠底
- 非 iBeacon : 橘色警告圖示
- 出現次數 : 綠色圓角標籤 ($\times N$)

2.3 非功能性需求

2.3.1 效能需求

- **掃描延遲** : < 2 秒內偵測到 Beacon
- **定位精度** : GPS 精度 < 20 米 (理想條件下)
- **上傳延遲** : < 5 秒內完成資料上傳
- **UI 更新** : 即時更新, 無明顯延遲
- **背景運行** : 連續運行 8 小時以上

2.3.2 相容性需求

****作業系統版本** :**

- iOS 13.0 以上
- Android 8.0 (API Level 26) 以上

****設備需求** :**

- 支援藍牙 4.0 (BLE)
- 支援 GPS / A-GPS
- 建議 2GB 以上記憶體

****測試設備範圍** :**

- iPhone 12 / 13 / 14 / 15 系列
- Samsung Galaxy S20+ 系列
- Google Pixel 6+ 系列
- 其他主流 Android 設備

2.3.3 安全性需求

- **資料傳輸** : 全程 HTTPS 加密
- **Firebase 安全規則** : 僅授權用戶可讀寫
- **敏感資料** : 不儲存個人識別資訊
- **權限最小化** : 僅請求必要權限

2.3.4 可靠性需求

- **錯誤處理** : 所有 API 調用都有 try-catch
- **回退機制** : Cloud Function 失敗時自動回退
- **離線處理** : 網路斷線時本地快取 (未來功能)
- **日誌記錄** : 詳細的 Console 日誌供除錯

2.3.5 可用性需求

- **學習曲線** : 新用戶 5 分鐘內上手
- **操作步驟** : 核心功能不超過 3 步
- **錯誤提示** : 友善且具指導性
- **多語言** : 優先支援繁體中文

三、技術架構

3.1 技術棧

```
**前端 (Mobile App) ** :  
  
- React Native 0.81.5  
- Expo SDK 54  
- TypeScript 5.9.2  
- React 19.1.0
```

藍牙與定位 :

```
- react-native-ble-px 3.5.0 (BLE 掃描)  
- expo-location 19.0.8 (GPS 定位)  
- expo-task-manager 14.0.9 (背景任務)  
- expo-background-fetch 14.0.9 (定期執行)
```

後端 :

```
- Firebase Firestore (資料庫)  
- Firebase Cloud Functions (通知邏輯)  
- Firebase Authentication (未來功能)
```

本地儲存 :

```
- AsyncStorage 2.2.0 (本地快取)
```

UI 組件 :

```
- React Native Picker 2.10.0 (下拉選單)  
- Expo Notifications 0.32.16 (通知)
```

3.2 專案結構

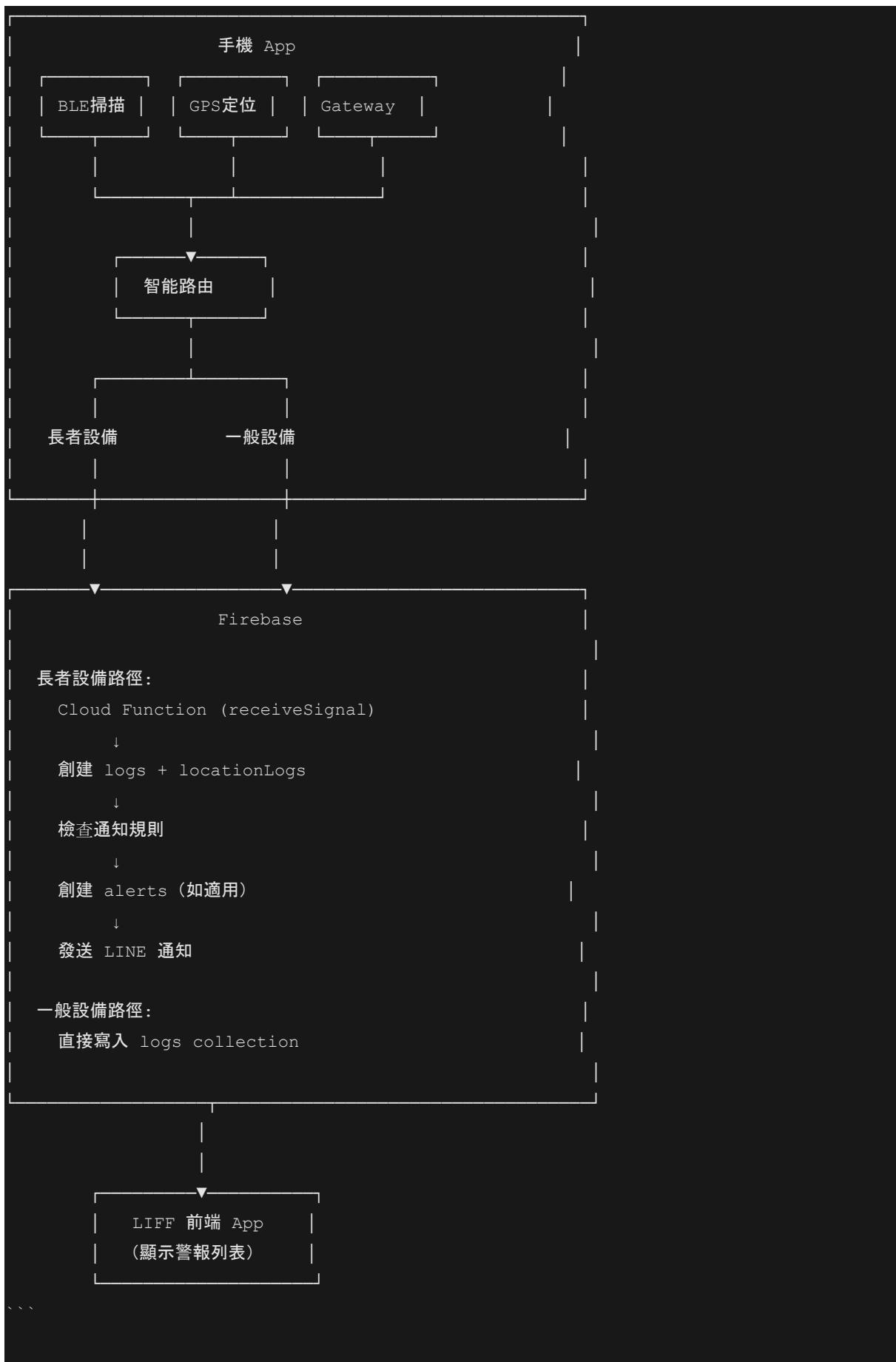
```
```  
safe-net-app/
| └── App.tsx # 主應用程式
| └── app.json # Expo 配置
| └── package.json # 依賴管理
| └── tsconfig.json # TypeScript 配置
|
| └── src/
| | └── config/
| | └── firebase.config.ts # Firebase 配置
| |
| └── services/
```

```
| | └── bleService.ts # BLE 掃描核心邏輯
| | └── locationService.ts # GPS 定位服務
| | └── firebaseService.ts # Firebase 操作
| | └── gatewayService.ts # Gateway 管理
| | └── mobileGatewayService.ts # 手機 Gateway
| | └── tenantService.ts # 社區資料服務
| | └── backgroundTask.ts # 背景任務管理
|
| └── hooks/
| └── useBLE.ts # BLE Hook
| └── useGateway.ts # Gateway Hook
| └── useLocation.ts # 定位 Hook
| └── useDebounce.ts # Debounce Hook
|
| └── types/
| └── beacon.types.ts # Beacon 型別
| └── gateway.types.ts # Gateway 型別
| └── patrol.types.ts # 巡守記錄型別
|
| └── utils/
| └── permissions.ts # 權限管理
| └── deviceId.ts # 設備識別
| └── notificationHelper.ts # 通知輔助
|
└── functions/ # Cloud Functions
 ├── src/
 | ├── index.ts # Functions 入口
 | ├── services/
 | | ├── lineMessagingService.ts # LINE 訊息
 | | └── notificationRules.ts # 通知規則
 | ├── triggers/
 | | ├── onLocationLogCreated.ts # 行蹤記錄觸發器
 | | └── onNotificationCreated.ts # 通知觸發器
 | └── types/
 | └── index.ts # 型別定義
 |
 └── package.json
 └── tsconfig.json
```
---  

```
```

### 3.3 資料流架構

```



```
---  
  
### 3.4 Cloud Function API  
  
**Endpoint**:  
  
```  
POST https://asia-east1-safe-net-test.cloudfunctions.net/receiveSignal
```  
  
**Request Body**:  
  
```typescript  
{
 macAddress: string, // MAC 地址 (標準化格式)
 rssi: number, // 訊號強度 (-100 ~ 0)
 gatewayId: string, // Gateway 序列號
 signalType: 'normal' | 'emergency',
 timestamp: string, // ISO 8601 格式
 metadata: {
 latitude: number, // GPS 緯度
 longitude: number, // GPS 經度
 accuracy: number, // GPS 精度 (米)
 distance?: number, // 距離 (米)
 volunteerId?: string, // 志工 ID
 uuid?: string, // iBeacon UUID
 major?: number, // iBeacon Major
 minor?: number // iBeacon Minor
 }
}
```  
  
**Response**:  
  
```typescript  
{
 success: boolean,
 data?: {
 logId: string, // 記錄 ID
 alertTriggered: boolean // 是否觸發警報
 },
 error?: string
}
```
```

四、使用流程

4.1 初次使用流程

```

1. 下載並安裝 App

↓

2. 開啟 App

↓

3. 點擊「請求權限」

↓

4. 授予藍牙、定位權限

↓

5. 點擊「註冊為接收點」

↓

6. 填寫註冊表單：

- 選擇所屬社區
- 填寫接收點編號
- 填寫位置名稱
- (選擇性) 勾選「設為邊界點」

↓

7. 點擊「註冊」

↓

8. 系統生成序列號並保存

↓

9. 點擊「開始掃描」

↓

10. 開始巡守任務

```

4.2 日常使用流程

```

1. 開啟 App

↓

2. 確認接收點狀態為「運作中」

↓

3. 點擊「開始掃描」

↓

4. 將手機放入口袋或包包

↓

5. 在社區內巡邏  
↓  
6. App 自動掃描並上傳資料  
↓  
7. (長者設備) 自動觸發通知  
↓  
8. (邊界點) 觸發邊界點警報  
↓  
9. 巡邏結束後點擊「停止掃描」  
` ` `

---

### ### 4.3 警報處理流程

` ` `

1. 系統偵測到長者設備  
↓  
2. Cloud Function 處理  
↓  
3. 檢查是否觸發警報條件  
↓  
4. (如觸發) 創建警報記錄  
↓  
5. 發送 LINE 通知給相關人員  
↓  
6. LIFF App 顯示警報  
↓  
7. 志工/管理員確認警報  
↓  
8. 前往關懷或處理  
↓  
9. 在 LIFF App 中標記「已處理」  
` ` `

---

## ## 五、測試計劃

### ### 5.1 功能測試

#### #### 測試案例 1：權限請求

- \*\*測試目標\*\*：驗證所有權限能正確請求
- \*\*前置條件\*\*：全新安裝 App

- \*\*測試步驟\*\*：
  1. 開啟 App
  2. 點擊「請求權限」
  3. 依序授予權限
- \*\*預期結果\*\*：所有權限顯示「 已授予」

#### ##### 測試案例 2 : Gateway 註冊

- \*\*測試目標\*\*：驗證手機能註冊為接收點
- \*\*測試步驟\*\*：
  1. 點擊「註冊為接收點」
  2. 選擇社區
  3. 填寫必填欄位
  4. 點擊「註冊」
- \*\*預期結果\*\*：
  - 顯示「註冊成功」
  - 接收點狀態變為「已註冊」
  - 顯示序列號

#### ##### 測試案例 3 : Beacon 掃描

- \*\*測試目標\*\*：驗證能掃描到 Beacon
- \*\*前置條件\*\*：已授予权限
- \*\*測試步驟\*\*：
  1. 點擊「開始掃描」
  2. 將 Beacon 靠近手機
- \*\*預期結果\*\*：
  - Beacon 出現在列表
  - 顯示 RSSI、距離等資訊
  - iBeacon 顯示 UUID、Major、Minor

#### ##### 測試案例 4 : 長者設備上傳

- \*\*測試目標\*\*：驗證長者設備能正確上傳
- \*\*前置條件\*\*：
  - Beacon MAC 已在 elders collection 註冊
  - 手機已註冊為 Gateway
- \*\*測試步驟\*\*：
  1. 開始掃描
  2. 掃描到長者 Beacon
- \*\*預期結果\*\*：
  - Console 顯示「找到長者設備」
  - Console 顯示「Cloud Function 調用成功」
  - Firebase logs 有新記錄
  - Firebase locationLogs 有新記錄

#### #### 測試案例 5：邊界點警報

- \*\*測試目標\*\*：驗證邊界點警報能觸發
- \*\*前置條件\*\*：
  - Gateway 設為邊界點
  - 長者設備已註冊
- \*\*測試步驟\*\*：
  1. 開始掃描
  2. 掃描到長者 Beacon
- \*\*預期結果\*\*：
  - Firebase alerts 有新警報
  - 警報類型為 'inactivity'
  - 訊息包含「經過邊界點」
  - 收到 LINE 通知

#### #### 測試案例 6：當日首次活動

- \*\*測試目標\*\*：驗證首次活動通知
- \*\*前置條件\*\*：
  - 長者今日尚無記錄
- \*\*測試步驟\*\*：
  1. 開始掃描
  2. 掃描到長者 Beacon
- \*\*預期結果\*\*：
  - 收到 LINE 通知「今日首次活動」
  - 不創建警報記錄

#### #### 測試案例 7：一般設備上傳

- \*\*測試目標\*\*：驗證一般設備直接上傳
- \*\*前置條件\*\*：
  - Beacon MAC 在 devices collection
  - 不在 elders collection
- \*\*測試步驟\*\*：
  1. 開始掃描
  2. 掃描到該 Beacon
- \*\*預期結果\*\*：
  - Console 顯示「找到一般設備」
  - Console 顯示「直接寫入 Firebase」
  - Firebase logs 有新記錄
  - 不調用 Cloud Function
  - 不觸發通知

#### #### 測試案例 8：Debounce 機制

- \*\*測試目標\*\* : 驗證 1 分鐘內不重複上傳

- \*\*測試步驟\*\* :

1. 開始掃描

2. 持續掃描同一 Beacon

- \*\*預期結果\*\* :

- 第一次掃描 : 上傳成功

- 1 分鐘內再次掃描 : 跳過上傳

- 1 分鐘後掃描 : 再次上傳

#### ##### 測試案例 9：背景運行

- \*\*測試目標\*\* : 驗證背景持續掃描

- \*\*測試步驟\*\* :

1. 開始掃描

2. 按 Home 鍵回到桌面

3. 鎖定螢幕

4. 將 Beacon 靠近手機

- \*\*預期結果\*\* :

- 背景持續掃描

- 資料正常上傳

- Android 顯示前景通知

#### ##### 測試案例 10：回退機制

- \*\*測試目標\*\* : 驗證 Cloud Function 失敗時回退

- \*\*前置條件\*\* :

- 暫時設置錯誤的 Cloud Function URL

- \*\*測試步驟\*\* :

1. 開始掃描

2. 掃描到長者 Beacon

- \*\*預期結果\*\* :

- Console 顯示「Cloud Function 處理失敗」

- Console 顯示「回退到直接上傳模式」

- Firebase logs 有新記錄

- 資料未遺失

----

#### ## 5.2 效能測試

- \*\*持續運行測試\*\* : 連續運行 8 小時，檢查記憶體洩漏

- \*\*電池消耗測試\*\* : 測量背景運行時的電池消耗率

- \*\*大量 Beacon 測試\*\* : 同時偵測 10+ 個 Beacon

- \*\*網路延遲測試\*\* : 弱網路環境下的上傳表現

```

5.3 相容性測試

- **iOS 測試** : iOS 13 ~ 17
- **Android 測試** : Android 8 ~ 14
- **不同品牌** : Apple、Samsung、Google、小米等
- **不同螢幕** : 小螢幕 (< 5 吋) 到大螢幕 (> 6.5 吋)

```

## ## 六、部署與配置

### ### 6.1 環境需求

\*\*開發環境\*\* :

- Node.js 18+
- npm 或 yarn
- Expo CLI
- Android Studio (Android 開發)
- Xcode (iOS 開發)

\*\*Firebase 專案\*\* :

- Firestore Database
- Cloud Functions
- (選用) Authentication

### ### 6.2 配置步驟

#### #### 步驟 1 : Clone 專案

```
```bash  
git clone https://github.com/your-repo/safe-net-app.git  
cd safe-net-app  
```
```

#### #### 步驟 2 : 安裝依賴

```
```bash  
npm install
```

```
```
步驟 3：配置 Firebase

編輯 `src/config.firebaseio.config.ts`：

```typescript
const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_AUTH_DOMAIN",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_STORAGE_BUCKET",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID",
};

```

步驟 4：配置 Cloud Function URL

編輯 `src/services/cloudFunctionService.ts` 或創建 `*.env`：

```bash
EXPO_PUBLIC_CLOUD_FUNCTION_URL=https://asia-east1-your-project.cloudfunctions.net/receiveSignal
```

步驟 5：部署 Cloud Functions

```bash
cd functions
npm install
firebase deploy --only functions
```

步驟 6：設置 Firestore 安全規則

編輯 `firestore.rules` 並部署：

```bash
firebase deploy --only firestore:rules
```

步驟 7：Prebuild

```bash
```

```
npx expo prebuild --clean
```

步驟 8：運行 App

```bash
# Android
npm run android

# iOS
npm run ios
```

6.3 Firebase Collections 初始化

創建 tenants

```javascript
{
  name: "測試社區",
  address: "台北市信義區...",
  contactPerson: "張先生",
  contactPhone: "0912345678",
  lineNotifyToken: "YOUR_LINE_TOKEN",
  status: "active",
  createdAt: new Date()
}
```

創建 elders

```javascript
{
  name: "測試長者",
  macAddress: "11:11:11:11:11:11",
  tenantId: "YOUR_TENANT_ID",
  status: "active",
  inactivityThreshold: 12,
  createdAt: new Date()
}
```

創建 devices
```

```
```javascript
{
  deviceNumber: "TEST001",
  macAddress: "11:11:11:11:11:11",
  assignedElderId: "YOUR_ELDER_ID",
  status: "assigned",
  createdAt: new Date()
}
```

```

## ## 七、維護與支援

### ### 7.1 日誌監控

#### \*\*Console 日誌層級\*\* :

- ✓ 成功 : 一般操作成功
- 💡 資訊 : BLE 掃描、訊號發現
- ● 位置 : GPS 定位相關
- 🔍 檢查 : 設備類型判定
- 🔔 通知 : Cloud Function 調用
- ⚠️ 警告 : 回退機制、非致命錯誤
- ✖️ 錯誤 : 失敗操作、異常

#### \*\*Firebase 監控\*\* :

- Cloud Functions 執行日誌
- Firestore 使用量
- 錯誤率統計

### ### 7.2 常見問題處理

#### #### Q1 : 掃描不到 Beacon

#### \*\*解決方案\*\* :

1. 檢查藍牙是否開啟
2. 確認藍牙權限已授予
3. 檢查 Beacon 是否有電
4. 確認 Beacon 是否在廣播

#### Q2：無法上傳資料

\*\*解決方案\*\*：

1. 檢查網路連線
2. 確認 Firebase 配置正確
3. 檢查 Firestore 安全規則
4. 查看 Console 錯誤日誌

#### Q3：背景掃描停止

\*\*解決方案\*\*：

1. 檢查背景定位權限 (iOS：始終允許)
2. Android：關閉電池優化
3. 確認前景服務通知顯示
4. 重啟 App

#### Q4：沒有收到 LINE 通知

\*\*解決方案\*\*：

1. 檢查 Cloud Function 是否正常運行
2. 確認 LINE Notify Token 已配置
3. 檢查 Gateway 序列號是否正確
4. 查看 Cloud Function 執行日誌

#### Q5：Gateway 註冊失敗

\*\*解決方案\*\*：

1. 確認所有必填欄位已填寫
2. 檢查社區列表是否載入
3. 確認網路連線正常
4. 查看 Console 錯誤訊息

----

## 7.3 更新計劃

\*\*v1.1 (計劃中) \*\*：

- 離線模式支援
- 巡守記錄查詢功能

- 地圖顯示功能
- 統計報表

\*\*\*v1.2 (計劃中) \*\*\* :

- Firebase Authentication 整合
- 用戶登入/註冊
- 多志工協作
- 訊息推送

\*\*\*v2.0 (長期) \*\*\* :

- 緊急按鈕手動觸發
- 電池電量監控 (需硬體支援)
- AI 異常偵測
- 預測性警報

## ## 八、成功指標

### ### 8.1 關鍵績效指標 (KPI)

#### - \*\*技術指標\*\* :

- Beacon 偵測成功率 > 95%
- 資料上傳成功率 > 98%
- 背景運行穩定性 > 99%
- App 崩潰率 < 0.5%

#### - \*\*業務指標\*\* :

- 志工採用率 > 80%
- 長者覆蓋率 > 90%
- 警報回應時間 < 10 分鐘
- 誤報率 < 5%

#### - \*\*用戶滿意度\*\* :

- 易用性評分 > 4.5/5
- 志工回饋滿意度 > 85%
- App Store 評分 > 4.0

## ## 九、風險與對策

### ### 9.1 技術風險

| 風險                | 影響 | 機率 | 對策              |
|-------------------|----|----|-----------------|
| 背景任務被系統終止         | 高  | 中  | 前景服務 + 用戶教育     |
| Beacon 訊號不穩定      | 中  | 高  | Debounce + 距離過濾 |
| GPS 定位不精確         | 中  | 中  | 高精度模式 + 精度檢查    |
| Cloud Function 超時 | 中  | 低  | 回退機制 + 錯誤重試     |
| 電池消耗過快            | 中  | 中  | 優化掃描間隔 + 用戶提醒   |

### ### 9.2 業務風險

| 風險            | 影響 | 機率 | 對策         |
|---------------|----|----|------------|
| 志工不願使用        | 高  | 中  | 培訓 + 誘因機制  |
| 長者拒絕佩戴 Beacon | 高  | 低  | 溝通 + 設計改善  |
| 隱私疑慮          | 中  | 低  | 透明化 + 同意書  |
| 維護成本高         | 中  | 中  | 自動化 + 社區分擔 |

## ## 十、附錄

### ### 10.1 術語表

- **BLE (Bluetooth Low Energy)**：藍牙低功耗技術
- **iBeacon**：Apple 定義的 Beacon 標準格式
- **RSSI (Received Signal Strength Indicator)**：接收訊號強度指標
- **Gateway**：訊號接收點
- **Debounce**：防抖動機制，避免短時間內重複處理
- **UUID (Universally Unique Identifier)**：通用唯一識別碼
- **Major / Minor**：iBeacon 的識別參數
- **Foreground Service**：Android 前景服務，確保背景運行

### ### 10.2 相關文件

- `README.md` - 專案總覽
- `PROJECT\_SUMMARY.md` - 專案完成總結
- `INTEGRATION\_SUMMARY.md` - Cloud Function 整合摘要
- `CLOUD\_FUNCTION\_INTEGRATION.md` - Cloud Function 整合詳解
- `GATEWAY\_FEATURE.md` - Gateway 功能說明
- `INSTALLATION.md` - 安裝指南
- `SETUP\_GUIDE.md` - 設置指南
- `API\_REFERENCE.md` - API 參考文件

### ### 10.3 聯絡資訊

- \*\*技術支援\*\* : daniel@example.com
- \*\*專案 GitHub\*\* : <https://github.com/your-repo/safe-net-app>
- \*\*問題回報\*\* : <https://github.com/your-repo/safe-net-app/issues>

----

### ## 版本歷程

| 版本  | 日期         | 變更內容      |
|-----|------------|-----------|
| 1.0 | 2026-01-15 | 初版 PRD 發布 |

----

\*\*文件結尾\*\*