

Community Guardian SaaS - 完整產品需求文件 (PRD)

版本: 2.0

最後更新: 2026-01-15

專案狀態:  已完成開發，可部署至生產環境

📁 目錄

1. [產品概述](#1-產品概述)
2. [核心價值主張](#2-核心價值主張)
3. [系統架構](#3-系統架構)
4. [用戶角色與權限](#4-用戶角色與權限)
5. [核心功能模組](#5-核心功能模組)
6. [資料庫設計](#6-資料庫設計)
7. [API 規格](#7-api-規格)
8. [硬體整合](#8-硬體整合)
9. [LINE 整合](#9-line-整合)
10. [安全性與權限控制](#10-安全性與權限控制)
11. [部署與維運](#11-部署與維運)
12. [技術堆疊](#12-技術堆疊)
13. [未來擴展規劃](#13-未來擴展規劃)

1. 產品概述

1.1 產品定位

Community Guardian 是一個專為社區/里長設計的 **B2B2C 物聯網安全監控平台**，透過 BLE Beacon 與 Gateway 的整合，實現 24/7 無感式監控獨居長者的活動狀態，並在偵測到異常時立即透過 LINE 通知管理人員。

1.2 問題陳述

目標問題:

- 獨居長者安全難以即時掌握
- 傳統巡視耗費人力且無法全天候監控
- 長者不擅使用智慧型手機 APP
- 社區管理員需要簡單易用的監控工具

解決方案:

- 長者配戴輕薄卡片式 Beacon(如信用卡大小)
- Gateway 自動接收訊號並上傳雲端
- 里長透過 LINE 即可查看所有長者狀態
- 異常時自動發送 LINE 推播通知

1.3 商業模式

平台方 (Super Admin)

↓ 提供 SaaS 服務 + 硬體支援

社區/里長 (Tenant Admin)

↓ 月費/年費訂閱

獨居長者 (End User)

收費模式 :

- 基礎版：每月 NT\$2,000 (最多 20 位長者)
- 專業版：每月 NT\$5,000 (最多 50 位長者)
- 企業版：客製化報價 (無上限 + 專屬支援)

2. 核心價值主張

2.1 對社區/里長

- ✓ **零下載門檻** - 透過 LINE 官方帳號，無需下載額外 APP
- ✓ **即時警報** - 緊急求救、長時間未活動、邊界點通過立即通知
- ✓ **行蹤記錄** - 自動記錄長者經過哪些接收點
- ✓ **輕鬆管理** - 簡潔的 LIFF 介面，隨時查看所有長者狀態
- ✓ **首日訊號通知** - 每日第一筆訊號自動通知，確認長者平安

2.2 對長者

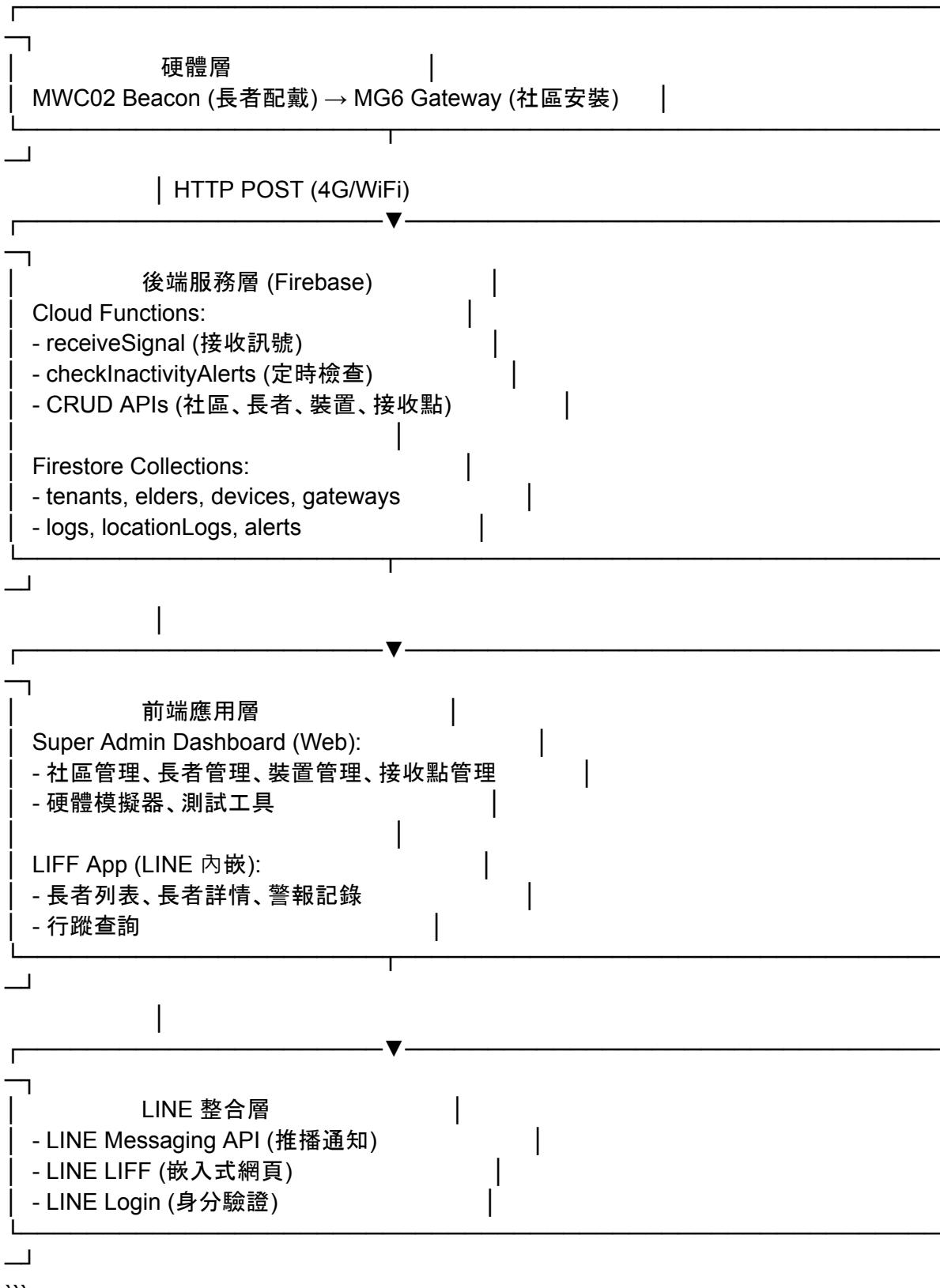
- ✓ **無感監控** - 無需操作任何裝置，只需隨身攜帶卡片
- ✓ **緊急求救** - 卡片上的按鈕可一鍵求救
- ✓ **保護隱私** - 僅記錄活動時間，不追蹤具體位置
- ✓ **輕薄舒適** - 信用卡大小，可放錢包或掛繩配戴

2.3 對平台方

- ✓ **多租戶架構** - 一套系統服務多個社區，資料完全隔離
- ✓ **易於擴展** - 新增社區僅需 5 分鐘設定
- ✓ **自動化運作** - Cloud Functions 自動處理所有訊號
- ✓ **硬體模擬器** - 內建測試工具，方便開發與展示

3. 系統架構

3.1 整體架構圖



3.2 資料流程

正常訊號接收流程:

...

1. Beacon 發送 BLE 訊號
 2. Gateway 接收並透過 4G/WiFi 上傳到 Cloud Function
 3. 查詢長者資料 (by MAC Address)
 4. 更新長者最後出現時間 (lastSeen)
 5. 記錄訊號 (logs collection)
 6. 記錄行蹤 (locationLogs collection)
 7. 更新裝置電量 (devices collection)
 8. 檢查是否為當日首筆訊號 → 發送通知
 9. 檢查是否為邊界點 → 發送警報
 10. 檢查電池電量 → 低於 20% 發送警報
- ...

4. 用戶角色與權限

4.1 Super Admin (平台管理員)

身分驗證: Firebase Authentication (Email/Password)

權限:

- 管理所有 Tenant (社區)
- 管理所有 Elder (長者)
- 管理所有 Device (裝置)
- 管理所有 Gateway (接收點)
- 查看所有系統日誌
- 使用硬體模擬器
- 使用測試工具

主要頁面:

- Admin Dashboard - 統計總覽
- Tenant Management - 社區管理
- Elder Management - 長者管理
- Device Management - 裝置管理
- Gateway Management - 接收點管理
- Testing Tools - 測試工具 (硬體模擬器、LINE 推播測試)

4.2 Tenant Admin (里長/社區管理員)

身分驗證: LINE User ID (透過 LIFF)

權限:

- 查看自己社區的長者列表
- 查看長者詳細資訊和行蹤
- 查看自己社區的警報記錄
- 確認和解決警報
- 無法查看其他社區資料
- 無法修改系統設定

****主要頁面 (LIFF): ****

- Elder List - 長者列表(含狀態燈號)
- Elder Detail - 長者詳情(最近 24 小時活動、行蹤記錄)
- Alert List - 警報列表

4.3 權限控制機制

****授權流程 :****

```

1. Super Admin 在 Tenant 設定中新增 LINE User ID 到 adminLineIds
2. 里長加入 LINE 官方帳號並點擊 LIFF 連結
3. LIFF 自動取得 LINE User ID
4. 查詢 Firestore: WHERE adminLineIds array-contains lineUserId
5. 找到對應的 Tenant → 授權成功
6. 未找到 → 顯示「權限不足」

```

5. 核心功能模組

5.1 社區管理 (Tenant Management)

****功能清單 :****

- 新增社區(名稱、地址、聯絡人)
- 設定 LINE 整合(Channel Access Token、LIFF ID)
- 管理授權管理員(adminLineIds)
- 設定警報參數(超過 N 小時未活動)
- 訂閱狀態管理(active / suspended / expired)
- 編輯 / 刪除社區

****資料結構 :****

```typescript

```
interface Tenant {
 id: string;
 name: string;
 address?: string;
 contactPerson?: string;
 contactPhone?: string;
 lineConfig: {
 channelAccessToken: string;
 channelSecret: string;
 lifId: string;
 };
 adminLineIds: string[];
 subscription: {
 plan: 'basic' | 'pro' | 'enterprise';
 };
}
```

```

 startDate: string;
 endDate: string;
 status: 'active' | 'suspended' | 'expired';
 };
 settings: {
 alertThresholdHours: number;
 enableEmergencyAlert: boolean;
 enableInactivityAlert: boolean;
 };
}
...

```

### ### 5.2 長者管理 (Elder Management)

**\*\*功能清單:\*\***

- 新增長者(姓名、年齡、地址、緊急聯絡人)
- 配發裝置(從裝置清單選擇)
- 查看長者狀態(active / offline)
- 查看最後出現時間
- 編輯長者資料
- 更換裝置
- 刪除長者(軟刪除, 保留歷史記錄)

**\*\*狀態燈號:\*\***

-  綠色: 1 小時內有訊號
-  黃色: 1-6 小時無訊號
-  橙色: 6-12 小時無訊號
-  紅色: 超過 12 小時無訊號

### ### 5.3 裝置管理 (Device Management)

**\*\*功能清單:\*\***

- 新增裝置(輸入 MAC Address)
- 自動生成裝置編號(社區名稱前3字 + 流水號)
- 裝置狀態管理(available / assigned / maintenance / retired)
- 查看配發對象
- 查看電池電量(自動更新)
- 編輯 / 刪除裝置(已配發的無法刪除)

**\*\*裝置與長者綁定機制:\*\***

...

**重要:**長者必須從裝置清單選擇, 不可手動輸入 MAC Address

**優點:**

1. 數據一致性 - MAC Address 統一由裝置管理維護
2. 狀態追蹤 - 清楚知道每個裝置的配發狀態
3. 電量監控 - 裝置電量資訊自動同步
4. 防止錯誤 - 避免手動輸入錯誤

...

### ### 5.4 接收點管理 (Gateway Management)

\*\*功能清單:\*\*

- 新增接收點(序號、地點名稱、地址)
- 設定邊界點(出入口、社區邊界)
- 自動生成接收點編號(社區名稱前3字 + 流水號)
- 查看最後訊號時間
- 查看接收點狀態(online / offline)
- 編輯 / 刪除接收點

\*\*邊界點警報:\*\*

...

當長者經過設為「邊界點」的接收點時：

1. 立即創建警報 (Alert)
2. 發送 LINE 推播通知
3. 訊息內容：「XXX 經過邊界點『社區大門』」

...

### ### 5.5 信號接收與處理 (Signal Processing)

\*\*API Endpoint:\*\* `POST /receiveSignal`

\*\*請求格式:\*\*

```
```json
{
  "macAddress": "AA:BB:CC:DD:EE:FF",
  "rssi": -70,
  "gatewayId": "MG6-001",
  "signalType": "normal",
  "timestamp": "2026-01-15T10:30:00.000Z",
  "metadata": {
    "batteryLevel": 85
  }
}
````
```

\*\*處理邏輯:\*\*

...

1. 驗證請求格式
2. 查詢長者 (WHERE macAddress == ?)
3. 更新長者 lastSeen
4. 記錄訊號 (logs)
5. 記錄行蹤 (locationLogs)
6. 更新裝置電量
7. 檢查是否為當日首筆訊號 → 發送通知
8. 檢查邊界點 → 創建警報

9. 檢查緊急按鈕 → 創建警報
10. 檢查電池電量 → 創建警報
- ...

### ### 5.6 警報系統 (Alert System)

**\*\*警報類型 :\*\***

| 類型   觸發條件   嚴重程度   通知方式                           |
|---------------------------------------------------|
| ----- ----- ----- -----                           |
| emergency   緊急按鈕被按下   Critical   LINE 推播(立即)      |
| inactivity   超過 N 小時未偵測   High   LINE 推播 + 定時任務檢查 |
| low_battery   電量 < 20%   Medium   LINE 推播         |
| low_battery   電量 < 5%   High   LINE 推播            |
| boundary   經過邊界點   Medium   LINE 推播               |

**\*\*定時任務 :\*\***

```
```typescript
// 每小時執行一次
export const checkInactivityAlerts = functions.pubsub
  .schedule('every 1 hours')
  .onRun(async (context) => {
    // 檢查所有 active tenants
    // 比對每位長者的 lastSeen
    // 超過 alertThresholdHours → 創建警報
  });
```
```

```

5.7 首日訊號通知

****功能說明 :****

每位長者每天的第一筆訊號會自動發送 LINE 通知，讓里長確認長者平安起床活動。

****實現邏輯 :****

```
```typescript
// 檢查今天是否已有訊號記錄
const todayStart = new Date().setHours(0, 0, 0, 0);
const todaySignals = await db.collection('logs')
 .where('elderId', '==', elderId)
 .where('timestamp', '>=', todayStart)
 .get();

if (todaySignals.empty) {
 // 發送首日訊號通知
 await sendFirstSignalNotification(tenantId, elderId, location);
}
```
```

```

\*\*通知內容:\*\*

...

長者活動通知

姓名:王大明

今日首次訊號

時間:2026-01-15 07:30

地點:社區活動中心

長者已開始今日活動，請放心。

...

---

## ## 6. 資料庫設計

### ### 6.1 Collection: tenants (社區)

```
```typescript
{
  id: "tenant001",
  name: "松柏社區",
  address: "台北市大安區復興南路一段 123 號",
  contactPerson: "王里長",
  contactPhone: "0912-345-678",
  lineConfig: {
    channelAccessToken: "xxx",
    channelSecret: "xxx",
    liffId: "1234567890-xxx"
  },
  adminLineIds: ["U1234567890abcdef"],
  subscription: {
    plan: "pro",
    startDate: "2026-01-01T00:00:00.000Z",
    endDate: "2026-12-31T23:59:59.000Z",
    status: "active"
  },
  settings: {
    alertThresholdHours: 12,
    enableEmergencyAlert: true,
    enableInactivityAlert: true
  },
  createdAt: "2026-01-01T00:00:00.000Z",
  updatedAt: "2026-01-15T00:00:00.000Z"
}
````
```

### ### 6.2 Collection: elders (長者)

```
```typescript
{
  id: "elder001",
  tenantId: "tenant001",
  name: "王大明",
  age: 78,
  gender: "male",
  address: "台北市大安區建國南路二段 50 號 3 樓",
  contactPhone: "0912-111-222",
  emergencyContact: "王小華 (兒子)",
  emergencyPhone: "0912-333-444",
  macAddress: "AA:BB:CC:DD:EE:FF",
  deviceId: "device001",
  status: "active",
  lastSeen: "2026-01-15T10:05:00.000Z",
  lastSignalRssi: -65,
  lastGatewayId: "GW-001",
  notes: "有高血壓病史，每日需服藥",
  createdAt: "2026-01-10T09:00:00.000Z",
  updatedAt: "2026-01-15T10:05:00.000Z"
}
```
```

```

6.3 Collection: devices (裝置)

```
```typescript
{
 id: "device001",
 tenantId: "tenant001",
 macAddress: "AA:BB:CC:DD:EE:FF",
 deviceNumber: "松柏社001",
 status: "assigned",
 assignedElderId: "elder001",
 assignedElderName: "王大明",
 lastBatteryLevel: 85,
 lastBatteryUpdate: "2026-01-15T10:05:00.000Z",
 notes: "2024年1月購入",
 createdAt: "2026-01-01T00:00:00.000Z",
 updatedAt: "2026-01-15T10:05:00.000Z"
}
```
```

```

### ### 6.4 Collection: gateways (接收點)

```
```typescript
{
  id: "gateway001",
```
```

```

```
tenantId: "tenant001",
serialNumber: "MG6-001",
gatewayNumber: "松柏社G001",
location: "社區大門",
address: "台北市大安區復興南路一段 123 號 1F",
isBoundary: true,
status: "online",
lastSeen: "2026-01-15T10:05:00.000Z",
notes: "主要出入口",
createdAt: "2026-01-01T00:00:00.000Z",
updatedAt: "2026-01-15T10:05:00.000Z"
}
```
``
```

### ### 6.5 Collection: logs (訊號記錄)

```
```typescript
{
  id: "log001",
  tenantId: "tenant001",
  elderId: "elder001",
  macAddress: "AA:BB:CC:DD:EE:FF",
  rssi: -70,
  gatewayId: "MG6-001",
  signalType: "normal",
  timestamp: "2026-01-15T10:05:00.000Z",
  metadata: {
    batteryLevel: 85
  },
  createdAt: "2026-01-15T10:05:02.000Z"
}
```
``
```

### ### 6.6 Collection: locationLogs (行蹤記錄)

```
```typescript
{
  id: "location001",
  tenantId: "tenant001",
  elderId: "elder001",
  elderName: "王大明",
  macAddress: "AA:BB:CC:DD:EE:FF",
  gatewayId: "gateway001",
  gatewayNumber: "松柏社G001",
  location: "社區大門",
  isBoundary: true,
  rssi: -70,
  timestamp: "2026-01-15T10:05:00.000Z",
}
```

```
    createdAt: "2026-01-15T10:05:02.000Z"
}
...

```

6.7 Collection: alerts (警報)

```
```typescript
{
 id: "alert001",
 tenantId: "tenant001",
 elderId: "elder001",
 elderName: "王大明",
 alertType: "inactivity",
 severity: "high",
 message: "王大明 已超過 12 小時未偵測到活動訊號",
 status: "acknowledged",
 acknowledgedBy: "U1234567890abcdef",
 acknowledgedAt: "2026-01-15T22:10:00.000Z",
 notificationSent: true,
 notificationSentAt: "2026-01-15T22:05:30.000Z",
 createdAt: "2026-01-15T22:05:00.000Z"
}
...
```
---
```

7. API 規格

7.1 接收訊號 API

Endpoint: `POST /receiveSignal`

已在第 5.5 節詳細說明。

7.2 Tenant CRUD APIs

- `POST /tenants/create`
- `PUT /tenants/:id/update`
- `GET /tenants/:id`
- `GET /tenants/list`
- `DELETE /tenants/:id` (軟刪除)

7.3 Elder CRUD APIs

- `POST /elders/create`
- `PUT /elders/:id/update`
- `GET /elders/:id`
- `GET /elders/list?tenantId=xxx`

- `DELETE /elders/:id` (軟刪除)

7.4 Alert APIs

- `GET /alerts/list?tenantId=xxx`
- `PUT /alerts/:id/acknowledge`
- `PUT /alerts/:id/resolve`

8. 硬體整合

8.1 硬體規格

MWC02 Ultra-thin Location Card (Beacon):

- 尺寸: 85.6mm × 54mm × 0.8mm (信用卡大小)
- 藍牙: BLE 4.0/5.0
- 訊號範圍: 10-50 公尺
- 電池: CR2032, 繽航 3-12 個月

MG6 4G Bluetooth® Stellar Gateway:

- 4G LTE 連接
- 藍牙 4.0/5.0
- 掃描範圍: 10-100 公尺
- 供電: DC 5V 或 PoE

8.2 整合步驟

1. **註冊裝置:** 在裝置管理中新增 Beacon, 輸入 MAC Address
2. **註冊接收點:** 在接收點管理中新增 Gateway, 輸入序號
3. **設定 Gateway:** 將 Cloud Function API 地址配置到 Gateway
4. **測試連接:** 使用硬體模擬器測試流程

詳細整合指南請參考 `HARDWARE_INTEGRATION_GUIDE.md`。

9. LINE 整合

9.1 LINE Messaging API

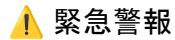
用途: 發送推播通知給里長

訊息類型:

- Text Message (警報內容)
- Flex Message (卡片式訊息)
- Template Message (包含按鈕連結)

範例訊息:

...



姓名:王大明

類型:緊急求救

時間:2026-01-15 14:30

地點:社區活動中心

請立即確認長者狀況！

[查看詳細資訊] (按鈕連結到 LIFF)

...

9.2 LINE LIFF

用途: 提供里長查看長者資訊的介面

授權流程:

...

1. 里長加入 LINE 官方帳號

2. 點擊 LIFF 連結

3. 自動取得 LINE User ID

4. 查詢 Tenant (WHERE adminLineIds contains lineUserId)

5. 授權成功 → 顯示長者列表

6. 授權失敗 → 顯示「權限不足」

...

10. 安全性與權限控制

10.1 Firestore Security Rules

```
```javascript
rules_version = '2';
service cloud.firestore {
 match /databases/{database}/documents {
 // Tenants: 僅 Super Admin 可寫入
 match /tenants/{tenantId} {
 allow read: if request.auth != null;
 allow write: if request.auth != null &&
 request.auth.token.role == 'admin';
 }

 // Elders: 資料隔離
 match /elders/{elderId} {
 allow read: if request.auth != null;
 }
 }
}
```

```
 allow write: if request.auth != null;
}

// Logs: 所有人可寫入(Gateway), 僅授權用戶可讀取
match /logs/{logId} {
 allow read: if request.auth != null;
 allow create: if true; // Gateway 發送訊號
}
}
}
...
...
```

### ### 10.2 API 安全

- CORS 設定(允許特定來源)
- API Key 驗證(可選)
- Rate Limiting(防止 DDoS)
- HTTPS 加密(Firebase 內建)

---

## ## 11. 部署與維運

### ### 11.1 部署流程

```
```bash  
# 1. 部署 Cloud Functions  
cd functions  
npm install  
npm run deploy  
  
# 2. 部署前端  
npm install  
npm run build  
firebase deploy --only hosting  
  
# 3. 設定環境變數  
cp env.example.txt .env  
# 填寫 Firebase 配置  
...  
...
```

詳細部署指南請參考 `DEPLOYMENT.md`。

11.2 監控指標

- API 回應時間
- Cloud Function 執行次數
- Firestore 讀寫量

- 警報發送成功率

12. 技術堆疊

層級	技術	版本
----- ----- -----		
前端框架 React 18.x		
語言 TypeScript 5.x		
建置工具 Vite 5.x		
狀態管理 Zustand 4.x		
樣式 SASS (BEM) 1.83.0		
後端 Firebase Cloud Functions Node.js 18		
資料庫 Firestore -		
身分驗證 Firebase Auth + LINE Login -		
推播通知 LINE Messaging API -		
硬體 BLE Beacon + Gateway -		

13. 未來擴展規劃

Phase 2

- AI 異常偵測(學習長者活動模式)
- 健康數據整合(血壓、血糖)
- 家屬 App(次要通知接收)

Phase 3

- 語音呼叫(緊急雙向通話)
- 跌倒偵測(加速度感測器)
- 社區服務整合(送餐、醫療)

☎ 聯絡資訊

開發團隊: Community Guardian Development Team

技術支援: 請參考 `TROUBLESHOOTING.md`

版本紀錄: 請參考 `PROJECT_SUMMARY.md`

文件結尾

這份 PRD 涵蓋了整個系統的完整設計與實現細節。如需特定功能的詳細說明，請參考對應的技術文檔。