



# PRD — NotGone

**Tagline:** *I'm not gone.*

## 1. 产品愿景

NotGone 是一个为年轻独居者设计的 “生命存在感系统”。

它不是检测“你死没死”，而是每天温柔地确认：

| 你还在，这个世界在看着你。

通过每天 5 秒钟的「生命体征」记录，让用户被看见、被关心，并在慢慢变冷时被轻轻牵住。

## 2. 目标用户

**25-45 岁独居人群**

- 单身、异地、海外生活
- 缺乏稳定照看者
- 不想社交，但害怕“如果我消失没人知道”

## 3. 核心用户需求 (JTBD)

| “我想在一个人生活的世界里，也能被这个世界确认我还在。”

## 4. 核心体验 (MVP)

### 4.1 生命体征打卡 (Home)

用户通过 **长按** 在一个中心圆形控件上记录今天的「生命温度」。

- 长按时间 → 0-100 的生命值
- 松手 → 记录今天状态
- 系统根据区间显示一句温柔文案（不显示数字）

## 生命值语义映射

### ## 生命值语义映射

区间	状态	文案示例
0-20	🖤 冰冷	"我只是在撑着" / "今天很难"
21-40	🌧 偏冷	"很累，有点低" / "需要一点安静"
41-60	☀ 常温	"还行，正常的一天"
61-80	🌿 温暖	"今天挺好" / "有能量"
81-100	🔥 炙热	"很享受" / "有成就感，活得发光"

## JSON

```
[
  {
    "min": 0,
    "max": 20,
    "label": "冰冷",
    "emoji": "🖤",
    "copies": ["我只是在撑着", "今天很难"]
  },
  {
    "min": 21,
    "max": 40,
    "label": "偏冷",
    "emoji": "🌧",
    "copies": ["很累，有点低", "需要一点安静"]
  },
  {
    "min": 41,
    "max": 60,
    "label": "常温",
    "emoji": "☀",
    "copies": ["还行，正常的一天"]
  },
  {
    "min": 61,
    "max": 80,
    "label": "温暖",
    "emoji": "🌿",
    "copies": ["今天挺好", "有能量"]
  },
  {
    "min": 81,
    "max": 100,
    "label": "炙热",
    "emoji": "🔥",
    "copies": ["很享受", "有成就感，活得发光"]
  }
]
```

```
        "min": 41,
        "max": 60,
        "label": "常温",
        "emoji": "☀️",
        "copies": ["还行，正常的一天"]
    },
    {
        "min": 61,
        "max": 80,
        "label": "温暖",
        "emoji": "🌿",
        "copies": ["今天挺好", "有能量"]
    },
    {
        "min": 81,
        "max": 100,
        "label": "炙热",
        "emoji": "🔥",
        "copies": ["很享受", "有成就感，活得发光"]
    }
]
```

## 4.2 趋势页 (Trend)

- 显示最近 7 天生命体征曲线
- 只展示趋势，不做诊断、不做解释

## 4.3 守望者 (Watchers)

用户可添加 1-3 个守望者（邮箱）

字段：

- email
- level (1 = 最亲密, 2 = 朋友, 3 = 互助)

# 5. 温柔关怀升级机制 (核心逻辑)

每日定时任务（后台）检测：

触发条件（任一）：

- 连续 3 天生命值 < 30
- 或 4 天未记录 + 最近一次生命值 < 40

触发流程：

1. App 内提醒

    | “你最近有点冷，我们在这。”

2. 24 小时无响应

    → 给 level=1 的守望者发送温柔邮件

3. 48 小时无响应

    → 升级发送给 level=2 / 3

## 邮件模板（固定）

**Subject**（固定）

- Just checking in 🌱

**Body**（模板 + 变量）

- 变量：{days} = 未更新天数, {trend} = “一直偏低/逐渐变冷/最近不太稳定”（三选一）

Hi,

Someone you care about has been sharing a small daily “life signal” in NotGone.

We noticed they haven't updated **for {days}** days, and their recent trend has been {trend}.

This doesn't mean anything bad — it may simply be a busy week.

If you can, a gentle check-in (a message or a quick call) might mean a lot.

Thank you **for** being there.

这个版本已经足够“有温度”，并且非常安全。

## 6. 付费模式

**一次性付费 App (无订阅)**

用户买的是：

| “被这个世界看见的权利”

## 7. 身份 & 隐私

- 无登录

- 使用匿名 device\_id

- 云端仅存：
  - anon\_id

- 日期

- 生命值

- 守望者邮箱

不存名字、不存日记、不存隐私文本

## 8. 技术架构（海外版）

**Frontend**

- React Native + Expo

**Backend (AWS)**

- API Gateway

- Lambda (Python FastAPI)

- DynamoDB
- EventBridge (每日检测)
- SES (发送邮件)

## API 设计草图 (FastAPI 路由 + DynamoDB 表结构 + 关键逻辑)

无登录、匿名 device\_id、一次性付费、邮件守望者提醒。

### 0) 总体架构 (v1)

- Client : Expo RN
- Backend : API Gateway → Lambda (FastAPI)
- DB : DynamoDB (3 张表)
- Scheduler : EventBridge (每日 1 次) → Lambda (/cron/daily-check)
- Email : AWS SES (向守望者发邮件)

### 1) 认证与匿名身份 (无登录)

- 客户端首次启动生成 anon\_id = `uuid_v4()`
- 存 SecureStore/Keychain
- 每次请求带 Header :
- **X-ANON-ID:** <uuid>
- 后端只校验格式 (uuid)，不做账号体系
- 重要：不要用 IP/设备信息做追踪，减少隐私风险

### 2) DynamoDB 表设计 (v1)

#### 2.1 ng\_users

用途：用户偏好、时区、提醒时间、暂停状态、通知状态

PK : `anon_id` (string)

字段：

- `anon_id` (PK)
- `created_at` (ISO string)
- `timezone` (string, e.g. "Europe/London")

- `reminder_time_local (string, "20:30")` // 可选，v1 可先不做 push，只做未来预留
- `pause_until (ISO string | null)` // 暂停关怀升级
- `last_checkin_date (string, "YYYY-MM-DD" | null)`
- `last_value (number | null)`
- `plan ("paid")` // 你是一次性付费 App, v1 可以默认 paid；或留字段未来做校验
- `updated_at`

v1 你可以直接默认所有用户 `plan="paid"`, 先不做收据校验, 后续再补 (风险取决于你策略)。

## 2.2 ng\_checkins

用途：每天生命值记录（趋势图 + 异常检测）

PK : `anon_id`

SK : `date (string "YYYY-MM-DD")`

字段：

- `anon_id (PK)`
- `date (SK)`
- `value (number 0..100)`
- `label (string, e.g. "冰冷/偏冷/常温/温暖/炙热")` // 服务端或客户端写入都行
- `created_at`

查询模式

- 获取最近 7 天：Query `PK=anon_id, ScanIndexForward=False, Limit=7`
- 获取指定日期：`GetItem (anon_id, date)`
- upsert 今日：`PutItem` 覆盖即可 (允许用户补写/修改)

## 2.3 ng\_watchers

用途：守望者列表 + 上次通知时间（避免重复轰炸）

PK : `anon_id`

SK : `level (number 1..3)` 或 `watcher_id (uuid)`

建议用 **SK=level** (简单、天然限制最多3个) :

- **anon\_id** (**PK**)
- **level** (**SK**, **number 1..3**)
- **label** (**string**) // 用户给守望者的称呼 (包含关系也行), 它允许“妈妈/姐姐/导师”这种不是严格姓名的称呼。)
- **email** (**string**)
- **verified** (**bool**, v1 可不做验证)
- **last\_notified\_at** (**ISO string** | **null**) // 最近一次给该守望者发信时间
- **created\_at**

## 2.4 GSI (用于 cron 扫描)

给 `ng_users` 加 **GSI**

- **GSI1PK** = "ACTIVE" (常量)
- **GSI1SK** = `last_checkin_date` 或 `updated_at`
- cron 用 **GSI Query** (更便宜更快)

## 3) FastAPI 路由设计 (v1)

### 3.1 Health

**GET** /health

返回 : { "ok": true }

### 3.2 用户初始化/获取

**POST** /v1/me/init

用途 : 首次启动创建用户记录 (幂等)

Header : **X-ANON-ID**

Body (可选) :

{ "timezone": "Europe/London" }

Response :

{ "anon\_id": "...", "created": true, "plan": "paid" }

逻辑 :

- 若 `ng_users` 无此 `anon_id` : Create

- 否则：Update timezone (如提供)

**GET /v1/me**

返回用户信息 (偏好/暂停状态)

```
{
  "anon_id": "...",
  "timezone": "Europe/London",
  "pause_until": null,
  "last_checkin_date": "2026-01-14",
  "last_value": 62
}
```

**PATCH /v1/me**

更新用户偏好 (v1 只做暂停模式就够)

Body : { "pause\_until": "2026-01-21T00:00:00Z" }

Response : 更新后的 me

### 3.3 Check-in (生命体征)

**POST /v1/checkins**

Header : X-ANON-ID

Body :

```
{
  "date": "2026-01-14",
  "value": 62
}
```

Response :

```
{
  "date": "2026-01-14",
  "value": 62,
  "bucket": "温暖",
  "copy": "今天挺好"
}
```

**GET /v1/checkins?days=7**

```
返回最近 N 天 (默认 7, 上限比如 90)
{
  "items": [
    { "date": "2026-01-14", "value": 62, "bucket": "温暖" },
    ...
  ]
}
```

### 3.4 Watchers (守望者)

GET /v1/watchers

返回 level 1..3

```
{
  "items": [
    { "level": 1, "label": "妈妈", "email": "a@example.com" },
    { "level": 2, "label": "Alex", "email": "b@example.com" }
  ]
}
```

PUT /v1/watchers/{level}

level = 1..3

Request Body :

```
{
  "label": "妈妈",
  "email": "a@example.com"
}
```

UI 文案：

“你怎么称呼 TA ?”

(比如：妈妈 / Alex / 室友 Lily)

Response

```
{
  "level": 1,
  "label": "妈妈",
  "email": "a@example.com"
}
```

验证：

- email 格式校验
- 限制总数 $\leq 3$  (天然)

**DELETE /v1/watchers/{level}**

删除该 level 守望者

### 3.5 Cron (仅内部调用)

**POST /v1/cron/daily-check**

用途：每天跑一次，找需要关怀升级的用户，发邮件

鉴权：用 **X-CRON-SECRET** (环境变量) 或 API Gateway **IAM** authorizer (更正式)

{ "scanned": 120, "notified": 7 }

## 4) 关键逻辑：触发规则 (v1)

### 4.1 触发条件 (任一命中)

- A : 连续 3 天 value  $< 30$
- B : 连续 no\_checkin\_days  $\geq 4$  且 last\_value  $< 40$

计算方式：

- 取最近 7 天 checkins (最多 7 条)
- 统计最近连续低温天数 (按日期连续性你可以简化：只按记录条连续；v1 可接受)
  - no\_checkin\_days : 今天日期 - last\_checkin\_date

### 4.2 暂停逻辑

- 若 pause\_until 存在且 now  $<$  pause\_until : 跳过

### 4.3 通知升级

- 优先 level=1
- 若 level=1 无, 才 level=2, 再 level=3
- 同一个守望者冷却时间：last\_notified\_at 距离现在  $< 72h \rightarrow$  不重复发

## 5) 邮件发送 (SES) v1

模板固定 (硬编码)，仅插变量：

- {days} : 未更新天数
- {trend} : 简单趋势描述 (三选一：quite low / cooling down / a bit unstable)

v2 再接 AI，必须有黑名单 & 回退模板。

## 数据模型

User: anon\_id, timezone, reminder\_time

Checkin: anon\_id, date, life\_value

Watcher: anon\_id, email, label, level, last\_notified\_at

## 9. AI 扩展 (v2 预留)

- 基于历史生命值生成更贴合用户的一句温柔回应
- 个性化关怀触发阈值

v1：只用规则

v2：引入模型

## 10. 核心产品原则

- 不说“死亡”
- 不说“风险”
- 不说“异常”
- 只说：

| 你还在吗？我们在这。

