

PlantPal: 系统设计模型

团队名称：SproutLab

团队成员：林琪 金恒宇 达思睿 胡宝怡



目录

- 一 项目概述与技术选型
- 二 系统架构与核心子系统设计
- 三 关键设计机制与工程实现
- 四 典型用例与设计模式应用
- 五 系统成效、原型与反思



项目概述与 技术选型



设计概述与技术选型

核心架构

端口与适配器架构 (Hexagonal Architecture)

以领域内核为中心，通过端口定义能力，通过适配器对接数据库、AI 模型与外部服务，实现业务与技术实现解耦。

设计焦点

1. 从逻辑视图到物理部署的可落地性设计

将领域模型、子系统划分与接口契约

完整映射到客户端、云服务与数据存储节点

2. 关键非功能性需求保障

本地识别处理延迟 Latency $\leq 2s$

弱网 / 离线场景下的核心功能可用性

云端 AI 服务异常情况下的系统降级与容错能力

3. 以用户核心使用流程为导向的架构设计

在保证性能、可靠性与可扩展性的前提下，重点优化首次使用与高频操作路径的响应速度与稳定性。

技术栈



移动端：Flutter

选型理由：满足iOS/Android 双端同构需求。利用Skia引擎的高性能渲染能力，支持复杂的“相似植物对比视图”和流畅交互。



后端核心：Java/Spring Boot

选型理由：承载应用服务层。利用Spring生态成熟的依赖注入(DI)和AOP特性，管理复杂的业务规则与事务一致性。



AI 服务：Python/FastAPI

选型理由：作为独立微服务运行，封装Cloud VLM 调用。Python 拥有丰富的向量计算库，便于处理 Embedding 生成与检索。



数据持久化：PostgreSQL&Milvus/Pgvector

选型理由：PostgreSQL 用于存储结构化的植物档案、养护计划等，保证强一致性。Milvus/Pgvector 用于存储植物特征向量，支持高效相似度搜索。

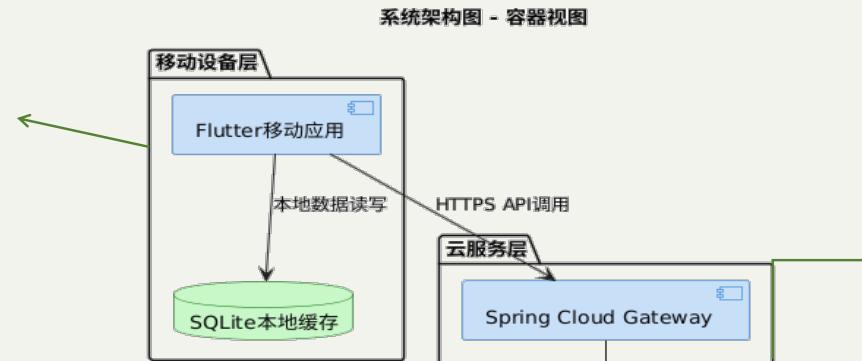


系统架构与 核心子系统设计

架构细化：从逻辑到物理

1 移动设备层：

包含 Flutter App 和本地 SQLite，用于离线缓存和实现 <2s 的本地识别。



2 云服务层：

由 Spring Cloud Gateway (API 入口，负责鉴权限流)，Spring Boot 核心服务（托管 Dossier 和 Care 核心域），以及 FastAPI AI 运配服务（独立节点，对接外部大模型）组成。

3 数据存储层：

双数据库策略，分离 PostgreSQL 的事务数据和 Milvus 的 AI 向量数据。

4 外部AI服务：

与 OpenAI/Google 等强大的视觉语言模型 (VLM) 集成。

核心子系统及其契约

子系统名称	职责简述	核心接口 (Interface)	依赖关系
Identification Subsystem	处理图像识别、特征提取与混淆判别	IdentificationService	依赖 CloudVLM
Care Subsystem	养护规则计算、日程调度与日志记录	ICarePlanService, ICareLogService	依赖 Weather Service
Dossier Subsystem	聚合根管理(CRUD)，维护档案完整性	IDossierRepository	被 CareSubsystem 引用
Diagnosis Subsystem	多轮问诊与风险评估	IDiagnosisSession	依赖 IdentificationSubsystem

如果还需要其他帮助，可以随时告诉我。

场景：系统需调用外部气象服务以调整养护计划。

```
// 外部气象服务端口
// 业务层只依赖此接口
public interface IWeatherProvider {
    /**
     * 获取指定坐标的当前及未来天气简报
     * @return WeatherDTO (包含温度、湿度、降水概率)
     */
    WeatherDTO getLocalWeather(double latitude, double longitude)
        throws ExternalServiceException;
}
```

设计原则：依赖倒置 (Design Principle: Dependency Inversion)

说明：业务逻辑(内核)仅依赖于抽象接口 (Ports)，而不依赖于外部工具(如天气API)的具体实现。这使得更换服务提供商变得轻而易举。

子系统详设：识别服务

Identification Subsystem

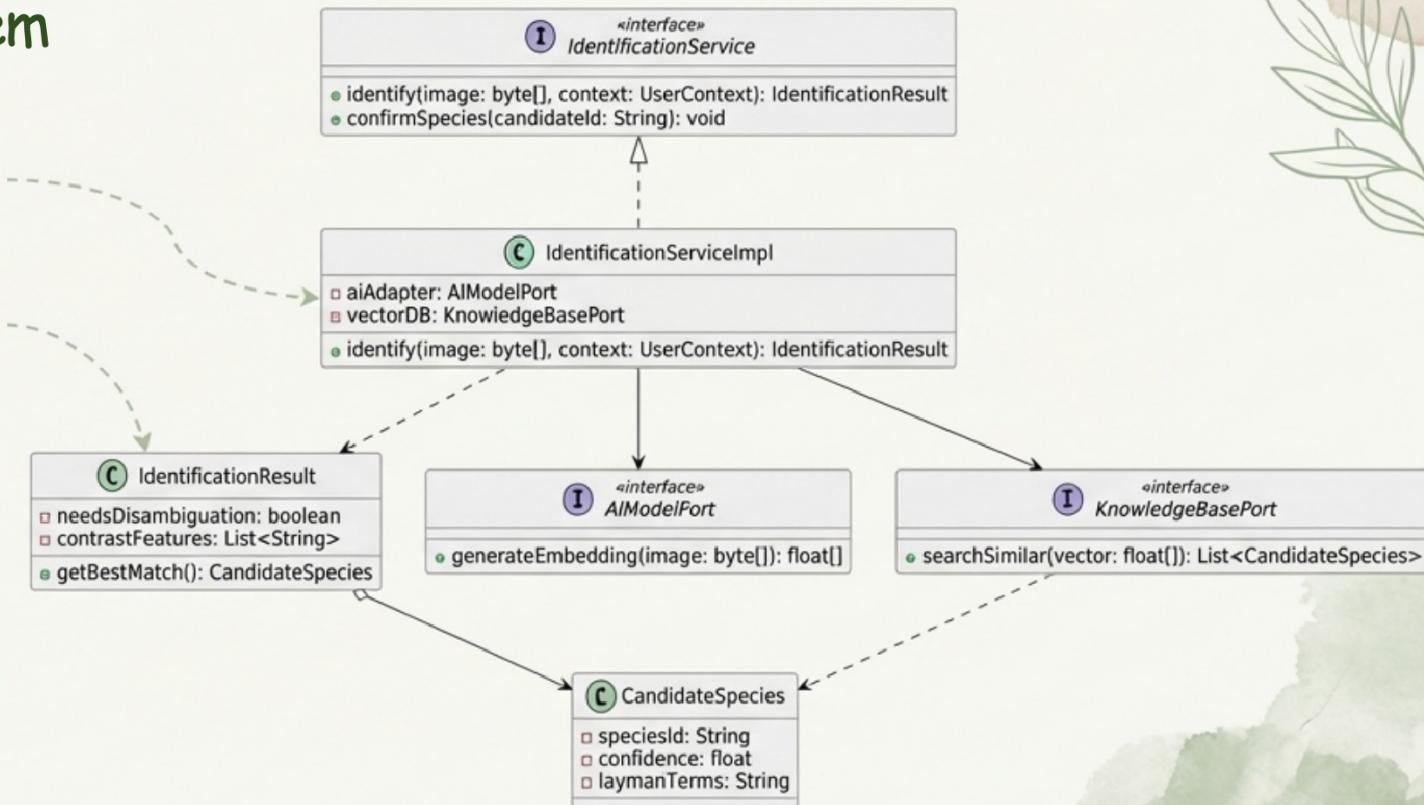
输入：image: byte[], context:

UserContext（包含 geoLocation 和 userPreferences）

输出：IdentificationResult 对象

核心逻辑：

- 当 Top-1 和 Top-2 候选者的置信度差异 < 15% 时，needsDisambiguation 字段将被标记为 true。
- 同时填充 contrastFeatures（如“叶缘锯齿” vs “全缘”）供前端渲染对比视图，实现人机协作消歧。





关键设计机制 与工程实现

关键设计机制(1):持久化与对象关系映射

1. 挑战

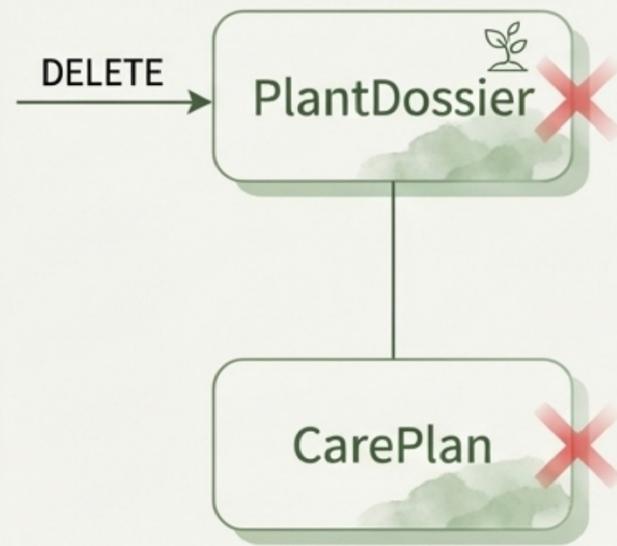
保证 PlantDossier、CarePlan、CareLog 生命周期一致

2. 解决方案

- ① 使用 Spring Data JPA 实现组合关系 & 懒加载
- ② Dossier 作为聚合根管理内部对象

3. 核心机制

- ① 组合关系 & 级联删除
 - @OneToOne(cascade = CascadeType.ALL, orphanRemoval = true)
 - 删除 Dossier → 自动删除 CarePlan, 防止孤儿数据
- ② 懒加载策略
 - FetchType.LAZY
 - CareLog 按需加载, 保证首屏 <60s



防止孤儿数据 (Preventing Orphan Data)

关键设计机制(2): AI服务适配与熔断

1. 挑战

云端 VLM 调用延迟或失败，不能阻塞用户核心流程

2. 解决方案

① 适配器模式 (Adapter Pattern)

- AIModelPort 统一接口
- 云端: CloudVLMAAdapter (调用 GPT-4)
- 本地: LocalTFLiteAdapter (调用 MobileNet)

② 断路器模式 (Circuit Breaker)

- 使用 Resilience4j
- 配置示例:

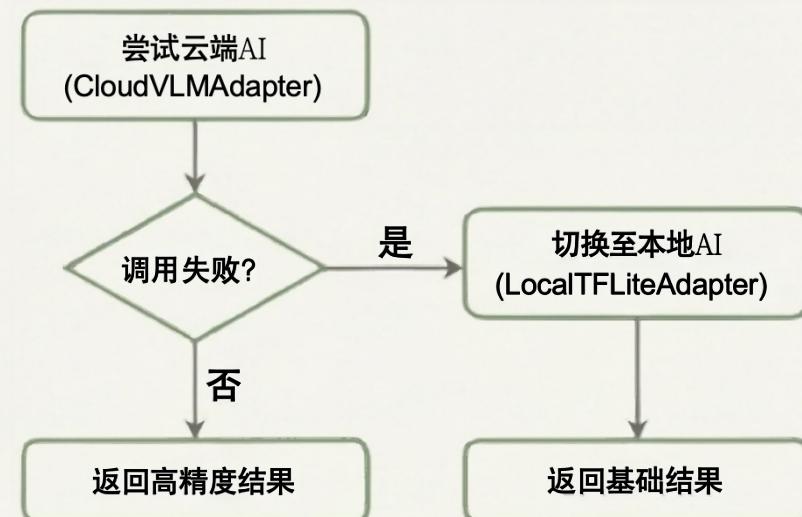
失败率 >50% → 开启断路器

打开状态等待 10 秒

调用超时 5 秒

③ 降级流程 (Fallback)

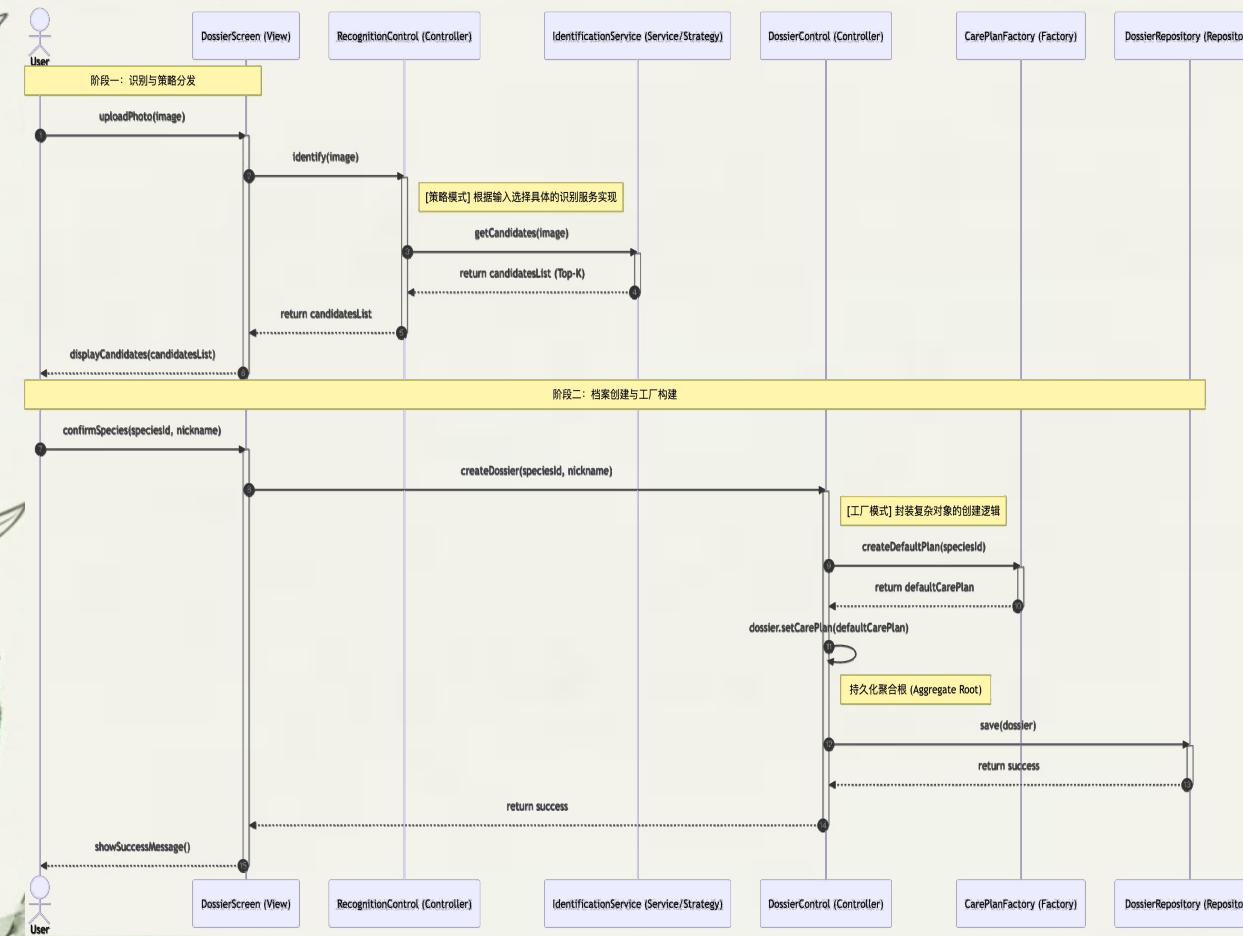
- 云端不可用 → 自动切换本地模型
- 本地模型返回 Top-1 属级分类，保证基本建档





典型用例与 设计模式应用

用例实现 (UC3): 创建植物档案



1. 用例概述

用户上传照片 → 系统识别 → 创建完整植物档案

2. 关键设计模式

① 策略模式 (Strategy Pattern)

- RecognitionControl 根据输入类型（图像/文本）动态选择识别策略
- 识别逻辑与业务控制解耦

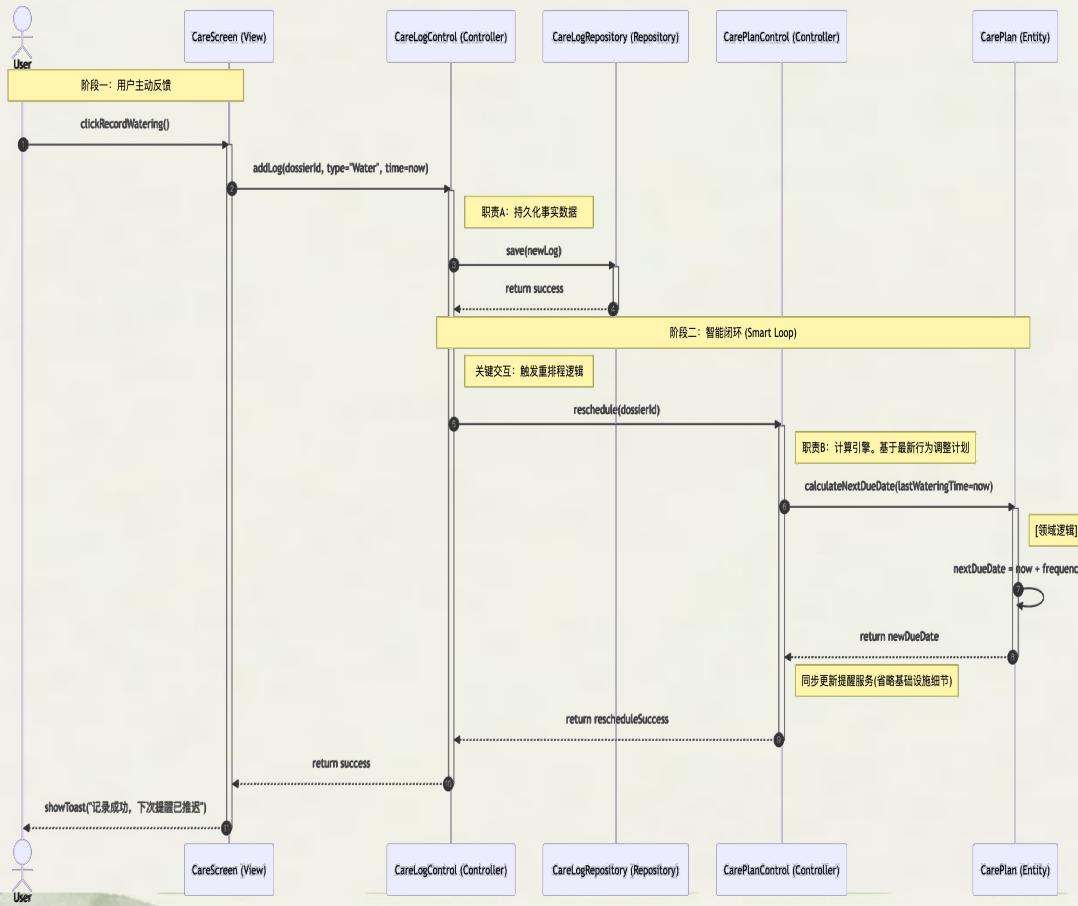
② 工厂模式 (Factory Pattern)

- CarePlanFactory 封装植物养护计划创建逻辑
- 支持不同植物类型（多肉、蕨类等）的默认养护计划

3. 持久化设计

- Dossier 作为聚合根
- 保存时自动级联保存 CarePlan
- 保证数据一致性

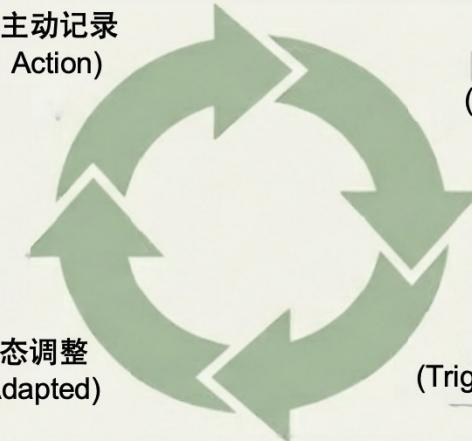
用例实现 (UC5-2): 主动记录与智能闭环



核心理念: 智能反馈闭环

用户主动记录
(User Action)

日志持久化
(Log Persisted)



触发重排程
(Trigger Reschedule)

计划动态调整
(Plan Adapted)

设计原则

① 职责分离

- 记录者 (Log Control): 负责持久化行为数据
 - 规划者 (Plan Control): 负责计算下一次养护时间
- ② 高内聚低耦合, 保证系统可维护性

用例实现 (UC 4): 人机协作式诊断

1. 用例概述

- 多步骤诊断: 上传照片 → 初步分析 → 交互问答 → 最终方案
- 采用人机协作, 用户参与决策, 提高准确性

2. 关键设计模式与原则

① 会话模式 (Session Pattern)

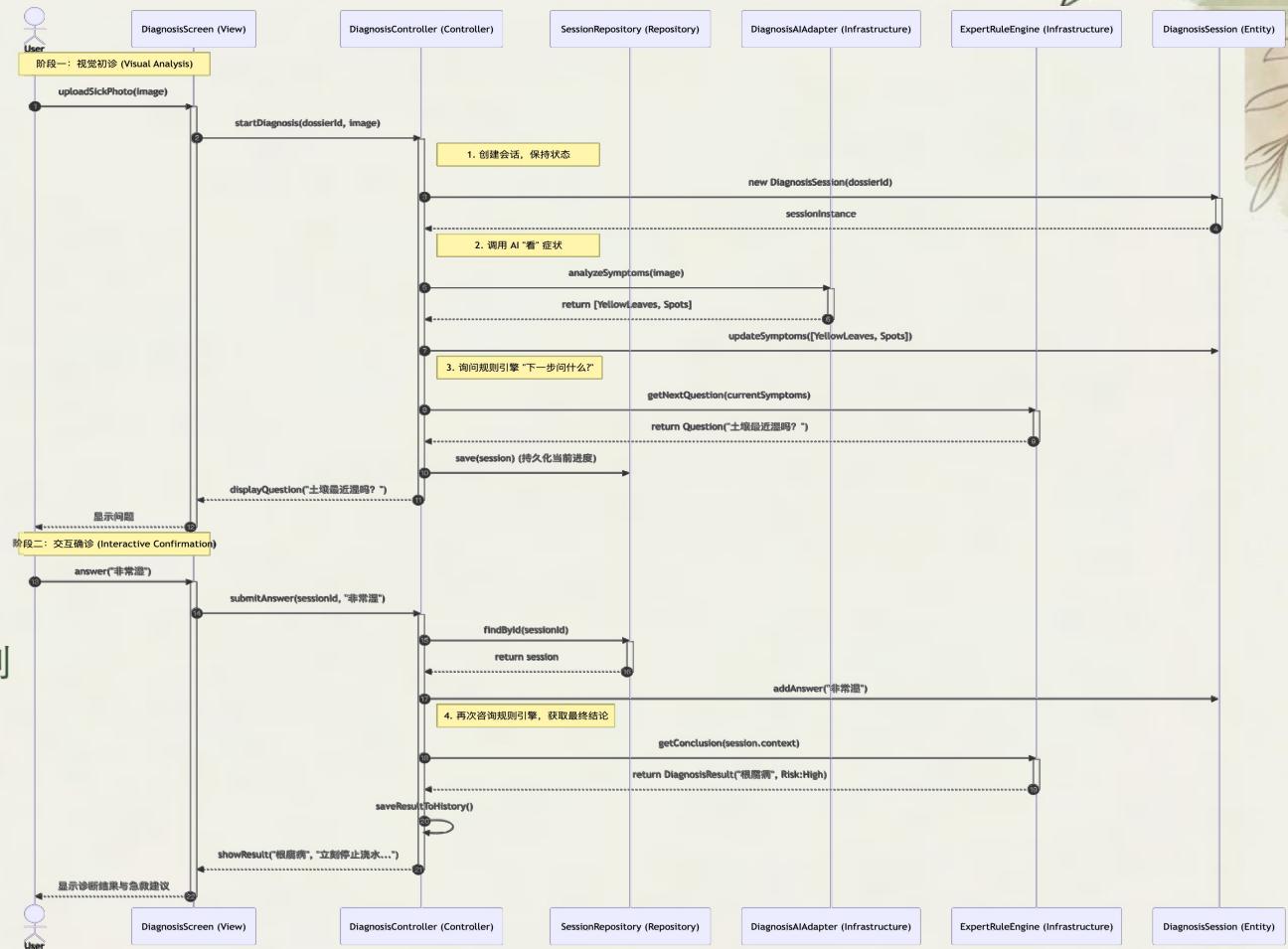
- 记录诊断上下文, 支持多轮问答
- 避免因用户切出 App 导致状态丢失

② 视觉/逻辑分离

- DiagnosisAIAdapter: 负责视觉分析, 输出结构化数据
- ExpertRuleEngine: 负责推理与结论, 规则可动态升级

3. 防御性设计

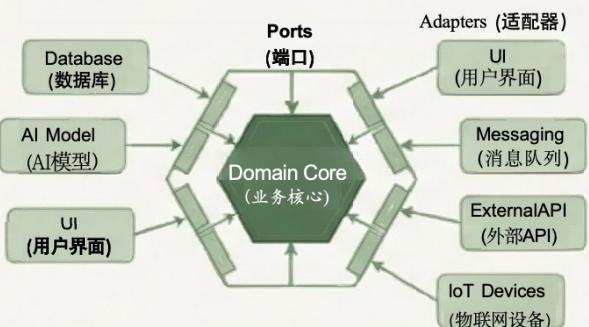
诊断结果保存到 CareLog, 即使 App 崩溃, 数据仍可靠



6

系统成效、 原型与反思

架构风格与关键决策

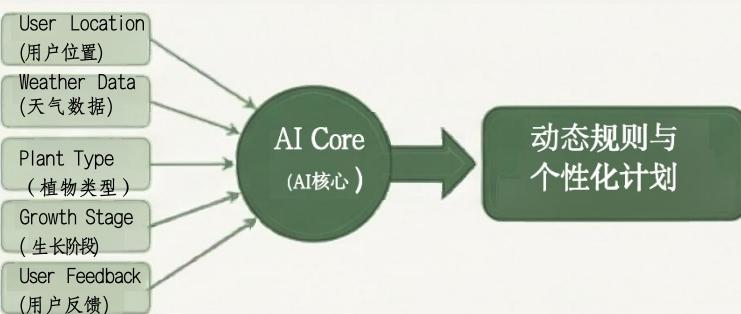


1. 六边形架构

将核心业务逻辑（内核）与易变的技术实现（外壳，如数据库、AI模型）隔离。更换AI模型只需新增一个Adapter，无需修改核心代码，保证业务稳定性。

2. 混合AI推理

- “云端大模型（高精度）+本地小模型（高可用）”的组合策略。
- 解决花鸟市场的“信号黑洞”问题。保证用户在弱网/离线环境下也能“先用起来”，完成核心识别流程。



在线(Online)



Cloud VLM
(高精度)

离线(Offline)



Local TFLite
(高可用)

3. AI 驱动的动态策略引擎

摒弃静态规则表，实现“千植千面”。AI根据用户微环境（地理位置、天气）和行为反馈动态生成个性化养护计划，解决环境差异化难题。

非功能需求：性能与可靠性

性能：本地识别<2s

1. 目标

弱网或离线环境下，识别响应 < 2 秒

2. 实现方案

本地向量匹配 + LRU 缓存：

- App 启动时预加载 Top-50 热门植物特征向量
- TFLite 轻量模型快速计算拍照图片的向量
- 首先在本地缓存中匹配，高相似度直接返回结果

3. 效果

用户体验 “按下快门即出结果”

可靠性：离线优先数据同步

1. 目标

支持离线操作，网络恢复后自动同步

2. 实现方案

本地 SQLite + SyncManager：

- 用户记录立即反馈“保存成功”
- 待同步队列：记录标记为 dirty，持久化存储
- 网络恢复：后台 Worker 批量上传云端，保证最终一致性

3. 效果

断网环境下也能正常记录，数据安全可靠

原型进展与用户界面

The image displays a sequence of six mobile application screens illustrating the development of a plant care application's user interface.

- Screen 1: Onboarding**

Tell us about your plant care experience

Have you ever owned plants before?

 - Yes, I've had plants
 - No, I'm new to plants

Next
- Screen 2: AI Recognition**

AI Recognition

Explore Create Profile Update Status

Get plant names and scientific classification. View detailed plant descriptions. Learn about care difficulty levels...

Recently Recognized

 - Fiddle Leaf Fig
 - Monstera Deliciosa
 - Sna

Explore

Recognition My Plants Profile
- Screen 3: Recognition Result**

Recognition Result

Small Hibiscus

Care Difficulty: ★★★

A flowering shrub native to South Africa. Known for its prolific and long-lasting blooms from spring to fall. Pink or white flowers, ideal for container gardening and topiary.

Care Guide

 - Sunlight: Requires full sun
 - Watering: Water when topsoil is dry. Avoid waterlogging and prolonged drought.
 - Ventilation: Crucial to prevent pests and diseases.
 - Pruning: Deadhead after flowering.

Add to My Plants Update My Plant
- Screen 4: My Plants**

My Plants

sort by filter select

 - Monstie** Monstera Deliciosa
 - Leaf Erikson** Fiddle Leaf Fig
 - Medusa** Snake Plant

Records

 - Oct 20, 2025: Watered Plant
 - Oct 17, 2025: Status Update: Healthy

Water Update
- Screen 5: Profile**

Profile

Emily Carter Joined 2022

My Plants

 - Collection 12 plants
 - Favorites 3 plants

Settings

 - Notifications
 - Privacy
 - Help & Support

Recognition My Plants Profile

人机协作：处理AI的不确定性

场景

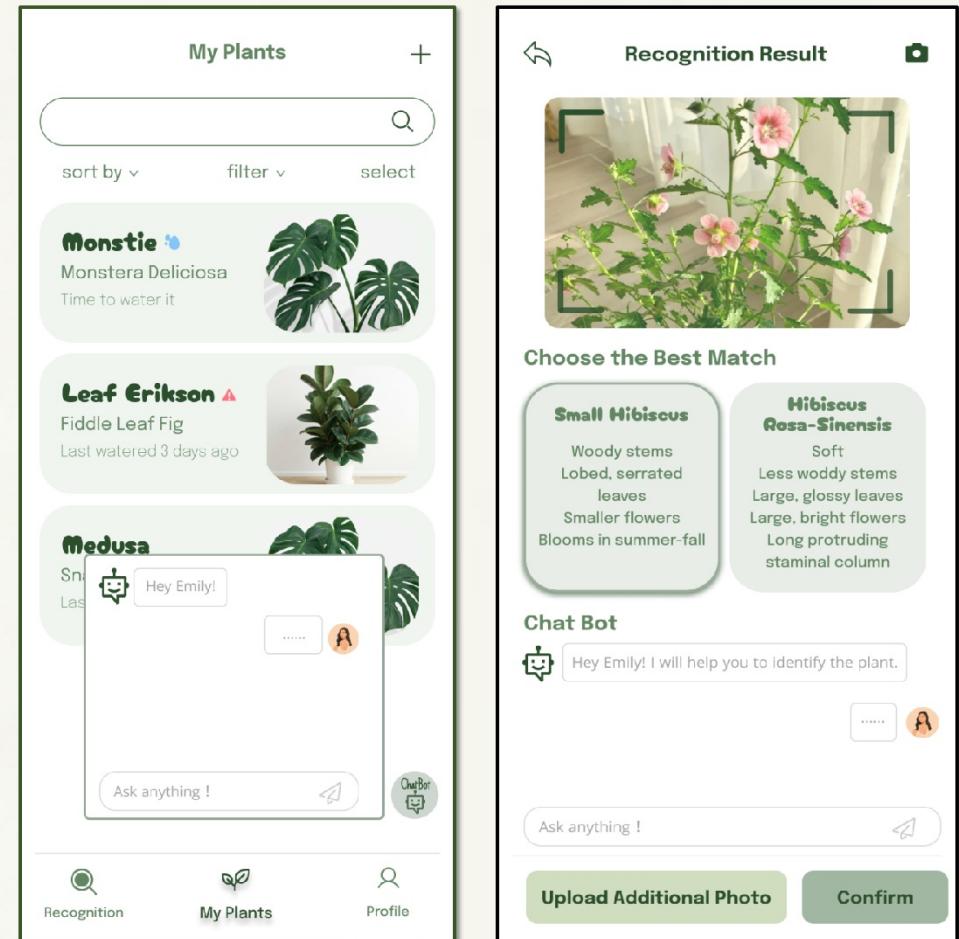
当AI识别结果的置信度差异<15%时，系统进入人工确认引导模式。

核心设计

- 并列对比：清晰展示两种最可能的植物，并列出关键外形特征帮助用户判断
- ChatBot 辅助：用户可与ChatBot 对话，询问两种植物的区别，辅助决策。
- 多种确认方式：
 - Upload Additional Photo:上传更多照片供系统进一步分析。
 - Confirm: 用户根据实物特征，手动选择并确认最终结果。

核心设计理念

通过交互流程弥补算法的不确定性，而非仅仅追求模型精度。



项目自省与待解决问题

项目自省

架构不仅仅是画图：
六边形架构提供核心稳定 + 外部适配器灵活替换。

- AI 不是万能的，流程才是：
- 现有 VLM 模型存在“幻觉”风险，识别或诊断可能出错。
 - 设计 Human-in-the-loop 流程：AI 给出候选集，用户确认，弥补算法不确定性。

- 细节决定落地的可能性：
- 处理断网记录时，必须保证数据唯一性与同步可靠。
 - 引入本地 UUID + SyncManager，确保离线操作也能安全同步至云端。

待解决问题

- 反馈数据噪声：用户延迟浇水 → 系统无法区分“土湿”还是“拖延”，可能误导智能排程
- 视觉引导性能：AI 对相似植物生成高亮对比图像时，端云协作难 <2s

感谢您的聆听

恳请批评指正

团队名称 : SproutLab

团队成员 : 林琪 金恒宇 达思睿 胡宝怡

