PlantCare Ecosystem - Полная документация

Автор: Денис Аниськов
Дата создания: 2025
Последнее обновление: 31 октября 2025
Статус: Production Ready

Содержание

1. Обзор экосистемы
2. PlantCare - Комплексный помощник
3. AI Plants Scanner - AI Сканер растений
4. Технологический стек
5. Сравнение проектов
6. Общая инфраструктура
7. Установка и настройка
8. Использование
9. Архитектура
10. История релизов
11. Roadmap
12. Поддержка и FAQ

Обзор экосистемы

PlantCare Ecosystem — это набор из двух взаимодополняющих Android приложений для ухода за растениями с интеграцией локального и облачного искусственного интеллекта.

Философия проекта

- **Приватность превыше всего** - локальные AI модели через LM Studio
- **Оффлайн-первый подход** - работа без интернета
- **Современный UI/UX** - Material Design 3, плавные анимации
- **Доступность** - поддержка слабовидящих пользователей
- **Мультиплатформенность** - Android + Desktop (Windows)
- **Open Source подход** - чистый, документированный код

Проекты в экосистеме

Проект	Назначение	Платформы	Статус
PlantCare	Комплексный помощник по уходу	Android, Desktop	v1.1
AI Plants Scanner	AI-сканер растений по фото	Android	v1.0

PlantCare - Комплексный помощник

О проекте

PlantCare — мультиплатформенное приложение для комплексного ухода за растениями с интеграцией локального и облачного ИИ.

GitHub: https://github.com/DenisAniskov/PlantCare
Локация: `C:\Users\User\Downloads\PlantCare`
Версия: 1.1.0
Дата релиза: 25 октября 2025

Возможности

1. **Мои растения**

   - Добавление, редактирование и удаление растений
   - Поля: название, тип, примечания
   - Локальное хранение в Room Database
   - **События ухода** с типами:
   - Полив
   - Подкормка
   - Опрыскивание
   - Пересадка
   - Отметка событий как выполненных
   - Автоматические рекомендации по типу растения

2. **Заметки**

   - Текстовые заметки с временными метками
   - Привязка к конкретным растениям

- Быстрое редактирование и удаление
- Сортировка по дате

3. **Справочник растений**

- **10 предустановленных растений** (UTF-8):
- Роза (Rosa)
- Томат (Solanum lycopersicum)
- Орхидея Фаленопсис (Phalaenopsis)
- Орхидея Дендробиум (Dendrobium)
- Фиалка узамбарская (Saintpaulia)
- Монстера деликатесная (Monstera deliciosa)
- Алоэ вера (Aloe vera)
- Спатифиллум (Spathiphyllum)
- Кактус (Cactaceae)
- Фикус Бенджамина (Ficus benjamina)
- Справочник болезней и вредителей
- Советы по уходу
- **LocalRagEngine** - локальный поисковый движок
- Работа полностью оффлайн

4. **Диагностика болезней**

- **13 симптомов** для выбора:
- желтеют листья, коричневые пятна
- мучнистый налёт, вялость
- опадают листья, паутина
- насекомые, гниль
- чёрные точки, деформация листьев
- белые пятна, липкие выделения
- дырки на листьях
- Автоматический подбор заболеваний
- Рекомендации по лечению

5. **AI-ассистент (Онлайн модель)**

- **LM Studio интеграция:**
- Модель: google/gemma-3-12b (7.33 GB)
- Адрес: `http://172.16.0.1:1234`
- OpenAI-совместимый API

- Timeouts: 60s connect, 300s read/write
- **Поддержка облачных API:**
- OpenAI (GPT-4o-mini, GPT-3.5-turbo)
- Groq (llama-3.1-70b-versatile)
- Together AI
- Anthropic Claude
- Chat interface с историей
- System prompt: "Ты — PlantCare Buddy, эксперт по уходу за растениями"

6. Локальная нейросеть (TFLite)

- TensorFlow Lite модель: `plant_disease_mobilenetv2.tflite`
- Python FastAPI сервер: `http://127.0.0.1:8000`
- Endpoints:
- `GET /health` - проверка доступности
- `POST /predict` - распознавание болезней
- Автозапуск сервера из Desktop версии
- Распознавание по фото

7. ☀ Погода

- Автоопределение геолокации:
- ipwho.is
- ip-api.com
- ipapi.co (с fallback)
- **Open-Meteo API** (без ключей):
- Температура, влажность, давление, ветер
- Weather code → описание на русском
- Учёт условий для растений

8. ◐ Тёмная тема

- Переключение светлой/тёмной темы
- Адаптивные цвета Material Design 3
- Анимированная кнопка (300ms fade)

9. Современный UI/UX

- **Плавные анимации:**
- Fade-in заголовков (800ms)
- Slide-in кнопок с задержкой (100-700ms)

- Slide transitions между экранами
- **Градиентные фоны:**
- Primary Gradient (зелёный → светло-зелёный)
- Nature Gradient (мятный → лесной)
- Sunset Gradient (лайм → светло-зелёный → зелёный)
- **Material Icons Extended**
- **Emoji в кнопках** для визуальной навигации
- **Крупные кнопки** (64dp) для доступности
- **Высокий контраст** для слабовидящих

Архитектура (PlantCare)

Модульная структура

```
PlantCare/
├── app/                     # Android приложение
│   ├── data/               # Room entities (Plant, CareEvent)
│   ├── db/                 # Database, DAO, Converters
│   ├── viewmodel/          # PlantCareViewModel
│   ├── ui/                 # Compose screens
│   ├── ai/                 # RemoteAiClient (LM Studio)
│   └── util/               # Utilities
├── core/                    # Kotlin Multiplatform
│   └── LocalRagEngine      # Локальный поиск
├── desktop/                 # Desktop (Compose for Desktop)
│   └── Main.kt             # Entry point
├── shared-ui/               # Общие UI компоненты
│   ├── DesignSystem.kt     # Цвета, градиенты, spacing
│   └── SharedComponents.kt # Кнопки, экраны
└── server/                  # TensorFlow Lite сервер
    └── tflite/
        └── start_server.bat
```

Технологический стек

Android (app/): - Kotlin 1.9.22 - Jetpack Compose + Material Design 3 - MVVM (ViewModel + Room Database) - Min SDK: 24 (Android 7.0) - Target SDK: 34 (Android 14) - Room (SQLite) - Navigation Compose - OkHttp3 - TensorFlow Lite 2.14.0 + GPU - Kotlinx Serialization + Gson

Desktop (desktop/): - Kotlin + Java 17 - Compose for Desktop 1.5.12 - MSI/EXE инсталляторы - Render API: SOFTWARE

Core (core/): - Kotlin Multiplatform (JVM + Android) - LocalRagEngine (RAG-like поиск)

Shared-UI (shared-ui/): - Kotlin Multiplatform Compose - Единая дизайн-система

Дизайн-система PlantCare

Цветовая палитра

```
kotlin
Primary = #4CAF50          // Основной зелёный
PrimaryVariant = #388E3C   // Тёмный зелёный
Secondary = #FF9800        // Акцентный оранжевый
Mint = #81C784             // Мятный
Lime = #9CCC65             // Лаймовый
Forest = #388E3C           // Лесной
Emerald = #00C853          // Изумрудный
```

Градиенты

- **PrimaryGradient** - Brush.linearGradient(Primary → PrimaryLight)
- **SuccessGradient** - Brush.linearGradient(Success → SuccessLight)
- **NatureGradient** - Brush.linearGradient(Mint → Forest)
- **SunsetGradient** - Brush.linearGradient(Lime → PrimaryLight → Primary)

Типографика

```
kotlin
H1 = 34.sp, Bold           // Заголовки первого уровня
H2 = 28.sp, SemiBold       // Заголовки второго уровня
H3 = 22.sp, Medium         // Заголовки третьего уровня
Body1 = 16.sp, Regular     // Основной текст
Body2 = 14.sp, Regular     // Вторичный текст
Caption = 12.sp, Regular   // Подписи
Button = 16.sp, Bold       // Текст кнопок
```

Билды PlantCare

Android APK

```
Путь: app/build/outputs/apk/debug/app-debug.apk
Размер: ~60 MB
Min SDK: 26 (Android 8.0)
```

```
Target SDK: 34 (Android 14)
Архитектуры: arm64-v8a, x86_64
```

Desktop

```
MSI: desktop/build/compose/binaries/main-release/msi/PlantCare-0.1.0.msi
EXE: desktop/build/compose/binaries/main-release/exe/PlantCare-0.1.0.exe
Размер: ~120 MB (с JRE)
ОС: Windows 10/11 (x64)
Java: 17 (включена)
```

Быстрый запуск

```
bash
PlantCare-Desktop.bat  # Автопоиск и запуск desktop версии
```

Документация PlantCare

В корне проекта:

1. **README.md** - Краткое описание
2. **FINAL-INSTALLATION-GUIDE.md** - Полная инструкция (330 строк)
3. **LM-STUDIO-SETUP.md** - Настройка локального ИИ (210 строк)
4. **API-CONFIGURATION.md** - Переключение API (206 строк)
5. **RELEASE_NOTES.md** - История изменений v1.1 (125 строк)
6. **DESKTOP-QUICK-START.txt** - Быстрый старт desktop
7. **INSTALL-DESKTOP.md** - Детальная инструкция desktop

AI Plants Scanner - AI Сканер растений

О проекте

AI Plants Scanner — Android приложение для анализа растений с помощью искусственного интеллекта через фото.

GitHub: https://github.com/DenisAniskov/Ai-Plants-Scanner
Локация: `C:\Users\User\Downloads\AiPlantsScaner`

Версия: 1.0
Дата релиза: 25 октября 2025

Возможности

1. **Фото растений**

- **Камера:**
- CameraX integration
- Разрешение камеры при первом запуске
- Предпросмотр в реальном времени
- **Галерея:**
- Выбор существующих фотографий
- Поддержка всех форматов изображений
- **Обработка:**
- Автоматическое сжатие до оптимального размера
- Конвертация в Base64 для передачи в AI

2. **AI анализ растения**

- **Vision AI через LM Studio:**
- Модель: google/gemma-3-12b (7.33 GB)
- OpenAI-совместимый API endpoint
- Передача изображения в Base64
- Структурированный JSON ответ
- **Определение растения:**
- Название на русском
- Латинское научное название (binomial nomenclature)
- **Оценка здоровья:**
- Статус: здорово / требует внимания / больно
- Health Score: 0-100 баллов
- Визуальная шкала здоровья
- **Обнаружение проблем:**
- Болезни (пятна, гниль, плесень)
- Вредители (тля, паутинный клещ)
- Недостаток питательных веществ
- Стрессовые факторы (переувлажнение, пересушка)

3. **Рекомендации по уходу**

- **Полив:**

- Частота (раз в неделю, 2 раза в неделю)
- Объём воды
- Особенности (тёплая вода, через поддон)
- **Освещение:**
- Интенсивность (прямой свет, рассеянный, тень)
- Продолжительность
- Рекомендуемое расположение
- **Температура:**
- Оптимальный диапазон (°C)
- Минимум и максимум
- Сезонные особенности
- **Удобрения:**
- Тип удобрений
- Частота подкормки
- Дозировка

4. **Лечение и профилактика**

- **Для здоровых растений:** пустой массив
- **Требует внимания:** профилактические меры
- Изменение режима полива
- Корректировка освещения
- Проверка дренажа
- **Больные растения:** конкретный план лечения
- Удаление пораженных частей
- Названия препаратов (Фитоспорин-М, Актара)
- Дозировки (4-5 мл на 1 л воды)
- Частота обработок (3 раза с интервалом 7 дней)
- Пересадка с добавлением биопрепаратов

5. **Интересные факты**

- Научные факты о растении
- Историческая справка
- Полезные свойства
- Особенности роста и размножения

6. **История сканирований**

- Локальное хранение в Room Database

- Фото + результат анализа
- Дата и время сканирования
- Возможность повторного просмотра
- Удаление из истории

7. **Современный интерфейс**

- **Material Design 3:**
- Dynamic colors
- Адаптивные компоненты
- Elevation и shadow
- **Темы:**
- Светлая тема
- Тёмная тема
- Автоматическое переключение по системе
- **Анимации:**
- Плавные переходы
- Loading indicators
- Shimmer effects

Архитектура (AI Plants Scanner)

Структура проекта

```
AiPlantsScaner/
├── app/src/main/java/com/plantscanner/
│   ├── data/
│   │   ├── api/              # LM Studio API сервисы
│   │   │   └── LMStudioApi.kt
│   │   ├── database/         # Room Database
│   │   │   ├── PlantDatabase.kt
│   │   │   └── PlantDao.kt
│   │   ├── model/            # Data классы
│   │   │   ├── PlantScan.kt
│   │   │   └── AnalysisResult.kt
│   │   └── repository/       # Репозитории
│   │       └── PlantRepository.kt
│   ├── ui/
│   │   ├── components/       # Переиспользуемые компоненты
│   │   │   ├── CameraPreview.kt
│   │   │   ├── HealthScoreBar.kt
│   │   │   └── ScanCard.kt
│   │   ├── screens/          # Экраны
│   │   │   ├── HomeScreen.kt
```

```
│   │   │   ├── ResultScreen.kt
│   │   │   └── HistoryScreen.kt
│   │   └── theme/            # Material Design 3 theme
│   │       ├── Color.kt
│   │       ├── Theme.kt
│   │       └── Type.kt
│   ├── viewmodel/            # ViewModels
│   │   └── PlantScanViewModel.kt
│   ├── util/                 # Утилиты
│   │   └── Constants.kt
│   ├── MainActivity.kt
│   └── PlantScannerApp.kt
└── app/build.gradle.kts
```

Технологический стек

Core: - Kotlin 1.9.20 - Min SDK: 26 (Android 8.0) - Target SDK: 34 (Android 14)

UI: - Jetpack Compose - Material Design 3 - Navigation Compose - Material Icons Extended

Architecture: - MVVM (Model-View-ViewModel) - Repository Pattern - Single Activity Architecture

Database: - Room 2.6.1 (SQLite) - KSP (Kotlin Symbol Processing) - Flow for reactive data

Network: - Retrofit 2.9.0 - OkHttp 4.12.0 - Gson converter - Logging interceptor

Camera: - CameraX 1.3.0 - camera-core - camera-camera2 - camera-lifecycle - camera-view

Image Loading: - Coil 2.5.0 (Compose integration)

Async: - Kotlinx Coroutines 1.7.3 - Lifecycle-aware coroutines

API интеграция

LM Studio API

```
kotlin
// Constants.kt
const val LM_STUDIO_BASE_URL = "http://172.16.0.1:1234/"
const val DEFAULT_MODEL = "google/gemma-3-12b"
```

```
// Timeouts
const val CONNECT_TIMEOUT = 30L  // секунд
const val READ_TIMEOUT = 120L    // секунд (для AI обработки)
const val WRITE_TIMEOUT = 60L    // секунд
```

Request format

```
json
{
  "model": "google/gemma-3-12b",
  "messages": [
    {
      "role": "system",
      "content": "Ты - профессиональный ботаник..."
    },
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "Проанализируй это растение"
        },
        {
          "type": "image_url",
          "image_url": {
            "url": "..."
          }
        }
      ]
    }
  ]
}
```

Response format

```
json
{
  "name": "Роза",
  "latinName": "Rosa",
  "healthStatus": "здорово",
  "healthScore": 95,
  "problems": [],
  "careInstructions": {
    "watering": "2 раза в неделю летом, раз в неделю зимой",
    "light": "Прямой солнечный свет 6-8 часов",
    "temperature": "18-24°C",
    "fertilizer": "Комплексное удобрение раз в 2 недели весной и летом"
```

```
  },
  "treatment": [],
  "facts": [
    "Роза - королева цветов",
    "Существует более 30,000 сортов роз",
    "Розовое масло используется в парфюмерии"
  ]
}
```

System Prompt

Файл: `LM_STUDIO_SYSTEM_PROMPT.txt` (175 строк)

Ключевые инструкции:

```
Ты - профессиональный ботаник-эксперт с опытом работы более 20 лет.

ВАЖНЫЕ ПРАВИЛА:
1. Отвечай ТОЛЬКО в формате JSON
2. НЕ добавляй текст до или после JSON
3. НЕ используй markdown блоки
4. НЕ добавляй приветствия или комментарии
5. Используй только русский язык

КРИТЕРИИ ОЦЕНКИ ЗДОРОВЬЯ:
- 90-100: Отличное состояние
- 70-89: Хорошее состояние
- 50-69: Требует внимания
- 30-49: Плохое состояние
- 0-29: Критическое состояние

АНАЛИЗИРУЙ:
1. Вид и сорт растения
2. Цвет и состояние листьев
3. Наличие вредителей
4. Признаки болезней
5. Общее состояние
6. Состояние почвы
```

Исправленные проблемы

Файл: `FIXES_APPLIED.md` (209 строк)

1. Краш при запросе камеры

- **Проблема:** Приложение крашилось при первом запросе разрешения

- **Причина:** Камера запускалась до получения разрешения
- **Решение:** Камера запускается в callback после разрешения

2. **Таймаут на телефоне**

- **Проблема:** На эмуляторе работает, на телефоне таймаут
- **Причина:** IP недоступен через WiFi, малый timeout
- **Решения:**
- Увеличены timeouts (120s read)
- Retry на connection failure
- Детальное логирование
- Понятные сообщения об ошибках

3. **Обработка ошибок**

- Connection Exception → "Проверьте сеть"
- Socket Timeout → "Сервер не отвечает"
- JSON Parse Error → "Некорректный ответ AI"

Билды AI Plants Scanner

Android APK

```
Путь: app/build/outputs/apk/debug/app-debug.apk
Размер: 16.71 MB
Дата: 25.10.2025 22:47
Min SDK: 26 (Android 8.0)
Target SDK: 34 (Android 14)
IP конфигурация: 192.168.1.126:1234
Подпись: Debug key
```

Документация AI Plants Scanner

В корне проекта:

1. **README.md** - Полная документация (161 строка)
2. **QUICK_START.md** - Быстрый старт (133 строки)
3. **START_HERE.md** - Готово к запуску (236 строк)
4. **BUILD_SUCCESS.md** - После сборки (261 строка)
5. **FIXES_APPLIED.md** - Исправления (209 строк)
6. **NETWORK_TROUBLESHOOTING.md** - Решение проблем с сетью
7. **LM_STUDIO_SYSTEM_PROMPT.txt** - System prompt (175 строк)

8. **ICON_GENERATION_PROMPT.txt** - Промпт для иконки

Технологический стек (сводная таблица)

Технология	PlantCare	AI Plants Scanner
Язык	Kotlin 1.9.22	Kotlin 1.9.20
Min SDK	24 (Android 7.0)	26 (Android 8.0)
Target SDK	34 (Android 14)	34 (Android 14)
UI Framework	Jetpack Compose	Jetpack Compose
Design System	Material Design 3	Material Design 3
Architecture	MVVM	MVVM
Database	Room 2.6.1	Room 2.6.1
Network	OkHttp3	Retrofit + OkHttp
Navigation	Navigation Compose	Navigation Compose
Camera	-	CameraX 1.3.0
Image Loading	-	Coil 2.5.0
ML	TFLite 2.14.0 + GPU	-
Multiplatform	(Android + Desktop)	(только Android)
Gradle	8.x	8.x
Java Target	17	17

⚖ Сравнение проектов

PlantCare vs AI Plants Scanner

Критерий	PlantCare	AI Plants Scanner
Назначение	Комплексный уход	AI-анализ по фото
Платформы	Android + Desktop	Android только
Размер APK	~60 MB	~17 MB
Функционал	9 модулей	Фокус на AI сканирование
База данных	Растения + События + Заметки	История сканирований
AI модели	LM Studio + TFLite + Cloud	LM Studio
Справочники	10 растений + болезни + вредители	-
Камера	Нет	CameraX
Offline работа	Полная	⚠ Только с локальным AI

Критерий	PlantCare	AI Plants Scanner
Сложность	Высокая (multimodule)	Средняя (single module)
Desktop версия	MSI/EXE	
Погода		
Диагностика	По симптомам	По фото

Когда использовать

PlantCare: - Нужен комплексный уход за коллекцией растений - Планирование событий (полив, подкормка) - Ведение заметок и дневника - Работа на компьютере (Desktop версия) - Доступ к справочникам без интернета - Погодные условия для растений

AI Plants Scanner: - Быстрая идентификация неизвестного растения - Диагностика болезней по фото - Получение мгновенных рекомендаций - Фокус на AI-анализе - Легкое приложение (~17 MB) - Сканирование множества растений

Взаимодополнение

Проекты отлично работают вместе:

1. **AI Plants Scanner** - идентифицируй растение по фото
2. **PlantCare** - добавь в коллекцию и планируй уход
3. **PlantCare AI Assistant** - задавай дополнительные вопросы
4. **PlantCare TFLite** - локальная диагностика болезней

Общая инфраструктура

LM Studio - Локальный AI сервер

Использование: - PlantCare: AI-ассистент + Chat - AI Plants Scanner: Vision анализ растений

Конфигурация:

```
yaml
Host: 0.0.0.0 или 172.16.0.1
Port: 1234
Model: google/gemma-3-12b (7.33 GB)
API: OpenAI-compatible
```

```
Endpoints:
  - GET  /v1/models
  - POST /v1/chat/completions
  - POST /v1/completions
  - POST /v1/embeddings
```

Установка: 1. Скачать с https://lmstudio.ai 2. Загрузить модель Gemma-3-12b 3. Запустить Local Server 4. Убедиться в доступности из сети

Требования: - Windows 10/11, macOS, Linux - 16+ GB RAM для Gemma-3-12b - 10 GB свободного места - GPU ускорение (опционально)

Network Security Configuration

Файл: `app/src/main/res/xml/network_security_config.xml`

```
xml
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="false">172.16.0.1</domain>
        <domain includeSubdomains="false">192.168.1.126</domain>
        <domain includeSubdomains="true">192.168.0.0</domain>
        <domain includeSubdomains="true">10.0.0.0</domain>
        <domain includeSubdomains="false">localhost</domain>
        <domain includeSubdomains="false">127.0.0.1</domain>
    </domain-config>
</network-security-config>
```

Зачем: Разрешение HTTP (не HTTPS) для локальной сети.

Firewall правила

Windows:

```
powershell
# Разрешить LM Studio (порт 1234)
netsh advfirewall firewall add rule name="LM Studio" dir=in action=allow protocol=TCP lo

# Проверить правила
netsh advfirewall firewall show rule name="LM Studio"

# Временно отключить (для теста)
Set-NetFirewallProfile -Profile Private -Enabled False
```

```
# Включить обратно
Set-NetFirewallProfile -Profile Private -Enabled True
```

Установка и настройка

Общие требования

Hardware: - Android устройство или эмулятор (API 24+ для PlantCare, API 26+ для Scanner) - ПК с Windows/macOS/Linux для LM Studio - 16+ GB RAM для AI моделей

Software: - Android Studio Hedgehog (2023.1.1+) - JDK 17+ - LM Studio - Git (опционально)

Шаг 1: Установка LM Studio

1. Скачать: https://lmstudio.ai
2. Установить и запустить
3. Вкладка "Discover" → Поиск "gemma-3-12b"
4. Скачать модель (7.33 GB)
5. Вкладка "Local Server" → Start Server
6. Настроить: - Address: `0.0.0.0` (не localhost!) - Port: `1234` - Model: google/gemma-3-12b

Шаг 2: Настройка сети

Узнать IP компьютера

```
bash
# Windows
ipconfig

# macOS/Linux
ifconfig
```

Найти IPv4 адрес (например, `192.168.1.100`)

Проверить доступность

На телефоне открыть браузер:

```
http://192.168.1.100:1234/v1/models
```

Должен появиться JSON с информацией о модели.

Шаг 3: Настройка PlantCare

Android APK

```
bash
cd C:\Users\User\Downloads\PlantCare

# Изменить IP (если нужно)
# Файл: app/src/main/java/com/example/plantcare/ai/RemoteAiClient.kt
# private const val DEFAULT_BASE_URL = "http://ВАШ_IP:1234"

# Собрать APK
gradlew assembleDebug

# Установить
adb install app/build/outputs/apk/debug/app-debug.apk
```

Desktop

```
bash
# MSI installer
gradlew :desktop:packageMsi

# EXE portable
gradlew :desktop:packageExe

# Быстрый запуск
PlantCare-Desktop.bat
```

Шаг 4: Настройка AI Plants Scanner

```
bash
cd C:\Users\User\Downloads\AiPlantsScaner

# Изменить IP (если нужно)
# Файл: app/src/main/java/com/plantscanner/util/Constants.kt
# const val LM_STUDIO_BASE_URL = "http://ВАШ_IP:1234/"

# Собрать APK
gradlew.bat assembleDebug
```

```
# Установить
adb install -r app/build/outputs/apk/debug/app-debug.apk
```

Шаг 5: Первый запуск

PlantCare

1. Открыть приложение
2. Главное меню → AI-ассистент
3. Задать вопрос: "Как ухаживать за розой?"
4. Подождать 15-30 секунд
5. Получить ответ от AI!

AI Plants Scanner

1. Открыть приложение
2. Разрешить доступ к камере
3. Нажать синюю кнопку (фото) или зелёную (галерея)
4. Сделать/выбрать фото растения
5. Подождать анализ (15-30 секунд)
6. Изучить результаты!

Использование

Сценарии использования

Сценарий 1: Новое растение

1. **AI Plants Scanner:** - Сделать фото растения - Получить название и рекомендации
2. **PlantCare:** - Добавить в "Мои растения" - Создать график полива на основе рекомендаций - Добавить заметку с начальным состоянием

Сценарий 2: Диагностика проблемы

1. **AI Plants Scanner:** - Сфотографировать больное растение - Получить диагноз и план лечения

2. **PlantCare:** - Отметить в заметках: "Обнаружена проблема X" - Использовать Диагностику по симптомам - Изучить справочник болезней - Применить лечение из рекомендаций Scanner

Сценарий 3: Обучение

1. **PlantCare:** - Открыть Справочник - Изучить 10 популярных растений - Прочитать о болезнях и вредителях
2. **PlantCare AI-ассистент:** - Задать вопросы: "Почему желтеют листья?" - Получить развёрнутые ответы

☀ **Сценарий 4: Учёт погоды**

1. **PlantCare:** - Открыть раздел "Погода" - Автоопределение местоположения - Проверить температуру и влажность
2. **Корректировка ухода:** - Увеличить полив в жару - Переместить растения при холоде - Опрыскивать при низкой влажности

Типичные вопросы к AI-ассистенту

PlantCare AI-ассистент: - "Как часто поливать кактус?" - "Почему у монстеры желтеют листья?" - "Какие удобрения нужны розе?" - "Признаки переувлажнения растения" - "Как размножить фиалку?"

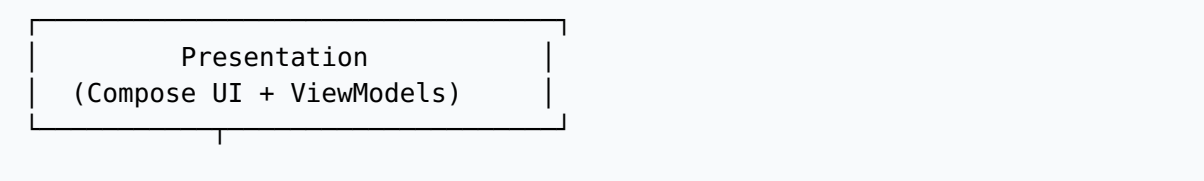
AI Plants Scanner (через фото): - "Что это за растение?" - "Здорово ли это растение?" - "Какие болезни у этого растения?" - "Как лечить эту проблему?"

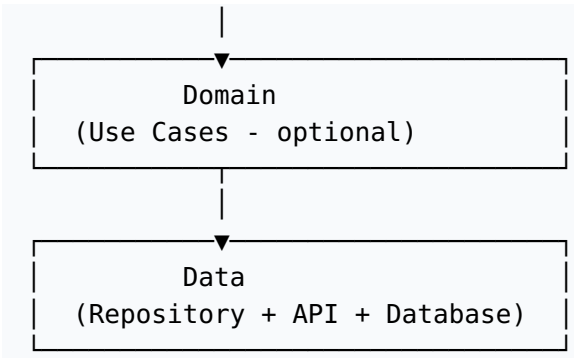
Архитектура (детально)

Общие принципы

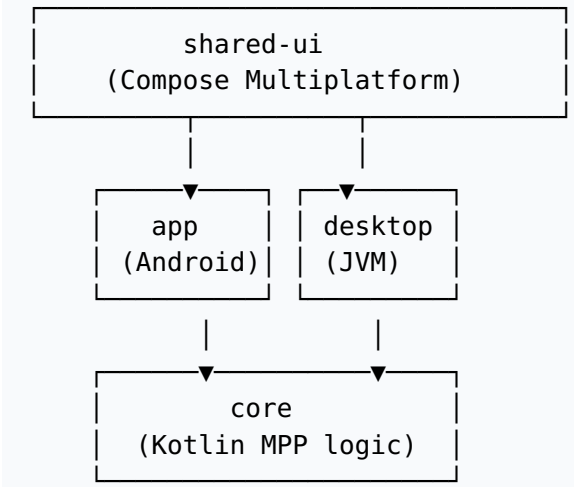
Оба проекта следуют: - **MVVM** (Model-View-ViewModel) - **Single Activity** с Jetpack Compose - **Repository Pattern** для data layer - **Dependency Injection** (manual, без Hilt/Koin) - **Clean Architecture** principles

Слои архитектуры





PlantCare Multiplatform



Data Flow

PlantCare:

```
UI (Compose)
  ↓ User action
ViewModel
  ↓ Call repository
Repository
  ↓ Query
Room Database ← → Local data
  ↓ or
API Client ← → LM Studio / Cloud AI
  ↓
Response
  ↑ LiveData/Flow
ViewModel
  ↑ State
UI (Compose) → Recompose
```

AI Plants Scanner:

```
HomeScreen (Compose)
  ↓ Take photo
CameraX → Image URI
  ↓
ViewModel.analyzeImage(uri)
  ↓
Repository.analyzePlant(uri)
  ↓ Load bitmap
Image → Base64
  ↓ API call
LM Studio (Vision AI)
  ↓ JSON response
Parse to PlantScan
  ↓ Save to DB
Room Database
  ↓ Emit result
Flow<Result<PlantScan>>
  ↑ Collect
ViewModel (State)
  ↑ Observe
ResultScreen → Display
```

История релизов

PlantCare

v1.1.0 (25 октября 2025) - UI/UX Update

Основные улучшения: - Единая дизайн-система (shared-ui) - Плавные анимации на главном экране - Поддержка тёмной темы - Градиентные фоны - Material Icons + Emoji в кнопках - Увеличенная высота кнопок (64dp)

Технические: - Расширена цветовая палитра (14 цветов) - Добавлены 5 градиентов - Исправлен дублированный код в PlantCareApp - Добавлен маршрут Neural Screen - 589 файлов изменено, +8307 строк

Документация: - FINAL-INSTALLATION-GUIDE.md (330 строк) - LM-STUDIO-SETUP.md (210 строк) - API-CONFIGURATION.md (206 строк) - RELEASE_NOTES.md (125 строк)

v1.0.0 (июль 2025) - Initial Release

- Android приложение (Jetpack Compose)
- Room Database (растения, события, заметки)
- 10 растений в справочнике
- Базовый UI (без анимаций)
- LM Studio интеграция
- Оффлайн работа

AI Plants Scanner

v1.0 (25 октября 2025) - Production Release

Функционал: - CameraX интеграция - Галерея поддержка - AI анализ через LM Studio - Определение растения - Оценка здоровья (0-100) - Рекомендации по уходу - План лечения болезней - Интересные факты - История сканирований (Room) - Material Design 3 - Светлая/тёмная темы

Исправления: - Краш при запросе камеры - Таймаут на телефоне (увеличены до 120s) - Детальное логирование - Retry на connection failure - Понятные сообщения об ошибках

Документация: - README.md (161 строка) - QUICK_START.md (133 строки) - BUILD_SUCCESS.md (261 строка) - FIXES_APPLIED.md (209 строк) - LM_STUDIO_SYSTEM_PROMPT.txt (175 строк)

Roadmap

PlantCare - Планы развития

v1.2 (Q1 2026)

- [ ] **Напоминания о поливе** через WorkManager
- [ ] **Push уведомления** о событиях
- [ ] **Виджеты** для главного экрана
- [ ] **Фото растений** в карточках
- [ ] **Экспорт данных** в JSON/CSV

v1.3 (Q2 2026)

- [ ] **Синхронизация** между устройствами (Firebase)

- [ ] **Облачный backup** данных
- [ ] **Мультиаккаунт** с разными коллекциями
- [ ] **Социальные функции** (поделиться растением)
- [ ] **Графики роста** и статистика

v2.0 (Q3 2026)

- [ ] **iOS версия** через Kotlin Multiplatform Mobile
- [ ] **Web версия** через Compose for Web
- [ ] **Интеграция с Plant.id API**
- [ ] **AR режим** для размещения растений
- [ ] **Голосовой ассистент**

AI Plants Scanner - Планы развития

v1.1 (Q1 2026)

- [ ] **Облачные AI** (Google Gemini, OpenAI GPT-4 Vision)
- [ ] **Переключение между AI** (локальный/облачный)
- [ ] **Batch анализ** (несколько фото сразу)
- [ ] **Сравнение состояния** растения во времени
- [ ] **Экспорт результатов** в PDF

v1.2 (Q2 2026)

- [ ] **Оффлайн режим** с TensorFlow Lite моделью
- [ ] **Локальная модель** для определения растений
- [ ] **Ускорение анализа** через quantization
- [ ] **Распознавание вредителей** отдельной моделью
- [ ] **Рекомендации удобрений** по NPK

v2.0 (Q3 2026)

- [ ] **Интеграция с PlantCare** (единое приложение)
- [ ] **Дневник растений** с фото
- [ ] **Таймлапс роста** из фотографий
- [ ] **Социальная сеть** садоводов
- [ ] **Геймификация** (достижения, уровни)

Общие улучшения

Ближайшие (2025-2026)

- [ ] **Единая кодовая база** (слияние проектов?)
- [ ] **CI/CD** через GitHub Actions
- [ ] **Автоматические тесты** (Unit, UI)
- [ ] **Localization** (English, Español)
- [ ] **Accessibility** улучшения (TalkBack)

Долгосрочные (2026+)

- [ ] **ML модели на устройстве** (TFLite)
- [ ] **Federated Learning** для улучшения моделей
- [ ] **Blockchain** для сертификации растений
- [ ] **IoT интеграция** (датчики влажности, света)
- [ ] **Smart pot** интеграция

Поддержка и FAQ

Общие вопросы

Q: Нужен ли интернет для работы приложений?

A: - **PlantCare:** Нет, работает полностью оффлайн (кроме AI-ассистента и погоды) - **AI Plants Scanner:** Нужен для подключения к LM Studio или облачному AI

Q: Как работает локальный AI?

A: Через LM Studio на вашем ПК. Данные не уходят в интернет, всё локально.

Q: Можно ли использовать облачный AI?

A: Да, оба приложения поддерживают OpenAI, Groq, Claude и другие.

Q: Сколько места занимают приложения?

A: - PlantCare: ~60 MB APK, ~150 MB установленное - AI Plants Scanner: ~17 MB APK, ~50 MB установленное - LM Studio модель: 7.33 GB (Gemma-3-12b)

Q: Какая точность AI распознавания?

A: Зависит от модели: - Gemma-3-12b: 85-90% для популярных растений - GPT-4 Vision: 95%+ для большинства растений - Локальная TFLite: 70-80% для болезней

PlantCare FAQ

Q: Как добавить растение?

A: Мои растения → кнопка + → заполните поля → Сохранить

Q: Как работает диагностика?

A: Выберите симптомы из списка → Подобрать диагноз → Результаты из справочника

Q: Можно ли импортировать данные?

A: Пока нет, запланировано в v1.2

Q: Как работает погода?

A: Автоопределение по IP → Open-Meteo API → температура, влажность, давление

Q: Поддерживается ли синхронизация?

A: Пока нет, запланировано в v1.3 через Firebase

AI Plants Scanner FAQ

Q: Почему анализ долгий?

A: Зависит от мощности ПК. Gemma-3-12b требует 15-30 секунд на слабых ПК.

Q: Можно ли использовать без LM Studio?

A: Да, измените URL на облачный API (OpenAI, Groq).

Q: Сохраняются ли фотографии?

A: Да, в Room Database с результатами анализа.

Q: Какие растения распознаёт?

A: Большинство комнатных и садовых растений. Точность зависит от модели.

Q: Можно ли сканировать несколько растений?

A: Пока по одному. Batch анализ запланирован в v1.1.

Решение проблем

Connection Failed / Timeout

Причины: 1. LM Studio не запущен 2. Неправильный IP адрес 3. Устройства в разных сетях 4. Firewall блокирует порт 1234

Решения:

```
bash
# 1. Проверить LM Studio
# Убедитесь, что сервер работает на 0.0.0.0:1234

# 2. Узнать IP ПК
ipconfig  # Windows
ifconfig  # macOS/Linux

# 3. Проверить в браузере телефона
http://ВАШ_IP:1234/v1/models

# 4. Добавить Firewall правило
netsh advfirewall firewall add rule name="LM Studio" dir=in action=allow protocol=TCP lo

# 5. Изменить IP в коде
# PlantCare: app/src/.../ai/RemoteAiClient.kt
# Scanner: app/src/.../util/Constants.kt

# 6. Пересобрать APK
gradlew assembleDebug
```

Camera Permission Denied

Решение: 1. Настройки Android → Приложения → AI Plants Scanner 2. Разрешения → Камера → Разрешить 3. Перезапустить приложение

CLEARTEXT not permitted

Решение: 1. Проверить `AndroidManifest.xml`:

```
xml
android:networkSecurityConfig="@xml/network_security_config"
```

1. Проверить `res/xml/network_security_config.xml`:

```
xml
<domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="false">ВАШ_IP</domain>
</domain-config>
```

1. Пересобрать APK

Gradle Sync Failed

Решение:

```
bash
# Очистить кэш
gradlew clean

# Invalidate caches в Android Studio
File → Invalidate Caches → Invalidate and Restart

# Проверить интернет соединение
# Проверить JDK 17+
```

Room Database ошибки

Решение:

```
bash
# Очистить данные приложения
adb shell pm clear com.example.plantcare
adb shell pm clear com.plantscanner
```

```
# Переустановить APK
adb install -r app-debug.apk
```

Логирование и диагностика

PlantCare

```
bash
# Android Studio Logcat
adb logcat -s PlantCareViewModel:D RemoteAiClient:D PlantRepository:D

# Сохранить логи в файл
adb logcat > plantcare_logs.txt
```

AI Plants Scanner

```
bash
# Фильтр: PlantRepository
adb logcat -s PlantRepository:D

# Подробные логи
adb logcat *:V | grep "plantscanner"
```

Контакты и поддержка

Автор: Денис Аниськов
GitHub: - https://github.com/DenisAniskov/PlantCare - https://github.com/DenisAniskov/Ai-Plants-Scanner

Инструменты: - Создано с помощью **Droid AI** (Factory.ai)

Поддержка: 1. Проверить документацию в репозитории 2. Изучить TROUBLESHOOTING.md 3. Проверить GitHub Issues 4. Связаться через GitHub

---

Статистика проектов

PlantCare

- **Модули:** 4 (app, core, desktop, shared-ui)
- **Экраны:** 15+
- **Строк кода:** 8307+ (последнее обновление v1.1)

- **Зависимости:** ~20 библиотек
- **Документация:** 7 файлов, 1000+ строк

AI Plants Scanner

- **Модули:** 1 (app)
- **Экраны:** 3 (Home, Result, History)
- **Строк кода:** ~3000
- **Зависимости:** ~15 библиотек
- **Документация:** 8 файлов, 1200+ строк

Общая экосистема

- **Проекты:** 2
- **Платформы:** Android + Desktop
- **Всего кода:** 11000+ строк
- **Документации:** 15 файлов, 2200+ строк
- **GitHub Stars:** ⭐ (поставьте!)

Для разработчиков

Структура кода

Соглашения: - Kotlin coding conventions - Compose best practices - MVVM architecture - Repository pattern - Clean code principles

Комментарии: - KDoc для public API - Inline comments для сложной логики - TODO для будущих улучшений

Именование: - PascalCase для классов - camelCase для функций и переменных - SCREAMING_SNAKE_CASE для констант - snake_case для ресурсов

Как внести вклад

1. **Fork репозитория**
2. **Создать ветку** feature/your-feature
3. **Внести изменения**
4. **Написать тесты**
5. **Создать Pull Request**

Что приветствуется: - Исправление багов - Новые функции - Улучшение UI/UX - Оптимизация производительности - Документация - Переводы

Сборка из исходников

PlantCare:

```
bash
git clone https://github.com/DenisAniskov/PlantCare.git
cd PlantCare

# Android
./gradlew assembleDebug

# Desktop
./gradlew :desktop:packageMsi
```

AI Plants Scanner:

```
bash
git clone https://github.com/DenisAniskov/Ai-Plants-Scanner.git
cd Ai-Plants-Scanner

./gradlew.bat assembleDebug
```

Тестирование

Unit тесты:

```
bash
./gradlew test
```

Instrumented тесты:

```
bash
./gradlew connectedAndroidTest
```

Lint:

```
bash
./gradlew lint
```

Лицензия

PlantCare: Лицензия не указана (см. GitHub)
AI Plants Scanner: MIT License - свободное использование

Заключение

PlantCare Ecosystem — это современная платформа для ухода за растениями с фокусом на: - **Приватность** (локальный AI) - **Удобство** (современный UI/UX) - **Функциональность** (комплексный уход + AI анализ) - **Доступность** (оффлайн, бесплатно)

Используйте оба приложения вместе для максимальной эффективности: - **AI Plants Scanner** - быстрая идентификация и диагностика - **PlantCare** - комплексный уход и планирование

Приятного использования!

Дата документа: 31 октября 2025
Версия документа: 1.0
Автор: Денис Аниськов.