

PlantCare — Полная документация

Автор: Денис Аниськов

Версия: 1.2

Дата: Апрель 2026

Статус: Production Ready

Содержание

1. [Обзор проекта]
 2. [Возможности]
 3. [Архитектура AI]
 4. [Технологический стек]
 5. [Структура проекта]
 6. [Установка и сборка]
 7. [Использование]
 8. [API-интеграции]
 9. [Минимальные требования]
 0. [История версий]
 1. [Устранение неполадок]
 2. [Лицензия и контакты]
-

1. Обзор проекта

1.1 Описание

PlantCare — это кроссплатформенное приложение для Android и Windows, предназначенное для помощи пользователям в уходе за комнатными и садовыми растениями. Приложение сочетает в себе функции управления коллекцией растений, AI-ассистента для консультаций, системы диагностики заболеваний, расширенного справочника растений и прогноза погоды.

Философия проекта основана на принципах доступности и конфиденциальности. Все данные хранятся локально на устройстве пользователя, без передачи на внешние серверы. Приложение разработано с учетом офлайн-первого подхода, что означает возможность использования основных функций без подключения к интернету.

1.2 Ключевые особенности

Мультиплатформенность представляет собой одну из ключевых особенностей PlantCare. Приложение доступно как для мобильных устройств на базе Android, так и для настольных компьютеров под управлением Windows. Это обеспечивает гибкость в использовании и синхронизацию данных между устройствами.

AI-ассистент с каскадной архитектурой представляет инновационное решение для обработки запросов пользователей. Система последовательно использует несколько AI-моделей, автоматически переключаясь между ними при недоступности или ошибках. Это гарантирует высокую надежность сервиса.

Офлайн-режим обеспечивает функционирование ключевых функций без интернет-соединения. Локальная база растений, справочник заболеваний и LocalRagEngine работают полностью автономно, что особенно важно в условиях нестабильного подключения.

1.3 Целевая аудитория

Приложение ориентировано на широкий круг пользователей:

- Владельцы комнатных растений, нуждающиеся в систематизации ухода
 - Садоводы-любители, выращивающие растения на приусадебном участке
 - Ботанические энтузиасты, интересующиеся классификацией и особенностями растений
 - Начинающие цветоводы, которым требуются базовые знания по уходу
 - Профессиональные озеленители, ведущие учет большого количества растений
-

2. Возможности

2.1 Мои растения

Раздел "Мои растения" представляет собой центральный компонент приложения для управления личной коллекцией растений. Функционал позволяет добавлять новые растения с указанием названия, типа и дополнительного описания. Каждое растение может быть привязано к определенной категории для удобства фильтрации.

Система событий ухода позволяет планировать и отслеживать регулярные мероприятия по заботе о растениях. Доступны четыре типа событий: полив, подкормка, опрыскивание и пересадка. Для каждого события можно установить периодичность выполнения и добавить примечание.

Отметка выполненных событий формирует историю ухода за каждым растением. Эта информация используется для анализа регулярности процедур и построения рекомендаций. Автоматическая система может подсказывать оптимальное время для следующего полива или подкормки на основе исторических данных.

Добавление растения осуществляется через диалоговое окно с полями для ввода названия, выбора типа из выпадающего списка и текстового поля для описания. При редактировании доступны все поля, включая возможность изменить привязанные события ухода.

Удаление растения требует подтверждения для предотвращения случайной потери данных. При удалении также удаляются все связанные события и заметки.

2.2 Заметки

Система заметок позволяет создавать текстовые записи с привязкой к конкретным растениям или без нее. Каждая заметка автоматически получает временную метку создания и может быть отредактирована позже.

Заметки без привязки к растениям хранятся в общем разделе и могут использоваться для общих записей, напоминаний или идей по озеленению.

Добавление заметки осуществляется через диалоговое окно с текстовым полем. Привязка к растению осуществляется через выпадающий список или может быть пропущена.

Редактирование заметки доступно из списка, при этом автоматически обновляется временная метка изменения.

Удаление заметки требует подтверждения аналогично удалению растений.

2.3 Chat Sessions

Функция сохранения истории чатов представляет собой одно из значимых обновлений версии 1.2. Система включает сущности базы данных для хранения сессий и сообщений.

ChatSession хранит информацию о каждой сессии: уникальный идентификатор, название (можно изменить), временные метки создания и последнего обновления. Максимальное количество сохраненных сессий ограничено 100 для оптимизации производительности.

ChatMessageEntity хранит отдельные сообщения в рамках сессии: идентификатор сессии-родителя, роль (user/assistant), содержимое сообщения, временная метка. История сообщений загружается постранично для обработки длинных диалогов.

Интерфейс пользователя предоставляет кнопки "Новый чат" и "История". Первая создает новую сессию с очисткой контекста, вторая открывает список прошлых сессий для выбора и продолжения.

Переименование сессий доступно из списка истории через долгое нажатие или контекстное меню. По умолчанию используется формат "Чат #N" с датой создания.

2.4 Справочник растений

Справочник растений объединяет несколько источников информации для обеспечения полными данными по уходу за растениями.

Локальная база содержит 10+ наиболее распространенных комнатных растений с базовой информацией по уходу. База работает полностью оффлайн и загружается вместе с приложением.

Perenual API предоставляет доступ к базе из более чем 300 000 растений. API бесплатный, не требует ключа для базового использования. Запросы осуществляются по названию растения с получением детальной информации о поливе, освещении, температурном диапазоне.

Wikimedia Commons используется для получения эталонных изображений растений. Система извлекает ключевые слова (KEYWORDS) из ответов AI и запрашивает соответствующие изображения. Лимит — 10 изображений на запрос.

LocalRagEngine представляет локальный поисковый движок для работы без интернета. Использует векторное представление текста для семантического поиска по встроенной базе знаний.

2.5 Диагностика болезней

Система диагностики заболеваний растений объединяет экспертные знания и возможности AI для точной постановки диагноза.

Пользователь выбирает симптомы из predetermined списка, содержащего 13 основных симптомов: желтеющие листья, коричневые пятна, мучнистый налет, вялость, опадающие листья, паутина, видимые насекомые, гниль, черные точки, деформация листьев, белые пятна, липкие выделения, дырочки на листьях.

Можно выбрать один или несколько симптомов для более точного анализа. Система учитывает комбинации при поиске заболеваний.

AI-диагностика использует каскадный подход с последовательным обращением к моделям. При недоступности AI LocalRagEngine выполняет поиск по локальной базе заболеваний.

Результат включает список возможных заболеваний с вероятностью, рекомендации по лечению и профилактические меры для предотвращения повторного заражения.

2.6 AI-ассистент (Cascade)

AI-ассистент представляет собой основной интерфейс взаимодействия пользователя с системой искусственного интеллекта. Каскадная архитектура обеспечивает надежность и доступность сервиса.

Модели в каскаде используются в строгом порядке приоритета:

Первая модель — NVIDIA Nemotron через провайдер OpenRouter. Основные характеристики: поддержка vision, бесплатный tier, высокая скорость отклика, приоритет 1.

Вторая модель — MiniMax также через OpenRouter. Обеспечивает vision поддержку как backup. Приоритет 2.

Третья модель — GLM 4.5 Air. 💎💎 бесплатная модель с vision возможностями. Приоритет 3.

Четвертая модель — Gemini 2.0 Flash через OpenRouter Free. Резервная опция. Приоритет 4.

Пятая опция — LocalRagEngine для оффлайн режима. Работает без интернета, использует встроенную базу знаний. Приоритет 5.

Системный промпт настраивает поведение модели: "Ты — опытный ассистент по уходу за растениями. Отвечай на русском языке, кратко и по существу. Используй эмодзи для иллюстрации. Структурируй ответ с заголовками."

Извлечение KEYWORDS происходит автоматически. Модель добавляет строку формата "KEYWORDS: ключевое_слово1, ключевое_слово2" для последующего поиска изображений.

Status callback обеспечивает визуальную обратную связь. Пользователь видит текущую модель в процессе обработки: "NVIDIA Nemotron...", "MiniMax...", "GLM...", "OpenRouter...", "Локальная база..." .

2.7 Neural Screen (ИИ-анализатор)

Neural Screen предоставляет функцию распознавания растений по фотографиям с использованием computer vision технологий.

Загрузка изображения возможна двумя способами: непосредственно через камеру устройства или выбором из галереи фотографий. Поддерживаются форматы JPEG, PNG.

Анализ изображения осуществляется через AI каскад с vision-поддержкой. Модели способны идентифицировать вид растения, оценивать общее состояние, выявлять визуальные проблемы (пожелтения, повреждения, паразиты).

При недоступности AI моделей активируется TFLite fallback — локальная модель машинного обучения. Возможности ограничены базовым анализом, но работает полностью оффлайн.

Результат анализа включает: предполагаемый вид растения, оценку здоровья (в процентах), выявленные проблемы, персонализированные рекомендации по уходу.

Сохранение результатов происходит автоматически. История анализов доступна для просмотра в соответствующем разделе.

2.8 Погода

Раздел погоды использует открытый API Open-Meteo для получения актуальных метеоданных без необходимости регистрации и получения ключей.

Автоопределение геолокации осуществляется через сторонние сервисы. Используется IP-адрес пользователя для определения примерных координат. При недоступности — ручной ввод координат.

Отображаемые данные включают: текущую температуру (Цельсии/Фаренгейт), влажность воздуха в процентах, атмосферное давление в гектопаскалях, скорость и направление ветра в метрах в секунду, УФ-индекс.

Weather code конвертируется в человекочитаемое описание: "Ясно", "Облачно", "Пасмурно", "Дождь", "Снег", "Гроза" и другие варианты.

Рекомендации для растений формируются автоматически на основе текущих условий. Учитывается влажность для планирования полива, температура для определения возможности выноса растений на улицу.

2.9 Тёмная тема

Поддержка тёмной темы реализована с использованием Material Design 3 и его возможностей адаптивных цветов.

Переключение между темами осуществляется одной кнопкой в интерфейсе. Анимация перехода плавная, длительностью 300 миллисекунд.

Адаптивные цвета автоматически подстраиваются под выбранную тему. Светлая тема использует светлые оттенки зеленого, тёмная — насыщенные темно-зеленые тона.

Сохранение выбора пользователя происходит в локальных настройках. При следующем запуске автоматически применяется сохраненная тема.

2.10 Современный UI/UX

Пользовательский интерфейс построен на принципах Material Design 3 с расширенными возможностями визуальной обратной связи.

Основные компоненты включают FilledTonalButton для важных действий (особенно для загрузки фото), OutlinedButton для второстепенных действий, FloatingActionButton для главного действия на экране, Cards для отображения контента, Dialogs для подтверждений и ввода данных, Bottom Sheets для дополнительных опций.

Анимации реализованы для различных сценариев: fade-in при загрузке контента, slide-in для элементов списков, scale-in для появления карточек, плавные переходы между экранами.

Prox Indicator визуально отображает статус подключения: зеленый (онлайн), желтый (ограничено), красный (офлайн). Располагается в верхней части экрана.

Сетка главного экрана организована как 2×4, обеспечивая быстрый доступ ко всем основным функциям с одного экрана.

Крупные кнопки размером 64dp обеспечивают удобство касания и доступность для пользователей с ограниченными возможностями.

Высокий контраст элементов гарантирует читаемость при различных условиях освещения и для пользователей с нарушениями зрения.

3. Архитектура AI (Cascade)

3.1 Общая схема каскада

Архитектура каскадного AI представляет собой многоуровневую систему отказоустойчивости, обеспечивающую доступность сервиса при различных условиях.

Схема потока запроса:

Старт → NVIDIA Nemotron (основная модель) |— Успех → Вернуть ответ |— Ошибка/таймаут
→ MiniMax (fallback 1) |— Успех → Вернуть ответ |— Ошибка/таймаут → GLM 4.5 Air (fallback 2) |—
Успех → Вернуть ответ |— Ошибка/таймаут → OpenRouter Free (fallback 3) |— Успех → Вернуть ответ
|— Ошибка/таймаут → LocalRagEngine (fallback 4, оффлайн) |— Вернуть ответ из локальной базы

3.2 Временные параметры

Таймаут каждой модели составляет 15 секунд. При отсутствии ответа в течение этого времени происходит переключение на следующую модель.

Задержка между этапами составляет 500-1000 миллисекунд. Это необходимо для визуализации процесса пользователю через status callback.

Максимальное время обработки при последовательном использовании всех моделей может достигать 75 секунд (5 × 15 секунд).

3.3 Для изображений (vision)

Обработка изображений имеет приоритетную схему использования моделей с поддержкой vision:

Старт → NVIDIA Nemotron (vision) |— Успех → Вернуть анализ |— Ошибка → MiniMax (vision) |—
Успех → Вернуть анализ |— Ошибка → TFLite fallback (NeuralScreen) |— Вернуть базовый анализ

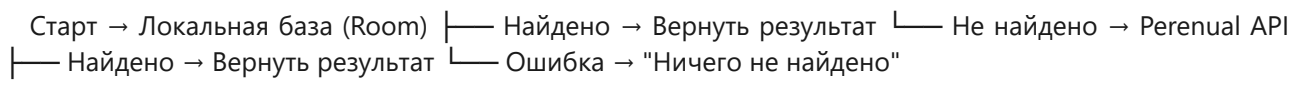
3.4 Status callback flow

Механизм обратной связи реализован через интерфейс callback:

UI регистрирует функцию через `setStatusCallback(функция)` Каскад вызывает функцию при переключении между моделями Функция обновляет состояние в UI через 500-1000ms Задержка обеспечивает визуализацию каждого этапа

3.5 Справочник растений

Поиск в справочнике также использует каскадный подход:



3.6 Эталонные изображения

流程 получения изображений:

AI диагностирует → Извлекает KEYWORDS из ответа Wikimedia Commons API → Поиск по ключевым словам Fallback: Pixabay API при ошибке Wikimedia

4. Технологический стек

4.1 Android (app/)

| Компонент | Версия | Назначение | |-----|-----|-----|-----| | Kotlin | 1.9.22 | Язык
разработки | | Jetpack Compose | 1.6.x | UI framework | | Material Design 3 | Да | Дизайн-система | | MVVM |
ViewModel + Room | Архитектура | | Min SDK | 24 | Android 7.0 | | Target SDK | 34 | Android 14 | | Room |
SQLite | База данных | | Navigation Compose | Да | Навигация | | OkHttp | 4.12.0 | HTTP клиент | | Kotlinx
Serialization | Да | JSON | | Coil Compose | Да | Изображения | | WorkManager | Да | Фоновые задачи |

4.3 Core (core/)

- Kotlin Multiplatform (JVM + Android)
- LocalRagEngine (векторный поиск)
- Векторная база (встроено)

4.4 Shared-UI (shared-ui/)

- Kotlin Multiplatform Compose
- Единая дизайн-система
- SharedChatAssistantScreen
- DesignTokens

4.5 API-интеграции

| API | Назначение | Бесплатно | |-----|-----|-----| | OpenRouter | AI cascade | Да | | Local RAG |
Офлайн | Да | | Perenual API | Справочник | Да | | Wikimedia | Изображения | Да | | Open-Meteo | Погода
| Да |

5. Структура проекта

5.1 Основные директории

app/ — Основное Android приложение core/ — Kotlin Multiplatform код desktop/ — Desktop приложение shared-ui/ — Общие UI компоненты server/ — Опциональный TFLite сервер

5.2 app/data/ — Сущности базы данных

Plant.kt — Растение (id, name, type, description, createdAt) CareEvent.kt — Событие ухода (id, plantId, type, description, isDone, doneAt) Disease.kt — Заболевание Pest.kt — Вредитель ChatSession.kt — Сессия чата ChatMessageEntity.kt — Сообщение чата

5.3 app/db/ — Работа с базой данных

AppDatabase.kt — Главная база данных PlantDao.kt — Операции с рас◆◆ен◆◆ями CareEventDao.kt
— Операции с событиями ChatDao.kt — Операции с чатами Converters.kt — Конвертеры типов

5.4 app/viewmodel/

PlantCareViewModel.kt — Главная ViewModel

5.5 app/ui/ — Экраны приложения

HomeScreen.kt — Главный экран PlantsScreen.kt — Список растений PlantDetailScreen.kt — Детали растения NotesScreen.kt — Заметки ReferenceScreen.kt — Справочник SymptomDiagnosisScreen.kt — Диагностика ChatGPTAssistantScreen.kt — AI чат NeuralScreen.kt — ИИ анализатор WeatherScreen.kt — Погода SettingsScreen.kt — Настройки

5.6 app/ai/ — AI клиенты

AIClient.kt — Интерфейс клиента CascadeAiClient.kt — Каскадный клиент AiClientProvider.kt — Провайдер OpenRouterAiClient.kt — OpenRouter клиент MiniMaxAiClient.kt — MiniMax клиент

5.7 app/util/ — Утилиты

[WeatherApi.kt](#) — Погода [PerenualApi.kt](#) — Справочник [WikipediaApi.kt](#) — [Wikipedia](#) [WikimediaApi.kt](#) — [Wikimedia](#) [PixabayApi.kt](#) — [Pixabay](#) [Prefs.kt](#) — Настройки [CareEventReminderWorker.kt](#) — Напоминания [ProxyStatusMonitor.kt](#) — Мониторинг статуса

6. Установка и сборка

6.1 Android APK

```
bash ./gradlew :app:assembleDebug
```

APK располагается в: `app/build/outputs/apk/debug/app-debug.apk` Размер: приблизительно 150 MB

6.2 Desktop (Windows)

```
bash ./gradlew :desktop:assemble # Сборка ./gradlew :desktop:run # Запуск
```

MSI: desktop/build/compose/binaries/main-release/msi/ EXE: desktop/build/compose/binaries/main-release/exe/ Размер: приблизительно 200 MB

6.3 Требования для сборки

- JDK 17+
 - Android SDK 34+
 - Android Studio Giraffe или новее
 - Gradle 8.11.1
-

7. Использование

7.1 Первый запуск

1. Установите APK или запустите Desktop версию
2. AI требует интернет для работы каскада
3. При отсутствии интернета используется LocalRagEngine
4. Справочник работает оффлайн через локальную базу

7.2 Добавление растения

1. Мои растения → "+"
2. Введите название (обязательно)
3. Выберите тип (опционально)
4. Добавьте описание (опционально)
5. Нажмите сохранить

7.3 События ухода

1. Откройте растение
2. "Добавить событие"
3. Выберите тип: Полив/Подкормка/Опрыскивание/Пересадка
4. Установите периодичность
5. Добавьте описание (опционально)
6. Отметьте как выполненное после свершения

7.4 Chat Sessions

1. AI-ассистент → "Новый чат"
2. Введите вопрос
3. Получите ответ от AI
4. Статус модели отображается в реальном времени
5. "История" → просмотр прошлых чатов
6. Переименование — длительное нажатие

7.5 AI-ассистент

1. Введите текстовый вопрос
2. Прикрепите фото (опционально)
3. Ответ приходит от каскада AI
4. Статус модели виден во время обработки

7.6 Neural Screen

1. Выберите фото (камера/галерея)
2. Нажмите "Анализировать"
3. При недоступности AI — TFLite fallback
4. Получите результат

7.7 Диагностика

1. Выберите симптомы из списка
2. Нажмите "Анализировать"
3. AI + LocalRagEngine обрабатывают
4. Получите диагноз и рекомендации

7.8 Справочник

1. Введите название растения
 2. Результат из локальной базы
 3. При отсутствии — Renewal API
-

8. API-интеграции

8.1 OpenRouter API

Endpoint: <https://openrouter.ai/api/v1/chat/completions> Модели: NVIDIA Nemotron, MiniMax, GLM 4.5 Air, Gemini Бесплатно: Да (с лимитами) Регистрация: openrouter.ai

8.2 Perenual API

Endpoint: <https://perenual.com/api/species-list> Покрытие: 300 000+ растений Бесплатно: Да

8.3 Wikimedia Commons

Endpoint: commons.wikimedia.org/w/api.php Назначение: Изображения растений Бесплатно: Да

8.4 Open-Meteo

Endpoint: api.open-meteo.com/v1/forecast Бесплатно: Да, без ключа

9. Минимальные требования

9.1 Android

- Android 7.0 (API 24) и выше
- 2 ГБ оперативной памяти
- 150 МБ свободного места

9.2 Desktop (Windows)

- Windows 10 и выше
 - 4 ГБ оперативной памяти
 - 200 МБ свободного места
 - Java 17 (включена в дистрибутив)
-

10. История версий

10.1 v1.2 — Апрель 2026

Новое: - AI Cascade с 4-stage failover - NVIDIA Nemotron → MiniMax → GLM → OpenRouter Free - Wikimedia Commons для эталонных изображений - Chat Sessions с сохранением в БД - Status callback на всех AI-экранах - TFLite fallback для NeuralScreen - AI + LocalRagEngine для диагностики - FilledTonalButton для фото - Proxy Indicator компоненты - Извлечение KEYWORDS из AI ответов - Переименование чатов - Кнопки "Новый чат" / "История"

10.2 v1.1 — Апрель 2026

Новое: - Groq API как основной AI (Llama 3.1/3.2) - Google Gemini как fallback - Perenual API для справочника - Pixabay API для фотографий - FallbackAiClient цепочка - UI сетка 2x4 - Улучшенная диагностика с AI

10.3 v1.0 — Октябрь 2025

- Первый релиз
 - Room Database
 - Compose UI
 - Локальный справочник
 - LM Studio интеграция
 - TFLite модель
 - Диагностика болезней
 - Погода (Open-Meteo)
 - Тёмная тема
-

11. Устранение неполадок

11.1 AI не работает

Проверьте интернет-соединение. Подождите 10-20 секунд для таймаута. Каскад автоматически переключится на LocalRagEngine. Попробуйте другой запрос.

11.2 Нет изображений

Проверьте ключевые слова в запросе. Попробуйте изменить формулировку. Используйте более распространенные названия.

11.3 Ошибки загрузки APK

Разрешите установку из неизвестных источников в настройках. Очистите кэш Play Store. Перезагрузите устройство.

11.4 Приложение вылетает

Очистите кэш приложения. Переустановите приложение. Проверьте совместимость версии Android. Проверьте свободное место.

11.5 Нет событий ухода

Проверьте привязку событий к растению. Убедитесь в правильности типа события. Перезагрузите приложение.

11.6 Погода не загружается

Проверьте интернет. Очистите кэш. Перезагрузите приложение.

12. Лицензия и контакты

12.1 Создатель

Денис Аниськов

12.2 Контакты

GitHub: <https://github.com/DenisAniskov/PlantCare> Email: denis.aniskov55@gmail.com

12.3 Лицензия

MIT License — свободное использование для некоммерческих целей.

Документация обновлена: Апрель 2026