«Айфэллоу Трекер»

Функциональные характеристики

2026 год

Содержание

Введение

Айфэллоу Трекер – это десктопное приложение для Windows, предназначенное для индивидуального управления задачами и проектами. Продукт позволяет создавать и структурировать задачи, декомпозировать их на подзадачи, прикреплять файлы, визуализировать рабочий процесс с помощью Канбан - доски. Приложение работает полностью в офлайн - режиме, а все задачи, файлы и настройки хранятся локально в базе данных.

Назначение ПО

Айфэллоу Трекер предназначен для:

1. Повышения эффективности работы сотрудников, путем приоритезации задач и организации рабочего процесса;
2. Визуализация бэклога для оценки текущей загруженности;
3. Структурирования задач по целям, рискам, срокам и важности выполнения;
4. Упрощению доступа к документам и файлам любого типа, связанным с задачей.

Продукт может использоваться в компаниях любых отраслей и видов деятельности, которым требуется надежный, быстрый и независимый инструмент для планирования.

Функциональные возможности ПО

Основные функции Айфэллоу Трекер:

Функция	Описание
Управление задачами	Создание редактирование и удаление задач и подзадач. Назначение сроков, приоритетов, рисков. Отслеживание и назначение прогресса выполнения задач (0 - 100%)
Управление подзадачами	Декомпозиция задач на подзадачи с собственным прогрессом с шагом 25% выполнения. Автоматический расчет прогресса родительской задачи по среднему из прогрессов подзадач

Функция	Описание
Управление файлами	Прикрепление к задачам файлов любого формата. Открытие файлов из папки задачи, удаление и хранение в структуре приложения при условии расположения файла на вашем локальном компьютере
Канбан-доска для задач	Визуализация группы задач на Канбан – доске по приоритетам: Критический, Срочный, Средний и Не срочный
Автоматическая оценка проиритета	Автоматический расчет приоритетов на основе четырех шкал с цветовой индикацией: Важность, Срочность, Сложность, Срыв сроков

Входные и выходные данные

Входные данные пользователя:

1. Данные, вводимые в формы (название, описание, сроки, цели, параметры шкал);

2. Файлы, добавляемые пользователем из файловой системы;

3. Действия по изменению прогресса подзадач.

Выходные данные:

1. Отображение списков задач на Канбан - доске с актуальным статусом, сроками и прогрессом;
2. Окна редактирования задач и подзадач;
3. Визуальные изменения Канбан - доски при редактировании процесса и удалении задач.

Архитектура приложения

Продукт является однопользовательским и работает полностью на локальном ПК. Все функции работают в полном объеме без подключения к сети Интернет.

Продукт представляет собой настольное приложение на базе Windows Presentation Foundation (WPF). Реализовано на C# с использованием .NET Framework 10.

Перечень компонентов и сервисов

Таблица 1 – Компоненты приложения

Компонент	Модуль / Тип	Описание	Основная функциональность
WPF UI	Настольное приложение (WPF)	Графический интерфейс пользователя	Отображение Канбан-доски (4 колонки по приоритету), окон создания/редактирования задач, списка подзадач, файлов и раздела настроек. Боковая навигация между страницами «Задачи» и «Настройки»
ViewModels	Библиотека классов (.NET)	Модели представления	Подготовка данных для UI: фильтрация задач по колонкам приоритета, сортировка по времени создания, расчёт прогресса (в том числе автоматический для родительских задач), управление состоянием выбранной страницы, реализация команд (создание, редактирование, удаление, изменение прогресса подзадач)
Сервисы	Классы в составе приложения	Слой бизнес-логики	- IAppSettingsService - загрузка/сохранение настроек пользователя в JSON-файл; - IThemeService - применение светлой или тёмной темы через MaterialDesign; - IAppVersionService - чтение версии приложения из version.json; - FileManager - операции с

Компонент	Модуль / Тип	Описание	Основная функциональность
			файлами, прикреплёнными к задачам (копирование, удаление, открытие)
Data Layer (Database)	Библиотека классов (.NET)	Слой доступа к данным	Определение моделей данных, настройка связей и ограничений (каскадное удаление, обязательные поля, максимальная длина). Предоставляет CRUD-операции через запросы
DatabaseInitializer (Database)	Класс в составе Database	Инициализация базы данных при первом запуске	Создание файла в папке \%LocalAppData%\, применение схемы через , заполнение тестовыми данными при первом создании
AppDataPathProvider (Database)	Класс в составе Database	Определение путей к системным папкам и файлам приложения	Возвращает пути к каталогу данных, к файлу БД, к файлу настроек, к папке для файлов задач
Installer	Отдельное WPF-приложение	Установщик приложения	Запускается пользователем вручную. Проверяет наличие предыдущей версии, копирует файлы основного приложения в целевой каталог, обновляет БД при необходимости. Интерфейс с кнопкой «Установить / Обновить» и индикацией прогресса. Использует CommunityToolkit.Mvvm для команд

Схема взаимодействия перечисленных компонентов приложения:

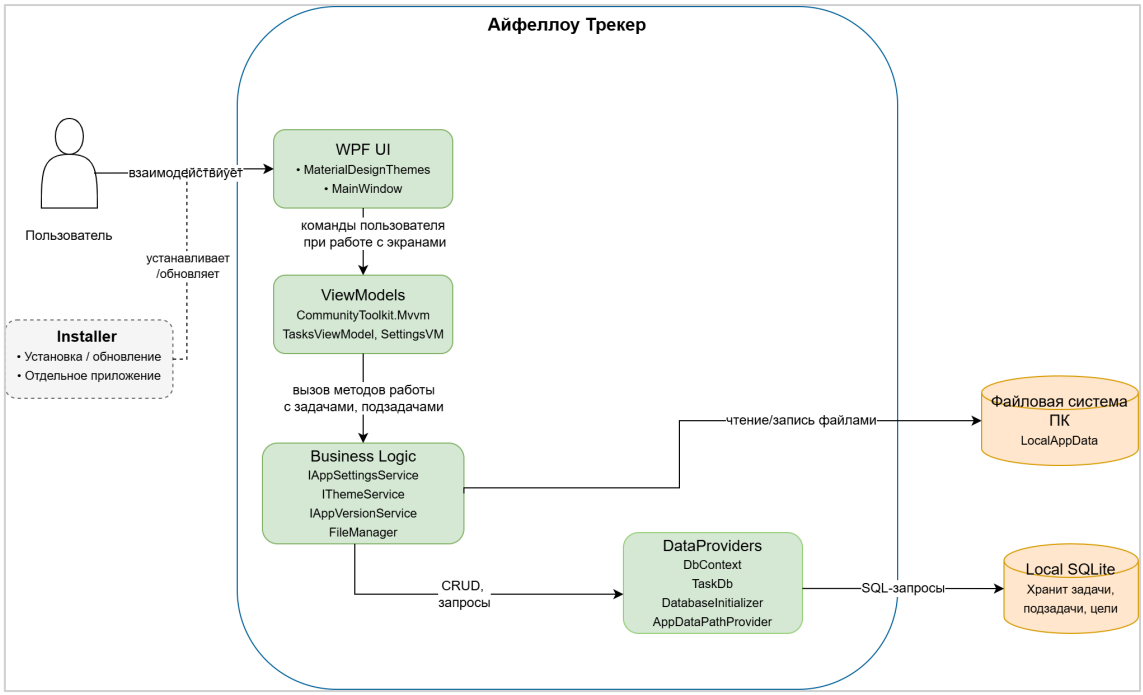

Рисунок 1 – Схема взаимодействия компонентов приложения

Цвет	Обозначение	Компоненты
Зелёный	Внутренние компоненты приложения	WPF UI, ViewModels, Business Logic, DataProviders
Оранжевый	Хранилища данных	SQLite, Файловая система ПК

Сервисы и интеграции

Таблица 2 – Сервисы приложения

Сервис	Функционал	Компонент	Интеграция
SQLite	Хранение всех данных пользователя (задачи, подзадачи, цели)	Database (Entity Framework Core)	SQLitePCLRaw
Файловая система	Хранение пользовательских файлов, прикреплённых к задачам, а также файла настроек	FileManager, SettingsService	System.IO
MaterialDesign	Оформление интерфейса (светлая/тёмная тема, стилизованные элементы управления)	ThemeService	MaterialDesignThemes / MaterialDesignColors

Сервис	Функционал	Компонент	Интеграция
CommunityToolkit.Mvvm	Реализация паттерна MVVM (команды, наблюдаемые коллекции, уведомления об изменениях)	ViewModels (TasksViewModel, SettingsViewModel)	CommunityToolkit.Mvvm

Примечания:

- **SQLite** используется через *Entity Framework Core*, который, в свою очередь, опирается на нативную библиотеку SQLite (поставляемую через *SQLitePCLRaw*). Сама SQLite распространяется как *Public Domain*.
- **Файловая система** используется для хранения:
  - файлов задач;
  - настроек приложения.
- **MaterialDesign** – библиотека визуальных тем, которая управляет переключением между светлой и тёмной темой.
- **CommunityToolkit.Mvvm** – библиотека, обеспечивающая удобную реализацию MVVM.

Требования к аппаратному и программному обеспечению

Таблица 3 – Минимальные системный требования

Параметр	Значение
Операционная система	Windows 10 версии 1809 (сборка 17763) или выше, Windows 11
Процессор	1.5 ГГц или выше, архитектура x64 или x86
Оперативная память	4 ГБ
Свободное место на диске	250 МБ (для приложения, базы данных и файлов настроек)
.NET Runtime	.NET 10.0 Desktop Runtime (если приложение не опубликовано как самодостаточное)

Таблица 4 – Рекомендуемые системный требования

Параметр	Значение
Операционная система	Windows 11 или Windows 10 с последними обновлениями
Процессор	2.0 ГГц или выше (многоядерный)
Оперативная память	8 ГБ
Свободное место на диске	500 МБ и более (с учётом хранения пользовательских файлов задач)

Таблица 5 – Требования к операционной системе и стороннему ПО

Требование	Описание
Операционная система	Требуется Windows 10 либо Windows 11. Поддержка .NET 10 и WPF гарантируется только на этих версиях.
.NET	Приложение разработано на .NET 10. Если распространяется в виде зависимого от платформы – требуется предварительная установка .NET 10 Desktop Runtime. При самодостаточной публикации все необходимые компоненты .NET входят в состав приложения, и отдельная установка не нужна.

Требование	Описание
Графическая подсистема	DirectX 9 (или выше) – входит в состав Windows и используется WPF для аппаратного ускорения.
Разрешения файловой системы	Приложение не требует прав администратора. Все данные сохраняются в папку файловой системы пользователя, которая создаётся автоматически.