#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **Техническое задание “To-Do List Bot”**

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

Выполнили: Маркин Даниил,

Кузин Сергей, Иванова Альмира

#### 

#### 

#### **1. Общие сведения**

#### **Наименование**: To-Do List Manager **Тип**: Telegram-бот **Назначение**: Управление персональными задачами с возможностью установки напоминаний, изменения статусов и совместного использования списков.

#### **2. Функциональные требования**

#### **2.1.** **Основные функции**:

#### **Создание и управление задачами:**

#### Добавление новых задач (➕ Добавить задачу).

#### Редактирование текста существующих задач (✏ Редактировать).

#### Удаление задач (❌ Удалить задачу).

#### Изменение статуса задач ("Не начата", "В процессе", "Выполнена").

#### **Напоминания:**

#### Установка времени напоминания в формате DD-MM-YYYY HH:MM.

#### Автоматическая отправка уведомлений при наступлении времени.

#### **Просмотр задач:**

#### Список активных задач (📋 Показать задачи).

#### Список выполненных задач (✅ Выполненные задачи).

#### 2.2. **Команды бота**:

#### /start – инициализация списка задач для нового пользователя.

#### Главное меню с кнопками:

#### Добавить/показать/удалить задачи.

#### Просмотр выполненных задач.

#### **3. Технические требования**

#### 3.1. **Используемые технологии**:

#### Язык программирования: **Python 3.11+**.

#### Библиотеки:

#### aiogram (для работы с Telegram API).

#### uuid (генерация уникальных ID списков задач).

#### asyncio (асинхронные операции, включая проверку напоминаний).

#### Хранение данных: локальный файл tasks.json (формат JSON).

#### 3.2. **Архитектура**:

#### **Состояния (FSM)**:

#### adding\_task, editing\_task, deleting\_task, setting\_reminder, changing\_status.

#### **Клавиатуры**:

#### Главное меню (ReplyKeyboardMarkup).

#### Интерактивные кнопки для задач (InlineKeyboardMarkup).

#### **Данные**:

#### to\_do\_lists: словарь с задачами и пользователями.

#### user\_to\_list: сопоставление пользователей с их списками.

#### 3.3. **Ограничения**:

#### Локальное хранение данных (не подходит для масштабирования).

#### Отсутствие мультиязычности.

#### **4. Интерфейс**

#### 4.1. **Главное меню**:

#### Кнопки:

#### Добавить задачу, Показать задачи, Выполненные задачи, Удалить задачу. 4.2. **Интерактивные элементы**:

#### Для каждой задачи:

#### Редактировать, Установить напоминание, Изменить статус, Завершить.

#### Выбор статуса через InlineKeyboardMarkup.

#### **5. Безопасность**

#### Валидация ввода времени напоминания.

#### Ограниченный доступ к данным (только для владельца списка).

#### **6. Тестирование**

#### 6.1. **Сценарии**:

#### Добавление/удаление задачи.

#### Установка напоминания и проверка его срабатывания.

#### Изменение статуса задачи. 7.2. **Ожидаемые результаты**:

#### Корректное сохранение данных в tasks.json.

#### Отправка уведомлений при наступлении времени.

#### **7. Дальнейшее развитие**

#### Добавление базы данных (SQLite/PostgreSQL).

#### Поддержка категорий задач.

#### Возможность совместного редактирования списков.

#### **Приложения**:

#### Исходный код: main.py.

#### Пример данных: tasks.json.

#### **ИТОГ**: Бот предоставляет удобный инструмент для управления задачами с функцией напоминаний и возможностью совместного использования списков. Реализован на Python с использованием библиотеки aiogram.

#### 

#### 