

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На создание инструментария для no-code разработки Telegram- ботов на Python

1. Основание для разработки

Разработка системы осуществляется в рамках выполнения выпускной квалификационной работы студента 4-го курса ИВТ 2.2 Киняева Ильи Константиновича на тему «Создание инструментария для no-code разработки Telegram-ботов на Python».

2. Цель и назначение разработки

Цель: Создание доступной и интуитивно понятной платформы для разработки Telegram-ботов без необходимости написания кода, что позволит расширить круг создателей ботов за счет пользователей без технического образования.

Назначение: Веб-инструментарий предназначен для визуального проектирования логики Telegram-бота, настройки команд, ответов, интеграций и автоматического генерации готового кода на Python для последующего развертывания.

3. Технические требования к системе

3.1. Требования к функциональным характеристикам

Система должна обеспечивать полный цикл разработки Telegram-бота: от проектирования до развертывания. Основные функции:

1. Визуальный редактор логики:

- Drag-and-drop интерфейс для создания сценариев диалога (диалоговых деревьев);
- Добавление и настройка триггеров (команды, текст, кнопки, inline-запросы);

- Настройка ответов бота (текст, изображения, документы, клавиатуры);
- Добавление условий и ветвлений в логику бота.

2. Генерация кода:

- Автоматическая генерация Python-кода на основе созданной логики с использованием библиотеки python-telegram-bot или aiogram;
- Возможность просмотра и ручного редактирования генерированного кода (опционально).

3. Управление проектами:

- Создание, сохранение и загрузка проектов ботов;
- Экспорт проекта в виде Python-файлов и зависимостей (requirements.txt).

4. Тестирование и отладка:

- Встроенный симулятор для тестирования логики бота внутри системы;
- Логирование действий и ошибок.

5. Развёртывание (Deployment):

- Интеграция с облачными платформами (например, Heroku, PythonAnywhere) для автоматического деплоя;
- Генерация инструкций для ручного развертывания на собственном сервере.

6. Управление пользователями:

- Регистрация и авторизация пользователей;
- Хранение проектов в личном кабинете.

7. Адаптивный интерфейс:

- Интерфейс должен корректно отображаться на ПК, планшетах и смартфонах.

3.2. Требования к надежности

Система должна обеспечивать сохранность проектов пользователей: автоматическое периодическое сохранение проектов, регулярное резервное копирование базы данных не реже одного раза в сутки. При возникновении ошибок ввода или логических конфликтов в конструкторе система должна выводить понятные сообщения без прекращения работы.

3.3. Условия эксплуатации

Серверная часть должна работать под управлением ОС Linux или Windows с интерпретатором Python версии 3.9 и выше. Для реализации логики приложения должен использоваться фреймворк FastAPI или Flask, для взаимодействия с базой данных — SQLAlchemy или Pony ORM. Клиентская часть должна быть доступна через современные браузеры (Chrome, Firefox, Safari).

3.4. Требования к аппаратному обеспечению

Для работы с системой пользователю необходимо устройство с доступом в Интернет и современным браузером. Серверная часть должна быть развернута на машине с минимум 2 ГБ ОЗУ, 2 ядрами CPU и 10 ГБ дискового пространства.

3.5. Требования к информационной и программной совместимости

Система должна быть реализована с использованием стека: Python, FastAPI/Flask, SQLAlchemy/Pony ORM. На этапе разработки допускается использование SQLite, для продакшена — PostgreSQL. Архитектура — монолитная с модульной структурой. Должна быть обеспечена совместимость с Telegram Bot API через библиотеку python-telegram-bot или аналоги.

3.6. Требования к программной документации

В состав документации должны входить: полный исходный код проекта, пояснительная записка к дипломному проекту, руководства пользователя по работе с конструктором, техническое описание архитектуры и алгоритмов генерации кода.

4. Технико-экономические показатели

Внедрение описанной системы позволит сократить время разработки Telegram-ботов, снизить порог входа для пользователей без навыков и опыта программирования, упростить процесс прототипирования, тестирования и модификации Telegram-ботов.

5. Стадии и этапы разработки

1. Анализ требований и исследование аналогов — 2 недели.
2. Проектирование архитектуры и БД — 3 недели.
3. Разработка front-end и backend части — 8 недель
4. Тестирование, отладка, написание документации — 3 недели.
5. Сдача проекта — 1 неделя.

6. Порядок контроля и приемки

Промежуточный контроль осуществляется по завершении каждого этапа. Приемка проекта осуществляется на основании полного соответствия требованиям ТЗ и наличия всей документации.