



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ  
Государственное бюджетное профессиональное образовательное учреждение города Москвы  
КОЛЛЕДЖ АРХИТЕКТУРЫ, ДИЗАЙНА И РЕИНЖИНИРИНГА № 26 (ГБПОУ «26 КАДР»)  
ул. Цимлянская, д. 7, стр. 1, Москва, 109559  
тел.: +7 (495) 679-47-21, тел.факс: +7 (495) 710-21-03, e-mail: spo-26@edu.mos.ru  
ОГРН 1057723001731 ИНН/КПП 7723356160/772301001



ОТДЕЛЕНИЕ ВЫСОКИЕ ТЕХНОЛОГИИ И КРЕАТИВНАЯ ЦИФРОВИЗАЦИЯ  
СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ 10

Допустить к защите  
Заведующий отделения  
Селезнева А.В.  
2025 год

## КУРСОВАЯ РАБОТА

Разработка VK-бота по оказанию административной помощи гражданам г. Калуги.

Специальность 09.02.07 Информационные системы и программирование  
код, название

КР.09.02.07.1.33.10  
(шифр работы)

Выполнил  
студент:

\_\_\_\_\_ Купцова В.Е.  
(подпись) (дата) (Ф.И.О.)

Проверил  
преподаватель:

\_\_\_\_\_  
(подпись) (дата) (Ф.И.О.)

Оценка:

\_\_\_\_\_

Москва 2025 г.

## Оглавление

Глоссарий.....	3
Введение .....	6
Глава 1 «Проектирование ВК-бота по оказанию административной помощи и ВК бот».....	8
1.1 О веб разработке.....	8
1.2 Сравнение и отличие веб-сайтов от ботов .....	9
1.3 Сравнение Telegram и ВК -ботов .....	11
1.4 Анализ и выбор языка программирования для разработки чат-бота ВКонтakte .....	13
Глава 2. «Разработка ВК-бота по оказанию административной помощи» .....	17
2.1 Инструменты создания проекта.....	17
2.2 Требования к проекту .....	18
2.3 Поставленные задачи .....	19
2.4 Настройка сообщества в ВК. ....	19
2.5 Создание сервера и его настройка .....	20
2.6 Подключение к FileZilla Client .....	21
2.7 Подключение к Callback API ВК в ВК.....	22
2.8 Создание ВК бота, структура проекта .....	23
2.9 Подключение к базе данных PostgreSQL .....	25
2.10 Создание форм .....	27
2.11 Keyboard.....	28
2.12 Обработчики .....	30
2.13 Тестирование бота.....	32
Заключение.....	35
Список используемых источников.....	37

## Глоссарий

Чат-боты — это виртуальные помощники, часть которых выполняет примитивные действия (напоминания, бронирование, запись информации, автоответчик), а другая построена на основе искусственного интеллекта. Чат-бота создают, чтобы выстроить диалог с пользователем. Он похож - конструктор из сообщений и кнопок, имитирующий разговор между реальными людьми. Чат-боты могут отвечать на простой запрос или конструировать сложную беседу с высоким уровнем персонализации;

ReplyKeyboardMarkup — разновидность клавиатуры. По факту это шаблоны сообщений. К примеру, ваш бот задаёт пользователю вопрос и предлагает варианты ответа. Пользователь может самостоятельно напечатать ответ, либо нажать на готовую кнопку. Такая клавиатура показывается вместо основной и не привязана ни к какому сообщению. В кнопки такой клавиатуры нельзя заложить никакой информации, нельзя запрограммировать для неё подобный если пользователь нажимает кнопку с текстом «abc» отправить текст «qwerty» алгоритм, отправлено будет только то, что написано на кнопке;

InlineKeyboardMarkup — это уже настоящая кастомная клавиатура. С её помощью мы можем выполнять более сложные действия. Она привязывается к сообщению, с которым была отправлена. В кнопки можно заложить любой текст размером. Inline кнопки позволяют скрыть в себе внутреннюю tg-ссылку, ссылку на внешний ресурс, а также shorcat для inline запроса;

Shorcat(Шорткаты) — это готовые ответы на запрос пользователя в поисковой выдаче. Система создаёт короткий готовый ответ, чтобы пользователь не переходил по ссылкам и быстро получал всю нужную информацию на странице поисковика;

Сессия (сеанс) — это один сеанс пользователя. Когда пользователь взаимодействует с ботом, фиксируется начало сессии. Если с момента

последнего взаимодействия пользователя с ботом прошло 30 минут, сессия заканчивается;

Сцена в боте — это сценарий, который позволяет разграничивать различные функции бота. Например, сцена позволяет заблокировать исполнение бота в рамках вопроса до тех пор, пока пользователь не выйдет из сцены или не пройдёт валидацию вопроса;

Токен или ключ доступа — это случайно сгенерированная строка, состоящая из цифр, латинских букв и специальных символов. Токен возвращается в качестве результата авторизации пользователя;

API (Application Programming Interface) — это посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать. API упрощает создание кода, поскольку предоставляет набор готовых классов, функций или структур для работы с имеющимися данными;

API ВКонтакте — это интерфейс, который позволяет получать информацию из базы данных vk.com с помощью HTTP-запросов к специальному серверу. Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса;

Callback API во «ВКонтакте» — это инструмент для отслеживания активности пользователей в сообществе. С его помощью можно, например, реализовать бота для отправки мгновенных ответов на поступающие сообщения, систему автоматической модерации контента или сервис для сбора и обработки показателей вовлечённости аудитории. Для работы с Callback API вам потребуется собственный сервер. Во вкладке Настройки сервера в поле Адрес нужно указать URL, по которому ваш скрипт обрабатывает входящие HTTP-запросы с событиями Callback API. Для создания скрипта потребуются навыки программирования и знание;

POST-запросы в «ВКонтакте» — это метод для отправки API-запросов. В API «ВКонтакте» POST и GET равнозначны;

					КР.09.02.07.1.33.10	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

Метод в ВК— это условная команда, которая соответствует той или иной операции с базой данных — получение информации, запись или удаление;

Сервер — это устройство, которое хранит данные и даёт доступ к ним большому числу клиентов. Серверы обрабатывают запросы пользователей и могут, например, отправлять и принимать письма, обеспечивать доступ к сайтам и базам данных, идентифицировать пользователей;

JSON — это формат записи данных в виде пар <ИМЯ\_СВОЙСТВА>: <ЗНАЧЕНИЕ>;

Express.js — это минималистичный и гибкий веб-фреймворк для Node.js, который предоставляет множество возможностей для создания веб-приложений и API.

## Введение

В современном мире информационные технологии играют ключевую роль в упрощении взаимодействия между государственными органами и гражданами. Одним из наиболее эффективных инструментов для достижения этой цели являются чат-боты, которые позволяют автоматизировать процессы предоставления информации и услуг. В связи с этим разработка чат-бота ВКонтакте, направленного на оказание административной помощи жителям Калуги, становится актуальной задачей, способствующей улучшению качества обслуживания населения и повышению доступности государственных услуг.

Актуальность данной работы обусловлена необходимостью оптимизации взаимодействия граждан с местными органами власти, особенно в условиях растущей цифровизации и перехода на электронные формы обслуживания. «ВКонтакте», как одна из самых популярных социальных сетей в России, предоставляет уникальную платформу для реализации данного проекта, позволяя охватить широкую аудиторию и обеспечить удобный доступ к информации.

Целью курсовой работы является разработка функционального ВК-бота, который будет предоставлять жителям города Калуги информацию о доступных административных услугах, а также отвечать на часто задаваемые вопросы. **Для достижения этой цели необходимо решить ряд задач:**

1. Провести анализ существующих решений и определить потребности целевой аудитории;
2. Разработать архитектуру и функциональные требования к ВК-боту;
3. Реализовать бота с использованием современных технологий и инструментов;
4. Провести тестирование и оценку эффективности работы бота;

					КР.09.02.07.1.33.10	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

5. Подготовить рекомендации по дальнейшему развитию и улучшению функционала.

Объектом исследования является процесс разработки чат-бота для социальной сети ВКонтакте, а предметом исследования — функциональные возможности и особенности реализации административной помощи с помощью этого инструмента.

Таким образом, данная курсовая работа направлена на создание инновационного решения, способствующего улучшению взаимодействия между гражданами и органами власти, что, в свою очередь, будет способствовать повышению уровня информированности и удовлетворённости населения.

					<i>KP.09.02.07.1.33.10</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

# Глава 1 «Проектирование ВК-бота по оказанию административной помощи и ВК бот».

## 1.1 О веб разработке

Веб-разработка — это процесс создания и поддержания веб-сайтов и веб-приложений. Она включает в себя: дизайн, программирование, управление контентом и оптимизация для поисковых систем. Веб-разработка делится на две основные категории: фронтенд – видимый контент для пользователя и бэкенд -не видимый контент.

Фронтенд-разработка отвечает за то, как веб-сайт выглядит и взаимодействует с пользователем. Основные технологии, используемые во фронтенде, включают:

HTML (HyperText Markup Language): Язык разметки, который используется для создания структуры веб-страниц.

CSS (Cascading Style Sheets): Язык стилей, который позволяет задавать внешний вид элементов на веб-странице, включая цвета, шрифты и макеты.

JavaScript: Язык программирования, который добавляет интерактивность на веб-страницы, позволяя пользователям взаимодействовать с контентом.

Современные фронтенд-разработчики также используют различные фреймворки и библиотеки, такие как React, Angular и Vue.js, которые упрощают процесс разработки и делают код более структурированным и поддерживаемым.

Бэкенд-разработка включает в себя серверную часть веб-приложений, которая отвечает за обработку данных, взаимодействие с базами данных и выполнение бизнес-логики. Основные технологии и языки, используемые в бэкенде, включают:

Серверные языки программирования: PHP, Python, Ruby, Java, Node.js и другие.

					КР.09.02.07.1.33.10	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



Системы управления базами данных (СУБД): MySQL, PostgreSQL, MongoDB и другие, которые используются для хранения и управления данными.

API (Application Programming Interface): Интерфейсы, которые позволяют различным программным компонентам взаимодействовать друг с другом.

Бэкенд-разработчики также занимаются настройкой серверов, обеспечением безопасности и оптимизацией производительности веб-приложений.

## 1.2 Сравнение и отличие веб-сайтов от ботов

Веб-сайты и боты представляют собой два различных подхода к взаимодействию с пользователями в цифровом пространстве, и их отличия можно рассмотреть с нескольких ключевых аспектов.

Во-первых, определение. Веб-сайт — это набор связанных веб-страниц, доступных через интернет, которые могут содержать текст, изображения, видео и другие мультимедийные элементы. Веб-сайты могут быть статическими или динамическими, предоставляя пользователям информацию, услуги или продукты. В отличие от этого, бот — это программа, которая автоматически выполняет задачи в интернете, чаще всего взаимодействуя с пользователями через текстовые или голосовые интерфейсы. Чат-боты, предназначены для общения с пользователями и могут работать на различных платформах, включая мессенджеры и социальные сети;

Во-вторых, цель. Основная цель веб-сайта заключается в предоставлении информации и услуг пользователям. Веб-сайты могут использоваться для различных задач, таких как ведение блогов, электронная коммерция или новостные ресурсы. Боты, в свою очередь, нацелены на автоматизацию взаимодействия с пользователями и выполнение

определенных задач, таких как предоставление информации, помощь в навигации или обработка заказов;

Третьим важным аспектом является взаимодействие с пользователем. Веб-сайты предлагают графический интерфейс, где пользователи могут кликать на ссылки, заполнять формы и просматривать контент. Пользовательский опыт зависит от дизайна и функциональности сайта. Боты же взаимодействуют с пользователями через текстовые или голосовые команды, что позволяет пользователям задавать вопросы или давать команды, а бот отвечает на них в реальном времени, создавая более интерактивный и персонализированный опыт.

Что касается технологий, создание веб-сайтов требует знаний в области HTML, CSS, JavaScript и других технологий веб-разработки. Веб-сайты могут использовать различные фреймворки и системы управления контентом. Разработка ботов, в свою очередь, требует знаний в области программирования, работы с API и обработки естественного языка (NLP). Боты могут быть созданы с использованием различных платформ и инструментов, таких как Dialogflow или Microsoft Bot Framework.

Наконец, примеры использования также различаются. Веб-сайты могут включать корпоративные сайты, интернет-магазины, блоги и новостные порталы. Боты же могут использоваться для поддержки клиентов, заказа еды, проведения опросов и получения обратной связи.

Таким образом, веб-сайты и боты служат различным целям и предоставляют разные способы взаимодействия с пользователями. Веб-сайты фокусируются на представлении информации и услуг, в то время как боты автоматизируют взаимодействие и выполнение задач. Оба инструмента могут быть использованы совместно для улучшения пользовательского опыта и повышения эффективности бизнеса.

					<i>KP.09.02.07.1.33.10</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		10

### 1.3 Сравнение Telegram и ВК -ботов

ВКонтакте (ВК) и Telegram (ТГ) боты имеют свои уникальные особенности и функционал. ВК боты чаще используются для взаимодействия с пользователями внутри социальной сети, предлагая различные сервисы и игры, тогда как ТГ боты более универсальны, позволяя интеграцию с внешними сервисами и автоматизацию процессов.

Критерии сравнения ВК ботов и ТГ ботов:

Таблица 1 – Основные функции

Функция	ВК боты	ТГ боты
Интерактивные игры и викторины	Возможны, но ограничены API ВК	Широкие возможности, включая геймификацию
Автоматизация задач и процессов	Ограничены возможностями ВК	Высокая степень автоматизации
Поддержка опросов и голосований	Возможна, но с ограничениями	Проста в реализации
Интеграция с API внешних сервисов	Ограничена политикой ВК	Высокая гибкость интеграции
Уведомления о событиях и новостях	Возможны, но ограничены	Широкие возможности для уведомлений
Поддержка команд и пользовательских интерфейсов	Ограничена	Высокая гибкость настройки
Возможность интеграции с другими приложениями внутри ВК	Ограничена	Не применимо напрямую

Таблица 2 – Уровень доступа и настройки

Критерий	ВК боты	ТГ боты
Доступ к данным пользователей	Ограничен политикой ВК	Гибкие настройки
Настройки бота	Зависят от API ВК, много ограничений	Большие возможности в Telegram, много библиотек, постоянные обновления от создателей
Создание уникальных команд и функционала	Ограничено	Возможность создания уникальных ботов

Таблица 3 -Аудитория и использование

Критерий	ВК боты	ТГ боты
Аудитория	Пользователи ВКонтакте. В данный момент становится популярнее в России, необходима интеграция с сервисом	Широкая аудитория
Используется	Чаще для социальных целей	Применяются в бизнесе и для поддержки клиентов

Таблица 4 - Преимущества и недостатки

Критерий	ВК боты	ТГ боты
Преимущества	Интеграция с ВК,	Высокая гибкость, интеграция с сервисами
Недостатки	Ограниченные возможности и доступ к данным	Необходимость технических знаний

#### 1.4 Анализ и выбор языка программирования для разработки чат-бота ВКонтакте

При разработке чат-бота для социальной сети ВКонтакте одним из ключевых этапов стал выбор языка программирования. Для реализации проекта рассматривались несколько популярных языков, таких как Python, JavaScript (Node.js), Java и PHP. Каждый из них имеет свои преимущества и недостатки, которые были проанализированы с учетом требований к проекту.

Основные критерии выбора языка программирования: для сравнения языков программирования были выделены следующие критерии:

##### **Простота разработки и поддержка библиотек:**

- Язык должен иметь готовые библиотеки и инструменты для работы с API ВКонтакте;
- Простота написания кода и его поддержка в будущем являются важными факторами.

##### **Производительность и асинхронность:**

- Чат-бот должен обрабатывать большое количество запросов в режиме реального времени;
- Поддержка асинхронных операций для повышения производительности.

### **Масштабируемость:**

Возможность расширения функционала бота в будущем без значительных изменений в архитектуре.

### **Сообщество и документация:**

Наличие активного сообщества и качественной документации упрощает процесс разработки и решение возникающих проблем.

### **Скорость разработки:**

Время, необходимое для создания MVP (минимально жизнеспособного продукта), является важным фактором.

### **Сравнение языков программирования**

Python

#### **Преимущества:**

- Богатая экосистема библиотек, таких как `vk_api`, `aiohttp` и `asyncio`, которые упрощают работу с API ВКонтакте и обеспечивают поддержку асинхронных операций;
- Простой и понятный синтаксис, что снижает порог входа для разработчиков;
- Высокая скорость разработки благодаря наличию готовых решений и инструментов;
- Активное сообщество и подробная документация.

#### **Недостатки:**

Производительность ниже, чем у компилируемых языков, таких как Java или C++, но для задач чат-бота этого недостатка недостаточно для отказа от языка.

					<i>KP.09.02.07.1.33.10</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

## JavaScript (Node.js)

### Преимущества:

- Отличная поддержка асинхронных операций благодаря событийно-ориентированной архитектуре;
- Широкий выбор библиотек для работы с API ВКонтакте;
- Хорошо подходит для проектов, связанных с веб-разработкой.

### Недостатки:

- Более сложный синтаксис по сравнению с Python;
- Меньшая универсальность для задач, не связанных с веб-разработкой.

## Java

### Преимущества:

- Высокая производительность и надежность;
- Хорошая масштабируемость для крупных проектов.

### Недостатки:

- Более сложный и многословный синтаксис.
- Отсутствие готовых библиотек для работы с API ВКонтакте, что увеличивает время разработки.

## PHP

### Преимущества:

- Исторически используется для взаимодействия с API социальных сетей.

### Недостатки:

- Устаревший язык с ограниченными возможностями для асинхронной обработки.
- Меньшая производительность и меньшая популярность среди разработчиков.

На основе проведенного анализа был выбран язык программирования Python. Это решение обусловлено следующими факторами:

Простота и скорость разработки: Python позволяет быстро создавать прототипы и реализовывать функционал благодаря интуитивному синтаксису и богатой экосистеме библиотек.

Поддержка асинхронности: Библиотеки `asyncio` и `aiohttp` обеспечивают эффективную обработку множества запросов в режиме реального времени.

Готовые решения для работы с API ВКонтакте: Библиотека `vk_api` предоставляет удобный интерфейс для взаимодействия с API, что значительно ускоряет процесс разработки.

Активное сообщество и документация: Python имеет огромное сообщество разработчиков, что упрощает поиск решений и помощь в случае возникновения проблем.

Универсальность: Python подходит не только для разработки чат-ботов, но и для других задач, таких как анализ данных, машинное обучение и автоматизация.

Таким образом, Python был выбран как оптимальный язык программирования для реализации чат-бота ВКонтакте, соответствующий всем требованиям проекта и обеспечивающий высокую эффективность разработки.

					КР.09.02.07.1.33.10	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		



## **Глава 2. «Разработка ВК-бота по оказанию административной помощи»**

### **2.1 Инструменты создания проекта**

Для разработки ВК-бота по оказанию административной помощи жителям города Калуги будут использоваться следующие инструменты и технологии:

#### **1. Языки программирования:**

- Python: один из самых популярных языков для разработки чат-ботов благодаря своей простоте и большому количеству библиотек, таких как `vk_api`, которые упрощают взаимодействие с API ВКонтакте.

#### **2. Среды разработки и другие программы:**

- PyCharm: интегрированная среда разработки для Python, которая предоставляет мощные инструменты для написания и отладки кода;
- Visual Studio Code: легкая и настраиваемая среда, подходящая для работы с JavaScript и другими языками;
- PuTTY — свободно распространяемый клиент для различных протоколов удалённого доступа, включая SSH, Telnet, rlogin. Также имеется возможность работы через последовательный порт. Программа позволяет подключиться и управлять удалённым узлом (например, сервером);
- FileZilla — бесплатный FTP-клиент для передачи файлов по одноимённому протоколу. Предназначен, для загрузки и скачивания файлов с FTP-серверов. Программа имеет удобный интерфейс с русскоязычной локализацией, поддерживает drag & drop, настройку прав доступа, удалённый поиск и редактирование файлов, многопоточную загрузку и при обрыве соединения и множество других опций.

#### **3. Библиотеки и фреймворки:**

					<i>KP.09.02.07.1.33.10</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

- vk\_api: библиотека для работы с API ВКонтакте, позволяющая легко отправлять и получать сообщения, а также управлять другими функциями платформы.

#### 4. Базы данных:

- SQLite или PostgreSQL: для хранения информации о пользователях, их запросах и предоставляемых услугах

### 2.2 Требования к проекту

При разработке ВК-бота необходимо учитывать следующие требования:

#### Функциональные требования:

- Бот должен предоставлять информацию о доступных административных услугах;
- Бот должен отвечать на часто задаваемые вопросы;
- Бот должен иметь возможность обрабатывать запросы пользователей и предоставлять обратную связь.

#### Нефункциональные требования:

- Высокая доступность и надежность работы бота;
- Удобный и интуитивно понятный интерфейс для пользователей;
- Защита данных пользователей и соблюдение требований законодательства о защите персональных данных.

#### Технические требования:

- Бот должен быть совместим с API ВКонтакте и соответствовать его требованиям;
- Необходимость в регулярном обновлении базы данных с актуальной информацией о государственных услугах.

## 2.3 Поставленные задачи

Для достижения поставленной цели разработки ВК-бота необходимо решить следующие задачи:

1. Провести анализ существующих решений и определить потребности целевой аудитории;
2. Разработать архитектуру и функциональные требования к ВК-боту;
3. Реализовать бота с использованием современных технологий и инструментов;
4. Провести тестирование и оценку эффективности работы бота;
5. Подготовить рекомендации по дальнейшему развитию и улучшению функционала.

Таким образом, анализ предметной области и выбор инструментов разработки являются ключевыми этапами в создании эффективного ВК-бота, который будет способствовать улучшению взаимодействия граждан с органами власти в городе Калуга.

## 2.4 Настройка сообщества в ВК.

На страничке сообщества, в основном меню по желанию можно добавить приветствие бота. Пользователь будет замечать эти изменения, когда впервые зайдет к боту и нажмет на кнопку «Начать».

В разделе «Настройки» откройте правое подменю и выберите пункт «Работа с API». Этот путь потребует использовать многократно, поэтому рекомендуется запомнить последовательность действий.

Перейдите во вкладку «Ключи доступа», где необходимо создать новый ключ. Для этого нажмите кнопку «Создать ключ». Система запросит указание прав доступа для нового ключа. В рамках данного проекта следует выбрать следующие разрешения:

					<i>КР.09.02.07.1.33.10</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

- Доступ к управлению сообществом;
- Доступ к сообщениям сообщества;
- Доступ к фотографиям сообщества.

После подтверждения создания ключ будет активирован и готов к использованию в работе.

Так же в этом разделе есть вторая вкладка «Callback API», это тоже требует настройки, но если перейти в раздел, то для этого необходимо иметь сервер, поэтому создадим свой сервер и после этого вернемся к callback.

## 2.5 Создание сервера и его настройка

Для того чтобы бот взаимодействовал с посетителями сообщества постоянно и запускать проект через IDE не приходилось, нужно выложить проект на сервер и запустить исполняемый файл.

Серверы бывают разные. Отличаются по цене, операционной системой, установленными программами.

Теперь, когда есть IP-адрес сервера можно заходить в PuTTY. При открытии приложения всплывает окошко. В строчку с «Host Name (or IP - Address)» прописываем IP. Так же чтобы сохранить подключение в окошке «Saved Session» прописать название, у меня это —«VK\_BOT» и нажать «Save», после этого не надо будет каждый раз прописывать адрес, достаточно нажать на название сессии.

Открывается окно терминала в нем прописывается пользователи и пароль. В данный момент мы работаем на сервере под учетной записью root, которая обладает максимальными привилегиями. Для безопасной разработки рекомендуется создать дополнительного пользователя с ограниченными правами, так как выполнение задач под root-аккаунтом считается небезопасной практикой.

Чтобы создать нового пользователя в Ubuntu с помощью терминала, нужно ввести команды:

					КР.09.02.07.1.33.10	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

Для создания пользователя: `sudo useradd new-username`. Вместо new-username нужно указать желаемое имя пользователя. (fr-например);

Для установки пароля: `sudo passwd new-username`. При этом будет предложено дважды ввести новый пароль.

Производить обновление операционной системы надо будет делать на протяжении всей работы, так как обновлять сервер полезно, а вот обновлять зависимости в действующем проекте не стоит, так как это может сделать его неработоспособным. Ввести в терминал следующие команды: `sudo apt-get update` и `sudo apt-get upgrade`.

2.6 Подключение к FileZilla Client

Для подключения сервера к FileZilla Client: запускаем программу и переходим на верхнюю панель, верхний левый угол, кнопка с зеленым треугольником – стрелкой, переходим на нее и выскакивает окно «Менеджер сайтов».

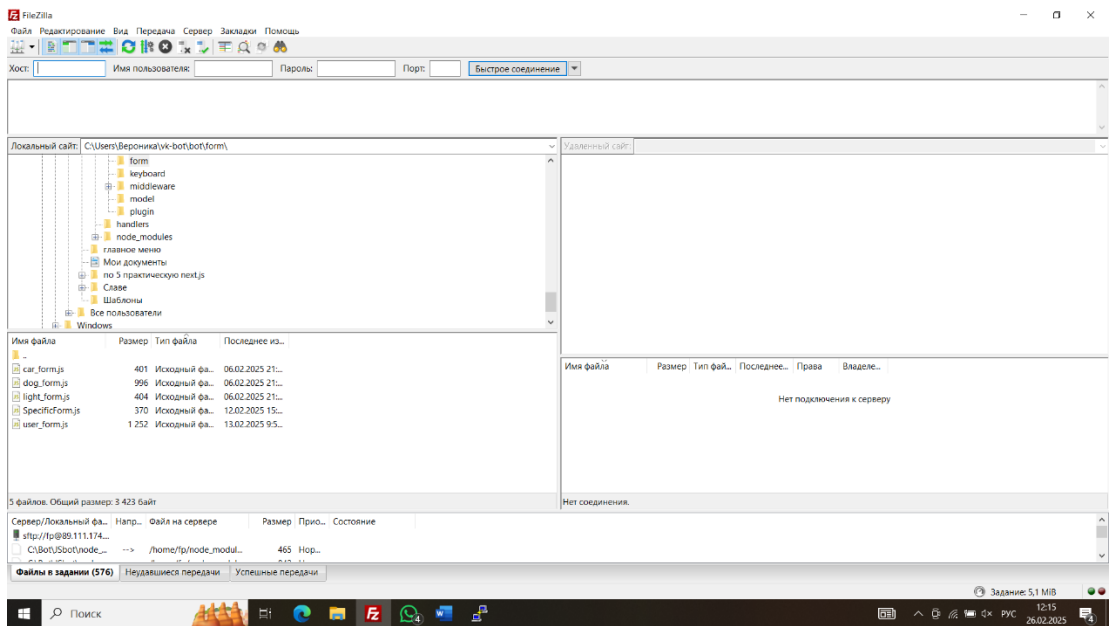


Рисунок 1 - Подключение к FileZilla Client

Прописываем хост, пользователя и его пароль (который создавали). Подключаемся к созданному проекту.

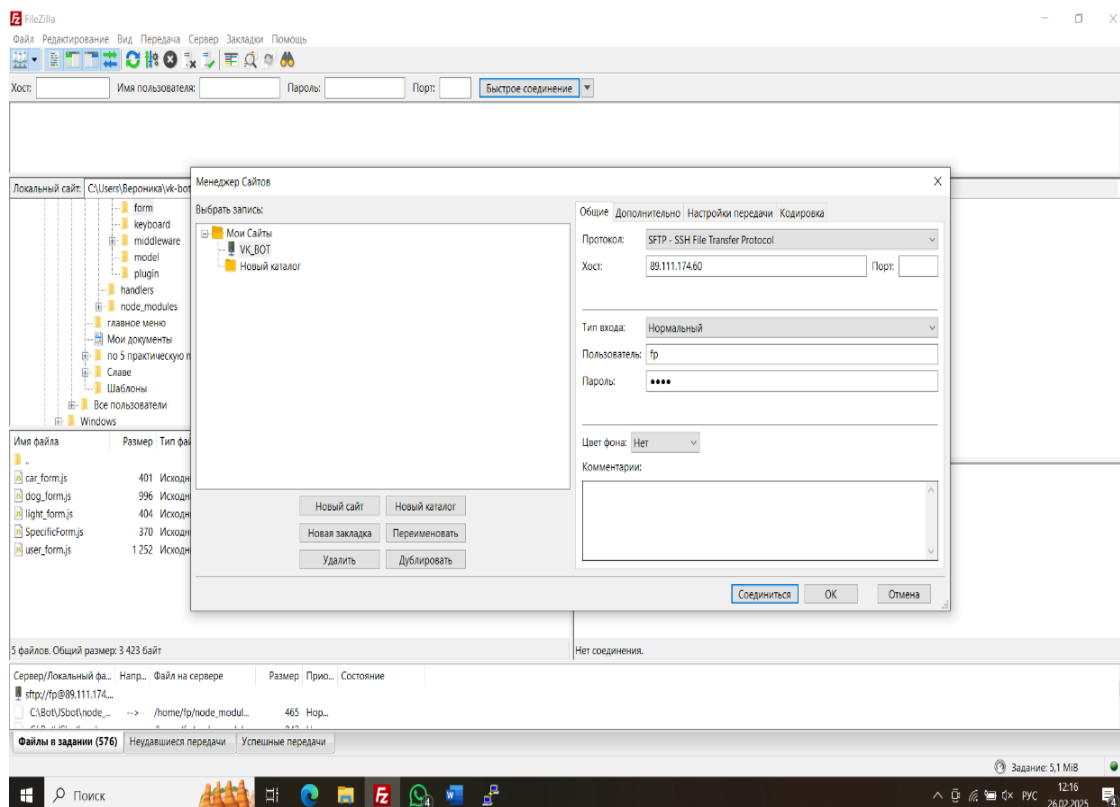


Рисунок 2 - Создание проекта на FileZilla Client

## 2.7 Подключение к Callback API ВК в ВК.

Для того чтобы работать с вк необходимо создать подключение к Callback API ВК. В разделе «Работа с API», вторая вкладка- «Callback API». Версию лучше выбирать более высокую так, как от поставленной версии зависит, то, как работать с запросами get и post. В окне «Адрес» необходимо прописать следующие команды:

`http://89.111.174.60/webhook`

Где, 89.111.174.60 – IP-адрес сервера, webhook – возвращаемое значение, его затронем позже - прописывается в коде.

ВК работает только с 80 портом, так как все приложения, чаты, боты и т.п – распознается как приложение, поэтому если прописать: `http://89.111.174.60:3000/webhook` то, это ошибочный вариант, ведь используется только 80 порт. Еще это влияет на запуск бота так как это порт с

					КР.09.02.07.1.33.10	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

особенными правами и запускать его надо только с правами root, поэтому надо будет в коде учитывать команду sudo.

Под окном «Адрес» находится справочная информация, в ней необходимо найти «Строка, которую должен вернуть сервер:» -эта строка меняется каждый день, поэтому лучше вынести ее в переменную. Она понадобится позже.

## 2.8 Создание ВК бота, структура проекта

При создании проекта создается понятная и логичная структура. Для того чтобы с ней ознакомиться, прописывается - README.md файл, в нем изложены все использованные папки и файлы для создания проекта, а также пошаговые действия для запуска ВК бота.

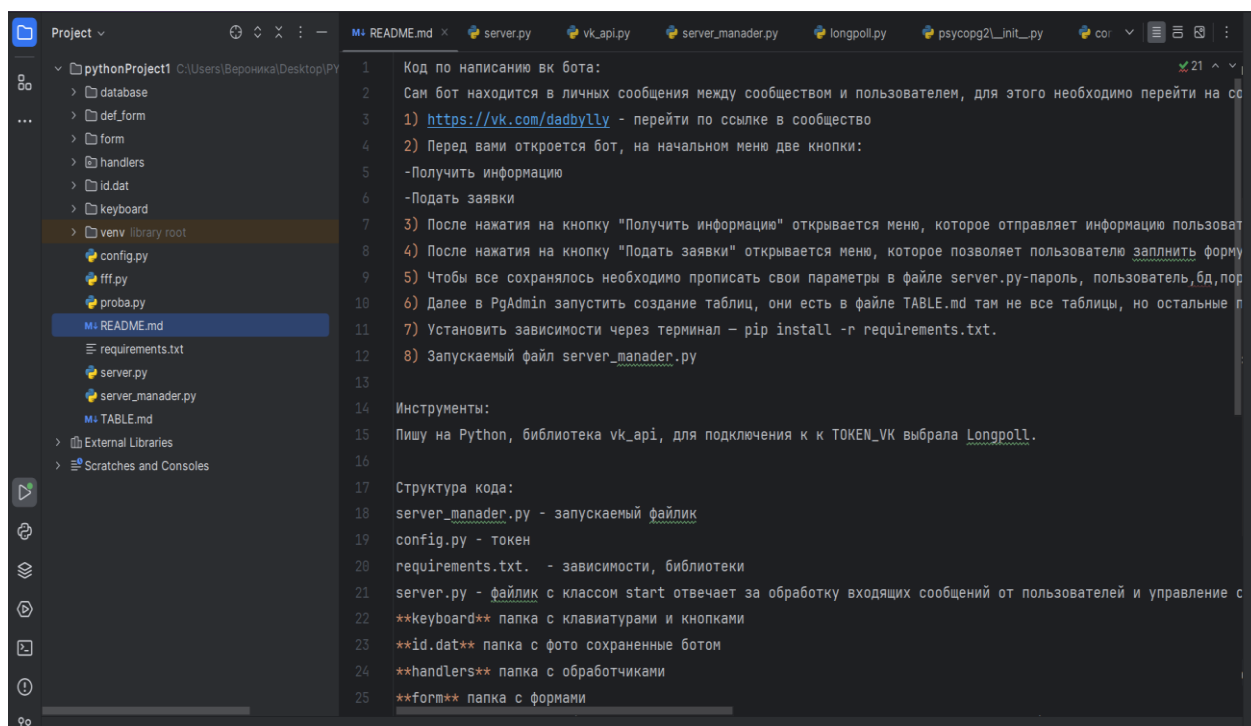


Рисунок 3 - Скриншот кода с его структурой и README.md файл

### Структура кода:

- server\_manader.py - запускаемый файл;

- config.py - находится токен бота, это является важным аспектом, так как иногда ключей у вк бота может быть несколько и если в них запутаться или же удалить, то работать приложение не будет;
- requirements.txt. - зависимости, библиотеки;
- server.py - файл с классом start отвечает за обработку входящих сообщений от пользователей и управление состояниями пользователей в зависимости от их взаимодействия с ботом;
- keyboard - папка с клавиатурами и кнопками;
- handlers - папка с обработчиками;
- form - папка с формами.

Иллюстрация «Отрывок из README.md файла, который описывает основные папки в коде, а также действующие файлы»

Так же добавлен файл TABLE.md, в котором уже прописан код для БД, чтобы можно было просто вставить скопированный скрипт и не создавать таблицы заново.

```

1 CREATE TABLE dog_forms (
2     id SERIAL PRIMARY KEY,
3     full_name VARCHAR(255) NOT NULL,
4     phone VARCHAR(50),
5     description TEXT,
6     breed VARCHAR(100),
7     latitude DECIMAL(9, 6),
8     longitude DECIMAL(9, 6),
9     created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP
10 );
11
12 CREATE TABLE car_forms (
13     id SERIAL PRIMARY KEY,
14     full_name VARCHAR(255) NOT NULL,
15     phone VARCHAR(50),
16     description TEXT,
17     number_avto VARCHAR(100),
18     latitude DECIMAL(9, 6),
19     longitude DECIMAL(9, 6),
20     created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP
21 );
22
23 CREATE TABLE light_forms (
24     id SERIAL PRIMARY KEY,
25     full_name VARCHAR(255) NOT NULL,
26     phone VARCHAR(50),
  
```

Рисунок 4 - Файл TABLE.md, в котором уже прописан код для БД



## 2.9 Подключение к базе данных PostgreSQL

Для корректной работы бота, необходимо реализовать базу данных (БД).

**При создании учитывались такие требования:**

1. Пользователи, заполняют формы и отправляют ее данные в БД;
2. Реализация отправки данных с геолокацией, используя ВК-кнопка – «Геолокация»);
3. В БД отражаются валидные данные, все столбцы заполнены, в соответствии с тем, что отправляет пользователь, корректное отображение id и геолокации (координаты).

Для подключения к PostgreSQL нужно, знать такие данные как:

dbname="postgres",

user="postgres",

password="1234",

host="localhost",

port="5432"

Вот как работает база данных в данном боте:

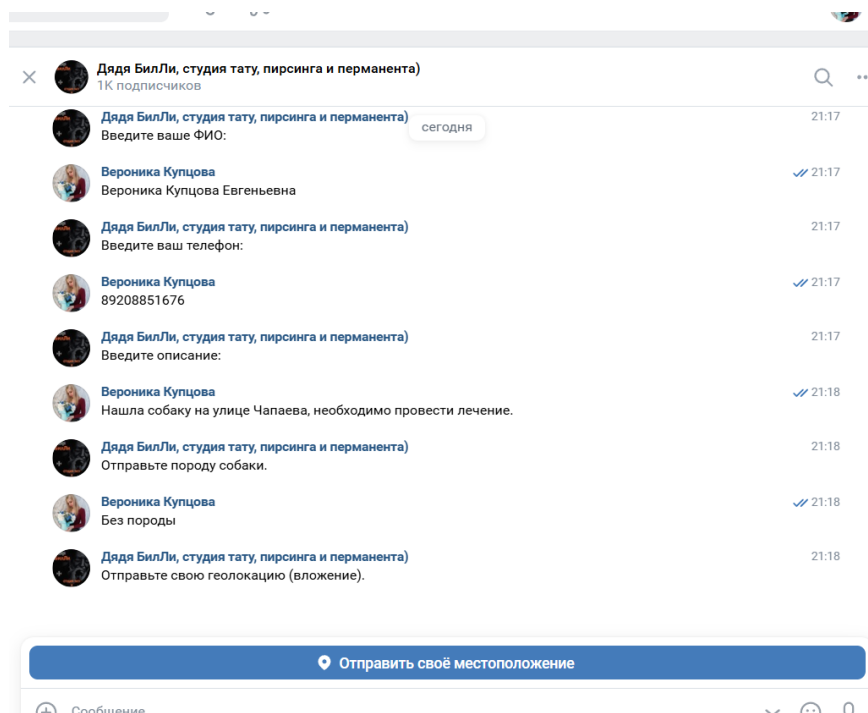


Рисунок 5 -Проверка заполнения формы в базу данных

					КР.09.02.07.1.33.10	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

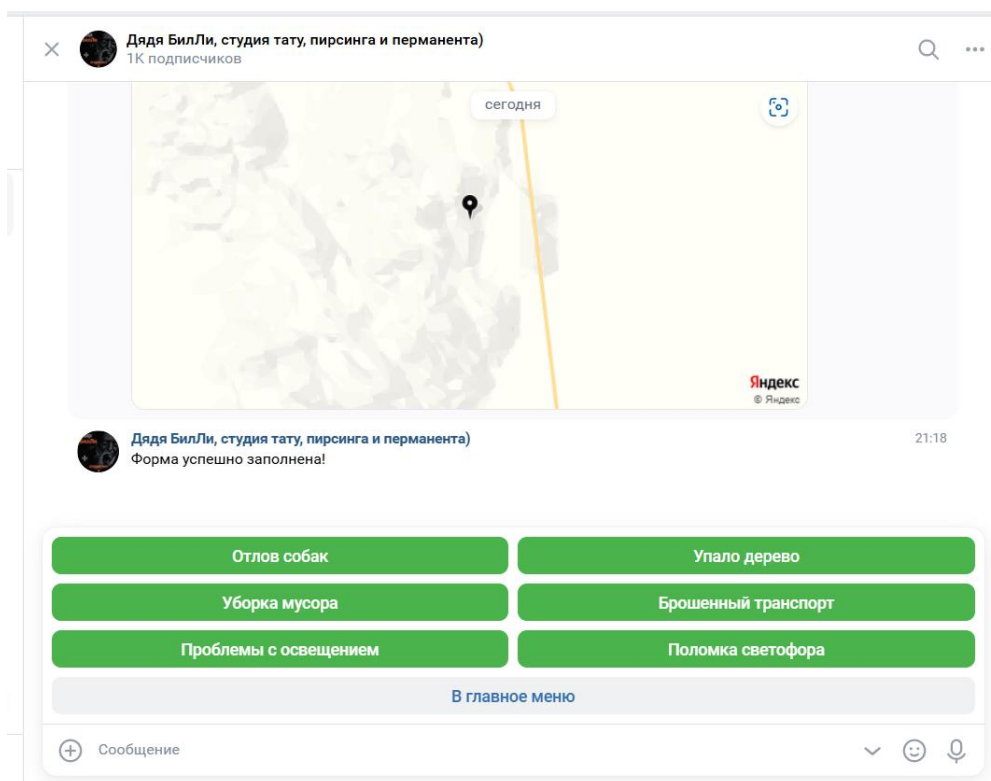


Рисунок 6 - Проверка заполнения формы в базу данных, отправка геолокации с помощью ВК-кнопка «Геолокация»

Отображение в RqAdmin:

public.dog_forms/postgres/postgres@PostgreSQL 11							
id	full_name	phone	description	breed	latitude	longitude	created_at
1	ghbyd	fojif	jdjd	jdjd	10.356820	9.843739	2025-04-10 21:13:56.859953
2	Вероника Купцова Евгеньевна	89208851676	Нашла собаку на улице Чапаева, необходимо провести лечени...	Без породы	12.092115	18.984377	2025-04-10 21:18:39.633295

Рисунок 7 - Проверка заполнения формы в RqAdmin

По рисунку 7 можно сказать, что данные пришли валидные, в полном объеме.

## 2.10 Создание форм

В боте присутствует несколько форм: «Отлов собак», «Упало дерево», «Уборка мусора», «Брошенный транспорт», «Проблемы с освещением» и «Поломка светофора»

Если проанализировать, то есть у всех форм одинаковые поля, следовательно создавать все по отдельности не является нормированным решением. Для того, чтобы данную задачу решить, необходимо использовать Объектно-ориентированное программирование. Главным родителем будет таблица так называемая – user\_form – ее использовать надо в формах, где только родительские столбцы: full name, phone, description, location. Остальные же формы с индивидуальными данными, будут наследовать от родителя столбцы и так же добавляться новые.

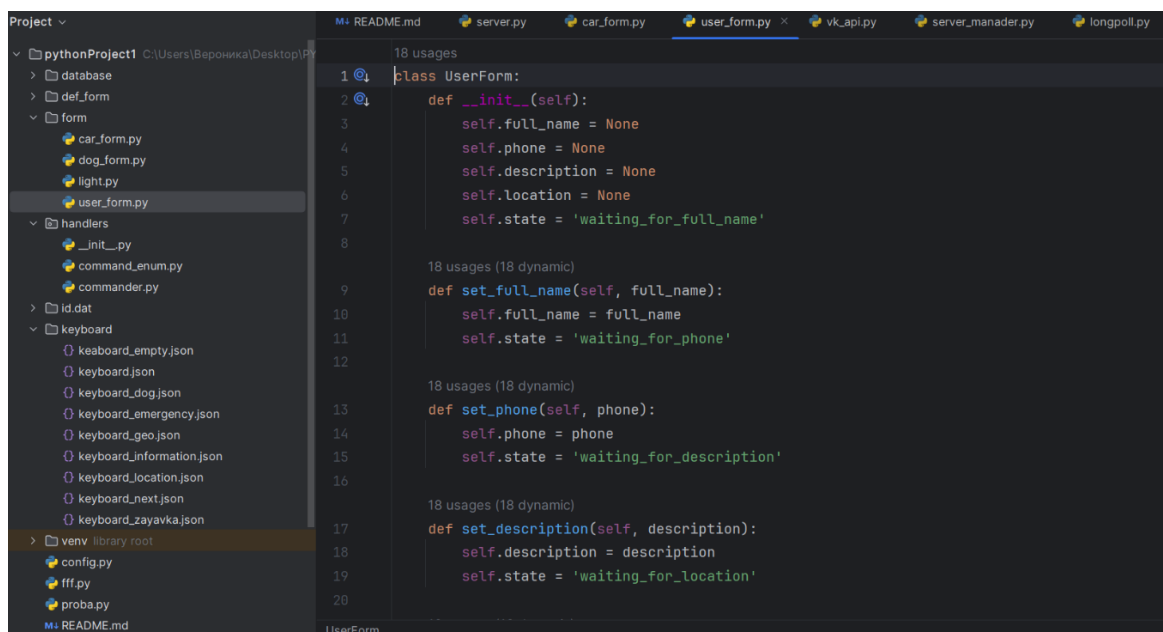


Рисунок 8 - Пример родительской формы – UserForm

Создается конструктор, в котором лежат переменные с названиями столбцов»

```
nd server.py car_form.py x user_form.py light.py dog_form.py vk_api.py longpoll.py
1 from form.user_form import UserForm
2
3
4 class CarForm(UserForm):
5     def __init__(self):
6         super().__init__()
7         self.number = None
8
9
10     def set_number(self, number):
11         self.number = number
12         self.state = 'waiting_for_number'
13
14     def is_car_completed(self):
15         return all([self.full_name, self.phone, self.description, self.location, self.number])
```

Рисунок 9 - Пример формы «Брошенный транспорт

Которая наследует данные от UserForm, а также добавляет индивидуальные столбцы»

## 2.11 Keyboard

Для создания клавиатуры в ВК боте, часто используется JSON формат.

JSON (JavaScript Object Notation) — это легковесный формат обмена данными, который легко читается и записывается человеком, а также легко парсится и генерируется машинами. Он широко используется в веб-разработке и API, включая создание клавиатур для ботов в ВКонтakte (VK)

В данном боте используется много клавиатур, они хранятся в папке – keyboard.

```

1
2 {
3   "one_time": false,
4   "buttons": [
5     [
6       {
7         "action": {
8           "type": "text",
9           "label": "Получить информацию",
10          "payload": "{\"button\": \"1\"}"
11        },
12        "color": "positive"
13      }
14    ],
15    [
16      {
17        "action": {
18          "type": "text",
19          "label": "Подать заявку",
20          "payload": "{\"button\": \"2\"}"
21        },
22        "color": "positive"
23      }
24    ]
25  ]
26 }

```

Рисунок 10 - Клавиатура в формате JSON

Payload отвечает за обработку кнопки, то есть действия, которые будут следовать после ее нажатия. Цвет кнопки можно изменить через color. ВК имеет ограниченную палитру, состоящую из следующих цветов: negative, positive, primary, secondary.

У ВК есть разные виды кнопок за это отвечает -type. Если использовать text, то у кнопки будет label-подпись, а вот у кнопки type : location, отсутствует текст и соответственно label, так как этот тип кнопки имеет встроенный текст и изображение.

```

1  {
2      "buttons": [
3          [
4              {
5                  "action": {
6                      "type": "location"
7                  }
8              }
9          ]
10     ],
11     "one_time": false
12 }
13
14

```

Рисунок 11 - Клавиатура в формате JSON с кнопкой - type: location

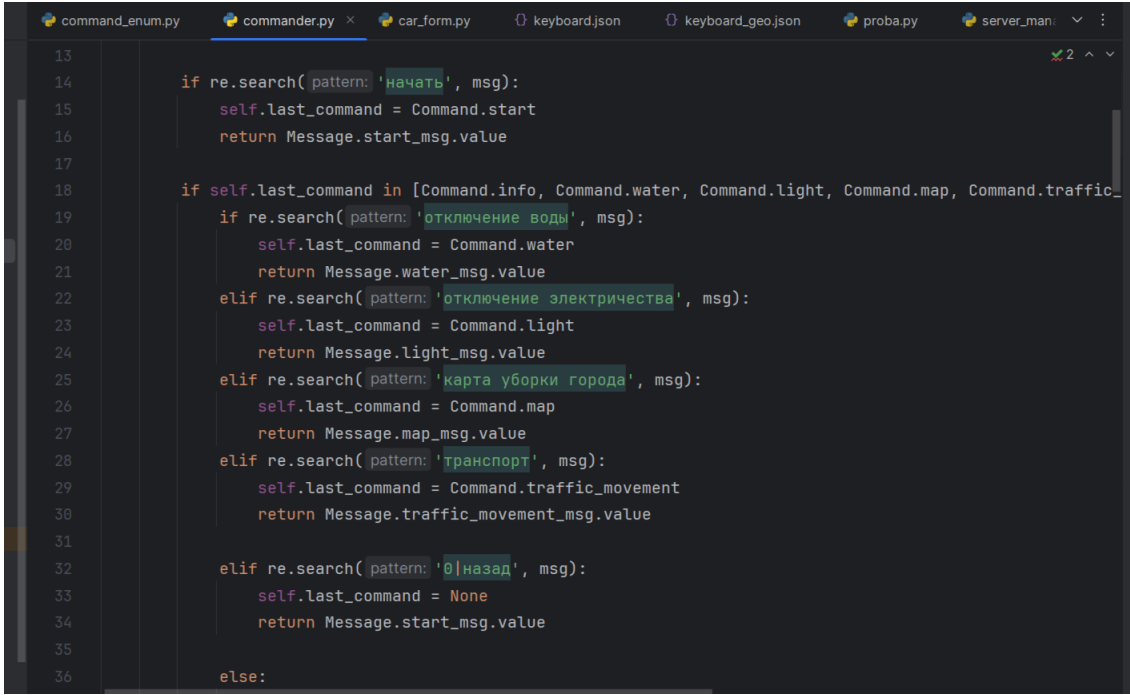
## 2.12 Обработчики

Обработчики или же по-другому handlers- отвечают за действие, которое следует после выполнения той или иной команды. В данном проекте реализованы разные виды обработчиков:

Первый это – отправка пользователю сообщения после нажатия кнопки. В боте это применяется, если перейти с основного меню на клавиатуру с получением информации. Для каждой кнопки с информацией прописывается отдельно команда и сам текст, то есть возвращаемое значение, оно хранится в переменных.

Второй это – обработчик кнопок, которые могут не возвращать текстовое сообщение, но выполнять другие действия, например переход на новую сцену. Сцена- этап в логике работы бота, который идёт по определённому сценарию. Сцены удобны, так как позволяют сократить

количество кода, необходимого для ответа пользователю и обработки данных. Например, если в сцене есть кнопки, то при переходе на следующий шаг предыдущие уже не будут срабатывать. За это отвечают кнопки «Получить информацию» и «Подать заявку», после нажатия на этих кнопок меняется клавиатура на новую.



```
13
14     if re.search( pattern: 'начать', msg):
15         self.last_command = Command.start
16         return Message.start_msg.value
17
18     if self.last_command in [Command.info, Command.water, Command.light, Command.map, Command.traffic_
19         if re.search( pattern: 'отключение воды', msg):
20             self.last_command = Command.water
21             return Message.water_msg.value
22         elif re.search( pattern: 'отключение электричества', msg):
23             self.last_command = Command.light
24             return Message.light_msg.value
25         elif re.search( pattern: 'карта уборки города', msg):
26             self.last_command = Command.map
27             return Message.map_msg.value
28         elif re.search( pattern: 'транспорт', msg):
29             self.last_command = Command.traffic_movement
30             return Message.traffic_movement_msg.value
31
32         elif re.search( pattern: '0|назад', msg):
33             self.last_command = None
34             return Message.start_msg.value
35
36     else:
```

Рисунок 12 – Обработчики первого вида, реагирующие текстовым сообщением на нажатие кнопки

```

1  from enum import Enum
2
3  42 usages
4  class Command(Enum):
5      """Класс, содержащий перечисление команд"""
6      """ Начать """
7      start = ["Начать", "начать", "start", "Start"]
8
9      # Первые кнопки две
10     """Получить информацию"""
11     info = ["1", "Получить информацию"]
12
13
14     """Подать заявку"""
15     заявка = ["2", "Подать заявку"]
16
17
18     #Информационная клавиатура
19     """карта уборки города"""
20     map = ["2", "карта уборки города"]
21
22     """ Отключение воды """
23     water = ["5", "Отключение воды"]

```

Рисунок 13 - Обработчики второго вида, меняющие сцену

## 2.13 Тестирование бота

В ходе тестирования были выявлены несколько ключевых областей, которые могут быть улучшены для повышения удобства использования бота. Ниже приведены основные предложения:

### 1. Реализация Inline-кнопок

На данный момент бот использует текстовые кнопки, которые срабатывают на основе текстового ввода. Однако более логичным решением будет внедрение Inline-кнопок.

#### Преимущества:

- Inline-кнопки работают не с текстом, а с payload (скрытыми данными), что делает их более надежными и менее подверженными ошибкам;
- Пользователь получает четко структурированный интерфейс, где каждый выбор четко обозначен;
- Это также упрощает логику обработки данных на стороне бота.



**Рекомендация:** Заменить текстовые кнопки на Inline-кнопки для всех ключевых взаимодействий (например, выбор региона, подтверждение действий и т.д.).

## 2. Обработка некорректного ввода региона

- Если пользователь вводит регион, отличный от допустимого (например, не «г. Калуга»), бот должен своевременно информировать его об ошибке.

**Проблема:** Пользователь может случайно или намеренно указать неверный регион, что может привести к некорректной работе бота или путанице.

### Решение:

При вводе региона, который не соответствует допустимым значениям, бот должен отправлять сообщение об ошибке, например: "Вы ввели некорректный регион. Пожалуйста, укажите 'г. Калуга' или выберите подходящий вариант из предложенных кнопок."

Это обеспечит корректность данных и минимизирует количество ошибок при заполнении формы.

## 3. Управление процессом заполнения формы

На данный момент пользователь имеет возможность покинуть процесс заполнения формы в любой момент. Это решение имеет как положительные, так и отрицательные стороны.

### Положительные стороны:

- Пользователь сохраняет свободу выбора и может прервать процесс без каких-либо ограничений;
- Это создает более комфортный опыт взаимодействия с ботом.

					КР.09.02.07.1.33.10	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

### Отрицательные стороны:

- Если пользователь начинает заполнять форму, но прерывает процесс, его данные сохраняются в текущем состоянии. Это может привести к накоплению незавершенных записей в системе.

### Рекомендация:

Добавить механизм подтверждения выхода например: «Вы уверены, что хотите покинуть процесс заполнения формы? Все введенные данные будут сохранены, и вы сможете продолжить позже».

Также можно реализовать функцию автоматического удаления незавершенных данных через определенный период времени (например, 24 часа).

Внедрение вышеописанных улучшений позволит сделать взаимодействие с ботом более удобным, интуитивно понятным и эффективным. Inline-кнопки упростят навигацию, обработка ошибок повысит точность данных, а управление процессом заполнения формы обеспечит баланс между свободой пользователя и целостностью данных.

## Заключение

В данной курсовой работе был рассмотрен процесс создания чат-бота для социальной сети ВКонтакте, предназначенного для оказания помощи гражданам по административным вопросам и проблемам. Проект реализован с использованием современного стека технологий, включающего Python 3.10, библиотеку VK API (vk\_api), асинхронные инструменты (asyncio), базу данных PostgreSQL. Основное внимание уделялось как техническим аспектам разработки, так и практической применимости решения для улучшения взаимодействия между гражданами и администрацией в кризисных условиях.

Проект включал несколько ключевых этапов: анализ требований к функционалу, разработку логики работы для пользователей и администраторов, создание информационной базы данных для хранения информации о пользователях, сообщениях, обращениях и экстренных уведомлениях, а также программирование функционала бота с использованием асинхронных возможностей языка Python. Особое внимание было уделено следующим аспектам:

**Оперативность и доступность коммуникации:** Бот разработан таким образом, чтобы граждане могли быстро отправлять обращения, получать актуальную информацию через простой и интуитивно понятный интерфейс. Логика бота минимизирует количество шагов для выполнения ключевых действий, таких как отправка заполненных заявок.

**Безопасность данных и защита персональной информации:** Применение современных методов аутентификации и шифрования, а также проверка входящих данных через механизм VK API, обеспечивают надежную защиту пользовательских данных. Это предотвращает несанкционированный доступ к конфиденциальной информации и гарантирует безопасность обмена данными между гражданами и администрацией.

**Высокая производительность и масштабируемость:** Асинхронная обработка запросов с использованием библиотеки asyncio и эффективная

работа с базой данных PostgreSQL обеспечивают стабильную работу бота даже при высоких нагрузках. Выбранная архитектура позволяет легко масштабировать систему для обслуживания большего количества пользователей в случае масштабных чрезвычайных ситуаций.

Эффективное управление обращениями и уведомлениями: Модульная архитектура бота позволяет администраторам оперативно отслеживать поступающие обращения. Это значительно упрощает администрирование и повышает оперативность реагирования на критические события.

Практический опыт, полученный в ходе разработки, подтвердил, что использование современного стека технологий на базе Python позволяет создать надежное и удобное решение для поддержки связи между гражданами и администрацией. Разработанный чат-бот демонстрирует потенциал для дальнейшего развития и внедрения инновационных решений в сфере обеспечения безопасности и координации действий населения, способствуя повышению эффективности реагирования на чрезвычайные ситуации и снижению рисков для граждан.

Таким образом, в работе были достигнуты поставленные цели и задачи, успешно демонстрируя полученные знания и навыки в области разработки, интеграции и обеспечения безопасности современных чат-ботов для социальных сетей. Реализованный проект подтверждает возможность использования автоматизированных систем для улучшения взаимодействия между гражданами и администрацией, что может быть полезным инструментом для повышения уровня безопасности и получение информации в различных ситуациях в скорые строки.

## Список используемых источников

1. Документация | ВК для разработчиков [Электронный ресурс]. URL: <https://dev.vk.com/> (дата обращения: 05.01.2025).
2. Интеграция | Чат-боты | Разработка | Клавиатура | VK для разработчиков [Электронный ресурс]. URL: <https://dev.vk.com/api/bots> (дата обращения: 05.01.2025).
3. npm | Home [Электронный ресурс]. URL: <https://www.npmjs.com/> (дата обращения: 05.01.2025).
4. Express - фреймворк веб-приложений Node.js [Электронный ресурс]. URL: <https://expressjs.com/> (дата обращения: 05.01.2025).
5. API | VK-IO [Электронный ресурс]. URL: <https://vk-io.github.io/vk-io/> (дата обращения: 05.01.2025).
6. Документация vk\_api — vk\_api 11.9.9 documentation [Электронный ресурс]. URL: <https://vk-api.readthedocs.io/en/latest/> (дата обращения: 05.01.2025).
7. Современный облачный хостинг провайдер | Aéza [Электронный ресурс]. URL: <https://aéza.net/> (дата обращения: 05.01.2025).
8. xTunnel - аналог ngrok [Электронный ресурс]. URL: <https://xtunnel.io/> (дата обращения: 05.01.2025).
9. Как пользоваться командой apt-get в Linux [Электронный ресурс]. URL: [https://youtu.be/dk\\_a8DVj3\\_g?si=AdkXD1zolA4gnSfD](https://youtu.be/dk_a8DVj3_g?si=AdkXD1zolA4gnSfD) (дата обращения: 05.01.2025).
10. Express application generator [Электронный ресурс]. URL: <https://expressjs.com/en/starter/generator.html> (дата обращения: 05.01.2025).
11. Академия Selectel, Ботоводство [Электронный ресурс]. URL: <https://selectel.ru/blog/botovodstvo/> (дата обращения: 29.04.2025).
12. Аудитория Telegram: исследование Mediascope [Электронный ресурс]. – URL: <https://mediascope.net/news/1601603/> (дата обращения: 03.01.2025).

13. Вигерс К., Разработка требований к программному обеспечению. 3-е изд., дополненное Битти Д., Вигерс К., Издательство «Русская редакция», 736 стр.

а. Руководство по UML-диаграммам и моделированию баз данных [Электронный ресурс]. URL: <https://www.microsoft.com/ru-ru/microsoft365/business-insights-ideas/resources/guide-to-uml-diagramming-and-databasemodeling/> (дата обращения: 01.04.2025).

14. СберБизнесБот [Электронный ресурс]. URL: <https://sberbb.ru/> (дата обращения: 01.04.2025).

15. Телеграмм: социальная сеть или что-то другое? [Электронный ресурс]. URL: <https://uchet-jkh.ru/i/telegram-socialnaya-set-ili-cto-to-drugoe/> (дата обращения: 03.01.2025).

16. Что такое чат-бот? [Электронный ресурс]. URL: <https://www.oracle.com/cis/chatbots/what-is-a-chatbot/> (дата обращения: 25.02.2025).

17. Python [Электронный ресурс]. URL: <https://www.python.org/> (дата обращения: 24.01.2025).