1. **Общая информация**

**Цель проекта:**

Разработать Telegram-бота, который поможет наладить эффективное взаимодействие между администраторами и стажерами. Бот будет выполнять функции сбора контактных данных, обмена документами, коммуникации и организации задач.

**Назначение системы:**

Система предназначена для автоматизации процессов работы HR отдела и оптимизации временных издержек при постановлении задачи, приеме на работу или увольнении с работы, получении необходимых сведений. Повысить эффективность и качество адаптации стажеров, сократив временные и организационные затраты администраторов на рутинные операции и улучшив коммуникацию между всеми участниками процесса.

* **Автоматизировать** рутинные процессы (сбор данных, ответы на частые вопросы, напоминания).
* **Стандартизировать** процесс onboarding-а (чтобы каждый стажер получал одинаково полную и структурированную информацию).
* **Централизовать** информацию и документооборот (исключить ситуацию, когда договор «где-то в чате»).
* **Ускорить**коммуникацию (стажер всегда знает, к кому и по какому вопросу обратиться).
* **Обеспечить** легкий доступ к учебным материалам и задачам для стажера.
* **Снизить**нагрузку на администраторов и HR, освободив их время для более важных и персонализированных задач.

**Функциональные требования:**

0. Роли в приложении:

* Администратор (админ) - пользователь с наивысшими привилегиями, имеет функционал осмотра всех стажеров, видит списки, задачи и дедлайны каждого;
* Стажер - пользователь-стажер, видит только себя и свои задачи;
* Суперадминистратор (суперадмин) - контролирует регистрацию админов, приходит сообщение, одобряющее или не одобряющее регистрацию администратора, удаление с роли админа, следит за базой данных.

1. Сбор контактных данных

* Кнопка "Поделиться контактом";
* После нажатия, бот должен запросить у пользователя разрешение на получение его username;
* Username стажера должен быть передан администратору.

2. Отправка резюме

* Функционал отправки резюме;
* Пользователь должен иметь возможность загрузить резюме в формате PDF или DOCX;
* Бот передает загруженное резюме на утвержденный контакт администратора.

3. Отправка сообщений от администратора

* Сообщения от "Администратора";
* Администратор может отправлять сообщения любому стажеру через бота;
* Имя отправителя будет отображаться как "Администратор стажировки".

4. Управление задачами

* Создание и отслеживание задач;
* Администратор может создавать задачи для одного или нескольких стажеров;
* У каждой задачи может быть установлен конкретный дедлайн;
* Стажеры получают уведомление о новой задаче и ее дедлайне.

5. Отправка результатов

* Прикрепление документов;
* Стажеры имеют возможность прикреплять файлы (документы, скриншоты и т.д.) и отправлять их через бот;
* Файлы должны быть переданы в утвержденное хранилище или контакт администратора.

**Нефункциональные требования:**

1. Безопасность

Доступ к управлению ботом (административные функции) только для уполномоченных лиц.

2. Производительность

Бот должен отвечать на команды пользователя не более чем за 5 секунды.

3. Интерфейс

* Удобный и интуитивно понятный интерфейс для пользователей;
* Легкий доступ к основным функциям через кнопки и команды.

4. Системные требования

Совместимость с Telegram API и актуальными версиями мессенджера.

**Технические детали разработки:**

1. Языки программирования и технологии

* Backend: Любой;
* Внешние сервисы: для хранения файлов могут использоваться облачные хранилища или что угодно.

2. Инфраструктура

* Размещен на сервере с доступом в интернет. (сервер можем предоставить);
* Использование баз данных для хранения информации о задачах и их статусах.

**Разработка и тестирование:**

* Дизайн и проектирование архитектуры бота: до 01.03.2025;
* Реализация функционала: до 20.04.2025;
* Тестирование всех функций и исправление ошибок: до 12.05.2025.

### **Потенциальные уязвимости:**

Метод тестирования: Черный ящик

### **1. Уязвимости Веб-сервера и Веб-хука**

* **Подделка веб-хуков (Webhook Spoofing):**
* **Раскрытие IP-адреса сервера:**

### **2. Уязвимости Приложения (Кода Бота)**

* **Небезопасная обработка пользовательского ввода (Injection Flaws):**
* **Небезопасная десериализация данных:**
* **Неправильное управление доступом (IDOR & Privilege Escalation):**
* **Утечка чувствительной информации:**

### **3. Уязвимости Инфраструктуры (Docker, Ubuntu)**

* **Привилегированный контейнер**
* **Уязвимости в базовом образе Docker**

### **4. Уязвимости Данных (SQLite)**

* **Хранение секретов в открытом виде**
* **Отсутствие шифрования базы данных**

1. **Реализованные функции:**

**2.1 Общая структура проекта:**

Telegram-бот, развернутый на сервере Ubuntu и работающий через веб-хуки, получает обновления от Telegram в реальном времени. Каждый раз, когда пользователь отправляет сообщение или выполняет действие, Telegram отправляет HTTP-запрос на заранее настроенный URL сервера. Сервер с ботом принимает этот запрос, обрабатывает данные, и в зависимости от заложенной логики выполняет определенные действия: отправляет ответное сообщение, обрабатывает команды, взаимодействует с базой данных или внешними API.

Бот разворачивается через Docker Compose, используя конфигурацию из docker-compose.yml. Сервис работает в изолированном контейнере с подключенными томами (volumes) для данных и переменными окружения из .env. При запуске (docker-compose up) система автоматически создает сеть, монтирует файлы и запускает контейнер с ботом, который взаимодействует с внешними API (Telegram и др.). Ошибки мониторятся через логи (docker-compose logs), а пересборка образа выполняется с очисткой кэша (--no-cache).

База данных (далее - БД) реализована через SQLlite и SQLAlchemy, локально хранящие в себе данные, ключи и тд.

Структура проекта построена так, что функционал делиться на 3 основные части:

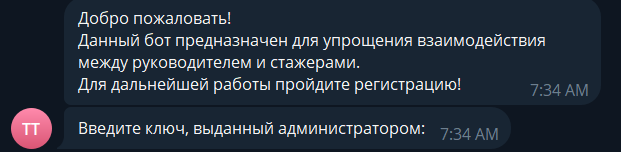
1. Использование бота со стороны стажера
2. Использование бота со стороны админа
3. Использование бота со стороны суперадмина

С кодом Telegram-бота можно ознакомиться в прилагаемом файле.

Сервер, на котором находится бот: 10.0.10.24

**3. Начало работы с ботом**

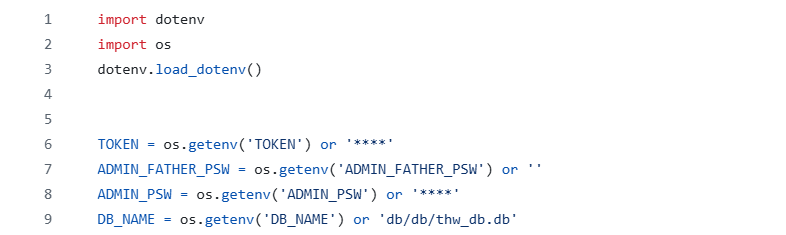
При активации бота командой “/start” будет выведено следующее:



Данный этап идет общим для всех видов пользователей(стажер, админ, суперадмин). Со стороны разработчика ключ указывается в docker-compose.yml файле.



После используется в файле [config.py](http://config.py)



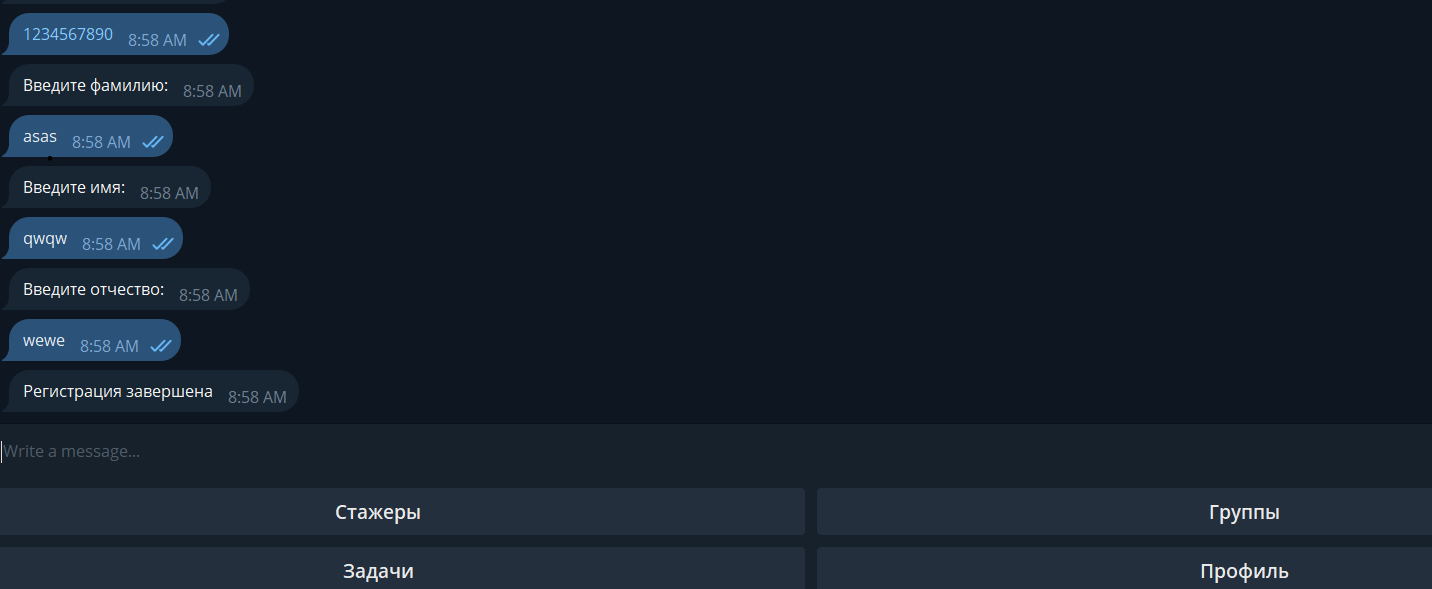
Далее рассмотрим функционал 3 разных пользователей.

*Большинство функций имеют обработку ошибок для стабильной работы бота, о некорректном вводе или выборе команды пользователю приходит уведомление.*

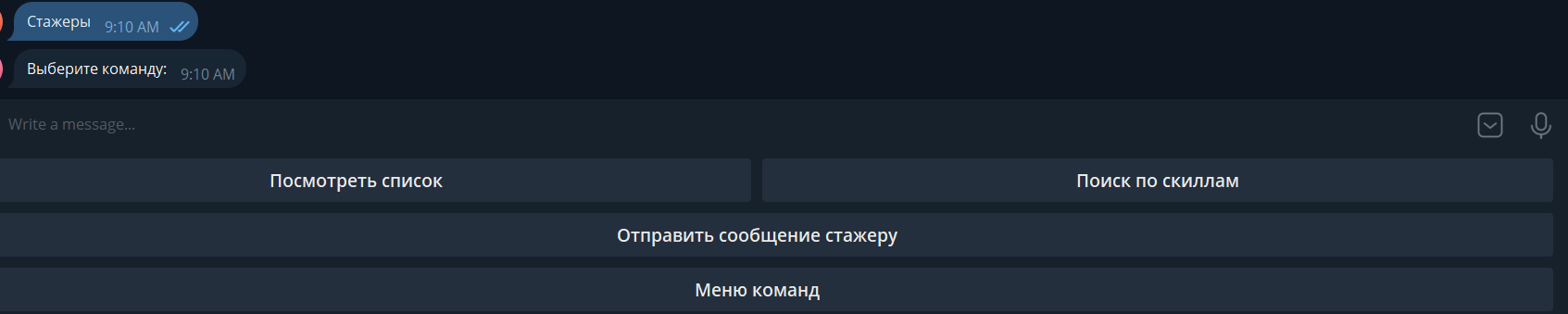


**3.1 Функционал администратора**

После заполнения информации о себе, будут доступны следующие действия:



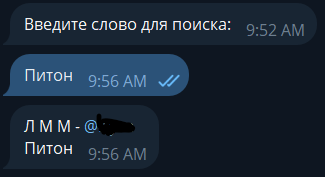
При переходе в стажеры предлагается следующий выбор команд:



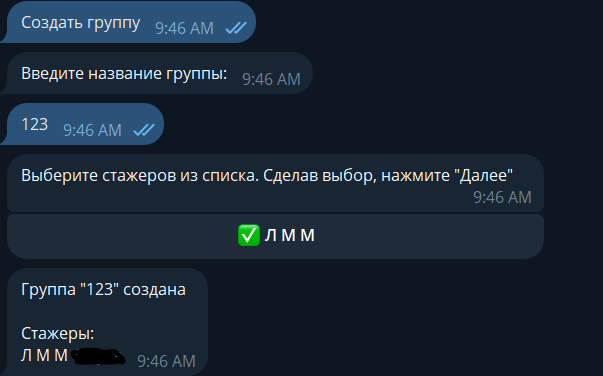
При регистрации новых пользователей администратору приходит уведомление в форме (Зарегистрирован новый пользователь:@username Никнейм):



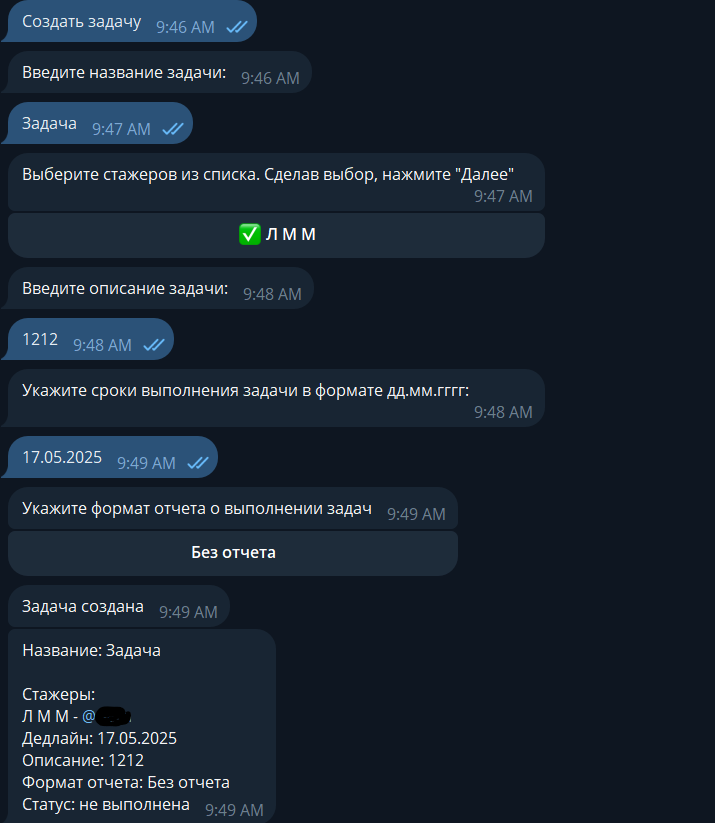
Помимо этого, админ может осуществлять поиск стажеров по скиллам. Поиск реализован на основе поиска с regex, который не чувствителен к целостности записи или регистру.



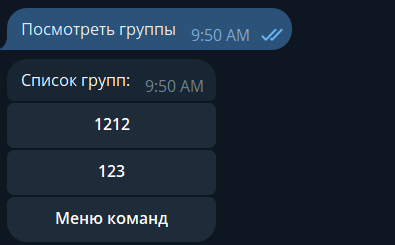
Админы также могут создавать группы из стажеров:



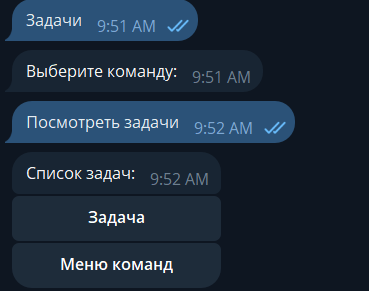
Администраторам доступна опция “Создание задачи”:



Работа команды “Просмотр групп”



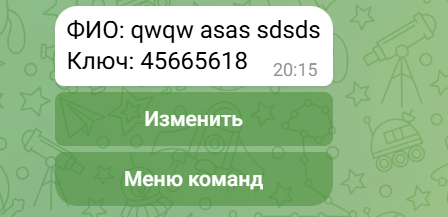
Просмотр всех доступных задач



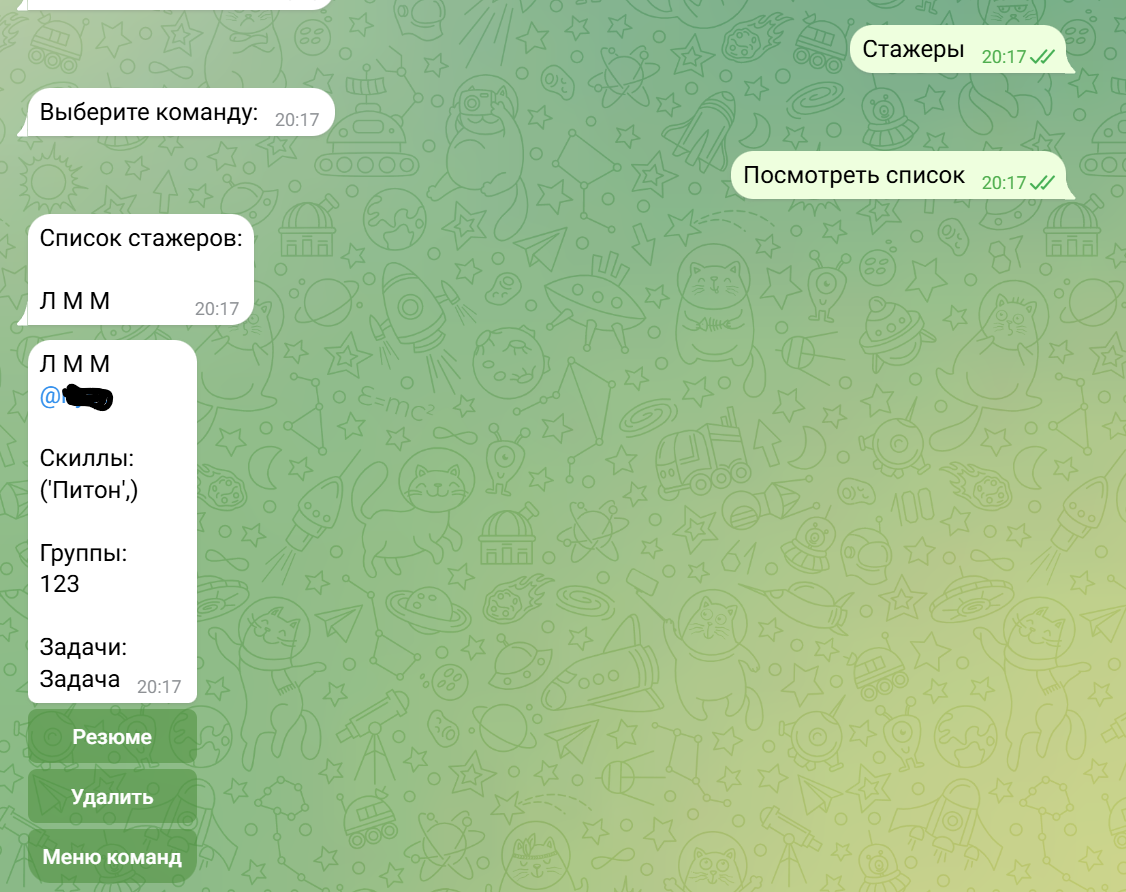
При выборе группы/задачи/стажера из выпадающего списка приходит отдельное сообщение с более подробной информацией.

У админа также есть возможность отправлять личные сообщения стажерам 

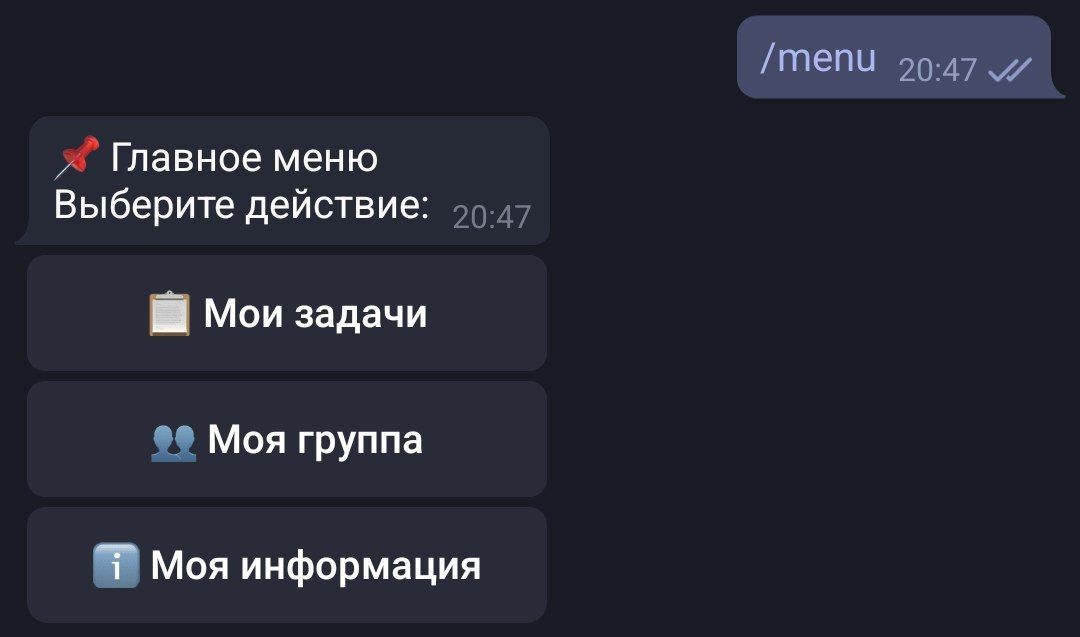
Ключ для стажера админ может получить в своем профиле. Профиль возможно изменить (фамилию, имя или отчество)



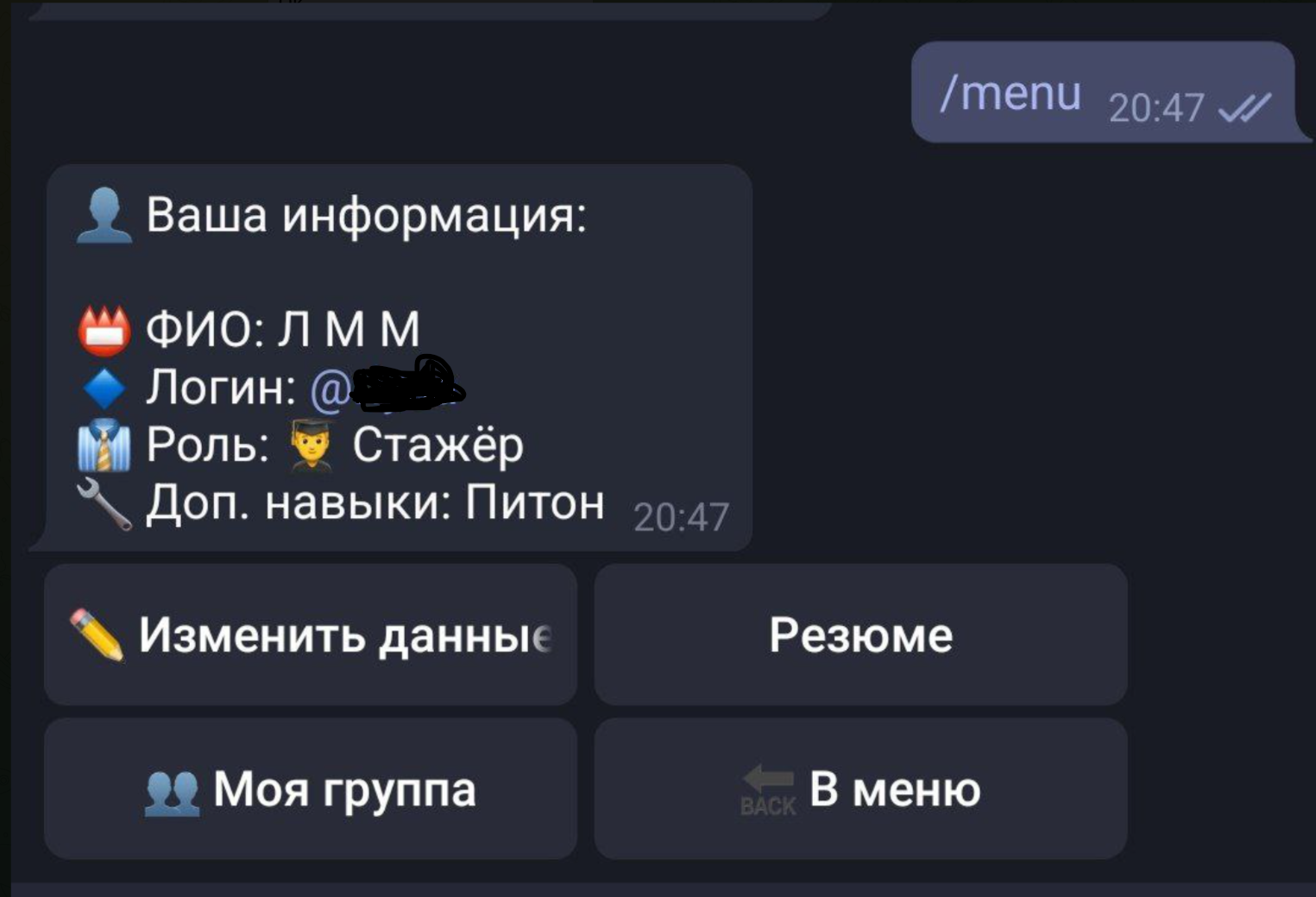
Далее продемонстрирован вывод информации о стажере, выбранном из выпадающего списка. Администратору доступно к просмотру резюме стажера, прикрепленное при регистрации в виде файла, администратор может удалить стажера.

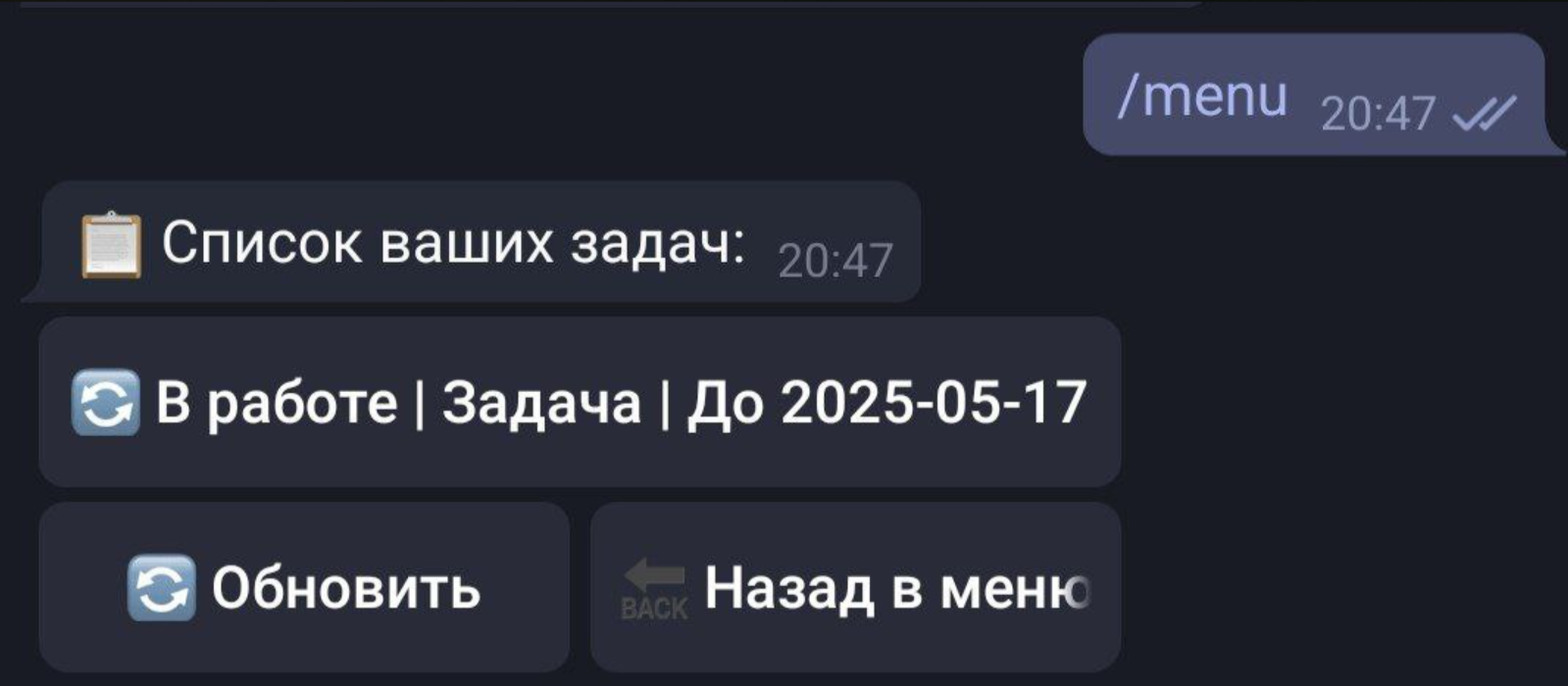


**3.2 Функционал стажера**

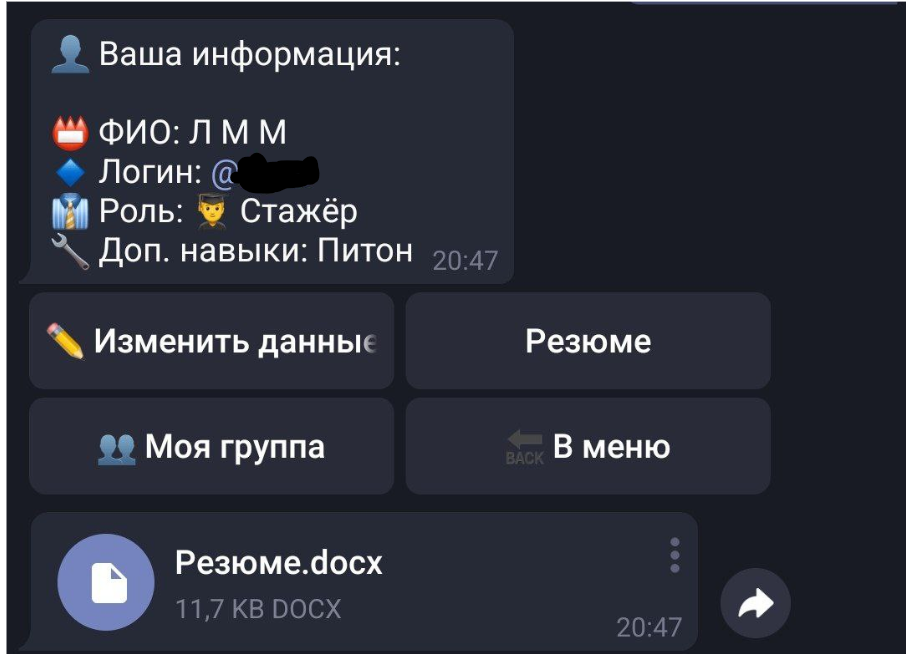
****

Если перейти в раздел “моя информация”, можно увидеть

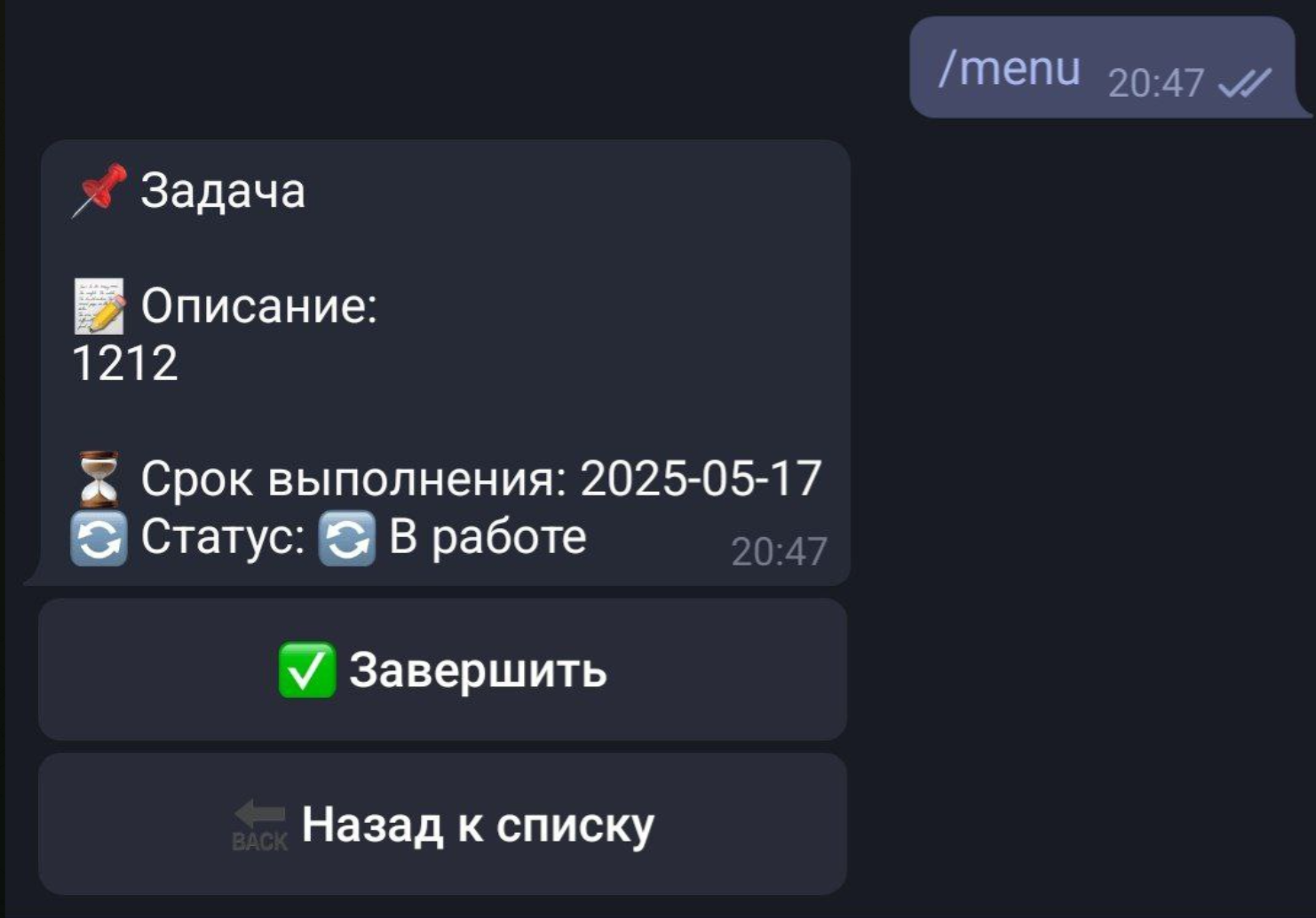
****

****

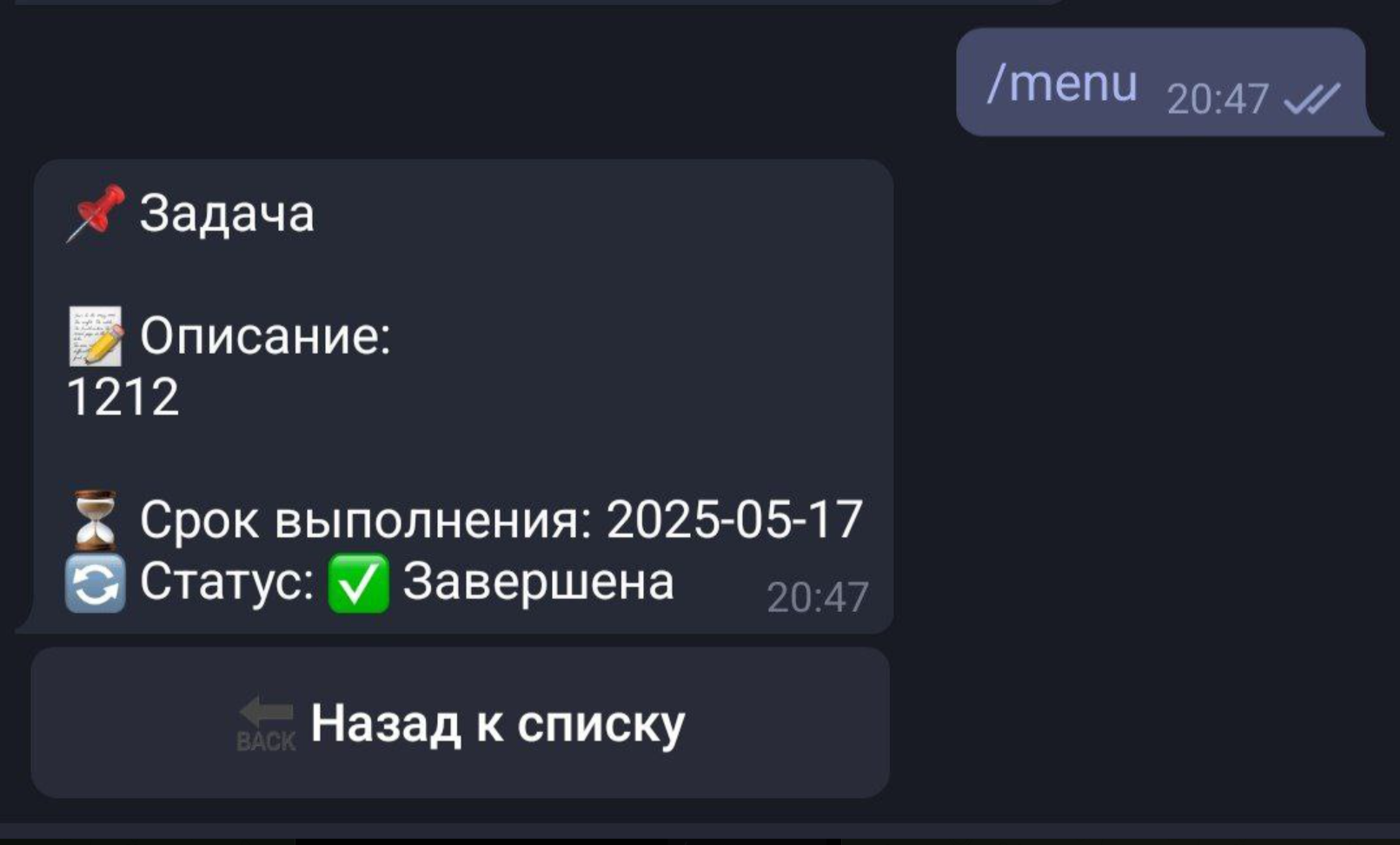
Отображение задач стажера и обновление статуса

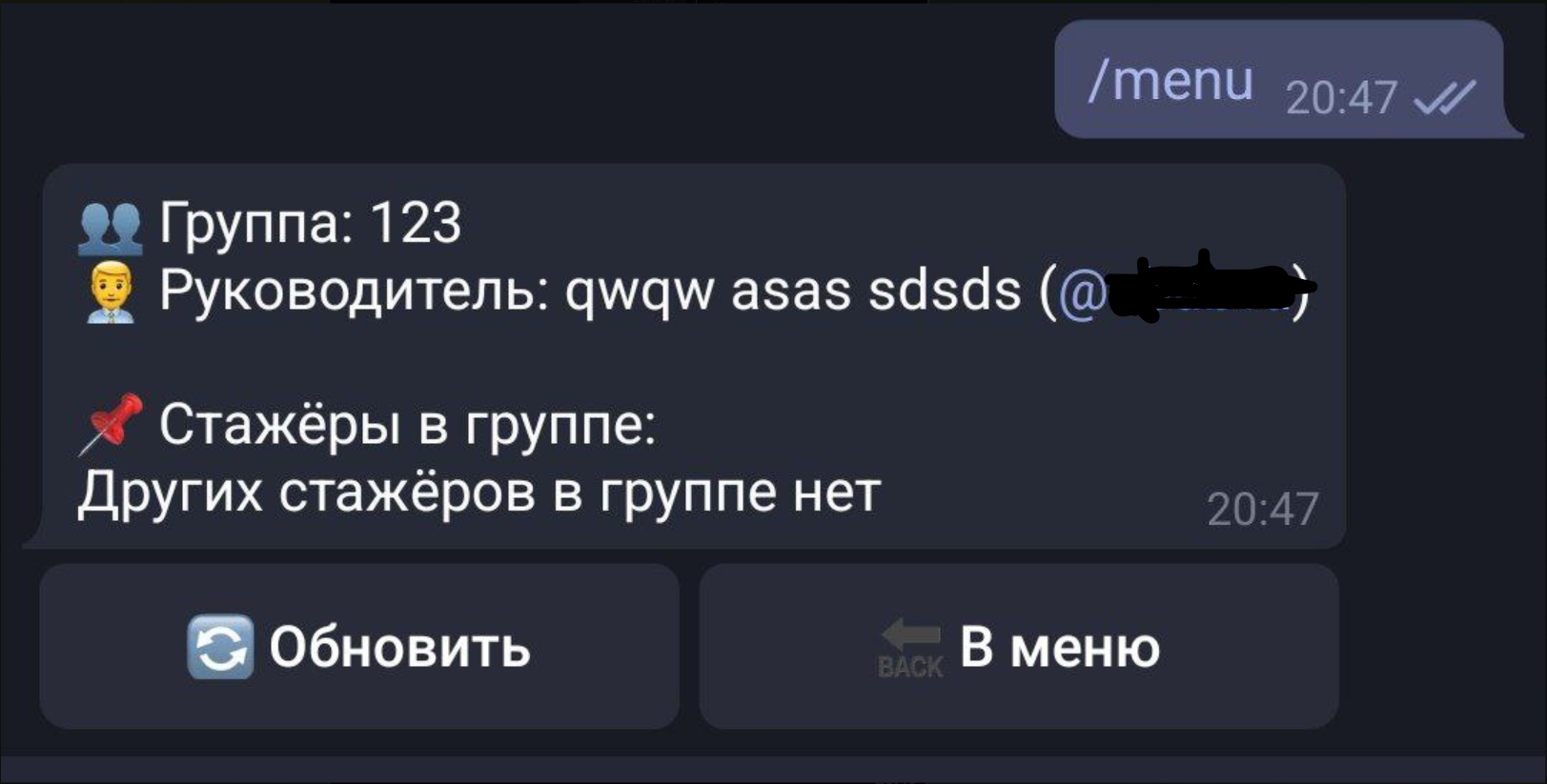


Возможность прикрепления резюме



Возможность завершения задачи



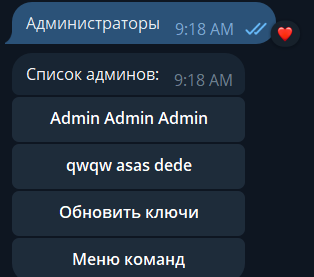


Отображение руководителя и стажеров по команде

**3.3 Функционал суперадмина**

Главным функционалом суперадмина является просмотр пользователей: стажеров и администраторов отдельно.

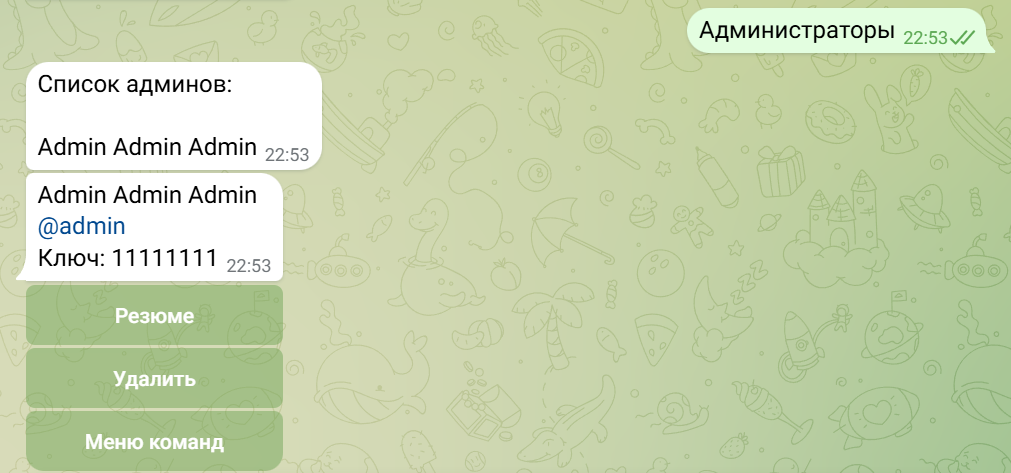




Обновление ключей администраторов происходит по нажатию соответствующей кнопки суперадмином.







Возможность удаления админов

4. Будущее развитие проекта

Проект можно развивать и масштабировать. Одной из идей является добавление подтверждение регистрации администратора главным администратором и стажера администратором, к которому он прикреплен. Также с целью повышения безопасности предлагается добавление функции. которая регулярно обновляла бы ключи администраторов, и сами администраторы по необходимости могли бы их обновлять.

Для упрощения выбора стажеров из списка при создании задачи администратору будет добавлена возможность выбрать группу, при нажатии на кнопку с названием группы будут подсвечены стажеры, состоящие в ней.

Кроме того, планируется переход на полноценную базу данных PSQL, которую планируется развернуть в отдельном контейнере и связать через docker-compose.yml.

* + 1. ТЗ обменной команды

**Введение**

Программа "Конвертер величин" представляет собой desktop-приложение для быстрого и удобного преобразования различных физических величин между единицами измерения. Приложение предназначено для использования в образовательных целях, в быту, а также в профессиональной деятельности, где требуется оперативный пересчет значений между различными системами измерений.

Области применения:

* Учебный процесс в школах и вузах
* Инженерные расчеты
* Научно-исследовательская деятельность
* Повседневное использование для бытовых расчетов

**Основания для разработки**

Разработка программы осуществляется в рамках учебного проекта для демонстрации практических навыков программирования на Python с использованием библиотеки tkinter для создания графического интерфейса.

Исходные документы:

* Задание на практическую работу по программированию
* Рекомендации по разработке пользовательских интерфейсов

**Назначение разработки**

Программа предназначена для автоматизации процесса конвертации физических величин, что позволит:

* Сократить время расчета по сравнению с ручными вычислениями
* Исключить ошибки при пересчете значений между системами измерений
* Обеспечить наглядное представление результатов преобразования
* Предоставить удобный интерфейс для работы с различными категориями величин

**Требования к программе**

**Функциональные требования**

* Конвертация величин в категориях: длина, масса, температура, время, объем, площадь
* Возможность выбора конкретных единиц измерения внутри каждой категории
* Функция обмена выбранных единиц измерения
* Отображение результатов преобразования с точностью до 8 знаков после запятой
* Обработка ошибок ввода и некорректных данных
* Валидация вводимых значений (проверка на отрицательные значения для определенных категорий)
* Очистка всех полей ввода по запросу пользователя

**Требования к надежности**

* Обработка исключительных ситуаций без аварийного завершения программы
* Корректное восстановление после ошибок ввода
* Защита от некорректных операций (деление на ноль, нечисловые значения)
* Стабильная работа при непрерывном использовании

**Условия эксплуатации**

* Операционная система: Windows 7/8/10/11, Linux, MacOS
* Минимальные требования: 512 МБ ОЗУ, 50 МБ свободного места на диске
* Обязательное наличие интерпретатора Python версии 3.6 и выше
* Наличие библиотеки tkinter в составе Python

**Требования к совместимости**

* Совместимость с операционными системами семейства Windows, Linux, MacOS
* Поддержка кодировки UTF-8 для корректного отображения символов
* Независимость от дополнительного программного обеспечения (кроме Python)

**Требования к интерфейсу**

**Основные элементы интерфейса**

* Выпадающий список для выбора категории величин
* Поле ввода исходного значения с возможностью использования точки или запятой как разделителя
* Два выпадающих списка для выбора единиц измерения (исходная и целевая)
* Кнопка обмена единицами измерения (↔)
* Кнопка "Конвертировать" зеленого цвета с белым текстом
* Поле вывода результата с зеленым текстом на светлом фоне
* Область дополнительной информации о результате преобразования
* Кнопка "Очистить все" красного цвета с белым текстом

**Особенности интерфейса**

* Все элементы управления должны быть подписаны на русском языке
* Размер окна фиксированный (500x650 пикселей)
* Шрифты: Arial различных размеров (9-16) и начертаний
* Цветовая схема: сочетание зеленых, серых и белых тонов
* Кнопка конвертации должна быть заметным акцентным элементом
* Поле результата должно визуально отличаться от поля ввода

**Критерии приемки**

* Успешное выполнение 100% тестов
* Отсутствие аварийных завершений программы при любых сценариях использования
* Соответствие времени отклика интерфейса требованиям (не более 0.5 сек)
* Полное соответствие функциональности заявленным требованиям

**Требования к документации**

**Обязательная документация**

* Запрос пользователя с описанием требуемой системы
* Техническое описание программы
* Техническое задание
* Исходный код с соответствующими комментариями

**Порядок контроля и приемки**

**Методы тестирования**

* Функциональное тестирование всех элементов интерфейса
* Тестирование корректности преобразования величин
* Тестирование обработки ошибок и исключительных ситуаций
* Тестирование пользовательского интерфейса на удобство использования

**Приемочные испытания**

* Проверка на соответствие техническому заданию
* Тестирование на различных операционных системах
* Проверка работы при различных разрешениях экрана
* Оценка удобства интерфейса целевой аудиторией

**Этапы и сроки разработки**

1. Проектирование архитектуры приложения – 1 день
2. Разработка графического интерфейса – 2 дня
3. Реализация логики преобразования величин – 3 дня
4. Реализация обработки ошибок и исключений – 1 день
5. Тестирование и отладка программы – 2 дня
6. Написание документации – 1 день

Общий срок разработки: 10 рабочих дней

**Дополнительная документация на программный продукт "Конвертер величин"**

**1. Руководство пользователя**

**Обзор приложения**

"Конвертер величин" — это графическое приложение для конвертации различных физических единиц измерения. Оно поддерживает категории: Длина, Масса, Температура, Время, Объем и Площадь. Приложение позволяет вводить значение, выбирать исходную и целевую единицы, выполнять конвертацию и отображать результат.

**Установка и запуск**

* **Требования:** Python 3.x с установленной библиотекой Tkinter (обычно входит в стандартную поставку Python).
* **Запуск:** Сохраните код в файл main.py и выполните команду python main.py в терминале. Приложение откроется в отдельном окне.

**Интерфейс пользователя**

* **Категория:** Выберите категорию из выпадающего списка (например, "Длина").
* **Исходное значение:** Введите числовое значение в поле ввода. Поддерживаются целые и дробные числа (используйте точку или запятую как разделитель).
* **Из:** Выберите исходную единицу измерения.
* **В:** Выберите целевую единицу измерения.
* **Кнопка "↔":** Обменяет исходную и целевую единицы.
* **Кнопка "Конвертировать":** Выполнит расчет и отобразит результат.
* **Раздел результата:** Показывает конвертированное значение и дополнительную информацию (например, "10 метры = 1000 сантиметры").
* **Кнопка "Очистить все":** Сбросит все поля.

**Примеры использования**

1. **Конвертация длины:** Выберите "Длина", введите "1" в поле значения, "километры" в "Из", "метры" в "В". Нажмите "Конвертировать". Результат: "1000".
2. **Конвертация температуры:** Выберите "Температура", введите "0", "°C" в "Из", "°F" в "В". Результат: "32".
3. **Обмен единиц:** Нажмите "↔" для быстрого переключения между "Из" и "В".
4. **Обработка ошибок:** Если значение не число, приложение покажет сообщение "Введите корректное число". Для отрицательных значений в "Масса" или "Длина" — "Значение не может быть отрицательным".

**Советы**

* Используйте клавишу Enter для конвертации после ввода значения.
* Результат форматируется для удобства: с разным количеством знаков после запятой в зависимости от величины.
* Для температуры учитывается абсолютный ноль (в Кельвинах результат не может быть отрицательным).

**2. Описание архитектуры системы**

**Общая структура**

Приложение построено на основе объектно-ориентированного подхода. Основной класс UnitConverter инкапсулирует всю логику: инициализацию данных, настройку UI и обработку конвертации. Главная функция main() создает корневое окно Tkinter и экземпляр класса.

* **Модули и библиотеки:**
  + tkinter и ttk для графического интерфейса.
  + messagebox для вывода ошибок.
* **Данные:**
  + Словарь units хранит списки единиц по категориям.
  + Словарь conversion\_factors содержит коэффициенты конвертации (относительно базовой единицы, первой в списке).
* **Методы:**
  + \_\_init\_\_: Инициализирует окно, данные и UI.
  + setup\_ui: Настраивает элементы интерфейса (фреймы, метки, комбобоксы, кнопки).
  + update\_units: Обновляет списки единиц при смене категории.
  + swap\_units: Обменивает единицы "Из" и "В".
  + clear\_fields: Очищает поля ввода и результата.
  + convert\_temperature: Специальная функция для температуры (конвертация через Цельсий).
  + convert: Основная логика конвертации, включая проверки и форматирование.

**Архитектурные принципы**

* **Модульность:** UI отделен от логики конвертации.
* **Событийно-ориентированный подход:** Используются привязки событий (e.g., <<ComboboxSelected>>, <Return>).
* **Обработка исключений:** Ловит ValueError (некорректное число), ZeroDivisionError и другие.
* **Базовая единица:** Для большинства категорий конвертация идет через базовую единицу (e.g., метры для длины).
* **Особенности:** Температура обрабатывается отдельно из-за нелинейных формул. Для объема реализована специальная логика (см. ошибки ниже).

**3. Схемы и диаграммы**

**Диаграмма классов (UML-в текстовом формате)**

|  |
| --- |
| UnitConverter |
| Атрибуты:  - root: Tk  - units: dict {категория: список единиц}  - conversion\_factors: dict {категория: коэффициенты}  - category\_var: StringVar  - from\_var: StringVar  - to\_var: StringVar  - input\_var: StringVar  - result\_var: StringVar  - result\_info\_var: StringVar |
| Методы:  + \_\_init\_\_(root): Инициализация окна и UI  + setup\_ui(): Создание интерфейса  + update\_units(event): Обновление единиц  + swap\_units(): Обмен единиц "Из" и "В"  + clear\_fields(): Очистка полей ввода/вывода  + convert\_temperature(value, from, to): Конвертация температуры  + convert(): Выполнение конвертации |

**Схема интерфейса (ASCII-арт)**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Содержимое, созданное искусственным интеллектом, может быть неверным.

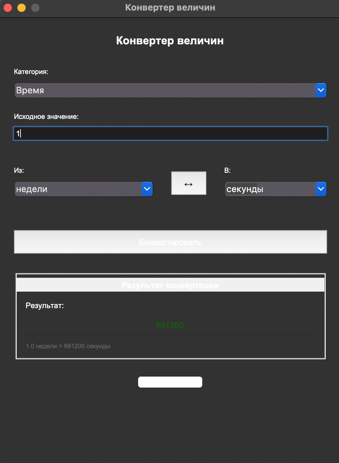
Рисунок 1 - Интерфейс

**Диаграмма потока данных для конвертации**

1. Пользователь выбирает категорию → update\_units обновляет комбобоксы.
2. Ввод значения + единицы → Нажатие "Конвертировать" → convert:
   * Проверки (значение, категория, единицы).
   * Если Температура: convert\_temperature.
   * Иначе: value \* factor\_from / factor\_to.
   * Форматирование → Отображение в result\_var и result\_info\_var.

Отчет по тестированию

Ошибка 1:



1. Идентификатор. НР-001

2. Название. Проверка корректности расчета по заданным параметрам.

3. Описание. Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений.

4. Предварительные условия: а) пользователь вводит корректные значения соответствующие условиям для конвертации(неотрицательные);

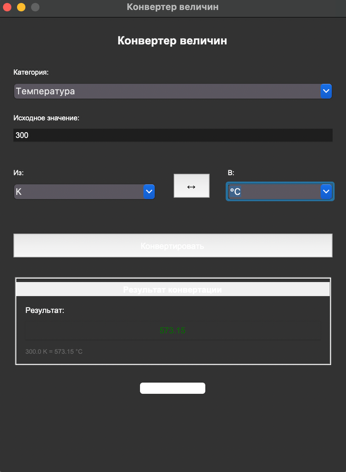
5. Шаги выполнения: а) в поле «Категории» выбрана интересующая категория для перевода; б) в поле «Исходное значение» введено корректное число; в) в поле «Из» выбрана единица измерения г) в поле «В» выбрана единица измерения д) нажать кнопку «Конвертировать».

6. Ожидаемый результат. В поле «результат конвертации» выводиться корректный, правильный математически, перевод

7. Фактический результат. В поле «результат конвертации» выводиться результат с ошибкой в расчетах

8. Статус. Failed

Ошибка 2:



1. Идентификатор. НР-002

2. Название. Проверка корректности расчета по заданным параметрам.

3. Описание. Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений.

4. Предварительные условия: а) пользователь вводит корректные значения соответствующие условиям для конвертации;

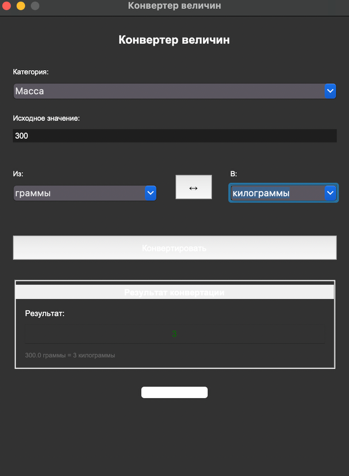
5. Шаги выполнения: а) в поле «Категории» выбрана интересующая категория для перевода; б) в поле «Исходное значение» введено корректное число; в) в поле «Из» выбрана единица измерения г) в поле «В» выбрана единица измерения д) нажать кнопку «Конвертировать».

6. Ожидаемый результат. В поле «результат конвертации» выводиться корректный, правильный математически, перевод

7. Фактический результат. В поле «результат конвертации» выводиться результат с ошибкой в расчетах

8. Статус. Failed

Ошибка 3:



1. Идентификатор. НР-003

2. Название. Проверка корректности расчета по заданным параметрам.

3. Описание. Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений.

4. Предварительные условия: а) пользователь вводит корректные значения соответствующие условиям для конвертации(неотрицательные);

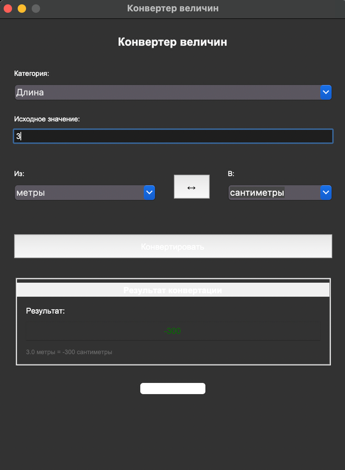
5. Шаги выполнения: а) в поле «Категории» выбрана интересующая категория для перевода; б) в поле «Исходное значение» введено корректное число; в) в поле «Из» выбрана единица измерения г) в поле «В» выбрана единица измерения д) нажать кнопку «Конвертировать».

6. Ожидаемый результат. В поле «результат конвертации» выводиться корректный, правильный математически, перевод

7. Фактический результат. В поле «результат конвертации» выводиться результат с ошибкой в расчетах

8. Статус. Failed

Ошибка 4:



1. Идентификатор. НР-004

2. Название. Проверка корректности расчета по заданным параметрам.

3. Описание. Убедиться, что пользователь увидит корректный результат конвертации при вводе корректных значений.

4. Предварительные условия: а) пользователь вводит корректные значения соответствующие условиям для конвертации (неотрицательные);

5. Шаги выполнения: а) в поле «Категории» выбрана интересующая категория для перевода; б) в поле «Исходное значение» введено корректное число; в) в поле «Из» выбрана единица измерения г) в поле «В» выбрана единица измерения д) нажать кнопку «Конвертировать».

6. Ожидаемый результат. В поле «результат конвертации» выводиться корректный, правильный математически, перевод

7. Фактический результат. В поле «результат конвертации» выводиться результат с ошибкой в рассчетах

8. Статус. Failed