**Государственное бюджетное общеобразовательное учреждение города Москвы " Школа № 1506"**

**АДМИНИСТРАТИВНЫЙ ПОМОЩНИК**

Работу выполнил:

Ученик 10 класса

ГБОУ Школы №1506

Химин Максим Владимирович

Руководитель:

Учитель информатики

ГБОУ Школы №1506

Игнатьев Андрей Александрович

**г. Москва, 2025**

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc190958110)

[ЦЕЛЬ И ЗАДАЧИ РАБОТЫ 4](#_Toc190958111)

[МЕТОДИКА ВЫПОЛНЕНИЯ ПРОЕКТА: 5](#_Toc190958112)

[РЕЗУЛЬТАТ 8](#_Toc190958113)

[ОПИСАНИЕ ПРОДУКТА 9](#_Toc190958114)

# ВВЕДЕНИЕ

Актуальность для школы заключается в необходимости автоматизации и упрощения административных и учебных процессов, таких как взаимодействие с учителями и родителями, управление расписанием, а также обработка различных запросов и заявок. Внедрение помощника позволит значительно повысить эффективность работы школы, снизить нагрузку на административный персонал и улучшить организацию образовательного процесса.

# ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель: Автоматизация взаимодействия между администрацией школы, учителями и учащимися, а также их родителями на основе Telegram-бота.

Задачи:

* Получить токены для доступа к платформе Telegram;
* Реализовать модуль создания чата/канала;
* Реализовать модуль взаимодействия с базой данных (хранение информации о пользователях, группах);
* Реализовать модуль модерации сообщений в чатах на основе стоп-листа
* Реализовать модуль по созданию тестов и автоматизированную проверку их результатов.

# МЕТОДИКА ВЫПОЛНЕНИЯ ПРОЕКТА:

Первый этап: «Опрос целевой аудитории».

Сроки этапа: «1» ноября 2024 г. – «15» ноября 2024 г.

Описание этапа: провести опрос среди учителей, учащихся, чтобы определить наиболее актуальные потребности и задачи, необходимые для повышения эффективности образовательного процесса. Полученные результаты позволят учесть мнения всех участников школьного сообщества и адаптировать функционал продукта к реальным запросам.

Результаты этапа: в опросе приняли участие ученики 10-11 классов и учителя. Результаты показали убедительное единство мнений: 90% участников поддержали идею внедрения такого бота, отмечая его полезность, в то время как лишь 10% посчитали его использование излишним. Эти данные подчеркивают высокую заинтересованность и готовность к нововведениям среди школьников и педагогов.

Второй этап: «Поиск готовых решений».

Сроки этапа: «16» ноября 2024 г. – «30» ноября 2024 г.

Описание этапа: поиск информации об имеющихся в открытом доступе продуктов, направленных на решение аналогичных проблемы.

Результаты этапа: исследование показало, что готовых решений проблемы на текущий момент не имеется.

Третий этап: «Разработка архитектуры проекта».

Сроки этапа: «1» декабря 2024 г. – «13» декабря 2024 г.

Описание этапа: создание схемы проекта на уровне абстракции, определение её основных компонентов, их функций и взаимодействия для обеспечения работы.

Результаты этапа: результатом разработки проекта становится продуманная модель системы, описывающая её компоненты, связи и выбранные технологии.

Четвертый этап: «Изучения языка программирования Python и компьютерного зрения».

Сроки этапа: «14» декабря 2024 г. – «1» января 2024 г.

Описание этапа: Необходимо освоить базовые элементы языка программирования Python для ускорения разработки продукта, а также углубиться в изучение компьютерного зрения, включая методы обработки изображений и автоматического анализа визуальных данных.

Используемое ПО: PyCharm.

Используемые технологии: Python, OpenCV.

Пятый этап: «Создание схемы базы данных SQLite»

Сроки этапа: «18» декабря 2024 г. – «26» декабря 2024 г.

Описание этапа: создание схемы базы данных SQLite включает в себя проектирование структуры данных, определение таблиц с полями и типами данных, установление связей между таблицами через внешние ключи и написание SQL-запросов для создания таблиц, после чего проводится тестирование и проверка правильности работы схемы.

Результаты этапа: спроектирована структура базы данных, включающая таблицы с необходимыми полями и типами данных, а также зависимостями.

Используемое ПО: PyCharm.

Используемые технологии: SQL, Python.

Шестой этап: «Создание функционала автоматизированной проверки результатов теста ученика на основе компьютерного зрения»:

Сроки этапа: «26» декабря 2024 г. – «13» января 2025 г.

Описание этапа: на основе отправленной фотографии теста ученика, бот распознает ответы, проводит их проверку и предоставляет учителю детальный отчет для последующей оценки работы.

Результаты этапа: разработан функционал автоматизированной проверки тестов на изображении и формирования детального отчета для учителя.

Используемое ПО: PyCharm.

Используемые технологии: Python, OpenCV.

Седьмой этап: «Реализация пользователя-бота (user-bot) на языке Python с использованием библиотеки pyrogram»:

Сроки этапа: «14» января 2025 г. – «9» февраля 2025 г.

Описание этапа: на основе изученных данных на предыдущих этапах необходимо создать проект на языке Python.

Результаты этапа: написана рабочая версия продукта, требующая дальнейшей оптимизации, тестирования и откладки.

Используемое ПО: PyCharm

Используемые технологии: Python, SQL.

Восьмой этап: «Тестирование и отладка бота»:

Сроки этапа: «10» февраля 2025 г. – «18» февраля 2025 г.

Описание этапа: найти недостатки, которые могут стать препятствием для стабильной работы и масштабируемости системы на глобальном уровне, а также для её использования в различных условиях и среди широкого круга пользователей.

Результаты этапа: по итогам тестирования выявленные проблемы были устранены, а также проведена оптимизация.

# РЕЗУЛЬТАТ

Итогом проекта является рабочая версия пользователя-бота. Полученный продукт обладает функционалом для автоматизации ряда административных и педагогических задач и работает на платформе Telegram, обеспечивая удобный интерфейс для пользователей. Бот интегрирован с базой данных, что позволяет оперативно обновлять информацию и взаимодействовать с учителями, учениками и родителями.

# ОПИСАНИЕ ПРОДУКТА

Наш продукт разработан для того, чтобы сократить затраты времени и усилий пользователей на выполнение повседневных школьных задач. Для удобного доступа к системе каждому пользователю создаётся личный аккаунт, связанный с его Telegram-аккаунтом. Это позволяет сохранять и обрабатывать всю информацию, которую пользователь вносит в систему, обеспечивая её доступность в любой момент. Чтобы начать работу, пользователю необходимо пройти регистрацию, как показано на рисунке 1. Для этого нужно написать боту команду “/начать”, после чего появится возможность выбрать роль. Такой процесс регистрации обеспечивает персонализированный подход, упрощает взаимодействие с системой и делает её использование максимально комфортным.

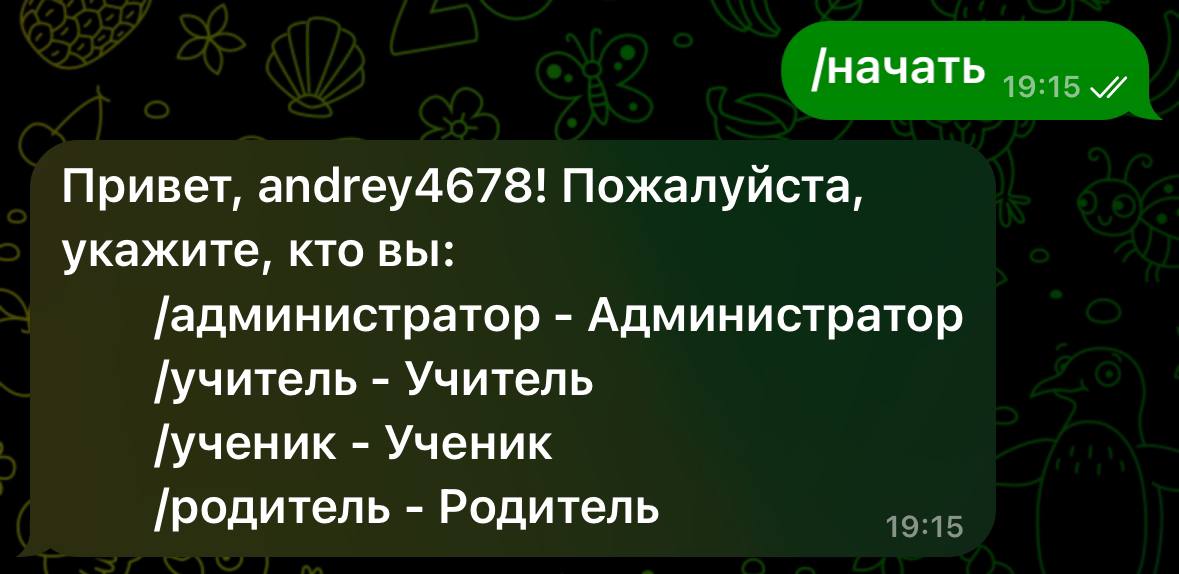


Рисунок 1. Приветственное сообщение

После того как вы выберете свою роль в системе, вам будет предложено ввести персональные данные: фамилию, имя, отчество (при наличии) и номер телефона. Эти данные необходимы для создания вашего профиля в системе. После того как вы введёте всю информацию, программа отобразит её для проверки. Вы сможете убедиться, что все данные введены правильно, и при необходимости внести изменения. Этот процесс помогает гарантировать точность и корректность информации, что исключает возможные ошибки при её дальнейшей обработке в системе. После этого необходимо дождаться активации вашего профиля, когда бот подтвердит принятие ваших данных и завершит процесс регистрации. Описанные действия продемонстрированы на рисунке 2.

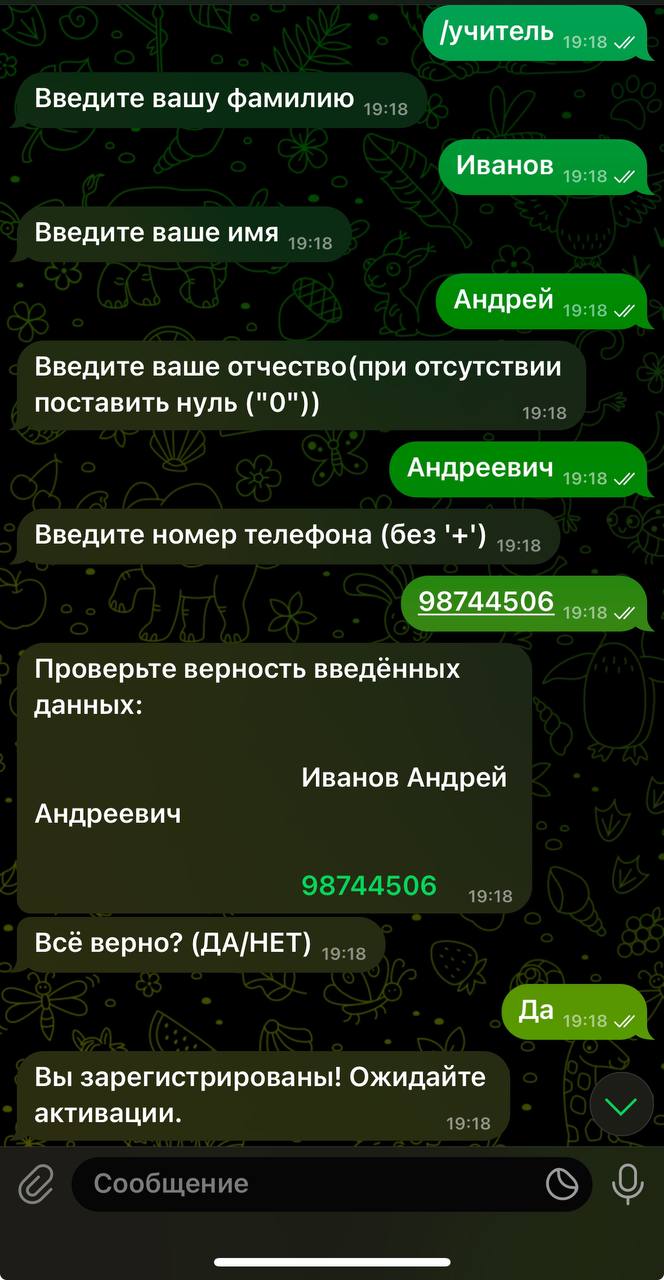


Рисунок 2. Ввод регистрационных данных

Когда боту поступает новая заявка, он получает все введённые данные пользователя: фамилию, имя, отчество (при наличии) и номер телефона. Данный функционал изображён на рисунке 3. Вы как администратор или пользователь с соответствующими правами сможете просмотреть эти данные и решить, принимать или отклонить заявку. В зависимости от вашего выбора заявка будет либо одобрена, либо отклонена, что позволит продолжить процесс регистрации или предотвратить дальнейшее взаимодействие с системой для этого пользователя.

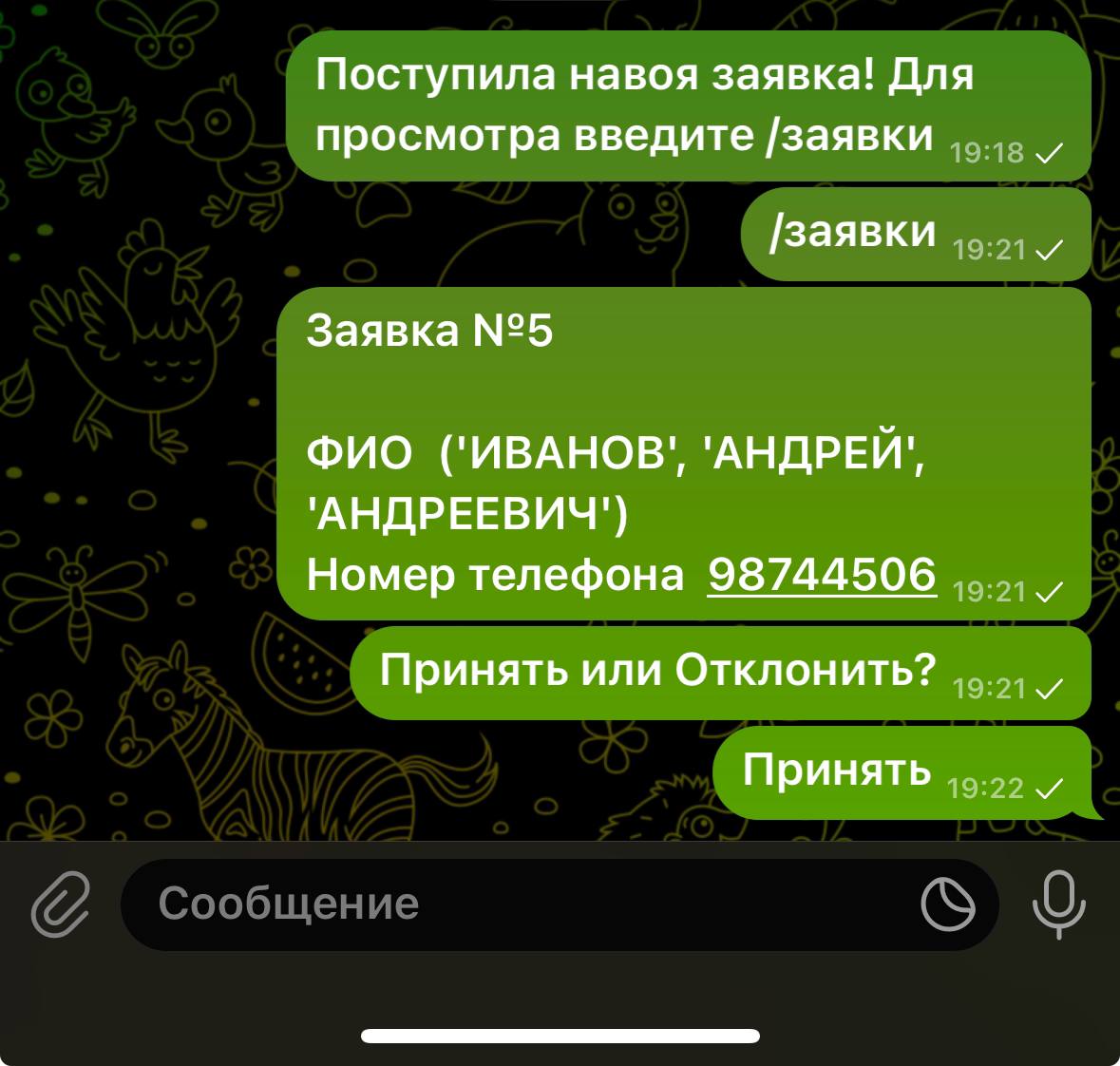


Рисунок 3. Подтверждение информации нового пользователя

После того как вашу заявку приняли и профиль был успешно активирован, вы получаете доступ к функционалу, соответствующему вашей роли в системе. Сообщение об активации учётной записи продемонстрировано на рисунке 4. Теперь вы сможете выполнять все функции, которые предусмотрены для вашей роли, будь то управление данными, доступ к определённым разделам или выполнение других задач, в зависимости от вашего уровня доступа.

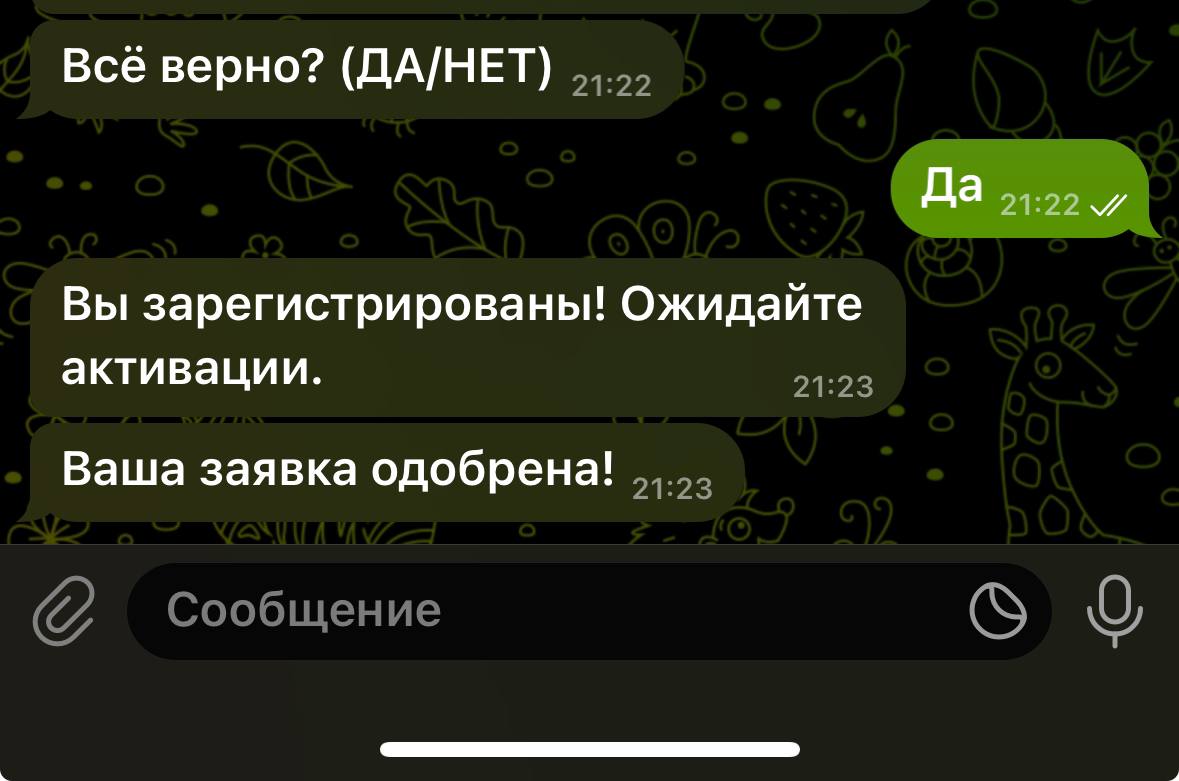


Рисунок 4. Информирование о результатах проверки данных

Учитель может зарегистрировать свой класс в боте, после чего он получит уникальный код класса. Механизм получения кодовой последовательности класса проиллюстрирован на рисунке 5.

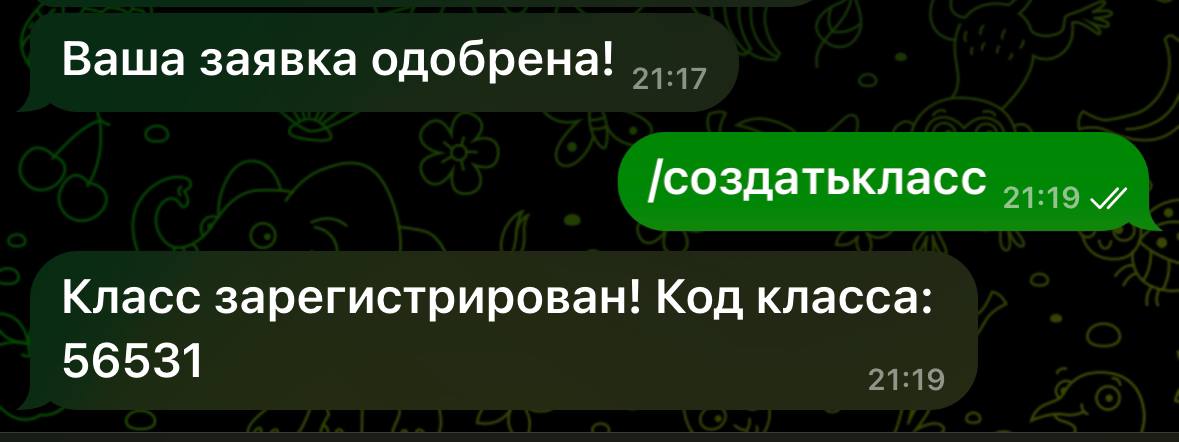


Рисунок 5. Создание нового класса и получение его кода

Этот код будет использоваться учениками для присоединения к классу, что подтверждается рисунком 6. Ученики смогут ввести полученный код в боте, чтобы стать частью соответствующего класса, и таким образом, учитель будет иметь возможность управлять классом и отслеживать прогресс учеников.

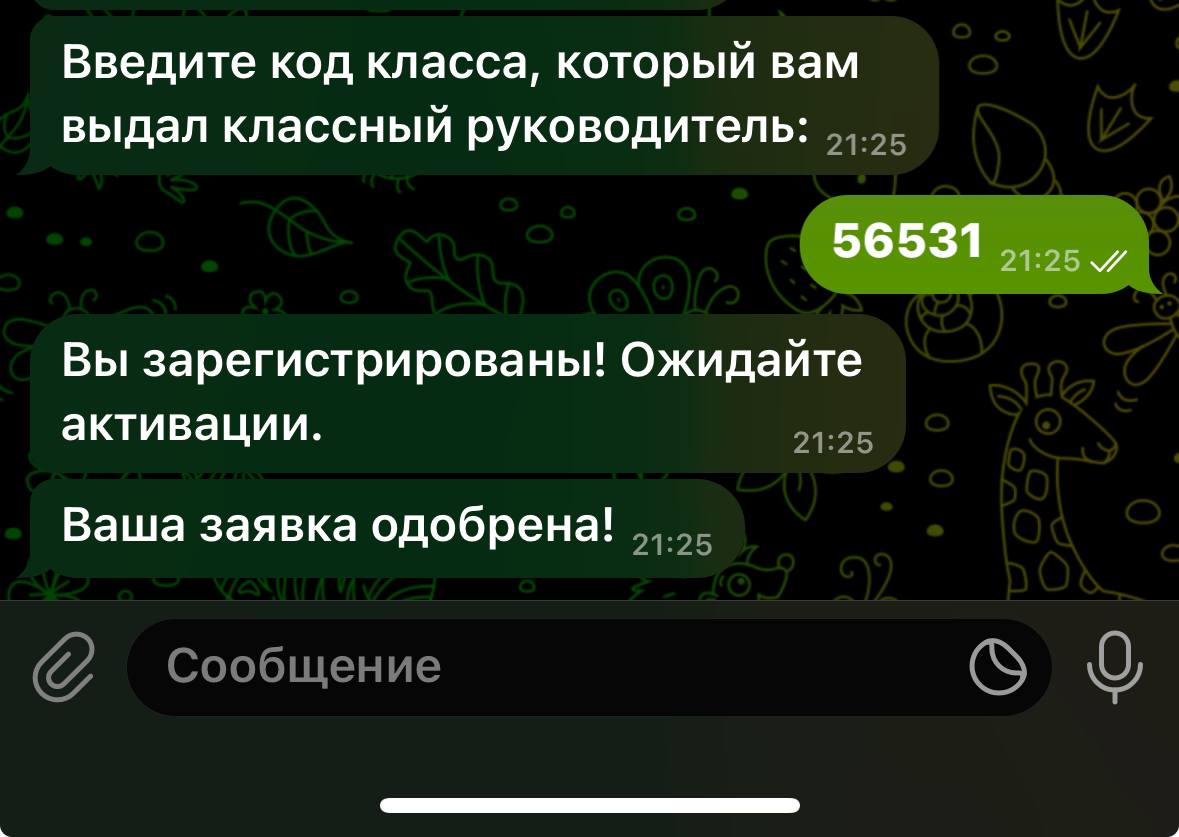


Рисунок 6. Пример регистрации ученика на этапе ввода кода класса, полученного выше



Рисунок 7. Создание шаблона для теста

У учителя есть уникальная возможность воспользоваться специальной функцией для создания шаблона теста, которая значительно упрощает подготовку материалов для проверки знаний учащихся. Чтобы начать процесс формирования шаблона, необходимо отправить боту команду “/создатьшаблонтеста”. После этого система предложит пользователю указать ключевые параметры будущего теста: количество вопросов, которое он планирует включить, число вариантов ответов для каждого вопроса, а также желаемое количество колонок, в которых будет организована структура документа, как показано на рисунке 7.

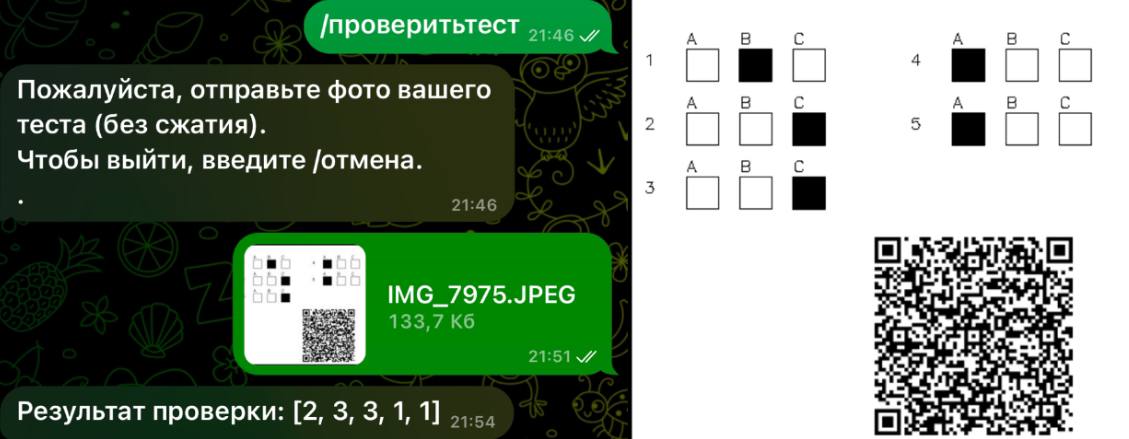


Рисунок 8. Отправка эталонного теста (теста учителя)



Рисунок 9. Проверка теста ученика 1



Рисунок 10. Проверка теста ученика 2

Кроме функции создания шаблона теста, учитель также может воспользоваться удобным инструментом для автоматической проверки выполненных работ. Для этого предусмотрена специальная функция, которая значительно упрощает процесс оценивания и минимизирует вероятность ошибок при ручной проверке. Чтобы воспользоваться этой возможностью, преподавателю необходимо отправить команду “/проверитьтест”. После этого система предложит учителю загрузить фотографию заполненного теста с правильными ответами. Важно, чтобы изображение было отправлено без сжатия, чтобы сохранить максимальную четкость и детализацию, что обеспечит корректное распознавание всех элементов теста. После получения эталонного варианта система зафиксирует правильные ответы и предложит учителю отправить аналогичным образом (без сжатия) фотографию работы ученика для дальнейшего анализа. Как только оба изображения будут загружены, бот выполнит детальное сравнение ответов учителя и ученика, используя современные алгоритмы анализа данных. В результате проверки бот предоставит подробный отчет, включающий общее количество заданий, выполненное учеником, количество верных ответов. Кроме того, система выделит номера заданий, в которых были допущены ошибки. Механизм проверки тестовых работ изображён на рисунках 8, 9, 10.

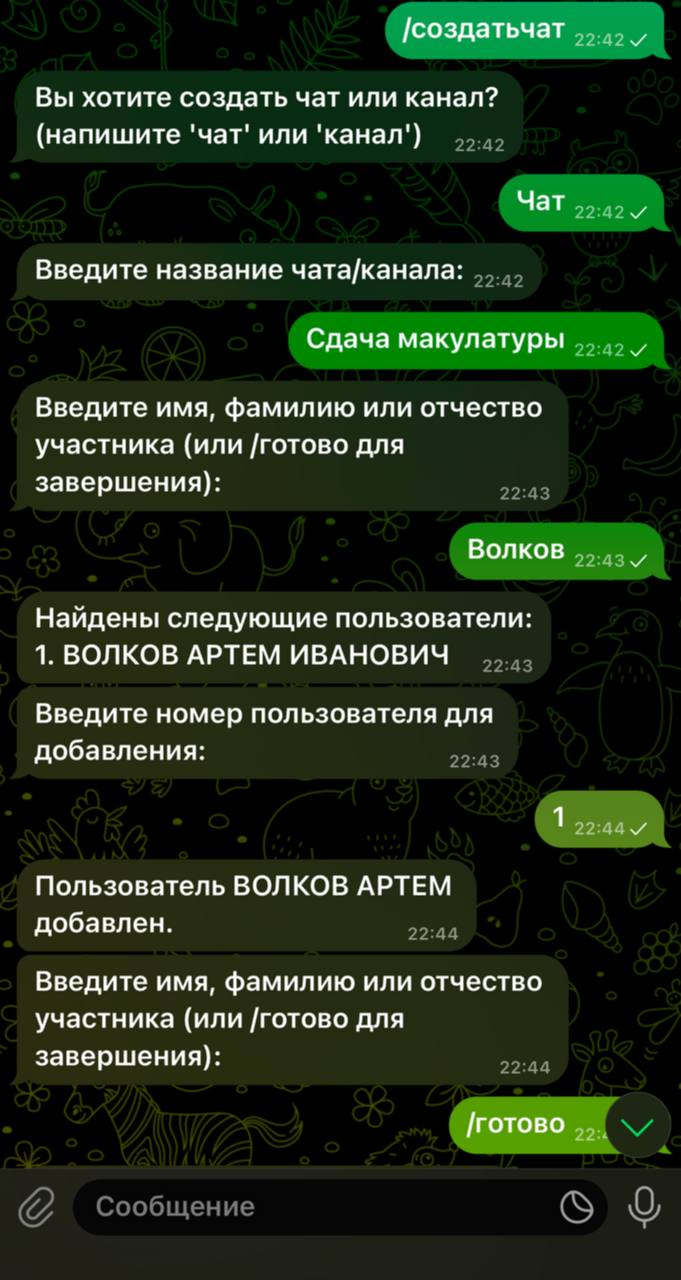


Рисунок 11. Создание чата

Каждый пользователь имеет возможность создать чат, для этого необходимо ввести имя, фамилию или отчество участника, который уже зарегистрирован в боте. После успешного поиска и добавления участника можно завершить создание чата. Пример функционала продемонстрирован на рисунке 11.

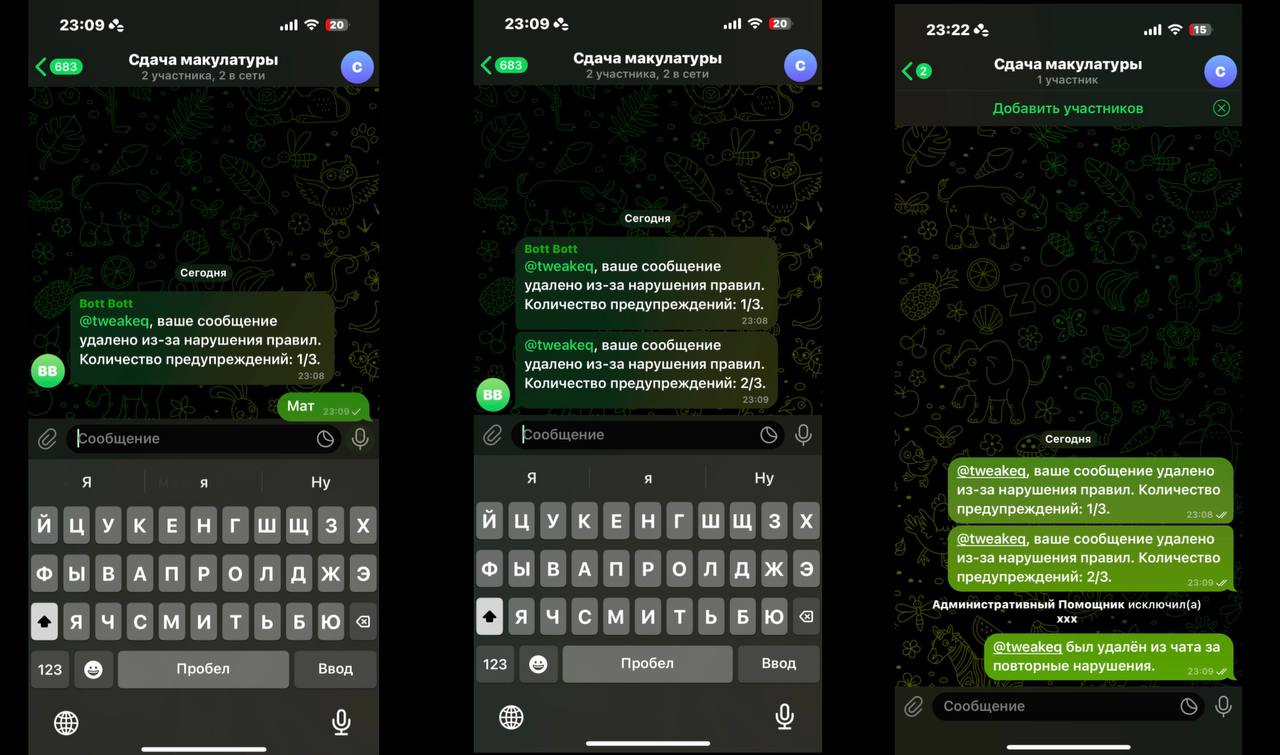


Рисунок 12. Фильтрация нежелательных слов в чате: стоп-лист

В рамках функционала чата внедрен механизм фильтрации нежелательного контента, основанный на использовании стоп-листа. Этот список содержит слова, которые запрещены к использованию в переписке, чтобы обеспечить общение в уважительной и корректной манере. В частности, система блокирует нецензурные выражения, в нашем случае слово «мат», автоматически удаляя его. Если пользователь попытается отправить сообщение с запрещенным словом, он получит уведомление о нарушении правил общения. После трех таких предупреждений участник будет автоматически удален из группы. Весь функционал продемонстрирован на рисунке 12.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Шумилина М.А, Коробко А.В. Разработка чат-бота на языке программирования Python в мессенджере «Telegram» // URL: <https://cyberleninka.ru/article/n/razrabotka-chat-bota-na-yazyke-programmirovaniya-python-v-messendzhere-telegram/viewer> (дата обращения: 12 декабря 2024)
2. Бейдер Д. Чистый Python. Тонкости программирования для профи. — СПб.: Питер, 2018. 288 с.