УДК 004

Разработка чат-бота на Python с возможностью сохранения данных пользователей в базу данных SQL

Зонов Николай Александрович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Научный руководитель:
Лучанинов Дмитрий Васильевич
Приамурский государственный университет имени Шолом-Алейхема
Старший преподаватель

Аннотация
В работе рассматривается задача по проектированию и разработке чат-бота в мессенджере telegram, с возможностью сохранения данных пользователей в базу данных SQL. Реализация проекта выполняется на языке Python. Описан алгоритм создания чат-бота telegram на языке Python.
Ключевые слова: Чат-бот, мессенджер, telegram, база данных, SQL, разработка.

Development of a chatbot in Python with the ability to save user data to SQL database

Zonov Nikolay Alexandrovich
Sholom-Aleichem Priamursky State University
Student

Scientific adviser:
Luchaninov Dmitry Vasilievich
Sholom-Aleichem Priamursky State University
Senior lector

Abstract
The work considers the task of designing and developing a chat bot in the telegram messenger, with the ability to save user data to a SQL database. The project is implemented in Python. The algorithm for creating a telegram chat bot in Python is described.
Keywords: Chatbot, messenger, telegram, database, SQL, development.

1. Введение

1.1 Актуальность

В настоящее время чат-боты как современные инструменты коммуникаций стали широко использоваться во многих сферах

жизнедеятельности человека с различной целью, такие как предоставление пользователю нужной информации или сбор данных, отправляя запрограммированный ответ на пользовательское сообщение или выбранную функцию.

В силу того, что контакт с чат-ботами является не только интересным занятие, но и полезным для человека, с точки зрения получения новой информации и решения потребительских задач, интернет-аудитория приняла их появление очень доброжелательно.

Многие организации в свою очередь увидели коммерческую выгоду в интересе аудитории к чат-ботам и стали разрабатывать собственные, используя их для привлечения клиентов, оказание консультаций по интересующих вопросам или товарам, выяснение потребностей клиентов, формирование лояльности к компании, вовлечение потребителя в рекламную коммуникацию, что повышает ее эффективность, автоматическая рассылка рекламы и многое другое.

Чат-бот становится мощным инструментом продвижения как для крупных брендов, так и для новичков.

Таким образом, исходя из современного уровня развития коммуникационных средств, можно сделать вывод о перспективности использования чат-ботов не только в бытовой практике, в популярных мессенджерах, но и в сферах экономики, образования, предоставления услуг и многих других областях деятельности человека.

1.3 Цель исследования

Целью данного исследования является создание функционала сохранения информации от пользователей чат-бота в базу данных, с помощью программного языка Python.

2. Инструменты разработки

Для написания кода использовался Atom – это многофункциональный текстовый редактор, с приятным интерфейсом и возможностью установки огромного количества расширений, позволяющих приспособить его практически к любому языку программирования. Однако для более полноценной работы – валидации кода, проверки синтаксиса, базовой отладки и снипетов, как и в большинстве редакторов кода, необходимо подключение дополнительных пакетов, но их установка не составляет труда.

Для Python рекомендуются следующие расширения:

1. Script – запускает код на Python.
2. Linter и Linter Python Pep8 – отображение синтаксических ошибок.
3. Atom Python Debugger – отладка и тестирование кода.

Библиотека pyTelegramBotAPI (telebot) на Python. Необходима для взаимодействия с Telegram Bot Api (представляет из себя HTTP-интерфейс для работы с ботами в Telegram.)

Стандартная библиотека sqlite3 на Python от версии 2.5, для создания базы данных SQLite без необходимости скачивания дополнительных инструментов.

3. Проектирование

С использованием перечисленных инструментов, необходимо разработать чат-бота для Telegram с проверкой созданной ранее базы данных, при запуске файла с кодом и созданием базы, если она отсутствует.

Структура базы данных будет состоять из двух таблиц, messages куда будут сохраняться все сообщения от зарегистрированных пользователей и таблица users для регистрации новых пользователей. Также необходимо реализовать вывод сообщения если пользователь был зарегистрирован ранее.

Предусмотреть возможность администратора чат-бота просмотреть сообщения от любого зарегистрированного пользователя.

3. Разработка

Предварительно необходимо зарегистрировать бота в мессенджере Telegram и получить токен (ключ) для работы с Telegram API. Регистрация происходит с помощью бота @BotFather

Токен — это секретный ключ-идентификатор бота, используется при обращении к Telegram API для идентификации бота.

В поисковой строке необходимо вести @BotFather и выбрать вариант поиска с соответствующим именем. После выбора бота и нажатия команды /start, бот покажет полный список команд для регистрации, настройки и удаления бота (Рис. 2).

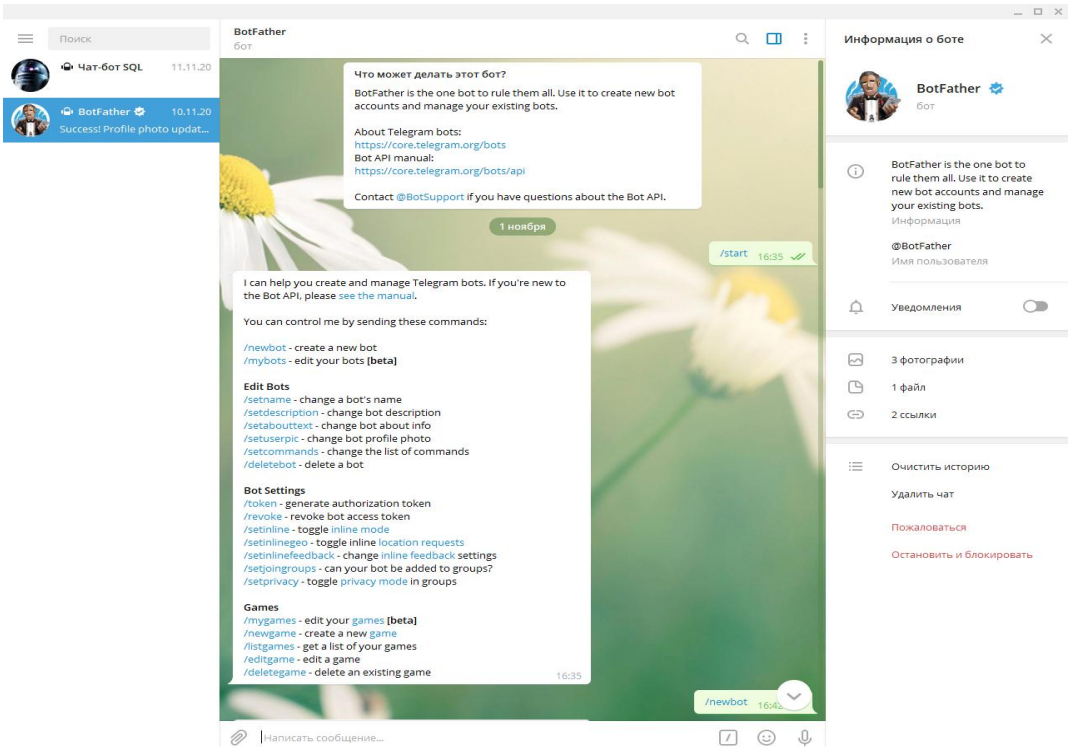

Рисунок 2. Список доступных команд

Далее нам понадобится указать название бота, задать адрес ссылки, описание, установить картинку бота и получить токен.

Для создания бота введите в чат с BotFather команду /newbot. Пользователя попросят ввести название для нового бота. Имя будет отображаться в заголовке и в информации о боте.

После того, как задано имя бота, нужно указать его сокращенное название для ссылок. Оно должно обязательно содержат приставку «bot» на конце. Например: «Test_Bot».

Если все выполнено верно, бот будет зарегистрирован в Телеграм и BotFather выдаст токен следующем формате:

API:764645301:AAGdRMMi_bF67lCkJjA0DKQNOwoATJQMWXk

Пример выполнения команд (Рис. 3).

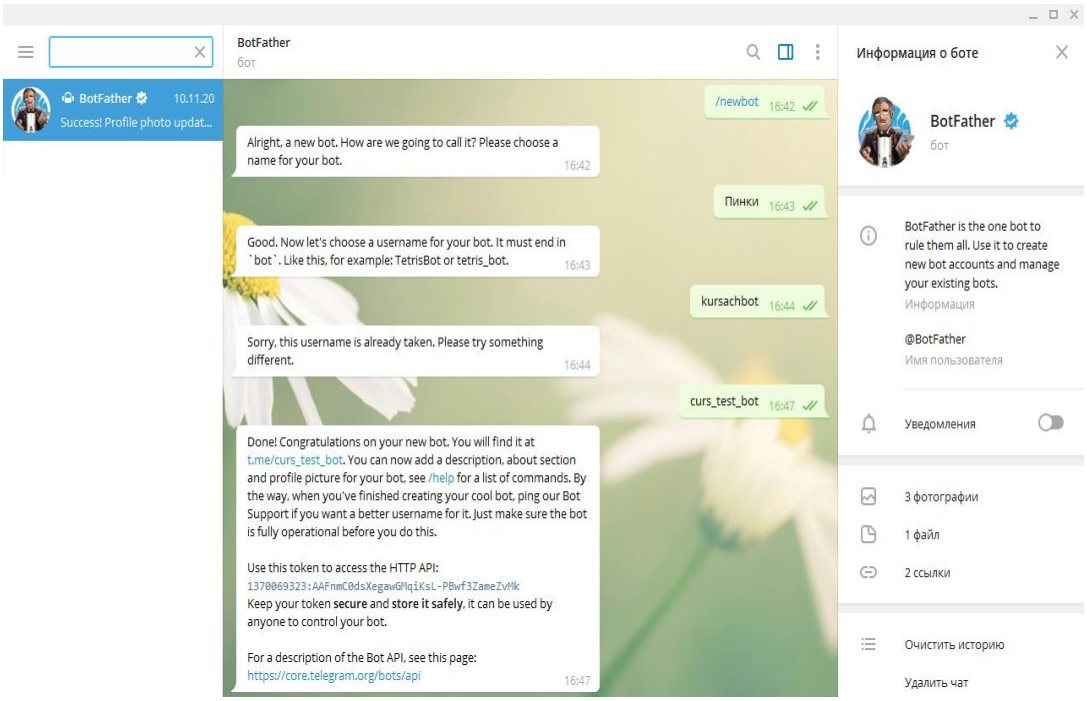

Рисунок 3. Пример выполнения команд

Этот токен используется для авторизации бота и отправки запросов к Bot API.

Как итог, бот зарегестрирован и получен токен. Далее в BotFather можно установить фото профиля, задать описание и приветствие. Токен также необходим для подключения Телеграм бота с его программной частью.

Подготовка к реализации проекта включает в себя установку и настройку необходимых программ и библиотек, а также получение токена.

Скачиваем с официального сайта установочный файл Python актуальной версии. При установке обязательно ставим отметку Add Python «version» to PATH, для дальнейшей установки библиотек через консоль cmd командой pip.

Пример окна установки (Рис. 7).

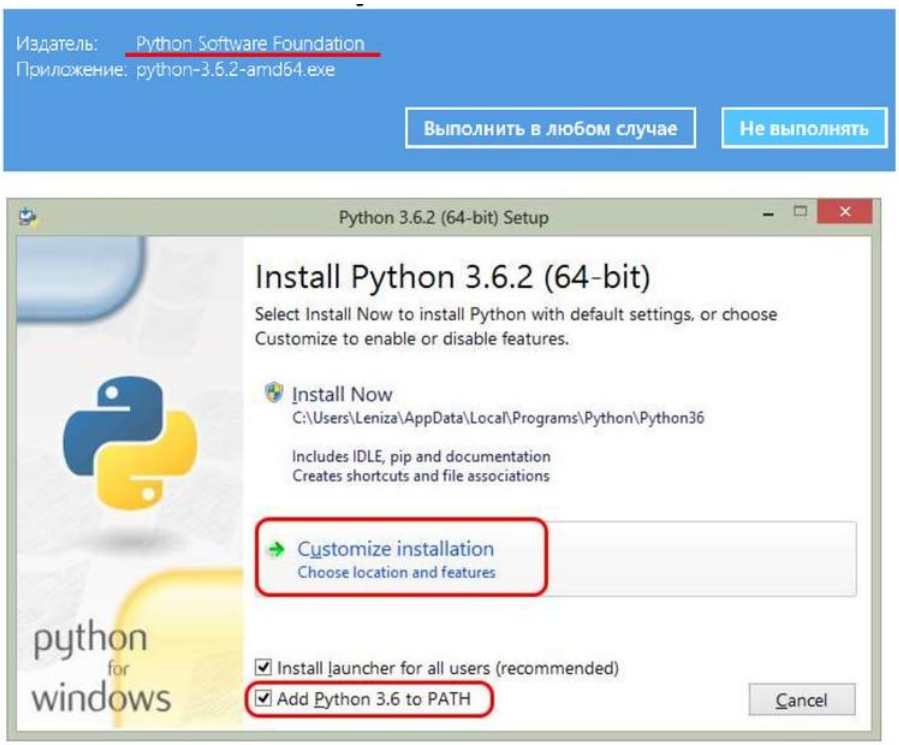

Рисунок 7. Окно установки

Для регистрации, настройки бота и получение токена в BotFather, необходимо выполнить следующие команды (Рис. 8).

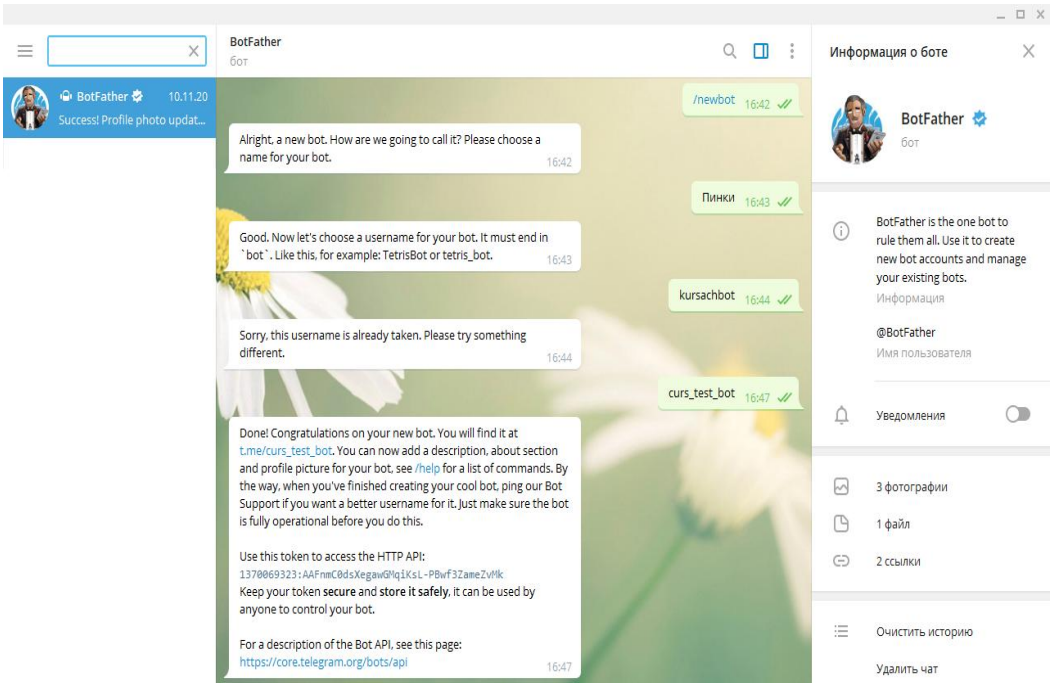

Рисунок 8. Команды для регистрации бота

После получения токена приступаем к коду, первым шагом будет подключение библиотек и токена.

Библиотека sqlite3 является стандартной для Python, начиная с версии 2.5, для ее интеграции достаточно указать в коде:

import sqlite3

Библиотека telebot устанавливается через консоль cmd, для этого, через cmd заходим в корневую папку с файлом кода, с помощью команды cd и прописываем команду *pip install pyTelegramBotAPI* в консоль, установка пройдет автоматически.

Пример установки библиотеки через cmd (Рис. 9).

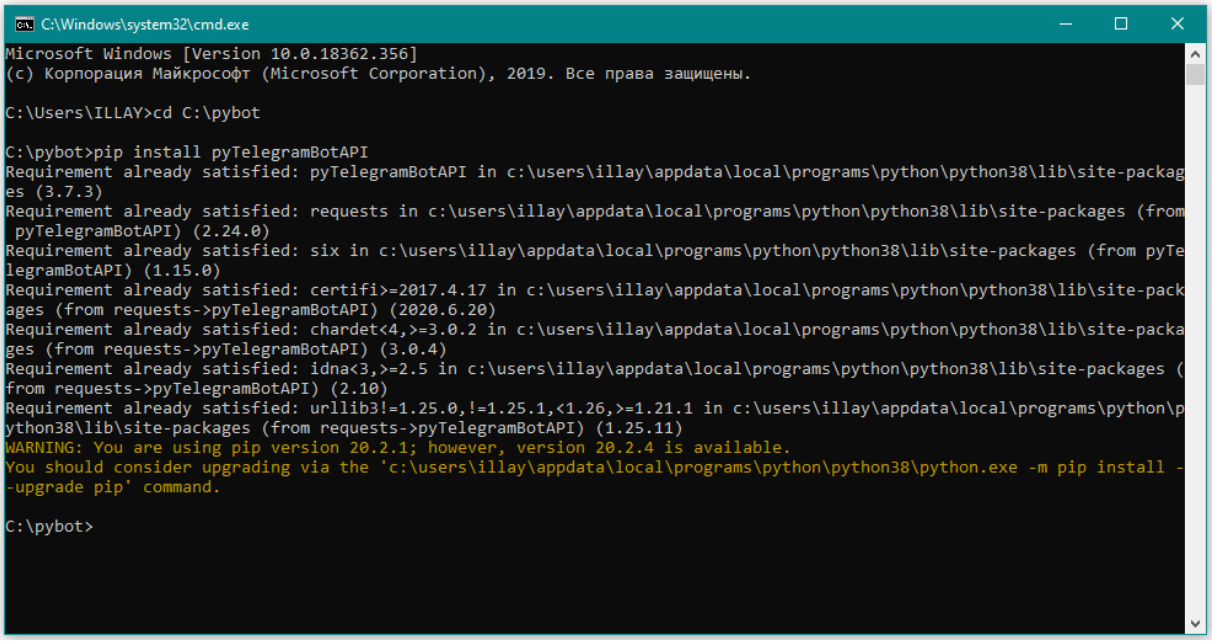

Рисунок 9. Пример установки библиотеки через cmd

В файле кода указываем необходимые библиотеки (Рис. 10).

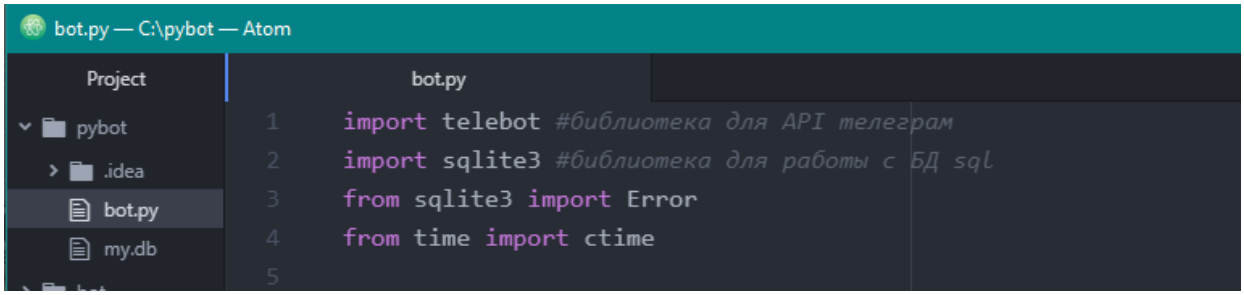

Рисунок 10. Пример подключения библиотек в коде

Прописываем в файле токен полученный от BotFather и указываем id администратора (Рис. 11).

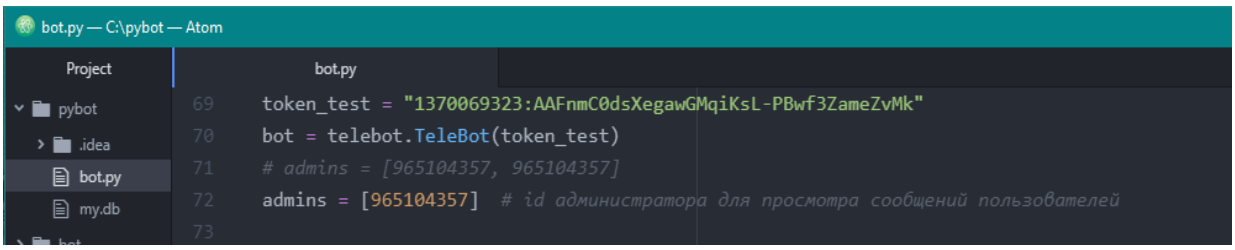

Рисунок 11. Пример подключения токена

Функции для подключения базы данных с таблицами или создания новой БД если она отсутствует в корневой папке (Рис. 12).

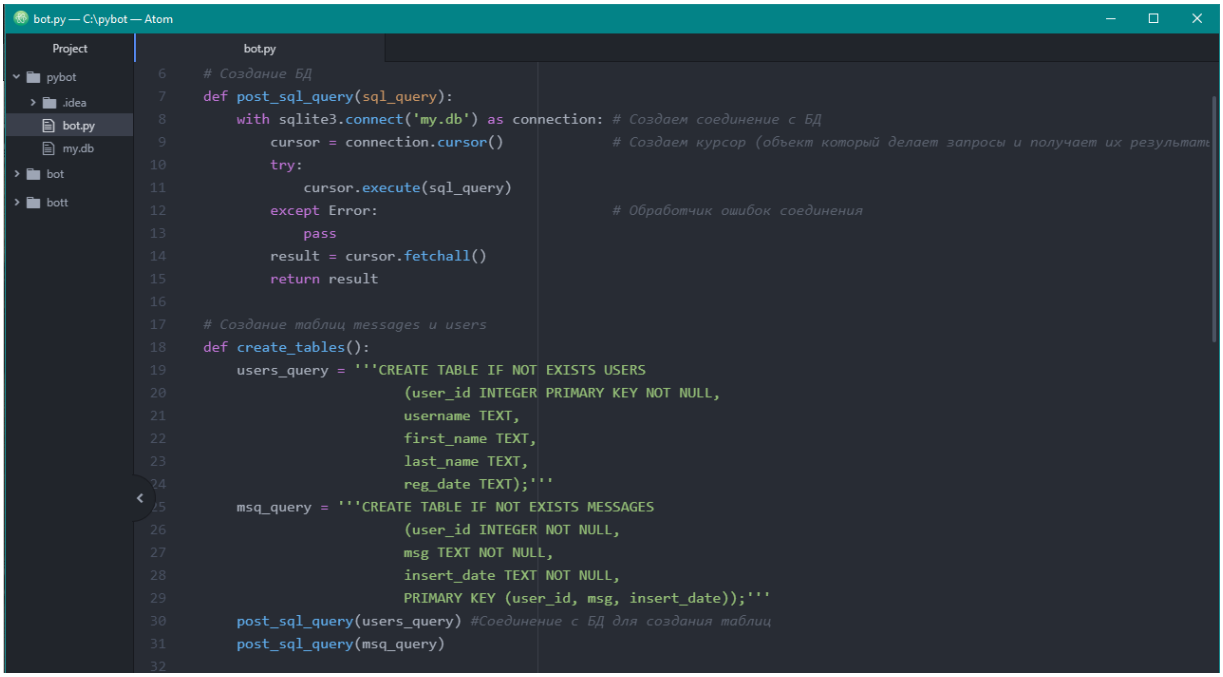

Рисунок 12. Пример функции для подключения БД

Функции для обработки сообщений пользователя, исключения записи команд и обработчики ошибок (Рис. 13).

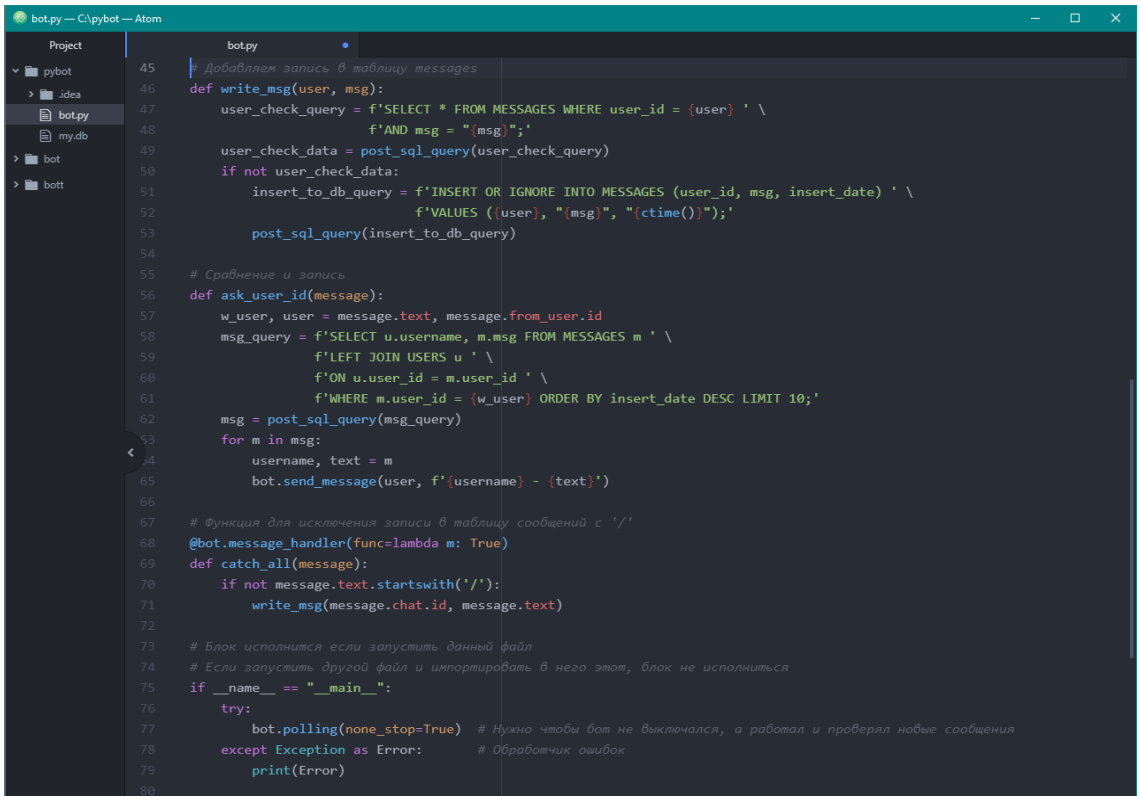

Рисунок 13. Пример функции для обработки сообщений

Обработчики команд регистрации и вывода сообщений «/start» и «/show» (Рис. 14).

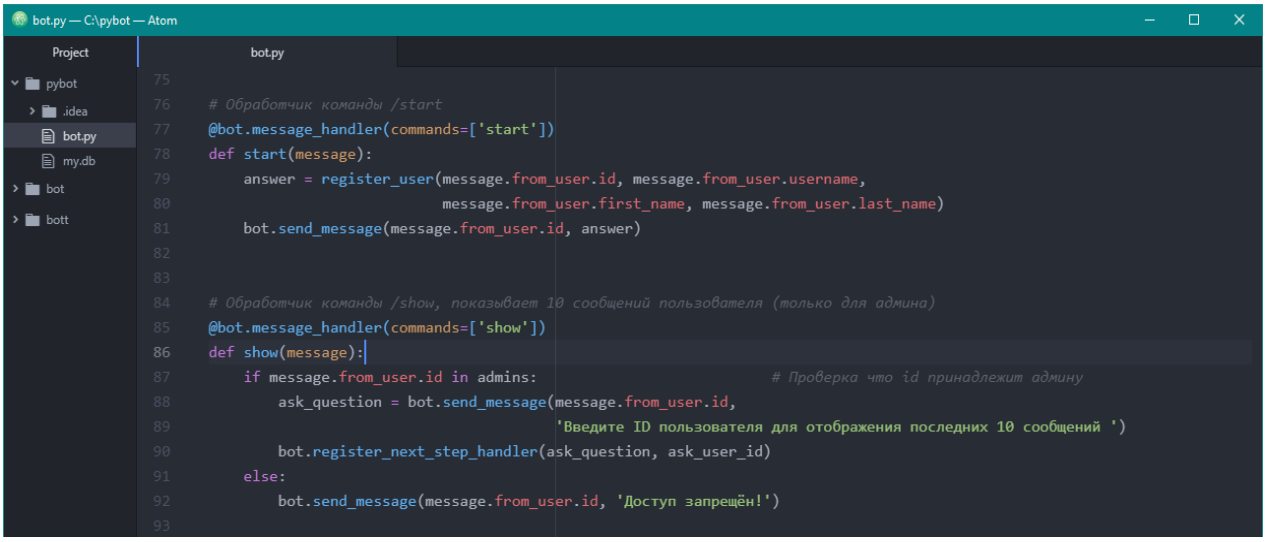

Рисунок 14. Пример обработчиков команд

После завершения кода, запустим файл и протестируем бота:

При запуске код подключается к базе данных в корневой папке, если файл БД отсутствует, создается новая база.

До запуска файла (Рис. 15).

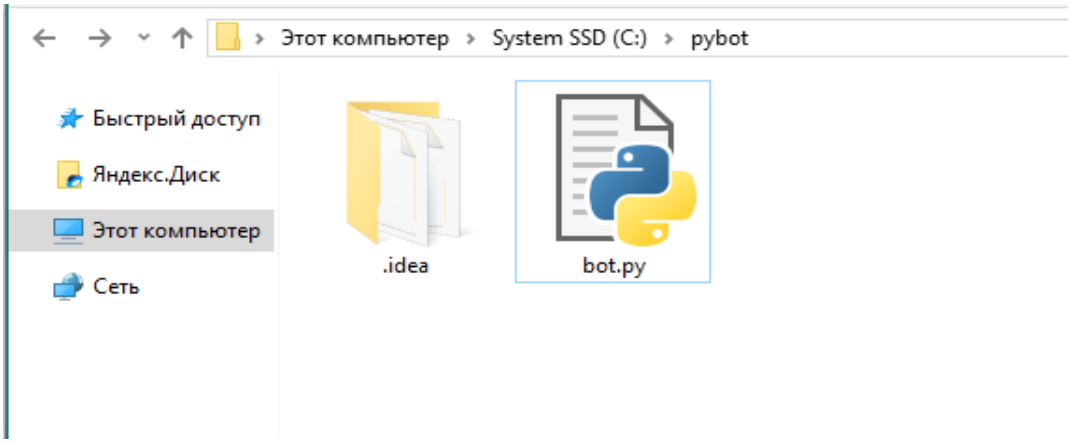

Рисунок 15. Корневая папка до запуска исполнительного файла

После запуска файла (Рис. 16).

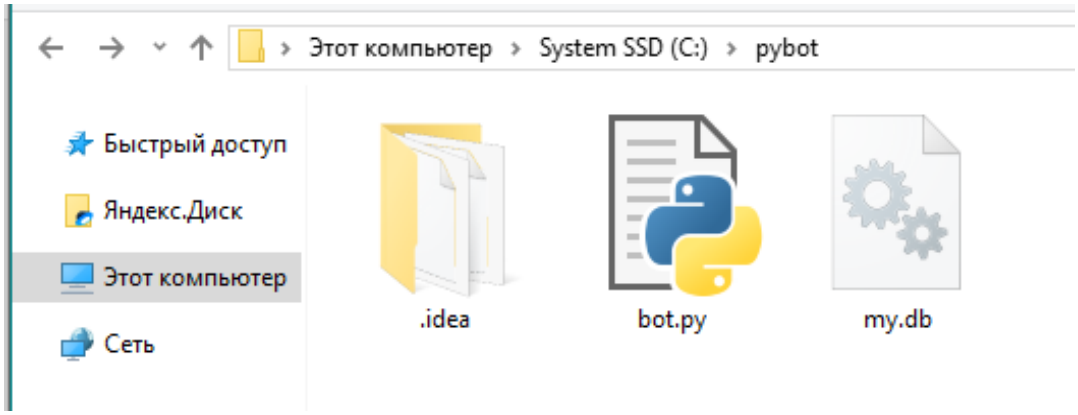

Рисунок 16. Корневая папка до запуска исполнительного файла

Как видно из скриншотов, база данных создается, откроем файл и посмотрим структуру таблицы users (Рис. 17) и messages (Рис. 18).

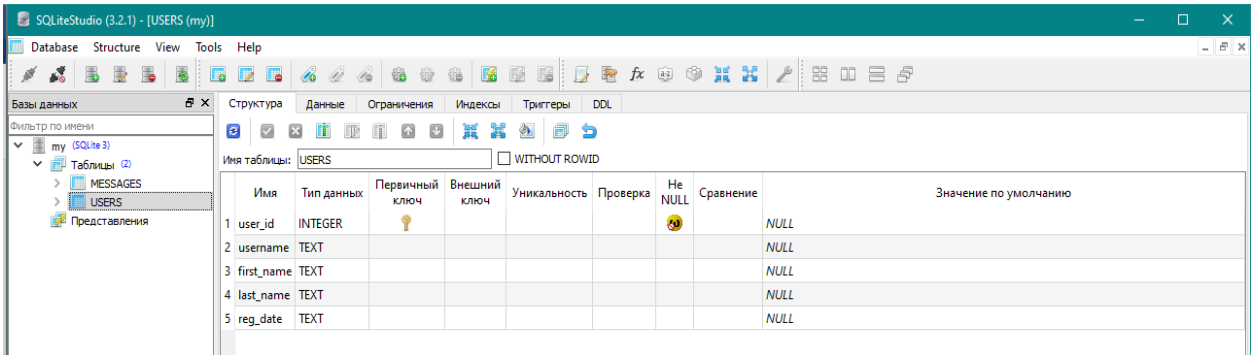

Рисунок 17. Структура таблицы users

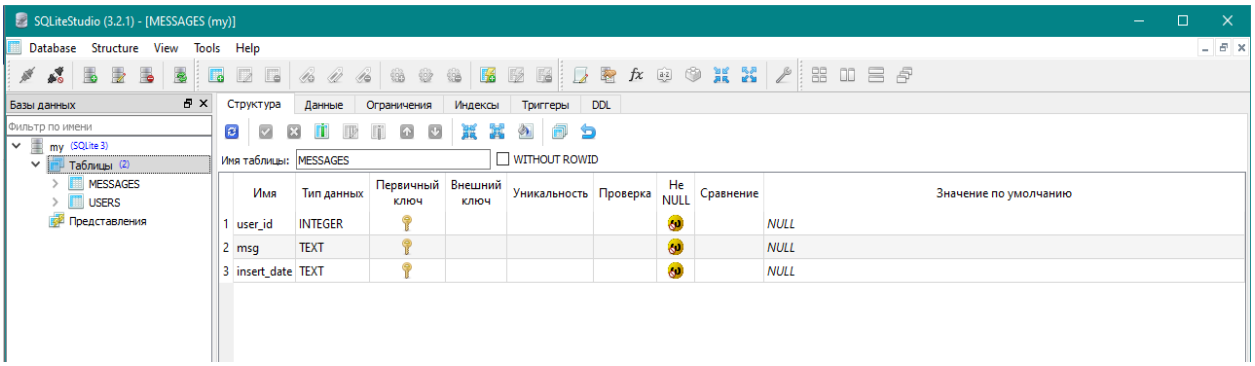

Рисунок 18. Структура таблицы messages

Создались таблицы для записи сообщений messages и регистрации пользователей users, а также первичные ключи для полей id.

Протестируем бота в мессенджере Telegram. В поисковой строке Telegram введем название @curs_test_bot и выбирем бота с соответствующим именем.

Нажимаем кнопку «Запустить» или вручную вводим команду /start.

Бот приветствует нас с выводом никнейма пользователя в Telegram (Рис. 19).

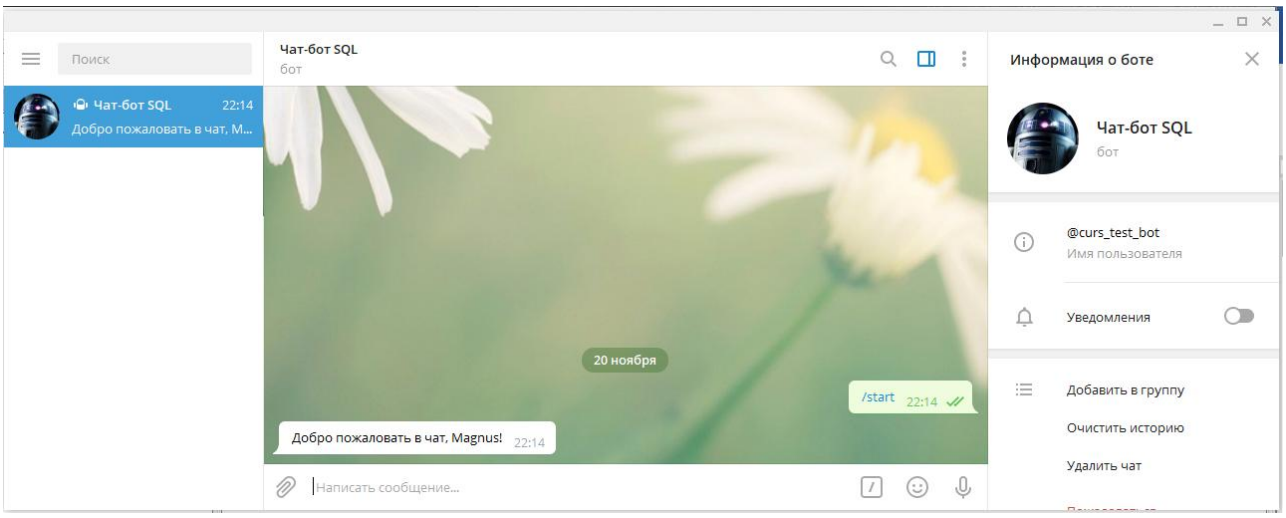

Рисунок 19. Приветствие бота. Регистрация

Проверяем таблицу users в базе данных (Рис. 20).

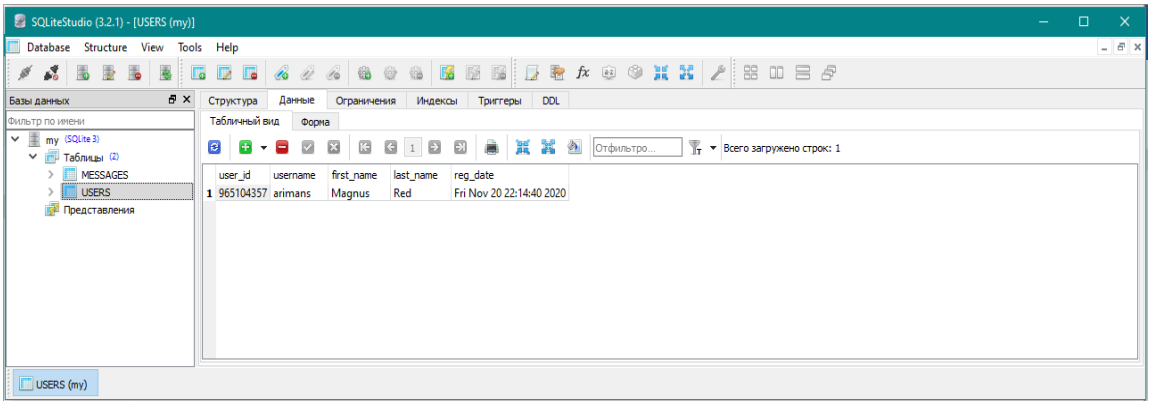

Рисунок 20. Запись в таблицу users

На скриншоте можно наблюдать, что пользователь зарегистрирован и присутствует в таблице.

При попытке снова зарегистрироваться, бот выдаст сообщение (Рис. 21).

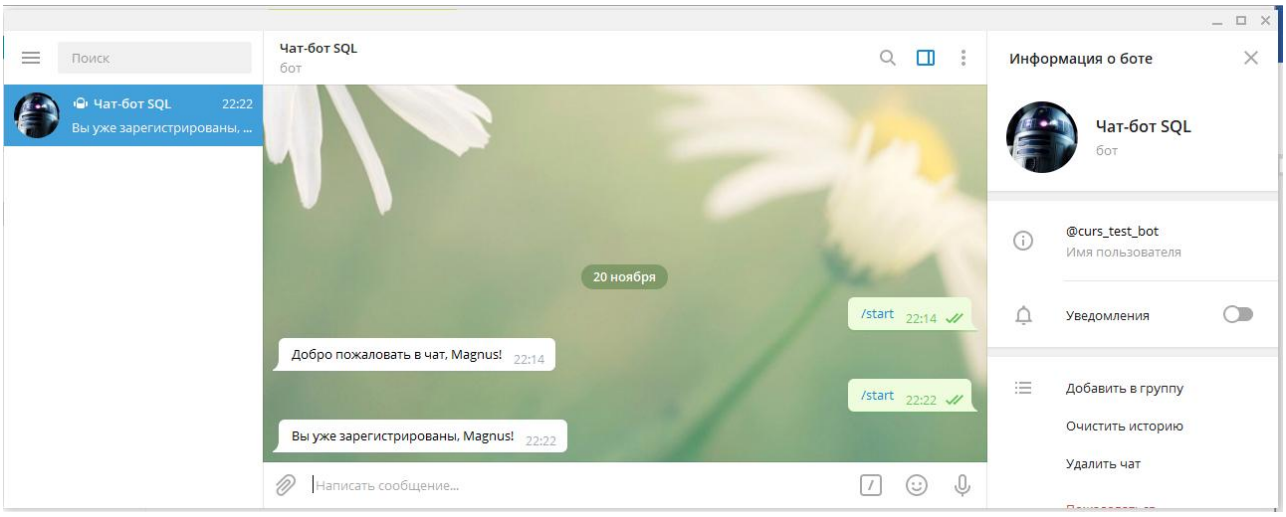

Рисунок 21. Ответ на повторную регистрацию

Напишем сообщения боту для тестирования записи в таблицу messages (Рис. 22).

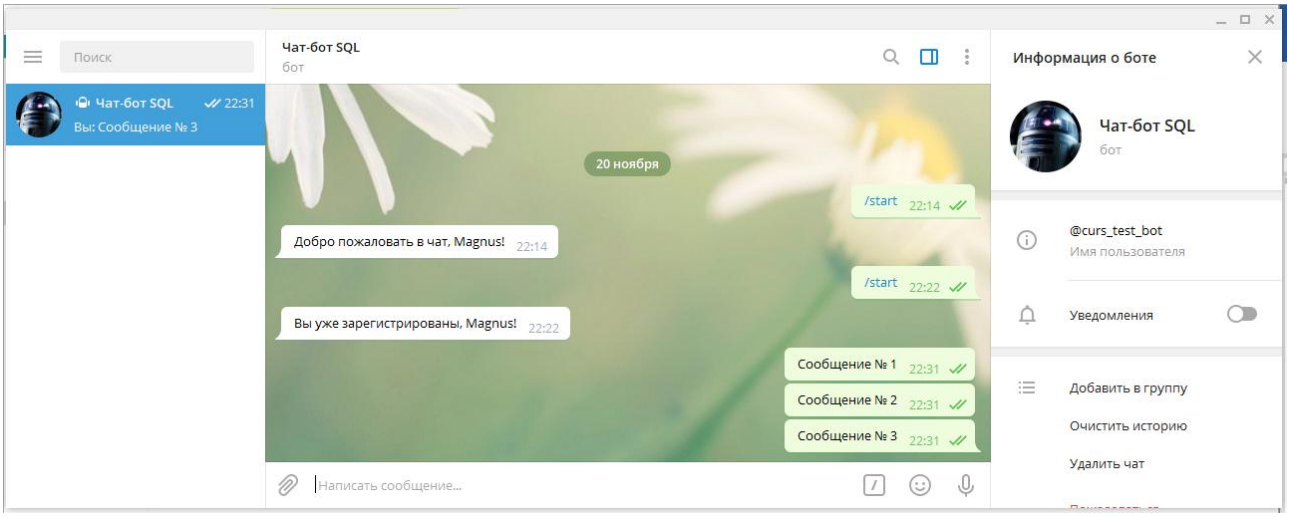

Рисунок 22. Ввод тестовых сообщений

Проверим запись в базе данных (Рис. 23).

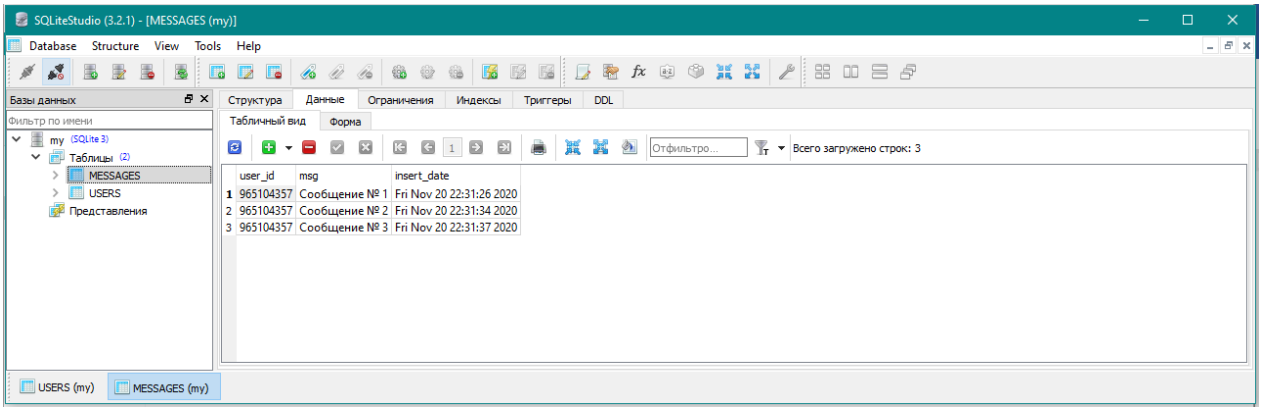

Рисунок 23. Запись тестовых сообщений в таблицу messages

Как видно из скриншотов, сообщения записались в таблицу.

Проверим функцию вывода сообщений пользователя для администратора.

В меню команд выбираем «\show» (Рис. 24).

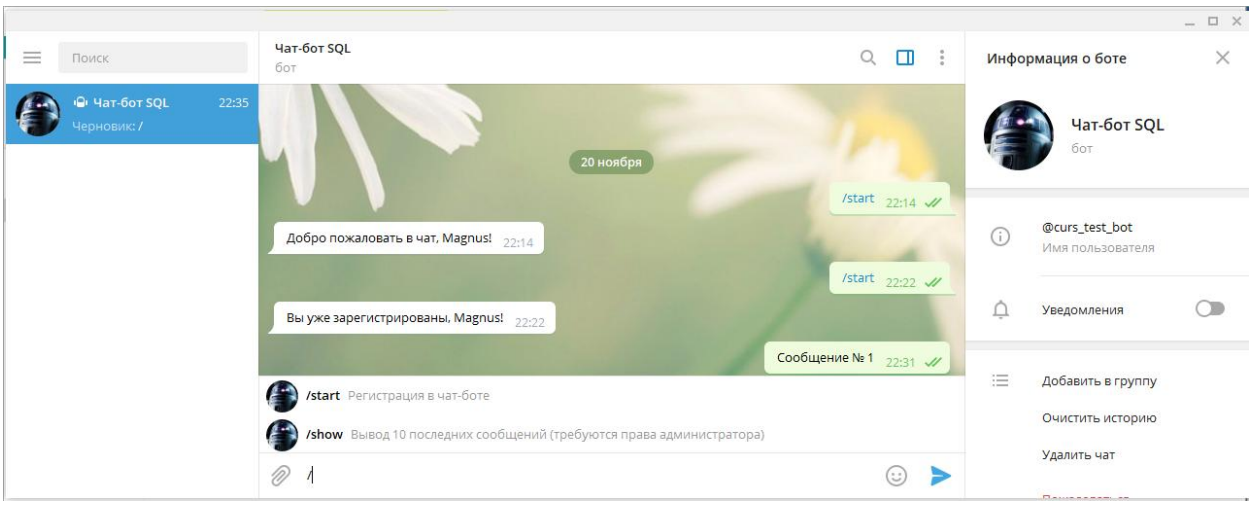

Рисунок 24. Пример интерфейса команд

Бот запросит id зарегистрированного пользователя из таблицы users и выведет последние 10 сообщений данного пользователя.

Пример выполнения команд (Рис. 25).

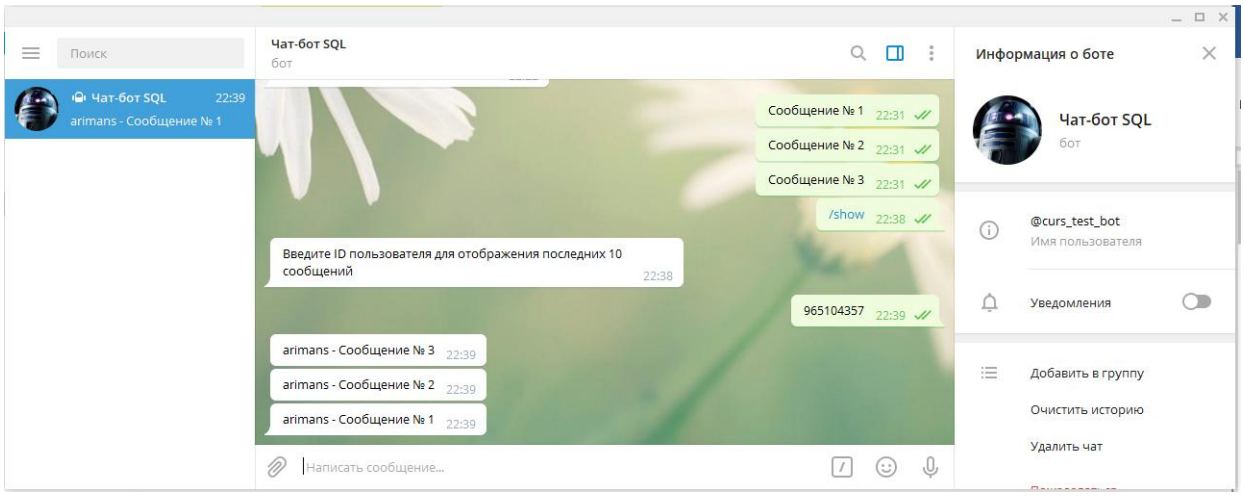

Рисунок 25. Вывод сообщений пользователя

В результате, мы протестировали весь функционал и убедились в работоспособности всех функций.

4. Вывод

В процессе выполнения данного проекта были решены задачи архивирования сообщений в базу данных SQL, регистрация пользователей, а также реализован функционал администрирования.

Библиографический список

1. Карвин Б. Программирование баз данных SQL. Типичные ошибки и их устранение. Рид Групп. 2012.
2. Дейт К. Введение в системы баз данных. М. Вильямс, 2006.
3. Документация. Боты: информация для разработчиков URL:

https://tlgrm.ru/docs/bots.
4. Джанарсанам С. Разработка чат-ботов и разговорных интерфейсов. М.: ДМК Пресс, 2019.
5. Руководство DB-API 2.0 interface for SQLite databases URL: https://docs.python.org/3/library/sqlite3.html.
6. Справочная документация по Bot API URL: https://tlgrm.ru/docs/bots/api.
7. Telegram-бот на python с помощью библиотеки telebot // Статья пользователя под ником kasad0r от 01.08.2018 https://habr.com/ru/post/418905/