Telegram Bot Server for Scheduled Ads

Aleksei Grachev
ČVUT FIT
grachale@fit.cvut.cz
2024-01-07

Problem Statement:

The primary goal of this project is to develop a Telegram bot server capable of scheduling and managing advertisements. In Telegram are very popular chats, which serve as marketplaces, where adds can be posted. Because of the large amount of adds, you have to regularly update the add in every chat, which is very time consuming (in case you have more than one add, you are an entrepreneur for example, it can be an impossible task). So the script, which serve as server for chat bot, solves that problem. The key functionalities include user authentication, creation and scheduling of advertisements, display of advertisement information, and the ability to delete advertisements. There is also an admin mode, which allows you to add or delete new users.

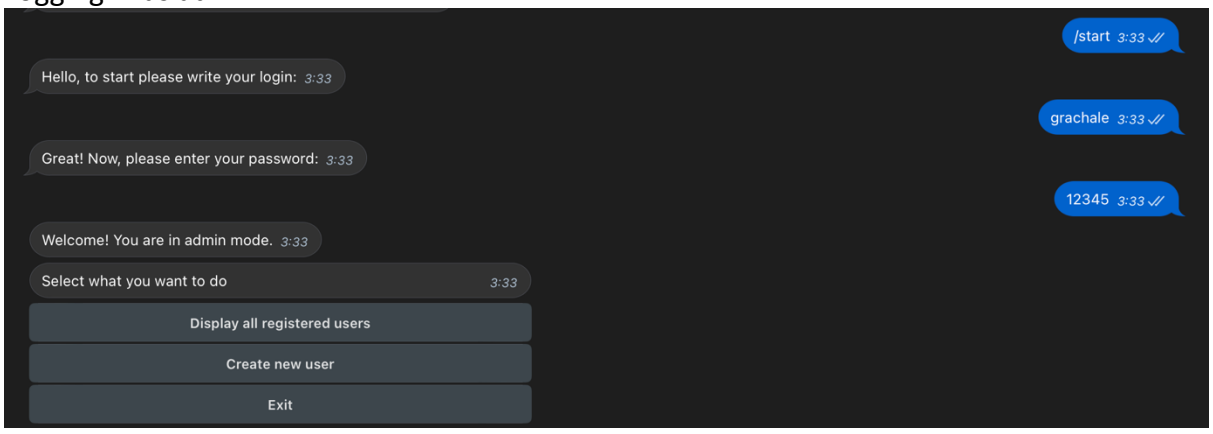
Methods and Approaches:

The project extensively employs the Telebot library for Telegram Bot API interactions, Schedule for task scheduling, and psycopg2 for seamless communication with a PostgreSQL database. The user authentication process is implemented to ensure secure access to the bot's functionalities. Advertisement creation involves user input for chat ID, text, scheduling interval, and time.

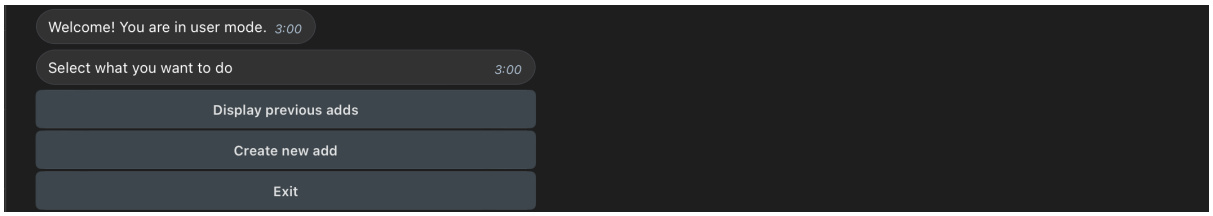
Examples of Usage:

Just run the server with configuration file, that has configs of started PostgreSQL database and token to your chat bot, which you can write to after running the servers. On the next screenshots below, you can see some examples of the interaction with bot in Telegram.

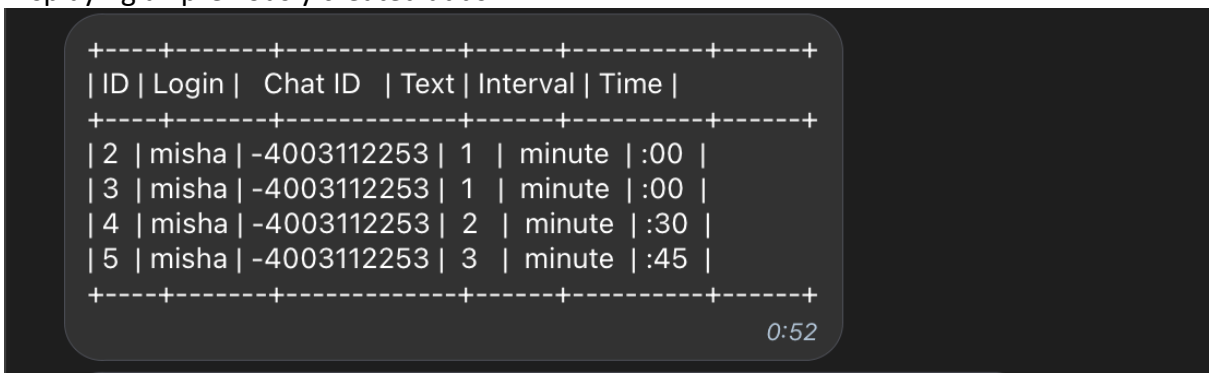
Logging in as admin:



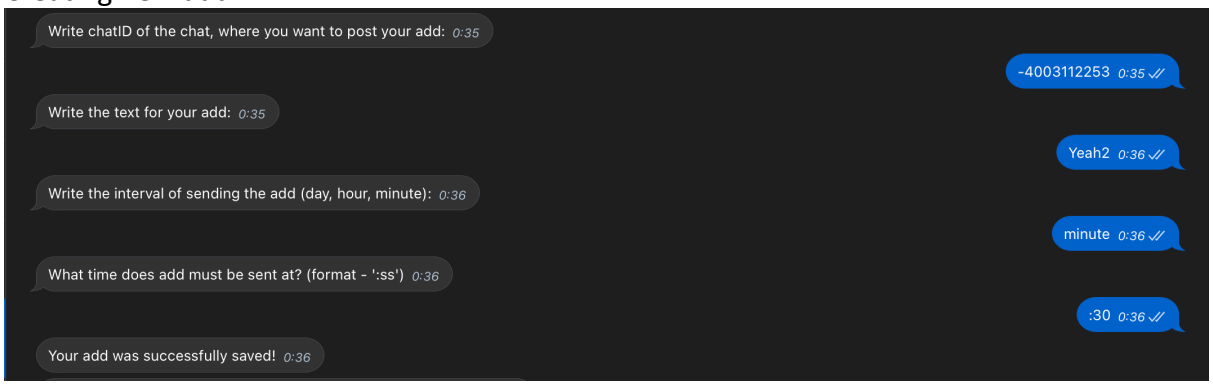
Menu in user mode:



Displaying all previously created adds:



Creating new add:



Results:

The script successfully creates, schedules, and manages advertisements as per user input. The user and advertisement information can be displayed, and deletions can be performed seamlessly. The script employs the Schedule library to execute scheduled tasks, ensuring timely delivery of advertisements.

Future Development:

The current implementation serves as a robust foundation for further enhancements. Future development could focus on additional features such as analytics of adds, automated reporting, and support for multimedia advertisements. The modular structure allows for easy integration of new functionalities.

Outcomes:
The project has been successful in creating a functional Telegram bot server for scheduled advertisements. The script is ready for use and can be extended to accommodate evolving requirements. The implementation demonstrates a deep understanding of bot development, database interaction, task scheduling and thread creation.

References:

1. Telebot Library Documentation - https://pytba.readthedocs.io/en/latest/
2. Schedule Library Documentation - https://schedule.readthedocs.io
3. Psycopg2 Library Documentation - https://www.psycopg.org/psycopg3/docs/
4. Chat GPT (questions about threading in python, error handling and usage of MagicMock in testing) - https://chat.openai.com