

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ТЕЛЕГРАМ БОТУ ДЛЯ УПРАВЛІННЯ
КОНТЕНТОМ РЕКЛАМНОЇ АГЕНЦІЇ»**

Виконав: студент 5 курсу, групи ППЗ-52
спеціальності

121 Інженерії програмного забезпечення

(шифр і назва спеціальності)

Левицький О. Л.

(прізвище та ініціали)

Керівник Щербина І. С.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

ЗАВДАННЯ НА БАКАЛАВРСКУ РОБОТУ СТУДЕНТУ

Левицькому Олегу Леонідовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка телеграм боту для управління контентом рекламної агенції»

Керівник роботи д.т.н., доцент Щербина І.С.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “12” березня 2021 року №65.

2. Строк подання студентом роботи «01» червня 2021 року.

3. Вихідні дані до роботи:

3.1. Документація мови програмування Python та додаткових бібліотек;

3.2. Документація Telegram Bot API;

3.3. Науково-технічна література;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1. Аналіз існуючих рішень та обґрунтування теми проекту;

4.2. Розроблення програмного забезпечення;

4.3. Тестування розробленої системи;

4.4. Висновки.

5. Перелік графічного матеріалу.

5.1. Об'єкт, предмет, та мета дослідження

5.2. Актуальність

5.3. Принцип роботи чат-бота

5.4. Аналоги чат-ботів

5.5. Діаграма роботи з чатом

5.6. Вимоги до чат-бота

5.7. Автоматизована робота з чатом

5.8. Автоматизована реклама

5.9. Інструменти та середовище розробки

5.10 Висновки

6. Дата видачі завдання «19» квітня 2021

7. КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Видача завдання на дипломне проектування	01.03.21	Виконано
2	Вивчення літератури за тематикою роботи	02.03.21	Виконано
3	Вибір засобів розробки	04.03.21	Виконано
4	Реалізація програми	06.03.21	Виконано
5	Тестування програмного комплексу	07.03.21	Виконано
6	Підготовка пояснювальної записки	10.03.21	Виконано
7	Здача роботи	01.06.21	Виконано

Студент _____ Левицький О.Л.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Щербина І.С.
підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота включає пояснювальну записку (49 с., 27 рис., 2 таблиці, список використаної літератури з 16 найменувань та 2 додатки).

Мета дипломного проекту полягає в дослідженні та структуризації теоретичних відомостей про розробку бота на платформі месенджера Telegram та на основі отриманих знань розроблення власного чат-бота для управління контентом рекламної агенції.

Для реалізації поставленої мети проведено аналіз аналогів ботів на платформі Telegram.

В результаті проведеної роботи було створено телеграм бот для управління контентом рекламної агенції.

Результати дипломного проекту використовуються в рекламних агенствах для управління контентом в месенджері Telegram. Користувач має змогу не лише автоматизовано керувати за допомогою бота в власному чаті, а й автоматично публікувати в різних чатах свої оголошення.

ЗМІСТ

ЗМІСТ	7
ВСТУП.....	8
1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	9
1.2 Месенджери	9
1.2 Чат-боти	12
1.3 Огляд аналогів чат-ботів	14
1.3.1 ZukuBot - бот модератор	14
1.3.2 Сервіс для управління чатом «Combot»	15
1.3.3 ChatKeeperBot.....	16
1.3.4 Group Butler	17
2. ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩЕ РОЗРОБКИ	19
2.1 Вибрані технології для розробки чат-бота.	19
2.1.1 Мова програмування - Python.....	19
2.1.2 Модуль Telegram Bot API.....	20
2.1.3 Heroku.....	22
2.2 Середовище розробки	23
2.2.1 Visual Studio Code	23
3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	25
3.1 Реєстрація бота чату для API Telegram Bot.....	25
3.2 Реалізація клієнтської частини чат-бота.....	27
3.3 Розміщення чат-бота на віддаленому сервері Heroku.	31
4 ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ	33
ВИСНОВОК.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	41
ДОДАТКИ.....	42

ВСТУП

Глобальна мережа інтернет - це універсальне середовище для пошуку різної інформації спілкування, розваг і навчання. На даний момент існує велика кількість засобів спілкування і значна частина з них тісно пов'язана з сучасними технологіями, які представлені використанням комп'ютерної мережі.

Інтернет є джерелом різноманітної та корисної для користувачів інформації і також, є основним засобом віртуального спілкування. Зв'язок з рідними та друзями, контакти з колегами по роботі, нові знайомства - все це є важливим компонентом у повсякденному житті сучасної людини. Сьогодні у користувача є великий вибір зручних засобів онлайн-спілкування.

Актуальність випускної кваліфікаційної роботи зумовлена високою популярністю месенджерів та таких засобів автоматизації як боти серед користувачів глобальної мережі інтернет.

Чат-бот це – комп'ютерна програма яка була розроблена на основі нейромереж і технологій машинного навчання, за допомогою неї можливо здійснювати комунікацію в аудіо- або текстовому форматі.[1]

Метою випускної дипломної роботи є розробка бота-помічника для автоматизованого управління контентом та автоматичне опублікування оголошень по чатам та групам рекламної агенції.

Виходячи з поставленої мети, були поставлені наступні завдання:

- аналіз обраної предметної області;
- порівняння наявних аналогів чат-ботів;
- вибір технологій і середовища розробки;
- розробка чат-бота на платформі Telegram;
- тестування чат-боту на коректність роботи.

1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.2 Месенджери

Месенджер - це додаток для смартфона або персонального комп'ютера, що дозволяє миттєво обмінюватися з друзями текстовими повідомленнями, телефонними дзвінками та навіть розмовляти з використанням відеозв'язку. [2]

Розвиток інтернет-сервісів для спілкування почався з чатів, потім месенджери, а далі соціальні мережі. Але на сьогодні месенджери знову стали найбільш популярні та перспективніші сервіси для онлайн спілкування. Це підтверджують дослідження компанії «Київстар» у 2020 році: кількість користувачів месенджерів в Україні збільшилася на 9%. Водночас з цим середня кількість месенджерів, якими користується один користувач, збільшилася за 2020 рік з 1,64 до 3 штук. месенджери використовуються переважно дорослими, так на частку осіб до 18 років припадає 8,3% від загальної кількості аудиторії, користувачі у віці від 18 до 25 років складають 10,1%, від 25 до 35 років - 31,9%, від 35 до 45 років - 27,1%. При цьому 12,9% користувачів месенджерів знаходяться у віці від 45 до 55 років, а 7,5% - від 55 до 64 років. І лише 2,6% аудиторії месенджерів знаходиться в віці старше 64 років. Причиною повторної хвилі популярності месенджерів стали зміни в області мобільного Інтернету, швидкість набагато збільшилась, ціни стали нижчими ніж раніше та широке використання та поширення смартфонів.

Сучасні месенджери мають безліч схожих функцій, таких як надсилання повідомлень, аудіо- та відео спілкування, поширення медіафайлів.

В наш час існує велика різноманітність месенджерів: Viber, WhatsApp, Facebook, Google Hangouts, але на особливу увагу заслугоує месенджер Telegram.

Telegram - це багатоплатформовий додаток, яким можна користуватись на телефонах, планшетах та ПК. Він розроблений мовою програмування C++, тому дозволяє поширювати повідомлення та файли більшості форматів. Telegram використовує спеціально розроблену серверну частину з закритим кодом, які працюють на серверах Німеччини і США. [3]

Telegram месенджер має ряд переваг:

- приватність – всі чати шифруються, а повідомлення видаляються через зазначений час;
- швидкість – швидкість доставки повідомлень набагато вище, ніж в аналогів;
- розподіл – Telegram сервера розташовані по всьому світу, що підвищує працездатність та відмовостійкість;
- відкритість – використання відкритого протоколу MTProto і API, безкоштовних для користувачів;
- відсутність підписок і зайвої реклами;
- немає обмежень на розмір вкладених файлів та повідомлень.

З мінусів можна відзначити, що месенджер не має функції відеодзвінків.

Засновниками Telegram є брати Павло та Микола Дурови. Павло є фінансова та ідеологічна опора проекту, а Микола працює з технічними аспектами.

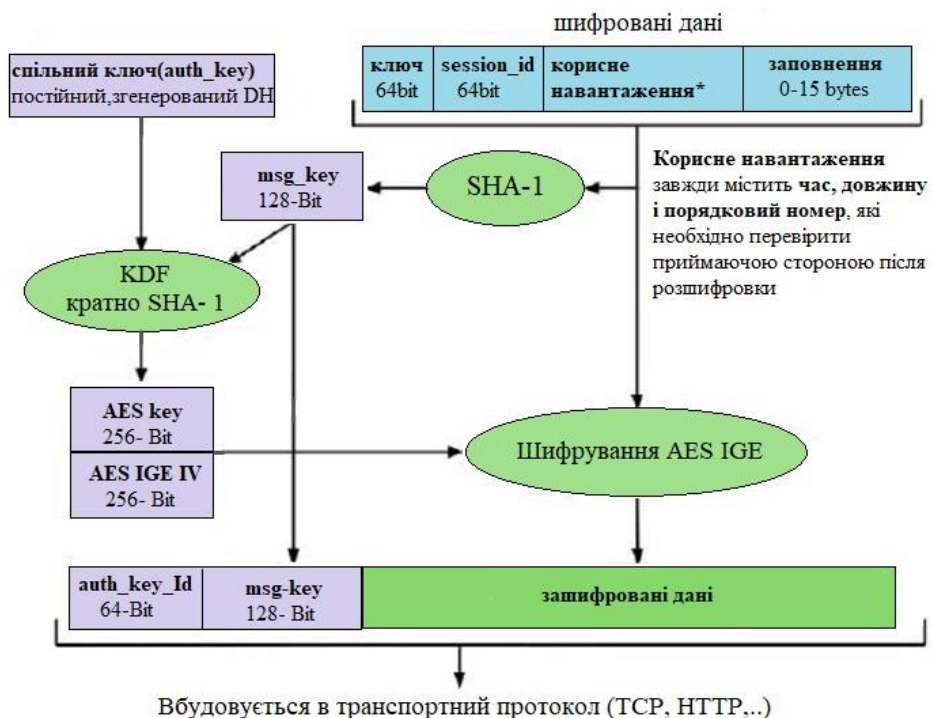


Рис. 1.1. – Протокол шифрування MTProto.

Особливість популярного месенджера це секретні чати та спеціально розроблений протокол шифрування MTProto (рисунок 1.1).

Секретні чати потрібні для тих користувачів, які бажають забезпечити себе максимально безпечним спілкуванням. Крім основного протоколу всі повідомлення проходять шифрування ключами за принципом від пристрою до пристрою. Це створено для того, щоб ніхто не зміг перехопити та не розшифрувати повідомлення, навіть співробітники Telegram. Повідомлення, фото та відео з секретних чатів пересилати та зберігати не можна і вони не зберігаються на серверах додатку. Але секретний чат має два мінуси:

- неможливо відкрити чат на пристрої, який не брав участі в його створенні;
- не захищеність від скриншоту інтерфейсу месенджера де відкритий секретний чат.

Месенджер Telegram можна використовувати майже на всіх популярних операційних систем, наприклад, Android, iOS, Windows, Ubuntu та інші.

Приклад інтерфейсу Telegram месенджера на Windows 7 та Android зображені на рисунках 1.2 та 1.3.

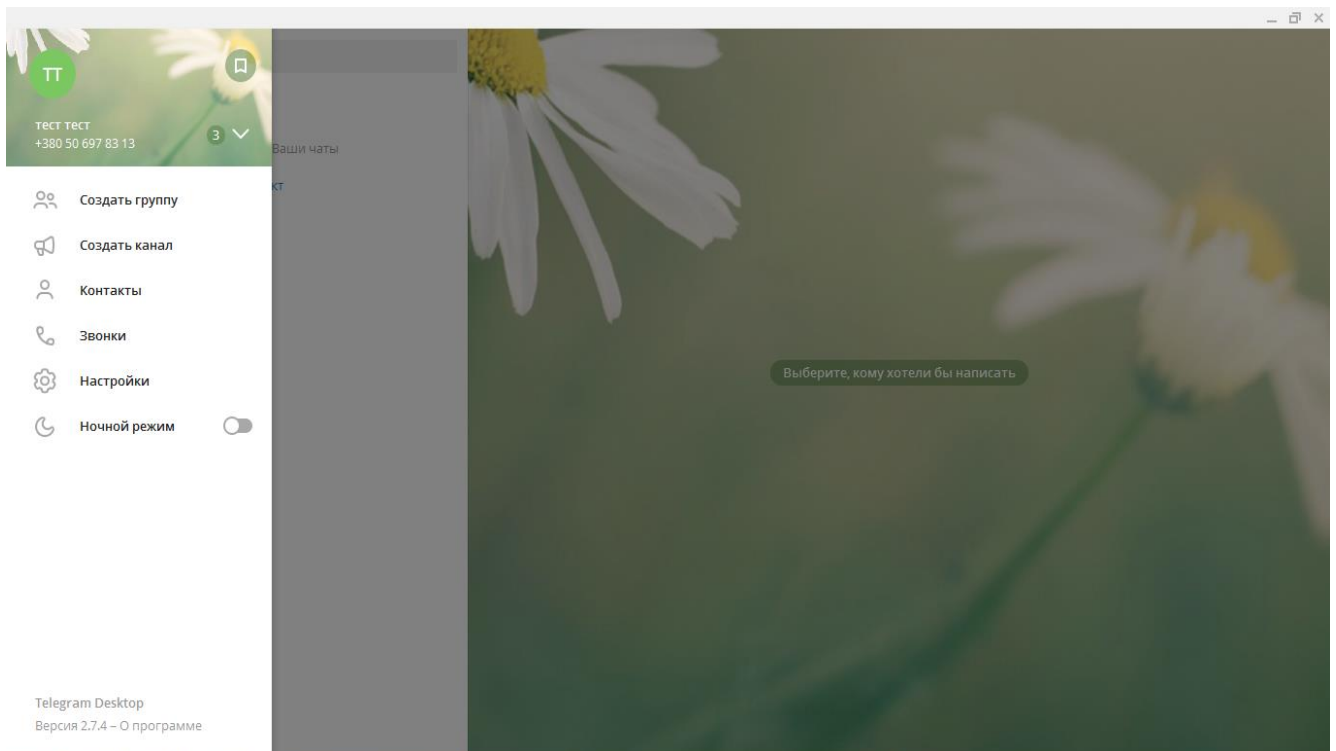


Рис. 1.2. – інтерфейс Telegram месенджера на Windows 7.

Також в більшості версія для різних платформ доступні: англійська, російська, польська та німецька мови та інші.

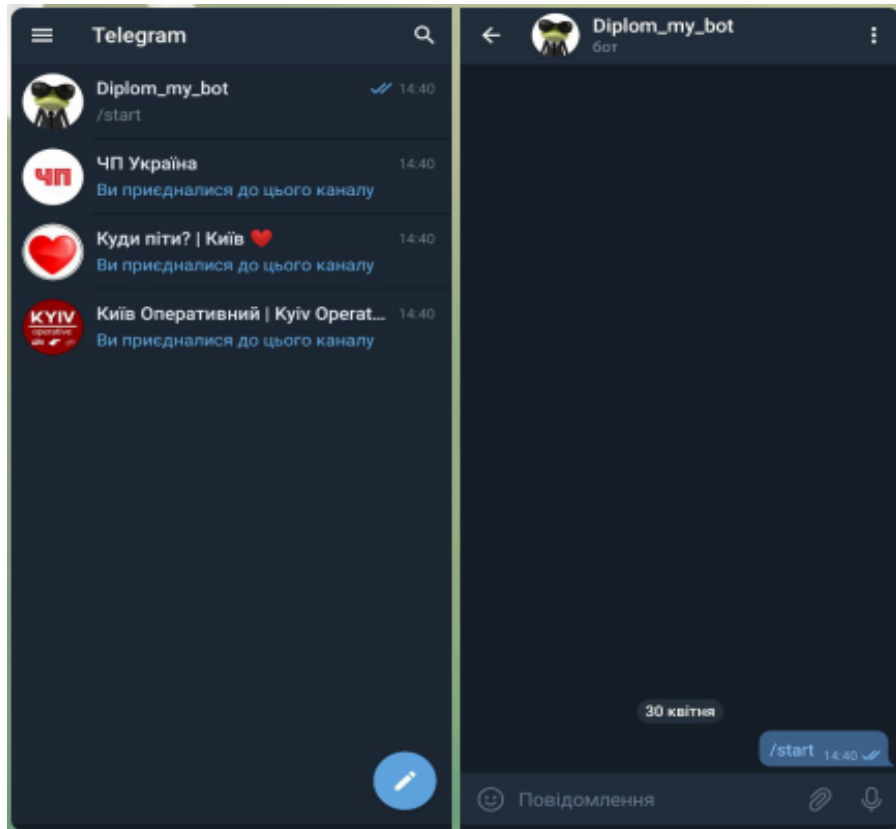


Рисунок 1.3. – інтерфейс Telegram месенджера на Android.

У 2018 році Telegram активно працював над розробкою своєї блокчейн-платформи TON(Telegram Open Network) і криптовалютою Gram на її основі. Але у 2020 році Павло Дуров заявив, про закриття проекту.

1.2 Чат-боти

Чат-бот - це програма, яка була розроблена на основі технологій машинного навчання та нейромереж. Вона створюється людиною для людей та навчається під певне коло цілей. Чат-бот імітує розмову з людиною в Інтернеті, саме тому даний сервіс найкраще зарекомендував себе в месенджерах. [4]

В останній час в додатках набирають популярність такі сервіси як чат-боти. Термін «чат-бот» був придуманий та створений Майклом Молдін в 1994 році для опису розмовних програм.

Пізнавши сучасний стан використання чат-ботів в додатках, можна прийти до висновку, що це є універсальний засіб, здатний до вирішення безліч різних

завдань. За допомогою чат-ботів можна спілкуватися, надавати медичну консультацію і навіть робити замовлення товарів та послуг.

В незалежності від операційної системи, чат-бот – це прикладна програма, яка, отримує інформацію від користувача та формує логічно обґрунтовані відповіді. На сьогодні є дуже величезна різноманітність ботів помічників. Деякі з можливих варіантів представлені нижче:

- ігрові чат-боти (квести, загадки);
- рекламні чат-боти (SAT, Pepsi Cola);
- новинні чат-боти (Укрінформ, Kievinfo);
- чат-боти для замовлень, магазинів, сервісів (таксі «Uber», піцерія «Мама-Мия», бронювання поїздів і літаків «Tickets.ua»);
- чат-боти для консультації та підтримки клієнтів (банк «Моно») та інші.

В останній час великі можливості мають інформаційні технології, пов'язані з ботами. Вони дуже міцно ввійшли в наше життя, що мають застосунок у всіх сферах нашої діяльності та з кожним днем їх роль стає все більшою. Тому, що основну частину свого часу люди проводять за комп'ютерами та смартфонами в email-клієнтах і месенджерах.

Боти - це програми, які використовуються для різних завдань користувача, що перебуває в месенджері. Бот в Telegram месенджеру виглядає як звичайний чат, але спілкування відбувається з програмою, а не з людиною. [5] За допомогою такого бота можна зробити замовлення машини, якщо це - бот таксі, або присилати свіжі новини, погоду, або замовити доставку їжі додому, якщо це - ресторанний бот.

Зовсім недавно чат-боти здобули не малу популярність, перетворившись з розваг в більш серйозну річ, такі боти в основному, стали використовуватися для вирішення серйозних завдань. В час інформаційних технологій - це нормальне явище, а тим більше для мережі Інтернет, адже сучасне суспільство перейшло на «нове» і ділове, і неділові спілкування. Зараз чат-боти це програми для вирішення бізнес-завдань. Чат-бот це додаток, який веде діалог з людьми, вибираючи логічні відповіді з бази даних, після питання де пообідати ви тут же отримуєте миттєву відповідь. Крім цього, чат-боти виконують багато корисних

операцій: виконання рутинної роботи, пошук інформації, об'єднання даних та роботою з клієнтами. Він використовується як віртуальний співрозмовник має базу знань та відповідей, за допомогою цього може дати відповідь на різні питань користувача. Для того щоб чат-бот міг дати логічну відповідь використовують ключові слова, збіг фрази та контексту. Постійно існують прості та легкі по виконання справи, на які не завжди є час, та бажання. На таку допомогу можна використати бота. Для пошуку та збору інформації можна використовувати чат-ботів. Наприклад, коли відбувається який-небудь захід чат-боти можуть повідомляти всім учасникам новини та довідкову інформацію.

1.3 Огляд аналогів чат-ботів

1.3.1 ZukuBot - бот модератор

ZukuBot - бот модератор. Займається модерацією чату, дозволяє додати обмеження новим користувачам, ввести капчу, тригери, попередження, налаштувати карму, додати вітання та інше. Інтерфейс бота на рисунку 1.4.

Основні плюси даного бота:

- простий для використання;
- безкоштовний у використанні;
- велика кількість параметрів для налаштування модерації.

З мінусів можна виділити:

- неможливість підключити свого бота;
- включення всіх функцій бота мінімальний тариф – 3\$ на місяць;
- багато спам реклами в чаті.

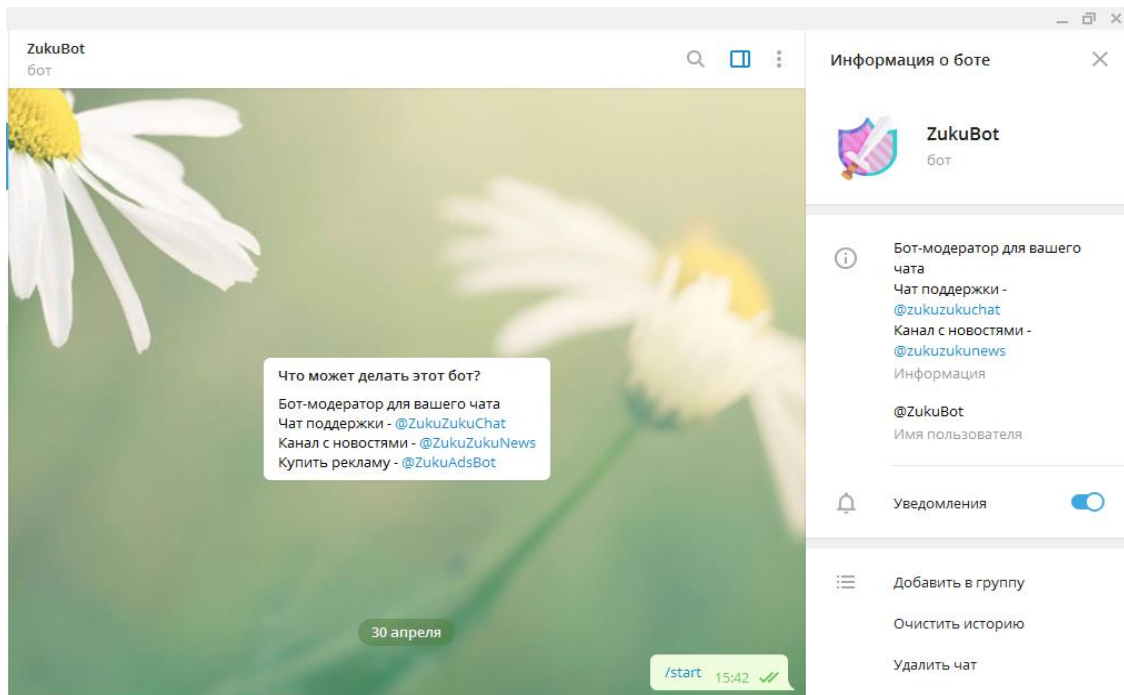


Рис. 1.4. – інтерфейс чат-боту «ZukuBot».

1.3.2 Сервіс для управління чатом «Combot»

Combot - сервіс для управління групами через сайт, в можливості якого входить: детальна аналітика, гнучкі настройки модерації, тригери, додавання рівнів і XP, створення системи репутації, різні попередження, капча, антиспам, звіти та багато іншого. Інтерфейс бота зображений на рисунку 1.5.

Плюсом у користуванні даного сервісу є те, що цей бот має:

- велику кількість функцій і можливих варіантів для їх використання;
- проведення статистики по групі, користувачів і їх повідомлення;
- зрозумілий інтерфейс налаштувань;
- наявність мобільної версії;
- безкоштовний тариф, але тільки до 200 учасників.

До мінусів «Combot» потрібно віднести:

- трохи складний (зважаючи на велику кількість параметрів) у користуванні;
- для відкриття додаткових функцій, мінімальний тарифний план - \$ 9,99 на місяць;
- не сумісний з іншими ботами на безкоштовному тарифі.

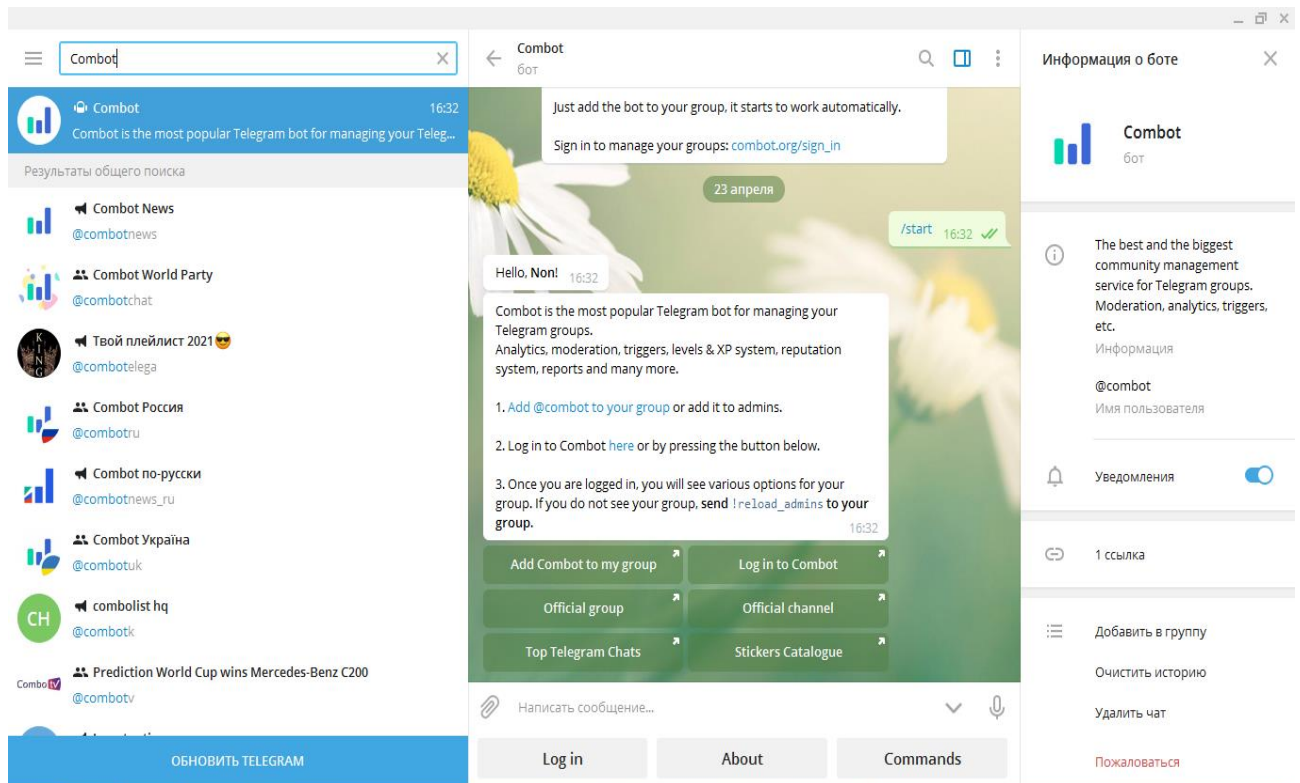


Рис. 1.5. – інтерфейс чат-бота «Combobot»

1.3.3 ChatKeeperBot

ChatKeeperBot - платформа для модерування чатів, яка вмiє: додавати антиспам, репутацію, вести статистику, налаштовувати вітання, капчу, тригери, фільтри і багато іншого. Вся взаємодія з ботом відбувається через сайт. Інтерфейс бота зображений на рисунок 1.6.

Із плюсів слід зазначити:

- безкоштовний тариф, але обмежено багато функцій;
- широкий простір можливостей, велика кількість функцій і налаштувань.
- Наявність фішок які відрізняють від конкурентів;
- наявність мобільної версії.

Однак існують мінуси:

- мінімальний платний тариф - \$ 3 на місяць;
- складний інтерфейс в особистому кабінеті;
- платне додавання свого бота, тариф від \$ 28 в місяць;
- присутньо багато спам реклами.
- взаємодія з ботом відбувається тільки через сайт.

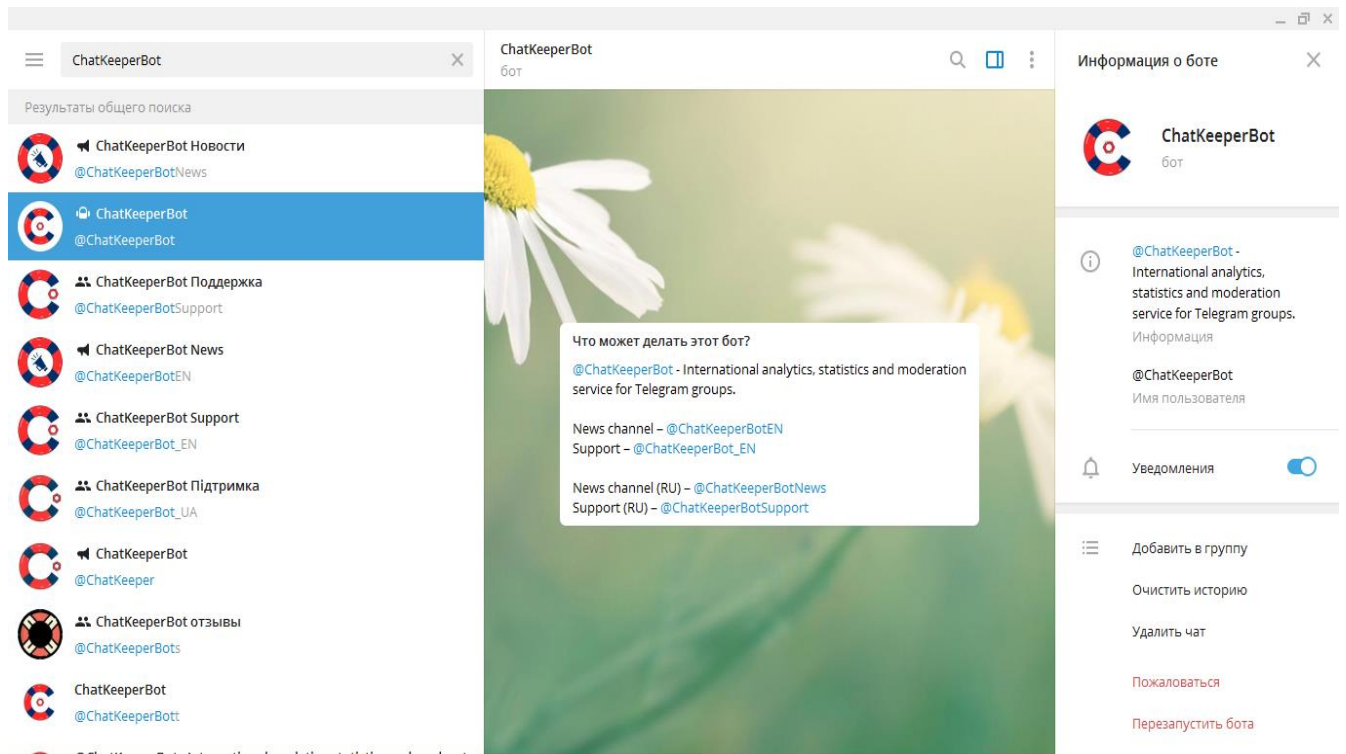


Рис. 1.6. – інтерфейс чат-бота «ChatKeeperBot».

1.3.4 Group Butler

Group Butler - бот для адміністрування груп, в список функцій якого входить: модерація, фільтри, вітання та попередження. Інтерфейс даного боту зображений на рисунку 1.7.

Плюсами даного чат-бота являються :

- наявність мобільної версії;
- простий у використанні;
- бот повністю безкоштовний.

З мінусів слід зазначити:

- налаштування боту відбувається в чаті;
- не можна додати свого бота;
- налаштування відбувається в чаті.

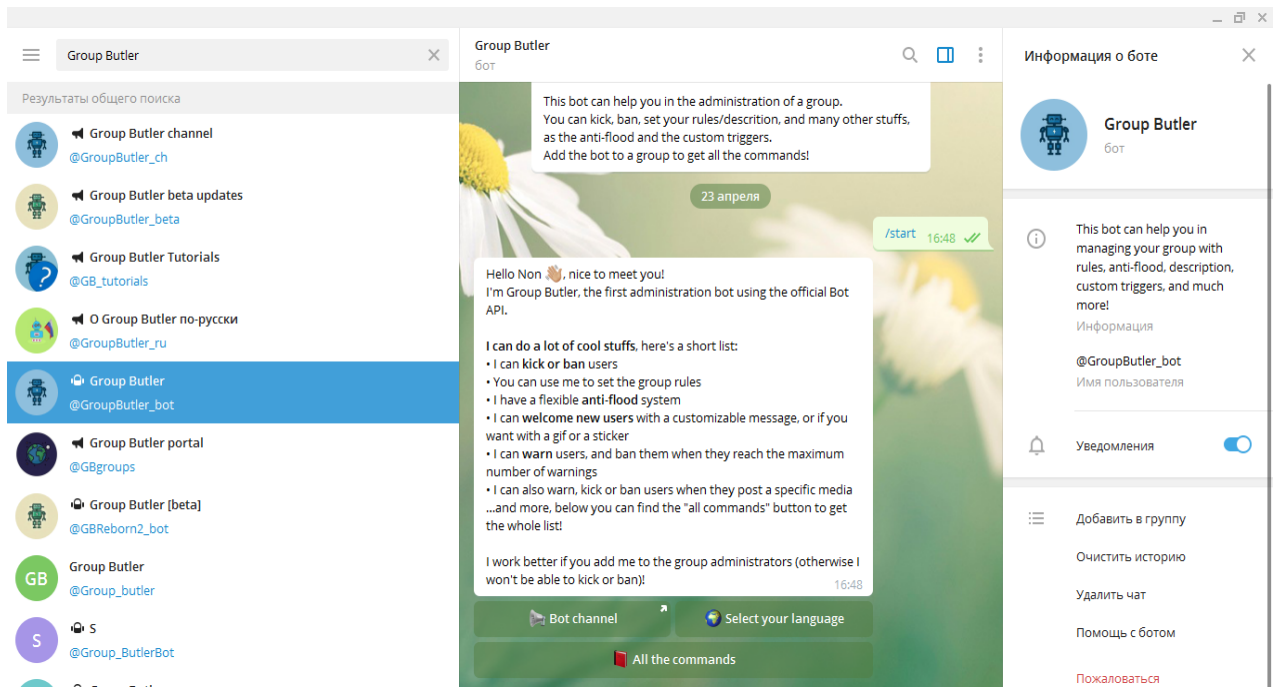


Рис. 1.7. – інтерфейс боту «Group Butler».

Після аналізу аналогів, було вирішено зробити бота для месенджера Telegram. Для виконання даної роботи буде використовуватись мова Python v3.6, а також будуть використанні окремі її модулі, які відповідають за обробку запитів TeleBot для роботи з Telegram API, та способами підключення до віддаленого серверу Heroku.

2. ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩЕ РОЗРОБКИ

2.1 Вибрані технології для розробки чат-бота.

2.1.1 Мова програмування - Python

Python – скриптова та високорівнева мова програмування, що фокусується на читабельності коду. У наш час Python є широко поширеною мовою, яка використовується в багатьох областях, наприклад, для розробки прикладного ПЗ, web-додатків та в якості вбудованого скрипта в багатьох іграх. Мова націлена на покращення продуктивності праці, вона допомагає розробникам робити кодування за кілька кроків у порівнянні з Java або C++.

Python - це динамічна програма програмування високого рівня, інтерпретована та загальноприйнята, що фокусується на читанні коду. Синтаксис в Python допомагає програмістам робити кодування за кілька кроків порівняно з Java або C++. Вона була заснована в 1990 році розробником Гвідо Ван Россумом, при її використанні програмувати легко та цікаво. [6]

Мова Python є широко використовувана у великих компаніях тому, що вона підтримує різні парадигми програмування: об'єктно орієнтовану, функціональну, структурну, імперативну та аспектно орієнтовану. Python має велику стандартну бібліотеку яка є комплексна та має автоматизоване керування пам'яттю та динамічні функції.

Більшість великих компаній які спеціалізуються на розробці програмного забезпечення, віддають перевагу мові Python тому, що вона є універсальною та через меншу кількість програмних кодів. Майже 15% розробників користуються нею в таких операційних системах, як UNIX, Linux, Windows та Mac OS. Розробники у великих компаніях користуються Python, оскільки вона має в розробці програмного забезпечення з характерні функції, такі як:

- інтерактивність;
- інтерпретовність;
- модульність;
- динамічність;

- об'єктно-орієнтованість;
- портативність;
- високорівневність;
- можливість розширення на C++ та C.

До переваг мови програмування Python можна віднести відкриту розробку коду, досить легка у навчанні, особливо на початковому рівні, особливості синтаксису допомагають програмісту писати код швидше, легко читається та безліч корисних бібліотек та розширень мови.

Наступна значна перевага мови Python це велика чисельність модулів, які можна імпортувати, щоб забезпечити багато різних додаткових можливостей. Ці модулі пишуться на мові C, на самому Python і можуть бути використані більш досвідченими програмістами. До приклада можна навести такі модулі:

- NumPy (Numerical Python) – дає розробнику набагато ширші можливості в математиці, такі як робота та маніпуляція з векторами та матрицями;
- Tkinter – він надає можливість розробляти додатки з додаванням графічного інтерфейсу користувача (GUI);
- OpenGL (Open Graphics Library) – є дуже великою бібліотекою графічного моделювання двовимірних та тривимірних об'єктів OpenGL.

Єдиним мінусом, про який говорив сам розробник Python, низька швидкість виконання Python програми порівняно з іншими. Але це не грає велику роль якщо порівнювати з перевагами мови при кодуванні програм не дуже критичних до швидкості виконання.

2.1.2 Модуль Telegram Bot API

Bot API це створений HTTP-інтерфейс для роботи з ботами в Telegram месенджері. Кожний бот - це спеціальний акаунт, розроблений для автоматизованого оброблення та відправлення повідомлень. Існує два різних за принципом способи отримати оновлень від бота:[7]

- long pulling - додаток який автоматично зв'язується з сервером Telegram на наявність нових оновлень для бота. Встановлена швидкість за замовчуванням 100мс;

- webhook - сервера месенджера які автоматично сповіщають додаток на сервері на наявність нових оновлень. Такі оновлення будуть збережені на сервері, до того часу поки вони не опрацюються, але не довше однієї доби. В будь якому разі, незалежно від способу отримання оновлень, буде відправлено об'єкт Update, серіалізовані в JSON.

Всі запити які будуть надходити до Telegram Bot API мають дійснюватися через HTTPS в наступному вигляді:

https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ.

Принцип роботи взаємодії чат-бота і користувача зображений на рисунку 2.1.

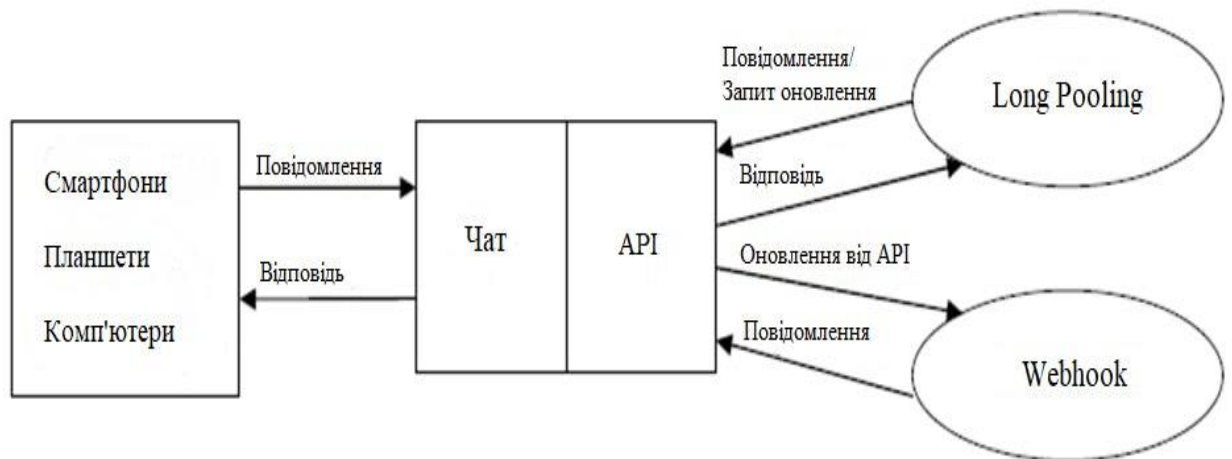


Рис. 2.1. - Принцип роботи чат-бота на платформі Telegram.

Для отримання унікального ключа потрібно написати спеціальному боту @BotFather. Зразки доступних для API методів можна побачити нижче:

- getUpdates - цей метод використовується для отримання оновлень за технологією long polling;
- setWebhook - метод який прив'язує до бота url домен, де знаходиться відкритий і запущений бот;
- sendMessage - метод який робить відправку текстового повідомлення в клієнт Telegram;

- `sendLocation` - метод відправляє повідомлення з координатами в клієнт Telegram;

- `getFile` - метод який робить повернення завантаженого файлу по його імені.

Допускаються також POST і GET запити. Існує 4 способи для відправки параметрів в Bot API:

- Запит в URL

- `application / x-www-form-urlencoded`

- `application / json` (не прийнятний для завантаження файлів)

- `multipart / form-data` (для завантаження файлів)

2.1.3 Heroku

Heroku – це хмарна PaaS-платформа, яка є сумісна з декількома мовами програмування. З 2010 року є дочірньою компанією Salesforce. Вона є однією з перших хмарних платформ яка з'явилася в липні 2008 року.[8] Спочатку Heroku мала підтримку тільки однієї мови програмування Ruby, але на сьогодні перелік мов які вона підтримує містить в собі Python, Java, Clojure, Node.js, PHP та інші. Сервера Heroku в основному використовуються на операційних системах Debian або Linux.

Платформа Heroku була заснована в 2008 році Джеймсом Лінденбаун, Адамом Віггінс та Оріон Генрі для роботи з проектами, створених на Rack. 10 січня 2010 року вона була куплена компанією Salesforce які зробили її своєю дочірньою компанією. У 2011 році Ю. Мацумото який був батьком мови програмування Ruby, перейшов до компанії в ролі інженера. Цього ж місяця було впроваджено підтримку Node.js та Clojure. Також платформа має підтримку таких баз даних, як Cloudant, Membase, Mongo та Redis.

Додатки, які працюють на цій платформі, використовують DNS - сервер (програми мають доменне ім'я типу «Ім'я_сервера.herokuapp.com»). Для таких програм виділяють малу частину віртуальних процесів, названі «dynos». Такі додатки розподіляються на спеціально розробленій віртуальній сітці, вона складається з декількох серверів. Платформа Heroku має власну систему контролю за оновленням версій.

2.2 Середовище розробки

2.2.1 Visual Studio Code

Visual Studio Code - це простий у використанні, але потужний редактор вихідного коду, який підтримує операційні системи такі як Windows, MacOS і Linux.[9]

IDE має підтримку великої кількості популярних мов програмування. Середовища виконання і розширення підтримувані IDE представлені нижче:

- JavaScript;
- TypeScript;
- Node.js;
- C ++;
- C #;
- Java;
- Python;
- PHP;
- Go;
- NET;
- Unity.

Для зручного використання програмних продуктів VS Code містить в собі вбудований відладчик, інструменти для роботи з Git-репозиторіями, підсвічування синтаксису, засоби для рефакторинга і IntelliSense (технологія автодоповнення, яка пропонує команду за першими літерами).

IDE VS Code була представлена 28 травня 2015 року компанією Microsoft в рамках конференції Blink. 19 жовтня 2015 року VS Code був відправлений у використання під ліцензію Массачусетського технологічного інституту.

Visual Studio Code був заснований на Electron – фреймворк який дозволяє використовуючи Python.py розробляти настільні додатки які працюють на движку Blink. Не дивлячись на це, що редактор створений на Electron, він не використовує редактор Atom. На заміну був реалізований вебредактор Monaco, розроблений для Visual Studio Online.

Інтерфейс IDE Visual Studio Code представлений на рисунок 2.2.

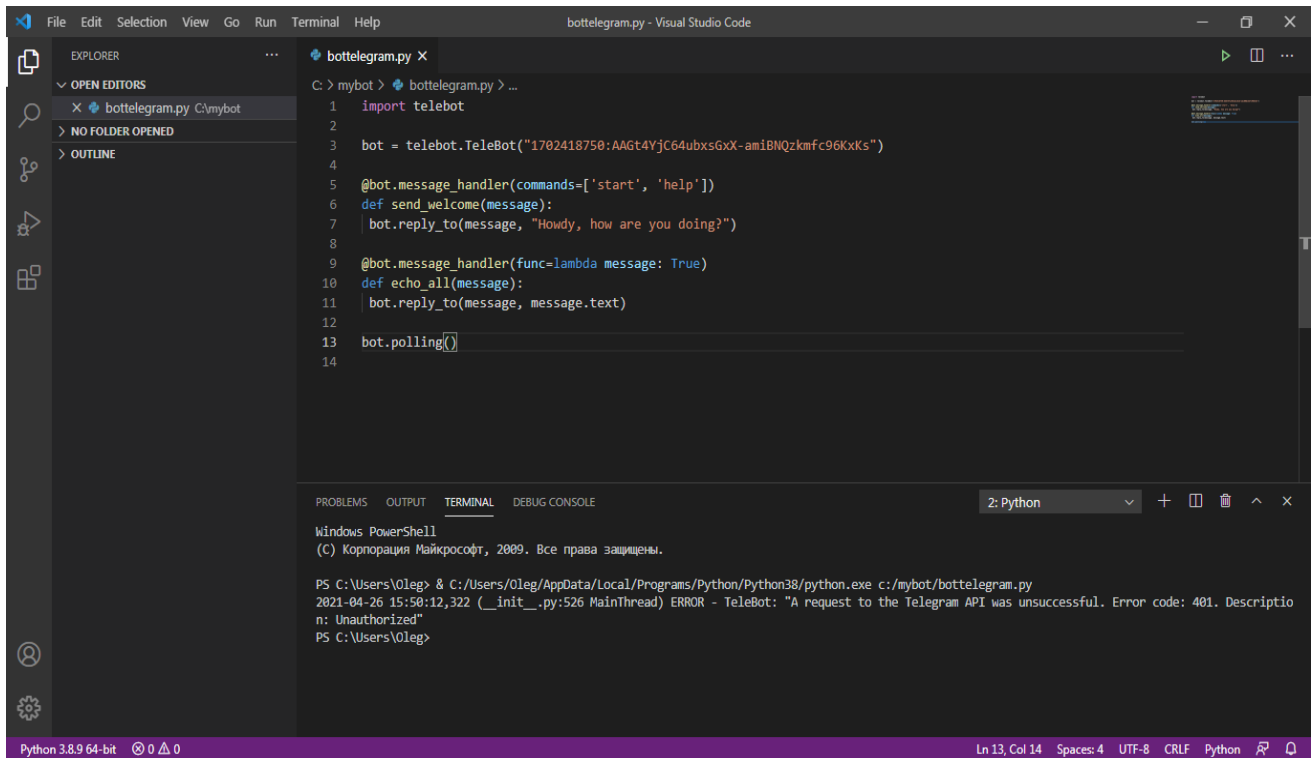


Рис. 2.2. - Інтерфейс IDE Visual Studio Code.

Перед початком написання коду були імпортовані такі бібліотеки:

- Configparser - ядро Python яке містить в собі модуль, який можна використовувати для створення роботи з файлами конфігурації
- Time - модуль Python, який використовується для вирішення завдань, пов'язаних з часом.
- Pathlib - покращує роботу з файлами, папками та поєднує в собі найкраще з модулів файлової системи Python - os, os.path, glob та інші.
- Pandas - вона надає дуже зручні з точки зору використання інструменти для зберігання даних і роботи з ними.
- Schedule – бібліотека для виконання коду який використовується, щоб підігнати під певний час або запускати його з певною періодичністю
- TelegramClient – збирає дані і повідомлення учасників чатів та каналів Telegram, а також зберігає цю інформацію у вигляді JSON-файлів.

3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Реєстрація бота чату для API Telegram Bot

Для початку роботи розробки програми потрібно зареєструватись в спеціальний чат-бот «BotFather». Реєстрація починається з команди `"/ Newbot"`, після чого потрібно ввести назву чату: `"Bot"` або `"_bot"` повинен бути вказаний в кінці, що є обов'язковою умовою. Після виконань всіх умов, BotFather видає ключ (спеціальний набір символів для доступу до HTTP API Telegram Bot) і URL-адресу доступу до чат-бота. Приклад реєстрації наведено на рисунку 3.1.

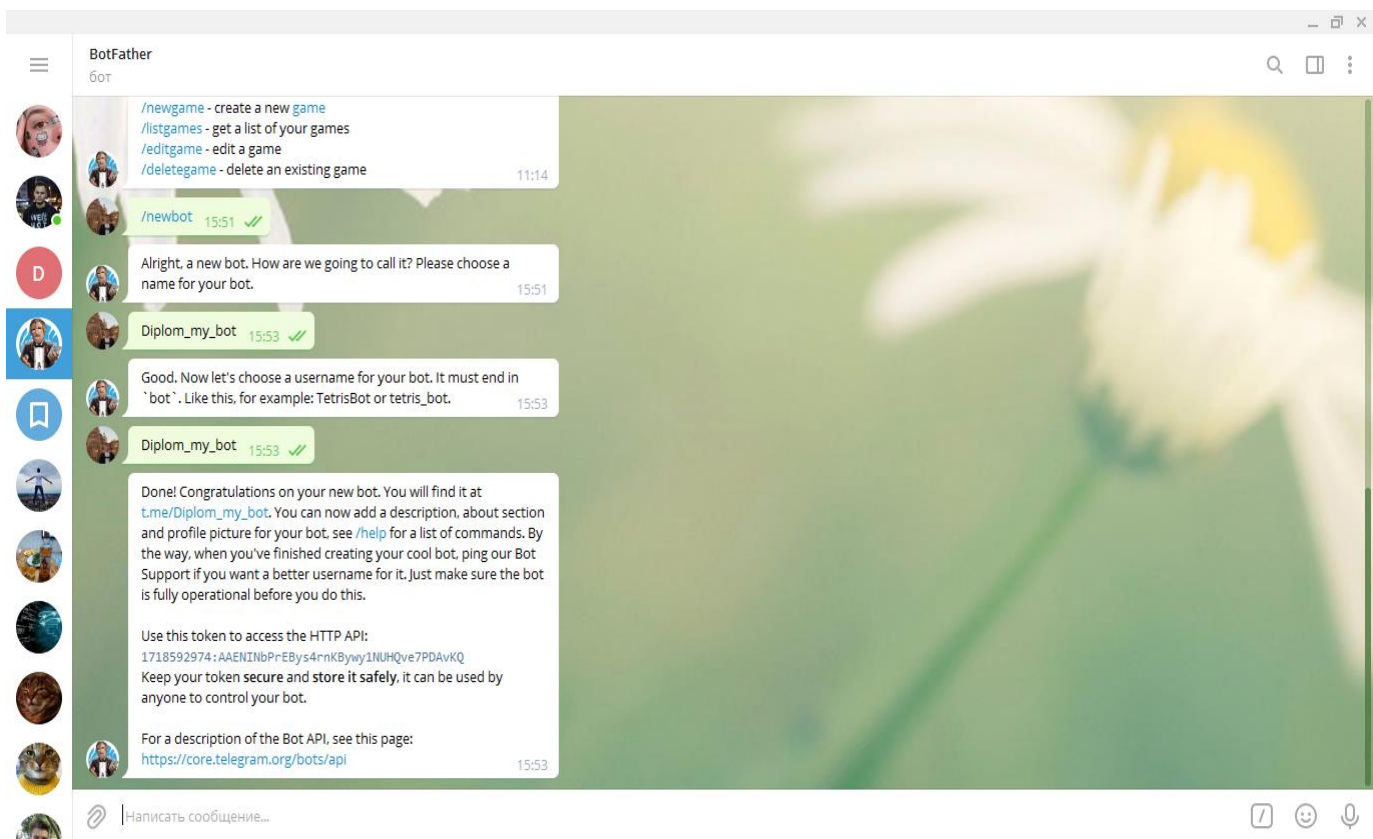


Рис. 3.1. – реєстрація чат боту в «BotFather»

Токен – це унікальний рядок із символів, який потрібний для того, щоб встановити справжність бота в системі. Токен генерується при створенні бота.[10]

Для встановлення додаткових параметрів: іконка чат-бота, повідомлення-вітання, опис чат-бота, а також можливість видалення існуючого бота існують наступні команди (Таблиця 1).

Таблиця 3.1 - Доступні команди для зміни чат-ботів

Команда	Опис
/setname	Зміна імені
/setdescription	Створює текст, який буде відображатись при першому відкритті чат-бота
/setabouttext	Зберігає текст в полі «Про чат-бота»
/setuserpic	Зберігає вибраний рисунок
/setcomands	Дозволяє створити список доступних команд
/deletebot	Видаляє вибраного чат-бота

Окрім команд на зміну основних параметрів чат-бота, існують команди, які дають змогу відобразити фіксовані параметри токєну, а також призначити значення, представлені в таблиці 2.

Таблиця 3.2. - Доступні команди для додаткової конфігурації чат-бота

Команда	Опис
/token	Повертає отриманий раніше токен у вибраного бота
/revoke	Анулює токен доступу до бота
/setinline	Включає чи вимикає можливість викликати бота з інших чатів
/setinlinegeo	Включає або вимикає можливість передавати місцєположення бота з іншого чату
/setinlinefeedback	Дає можливість отримати інформацію про кількість вибраних користувачами команд

Продовження таблиці 3.2

/setjoingroup	Визначає чи є можливість додати в групові діалоги бота
/setprivacy	Включає режим конфіденційності. В цьому режимі бот отримує, обробляє та відсилає назад інформацію окремо для кожного користувача в чаті

Отримавши налаштування та токен на стороні Telegram, можна розпочати розробку програмної частини чат-бота.

3.2 Реалізація клієнтської частини чат-бота

Telegram користувачі можуть спілкуватись з чат-ботами двома способами: команди («/start», «/help» та інші) з відповідними параметрами, або вбудованою клавіатурою (inline keyboards). Для швидкості та зручності користувачів було прийнято рішення зробити інтерфейс з вбудованою клавіатурою.

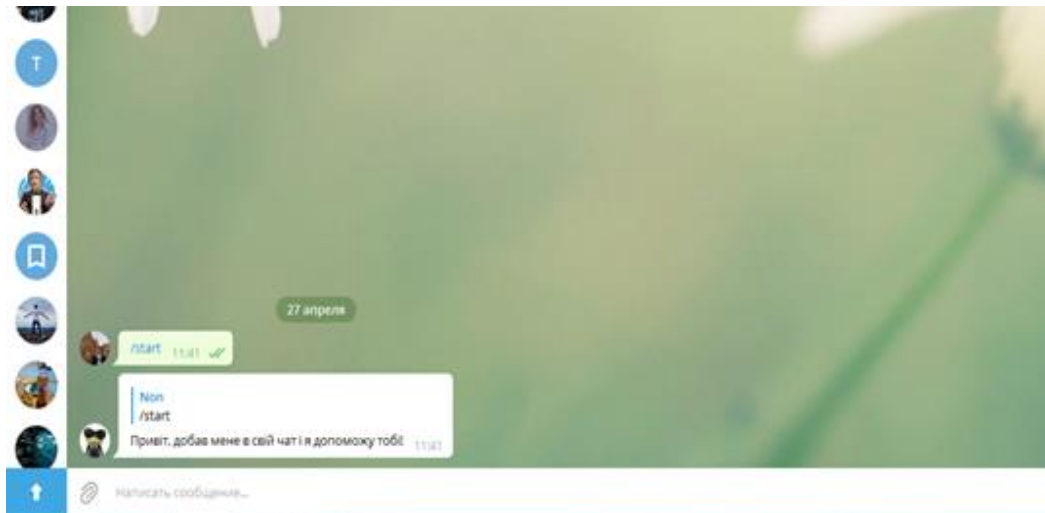


Рис. 3.2. – Запуск команди «/start»

Для реалізації поставлених цілей потрібно зробити меню в якому після першого запуску «/start» чат-бот буде вітатись, та пропонувати ознайомитись з функціями які в ньому запрограмовані, як показано на рисунку 3.2.

У Telegram месенджері клавіатура розроблена як об'єкт, а кнопки як JSON рядок. У інструкції Telegram Bot API написано, що всі клавіатури повинні мати обов'язковий параметр як назва кнопки (text), та п'ять необов'язкових - посилання (url), зворотні дані (callback_data), можливість вбудованого запиту (switch_inline_query), можливість виведення клавіатури з іншого чату (switch_inline_query_current_chat), та кнопки з можливістю покупки (pay). Приклад створення клавіатури з меню чат-бота зображений на рисунку 3.1.

```
import telebot

bot = telebot.TeleBot("1718592974:AAENINbPrEBys4rnKBywy1NUHQve7PDAvKQ")

@bot.message_handler(content_types=["text"])
def any_msg(message):
    keyboard = types.InlineKeyboardMarkup()
    callback_button = types.InlineKeyboardButton(text="Нажми на мене та дізнаєшся що я вмію", callback_data="test")
    keyboard.add(callback_button)
    bot.send_message(message.chat.id, "Мої функції: видалення по ключовим словам, видалення системних повідомлень", reply_markup=keyboard)
```

Рис 3.1. – Код з створенням клавіатури з меню чат бота.

Для того щоб чат-бот виконував свої функції в чаті, потрібно додати бота в сам чат, та надати йому права адміністратора.

Наприклад, коли новий користувач приєднується до чату або виходить з нього, системне повідомлення повідомляє про це в чат і це не завжди коректно, коли в діалозі між людьми приєднується новий користувач і це повідомлення з'являється в сотні людей. Тому наступною задачею було створення програмного коду який буде видаляти ці повідомлення. Було використано ряд основних функцій для видалення системних повідомлень:

- new_chat_members - нові учасники, які були додані до групи або супергрупу, і інформація про них
- left_chat_member - учасник видалений з групи, інформація про нього
- new_chat_title – назва чату була змінена на іншу
- new_chat_photo – фотографія чату була змінена
- delete_chat_photo – службове повідомлення: фото чата видалено
- group_chat_created – службове повідомлення: група створена

Приклад автовидалення ботом системного повідомлення про нового учасника чату на рисунку 3.3

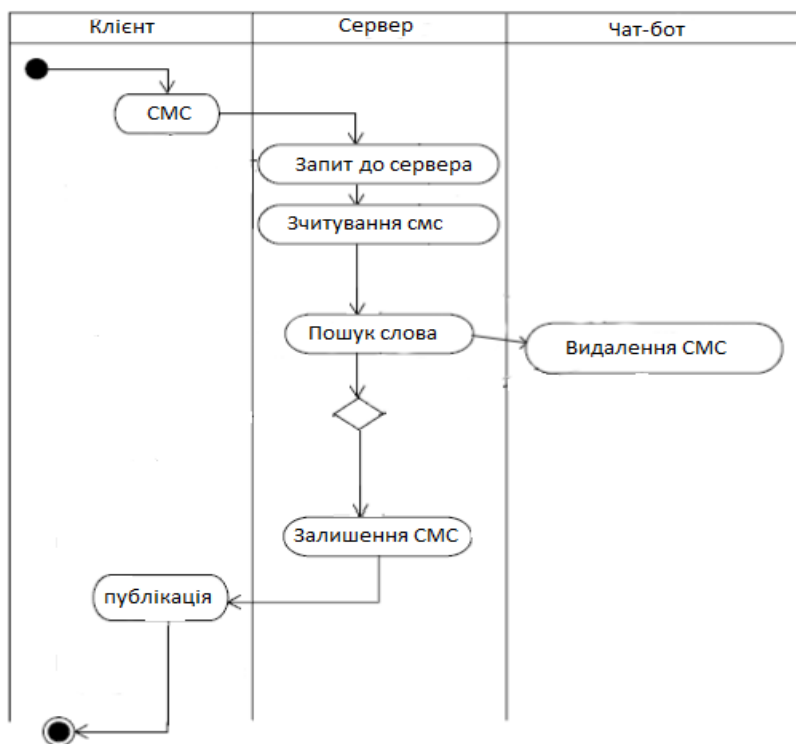


Рис. 3.3. – Видалення системних повідомлень ботом.

В чатах, де спілкуються користувачі месенджерів проходить багато різної та не потрібної інформації, такої як: спам не на тему розмови, реклама, погані слова які ображають інших співрозмовників. Тому наступним заданням було розробити спам-фільтр, який містить в собі ключові слова, які буде заборонено писати в чат. Для цього потрібно, щоб бот зчитував всі повідомлення які пишуть користувачі в чат і коли буде заборонене слово або адреси на сторонні групи, видаляв повідомлення. Для того, щоб бот працював коректно працював з забороненими словами була застосована функція «in message.text.lower():». За допомогою її бот зчитує текст написаний користувачем, та шукає ключові слова з фільтра.

Основною проблемою всіх рекламних агентств є автоматизація для безперервної реклами товарів та послуг по різних каналах та чатах. В наш час Telegram є одним з головних месенджерів для спілкування з друзями та колегами, через це вже існує безліч створених чатів та груп направлених на

рекламу. Тому було прийнято рішення створити функцію для бота для того щоб автоматизовано публікувати оголошення від імені рекламного агентства по різних чатам.

Telegram месенджер блокує повідомлення відправлені ботом по різним. Тому було прийнято рішення використати бібліотеку TelegramClient. Посилаючись на неї можна зробити так щоб бот автоматично підключався через сервер до акаунта користувача з якого будуть публікуватись оголошення. Для цього потрібно надано доступ для бота від акаунта.

Було створено файл формату Excel та підписано лист реклама». Далі записано в сам документ три функції:

- folder_name – ім'я з назвою телеграм месенджеру
- number_of_account – номер від акаунтf на який буде посилатись Telegram бот
- text_for_bot – текст який буде чат-бот використовувати для публікації

Внесення даних у функції зображено на рисунку 3.4.

	A	B	C
1	folder_name	number of account	text
2	1	+380 91 612 08 06	Здам квартиру 3х кімнатну в центрі ціна 1500\$
3			
4			
5			

Рис. 3.4. – Внесення даних в функції.

Для того, щоб бот автоматично приєднувався до груп, треба створити список посилань на ці чати та зберегти їх, для того, щоб бот міг зчитати та приєднатися до них. Для цього було створено Excel документ в який записано перелік чатів в які бот після того, як приєднається буде публікувати оголошення. Перелік посилань на ці групи записаних в документ на рисунку 3.5.

	A	B
1	группы	
2	dliia_orendu	
3	orenda_kvarturr	
4	kiev_kvartiraa	
5	orenda_kievv	
6		
7		
8		
9		
10		
11		

Рис. 3.5. – Посилання на чати в які має вступити бот.

Після успішного внесення даних, збережемо Excel документ під назвою Telegram.xlsx.

3.3 Розміщення чат-бота на віддаленому сервері Heroku.

Після реалізації програми, її слід розмістити на віддаленому сервері для того, щоб мати відкритий доступ до чат-бота. Платформа Heroku була обрана як об'єкт дослідження. Хостинг на хмарній платформі PaaS на базі Heroku нагадує розподілену систему контролю версій. Можна виділити такі способи його розгортання:

- відкрити Heroku Git та Container Registry за допомогою Heroku Command Line Interface (CLI), що спрощує розробку та керування програмами Heroku безпосередньо з термінала;
- пов'язати обліковий запис GitHub з автоматичним розгортанням;
- пов'язати обліковий запис з автоматичним розгортанням Dropbox.

Для розгортання чат-бота були зроблені такі кроки:

- реєстрація на платформі Heroku Cloud SaaS;
- завантаження та встановлення Heroku CLI;
- інтеграція в термінал операційної системи або в IDE виконаний вхід, в обліковий запис Heroku за допомогою запису команди «heroku login»;

- клонування сховища командою «`heroku git: clone -a <ADD_NAME>`» з віддаленого сервера Heroku Git на локальний комп'ютер;
- заміну в коді можна здійснити за допомогою команд "`git add`". та "`git commit -am <COMMIT_NAME>`";
- всі зроблені зміни були передані на віддалений хмарний сервер Heroku за допомогою команди "`git push heroku master`". Після того, як було правильно виконано всі вищезазначені команди в правильному порядку розпочинається завантаження даних на віддалені сервери. Після успішного розгортання, на терміналі з'явиться довідкова інформація про стан програми та режим доступу до неї (рисунок 3.6)

```
remote: ----> Build succeeded!
remote: ----> Discovering process types
remote:   Procfile declares types   -> (none)
remote:   Default types for buildpack -> web
remote:
remote: ----> Compressing...
remote:   Done: 19.4M
remote: ----> Launching...
remote:   Released v3
remote:   https://sfutelegrambotdip.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sfutelegrambotdip.git
* [new branch]      master -> master
```

Рис. 3.6. - Інформацію про стан програми та режим доступу

Таким чином, чат-бот був розгорнутий на віддаленому сервері хмарних обчислень Heroku, що дозволяє користувачам легко отримувати доступ та безперервна роботу.

4 ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

Тестування працездатності відбувалось після кожного етапу розробки за підготовленими тест-кейсами. Тест-кейси були складені заздалегідь, при визначенні цілей розробки та інструментів, які використовувались при розробці.

Основні тест-кейси:

- а) коректна обробка повідомлень від користувачів в чаті та робота з ними.
- б) коректна робота з видаленням системних повідомлень;
- в) робота з ключовими словами та видаленням їх;
- г) автоматичне приєднання до чатів та груп та опублікування оголошень;

Тестування чат боту відбувалось на операційній системі Windows 7.

Технічні характеристики:

- а) операційна система: Windows 7 з версією месенджера ;
- б) оперативна пам'ять: 4 ГБ;
- в) вбудована пам'ять: 500 ГБ.
- г) Версія месенджера Telegram 2.7.4.

Після початку чату з ботом на екрані з'являється опис та команда «/Start» .

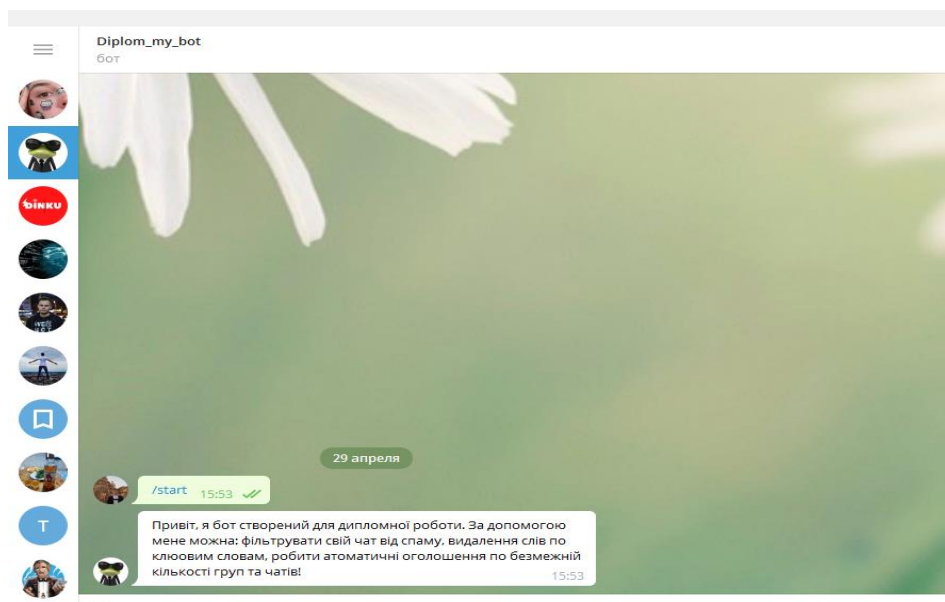


Рис. 4.1. – Натискання старт та інформація про бота

При натисканні на клавішу «/Start» отримуємо основну інформацію, про роботу боту та його функції (рисунок 4.1).

Наступним кроком пишемо команду «/help», після цього бот дає інструкцію по роботі з ним (рисунок 4.2).

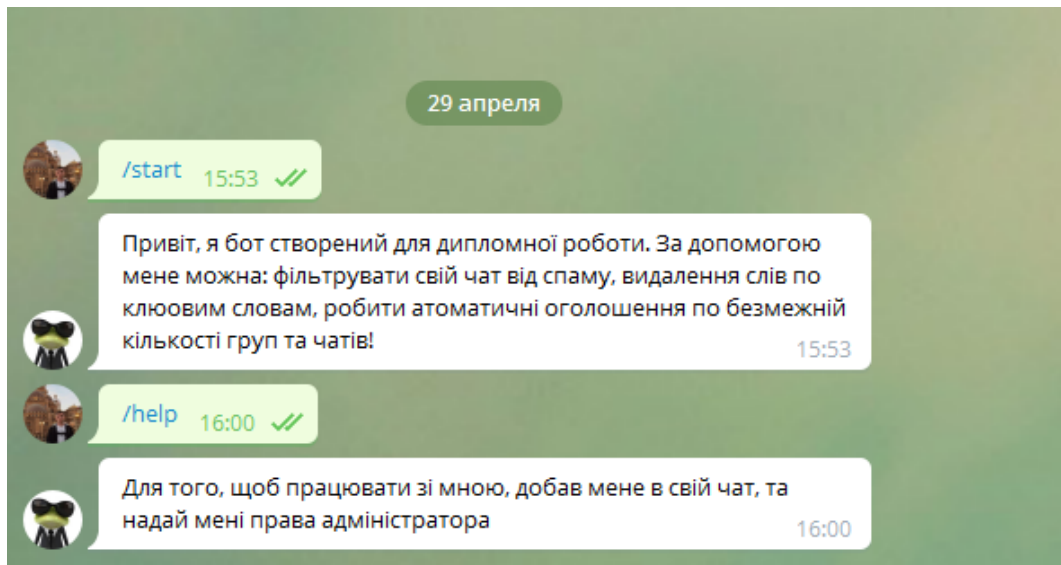


Рис. 4.2 – Інструкція по роботі з чат-ботом.

Далі створюємо власний чат, щоб перевірити та протестувати як бот справляється з видаленням системних повідомлень та роботою з ключовими словами.

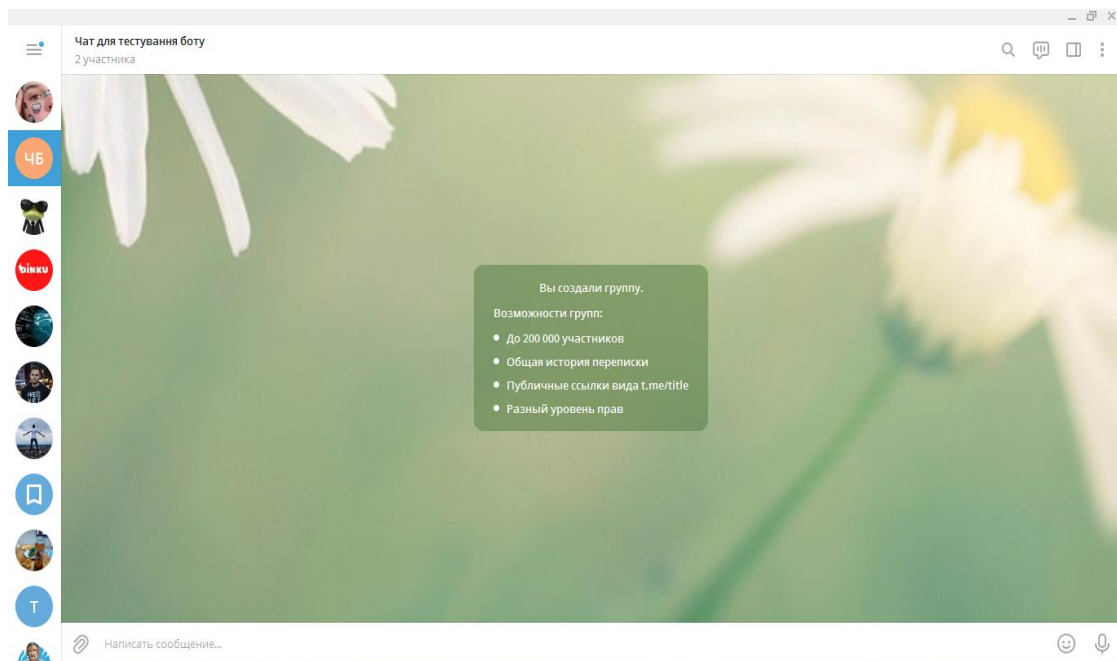


Рис. 4.3. – Інтерфейс створеного чату.

Для створення власного чату потрібно перейти в головне меню акаунту, та вибрати пункт «Створення чату» та дати назву для нього (рисунок 4.3)

Для того щоб, бот міг працювати з чатом, та керувати повідомленнями, його потрібно додати до чату, та надати йому права адміністратора (рисунок 4.4).

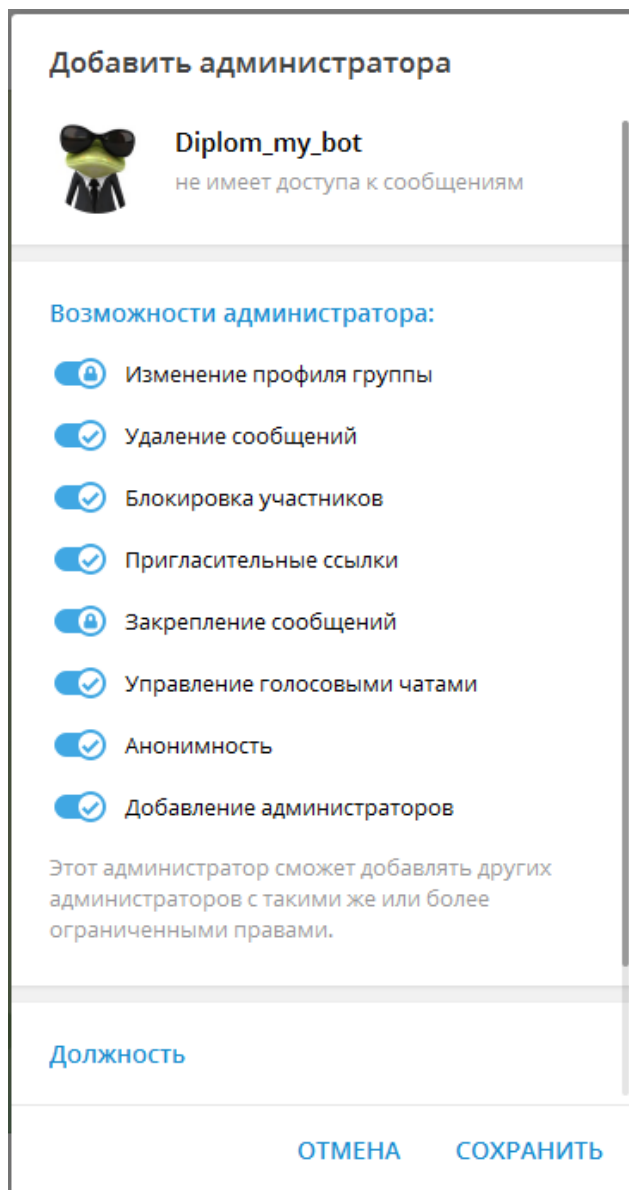


Рис. 4.4 – Надання боту прав адміністратора чату.

Далі перевіряємо бота на видалення системних повідомлень, таких як «Приєднався до чату», «Видалився з чату», «Зміна аватару в чаті» та інші.

Успішне видалення системних повідомлень можна побачити в меню «Дії з чатом», там можна переглядати історію за остані 3 доби які відбувалися з чатом (рисунок 4.5).

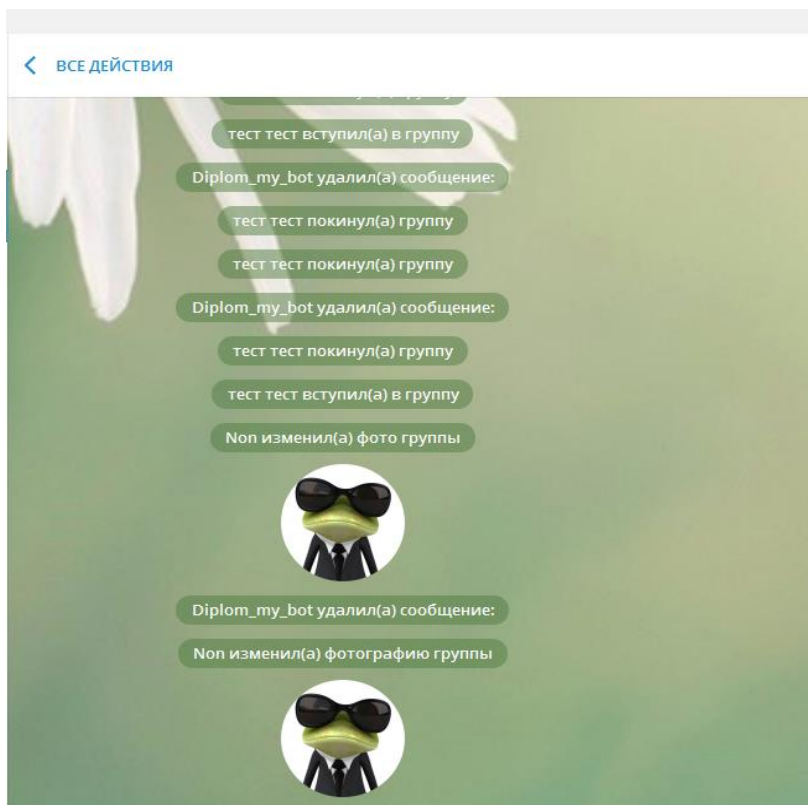


Рис.4.5. – Історія видалення повідомлень з чату.

Перевіряємо бота на видалення ключових слів. Занесемо до забороненого списку таке слово як «СПАМ» та напишемо слова окремим повідомленням та в реченні (рисунок 4.6).

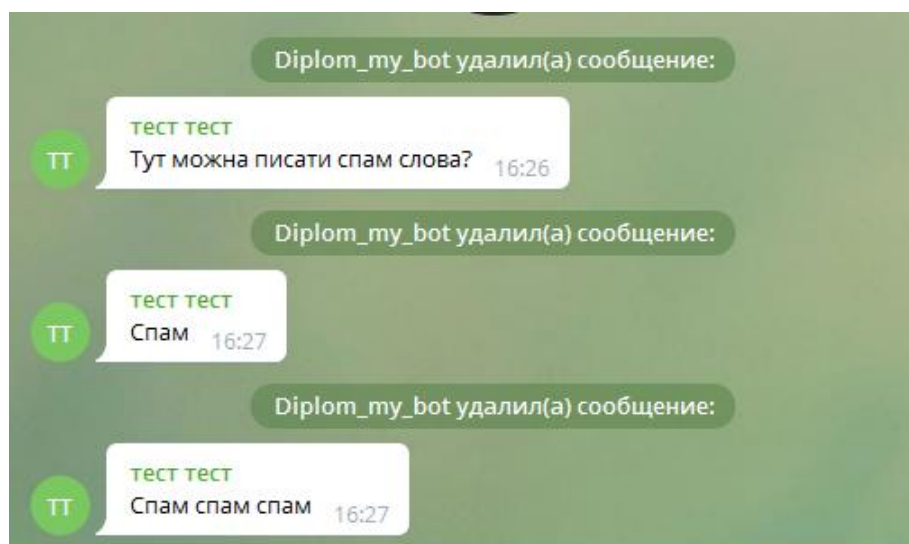


Рисунок 4.6. – Видалення ключових слів.

Далі перевіряємо роботу з приєднанням до чатів які знаходяться в Excel документі. Для цього пишемо команду в діалозі «/spam».

Бот автоматично підєднається до серверу та запустить MyBot.exe файл, після того потрібно надати доступ для нашого боту який автоматично приєднається до нашого акаунту.

Після написання команди в головному меню нашого бота команди «/spam» автоматично відкриється вікно з підтвердженням доступу до нашого акаунту. Для підтвердження стару натискаємо «1» (рисунок 4.7).

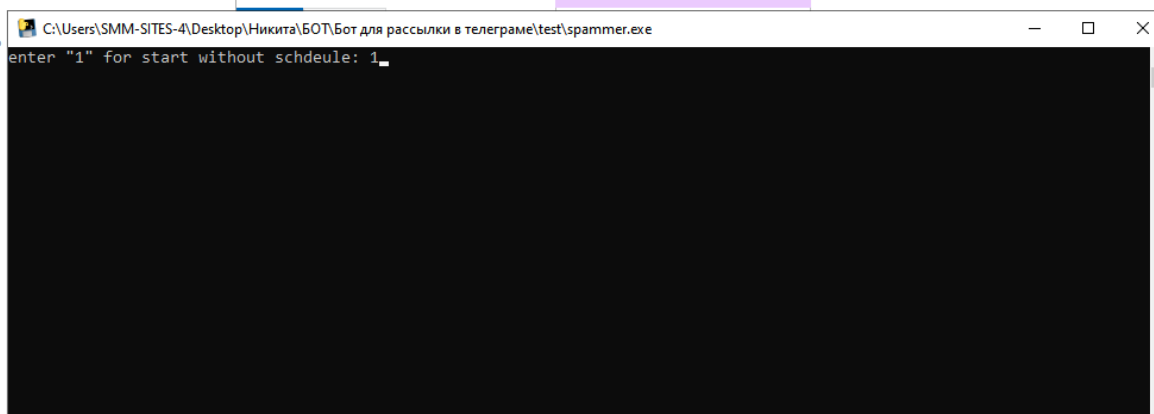


Рис. 4.7. – Підтвердження запуску програми.

Потрібно надати доступ для входу в наш акаунт, введемо код який прийшов в Telegram месенджер (рисунок 4.8)

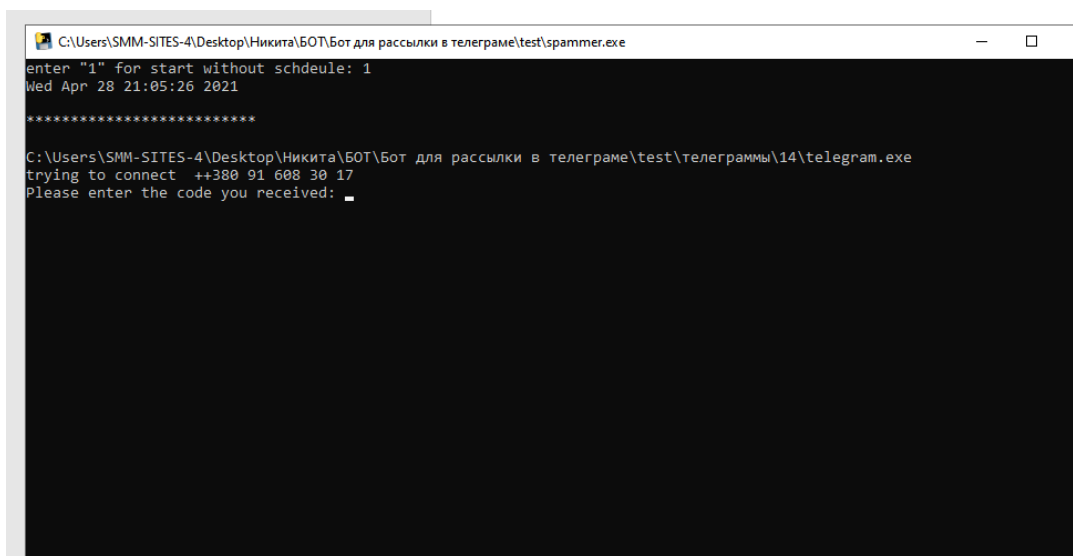


Рис. 4.8. – Підтвердження входу в Telegram акаунт.

Шестизначний код приходить на номер телефону, який був вказаний в функції «number of account». Після введення коду, натискаємо Enter.

Після успішної авторизації бота до нашого акаунту, автоматично було приєднано до груп які були вказані в excel документі з нашим переліком груп та чатів (рисунок 4.9).

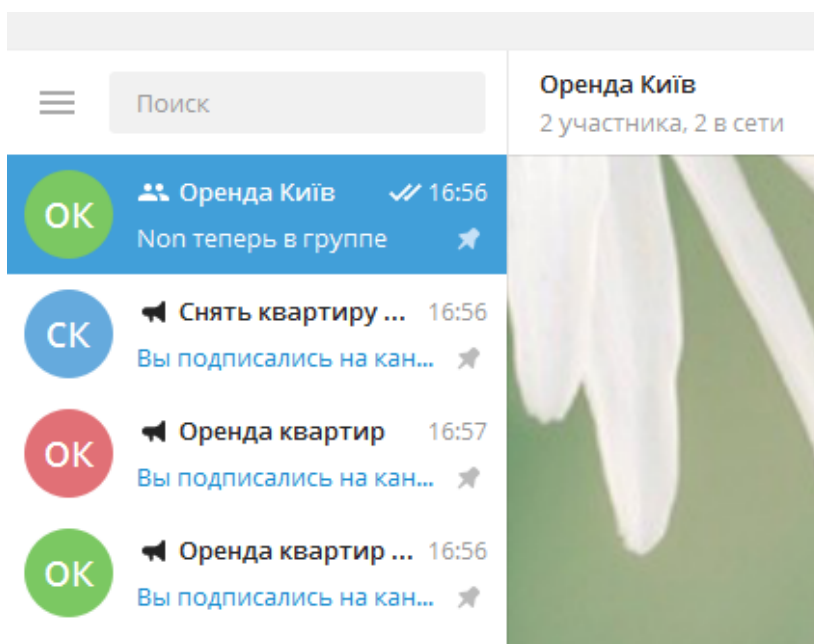


Рис. 4.9. – Автоматичне приєднання до груп та чатів.

Далі можна побачити, що після того як бот пройшов цикл приєднання до груп, він автоматично розпочав писати в чати та групи оголошення яке ми зазделегідь записали в файл (рисунок 4.10).

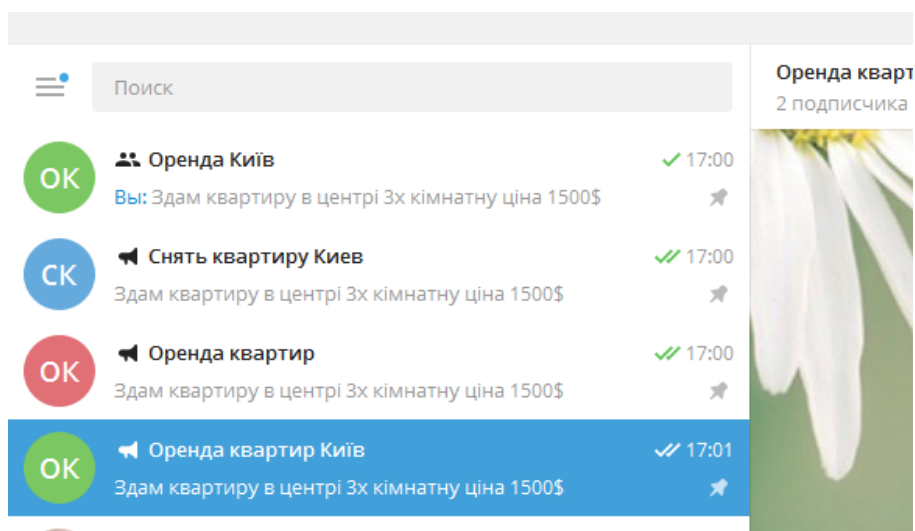


Рис. 4.10. – Автоматичне опублікування оголошень в чати та групи.

Текст оголошень бот зчитує з функції «text» та відправляє по групам та чатам.

Після того як бот проходить всі групи та чати, вмикається зворотній таймер на 60хвилин, після цього часу буде проведено автоматичне опублікування оголошень в чатах. Для завершення роботи з опублікуванням, потрібно написати в діалоговому вікні з ботом команду «/stop» (рисунок 4.11).

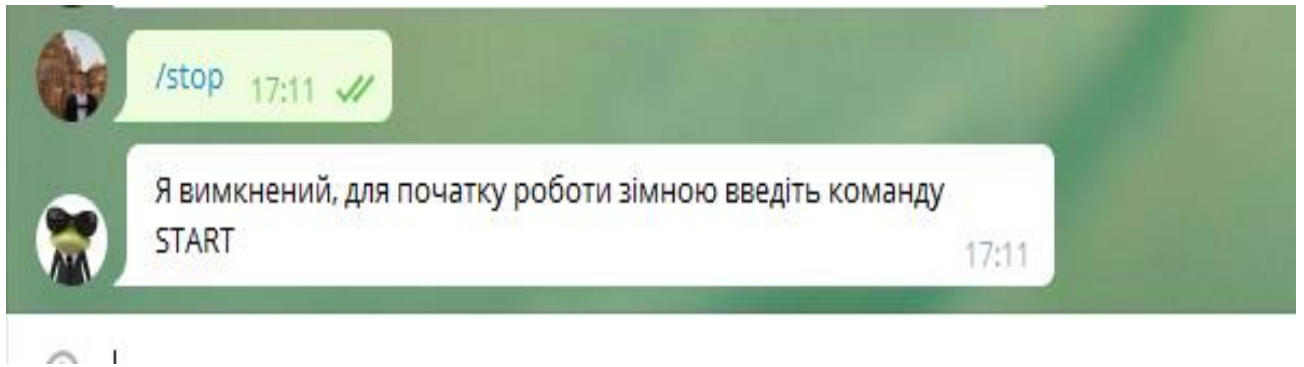


Рис. 4.11. – Вимкнення роботи бота.

Після успішного вимкнення командою «/stop» бот перестав працювати з адмініструванням чатом, та припинив публікування оголошення.

ВИСНОВОК

У наш час репутація месенджерів як засобу спілкування постійно росте і розвивається. Компанії, сім'ї та друзі щодня здійснюють спілкування в аудіо- або текстовому форматі на відстані без жодних обмежень. Найстрімкіше розвиваються такі програмні продукти, як чат-боти, які працюють на платформі месенджерів.

Підтримка обміну користувачами в режимі реального часу, перетворення даних, замовлення прокату автомобілів, завантаження файлів, пошук необхідної інформації та багато іншого тепер можна використовувати в одному месенджері. Для цього не потрібно завантажувати безліч різних додатків на свій смартфон, які будуть займати більшу частину пам'яті телефону. Тому що вже існує месенджер з необхідною кількістю чат-ботів.

В ході випускної кваліфікаційної роботи були виконані всі поставлені задачі. Спочатку було вивчено месенджери, проведена робота над з'ясуванням переваг та недоліків, їх порівняння та оцінювання. У результаті найзручнішим для використання виявився месенджер Telegram. Він є найбільш простий та зрозумілий на основі даних API Telegram Bot. Було проаналізовано аналоги чат-ботів на платформі Telegram та виявлено їх переваг та недоліки та цікаві рішення розробників цих ботів. На основі цього було прийнято рішення для розробки авторського чат-боту, для управління контентом рекламної агенції, за допомогою якого можна слідкувати за чистотою власних чатів, та автоматично проводити опублікування оголошень. Для виконання даних задач, були вибрані середовища для розробки чат-бота серед яких Python, VS Code, Heroku.

Таким чином, результатом випускної кваліфікаційної роботи є повністю реалізований чат-бот, за допомогою якого можна з легкістю керувати власними чатами, та автоматично розміщувати оголошення в необмеженій кількості чатів та груп.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Топ-10 корисних Telegram-ботів для українських користувачів: URL: <https://www.epravda.com.ua/publications/2017/08/7/627822>
2. Telegram Bot API: URL: <https://core.telegram.org/bots/api>
3. Марк Лутц, Learning Python :2019, 720с.
4. Telegram FAQ: URL: <https://telegram.org/faq>.
5. Telegram-bot: URL: <https://habr.com/ru/post/442800/>
6. Телеграм бот за допомогою TeleBot: URL: <https://habr.com/ru/post/448310/>
7. Git: URL: <https://uk.wikipedia.org/wiki/Git>
8. Python Requests : URL : <https://2.python-requests.org/en/master/>
9. Огляд протоколу HTTP :
URL:<https://developer.mozilla.org/ru/docs/Web/HTTP/Overview>
10. Черга повідомлень : URL: <https://aws.amazon.com/ru/message-queue/>
11. What is REST: URL: <https://restfulapi.net/>
12. REST : URL: <https://habr.com/ru/post/351890/>
13. JSON: URL: <https://www.json.org/json-ru.html>
14. Тестування телеграм-бота: URL: <https://habr.com/ru/post/322816/>
15. What is Artificial Intelligence :URL:
<https://www.techopedia.com/definition/190/artificial-intelligence-ai>
16. Subproces management: URL: <https://docs.python.org/3/library/subprocess.html>

ДОДАТОК А

Програмний код

```
import configparser
import time
import telebot
from pathlib import *
import pandas as pd
import schedule
from telethon.sync import TelegramClient
from telethon.tl.functions.channels import JoinChannelRequest

bot =
telebot.TeleBot("1718592974:AAENINbPrEBys4rnKBywy1NUHQve7PDAvKQ")

@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.send_message(message.chat.id, "Привіт, я бот створений для дипломної роботи. За допомогою мене можна: фільтрувати свій чат від спаму, видалення слів по ключовим словам, робити атоматичні оголошення по безмежній кількості груп та чатів! ")

@bot.message_handler(commands=['stop'])
def send_welcome(message):
    bot.send_message(message.chat.id, "Я вимкнений, для початку роботи зімною введіть команду START")

@bot.message_handler(content_types = ['new_chat_members', 'left_chat_member', 'new_chat_title', 'new_chat_photo', 'delete_chat_photo', 'group_chat_created', 'voice_chat_scheduled'])
def delete(message):
    bot.delete_message(message.chat.id, message.message_id)

@bot.message_handler(content_types=['text'])
def filter(message):
    if "Привет".lower() in message.text.lower() or "спам".lower() in message.text.lower() or "Что".lower() in message.text.lower():
        bot.delete_message(message.chat.id, message.message_id)

bot.polling()

config = configparser.ConfigParser()
config.read("config.ini")
```

```

def main():
    print(time.asctime())
    api_id = config["config"]["api_id"]
    api_hash = config["config"]["api_hash"]
    df = pd.read_excel("Telegram.xlsx",
                      sheet_name="реклама", dtype={'number of account': str,
'folder_name': str})[::-1]
    df_groups = pd.read_excel("Telegram.xlsx",
                              sheet_name="группы")

    count = 0
    for group in df_groups['группы'].dropna().tolist():
        for i, number in enumerate(df['number of account'].dropna().tolist(), start=0):
            print()
            print("*****")
            print()
            try:
                print(f" {Path.cwd()}\\телеграм\\{df['folder_name'].dropna().tolist()[i]}\\telegram.exe
")
                except:
                    continue
                print(f"trying to connect  +{number}")
                try:
                    client = TelegramClient(f"+{number}", api_id, api_hash)
                except ValueError as e:
                    return print(e)
                try:
                    client.start(f"+{number}")
                except:
                    print(f"can`t connect to +{number}")
                try:
                    with client:
                        try:
                            print(group)
                            client(JoinChannelRequest(channel=group))
                        except:
                            print("can`t join to group")
                        try:
                            if type(df["text"].tolist()[i]) == float:
                                continue
                            client.send_message(group, df["text"].tolist()[i])
                        except:
                            print("неможливо відправити повідомлення")

                except:
                    continue
                time.sleep(1)

```

```
    count += 1
    print(time.asctime())

if __name__ == '__main__':
    if input("enter \"1\" for start without schdeule: ") == "1":
        main()
    print("\nstarted schedule spam")
    print(time.asctime())
    schedule.every(60).minutes.do(main)
    while True:
        time.sleep(1)
        schedule.run_pending()
```

ДОДАТОК Б

Презентація



Державний
Університет
Телекомунікацій

РОЗРОБКА ТЕЛЕГРАМ БОТУ ДЛЯ УПРАВЛІННЯ КОНТЕНТОМ РЕКЛАМНОЇ АГЕНЦІЇ

Виконав студент: Левицький О.Л. ППЗ-52

Керівник: Щербина І.С.

1

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА ДОСЛІДЖЕННЯ

Об'єкт дослідження – процеси автоматизації та управління контентом рекламної агенції

Предмет дослідження – методи проектування та технології створення власного чат-боту для автоматизації процесів пов'язаних з автоматичним управлінням чату та публікування оголошень

Мета роботи – створення чат-боту для управління контентом рекламної агенції

3

АКТУАЛЬНІСТЬ

Переваги чат-ботів:

- простий доступ до інформації;
- зрозумілий інтерфейс;
- мінімальне використання трафіку;
- велика швидкість відповіді;
- немає необхідності скачувати та встановлювати додаткові додатки;
- автоматизація діяльності.

5

Аналоги чат-ботів

	ZukuBot	Combot	ChatKeeperBot
Ведення статистики	-	-	+
Складний у використанні	-	+	-
Спам реклама	-	-	+
Мобільна версія	+	+	-
Робота з іншими ботами	-	-	-
Безкоштовний	+	+	-

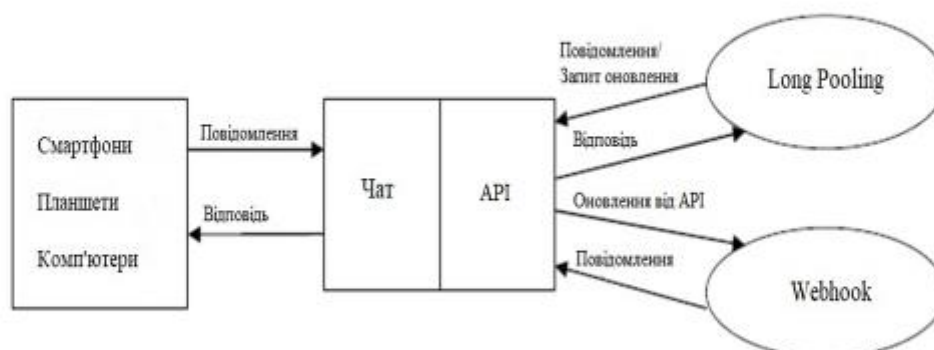
7

Вимоги до чат-бота

- Автоматичне видалення системних повідомлень
- Автоматичне видалення заборонених слів
- Автоматичне приєднання до груп та чатів
- Автоматичне публікування оголошень

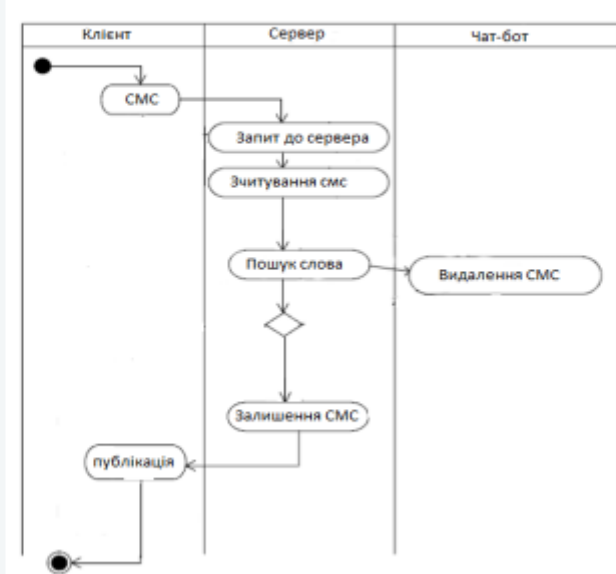
9

ПРИНЦИП РОБОТИ ЧАТ БОТА



11

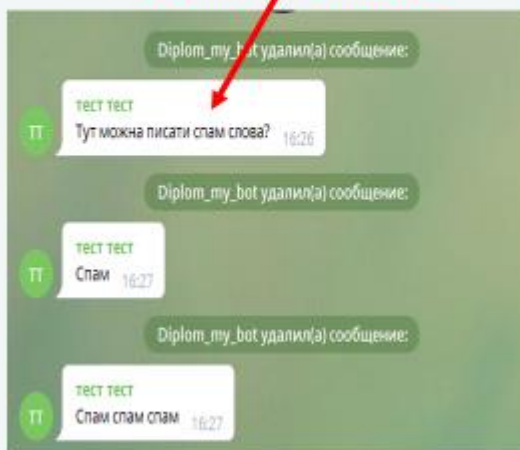
Діаграма роботи з чатом



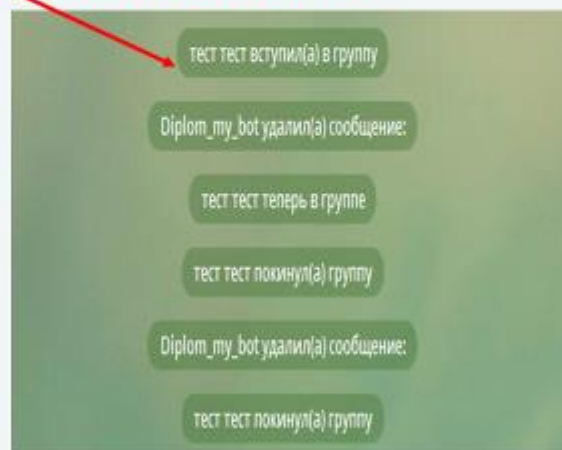
13

АВТОМАТИЗОВАНА РОБОТА З ЧАТОМ

Видалення спам слів



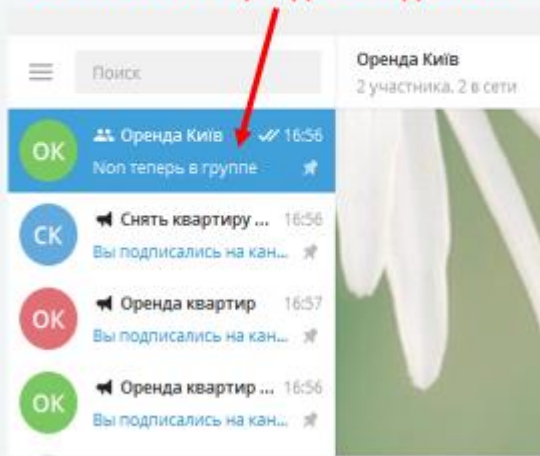
Видалення системних повідомлень



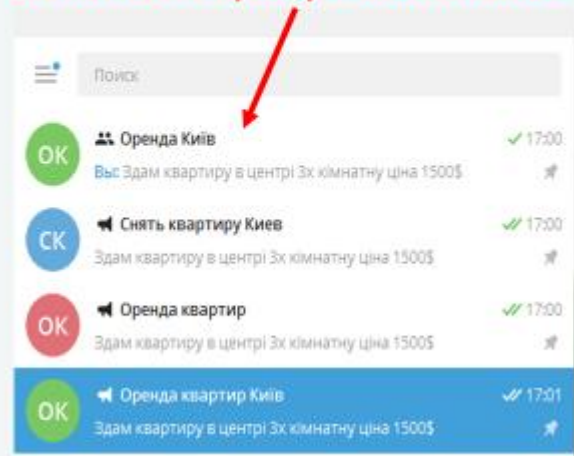
15

АВТОМАТИЗОВАНА РЕКЛАМА

Автоматичне приєднання до чатів



Автоматичне публікування оголошень

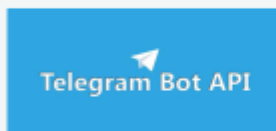


17

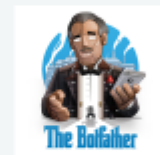
ІНСТРУМЕНТИ ТА СЕРЕДОВИЩЕ РОЗРОБКИ

Клієнтська частина:

TelegramBot API



FatherBot



Серверна частина:

Python



Heroku



19

ВИСНОВКИ

Протягом виконання дано роботи були виконані такі цілі та задачі:

- Визначення вимог до розроблюваного бота
- Вибір технології та середовища розробки
- Представлена діаграма роботи з чатом
- Представлена розробка бота

21

ДЯКУЮ ЗА УВАГУ!

23