**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВИЙ ПРОЕКТ**

з дисципліни “Бази даних”

спеціальність 121 – Програмна інженерія

на тему: **Моніторингова система Telegram-каналу**

**Студент групи** **КП-61 Садрицький С.В.**  \_\_\_\_\_\_\_\_ (підпис)

**Викладач к.т.н,**

**доцент кафедри**

**СПіСКС Петрашенко А.В.** \_\_\_\_\_\_\_\_ (підпис)

Київ – 2019

**Анотація**

У даній курсовій роботі була створена моніторингова система система для аналізу даних каналу в месенджері Телеграм. Галуззю застосування даної розробки єсоціологія та маркетинг.

Були отримані реальні дані з вибраного для прикладу Telegram-каналів КПІ-спільнот kpilive, kpi\_777 за допомогою бібліотеки Python, що використовує API Telegram під назвою Telethon[[2]](https://telethon.readthedocs.io/en/latest/). В ході роботи було отримано масив даних, що містять такі числові параметри як кількість пілписників каналу, постів та переглядів для відповідних постів, а також інші метрики для проведення подальшого аналізу і обробки. Усі дані були поміщені у базу даних. Далі методами фільтрації і аналізу система одержує потрібні результати, оформлюючи їх у текстовому та графічному вигляді.

У основному модулі програми відбуваєтьсязбір, фільтрація та аналіз активності на каналі, простежуються зв’язки між активністю та днями, а також такі характеристики, як лексичний вміст постів. В даниому проекті продемонстровано лише частину функціоналу, який може містити моніторингова система такого напряму , щоб надавати стислий і інформативний звіт про зацікавленість аудиторії каналом відповідним заінтересованим особам у процесі обробки даних.

**ЗМІСТ**

Зміст

[**Зміст 3**](#_gjdgxs)

[**Вступ 4**](#_30j0zll)

[**1.**](#_1fob9te) **Аналіз інструментарію для виконання курсового проекту 6**

[**2.**](#_3dy6vkm) **Структура бази даних 7**

[**3.**](#_4d34og8) **Опис программного забезпечення 8**

[**3.1.**](#_2s8eyo1) **Загальна структура ПЗ 8**

[**3.2.**](#_17dp8vu) **Опис модулів 8**

[**3.3.**](#_3rdcrjn) **Опис основних алгоритмів роботи 9**

[**4.**](#_26in1rg) **Аналіз функціонування засобів масштабування 10**

[**5.**](#_lnxbz9) **Опис результатів аналізу предметної галузі 11**

[**Висновки 12**](#_35nkun2)

[**Список використаних джерел 13**](#_1ksv4uv)

[**Додатки 14**](#_2jxsxqh)

# **Вступ**

Була створена моніторингова система telegram-каналу на прикладі каналів КПІ-спільнот у Телеграмі.

**Актуальність.** На даний момент розвитку технологій, зокрема швидкого розповсюдження і перспектив інтернет-маркетингу у світі використовується на повну арсенал такого нині популярного месенджера і за сумісництвом агрегатора інформації, як Телеграм. На відміну від традиційних соцмереж та застарілих форм передачі інформації Телеграм відчутно зручніший через його спосіб подачі інформації і модливості для рекламодавців та соціологів. Вони, в свою чергу, маючи доступ до даних аудиторії, зможуть розуміти більш ясно, спільноти яких телеграм каналів зможуть стати їх потенційними клієнтами і, зрештою, будуть більш точно прогнозувати свої продажі. Даний проект призначений для збору і обробки інформації про аудиторію телеграм- каналу для подальшого використання зацікавленими особами.

**Мета розробки**:створення програмного забезпечення, що забезпечить роботу наведених далі пунктів:

1. Попередня обробка даних

* Засоби генерації даних.
* Розроблення утиліти для збору інформації про активність на каналі за допомогою API Telethon [[1]](https://telethon.readthedocs.io/en/latest/)
* Встановлення зв’язку між даними, отриманими із API і розміщення їх у колекціях бд.
* Засоби фільтрації і валідації даних. Розроблення додаткового функціоналу у вищезазначеній утиліті задля корегування отриманих даних та переходу до їх подальшої обробки та структуризації.

1. Основний модуль: фільтрація за допомогою Aggregation Framework та аналіз , кількість постів за певний проміжок часу, зокрема, у середньому поденно за тиждень, середня кількість відвідувань каналу за певний проміжок часу, встановлення найвживаніших слів у постах на каналі.
2. Забезпечення масштабування та реплікації даних: шардинг та створення репліка сетів у Mongo DB

# **Аналіз інструментарію для виконання курсового проекту**

*Обґрунтування вибору СКБД*

Була обрана найпопулярніша серед нереляційних система керування базами даних – MongoDb, документо-орієнтована СКБД із відкритим кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Вибір нереляційної СКБД обгрунтовується тим, що MongoDB надає можливість у короткі строки розробити прототип системи і посприяє набуттю навичок роботи із NoSQL базами даних.

*Обґрунтування вибору мови програмування*

Мовою програмування було обрано Python, оскільки вона є одним із лідерів по використанню у Data Science, активно розвивається та має значний арсенал бібліотек для збору, аналізу та прогнозування даних.

*Обґрунтування вибору бібліотек*

Telethon - для роботи з API Telegram і отримання даних про канали

* NumPy - для роботи із великими масивами даних
* Matpotlib – візуалізація у вигляді графіків та діаграм
* ntlk – обробка текстових даних, зокрема встановлення ключових слів
* dotenv - для зберігання змінних оточення, таких як ключі до API
* asyncio - для забезпечення асинхронної обробки запитів через Telethon
* pymorphy - для морфологічної обробки слів(зокрема приведення слів до нормальної форми при аналізі ключових)
* Pymongo – взаємодія із базою даних

# **Структура бази даних**

# База даних складається з колекції повідомлень проаналізованих каналів.

* 1. **Колекція Telegram каналів**

Була отримана шляхом запитів через Telethon.

Колекціяь messages зберігає дані про канали, що було проаналізовано. Містить наступні поля:

* \_id– ідентифікатор запису, унікальне поле у базі даних;
* date - дата посту
* message – рік заснування
* views – кількість переглядів поста
* сhannel – назва каналу, на якому був опублікований пост
* media - чи є у пості медіафайл, як-то зображення чи аудіо тощо

Графічне представлення бази даних можна побачити на рис. 1 (додаток А).

# **Опис программного забезпечення**

# **Загальна структура ПЗ**

Розроблене ПЗ має відносно просту структуру, що складається з таких файлів, кожен з яких репрезентує логічний модуль системи:

* **main.py**Даний файл відповідає за запуск програмного забезпечення і організацію взаємодії із ним користувача.
* **analysis.py**

Даний файл відповідає за виконання аналізу даних після іх фільтрації а також за візуалізацію у вигляді графіків та діаграм

* **filter.py**

Даний файл відповідає за фільтрацію даних методами Aggregation Framework.

* + - **parser.py**

Даний файл відповідає за збір даних для аналізу .

* + - **database.py**

Даний файл відповідає за взаємодію з БД, імпорт та експорт колекції даних .

# **Опис модулів**

У програмі можна виділити декілька основних логічних модулів, що мають певну самостійність і обмінюються один з одним даними. У програмі є такі основні модулі: модуль генерації даних, основний модуль, модуль роботи з графіками та модуль бази даних. Взаємодія між модулями відбувається за допомогою виклику методів.

* + 1. **Модуль бази даних**. Завданням цього модулю є встановлення зв’язку з базою даних та запис і зчитування даних із бази даних, фільтрування отриманих даних, їх збір у відповідну для відображення форму та для побудови графіків, що відображають проаналізовані дані.
    2. **Основний модуль**. У функції даного модулю входить відображення GUI і організація взаємодії через нього із користувачем.
    3. **Модуль генерації даних**. Даний модуль відповідає за збір даних за допомогою спеціалізованого прикладного програмного інтерфейсу(API).
    4. **Модуль роботи з графіками** реалізовує вивід графіків аналітики в інтерактивному вигляді за допомогою бібліотек numpy, pandas та matplotlib.

# **Опис основних алгоритмів роботи**

Алгоритми реалізовані з використанням відкритих бібліотеках Python, а саме numpy, matplotlib, pandas, requests, stop\_words, npyscreen.

* + 1. **Алгоритм побудови графіків** було реалізовано за допомогою математичного пакету matplotlib та її підсистеми pyplot. Цей алгоритм задає назву графіку, встановлює панель управління, вказує атрибути відповідні за осі Х та У, підписує їх та виводить на екран. Всю внутрішню реалізацію містять у собі функції математичного пакету matplotlib.
    2. **Алгоритм збору даних**, що описують канали реалізовано за допомогою бібліотеки Telethon, що здійснює запити до офіційного API Telegram. Алгоритм збору даних з ресурсу Telethon реалізований на основі Telegram API, що виконує запити до API даного ресурсу. Далі дані записуються у БД.
    3. **Алгоритм обробки великої кількості даних** реалізований з використанням бібліотеки numpy, що підготовлює дані для створення графіків на їх основі.

# **Аналіз функціонування засобів масштабування**

Було створено декілька шардів MongoDB, куди були записані дані. Через те, що кількість записів у БД є відносно малою, а характеристики машини, на якій була запущена система є відносно хорошими(наявність диску з великою швидкістю доступу до даних, великою кількістю ОЗП тощо) наявність шардів вплинула на швидкодію непомітно або й зовсім не вплинула.

# **Опис результатів аналізу предметної галузі**

У додатку А представлено 5 графіків демонстрації роботи моніторингової системи Telegram-каналів.

На рис. 1 представлено діаграму структури бази даних, в якій зберігаються дані про пости на каналі.

На рис. 2 представлено діаграму найвживаніших слів у постах первного телеграм-каналу(для прикладу було взято канал @kpi\_777 та оброблено 304 останніх постів). З діаграми можемо бачити, що найвживанішими є саме слова навчальної лексики.

На рис. 3 подано графік кількості переглядів постів каналу за 27 місяців.

Для прикладу взято інший канал, а саме @it\_kpi. Як можемо бачити, максимум відвідувань припав на вересень 2019 року.

На рис. 4 представлено діаграму середньої кількості постів за кожен день тижня(проаналізовано всього 101 тиждень). Для прикладу взято канал @kpi\_live, @kpi\_777, @it\_kpi. Як видно з діаграми, адміністратори каналу постять ммайже з однаковою інтенсивністю з понеділка по пятницю, а у вихідні спостерігаємо меншу активність, що й закономірно.

На рис.5 представлено приклад вигляду графічного інтерфейсу.

# **Висновки**

Було створено програмне забезпечення, яке забезпечило виконання наведених далі пунктів:

1. Попередня обробка даних

* Засоби генерації даних.
* Розроблення утиліти для збору інформації про телеграм-канали за допомогою Telethon API.
* Встановлення зв’язку між даними, отриманими із двох джерел і розміщення їх у колекціях бд.
* Засоби фільтрації і валідації даних. Розроблення додаткового функціоналу у вищезазначеній утиліті задля корегування отриманих даних та переходу до їх подальшої обробки та структуризації.

1. Основний модуль: фільтрація та аналіз постів каналу Телеграм, а саме кількості їх переглядів за певний проміжок часу, середню кількість відвідувань каналу за певний проміжок часу, знаходження найвживаніших слів на каналі.

Візуалізація отриманих результатів шляхом побудови графіків.

1. Забезпечення масштабування та реплікації даних: шардинг та створення репліка сетів у Mongo DB

В ході виконання даного курсового проекту було досягнуто поставленої мети: було набуто практичних навичок розробки сучасного програмного забезпечення, що взаємодіє з нереляційними базами даних, а також були здобуті навички оформлення відповідного текстового, програмного та ілюстративного матеріалу у формі проектної документації. У результаті виконання курсового проекту також були здобуті навички розробляти програмне забезпечення для постреляційних баз даних, володіння основами використання СКБД, а також інструментальними засобами підтримки розробки додатків для подібних баз даних.

# **Список використаних джерел**

1. Python Numpy Tutorial

[online] Available at: <http://cs231n.github.io/python-numpy-tutorial/>.

1. Python Graph Gallery

[online] Available at: <https://python-graph-gallery.com/>.

1. Python data visualizations with Matplotlib

[online] Available at: <https://towardsdatascience.com/a-step-by-step-guide-for-creating-advanced-python-data-visualizations-with-seaborn-matplotlib-1579d6a1a7d0>.

1. Tkinter tutorial

[online] Available at: <https://tkdocs.com/tutorial/>.

1. Npyscreen tutorial

[online] Available at: <https://npyscreen.readthedocs.io/introduction.html>.

1. Telethon documentation

[online] Available at: <https://telethon.readthedocs.io/>.

1. Pymongo documentation

[online] Available at:<https://api.mongodb.com/python/current/>.

1. Matplotlib documentation

[online] Available at:<https://matplotlib.org/>.

1. Extracting Telegram data with Telethon

[online] Available at: <https://medium.com/@dc.aichara/telegram-channel-data-extraction-users-information-chats-and-specific-messages-and-data-21bb54710fd3>.

# **Додатки**

*Додаток А*

  
Рис. 1 – ER-діаграма бази даних

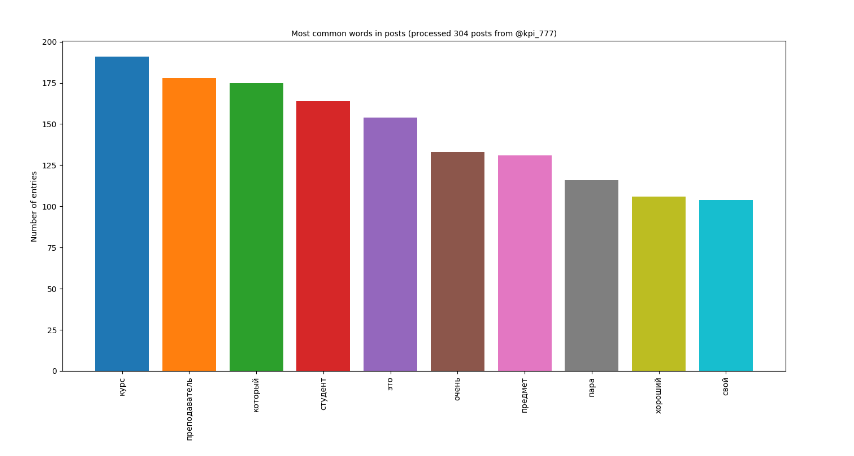


Рис. 2 – Діаграма найвживаніших слів у постах первного телеграм-каналу



Рис. 3 – Графік кількості переглядів постів каналу за 27 місяців

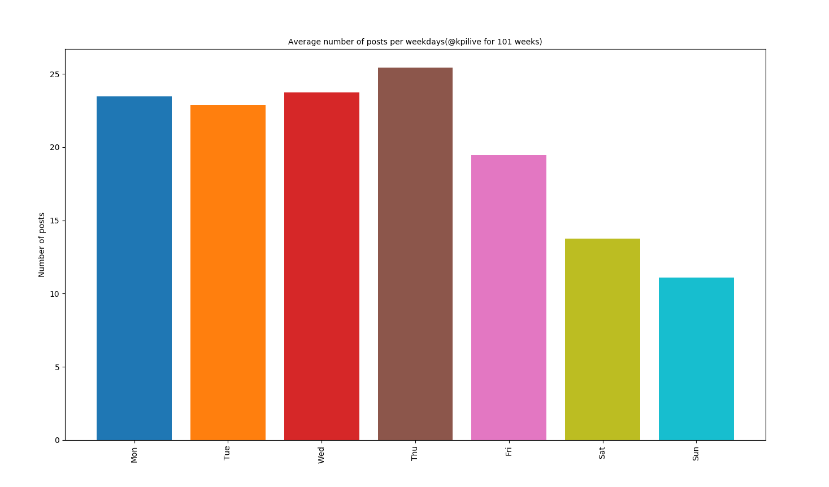


Рис. 4 - Діаграма середньої кількості постів за кожен день тижня(проаналізовано 101 тиждень)

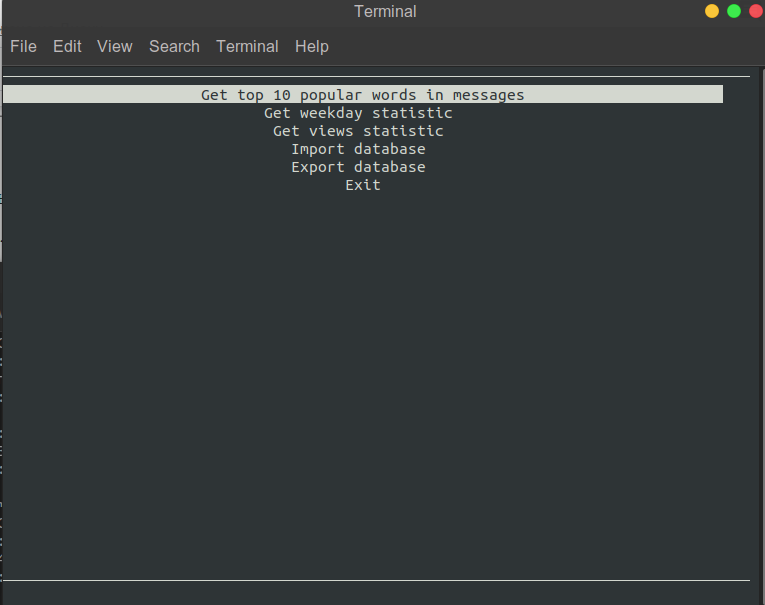


Рис. 5 – Приклад графічного інтерфейсу програми