

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота**

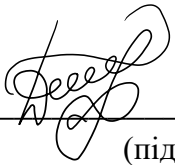
**на здобуття освітнього ступеня «бакалавр»**

За спеціальністю: 121 Інженерія програмного забезпечення

на тему:

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГУВАННЯ  
НОВИН З ПРОВІДНИХ ІНФОРМАЦІЙНИХ ПОРТАЛІВ З  
ІНТЕРФЕЙСОМ ТЕЛЕГРАМ-БОТА**

Виконала студентка 4-го курсу  
Дарина БОНДАРЕЦЬ

  
\_\_\_\_\_  
(підпис)

Науковий керівник:  
к.т.н., доцент кафедри інтелектуальних програмних систем,  
Євген ДЕМКІВСЬКИЙ

\_\_\_\_\_  
(підпис)

Засвідчую, що в цій роботі немає запозичень  
з праць інших авторів без відповідних  
посилань.

Студент

  
\_\_\_\_\_  
(підпис)

Роботу розглянуто й допущено до захисту  
на засіданні кафедри інтелектуальних  
програмних систем  
«25» травня 2022 р.,  
протокол № 10

Завідувач кафедри  
Олександр ПРОВОТАР

\_\_\_\_\_  
(підпис)

## РЕФЕРАТ

Обсяг роботи 51 сторінка, 20 ілюстрацій, 7 джерел посилань.

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГУВАННЯ НОВИН З ПРОВІДНИХ ІНФОРМАЦІЙНИХ ПОРТАЛІВ З ІНТЕРФЕЙСОМ ТЕЛЕГРАМ-БОТА.**

**Об'єктом роботи** є задача оптимізації процесу перегляду розосередженої інформації та вирішення проблеми надлишкової кількості публікацій в інформаційних порталах. **Предметом роботи** є програмне забезпечення для агрегування новин з провідних інформаційних порталів з інтерфейсом телеграм-бота.

**Метою роботи** є розробка програмного забезпечення для агрегування новин з провідних інформаційних порталів з інтерфейсом телеграм-бота, яке було б корисним для користувачів та простим у використанні.

**Методи розроблення:** програмування телеграм-боту, розробка продукту за шаблоном проектування архітектури програми. Інструменти розроблення: ліцензійне, вільно поширюване інтегроване середовище розробки WebStorm 2022.1, мова програмування JavaScript на платформі Node.js. В якості бази даних було обрано MongoDB. Для стабільної роботи чат-бот було розміщено на віддаленому сервісі EvenNode.

**Результати роботи:** було досліджено існуючі засоби для агрегування новин та їх принцип роботи. Під час виконання роботи було визначено основні принципи створення телеграм-ботів. Також було розроблено технічне завдання, спроектовано та реалізовано програмне забезпечення для агрегування новин з інтерфейсом телеграм-бота, що полегшує процес отримання інформації з різних джерел, зокрема було реалізовано фільтрування публікацій за категоріями.

**Програмний продукт** «News of UA» може використовуватись як у сфері інформаційних технологій так і у розважальних або пізнавальних цілях. Отримана архітектура є досить гнучкою і надає можливість розширення

функціоналу в майбутньому та стати прикладом для реалізації аналогічних сервісів для агрегації новин.

## ЗМІСТ

|   |    |
|---|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ   | 6  |
| ВСТУП   | 7  |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ   | 9  |
| 1.1. Шляхи ефективного використання інформаційних порталів  | 9  |
| 1.2. Агрегатор новин  | 12 |
| 1.3. Чат-боти в месенджерах   | 18 |
| РОЗДІЛ 2. ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГУВАННЯ НОВИН  | 21 |
| 2.1. Огляд існуючих рішень для отримання новин з інформаційних каналів за допомогою телеграм-ботів          | 21 |
| 2.1.1. Огляд телеграм-боту AximoBot   | 22 |
| 2.1.2. Огляд телеграм-боту Feed2Telegram  | 23 |
| 2.1.3. Огляд телеграм-боту RSS2tg Bot   | 25 |
| 2.2. Формування вимог до програми   | 26 |
| 2.2.1. Вимоги до функціоналу програми   | 26 |
| 2.2.2. Вимоги щодо інтерфейсу користувача   | 26 |
| РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГАЦІЇ НОВИН З ІНТЕРФЕЙСОМ ТЕЛЕГРАМ-БОТА | 29 |
| 3.1. Діаграма варіантів використання чат-боту   | 29 |
| 3.2. Обґрунтування вибору засобів для реалізації  | 30 |
| 3.2.1. Платформа Node.js на мові JavaScript   | 30 |
| 3.2.2. Прикладний програмний інтерфейс Telegram Bot API   | 32 |
| 3.2.3. База даних – MongoDB   | 33 |
| 3.2.4. Хостинг – EvenNode   | 35 |
| 3.3. Структура бази даних   | 36 |

|  |    |
|--|----|
|  | 5  |
| 3.4. Реєстрація телеграм-бота                              | 38 |
| 3.5. Розробка програмного забезпечення для агрегації новин | 39 |
| РОЗДІЛ 4. ТЕСТУВАННЯ Й ОГЛЯД РОЗРОБЛЕНОГО БОТА             | 42 |
| 4.1. Функції та команди бота                               | 42 |
| 4.2. Тестування системи                                    | 47 |
| 4.3. Розширення функціоналу                                | 49 |
| ВИСНОВКИ   | 50 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ                                 | 51 |

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ**

API – Application Programming Interface, прикладний програмний інтерфейс;

HTML – Hypertext Markup Language, мова розмітки гіпертексту;

HTTP – HyperText Transfer Protocol, протокол передачі гіпертекстових документів;

JS – Java Script, мова програмування;

RSS – Really Simple Syndication, насправді проста синдикація;

URL – Uniform Resource Located, єдиний вказівник на ресурс;

XML – Xtensible Markup Language, розширювана мова розмітки.

## ВСТУП

**Оцінка сучасного стану об'єкта розробки.** На сьогоднішній день соціальні мережі перебувають у дуже активної стадії розвитку. Щодня загальна кількість користувачів і популярних соціальних мереж швидко зростає. Як наслідок, виникає проблема, яка пов'язана із споживанням користувачами зайвої інформації або неякісного контенту, оскільки час, який людина виділяє на використання соціальних мереж, є недостатнім аби повністю контролювати потік інформації.

Крім того, розкид інформації з різних платформ призводить до дискомфорту і втрати часу при перемиканні між платформами. Останніми роками траплялися випадки, коли великі соціальні мережі у рамках співробітництва реалізовували агрегацію контенту за допомогою інтегрування у свій продукт елементи інших соціальних мереж. Таким чином, проблема поширення контенту з різних інформаційних порталів також є дуже актуальною.

У даній роботі розглядається розробка програмного забезпечення для агрегування новин з провідних інформаційних порталів з інтерфейсом телеграм-бота, що являється дуже популярним підходом розробки у наш час.

**Актуальність роботи та підстави для її виконання** полягає у відсутності безкоштовних робочих аналогів телеграм-ботів для агрегування новин, де користувачі могли б відфільтрувати контент за допомогою категорій.

**Мета й завдання роботи.** Метою роботи є розробка програмного забезпечення для агрегування новин з провідних інформаційних порталів з інтерфейсом телеграм-бота, яке було б корисним для користувачів та простим у використанні. Для досягнення цієї мети поставлено такі завдання:

- дослідити існуючі засоби для агрегування новин та їх основний принцип роботи;
- розробити технічне завдання до продукту;

- спроектувати та розробити телеграм-бота для агрегування новин з провідних інформаційних порталів.

**Об'єкт, методи й засоби розроблення.** Об'єктом розроблення програмного засобу для агрегації новин з провідних порталів є задача оптимізації процесу перегляду розосередженої інформації та вирішення проблеми надлишкової кількості публікацій в інформаційних порталах.

Розробці програмного засобу передувало створення моделі архітектури проєкту, що описує об'єкт. Для цього було використано уніфіковану мову моделювання.. Основу для цього склав аналіз деяких аналогічних продуктів, що дозволило виділити основний функціонал програми та вимоги щодо інтерфейсу користувача.

Даний програмний продукт розроблявся з використанням еволюційної моделі, заснованої на принципі створення початкової версії продукту, яка тестується кінцевими користувачами для оцінки, після чого програмний продукт підлягає додатковій розробці, враховуючи побажання та думки споживача. Аналогічно розробляються і проміжні версії програмного продукту, які оцінюються та доробляються до того рівня, поки не з'явиться програмний продукт, який повністю відповідає всім вимогам користувача. Паралельно, з постійним обміном інформації між ними, ведуться процеси специфікації, розробки та атестації програмного продукту.

Програмний продукт створено у вигляді телеграм-бота. Для його реалізації було обрано мову програмування JavaScript на платформі Node.js з використанням фреймворків grammY (для комфортної роботи з Telegram Bot API) Node Fetch та Fast Xml Parser (для швидкого парсингу RSS-стрічок). В якості бази даних було обрано MongoDB. Для стабільної роботи чат-бот було розміщено на віддаленому сервісі EvenNode.

**Можливі сфери застосування.** Розроблений телеграм-бот може як у сфері інформаційних технологій так і у розважальних або пізнавальних цілях.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Шляхи ефективного використання інформаційних порталів

В останні роки інформаційний простір в Україні та світі перенасичений подіями, які висвітлюються у стрічках новин великої кількості сайтів та соціальних мереж. Український медіаландшафт перенасичений усіма типами ЗМІ — від традиційних (газети, радіо та телебачення) до сучасних (новинні веб-сайти, програми та платформи соціальних мереж). Через це люди не можуть визначитись, який інформаційний ресурс надає найбільш швидкий та повний огляд останніх подій. Іноді навіть трапляється, що різні ресурси надають суперечливу інформацію про одні й ті ж самі події. Тому переважна більшість людей надає перевагу певним інформаційним порталам, перевіреним часом або тим, що викликають довіру.

Станом на 2021 рік споживання новин за допомогою традиційних медіа, таких як телебачення, радіо та друкованих видань — систематично зменшується протягом останніх 5 років. Це ствердження засноване на статистичних дослідженнях міжнародних та українських соціологічних компаній. Як приклад, можемо привести наступні соціологічні дослідження, метою яких було вивчення звичок українців, що стосуються споживання медіа, а також їхню довіру до медіа.

Компанією InMind на замовлення міжнародної неприбуткової організації Internews було проведено дослідження на тему «Ставлення населення до ЗМІ та споживання різних типів медіа у 2020 році» (рис 1.1), а також дослідження USAID-Internews про «Ставлення населення до медіа та споживання різних типів медіа 2021», яке було презентоване 17 листопада 2021 року (рис 1.2).

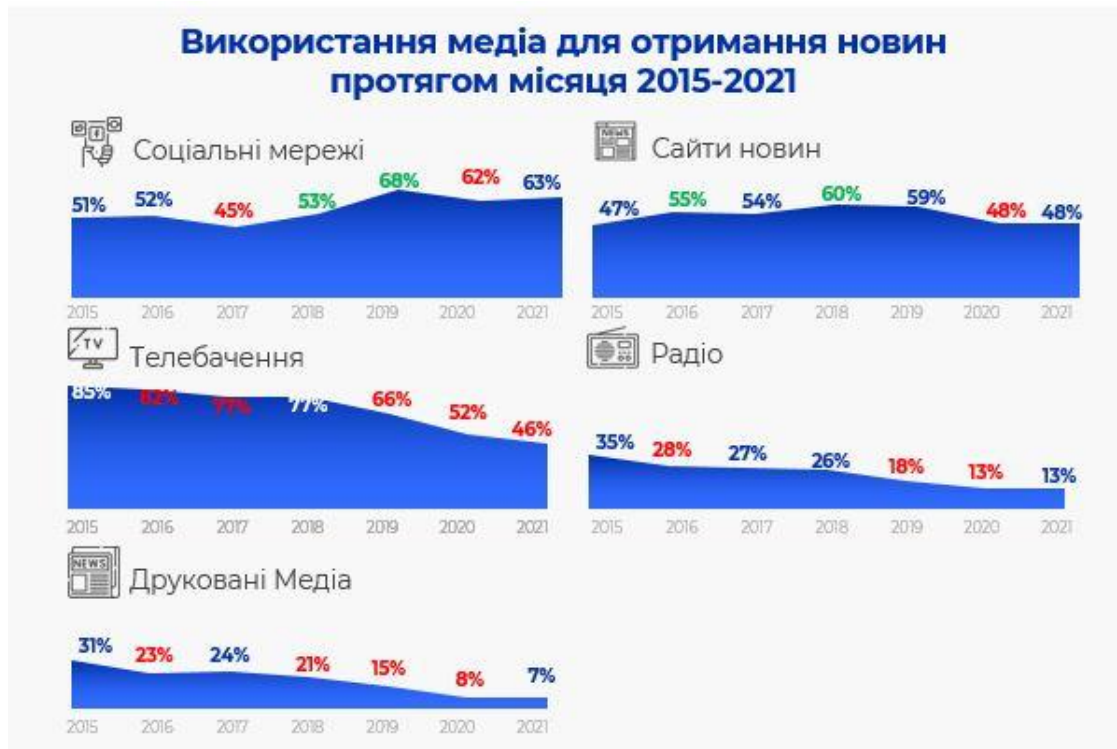


Рисунок 1.1 – Використання медіа для отримання новин протягом місяця 2015-2021 (USAID) [1]

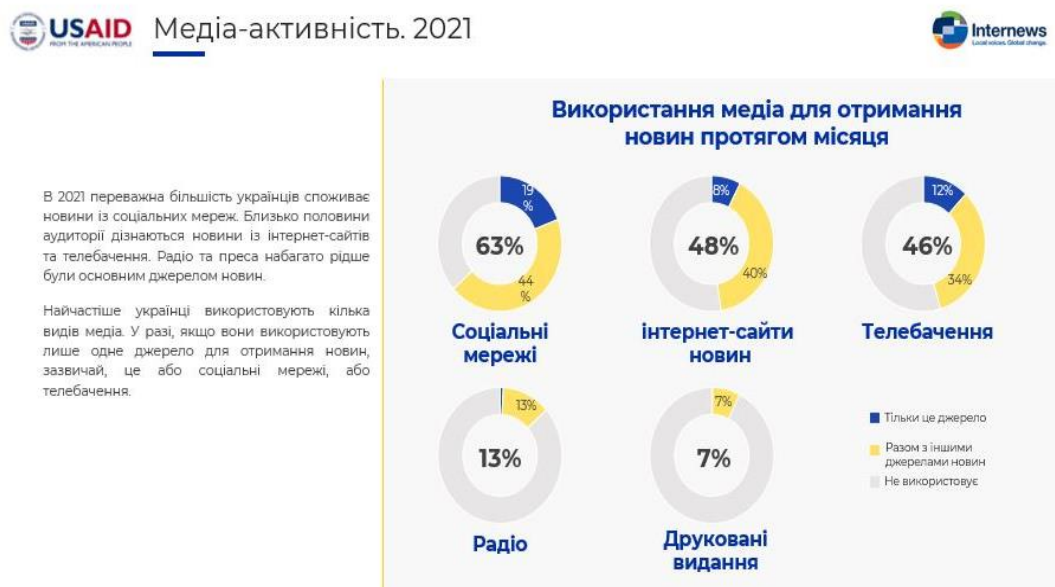


Рисунок 1.2 – Використання медіа для отримання новин протягом місяця (USAID) [1]

Оглянувши результати, можна помітити, що переважна більшість українців отримує інформацію та новини із соціальних мереж, але найчастіше

українці використовують кілька видів медіа. Так, новини дізнаються з соцмереж 63% людей, лише 19% – виключно з них. З телебачення – 48% респондентів і лише 8% – виключно з нього. Інші джерела – радіо та друковані видання – разом використовують лише 20% українців. [1]

Найбільш популярними загальнонаціональними онлайн-медіа та телебачення станом на 2021 рік став канал «1+1» (близько 66% опитуваних респондентів назвали канал серед трьох джерел, де дивляться новини найчастіше) та сайт каналу (24%). З повним списком рейтингів медіа, які використовували у 2021 році можна ознайомитися нижче (рис 1.3).



Рисунок 1.3 – Медіа, які використовували у 2021 році (USAID) [1]

Щороку кількість користувачів Інтернету зростає. Згідно з дослідженнями у 2021 році, 82% населення України користується інтернетом щоденно (серед них 97% молодих людей у віці 18-35 років). [1]

Ознайомившись із результатами дослідження, можна помітити не тільки збільшення частки споживачів, які дізнаються новини з Інтернету, але й зменшення частки тих, хто користується двома джерелами одночасно.

Оскільки щодня додається велика кількість нового контенту, можуть виникати труднощі з тим, щоб не відставати від того, що відбувається в Інтернеті на даний момент. Користувачі намагаються раціоналізувати даний процес кількома різними способами, зокрема щодня відвідувати певні веб-сайти, виконувати пошук у Google або покладатися на соціальні мережі, щоб отримати свіжі новини.

Враховуючи те, що інформаційних джерел може бути декілька, виникає постійна потреба у відвідуванні певних сайтів та соціальних мереж для отримання свіжої та достовірної інформації. Дана процедура займає досить велику кількість часу та зусиль у пошуку останніх новин. Саме тому використання агрегаторів новин, які б збирали усі новини в одному місці, набуває особливої актуальності.

## **1.2. Агрегатор новин**

Припустимо, що ви маєте бажання регулярно спостерігати за новинами з десятків різних джерел. Тоді, аби їх зібрати та ознайомитися з вмістом кожного посту, вам доводиться систематично відвідувати велику кількість новинних порталів. Крім того, може статися така ситуація, коли ви знову відвідуєте сторінку, проте ніяких оновлень не було додано. Тобто той час, який було витрачено на повторний пошук був марним.

Розглянемо інший спосіб, де оновлення з блогів і новинних сайтів самі приходять до вас. Мова йде про використання одного єдиного додатка або платформи для читання усіх оновлень в інтернет мережі. За допомогою RSS-каналу (RSS часто називають стрічкою новин або RSS-канал) вони можуть швидше перевірити ваш сайт за допомогою агрегатора RSS (сайт або програма, яка збирає та сортує RSS-канали).

**Агрегатор новин** (іноді RSS-агрегатор або RSS-reader) — клієнтське програмне забезпечення або веб-застосунок, який об'єднує синдигований веб-контент (такий як блоги, подкасти, онлайн-газети або відеоблоги) із джерел, що експортують інформацію у форматах Atom або RSS. Щодня агрегатор шукає на зареєстрованих веб-сайтах документи RSS, перевіряє посилання та відображає інформацію про канал, щоб клієнти могли отримувати статті, які їх цікавлять.

**Atom (Atom Syndication Format)** — це формат заснований на XML, яка використовується для агрегації інформації. Може використовуватися для будь-яких періодичних видань в Інтернеті. Канал містить записи, які можуть бути заголовками, повнотекстовими статтями, уривками, резюме або посиланнями на вміст веб-сайту разом із різними метаданими. Формат Atom був розроблений як альтернатива RSS. Особливістю специфікації є протокол, що автоматизує ведення блогу та працює поверх HTTP.

**RSS (Really Simple Syndication)** — це родина XML-форматів, що дозволяє користувачам і додаткам отримувати доступ до оновлень веб-сайтів у стандартизованому, читабельному комп'ютером форматі. Підписка на канали RSS може дозволити користувачеві відстежувати багато різних веб-сайтів в одному агрегаторі новин, який постійно відстежує сайти на наявність нового вмісту, позбавляючи користувача від необхідності перевіряти їх вручну. [2]

RSS використовується для обміну вмістом між веб-сайтами, а також визначає простий спосіб ділитися та переглядати заголовки та вміст нових постів. Канал був розроблений для відображення вибраних даних, що дозволяє персоналізувати перегляди для різних сайтів.

RSS-файли можна автоматично оновлювати. Оскільки дані RSS невеликі і швидко завантажуються, їх можна легко використовувати з такими службами, як мобільні телефони або ПК. Вебкільця з подібною інформацією

можуть легко обмінюватися даними на своїх веб-сайтах, щоб зробити їх кращими та кориснішими.

За допомогою RSS ви можете вибрати для перегляду потрібні вам новини й публікації, які вас цікавлять або мають відношення до вашої роботи. Також він надає змогу відокремити потрібну інформацію від небажаної (спаму). За допомогою RSS ви можете створити власний канал новин і публікувати його в Інтернеті.

На цей момент немає офіційного стандарту для RSS. Близько 50% всіх RSS-каналів використовують RSS 0.91, приблизно 25% використовують RSS 1.0, а останні 25% розподіляються між версіями RSS 0.9x і RSS 2.0.

У RSS-документах використовується простий синтаксис із самоописом. Розглянемо приклад RSS-стрічки (рис 1.4).

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">

<channel>
  <title>My Home Page</title>
  <link>https://www.mypage.com</link>
  <description>My blog is the best</description>
  <item>
    <title>RSS Tutorial</title>
    <link>https://www.mypage.com/rss</link>
    <description>New RSS tutorial</description>
  </item>
  <item>
    <title>XML Tutorial</title>
    <link>https://www.mypage.com/xml</link>
    <description>New XML tutorial</description>
  </item>
</channel>

</rss>
```

Рисунок 1.4 – Простий приклад RSS-стрічки

Перший рядок в документі — декларація XML — визначає версію XML і кодування символів, що використовуються в документі. У цьому випадку документ відповідає специфікації XML 1.0 і використовує набір символів

UTF-8. Наступний рядок — це декларація RSS, яка визначає, що це документ RSS (у даному випадку RSS версія 2.0).

Документ RSS-стрічки складається із двох частин: каналу та статей.

Елемент `<channel>` використовується для опису каналу RSS та має три обов'язкових дочірніх елементи:

- `<title>` — визначає назву каналу;
- `<link>` — визначає гіперпосилання на веб-сайт, з яким встановлений канал зв'язку;
- `<description>` — текст, що описує канал.

Крім того, є кілька необов'язкових дочірніх елементів `<channel>` :

- `<category>` — визначає одну або кілька категорій для каналу;
- `<cloud>` — реєструє процеси, щоб негайно отримувати сповіщення про оновлення каналу;
- `<copyright>` — повідомляє про матеріали, захищені авторським правом;
- `<docs>` — вказує URL-адресу документації у форматі, який використовується в каналі;
- `<generator>` — визначає програму, яка використовується для створення каналу;
- `<image>` — дозволяє відображати зображення, коли агрегатори представляють стрічку;
- `<language>` — визначає мову, якою написаний канал;
- `<lastBuildDate>` — визначає дату останньої зміни вмісту каналу;
- `<managingEditor>` — визначає адресу електронної пошти редактору вмісту стрічки;
- `<pubDate>` — визначає дату останньої публікації вмісту каналу;
- `<rating>` — PICS рейтинг стрічки;

- `<skipDays>` — визначає дні, коли агрегатори повинні пропускати оновлення каналу;
- `<skipHours>` — визначає години, коли агрегатори повинні пропускати оновлення каналу;
- `<textInput>` — визначає поле введення тексту, яке має відобразитися разом зі стрічкою;
- `<ttl>` — визначає кількість хвилин, протягом яких канал може залишатися кешованим, перш ніж оновити його з джерела;
- `<webMaster>` — визначає адресу електронної пошти для веб-майстра каналу.

Кожен елемент `<channel>` може мати один або кілька елементів `<item>`.

Кожен елемент `<item>` визначає статтю або публікацію в RSS-каналі.

Елемент `<item>` має три обов'язкових дочірніх елементи:

- `<title>` — визначає назву елемента;
- `<link>` — визначає гіперпосилання на елемент;
- `<description>` — описує елемент.

Крім того, є кілька необов'язкових дочірніх елементів `<item>` :

- `<author>` — вказує адресу електронної пошти автора елемента;
- `<category>` — визначає одну або кілька категорій, до яких належить елемент;
- `<comments>` — дозволяє елементу посилатися на коментарі щодо цього елемента;
- `<enclosure>` — дозволяє додати медіафайл до елемента;
- `<guid>` — визначає унікальний ідентифікатор для елемента;
- `<pubDate>` — визначає дату останньої публікації для елемента;
- `<source>` — вказує стороннє джерело для елемента.

Чим більше даних надається, тим якість відображення RSS-стрічки покращується. Наприклад дочірній елемент `<category>` використовується для

визначення категорії для вашого каналу, що дає змогу агрегаторам RSS групувати сайти за категоріями.

Нарешті, два останні рядки закривають елементи `<channel>` і `<rss>`.

Оскільки RSS є XML, то потрібно пам'ятати, що:

- усі елементи повинні мати закриваючий тег;
- елементи чутливі до регістру;
- елементи повинні бути правильно вкладені;
- значення атрибутів завжди мають бути в лапках.

Агрегатори контенту поділяють на наступні типи:

1. Програми агрегатори медіа контенту. Вони дозволяють сформувати контент для окремого користувача. Платформи такого типу більш популярні в користувачів, тому що дають можливість використання на мобільних пристроях. Основним недоліком є обмеження можливостей в налаштуваннях інтерфейсу та фільтрації публікацій з соцмереж, а також наявність великої кількості реклами.

2. Платформи веб агрегації. Вони створені для ведення блогів та сайтів, що дозволяє керувати контентом, який потрапляє на веб сторінку. Цей агрегатор не є персональним, тому що він автоматизує відбір, але не має особистих налаштувань споживача контенту і не підходить для повсякденного користування середньостатистичним користувачем.

3. Веб-сайти агрегатори. Вони формують персоналізований потік контенту з різних соцмереж і орієнтовані на використання окремими користувачами, мають широкі можливості в плані налаштувань. Використання браузера для перегляду контенту на веб-сторінці є основним недоліком даного виду агрегаторів, тому що є незручним для частини потенціальних користувачів, особливо при користуванні з мобільного пристрою.

4. Системи, інтегровані в соціальні мережі чи месенджери. В них контент переадресовується з однієї соцмережі в іншу, або збирається з кількох джерел та відображається в одному місці. Боти – алгоритми складають

основну частину таких систем. Основним недоліком таких систем являється обмежений доступ до публікації інформації, що використовується інформаційними платформами для запобігання різного роду зловмисних дій зі сторони ботів.

### 1.3. Чат-боти в месенджерах

Не зважаючи на велику кількість інформаційних сайтів, сучасне населення надає перевагу перегляду новин в соціальних мережах, про що також свідчать соціологічні опитування.

Наприклад, дослідження «Ставлення населення до ЗМІ та споживання різних типів медіа у 2020 році», що було виконано соціологічною компанією InMind на замовлення міжнародної неприбуткової організації Internews опублікувало наступні результати досліджень:



Рисунок 1.5 – Дослідження популярності соціальних мереж 2020-2021 роках (USAID) [1]

«Найпопулярнішою платформою для отримання новин залишається Facebook — принаймні раз на місяць 66% і 65% респондентів користуються Viber або Facebook відповідно, а 43% українців отримують новини з Facebook.

Але в 2021 році споживачі менше, ніж у 2020 році, будуть використовувати Facebook, YouTube і Viber для спілкування та перегляду новин.

До недоліків використання Facebook, на думку респондентів, можна віднести труднощі з управлінням, велику кількість реклами та фейкових новин, затримки з публікацією новин. Інстаграм частіше використовують молодь (частіше 18-30 років, рідше 31-40 років) як розважальний контент. Але останнім часом Instagram став джерелом новин через швидку доставку новин. На думку респондентів, YouTube занадто схожий на телебачення і не підходить для пошуку останніх новин. Telegram продовжує залишатися популярним каналом новин майже для всіх вікових груп (крім 50+). Збільшується інтерес до національних та регіональних Telegram-каналів. [1]

Telegram фокусується на швидкості та безпеці. Він надає дуже зручний функціонал для роботи з ботами, оскільки він має зручний API для розробки ботів та стрімко набирає популярність серед користувачів та розробників.

У Telegram зареєстровано понад 180 мільйонів користувачів, переважна більшість яких проживає в Західній Європі. Розробники Telegram повідомляють, що цей месенджер є найбільш захищеним серед подібних продуктів на ринку.

Основні переваги, які утворився за час існування месенджера: конфіденційність, оптимізація, швидкість, прозорість, доступність, відсутність великих обмежень щодо розміру повідомлення та файлів, що надсилаються.

Враховуючи написане вище, виникла ідея створення чат-боту для агрегування новин саме для Telegram.

Останнім часом, на перший план виходять чат-боти, яким пророкують велике майбутнє. Із зростанням популярності месенджерів, зріс і попит на чат-боти. Переважна більшість працює на наступних платформах месенджерів: Facebook Messenger, Telegram, Viber, Skype, Slack.

Чат-бот – це спеціалізована програма на платформі обміну повідомленнями, що дозволяє взаємодію користувачів зі сторонніми сервісами за допомогою звичного інтерфейсу у вигляді чату. Обмін повідомленнями здійснюється в системі "людина" – "комп'ютер". Чат-боти автоматизують роботу, виконуючи попередньо заданий алгоритмом, а також виконують запити користувача, імітуючи діалог.

Боти можуть бути запущені як окремі програми або вбудовані у функціональність пошукової системи. Чат-боти використовують в різноманітних галузях, таких як сервісні-центри, ігрова або комерційна індустрія тощо. Використання чат-ботів для таких цілей зазвичай обмежене вузькою спеціалізацією, і їх не можна використовувати для широкого кола людського спілкування.

## РОЗДІЛ 2. ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГУВАННЯ НОВИН

### 2.1. Огляд існуючих рішень для отримання новин з інформаційних каналів за допомогою телеграм-ботів

При огляді існуючих телеграм-ботів, що вирішують проблему отримання новин з інформаційних каналів було обрано 3 основних представника, а саме: AximoBot (безкоштовна версія), Feed2Telegram (безкоштовна версія) та RSS2tg Bot.

Основні критерії по яким буде проводитися оцінка аналогів:

- швидкість оновлення стрічки;
- можливість додати власну RSS-стрічку;
- підтримка інших форматів новин;
- кількість функціоналу;
- наявність готових рекомендацій;
- фільтрування новин за категоріями;
- коректне форматування публікацій;
- наявність короткого опису новини;
- інтерфейс користувача;
- доступність.

Під час аналізу аналогів я підключила усі 3 бота для перевірки справності їх роботи. Боти були підключені до однакових сайтів для чистоти експерименту. Проблем з оновленням не спостерігалось ні в одного з протестованих телеграм-ботів.

Для підписки на певні джерела необхідно надати боту посилання на нього, а для її відписки – треба обрати джерело зі списку або ввести власноруч.

Розглянемо детальніше кожен з представників.

### 2.1.1. Огляд телеграм-боту AximoBot

AximoBot – це популярний агрегатор новин з інтерфейсом телеграм-бота. Станом на травень 2022 року має приблизно 44,380 активних користувачів. У бота наявна можливість додати власну RSS стрічку, а також присутня підтримка інших форматів новин, а саме з таких соціальних мереж як Instagram, TikTok, Youtube, Twitter, Facebook, Twitch, Reddit, Livejournal, та ще декілька російських соцмереж.

Приклад відображення публікацій наведено на рис. 2.1. Присутній заголовок новини, оформлений у вигляді посилання на джерело, наступним рядком іде назва RSS-стрічки. В останньому абзаці короткий опис події. Також відображено зображення, що було прикріплене до публікації.

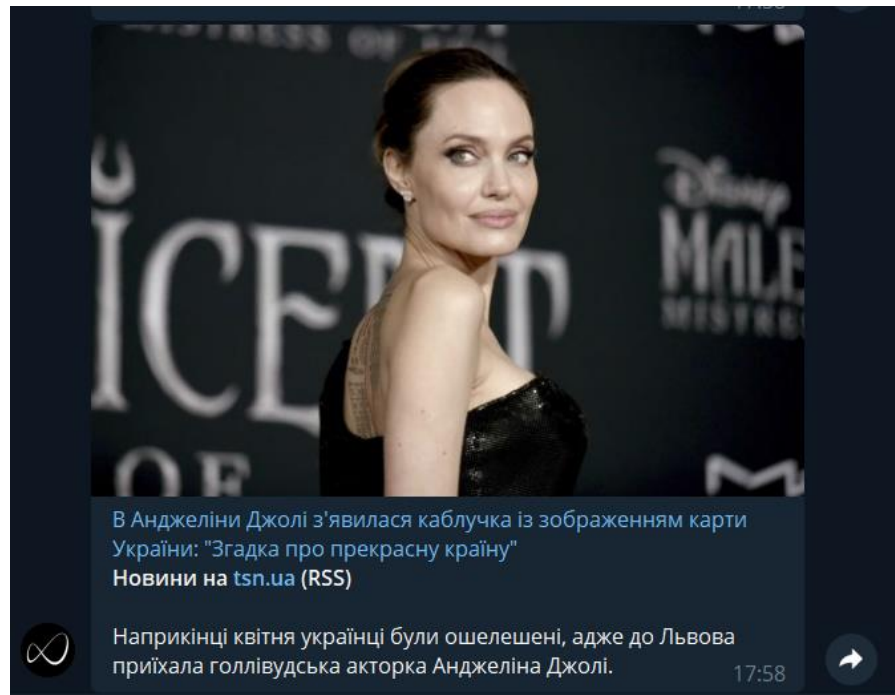


Рисунок 2.1 – Приклад оформлення публікацій у бота @AximoBot

Серед переваг можна назвати:

- відслідковування публікацій не лише за допомогою, але й великої кількості соціальних мереж;
- правильне форматування новин, без дивних символів і зайвих тегів;
- наявність короткого опису новини;

- кількість функціоналу достатня, проте для безкоштовної версії він був сильно урізаний;
- наявність готових рекомендацій з різних джерел та соціальних мереж;
- зручний інтерфейс користувача за допомогою кнопок та команд;
- є можливість переглянути статистику.

Основними недоліками є:

- для початку роботи з ботом змушують підписатися на зайвий канал;
- швидкість оновлення стрічки раз в 3 години, що є дуже не зручним способом. З таким довгим інтервалом збирається велика кількість невідправлених новини, які не тільки приходять дуже пізно й втрачають свою актуальність, але й сильно заспамлює чат, бо усі публікації приходять одночасно;
- відсутність функції фільтрації та відбору контенту за категоріями;
- немає функції призупинити розсилку;
- у безкоштовній версії кількість підписок обмежена 15 джерелами.

### **2.1.2. Огляд телеграм-боту Feed2Telegram**

Хоча у бота @Feed2Telegram\_bot немає великого набору команд, проте він є відносно простим у користуванні і не потребує довгих налаштувань. У бота також наявна можливість додати власну RSS стрічку й присутня підтримка соцмереж. Проте, порівняно з попереднім представником, список соціальних мереж, що підтримує бот набагато менший (лише Instagram, Youtube, Twitter та Facebook).

Якщо розглянути приклад відображення публікацій (рис 2.2), то можна помітити основну проблему, що полягає у неправильному неправильному форматуванні тексту. Наприклад, замість апострофу та латок відображається HTML-код. У новини присутній заголовок, посилання на джерело ніяк не оформлене, назва RSS-стрічки відображається у першому рядку. Короткий

опис події відсутній. Зображення відображається лише як частина посилання, а не було прикріплене до публікації.

Кнопки, що знаходяться під постом, не виконують заданий функціонал, тому що використовується безкоштовна версія агрегатора.

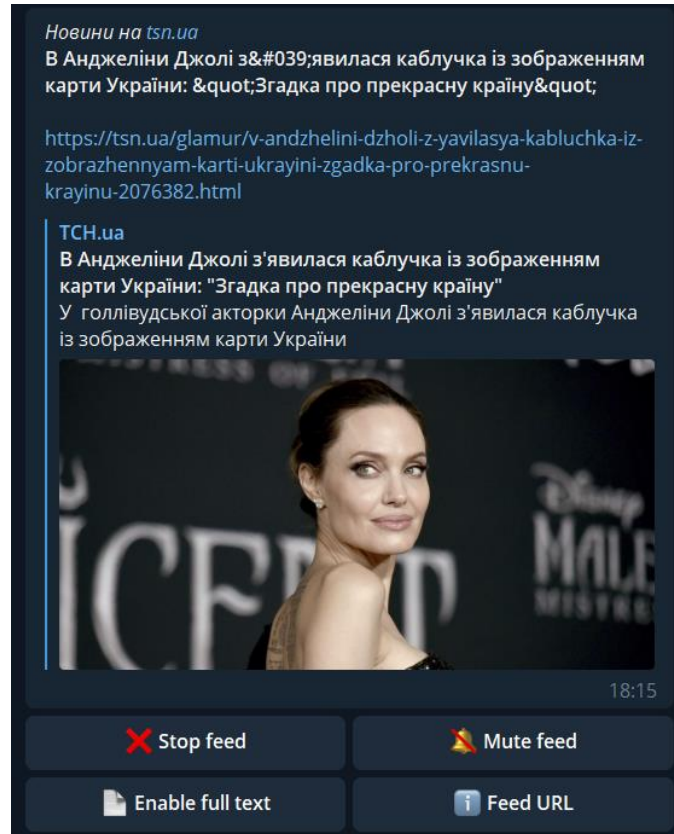


Рисунок 2.2 – Приклад оформлення публікацій у бота @Feed2Telegram\_bot

Переваги бота: дозволяє відслідковувати контент профілів в таких соцмережах як Instagram, YouTube, Facebook, Twitter, а також потоки RSS та публічні канали в Telegram.

Основні недоліки, що вдалося виділити: неправильне форматування тексту; відсутні картинки; довго оновлюється (раз в годину); доступні лише 3 безкоштовні підписки; дуже багато функціоналу обрізано у безкоштовній версії; лише англійська мова інтерфейсу, рекомендації не працюють, бот не має функції фільтрації новин за категоріями.

### 2.1.3. Огляд телеграм-боту RSS2tg Bot

@rss2tg\_bot – RSS-бот в Телеграмі. Нині має 9752 активних користувачів. Дуже простий у користуванні бот. Його головною перевагою є можливість використовувати необмеженої кількості RSS-посилань. Частота оновлень складає 10 хвилин. Присутня можливість кастомізації налаштувань відображення новин. Оформлення публікацій у бота @rss2tg\_bot (рис 2.3) має ті ж самі проблеми, що і попередній представник @Feed2Telegram\_bot.

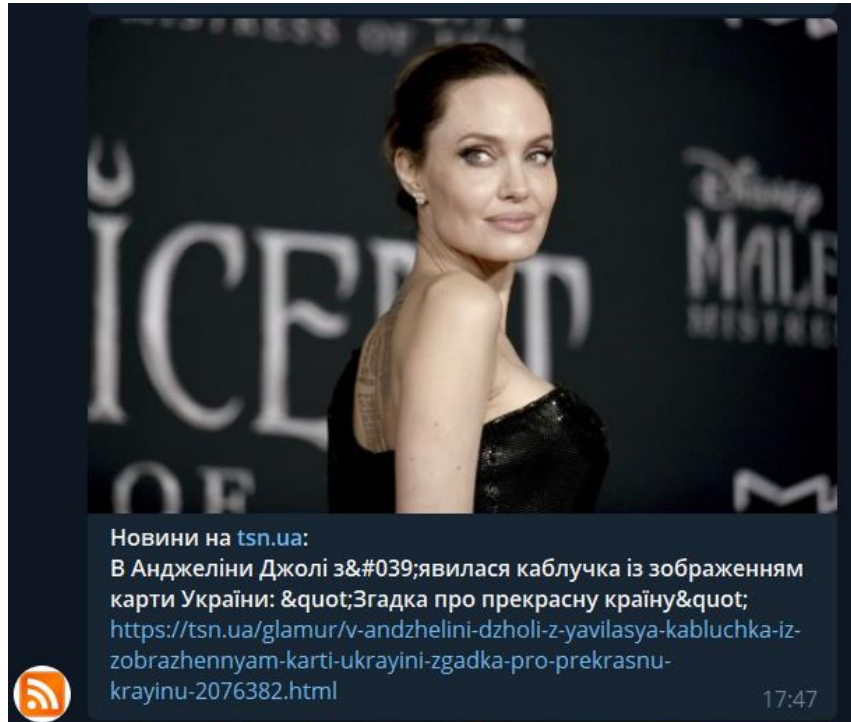


Рисунок 2.3 – Приклад оформлення публікацій у бота @rss2tg\_bot

Головні недоліки даного бота полягають у відсутності підтримки публікацій з соцмереж, неправильному форматуванні тексту, присутня лише англійська мова інтерфейсу, рекомендації не працюють, відсутній інтерфейс (лише команди), бот не має функції фільтрації та відбору контенту, а можливість його переадресації в інші Telegram канали чи чати поки знаходиться у стадії розробки.

## **2.2. Формування вимог до програми**

### **2.2.1. Вимоги до функціоналу програми**

Основні пункти які вдалося виділити щодо основного функціоналу, що не обхідно реалізувати:

- простий та зрозумілий функціонал;
- реалізувати не лише можливість додавати власні rss-стрічки, але й отримувати список перевірених офіційних джерел, аби запобігти дизінформуванню;
- також необхідно забезпечити актуальність інформації, що надається користувачу, тобто здійснювати перевірку оновлень щонайменше раз у 5 хвилин;
- реалізувати роботу адміністраторів боту з редагуванням списку офіційних/рекомендованих джерел (давати або видаляти RSS-стрічки);
- для комфортної роботи потрібно реалізувати можливість призупинити розсилку новин, але щоб при цьому зберігалися усі попередні налаштування стрічки;
- надати користувачу можливість відфільтровувати контент за допомогою категорій (якщо вони присутні для обраної RSS-стрічки);
- відображати статистику найпопулярніших інформаційних порталів та їх категорій (якщо такі наявні);
- організувати зворотній зв'язок користувачів із адміністраторами боту.

### **2.2.2. Вимоги щодо інтерфейсу користувача**

Ознайомившись з аналогами, було виявлено, що для ефективної та комфортної роботи, необхідно, щоб люди без спеціальної підготовки могли користуватися ботом. Саме тому було вирішено використовувати кнопочний інтерфейс, адже він є більш інтуїтивно зрозумілим, аніж команди (наприклад

«/help»). Бажано підібрати графічні піктограми (Емої) для кожної функції, аби покращити розуміння призначення кожної кнопки бота.

Вибір мови інтерфейсу також є важливим критерієм для комфортної роботи з чат-ботом. Для розробки було обрано англійську та українські мови, проте гнучкість архітектури дозволяє додати будь-яку кількість мов.

Так як потік інформації дуже стрімкий, необхідно лаконічно оформити публікацію, аби користувач швидко зрозумів про що йдеться у новині. Для цього була запропонована наступна схема (рис 2.4). За допомогою форматування тексту можна візуально виділити усі компоненти публікації.

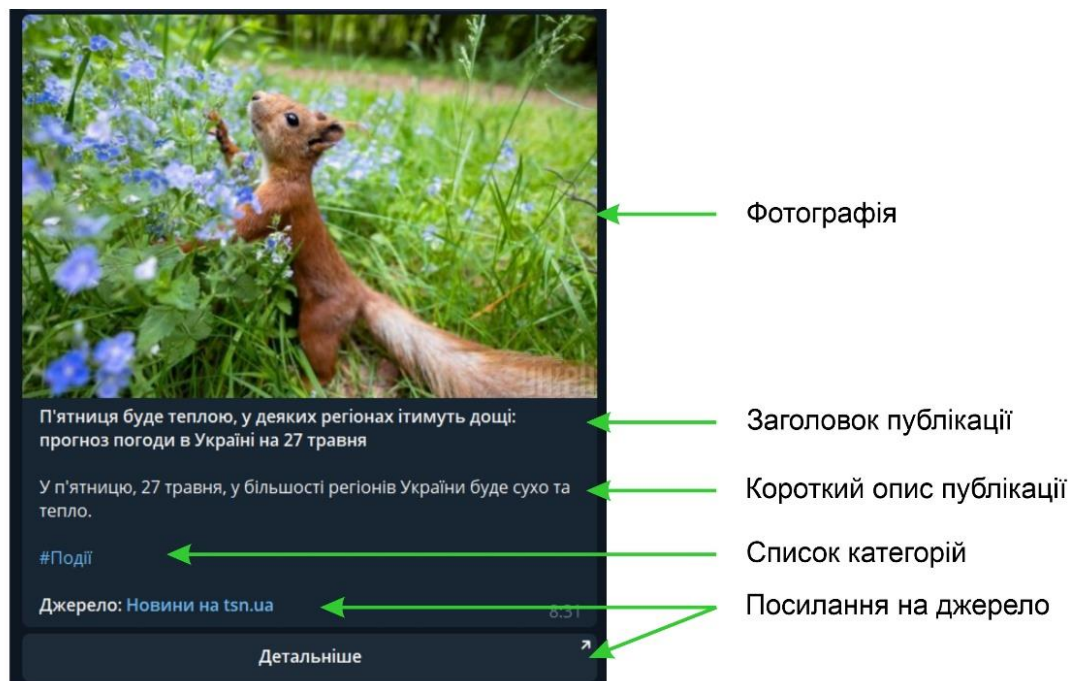


Рисунок 2.4 – Схема оформлення публікації

Для демонстрації категорій можна використовувати хештеги. Хештег (від англ. Hashtag – знак "решітка") – це ключове слово або кілька слів, що записуються після знака "#". Просунуті користувачі та команда Telegram активно використовуює хештеги. Головна їхня зручність полягає у легкому пошуку. Пошук по хештегах здійснюється у двох режимах – пошук по конкретному каналу або чату, або пошук по всіх повідомленнях у всіх каналах

і чатах, в яких ви складається. Це істотно спрощує координацію по величезних масивам публікацій, адже користувачі знаючи хештег можуть шукати інформацію по потрібній темі не докладаючи зусиль.

Для того щоб користувач міг детальніше ознайомитися з новиною необхідно додати посилання на джерело. Так як у користувача можуть виникати проблеми при натисканні курсором або пальцем по лінку, то можна також розмістити лінк у великій кнопці, яка прикріплюється внизу посту.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГАЦІЇ НОВИН З ІНТЕРФЕЙСОМ ТЕЛЕГРАМ-БОТА

### 3.1. Діаграма варіантів використання чат-бота

Для початку розробки бота треба спочатку продумати логіку програми та розробити сценарії взаємодії користувача з чат-ботом.

Визначивши основні функції, що мають бути присутніми у програмному продукті для агрегації новин, було побудовано діаграму варіантів використання (рис 3.1).



Рисунок 3.1 – Діаграма варіантів використання телеграм-бота для агрегування новин

Також було визначено, що для ефективної роботи необхідно реалізувати 2 ролі: користувач та адміністратор.

Актор Клієнт має такі високорівневі наміри:

- оперативно отримувати новини з сайтів на які він підписаний;
- підписуватися або відписуватися від різних RSS-стрічок;
- керувати власними підписками, що включає в себе фільтрування новин, перегляд списку підписок, додавання нових та видалення власних RSS-каналів;
- призупиняти/відтворювати розсилку.

Додатковий функціонал, що потребує реалізації:

- отримування рекомендованих інформаційних джерел;
- дізнатися статистику найпопулярніших каналів;
- можливість відправити фідбек розробнику;
- ознайомитися з інформацією про бота.

Для ролі адміністратора доступний весь функціонал звичайного користувача.

Додаткові функції, що реалізовано для адміністратора, а саме: отримування відгуків від користувачів й можливість редагування списку рекомендованих джерел, створені виключно для того, аби вносити глобальні зміни до бази даних та отримувати зворотній зв'язок з іншими користувачами агрегатора використовуючи лише телеграм-бот.

## **3.2. Обґрунтування вибору засобів для реалізації**

Далі в роботі наведено основні технології, що використовувались при розробці чат-боту для агрегування новин. Також було додано обґрунтування вибору обраних технологій.

### **3.2.1. Платформа Node.js на мові JavaScript**

Node.js – це серверна платформа, побудована на движку JavaScript V8 Chrome. Він використовує неблокуючу модель вводу-виводу, керовану подіями. Це дозволяє розробникам створювати додатки у реальному часі з

інтенсивним використанням даних, які запускаються на розподілених пристроях.

Node.js — це кросплатформне середовище виконання з відкритим вихідним кодом, яке використовується для розробки серверних веб-додатків. Програми Node.js написані на JavaScript і можуть запускатися в широкому діапазоні операційних систем таких, як: OS X, Microsoft Windows і Linux. [3]

Node.js базується на архітектурі, що керується подіями, і неблокуючому API вводу/виводу, який призначений для оптимізації пропускну здатності та масштабованості програми для веб-додатків у реальному часі.

Протягом тривалого періоду часу всі фреймворки, доступні для веб-розробки, базувалися на моделі без стану, коли дані, згенеровані під час одного сеансу (наприклад, інформація про налаштування користувача та події, що відбулися), не зберігаються для використання в наступному сеансі з цим користувачем.

Довелося багато працювати, щоб підтримувати інформацію про сеанс між запитами для користувача. Але з Node.js, нарешті, є спосіб для веб-додатків мати двостороннє з'єднання в режимі реального часу, коли клієнт і сервер можуть ініціювати зв'язок, дозволяючи їм вільно обмінюватися даними.

Асинхронне введення, керований подіями, допомагає одночасно обробляти запити – це, мабуть, найважливіша перевага Node.js. Ця функція в основному означає, що якщо Node отримує запит на якусь операцію введення/виведення, він виконає операцію у фоновому режимі та продовжить обробку інших запитів.

Node використовує механізм виконання JavaScript V8, який використовується в Google Chrome. Node має обгортку над механізмом JavaScript, що робить механізм виконання набагато швидшим, а отже, обробка запитів у Node також стає швидшою. [3]

Обробка одночасних запитів. Іншою ключовою функціональністю Node є можливість обробляти одночасні з'єднання з мінімальними витратами на один процес.

Бібліотека Node.js використовує JavaScript – це ще один важливий аспект розробки в Node.js. Значна частина спільноти розробників вже добре розбирається в javascript, і, отже, розробка на Node.js стає легшою для розробника, який знає javascript.

Для фреймворка Node.js існує активне та динамічне співтовариство. Завдяки активній спільноті для фреймворка завжди доступні оновлення ключів. Це допомагає підтримувати фреймворк завжди в курсі останніх тенденцій у веб-розробці.

Саме тому для реалізації чат-боту для агрегації новин було вирішено використовувати Node.js на мові JavaScript, так як він найкраще підходить для використання в потокових програмах або програмах реального часу на основі подій.

### **3.2.2. Прикладний програмний інтерфейс Telegram Bot API**

Після створення кожному боту призначається унікальний токен, який має схожу структуру: 123321:ABC-DEF4567pitOp-zyx54X2v1o123ew11. Більш детально про те, як отримати або замінити токен розписано у наступних розділах.

Всі запити до Telegram Bot API повинні здійснюватися через HTTPS у наступному вигляді: [https://api.telegram.org/ТОКЕН/НАЗВА\\_МЕТОДУ](https://api.telegram.org/ТОКЕН/НАЗВА_МЕТОДУ). [4]

Допускаються GET та POST запити. Для передачі параметрів у Bot API доступні 4 способи:

- запит в URL;
- application/x-www-form-urlencoded;
- application/json (не підходить для завантаження файлів);
- multipart/form-data (для завантаження файлів).

Відповідь прийде у вигляді JSON-об'єкта, в якому завжди буде булеве поле `ok` і опціональне поле `description`, що містить опис людини результату. Якщо поле `ok` істинно, запит пройшов успішно і результат його виконання можна побачити в полі `result`. У разі помилки поле `ok` дорівнює `false`, а причини помилки будуть описані в полі `description`. Крім того, у відповіді буде ціле поле `error_code`, але коди помилок будуть змінені в майбутньому.

Для полегшення роботи з Telegram Bot API використовувався фреймворк `grammY`. Він простий у використанні, значно спрощує та пришвидшує процес створення телеграм-ботів, гнучкий й масштабований. Його можна використовувати з TypeScript та JavaScript, також працює на Node.js, Deno і в браузері. Фреймворк `grammY` підтримує Telegram Bot API 6.0, який був випущений 16 квітня 2022 р.[6]

Боти взаємодіють з Telegram за допомогою HTTP-запитів. Щоразу, коли ваш бот надсилає або отримує повідомлення, HTTP-запити переходять назад і вперед між серверами Telegram і вашим сервером/комп'ютером. По суті, `grammY` реалізує все це спілкування за вас, тому ви можете просто ввести свій код, і повідомлення буде надіслано.

### **3.2.3. База даних – MongoDB**

Існує два основних типи баз даних: SQL (реляційна) і NoSQL (нереляційна). Реляційні бази даних зберігають дані в стовпцях і рядках. Бази даних NoSQL зберігають неструктуровані дані без схем у кількох колекціях і вузлах. Нереляційні бази даних не потребують фіксованих таблиць `schemas`. Бази даних NoSQL масштабуються по горизонталі і підтримують обмежені запити на приєднання.

Повністю всі дані розробленого телеграм-боту зберігаються у нереляційній (NoSQL) базі даних MongoDB.

MongoDB була створена в 2009 році як безкоштовна нереляційна база даних з відкритим вихідним кодом, яка може бути масштабована, надійна і безкоштовна. Існує також комерційна версія з ширшим функціоналом.[7]

MongoDB здобула репутацію універсальної, гнучкої бази даних і сьогодні використовується як серверне сховище даних багатьох високопоставлених компаній і організацій, таких як Forbes, Facebook, Google, IBM, Twitter та багатьох інших.

MongoDB використовує документи, які можуть містити піддокументи в складних ієрархіях, що робить його дуже гнучким. MongoDB може відображати об'єкти з будь-якої мови програмування, забезпечуючи легку реалізацію та обслуговування.

Користувач може вибірково індексувати деякі частини кожного документа або запиту на основі регулярних виразів, діапазонів або значень атрибутів і мати стільки властивостей для кожного об'єкта, скільки потрібно прикладному шару.

Бази даних NoSQL дешевші та простіші в обслуговуванні. Вони мають такі функції, як: легший розподіл даних, простіші моделі даних та автоматичне відновлення. Ці переваги вимагають менших адміністративних витрат. Бази даних NoSQL використовують дешевші сервери, тому ціна зберігання та обробки даних за один концерт значно нижча.

MongoDB підтримує інтегроване кешування, що підвищує продуктивність виведення даних.

Оскільки бази даних NoSQL, такі як MongoDB, розширюються горизонтально, ви можете масштабувати, додаючи більше машин до свого пулу ресурсів. Також важливою перевагою є відсутність проблем зі схемами. Ви можете помістити дані в базу даних NoSQL, не вимагаючи попередньо визначеної схеми, тож ви можете змінити модель даних і формати, не порушуючи роботу програм.

MongoDB пропонує безліч корисних функцій (спеціальні запити, агрегація, обмежене збирання, зберігання файлів, індексація, балансування навантаження, реплікація, виконання JavaScript на стороні сервера), що робить його зручною базою даних.

#### **3.2.4. Хостинг – EvenNode**

Так як агрегатори новин вимагають, щоб бот був у мережі весь час, то слід завантажити свій код до провайдера хостингу (іншими словами, на чужий комп'ютер, також відомий як сервер), і дозволити цим їм запуснути його за вас. Існує незліченна кількість компаній, які дозволяють безкоштовно запускати свого бота Telegram.

Для стабільної роботи чат-бот було розміщено на віддаленому сервісі EvenNode. Він дозволяє просто та швидко розгортати програми Node.js і Python у хмарі за допомогою git або FTP та дає змогу легко масштабувати їх у будь-який час. EvenNode підтримує роботу з базою даних MongoDB.

Також там присутнє автоматичне перемикання на помилку, тобто більше не доведеться турбуватися про неочікувані простої сервера. Ваша програма автоматично переходить до працездатного сервера, щойно виявлено відключення сервера.

EvenNode дозволяє зручно завантажувати свою програму через FTP або SFTP за допомогою TLS/SSL. WinSCP — популярний клієнт SFTP і FTP-клієнт для Microsoft Windows. Можна просто скопіювати файл між локальним комп'ютером та віддаленими серверами за допомогою протоколів передачі файлів FTP, FTPS, SCP, SFTP, WebDAV або S3. Щойно файл завантажено, автоматично перебудовується програма на EvenNode, щоб розгорнути останню версію.

### 3.3. Структура бази даних

Для реалізації поставлених задач нам необхідно зберігати дані про джерела та користувачів. Тому було розроблено наступну структуру бази даних, що складається з двох колекцій News та User. Кожна таблиця виконує свою функцію.

У колекції News зберігається інформація про усі RSS-порталами, що присутні. Кожен запис у таблиці News має наступні поля (рис 3.2):

- `_id`: (String) — унікальний ідентифікатор, що визначає гіперпосилання на RSS-стрічку;
- `title`: (String) — визначає назву каналу;
- `link`:(String) — визначає гіперпосилання на веб-сайт, з яким встановлений канал зв'язку;
- `time`: (Int) — час останньої відправленої публікації;
- `categories`: (Array of String) — масив усіх доступних категорій для даного джерела;
- `isPrivate`: (Boolean) — визначає чи буде відображатися канал у списку рекомендованих джерел.

```

_id: "https://tsn.ua/rss/full.rss"
title: "Новини на tsn.ua"
link: "https://tsn.ua/rss/full.rss"
time: 1654411320000
categories: Array
  0: "Війна"
  1: "Спорт"
  2: "Економіка"
  3: "Львів"
  4: "Події"
  5: "Гламур"
  6: "за кордоном"
  7: "Ексклюзив ТСН"
  8: "Політика"
  9: "Київ"
  10: "Астрологія"
  11: "ТСН. Спецвипуск"
  12: "Здоров'я"
  13: "Вікно"
  14: "Корисні статті"
  15: "Технологія"
  16: "Іжа"
  17: "Коронавірус"
isPrivate: false

```

Рисунок 3.2 – Приклад структури запису в таблиці News

У колекції Users зберігається інформація про усіх користувачів, що користуються ботом на даний момент. Кожен запис у таблиці Users має наступні поля (рис 3.3):

- `_id`: (String) — унікальний ідентифікатор користувача;
- `firstName`: (String) — визначає назву каналу;
- `news`: (Array of Objects) — масив, що включає в себе об'єкти, котрі репрезентують усі джерела (link) й відповідні їм категорії (categories) на які підписаний користувач;
- `state`: (String) — визначає у якому розділі меню знаходиться користувач;
- `stateData`: (String) — допоміжне поле для коректного відображення меню з фільтрами;
- `isPause`: (Boolean) — визначає чи призупинено розсилання новин;
- `language`: (String) — мова інтерфейсу користувача.

```

_id: 299091890
firstName: "Daryna"
news: Array
  0: Object
    link: "https://gordonua.com/ukr/xml/rss_category/top.html"
    categories: Array
  1: Object
    link: "https://24tv.ua/rss/all.xml"
    categories: Array
  2: Object
    link: "https://tsn.ua/rss/full.rss"
    categories: Array
  3: Object
    link: "https://www.pravda.com.ua/rss/view_news"
    categories: Array
state: "main_menu"
stateData: null
isAdmin: true
isPause: false
language: "ua"

```

Рисунок 3.3 – Приклад структури запису в таблиці Users

### 3.4. Реєстрація телеграм-бота

Щоб створити власного телеграм-бота, необхідно спочатку завантажити програму Telegram для комп'ютера. Коли це буде зроблено, вам потрібно буде взаємодіяти з основним телеграм-ботом, BotFather. Необхідно знайти його на вкладці пошуку в месенджері Telegram. Щоб запустити бота, введіть команду **/start**. Botfather дасть вам список команд, за допомогою яких можна налаштувати свого власного бота.

Команда **/newbot** попросить вас назвати свого бота та дати йому ім'я користувача.

Назва – це те, що користувачі побачать у розмові з ботом. Вони знайдуть бота за ім'ям користувача. Бажано дати боту дружнє ім'я, яке містить пробіли, наприклад, «New of UA».

Ім'я користувача слугує ідентифікатором вашого бота. У ньому не повинно бути пробілу, а суфіксом має бути слово «bot». Він має містити від 5 до 32 символів і може включати латиницю, цифри або символи підкреслення.[5] Ми дали нашому боту ім'я користувача @newofUA\_bot.

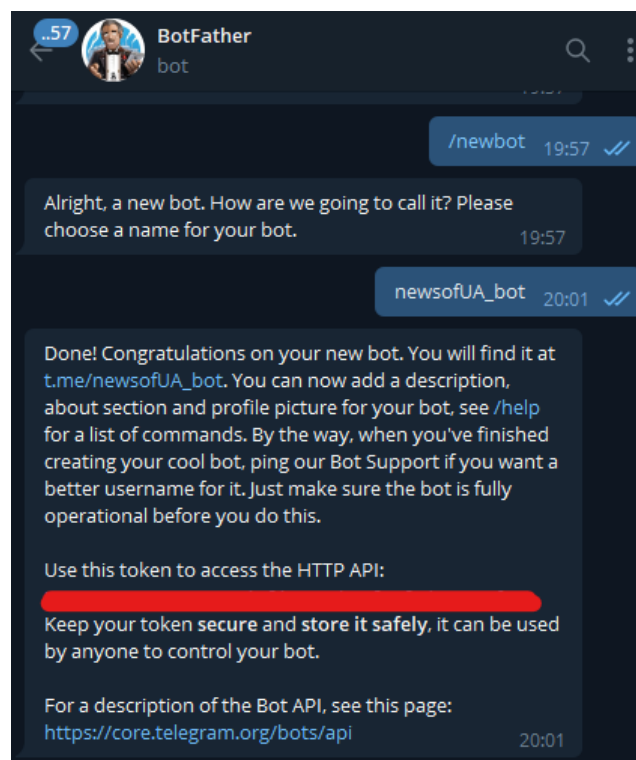


Рисунок 3.4 – Присвоєння унікального токєну

Після створення імені користувача, ви отримуєте токен (рис 3.4), який потрібен для керування ботом та надсилання його в API ботів. Його необхідно зберегти у окремому файлі конфігурації і нікому не показувати, щоб захистити телеграм-бот від зловмисників.

### 3.5. Розробка програмного забезпечення для агрегації новин

Після отриманню токена, перейдемо до реалізації самого телеграм-бота. Для початку зробимо графічне представлення інфраструктури, на яку буде розгорнуто додаток (рис 3.4). Діаграма розгортання — це діаграма, яка показує конфігурацію вузлів обробки під час виконання та компонентів, які на них працюють. Тобто це різновид структурної діаграми, яка використовується при моделюванні фізичних аспектів об'єктно-орієнтованої системи. Вони часто використовуються для моделювання статичного вигляду розгортання системи та допомагають оптимізувати компоненти, їхні залежності один від одного та продуктивність системи.

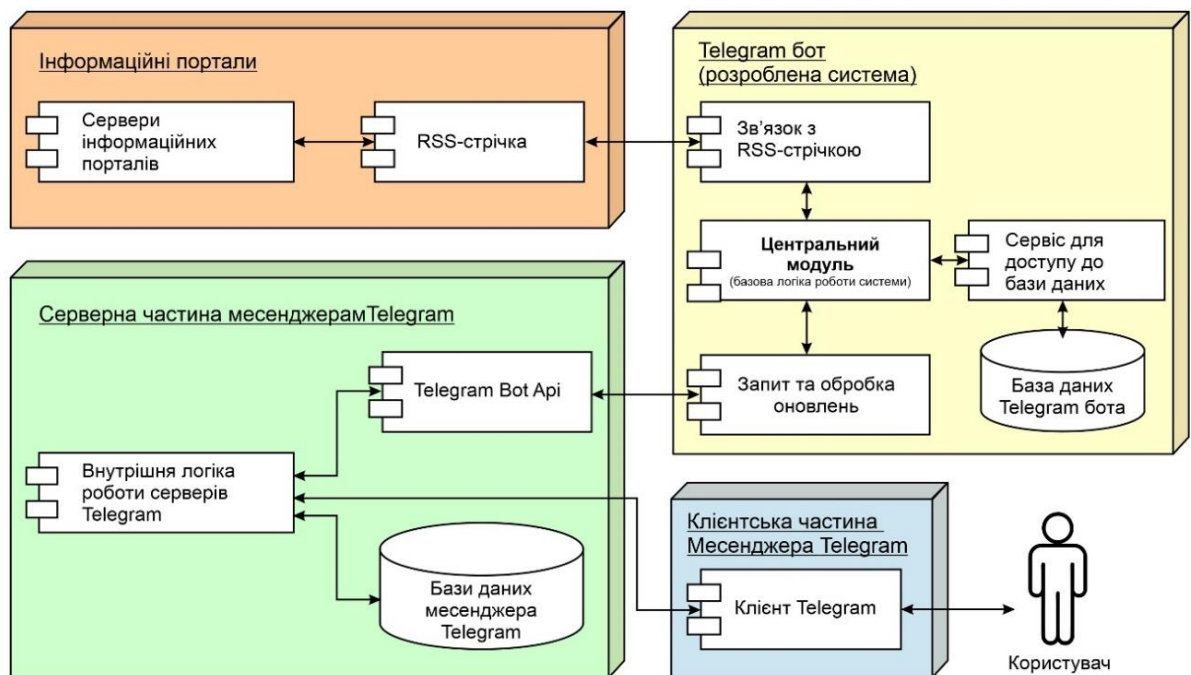


Рисунок 3.4 – Діаграма розгортання телеграм-боту для агрегування новин

Для роботи з базою даних телеграм-бота реалізовано модуль `dbApi`. У ньому відбувається підключення до бази `MongoDB` та описані основні необхідні запити для коректної роботи програми.

Додатку часто потрібно взаємодіяти з веб-серверами, щоб отримати різні ресурси. Використовуючи `JavaScript` на стороні клієнта, цього можна досягти за допомогою `API` вибірки та `window.fetch()` функції. У `NodeJS` кілька пакетів/бібліотек можуть досягти того ж результату. Одним з них є `node-fetch` пакет.

`node-fetch` — це легкий модуль, який дозволяє нам використовувати цю `fetch()` функцію в `NodeJS`, з дуже схожою функціональністю, як і `window.fetch()` в рідному `JavaScript`, але з деякими відмінностями.

Для реалізації зв'язку з `RSS`-стрічками було написано окремий модуль `handleNewsUpdate`, що періодично запускає модуль `getRSSInfo` (рис 3.5). Його основна мета перевіряти канали на наявність оновлень. Замість впровадження `XMLHttpRequest` в `Node.js` для запуску специфічного для браузера `Fetch polyfill`, чому б не перейти від рідного `http` до `fetchAPI` безпосередньо.

```
import { XMLParser } from "fast-xml-parser";
import fetch from "node-fetch";

export const getRSSInfo = async (url) => {
  const response = await fetch(url);
  const coding = parseCharsetFromContentType(
    response.headers.get("content-type")
  );
  const arrayBuffer = await response.arrayBuffer();
  const decoder = new TextDecoder(coding);
  const textDecoded = decoder.decode(arrayBuffer);
  const parser = new XMLParser({
    ignoreAttributes: false,
  });
  const result = parser.parse(textDecoded);
  return {
    title: result.rss.channel.title,
    link: url,
    time: Date.now(),
    categories: getAllCategories(result),
  };
};
```

Рисунок 3.5 – Фрагмент коду що описує парсинг `RSS`-стрічки

fast-xml-parser використовується для перевірки та перетворення XML до об'єкта JS без бібліотек на основі C/C++ і без зворотного виклику. Він ніж будь-яка чиста реалізація JS, може обробляти великі файли та підтримує непарні теги (наприклад `<br>`, у HTML), вузли зупинки (наприклад, `<script>` у HTML, коментарі і багато інших функцій.

При коректній обробці RSS-стрічка перетворюється на з XML на JS Object, що значно полегшує подальшу роботу. Далі вилучаються останні новини в період з попередньої обробленої публікації. Обробляються категорії, щоб не було повторів і відправляється до центрального модулю, де новина форматується і розсилається користувачам.

## РОЗДІЛ 4. ТЕСТУВАННЯ Й ОГЛЯД РОЗРОБЛЕНОГО БОТА

### 4.1. Функції та команди бота

Після закінчення розробки отримано робочого бота, який має наступний функціонал:

- підписка на необмежену кількість джерел;
- можливість додати власну RSS-стрічку;
- список рекомендованих джерел;
- фільтрування новин за категоріями;
- можливість призупинити/відновити розсилку новин;
- перегляд підписок користувача (джерела та категорії);
- статистика найпопулярніших джерел;
- вибір мови інтерфейсу (українська, англійська);
- зворотній зв'язок з адміністраторами;
- інформація про бота.

Аби знайти бота, необхідно в пошуку в месенджері Telegram ввести тег @newsofUA\_bot та обрати бота з назвою «News of UA». Для початку роботи з ботом (або для його перезапуску) потрібно ввести команду /start або натиснути на однойменну кнопку. Після того, як новий користувач заходить в чат-бот – йому надають можливість обрати мову інтерфейсу. На даний момент доступно лише дві мови, а саме: українська та англійська. Після цієї дії, відображається повідомлення з короткою інформацією про бота та головне меню у вигляді кнопок (рис 4.1).

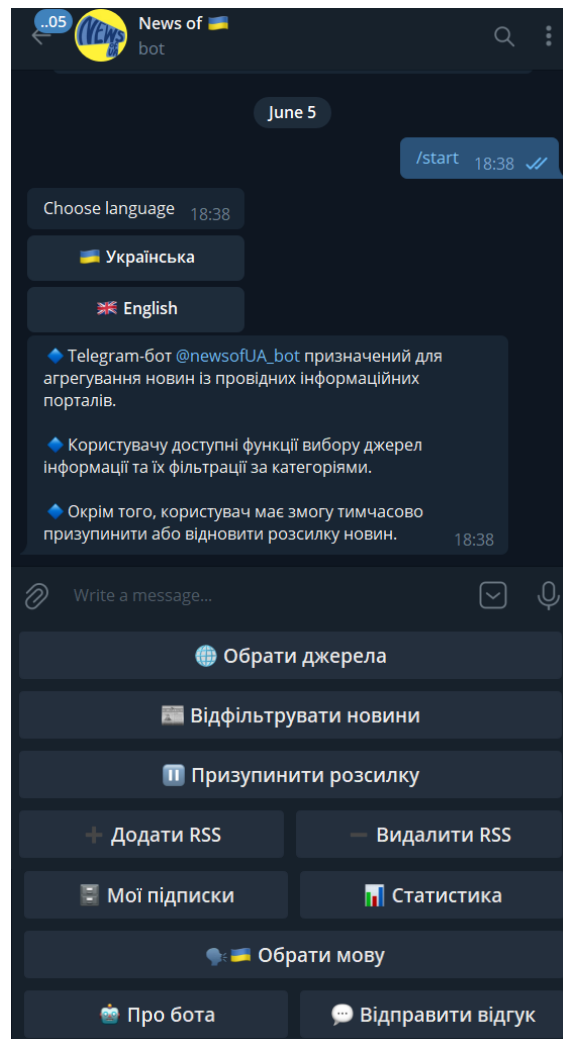


Рисунок 4.1 – Головне меню телеграм-бота

Кожна кнопка відповідає за певну функцію яку може виконати бот.

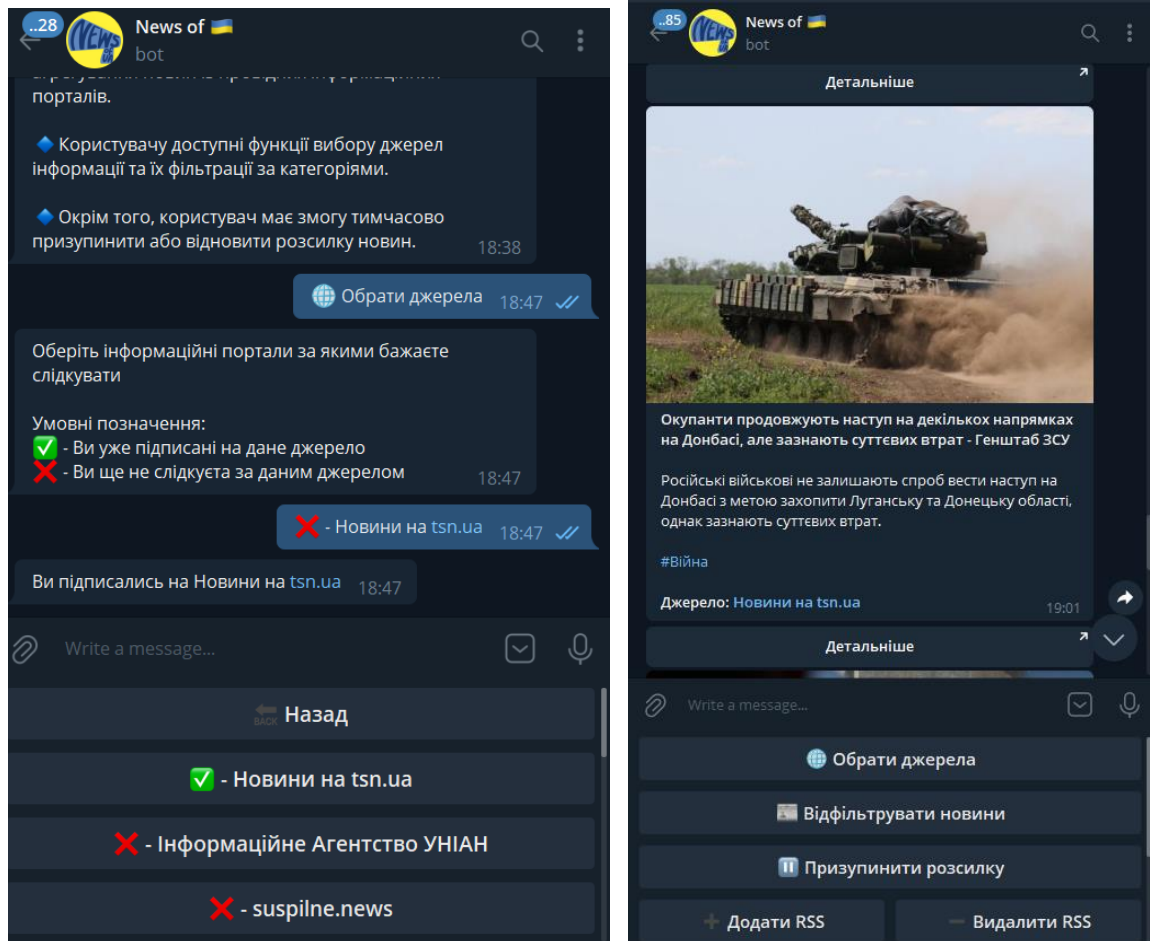
**"Обрати джерела"** — показує список джерел, на які можна підписатися або відписатися. Даний список виконаний у вигляді кнопок, щоб користувачу було зручніше керувати своїми підписками (рис 4.2).

Далі необхідно інформаційні портали за якими бажаєте слідкувати. Біля кожного джерела є умовні позначення:

✓ — означає, що ви уже підписані на дане джерело;

X — вказує, що ви ще не слідкуєте за даним джерелом.

Інтервал перевірки на наявність оновлень на RSS-стрічках складає 5 хвилин. Тому через невеликий проміжок часу можна побачити нові публікації.



а)

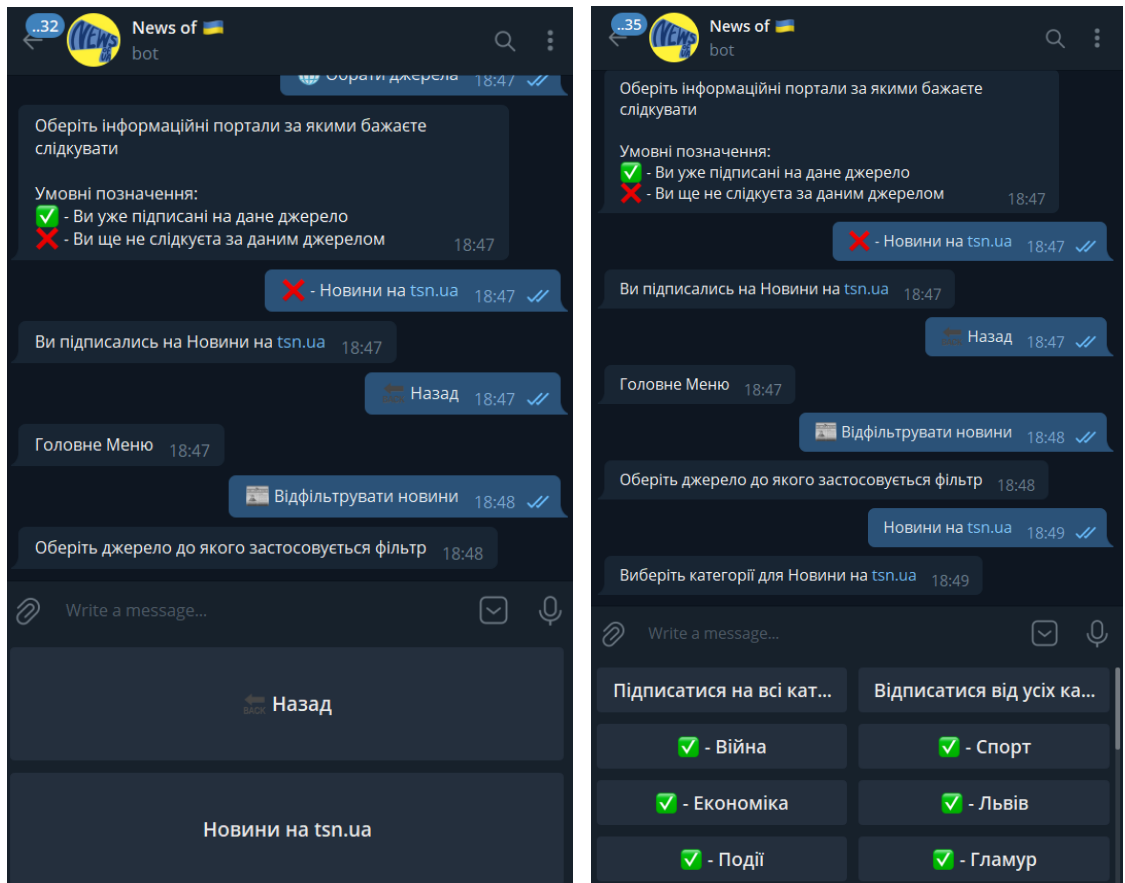
б)

Рисунок 4.2 – Приклад: а) процесу обрання джерел;  
б) відображення новин

**"Відфільтрувати новини"** — переводить у меню вибору джерела, до якого хочете застосувати фільтр (рис 4.3). Список виконаний у вигляді кнопок, тому після натискання на необхідне джерело висвічується кнопкове меню де відображені усі доступні категорії обраного порталу. Також для зручності є кнопки «Підписатися на всі категорії» та «Відписатися від усіх категорій» для швидкого вибору необхідних категорій. Для того щоб повернутися до головного меню в усіх меню присутня кнопка **"Назад"**.

**"Призупинити розсилку"** — скасування відправки новин.

**"Відновити розсилку"** — відновлення відправки новин.



а)

б)

Рисунок 4.3 – Приклад процесу фільтрації новин: а) вибір джерела до якого застосувати фільтр; б) вибір категорій

Кнопка "**Додати RSS**" відповідає за додавання власного RSS до списку обраних джерел (відображається лише для користувача, що його додав). Після її натискання вам необхідно ввести посилання на RSS-стрічку. Якщо стрічку успішно розпізнано, то висвітиться відповідне повідомлення та користувача підпише на доданий RSS-канал (рис 4.4).

"**Видалити RSS**" — видалення зі списку обраних джерел раніше доданої користувачем RSS-стрічки. Процедура схожа на попередній пункт, де ми додавали власний канал. Для цього потрібно ввести посилання на RSS-стрічку, яку бажаєте вилучити. Якщо стрічку успішно видалено, тоді бот відправить повідомлення про те, що операція пройшла успішно. Видалений канал більше не буде відображатися у списку обраних джерел користувача.

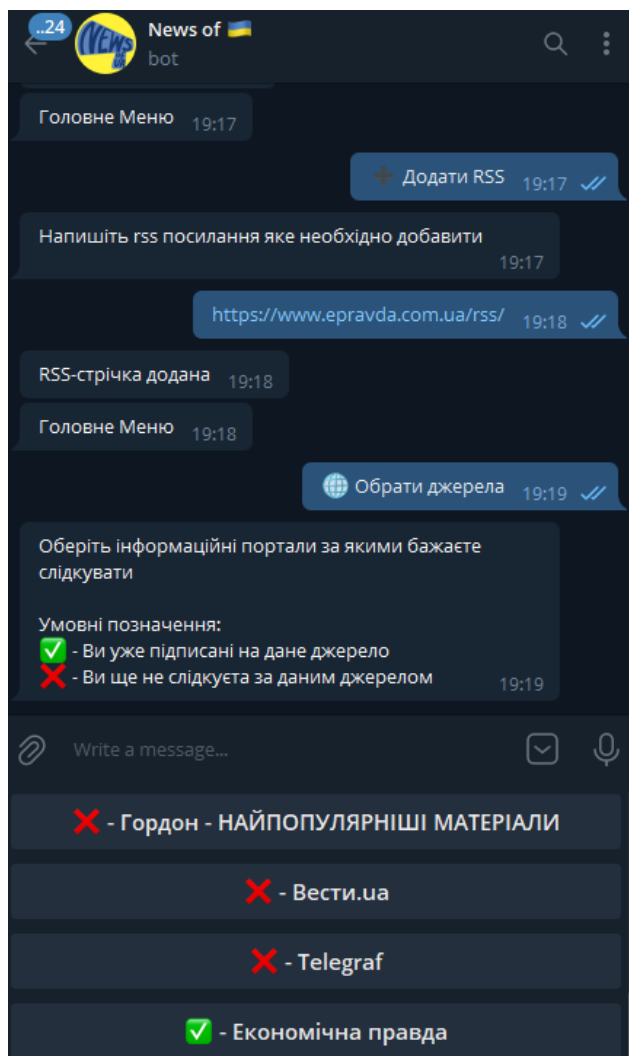


Рисунок 4.4 – Приклад додавання власної RSS-стрічки

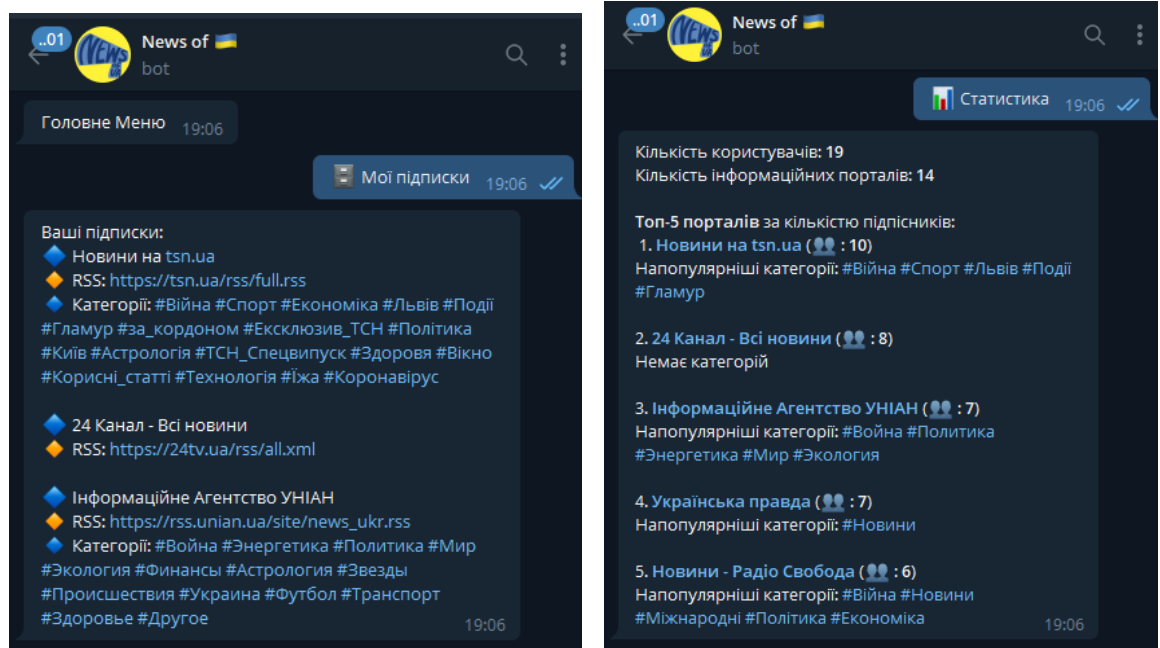
**"Статистика"** — показує статистику використання бота, а саме кількість активних користувачів, кількість порталів, що відслідковує бот у даний момент та рейтинг найпопулярніших RSS-стрічок та їх категорій серед користувачів даного бота.

**"Мої підписки"** — друкує список усіх інформаційних джерел та категорії на які підписаний користувач.

**"Про бота"** — коротка інформація про даний телеграм-бот.

**"Відправити відгук"** — зворотній зв'язок з адміністраторами боту.

**"Обрати мову"** — вибір мови інтерфейсу бота. На цей момент бот підтримує українську та англійську мови.



а)

б)

Рисунок 4.5 – Приклад: а) відображення підписок користувача; б) приклад відображення статистики

У адміністратора присутні ще 2 додаткові кнопки:

- **"Додати RSS (admin)"** — додавання RSS-стрічки до списку рекомендованих джерел (відображається для усіх користувачів);
- **"Видалити RSS (admin)"** — видалення RSS-стрічки зі списку рекомендованих джерел.

## 4.2. Тестування системи

Перевірка коректності роботи розробленого чат-боту для агрегування новин тестувалася вручну на мобільному пристрої Redmi Note 8 Pro з версією Telegram v 8.7.4, а також на офіційному застосунку для комп'ютера Telegram Desktop v 3.7.3. Листування у десктопній версії синхронізується з мобільним пристроєм.

Було перевірено роботу всіх доступних команд, а також розглянуто поведінку бота при деяких крайніх випадках, а саме коректну поведінку бота при наступних сценаріях:

- Додавання некоректного посилання на RSS-стрічку.

Реакція боту: Користувачу приходить повідомлення, що посилання на RSS-стрічку не правильне та дає додаткову спробу для введення посилання.

- Можливість повторно додати власну RSS-стрічку.

Реакція боту: Користувачу приходить повідомлення, що дана RSS-стрічка уже додана у користувача.

- Можливість повторно додати власну RSS-стрічку, коли вона уже додана адміністратором до списку рекомендованих джерел.

Реакція боту: Користувачу приходить повідомлення, що дана RSS-стрічка уже додана до списку рекомендованих джерел та автоматично підписує користувача на цю стрічку.

- Видалення RSS-стрічки, коли її не існує.

Реакція боту: Користувачу приходить повідомлення, що даної RSS-стрічки немає у базі даних.

- Спроба видалення RSS-стрічки, що вже додана адміністратором до списку рекомендованих джерел.

Реакція боту: Якщо користувач є адміністратором, то стрічка видаляється зі списку рекомендованих джерел, але не видаляється з бази даних, щоб користувачі які були підписані на дане джерело продовжували отримувати новини (статус джерела змінюється з публічного на приватний). Якщо користувач не є адміністратором, то приходить повідомлення, що у нього недостатньо прав для здійснення команди.

- Видалення приватної RSS-стрічки.

Реакція боту: Користувачу приходиться повідомлення, що дана RSS-стрічка видалена, якщо на стрічці був підписаний лише один користувач, то її запис видаляється у базі даних.

- Коректне відображення меню з «Відфільтрувати новини».

Реакція боту: відображати лише ті джерела, які мають можливість фільтрації за категоріями, аби користувач не робив зайвих рухів.

- Справність роботи телеграм-боту при перемиканні мови.
- Реакція боту: зміна мови інтерфейсу відбувається швидко й без затримок.

Усі тести, що стосувалися коректної роботи, працездатності та релевантності отриманих відповідей від чат-бота успішно пройшли перевірку.

### **4.3. Розширення функціоналу**

Отримана архітектура являється досить гнучкою і надає можливість розширення функціоналу в майбутньому. Наприклад, впровадити моніторинг не лише за допомогою RSS або Atom стрічок, але й соціальних мереж, таких як Instagram, Facebook та YouTube. Також існує перспектива додати можливість користувачам використовувати даний телеграм-бот для своїх каналів або групових чатів.

## **ВИСНОВКИ**

У результаті виконання роботи було досягнуто поставленої мети, яка полягала у проектуванні та розробці програмного забезпечення для агрегування новин з інтерфейсом телеграм-бота. Було досліджено існуючі засоби для агрегування новин та їх принцип роботи.

Основний функціонал, який надає бот, полягає у оперативному отриманні останніх публікацій з провідних інформаційних порталів. Для цього було розроблено технічне завдання, спроектовано та розроблено бота.

Під час виконання роботи було визначено основні принципи створення телеграм-ботів. Також було реалізовано та протестовано чат-бот який надає змогу виконувати основні функції для агрегації новин, що полегшують процес отримання інформації з різних джерел, зокрема було реалізовано фільтрування публікацій за категоріями. Для стабільної роботи чат-бот було розміщено на віддаленому сервісі EvenNode.

Отже, даний розроблений проєкт є актуальним та перспективним, адже отримана архітектура являється досить гнучкою і надає можливість розширення функціоналу в майбутньому для покращення та полегшення процесу отримання новин з провідних інформаційних порталів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Опитування USAID-Internews щодо споживання медіа «Ставлення населення до медіа та споживання різних типів медіа 2021» [Електронний ресурс]. – Режим доступу: <https://detector.media/infospace/article/193918/2021-11-17-bilshist-ukraintsiv-otrymuie-novyny-z-sotsmerezh-ta-onlayn-vydan-doslidzhennya/> – Дата доступу: травень 2022.
2. Що таке RSS? Інформація про RSS в популярній формі [Електронний ресурс]. – Режим доступу: <https://web.archive.org/web/20090129120059/http://rssfeed.in.ua/>
3. Документація Node.js v17.9.1 [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest-v17.x/api/documentation.html> – Дата доступу: травень 2022.
4. Документація Telegram Bot API [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/bots/api>. – Дата доступу: травень 2022.
5. Telegram FAQ [Електронний ресурс]. – Режим доступу: <https://telegram.org/faq> – Дата доступу: травень 2022.
6. Документація grammY [Електронний ресурс]. – Режим доступу: <https://grammy.dev/guide/introduction.html#how-to-write-a-bot> – Дата доступу: травень 2022.
7. Документація MongoDB [Електронний ресурс]. – Режим доступу: <https://www.mongodb.com/docs/manual/introduction/> – Дата доступу: травень 2022.