

СИСТЕМА ПОШУКУ МЕДІАКОНТЕНТУ У МЕСЕНДЖЕРІ TELEGRAM

* Національний університет “Чернігівська політехніка”, м. Чернігів, Україна

** Таврійський національний університет ім. В.І. Вернадського, м. Київ, Україна

Анотація. Розроблено автономний Telegram-бот, який буде виконувати пошук медіаконтенту (фільмів, мультфільмів та серіалів) із відображенням назви відеоматеріалу, категорій, основних тегів, опису, касту, рейтингів, посилань на сторонні ресурси та списку схожого контенту. Telegram-бот має спеціальний інтерфейс для адміністратора зі статистикою використання та можливістю розсилки повідомлень. Для реалізації чат-бота було вирішено використати середовище розробки WebStorm. Як мову програмування було обрано мову TypeScript на платформі NodeJS, так як вона є сумісною з JavaScript і компілюється в останній. Після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою. Мова TypeScript відрізняється від JavaScript можливістю явного статичного визначення типів, підтримкою використання повноцінних класів, а також підтримкою підключення модулів, що дає можливість підвищити швидкість розробки, полегшити читабельність, рефакторинг і повторного використання коду, допомогти здійснювати пошук помилок на етапі розробки і компіляції і, можливо, прискорити виконання програм. Для реалізації клієнтської частини було створено й налаштовано бота в Telegram. Ця можливість реалізована в середині самого месенджера. Для прискорення розробки та спрощення взаємодії з Telegram використовувався сучасний фреймворк для NodeJS – Telegraph. Базою даних виступала Cloud Firestore – це гнучка хмарна NoSQL-база даних, що масштабується, від Firebase і Google CloudPlatform для інтернету, мобільних платформ і серверних додатків. База даних Cloud Firestore підтримує гнучкі ієрархічні структури даних, зберігає дані в документах, які, у свою чергу, зберігаються в колекціях. Документи можуть мати вкладення, об'єкти і підколекції.

Ключові слова: Telegram-бот, хмарна NoSQL-база даних, TypeScript, веб-сервер.

Abstract. An autonomous Telegram bot aimed at searching for media content (movies, cartoons and TV series) displaying video titles, categories, main tags, description, cast, ratings, links to third-party resources and a list of similar content is developed. This Telegram-bot has a special interface for the administrator with usage statistics and the ability to send messages. It was decided to use the WebStorm development environment to implement the chatbot. The TypeScript language on the NodeJS platform was chosen as the programming language, because it is compatible with JavaScript and is compiled in it. After the compilation, the program on TypeScript can be run in any modern browser or used in conjunction with the server platform. TypeScript differs from the JavaScript language by its ability to explicitly statically define types, support the use of full-fledged classes and support the connection of modules, which is designed to increase the speed of development, facilitate readability, refactoring and reuse of the code, help to debug at the development and compilation stage, and possibly speed up program execution. The Telegram bot was created and configured to implement the client part. This feature is implemented inside the messenger itself. To speed up the development and simplify the interaction with Telegram, a modern framework for NodeJS – Telegraph – was used. Cloud Firestore was used as a database. It is a flexible, scalable cloud-based NoSQL database from Firebase and Google CloudPlatform for the Internet, mobile platforms and server applications. Cloud Firestore database supports flexible, hierarchical data structures, stores data in documents which, in their turn, are stored in collections. Documents can have attachments, objects and subcollections.

Keywords: Telegram-bot, NoSQL cloud database, TypeScript, web-server.

1. Вступ. Постановка проблеми

Системи пошуку інформації щоденно використовуються сотнями мільйонів людей по всьому світу. У свою чергу, сучасні месенджери всіляко намагаються інтегрувати в середину себе якомога більше сервісів, щоб кінцевий споживач не відчував потреби залишати додаток, а міг все зробити в одному місці. Месенджери перетворилися з засобів для спілкування між людьми у засоби для отримання інформації, а теоретично навіть більше. Ключовою частиною при їх використанні є спілкування із створеними ботами, що значно розширює можливості месенджера.

Метою статті є розробка чат-бота, який абсолютно безкоштовний, пропонує базовий функціонал платних платформ та доступний максимально широкій аудиторії і може працювати на будь-яких платформах. У статті розглядаються основні принципи розробки ботів та можливості використання інтеграцій для максимізації позитивного досвіду користувача.

2. Аналіз досліджень і публікацій

Telegram – унікальний за своєю структурою месенджер, що є кросплатформним додатком. Крім стандартного обміну повідомленнями та стікерами в діалогах і групах, у месенджері можна зберігати необмежену кількість файлів, вести канали (мікроблоги), створювати і використовувати боти [1].

За допомогою спеціального API сторонні розробники можуть створювати боти та спеціальні акаунти, які керовані програмами. Типові боти відповідають на спеціальні команди в персональних і групових чатах, також вони можуть здійснювати пошук в інтернеті або виконувати інші завдання, застосовуються в розважальних цілях або в бізнесі [2].

Telegram-боти являються різновидом чат-ботів. Їх суть полягає у відповідній реакції на певні повідомлення від користувачів. Таким чином, сфера їх застосування безмежна. Використання бота в Telegram просте: достатньо вибрати його з каталогу, перейти за посиланням або знайти за ім'ям через пошук і вступити з ним у переписку [3].

Для запуску бота вводиться команда /start або пропонується натиснути кнопку старту на віртуальній клавіатурі. Для роботи можуть використовуватись програмовані кнопки віртуальної клавіатури, за рахунок яких бот перетворюється у міні-додаток з інтуїтивно зрозумілим інтерфейсом. Далі бот надішле вам інформацію про себе, інструкції використання, а також список команд або виведе доступні команди (кнопки на екрані). Що ж стосується безпеки, то в рамках самого Telegram боти не можуть завдати шкоди, тому що за своєю суттю це ті ж акаунти користувачів, тільки управляються алгоритмічно. Як і звичайного користувача, бот можна видалити і заблокувати. Важливо, що бот не виступає ініціатором бесіди. Це користувач перший відправляє йому запит, тому без дозволу користувача бот не може потурбувати.

Компанія Uber, деякі банки, великі магазини і багато інших організацій використовують Telegram-боти для спрощення та автоматизації внутрішніх робочих процесів. При реєстрації бота видається унікальний ключ, за допомогою якого в подальшому і буде відбуватися зв'язок між клієнтом і сервером. Така схема виключає необхідність додаткових налаштувань клієнт-серверної архітектури, так як все відбувається автоматично і займає декілька рядків програмного коду. Також важливим пунктом для кожного розробника є документація, проте в обраному месенджері вона досить інформативна і детальна.

3. Огляд існуючих рішень пошуку медіаконтенту

Серед величезного різноманіття ботів для Telegram знайти бота з детальною інформацією про фільми, серіали, акторів, режисерів – це дійсно непросте завдання. Розглянемо більш розповсюджені варіанти ботів [4].

1. Бот KіnoxaBot володіє таким функціоналом:

- дозволяє знайти фільм;
- має детальну інструкцію щодо використання;
- у складі має списки популярних і нових фільмів.

Проте можливості такого бота надто обмежені, і він підтримує тільки одну мову (рис. 1).

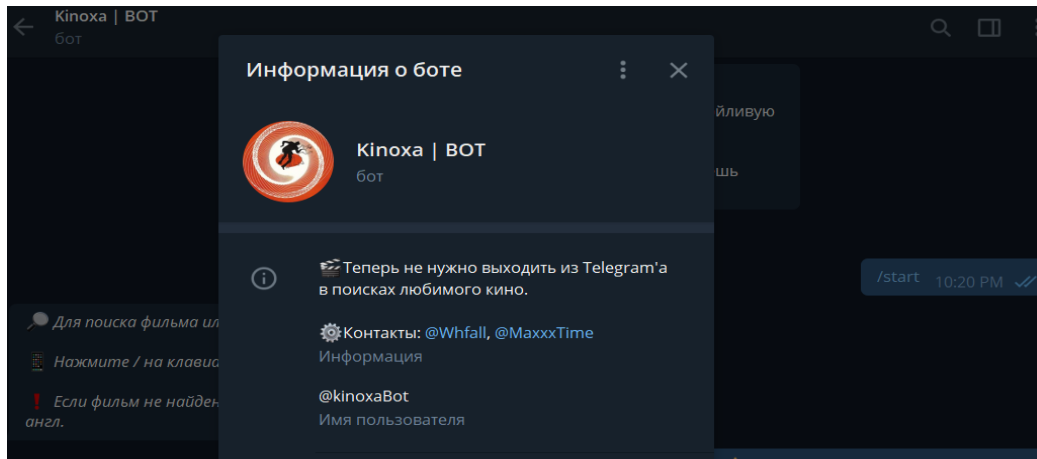


Рисунок 1 – Головна сторінка бота KіноxaBot

2. У боті CinemaKing багато можливостей, такі як:

- пошук фільмів із зручними фільтрами;
- можливість перегляду деяких фільмів;
- детальний опис статусу користувача;
- зручна навігація;
- відображення популярних запитів;
- розділ допомоги (зі зворотним зв'язком і рекламою);
- топи популярних сервісів за часом.

Проте цей бот має обмеження за кількістю фільмів, рекламою, платною системою підписки, формально підтримка англійської мови заявлена, але вона працює некоректно (рис. 2).

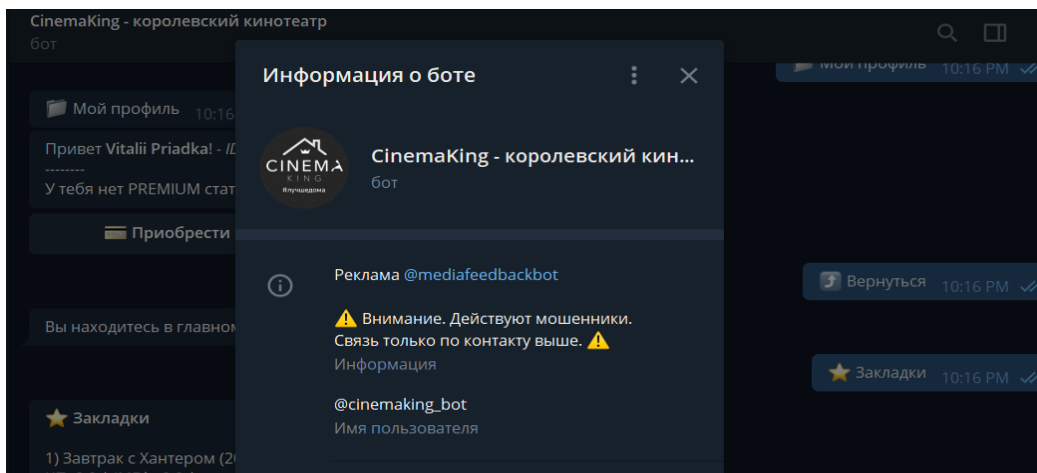


Рисунок 2 – Головна сторінка бота CinemaKing

3. Бот Кіноман включає в себе (рис. 3):

- пошук фільмів та серіалів;
- просту систему навігації;
- ігри, а саме «Пошук фільму» і «Правда чи брехня»;
- афішу того, що зараз йде в кіно або ж скоро буде;
- пошук кінотеатрів у вказаному радіусі по локації;
- відображає список джерел, на яких можна подивитися знайдений фільм.

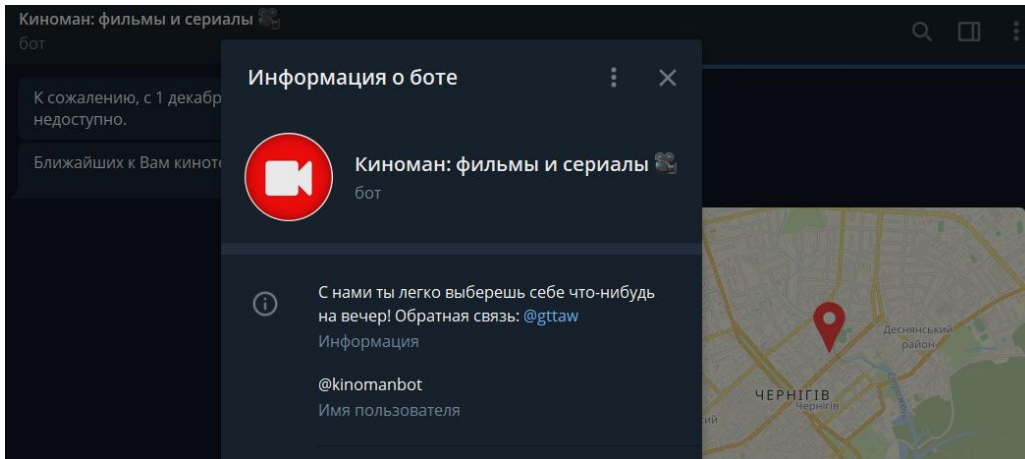


Рисунок 3 – Головна сторінка бота Кіноман

Розглянуті боти, безумовно, можуть бути конкурентами, проте всі вони мають недоліки, які можна було б виправити та за підтримкою англійської мови зібрати усе корисне і в одному місці.

Якщо ж брати до уваги рішення, не пов'язані із платформою Telegram, то якісних сервісів вже стає набагато більше, наприклад: Netflix, Kinopoisk та ін.

Для реалізації проєкту було вирішено зупинити свій вибір на середовищі розробки WebStorm, так як воно досить поширене, має безкоштовну версію для використання в освітніх цілях за підпискою і до того ж має величезну кількість плагінів, розгорнуту функціональність та підтримує багато мов програмування.

Як мова програмування було вибрано TypeScript на платформі NodeJS, так як вона є сумісною з JavaScript і компілюється в останній. Фактично після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою [5, 6]. TypeScript відрізняється від JavaScript можливістю явного статичного визначення типів, підтримкою використання повноцінних класів (як у традиційних об'єктно-орієнтованих мовах), а також підтримкою підключення модулів, що покликане підвищити швидкість розробки, полегшити читабельність, рефакторинг і повторне використання коду, допомогти здійснювати пошук помилок на етапі розробки і компіляції і, можливо, прискорити виконання програм.

Для реалізації клієнтської частини необхідно створити та налаштувати бота в Telegram. Ця можливість реалізована в середині самого месенджера.

Для прискорення розробки та спрощення взаємодії з Telegram будемо використовувати сучасний фреймворк для NodeJS – Telegraph.

Базою даних буде виступати Cloud Firestore – це гнучка, що масштабується, хмарна NoSQL-база даних від Firebase і Google CloudPlatform для інтернету, мобільних платформ і серверних додатків. База даних Cloud Firestore підтримує гнучкі, ієрархічні структури даних, зберігає дані в документах, які, у свою чергу, зберігаються в колекціях. Документи можуть мати вкладення, об'єкти і підколекції. Даний варіант бази даних практично ідеально підходить для використання у запланованому проєкті. У свою чергу, вона є одним із

сервісів, що надає Google CloudPlatform, тому через тісну інтеграцію необхідно використовувати хмарну платформу Google для хостингу бота після його розробки [7].

4. Методи взаємодії з ботом

Кожен бот є автономним незалежним додатком і може працювати у двох режимах.

1. Режим автономного додатка (getUpdates) – це тоді, коли запускається скрипт, який на регулярній основі опитує сервер Telegram. Цей метод використовується для отримання оновлень через так званий longpolling. Відповідь повертається, як масив об'єктів Update. Проте у даного способу оновлень є маса недоліків. Для цього необхідно запитувати дані кожні n секунд із сервера Telegram. Після чого, відповідно до внутрішніх алгоритмів, виконуються різні функції, що є неефективним і нераціональним. Тож цей спосіб буде використаний тільки на етапі розробки (рис. 4).

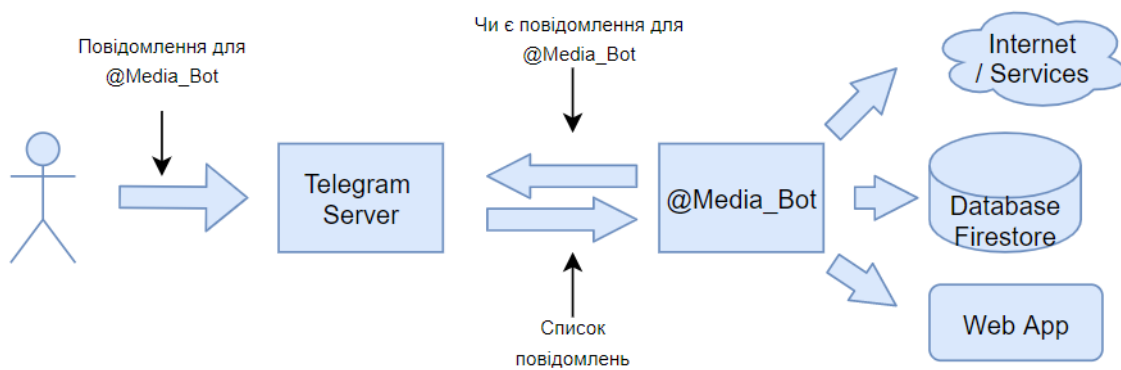


Рисунок 4 – Діаграма роботи Telegram-бота у режимі автономного додатка

2. Режим WebHook – це варіант, в якому дозволяється налаштувати Telegram-бот таким чином, що всі запити пересилаються на певний веб-адрес. Кожен раз при отриманні оновлення на цю адресу буде відправлений HTTPS-POST із серіалізованим в JSON об'єктом Update. В основному використовується об'єкт Message, який отримується з Update. Даний режим також має певні обмеження: потрібен виділений веб-сервер, який повинен приймати запити тільки за протоколом https, що, у свою чергу, вимагає валідний SSL-сертифікат (рис. 5).

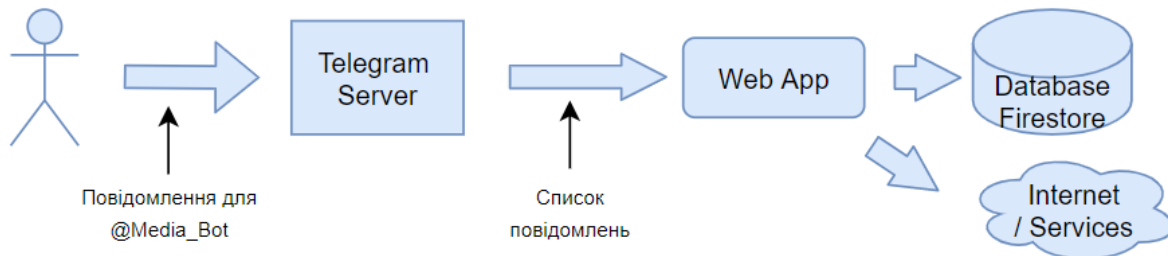


Рисунок 5 – Діаграма роботи Telegram-бота у режимі WebHook

5. Архітектура бота

Коли користувач відправляє повідомлення в зашифрованому вигляді на сервер Telegram, воно автоматично перенаправляється в Telegram-бот. Далі в боті відбувається обробка по-

повідомлення. Якщо воно відправлено для внесення налаштувань, то відбуваються запис нових даних і відправка повідомлення у відповідь користувачеві. Якщо ж це повідомлення направлено на пошук інформації, то бот відправляє відповідний запит на інформацію одному або ж декільком провайдерам, і після отримання та обробки її відсилає відповідь користувачу (рис. 6).

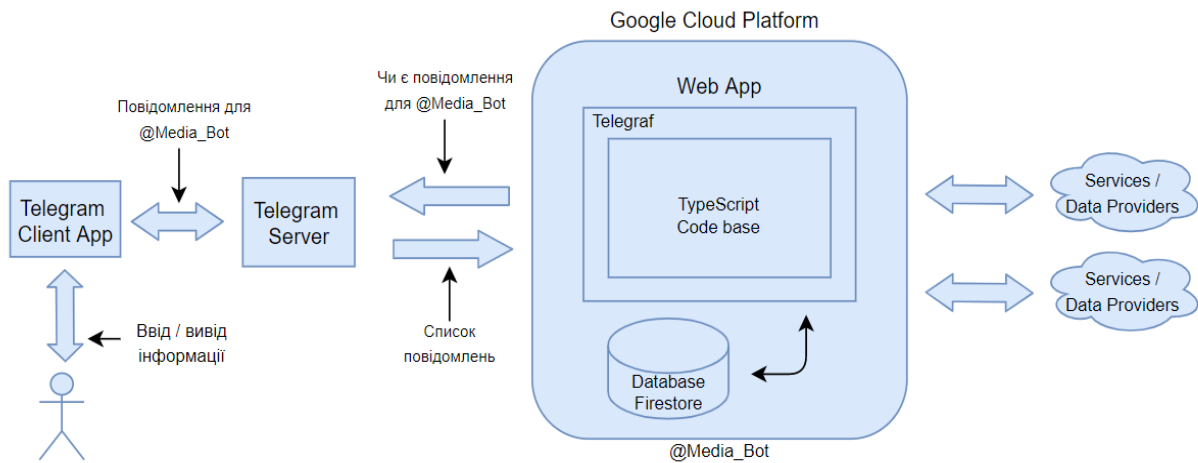


Рисунок 6 – Архітектура бота

Як основна NoSQL-база даних була обрана розробка Google. Це розташована база даних у хмарі DataBase Firestore.

Підключення і взаємодія відбуваються через доступні SDK для NodeJS, а також REST API. Так як усі клієнти працюють із базою даних напряму, то необхідно додати правила безпеки, які контролюватимуть права читання і запису документів і колекцій.

Модель даних Cloud Firestore підтримує гнучкі ієрархічні структури даних та зберігає дані в документах, які, у свою чергу, зберігаються в колекціях. Також документ може містити вкладені колекції. Крім того, модель даних Cloud Firestore надає потужну інфраструктуру Google CloudPlatform, яка забезпечує автоматичну мультирегіональну реплікацію даних, надійні гарантії цілісності, атомарні batch-операції, підтримку транзакцій [8, 9].

6. Архітектура програмного додатка

Згідно з зазначеним функціоналом, для розробки Telegram-бота знадобиться лише два рівні, а саме:

- рівень-сцен – це верхній рівень веб-додатка. Він відповідає за обробку запитів користувача від клієнта Telegram та повернення коректної відповіді назад. Даний рівень має також обробляти виняткові ситуації, які виникли на інших рівнях.
- рівень-сервісів – це рівень, що знаходиться нижче за перший. Він представляє собою допоміжний проміжний рівень між базою даних та обробленою інформацією з першого рівня.

Початковою точкою для взаємодії з ботом виступає файл `index.ts`. У ньому виконуються реєстрація сцен бота та реєстрація бота для Telegram через WebHook. Під сценою розуміється ізольована область, що має заздалегідь визначений список команд.

У створеному додатку були впроваджені такі сцени:

- `startScene` – перша сцена для взаємодії з користувачем, тому що для подальшої роботи користувач повинен бути зареєстрованим;
- `extraScene` – допоміжна сцена, в якій користувач може знайти більше інформації, а також залишити відгук;

- `adminScene` – сцена, в яку може потрапити тільки адміністратор, оскільки саме звідти можна буде отримати статистику використання та відправляти повідомлення кінцевим користувачам;

- `settingsScene` – допоміжна сцена, в якій можна змінити мову й подивитися власну інформацію;

- `contactScene` – сцена, яка необхідна для збору відгуків для поліпшення роботи бота;

- `searchScene` – одна з головних сцен, оскільки саме тут користувач буде шукати цікавий для нього контент, такий як фільми, серіали, аніме тощо;

- `personScene` – сцена, що відображає кінцеву інформацію про обрану персону з додатковими можливостями;

- `mediaScene` – сцена, що відображає кінцеву інформацію про обрану медіа з додатковими можливостями.

Основних сервісів було виділено лише два, а саме:

- `DatabaseService` – сервіс, який використовується для зв'язку із хмарною базою даних, тобто Cloud Firestore.

- `DataproviderService` – сервіс, що використовує медіаконтент, який розроблений провайдером, а також сервіси для отримання актуальних даних про певну людину в залежності від запиту.

На етапі проєктування було визначено, що необхідно буде реєструвати користувачів, проте сам медіаконтент було прийнято не зберігати у власній базі даних через такі причини:

- величезний обсяг даних, які треба зберігати, що заняло б багато часу;
- необхідність підтримки даних в актуальному стані, що означало б необхідність збирати на регулярній основі величезну кількість інформації, локалізувати її, а також проводити нормалізацію.

Незважаючи на це, для майбутньої масштабності проєкту були реалізовані класи для основних сутностей, в які увійшли:

- `Person` – базовий клас для конкретної персони;
- `User` – клас для користувача бота;
- `Media` – базовий клас для медіаконтенту;
- `Movie` – клас для фільмів;
- `Series` – клас для серіалів;
- `Season` – клас для окремого сезону серіалу;
- `Video` – клас для відеоматеріалу, зв'язаний з медіаконтентом;
- `Keyword` – клас для ключового слова, яке може мати будь-яка сутність;
- `Genre` – клас для певного жанру;
- `Character` – клас для персонажу;
- `Crew` – клас для персоналу у медіа;
- `Credit` – клас для ролі персони у медіа;
- `Certificate` – клас для рейтингу медіаконтенту;
- `Company` – клас для компанії;
- `ExternalLink` – клас для зовнішніх посилань, що може мати будь-яка сутність;
- `Money` – клас для грошей;
- `WatchProvider` – клас для джерела на перегляд медіа;
- `PaginationModel` – модель, що може формувати списки по сторінках з переданого контенту.

7. Головні елементи Telegram-бота

У першому повідомленні Telegram-бот містить у собі вітання і в неформальній манері пояснює, як користуватися ботом. Головне меню Telegram-бота зображене на рис. 7.

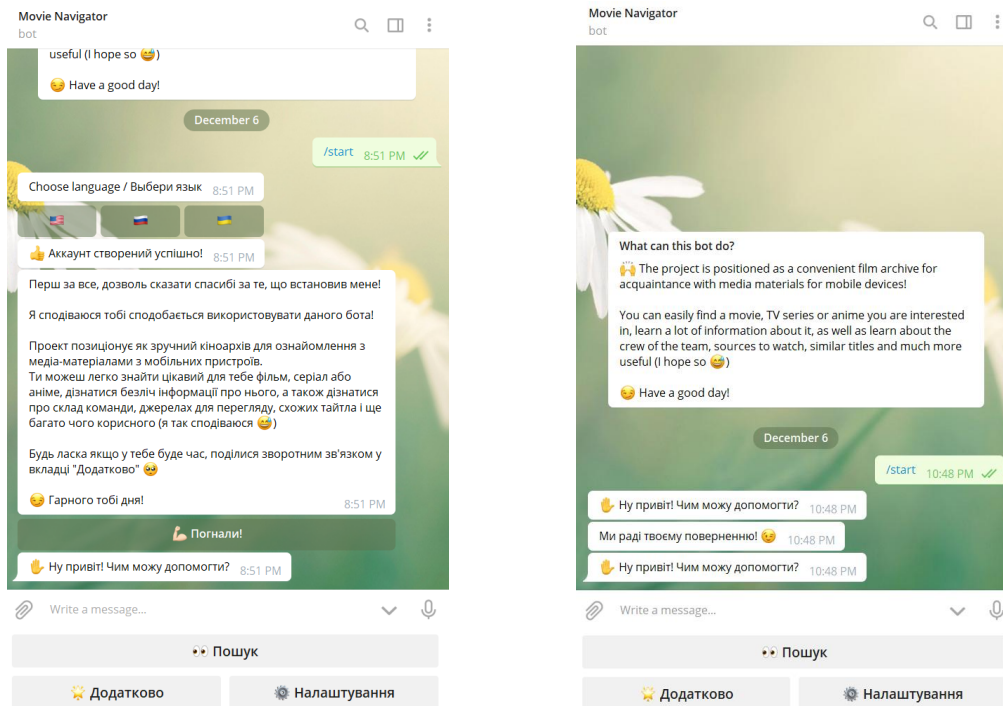


Рисунок 7 – Головне меню Telegram-бота

Сцена «Медіа» відображає кінцеву інформацію про обрану медіа з додатковими можливостями (рис. 8).

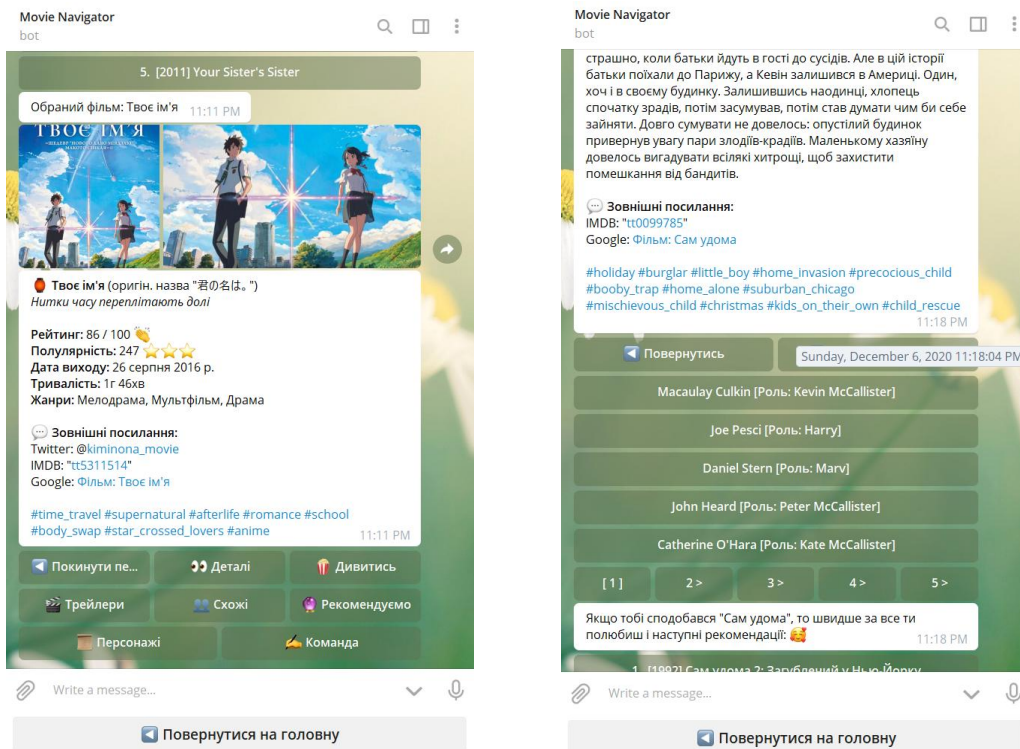


Рисунок 8 – Сцена «Медіа»

Сцена «Персона» відображає кінцеву інформацію про обрану персону з додатковими можливостями (рис. 9).

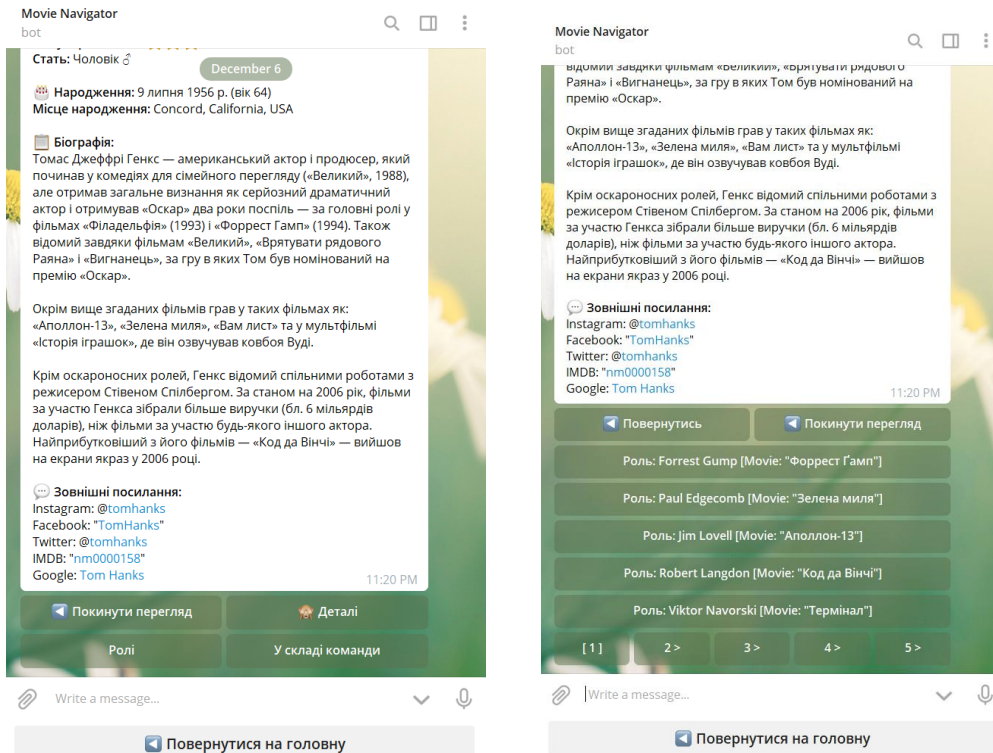


Рисунок 9 – Сцена «Персона»

8. Висновки

Месенджери є соціальними мережами, що найбільш швидко розвиваються та охоплюють мільярди користувачів по всьому світу. Вони намагаються включити в себе якомога більше сервісів для комфорту кінцевого споживача. У свою чергу, користувач прагне отримати якомога більше функціональних елементів за менших витрат. Чат-бот був реалізований за допомогою скриптової мови TypeScript на платформі NodeJS, а також сервісів Google. Бот надає технічну підтримку, допомагає у виборі продукту, консультує з питань, оформляє замовлення та розважає користувачів.

Крім того, створений чат-бот є абсолютно безкоштовним, пропонує базовий функціонал платних платформ, має варіанти взаємодії на трьох мовах і буде у відкритому доступі для всіх верств населення, у яких є смартфон, підключення до мережі Інтернет, а також месенджер Telegram. Також Telegram-бот буде працювати на будь-яких платформах.

Для тестування розробленого бота були залучені звичайні користувачі, що цікавляться фільмами та серіалами. Зібраний зворотний зв'язок дозволив внести поправки, такі як:

1. Можливість відображення не повної інформації, а лише її частини при пошуку медіаконтенту.
2. Була реалізована модель для пагінації при відображенні ролей у фільмах та серіалах, тому що користувачам була незручна навігація через велику кількість персонажів.

СПИСОК ДЖЕРЕЛ

1. Онлайн консультант для Slack, HipChat и Telegram-LinkedChat. URL: <https://linked.chat/>.
2. Документація Telegram: довідник по Bot API. URL: <https://tigrm.ru/docs/bots/api>.
3. Документація Telegram: приклади ботів. URL: <https://tigrm.ru/docs/bots/samples>.

4. ТОП месенджери в Україні та у світі 2020. URL: <https://marketer.ua/top-messengers-in-ukraine-and-the-world/>.
5. Java|Oracle. URL: <https://www.java.com/ru/>.
6. Телеграм-бот за допомогою NodeJs і Firebase Cloud Functions. URL: <https://www.udemy.com/course/nodejs-firebase-cloud-functions-ru/>.
7. GCP App Deployment, Debugging and Performance. URL: <https://www.coursera.org/learn/app-deployment-debugging-performance>.
8. Google Cloud Platform Fundamentals: Core Infrastructure. URL: <https://www.coursera.org/learn/gcp-fundamentals>.
9. NoSQL systems. URL: <https://www.coursera.org/learn/nosql-databases>.

Стаття надійшла до редакції 08.02.2021