

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра інженерія програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр
на тему: «Розробка Telegram чат-боту для пошуку цікавого відео-контенту
мовою Python»

Виконав: студент 4 курсу, групи ПД-43
спеціальності

121-інженерія програмного забезпечення

(шифр і назва спеціальності)

Данилюк.О.С

(прізвище та ініціали)

Керівник Золотухіна.О.А

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інженерія Програмного Забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки - 121 - Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедру

Інженерія програмного пабезпечення

О.В.Негоденко

_____ 2023 року

Завдання

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Данилюк Олег Сергійович

(прізвище, ім'я, по батькові)

Тема роботи: «Розробка Telegram чат-боту для пошуку цікавого відео-контенту мовою Python»

Керівник роботи Золотухіна Оксана Анатоліївна, к.т.н., доц.

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

затверджені наказом вищого навчального закладу від —24 лютого 2023 року №26.

1. Строк подання студентом роботи 1 червня 2023 року
2. Вхідні дані до роботи:
 - 3.1 Науково-технічна література, пов'язана з розробкою чат ботів .
 - 3.2 Технічна документація.
4. Зміст розрахунково-пояснювальної записки (перелік питань які потрібно розробити).
 - 4.1 Огляд предметної галузі пошуку цікавого відео-контенту.
 - 4.2 Огляд та аналіз наявних аналогів Telegram чат-ботів.
 - 4.3 Визначення вимог до структури та алгоритму роботи чат-боту, формування технічного завдання.
 - 4.4 Розробка Telegram чат-боту для пошуку цікавого відео-контенту.
 - 4.5 Тестування розробленого чат – бота

5. Перелік демонстраційного матеріалу
 - 5.1 Мета, об'єкт та предмет дослідження
 - 5.2 Задачі дипломної роботи
 - 5.3 Таблиця порівняння аналогів
 - 5.4 Вимоги до додатку
 - 5.5 Програмні засоби реалізації
 - 5.6 Діаграма варіантів використання
 - 5.7 Структура чат-боту
 - 5.8 Діаграма послідовностей
 - 5.9 Алгоритм підбору контенту
 - 5.10 Діаграма пакетів
 - 5.12 Екрані форми розробленого телеграм чат боту
 - 5.13 Апробація результатів дослідження
 - 5.14 Висновки

Дата видачі завдання 25 лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.23-27.02.23	Виконано
2	Огляд предметної галузі пошуку цікавого відео-контенту.	28.02.23-10.03.23	Виконано
3	Огляд та аналіз наявних аналогів Telegram чат-ботів.	13.03.23-24.03.23	Виконано
4	Визначення вимог до структури та алгоритму роботи чат-боту, формування технічного завдання.	27.03.23-28.04.23	Виконано
5	Тестування розробленого чат – бота	27.03.23-28.04.23	Виконано
6	Попередній захист роботи	24.05.23	Виконано
7	Подання роботи в деканат	01.06.2023	Виконано

Студент _____ Данилюк О.С
(підпис) (прізвище та ініціали)

Керівник роботи _____ Золотухіна О.А
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 60 с., 29 рис., 1 табл., 13 джерел.

TELEGRAM ЧАТ-БОТ, PYTHON, ВІДЕО-КОНТЕНТ, ПОШУК ЗА ЖАНРОМ.

Об'єкт дослідження - процес автоматизації процесу пошуку цікавого відео контенту.

Предмет дослідження - програмне забезпечення для допомоги автоматизованого пошуку інформації про цікаве відео контент.

Мета роботи - спрощення процесу пошуку фільмів , мультфільмів та інших відео за рахунок Telegram чат боту на мові Python.

Визначено, що ключовими недоліками існуючих ботів для вирішення подібних задач є те, що вони не мають категорії жанрів про мультфільми, рейтингу фільмів , а також не мають відгуків від професійних критиків щодо фільму

На основі результатів виконаних досліджень розроблено Telegram чат-бот, який дозволяє вибирати різні жанри фільмів, додаткові жанри мультфільмів та мультимедіа, виконувати пошук за рейтингом фільмів, надавати інформацію з відгуками про фільм від критиків. Для розробки використовувалась мова програмування Python та Telegram API.

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ПОШУКУ ЦІКАВОГО ВІДЕО- КОНТЕНТУ	11
1.1 Аналіз концепції та видів чат-ботів	11
1.2 Типи чат-ботів	13
1.3 Характеристики месенджерів	17
1.4 Аналіз аналогів та визначення вимог до додатку	18
1.4.1 The Movie Bot	18
1.4.2 FilmAffinity.bot	19
1.4.3 acutebot	19
2 ОГЛЯД ЗАСОБІВ РОЗРОБКИ ЧАТ-БОТУ	23
2.1 Аналіз засобів програмної реалізації	23
2.1.1 BotFather	23
2.1.2 API Telegram Bot.....	28
2.2 Вибір засобів програмування	28
2.3 Вибір середовища розробки	30
3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ...	31
3.1 Моделювання вимог до системи	31
3.2 Розробка алгоритму підбору контенту	32
3.3 Моделювання процесу обробки запиту до бота	33
3.4 Розробка архітектури бази даних	34
3.4 Розробка архітектури системи	34
3.5 Розробка логіки функціонування чат-боту	35
3.6 Опис функціонування чат-боту	37
3.7 Тестування чат-боту	41
ВИСНОВКИ	43
ПЕРЕЛІК ПОСИЛАНЬ	44

ДОДАТОК А ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ).....	46
ДОДАТОК Б ЛІСТИНГИ ПРОГРАМНОГО КОДУ	54

ВСТУП

У сучасному світі мобільні пристрої перетворилися на величезну інфраструктуру, і існує величезний попит на мобільні програми. Особливою популярністю користуються соціальні мережі та месенджери. Одним з найпопулярніших месенджерів цього типу є Telegram. Зростання популярності чат-ботів як ефективного інструменту спілкування в повсякденному житті пояснюється тим фактом, що чат-боти зараз широко використовуються в багатьох регіонах для створення розмов з користувачами Інтернету. Більшість сучасних підприємств, організацій та інтернет-магазинів додають у свою діяльність Telegram-ботів для вирішення конкретних завдань. Чат-боти стали особливо популярними, коли їх інтегрували в месенджери та соціальні мережі, особливо Telegram і Facebook, і з'явилися цілі вітрини та магазини чат-ботів.

Чат-боти можна використовувати як для розваги, так і для іншої мети. Також є функціональні чат-боти для спілкування, який може сповістити про погоду на цей день, про актуальний курс валют або про іншу цікаву інформацію.

Об'єкт дослідження - процес автоматизації процесу пошуку цікавого відео контенту.

Предмет дослідження - програмне забезпечення для допомоги автоматизованого пошуку інформації про цікаве відео контент.

Мета роботи - спрощення процесу пошуку фільмів, мультфільмів та інших відео за рахунок Telegram чат боту на мові Python.

Виходячи з поставленої мети, були поставлені такі завдання:

- Огляд предметної галузі пошуку цікавого відео-контенту.
- Огляд та аналіз наявних аналогів Telegram чат-ботів.
- Визначення вимог до структури та алгоритму роботи чат-боту, формування технічного завдання.
- Розробка Telegram чат-боту для пошуку цікавого відео-контенту.
- Тестування розробленого чат боту.

Практичне значення - чат бот буде використовуватися для загального користування різних категорій населення. Налаштування спілкування та робота з чат-ботом допоможе зменшити кількість витраченого часу для пошуку інформації про фільми або інший відео контент.

Структура роботи. У вступі наведено актуальність розробки телеграм бота та застосування його в загальному житті. Перший розділ дипломної роботи присвячена проблемам проекту, які вирішуються за допомогою розробки чат-ботів. Друга частина присвячена засоби розробки чат-ботів. Розділ 3 дає детальний опис процесів проектування та розробки програмного забезпечення чат-боту.

Чат-бот, представлений в дані дипломній роботі, дозволить спростити щоденні рутинні дії для пошуку цікавого відео-контенту для людини. Головною перевагою серед інших методів отримання інформації є те, що вся необхідна інформація та посилання на відео контент знаходиться на платформі єдиного месенджера.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ПОШУКУ ЦІКАВОГО ВІДЕО-КОНТЕНТУ

1.1 Аналіз концепції та видів чат-ботів

Чат-боти – це інструменти для спілкування, які ефективно виконують рутинні завдання. Людям вони подобаються, тому що вони допомагають їм швидко виконувати ці завдання, щоб вони могли зосередити свою увагу на високорівневих, стратегічних і захоплюючих заходах, які вимагають людських можливостей, які не можуть бути відтворені машинами.

На найпростішому рівні чат-боти (рис.1.1) -це комп'ютерні програми ,які імітують і обробляють людське мовлення (письмове чи усне),дозволяючи людям взаємодіяти з цифровими пристроями так, ніби вони розмовляють з реальною людиною. Чат-боти можуть бути простими, як елементарні програми ,що відповідають найпростіші запити одним рядком ,або складними, як цифрові асистенти, які навчаються і розвиваються в міру обробки інформації, підвищуючи рівень своєї персоналізації.

Чат-боти часто використовуються для покращення досвіду управління ІТ-послугами, що спрямоване на самообслуговування та автоматизацію процесів, які пропонуються внутрішньому персоналу. За допомогою інтелектуального чат-бота такі типові завдання, як оновлення паролів, стан системи, сповіщення про збої та керування знаннями, можна легко автоматизувати та зробити доступними цілодобово та без вихідних, одночасно розширюючи доступ до широко використовуваних голосових і текстових розмовних інтерфейсів.

Що стосується бізнесу, чат-боти найчастіше використовуються в контактних центрах для керування вхідними повідомленнями та спрямування клієнтів до відповідного ресурсу. Вони також часто використовуються для внутрішніх цілей, таких як адаптація нових співробітників і допомога всім співробітникам у звичайних заходах, включаючи планування відпусток, навчання, замовлення

комп'ютерів і бізнес-товарів, а також інші дії самообслуговування, які не потребують втручання людини.

Приклад чат-боту наведено на рис.1.1.



Рисунок 1.1 – Як виглядає чат бот

З боку споживачів чат-боти надають різноманітні послуги клієнтам, починаючи від замовлення квитків на заходи до бронювання та поселення в готелях до порівняння продуктів і послуг. Чат-боти також широко використовуються для виконання рутинних операцій з клієнтами в банківській сфері, роздрібній торгівлі, а також секторах харчування та напоїв. Крім того, багато функцій державного сектора доступні чат-ботами, наприклад, подання запитів на міські послуги, обробка запитів, пов'язаних з комунальними послугами, і вирішення проблем з рахунками.

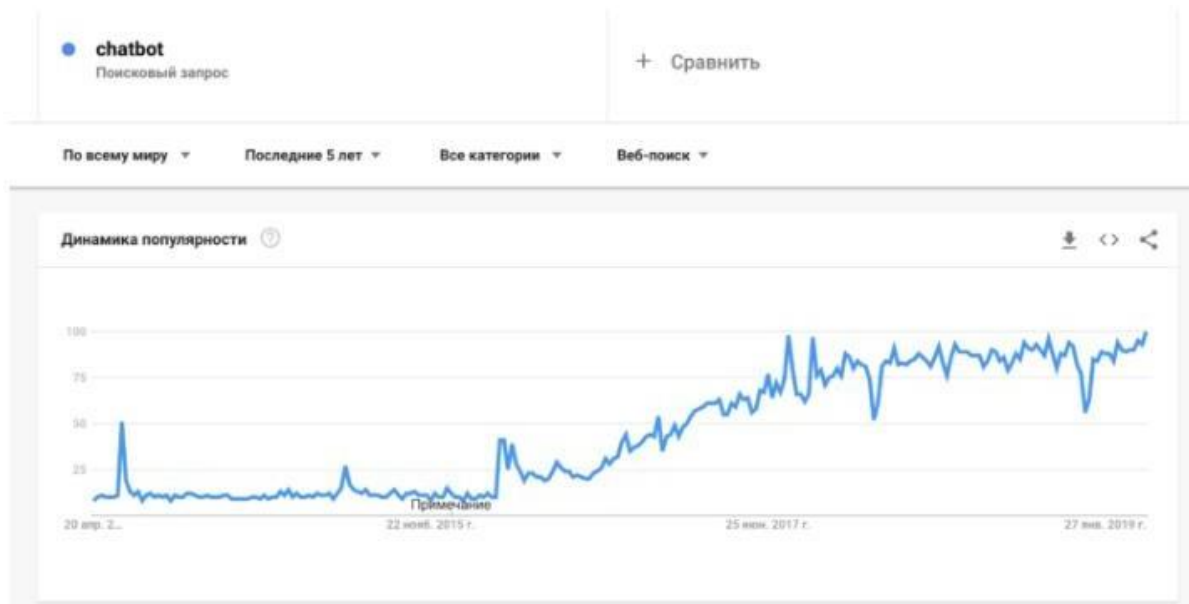


Рисунок 1.2 – Динаміка популярності чат-ботів

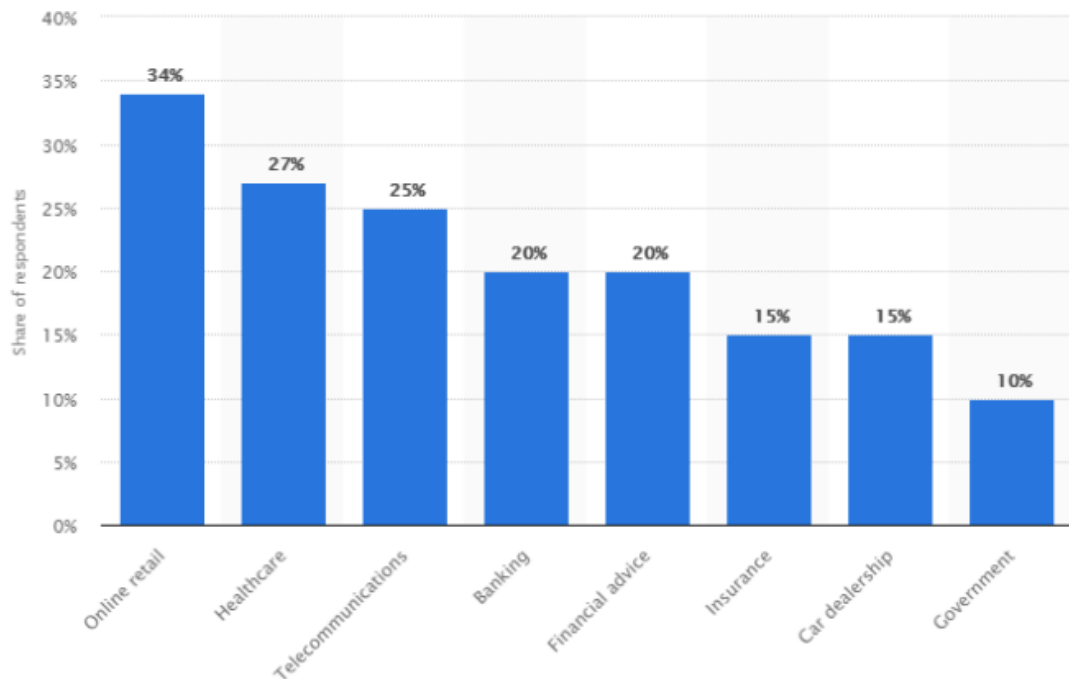


Рисунок 1.3 – Данні опитування Statista

1.2 Типи чат-ботів

У цьому розділі описані різні типи чат-ботів, їхнє використання та те, яке програмне забезпечення для чат-ботів буде більш корисним для вирішення певних задач.

Задачно-орієнтовані (декларативні) чат-боти — це одноцільові програми, які зосереджені на виконанні однієї функції. Використовуючи правила, NLP і дуже мало ML, вони генерують автоматизовані, але розмовні відповіді на запити користувачів. Взаємодія з цими чат-ботами є дуже специфічною та структурованою та найкраще підходить для функцій підтримки та обслуговування. Чат-боти, орієнтовані на завдання, можуть відповідати на поширені запитання, як-от запити про робочі години або прості транзакції, які не включають низку змінних. Хоча вони використовують NLP, щоб кінцеві користувачі могли випробувати їх у розмові, їхні можливості досить базові. Зараз це найпоширеніші чат-боти.

Керовані даними та прогнознi (розмовні) чат-боти часто називають віртуальними помічниками або цифровими помічниками, і вони набагато складніші, інтерактивні та персоналізовані, ніж чат-боти, орієнтовані на завдання (рис.1.2).

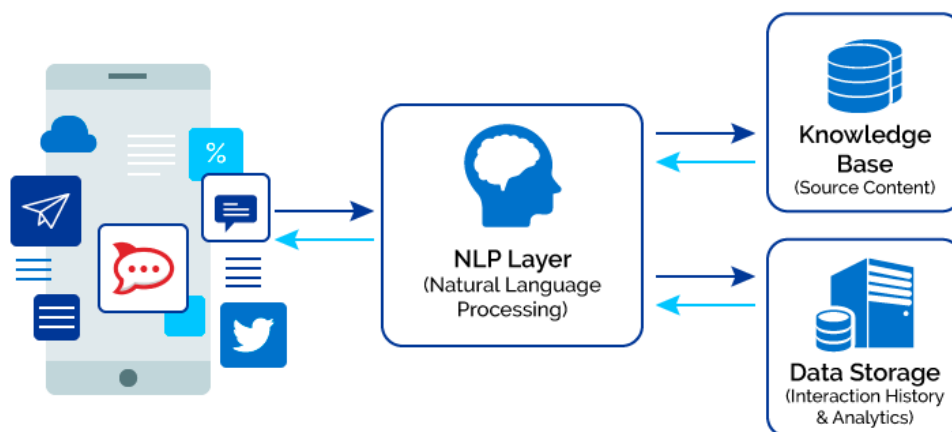


Рисунок 1.2 – Схема роботи розмовного чат-боту

Ці чат-боти розпізнають контекст і використовують розуміння природної мови (NLU), NLP і ML, щоб навчатися по ходу. Вони застосовують інтелектуальні дані та аналітику, щоб забезпечити персоналізацію на основі профілів користувачів і минулої поведінки користувачів. Цифрові помічники можуть з часом дізнаватися про вподобання користувача, надавати рекомендації та навіть передбачати потреби. Окрім моніторингу даних і намірів, вони можуть ініціювати розмови. Siri

від Apple і Alexa від Amazon є прикладами орієнтованих на споживача, керованих даними, прогнозуючих чат-ботів.

Голосовий бот — це розмовне рішення, яке використовує штучний інтелект (AI) і розуміння природної мови (NLU), щоб допомогти інтерпретувати наміри та значення мовних команд. Ця технологія, також відома як розмовна інтерактивна голосова відповідь (IVR), дозволяє користувачам взаємодіяти з пристроєм просто промовляючи. Оскільки голос є однією з найшвидших форм людського спілкування, голосові боти пропонують вторинний рівень обслуговування клієнтів, представляючи інший спосіб відображення вашого бренду.

Гібридний чат-бот — це програма, яка спілкується з людьми, щоб надати автоматизовану та персоналізовану відповідь. Він втручається лише через канал обміну миттєвими повідомленнями на додаток до агентів (чат, обмін повідомленнями або соціальні мережі).

Чат-боти соціальних медіа – це інструменти на основі штучного інтелекту, які забезпечують автоматизовану підтримку клієнтів у кількох соціальних мережах. Ці системи можуть бути розгорнуті на соціальних платформах (наприклад, Facebook, Instagram, Twitter) для обробки великих обсягів поширених запитань і основних проблем без втручання.

Чат-боти на основі меню/кнопок — це найпростіший тип чат-ботів, які легко створити та використовувати. У більшості випадків ці чат-боти мають уже готову базу знань із питаннями, представленими користувачеві у вигляді кнопок. Кнопковий бот спілкується відповідно до заздалегідь визначених правил.

Чат-бот на основі розпізнавання ключових слів використовує обробку природної мови (NLP) для обслуговування своїх користувачів. Тут користувачі можуть взаємодіяти, вводячи вільний текст. Потім чат-бот аналізує текст користувача на основі ключових слів, оскільки вони залежать від ключових слів, і дає найкращу відповідь.

Чат-боти на основі правил структуровані як дерево діалогів і часто використовують регулярні вирази, щоб зіставити введені користувачем відповіді,

подібні до людських. Мета полягає в тому, щоб імітувати розмову в реальному житті, часто в певному контексті, як-от розповісти користувачеві, яка надворі погода.

Чат-боти, які можуть зрозуміти контекст розмови та вивести правильне значення запиту користувача, називаються контекстними чат-ботами. Контекстні чат-боти також можуть запам'ятовувати попередні взаємодії (якщо такі були) і використовувати їх, щоб залишатися актуальними під час взаємодії з повторними відвідувачами.

Чат-боти служби підтримки клієнтів дозволяють клієнту прийняти запитання «як є» і помістити його в повідомлення, а потім миттєво отримати відповідь. Поєднання штучного інтелекту та методів розуміння природної мови на основі правил може спробувати інтерпретувати запитання та автоматично відповісти на нього.

Транзакційні боти створені для роботи від імені людини. Їм потрібно взаємодіяти з системами для виконання певних завдань, часто пов'язаних із переміщенням даних. Оскільки транзакційні боти створені для конкретного завдання, вони можуть бути розроблені для різних індивідуальних рішень.

Завдяки функціональним чат-ботам стає можливим здійснювати певні дії, такі як купівля товарів чи послуг, проведення оплати тощо:

- допомогти користувачеві зробити замовлення;
- прийняти заявку;
- зібрати зворотний зв'язок від користувачів;
- дати консультацію;
- зібрати контактні дані;
- реалізувати різні товари та послуги;
- допомогти користувачеві записатися в салон краси/ прийом до лікаря та ін;
- завантажити музику, фільм чи книгу.

API чат-бота — це інтерфейс програмування додатків, який дозволяє підключати канали обміну повідомленнями (SMS, веб-чат, WhatsApp і месенджери соціальних мереж) із програмним забезпеченням і функціями чат-бота. Це дає змогу вашому індивідуальному чат-боту взаємодіяти з клієнтами на різних веб-сайтах і платформах.

1.3 Характеристики месенджерів

Месенджер – це програма для обміну повідомленнями. Обмін миттєвими повідомленнями – це тип онлайн-чату, який забезпечує передачу текстових повідомлень у реальному часі через Інтернет. З появою смартфонів і подальшого вибуху мобільних додатків недорогі або безкоштовні програми для чату та соціальних повідомлень стали недорогою альтернативою SMS-повідомленням через оператора. Багато програм для обміну повідомленнями пропонують такі функції, як групові чати, обмін зображеннями, відео та навіть аудіоповідомленнями разом із наклейками та емодзі.

Telegram займає третє місце в загальному рейтингу месенджерів, він входить до п'ятірки найкращих програм у 21 із 100 країн і в десятку найкращих у 53 із 100 країн.

Telegram	UK	US	CA	AU
Install Penetration	19.05%	18.87%	17.44%	14.86%
Downloads	5.82M	35.98M	2.23M	1.35M
Unique Installs	845.44K	3.97M	345.01K	228.36K
Daily Active Users	1.95M	5.14M	992.71K	365.15K
Monthly Active Users	4.41M	10.89M	1.97M	842.44K

Рисунок 1.3 – Загальна характеристика популярності Telegram

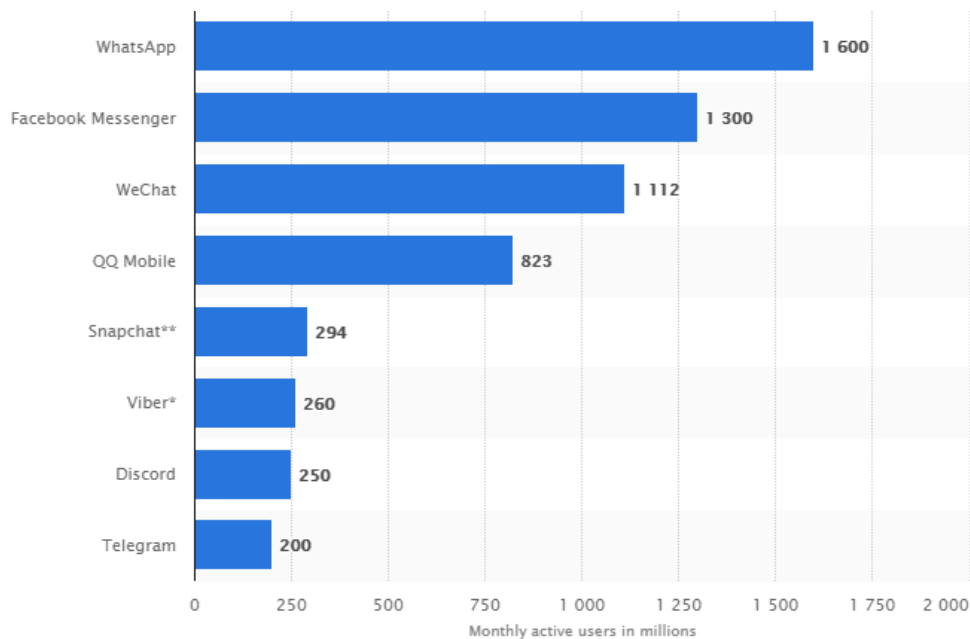


Рисунок 1.4 – Кількість користувачів в месенджерах за місяць

1.4 Аналіз аналогів та визначення вимог до додатку

1.4.1 The Movie Bot

Цей бот може допомогти отримати інформацію про ваші улюблені фільми , телешоу , серіали . Бот має налаштування та опис фільмів та багатьох мовах світу.

Інтерфейс додатку наведено на рис. 1.5.

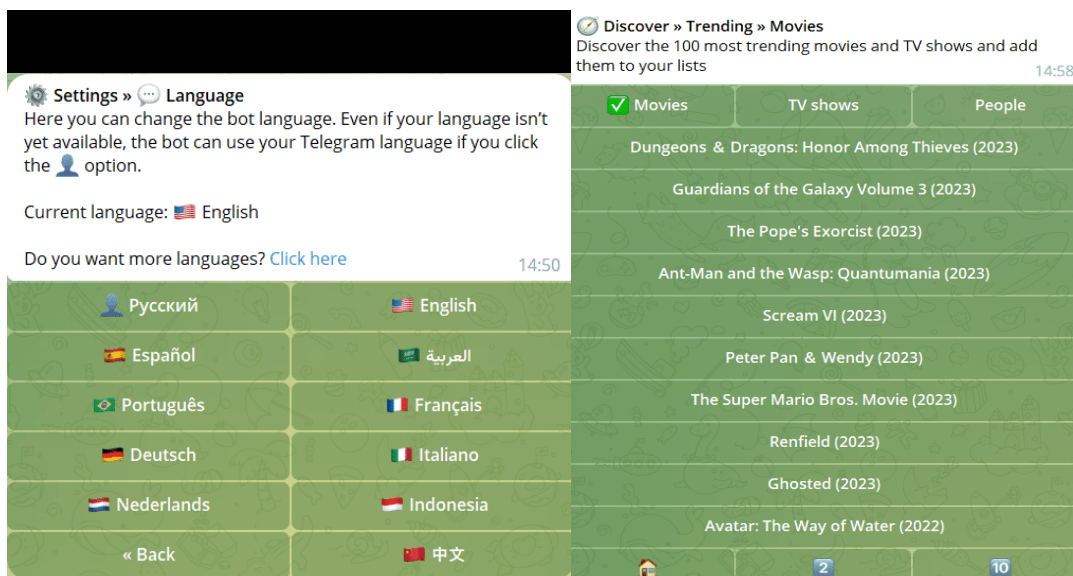


Рисунок 1.5 – Функціонал налаштування додатку The Movie Bot. Форма показу рекомендованих фільмів

1.4.2 FilmAffinity.bot

Цей бот може допомогти отримати інформацію про ваші улюблені фільми , телешоу та багато інших пов'язаних речей .Також цей бот може допомогти знайти фільм з улюбленим актором користувача. Інтерфейс додатку наведено на рис. 1.6.

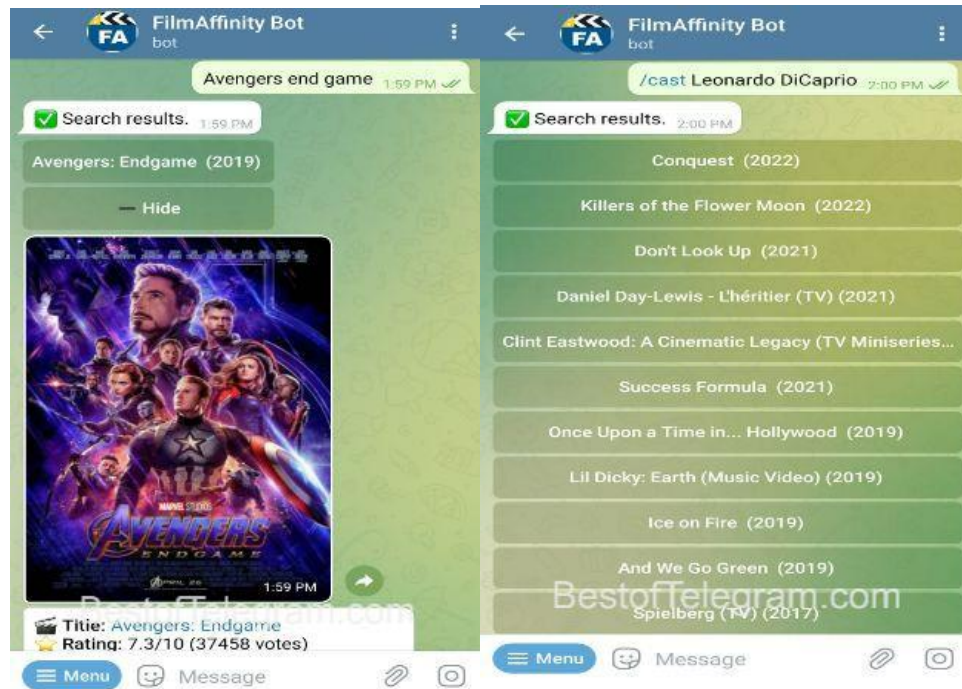


Рисунок 1.6 – Функціонал пошуку фільмів по актору . Форма показу рекомендованих фільмів

1.4.3 acutевот

Цей бот може допомогти отримати інформацію про ваші улюблені аніме, мангу, фільми, телешоу та багато інших пов'язаних речей.

Команди чат-бота:

/start: Щоб запустити бота.

/movies : пошук інформації про ваші улюблені фільми.

/tvshows: Отримайте інформацію про свої улюблені телешоу.

/recommend: дає вам рекомендації щодо фільмів і телешоу на основі ваших уподобань!

trendingmv : показує популярні фільми сьогодні або цього тижня.

/trendingtv : Те саме, що й вище, але для телешоу!

/collection : Отримати всі фільми в колекції, наприклад: Месники.

/lyrics: пошук текстів улюбленої музики.

Інтерфейс додатку acutebot наведено на рисунку 1.7.



Рисунок 1.7 – Функціонал пошуку фільмів. Форма показу рекомендованих фільмів acutebot

В таблиці 1.1 представлено зведені результати аналізу характеристик додатків для пошуку фільмів.

Таблиця 1.1 – Зведені результати аналізу характеристик додатків для пошуку фільмів

Показник	The Movie Bot	FilmAffinity.bot
Локалізація українською мовою	-	-

Наявність пошуку по категорії мультфільм	-	-
Відображення рейтингу фільмів	-	-
Наявність пошуку по жанрам фільмів	+	+
Наявність пошуку по акторами фільмів	+	-
Загальна інформація про фільм (короткий сюжет, головні герої)	+	+
Посилання на фільм	+	+

1.5 Аналіз вимог до додатку

Оскільки додаток для загального користування вимоги до нього залежать від попиту цільової аудиторії.

Цільова аудиторія – це коло потенційних клієнтів, яким потрібен певний товар чи послуга задоволення певних потреб. Такі споживачі мають деякі спільні особливості. Цільова аудиторія для телеграм чат-боту для пошуку цікавого відео контенту визначається за таких критеріїв

- стать;
- вік;
- товарні вподобання (жанри);
- цінності.

Основні вимоги до додатку, які потрібно реалізувати

- можливість пошуку та читання інформації на українській мові;
- посилання на фільмів , мультфільми та мультимедіа
- можливість вибирати різні жанри фільмів мультфільми та мультимедіа,

жанри фільмів є :

- комедійні фільми;
 - фентезі;
 - жахи;
 - бойовик;
 - детектив;
 - мультфільми.
- додати до додатку жанр мультфільми та мультимедіа;
 - додати рейтинг фільмів , мультфільми та мультимедіа ;
 - додати інформацію з відгуками про фільм від інших користувачів.

2 ОГЛЯД ЗАСОБІВ РОЗРОБКИ ЧАТ-БОТУ

Для розробки програми необхідно спершу визначити певні засоби та інструменти які використовуватимуться. До таких можна віднести:

- мова програмування;
- середовище розробки;
- використані інструменти розробки.

У цьому розділі буде проаналізовано можливості вибору таких засобів та наведено приклад розробки бота за їх допомогою.

2.1 Аналіз засобів програмної реалізації

2.1.1 BotFather

BotFather – найпростіший спосіб для реєстрації, налаштування та керування іншими telegram-ботами. Робота з ним проста і не потребує специфічних навичок. За допомогою BotFather можна зареєструвати необмежену кількість нових роботів. Єдиною умовою для реєстрації нового бота є його унікальний username.

BotFather стане відмінним рішенням для тих, хто не розуміється на програмуванні і не хоче довіряти створення та управління своїм ботом стороннім людям або стороннім організаціям.

Ім'я користувача бота - це логін бота і є коротким ім'ям, яке буде використовуватися в згадках і посиланнях на t.me. Надалі ім'я користувача не можна змінити. Імена користувачів (логіни) складаються з 5–32 символів та нечутливі до регістру, і можуть включати лише латинські символи, числа та символи підкреслення. Ім'я користувача бота має закінчуватися на bot, наприклад tetris_bot чи TetrisBot (рис.2.2).

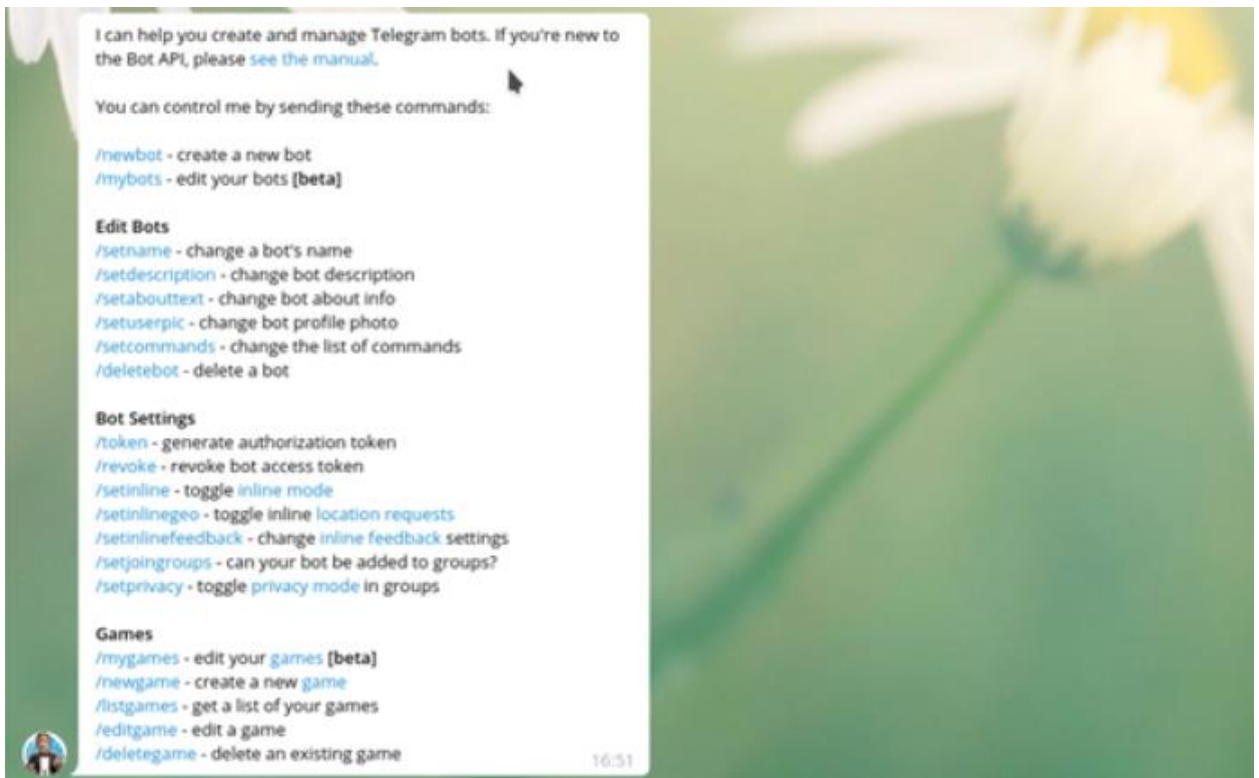


Рисунок 2.1 – Початок роботи із ботом BotFather та доступні команди

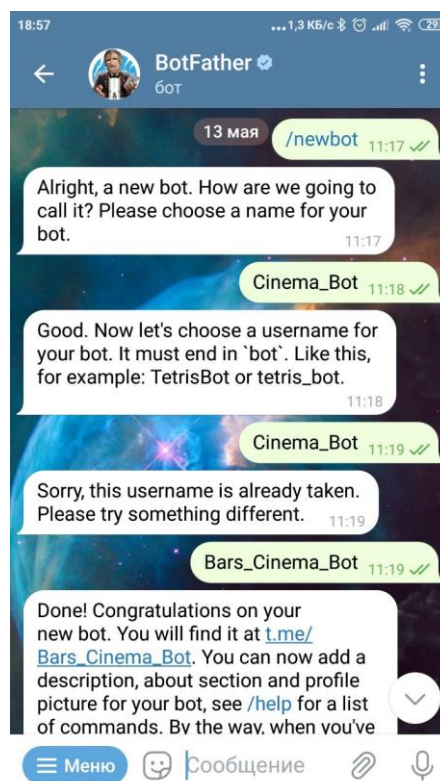


Рисунок 2.2 – Приклад Розробки ім'я для бота

Токен являє собою рядок типу 5991811368:AAFJATzL1o8XcZeiGwEDGPtGd2YeJKjVj6Q, який потрібний для авторизації бота та надсилання запитів до API бота (рис.2.3).

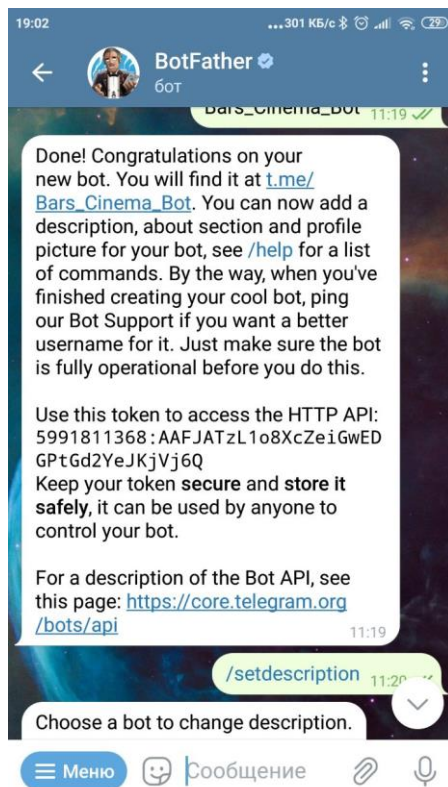


Рисунок 2.3 – Приклад токена для бота

Команди @BotFather:

/mybots - повертає список ботів із зручними елементами керування для редагування їх налаштувань;

/mygames - робить те саме для ігор.

Команди редагування Telegram бота:

/setname – змінити ім'я бота;

/setdescription - змінити опис бота, текст до 512 символів, що описує бота.

Користувачі побачать цей текст на початку розмови з ботом під назвою "What can this bot do?" (рис. 2.4).

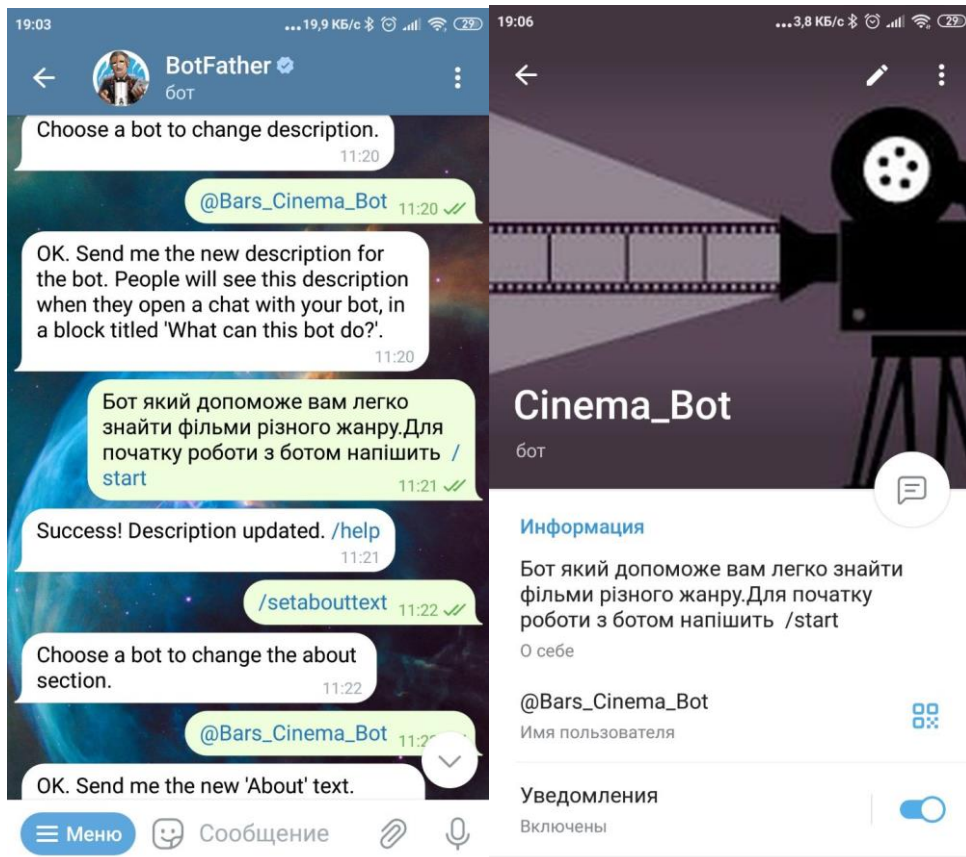


Рисунок 2.4 Змінити опис бота

`/setabouttext` - змінити інформацію про роботу, більш короткий текст до 120 символів. Користувачі побачать цей текст на сторінці профілю бота. Коли вони діляться ботом із кимось, цей текст відправляється разом із посиланням.

`/setuserpic` - Змінити фото профілю бота (рис. 2.5).

`/setcommands` - Змінити список команд, що підтримуються ботом. Користувачі будуть бачити список команд щоразу, коли вони набирають '/' у розмові з ботом. Кожна команда має ім'я (має починатися з косої межі '/', буквено-цифрове значення плюс підкреслення, не більше 32 символів, без урахування регістру), параметри та текстовий опис.

`/deletebot` - команда видаляє бота та звільняє його логін.

Команди зміни налаштувань Telegram бота:

`/setinline` - перемикає вбудований режим запитів для робота.

`/setinlinegeo` - запитує дані про місцезнаходження для надання вбудованих результатів на основі розташування.

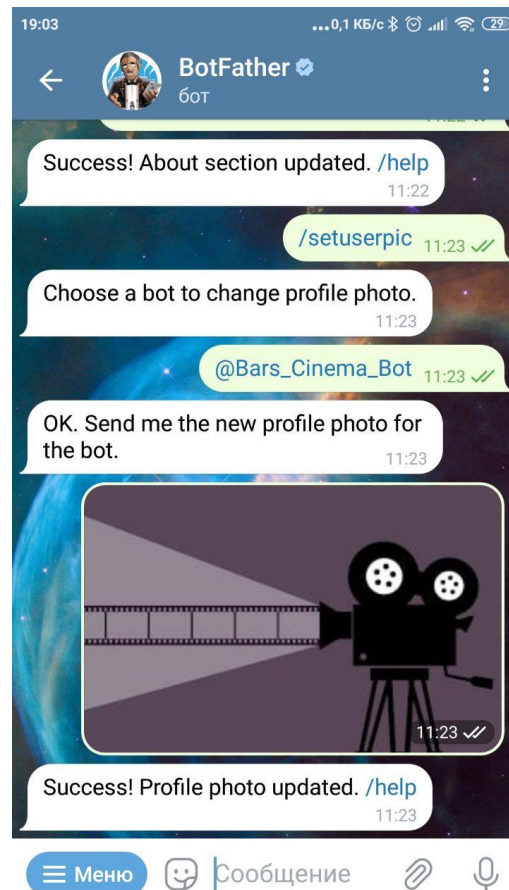


Рисунок 2.5 Змінити фото профілю бота

`/setjoingroups` - перемикач, який відповідає за те, чи можна додавати бота до групи чи ні. Будь-який бот повинен вміти обробляти особисті повідомлення, але якщо створений бот не призначений для роботи в групах, можна вимкнути цей режим.

`/setprivacy` - встановити, які повідомлення отримуватиме бот при додаванні до групи. Якщо режим конфіденційності вимкнено, то бот отримуватиме всі повідомлення. Рекомендується залишити режим конфіденційності увімкненим.

Команди керування іграми.

`/newgame` - Створити нову гру.

`/listgames` - Отримати список ваших ігор.

`/editgame` – редагувати гру.

`/deletegame` - Видалити існуючу гру.

2.1.2 API Telegram Bot

Telegram використовує два основних інструменти API - API Telegram Bot і API Telegram. Перший призначений для розробки чат-ботів, а другий дозволяє створити повністю індивідуальний клієнт Telegram. Для розробників також доступна відкрита бібліотека TDLib (Telegram Database Library), за допомогою якої можна створювати побудований на TDLib) Telegram Bot API 34-це надбудована над Telegram API, тому Bot API можна використовувати, незважаючи на те, як працює протокол MTProto. Він використовує проміжний сервер з інтерфейсом HTTPS для шифрування трафіку та забезпечення зв'язку з Telegram API. Bot API дозволяє легко створювати програми, які використовують інтерфейс Telegram для виконання коду на локальному сервері. Користувачі можуть взаємодіяти з ботом, надсилаючи повідомлення, команди та вбудовані запити. Ось як це працює: будь-який робот постійно надсилає запити на сервер і періодично отримує оновлення. Отримати їх можна двома способами. Перший - це використання веб-хуків. Це коли сервер виконує зворотний виклик на певну URL-адресу. По-друге, ви можете "закинути" запит в Telegram і отримувати безперервну відповідь. Пам'ятайте, що ви можете отримати лише одне повідомлення про нове повідомлення, тобто на роботу. Тому, якщо ви вважаєте дані чату дуже важливими, варто самостійно вести список чатів та історію старих повідомлень. Якщо цю інформацію випадково видалити/втратити, її неможливо відновити. На відміну від API бота, який може отримувати оновлення лише один раз, API Telegram може обійти це обмеження, використовуючи кілька клієнтів. У цьому випадку бот отримує всі оновлення на кожному запущеному клієнті. Також бот API не може надсилати повідомлення всім користувачам одночасно.

2.2 Вибір засобів програмування

Мова Python має багато переваг перед іншими мовами програмування. Розробники можуть легко читати та розуміти програми на Python, оскільки мова має базовий синтаксис, схожий на синтаксис англійської. Python допомагає

розробникам бути більш продуктивними, оскільки вони можуть писати програми на Python, використовуючи менше рядків коду, ніж іншими мовами. Python має велику стандартну бібліотеку, що містить багаторазові коди практично для будь-якого завдання. В результаті розробникам не потрібно писати код із нуля. Розробники можуть легко поєднувати Python з іншими популярними мовами програмування: Java, C та C++. Активна спільнота Python складається з мільйонів розробників з усього світу. У разі виникнення проблем співтовариство допоможе в їх вирішенні. Крім того, в Інтернеті є безліч корисних ресурсів для вивчення Python. Python можна переносити на різні операційні системи: Windows, MacOS, Linux і Unix.

Дані наведено огляд найбільш популярних бібліотек Python.

Matplotlib використовують для відображення даних у високоякісній двох- і тривимірній (2D і 3D) графіці. Ця бібліотека використовується при вирішенні наукових задач. За допомогою Matplotlib дані можна візуалізувати у вигляді різних діаграм (наприклад, стовпчастих і лінійних). Також можна побудувати кілька діаграм відразу, а графік - переносити на будь-які платформи.

Pandas містить оптимізовані та гнучкі дані структури даних, які можна використовувати для керування даними тимчасових рядів і структурованими, такими як таблиці та масиви. Наприклад, Pandas можна використовувати для читання, запису, об'єднання, фільтрації та групування даних. Також дана бібліотека широко застосовується в науці про дані, аналізі даних і задачах машинного навчання.

NumPy - це популярна бібліотека, яка використовується розробниками для простого створення масивів і керування ними, а також керування логічними фігурами та виконання операцій лінійної алгебри. NumPy підтримує інтеграцію з багатьма мовами. Наприклад, C і C++.

OpenCV-Python - це бібліотека, яка використовується для обробки зображень при роботі з машинним виглядом. Вона містить функції обробки

зображень, такі як безліч одночасних читань і запис зображень, перетворення двовимірної в тривимірну, а також захоплення та аналіз зображень із відео.

2.3 Вибір середовища розробки

PyCharm - це міжплатформне інтегроване середовище розробки для мови програмування Python, розроблене JetBrains на основі IntelliJ IDEA. Надає користувачам набір інструментів для написання коду та візуального відладчика. Існує дві версії продукту: PyCharm Community Edition - безкоштовна версія, ліцензована за ліцензією Apache, PyCharm Professional Edition - розширена версія продукту з додатковими функціями, яка є пропрієтарним програмним забезпеченням. PyCharm дбає про щоденні завдання, щоб розробник міг зосередитися на більш важливих речах. Розумний механізм аналізу коду забезпечує точне автозаповнення, виявлення помилок і швидке їх виправлення, легку навігацію по коду та інші корисні функції. PyCharm був створений програмістами для програмістів, тому він містить усе необхідне для продуктивної розробки на Python.

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Моделювання вимог до системи

Щоб описати дії, які користувач може виконувати з чат-ботом, була розроблена UML-схема випадку. Функціональність системи представлена UML-схемою використання (рис. 3.1).

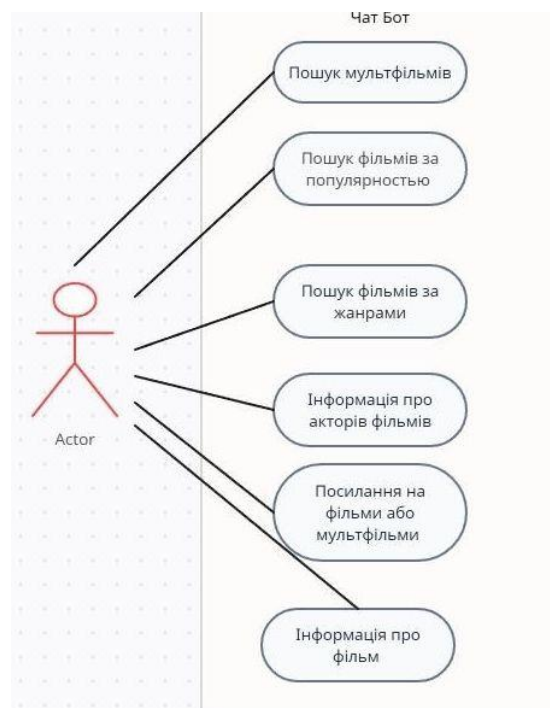


Рисунок 3.1 – Діаграма варіантів використання

Діаграма прецедентів зображує основні дії, доступні користувачеві в процесі спілкування з ботом:

- пошук фільмів за жанрами;
- пошук мультфільмів;
- пошук фільмів за рейтингом (популярність);
- шукати необхідну інформацію про фільм або мультфільм;
- отримати автоматичне посилання на фільм.

3.2 Розробка алгоритму підбору контенту

Алгоритм роботи чат-боту є послідовністю того, як використовує бота користувач. При старті боту перевіряється наявність побажань користувача. Якщо у користувача відсутнє побажання щодо фільму, то йому запропонує вибрати фільм які подобаються іншим користувачам та мають високі рейтинг фільму. Для користувача який має конкретну мету перегляду чат бот запропонує вибір між фільмом та мультфільмом. Після вибору категорії фільмів користувачу запропонують обрати жанр фільму, далі чат бот запропонує обрати фільм з декількох варіантів, після вибору фільма користувач отримає необхідну інформацію про сюжет та головних героїв, а також матиме посилання на фільм. При виборі цікавого відео контенту, користувач може повертатися на функцію назад, після чого тій людині прийде сповіщення (рис.3.2).

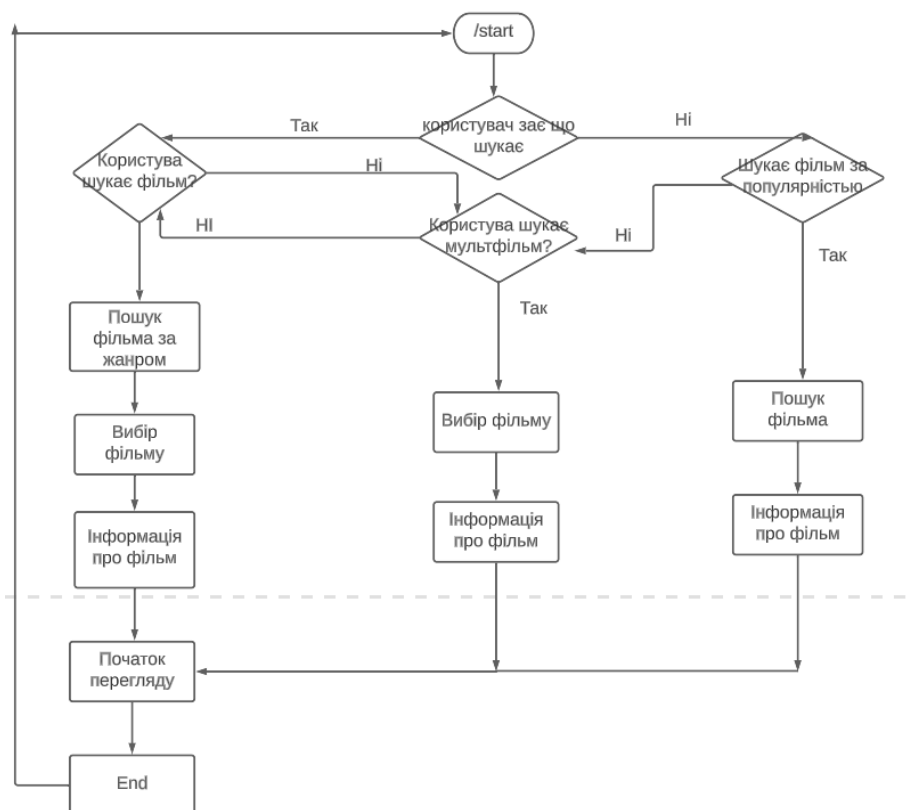


Рисунок 3.2– Алгоритм підбору контенту

3.3 Моделювання процесу обробки запиту до бота

Коли користувач надсилає повідомлення телеграм-боту, телеграм-бот перенаправляє його на Telegram API як код. Після цього бот за допомогою токена API отримується HTTP-запит з інформацією про користувача, час, інші деталі та його повідомлення. Після обробки запиту користувача, код бота надсилає HTTP-запит у відповідь, також використовуючи Маркер API, де Telegram API надає отриману відповідь від імені Telegram бота (рис.3.3).

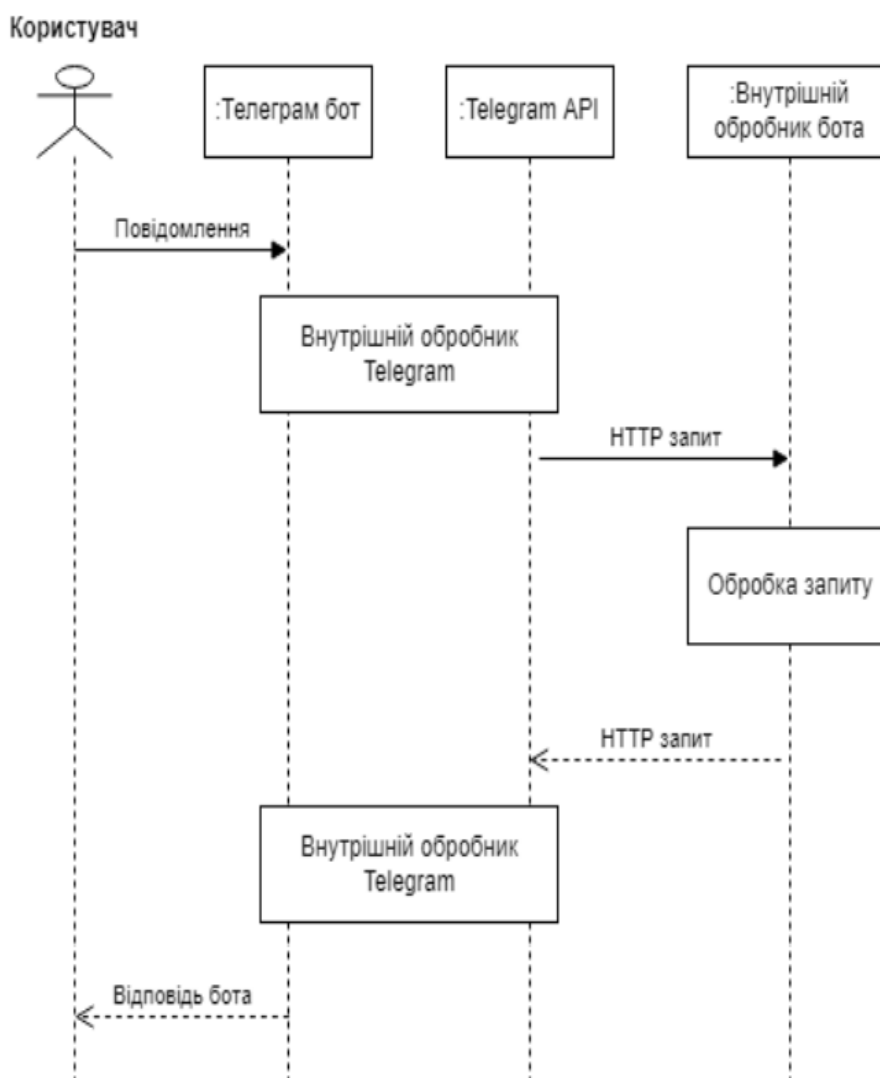


Рисунок 3.3 – Діаграма послідовності процесу обробки запиту до бота

3.4 Розробка архітектури бази даних

База даних використовується для зберігання анкет користувачів та інших даних про роботу робота. Для нього була розроблена схема структури бази даних. Таблиця користувачів зберігає профілі користувачів та інформацію про них. поле `user_id` і `chat_id` зберігають інформацію про облікові записи користувачів Telegram і використовуються для запиту щодо пошуку цікавого відео контенту (рис.3.4).

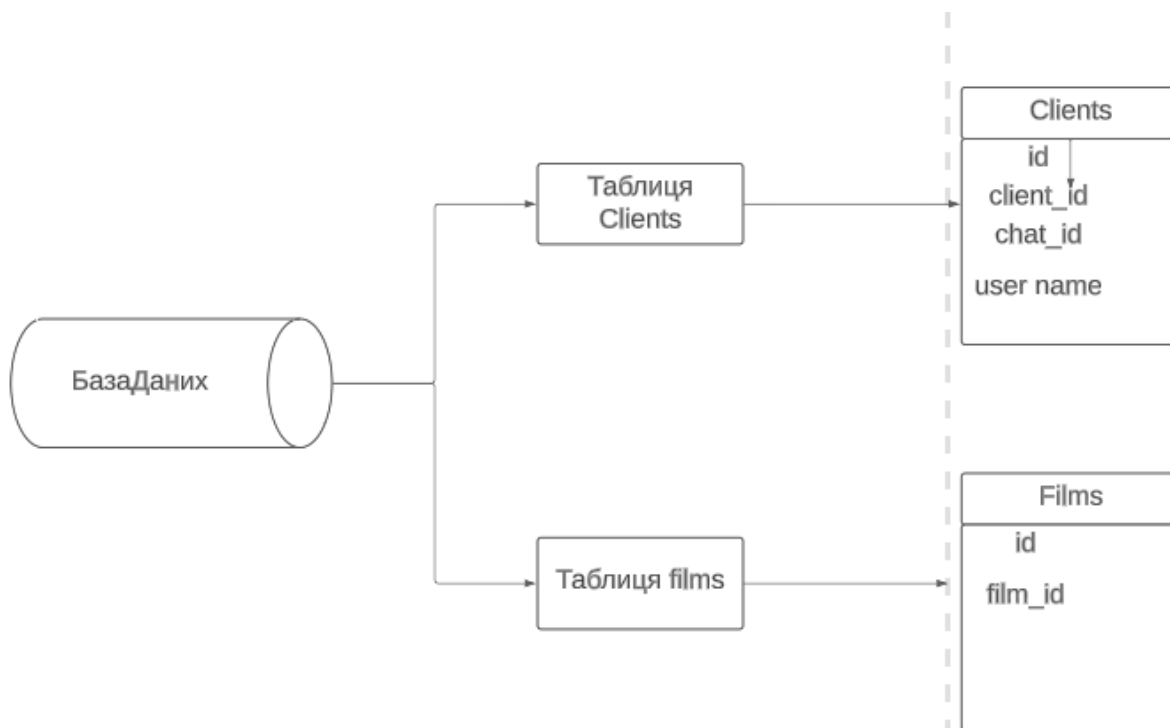


Рисунок 3.4 – Архітектура бази даних

3.4 Розробка архітектури системи

Діаграми пакетів — це структурні діаграми, які використовуються для відображення організації та розташування різних елементів моделі в пакетах. Пакет — це група пов'язаних елементів UML, таких як діаграми, документи, класи або навіть інші пакети. Кожен елемент вкладено в пакет, описаний на схемі як папку файлів, які потім упорядковуються на схемі ієрархічно. Пакетні діаграми використовуються для забезпечення візуальної організації багаторівневих

архітектур у будь-якому UML-класифікаторі, такому як програмна система. На рисунку 3.5 наведено архітектуру системи, описано за допомогою діаграми пакетів.

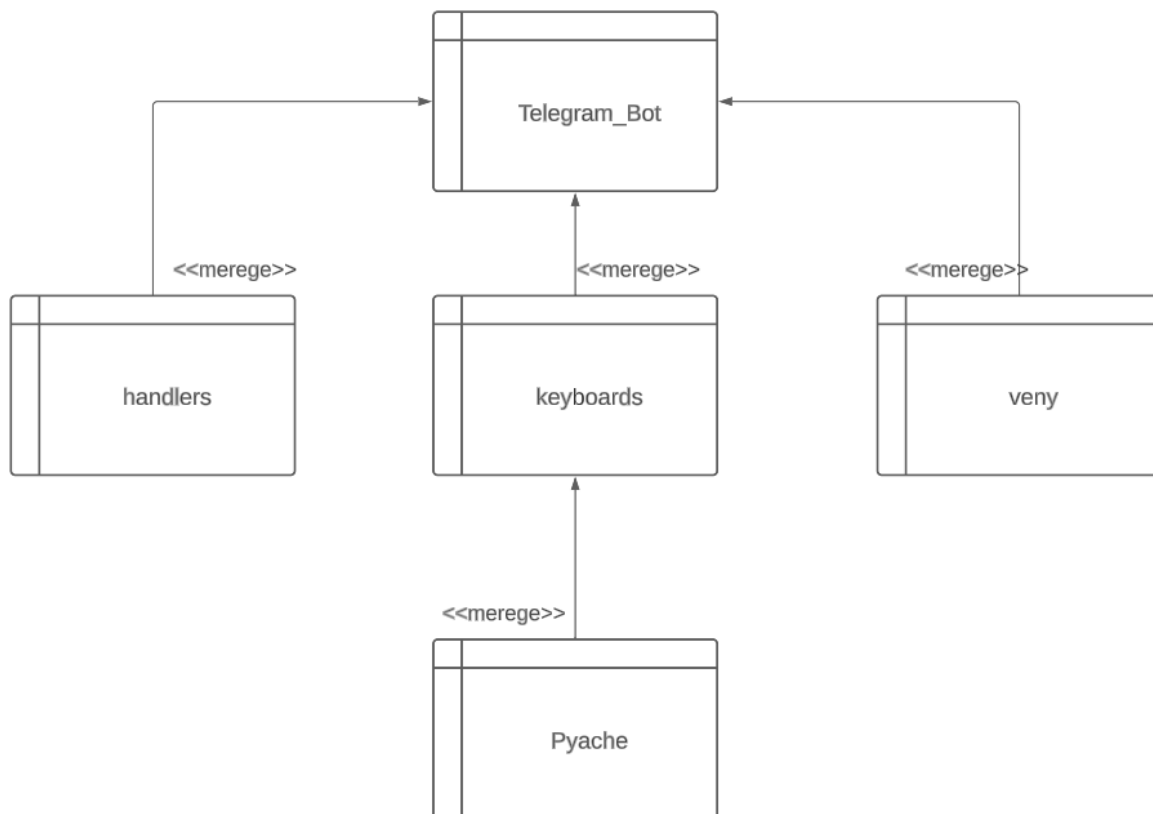


Рисунок 3.5 – Діаграма пакетів системи

3.5 Розробка логіки функціонування чат-боту

Структура бота - це блок-схема всієї логіки бота, на основі якої створюється чат-бот, що представляє собою технічне завдання створення чат-бота або налаштування бота на конструкторі. На схемі на рис.3.16 показано три основних сценарії.



Рисунок 3.6 – Структура чат-боту

Перша сцена . Пошук фільмів за найвищим рейтингом відбувається так

- Обрати категорію за найвищим рейтингом
- Обрати зацікавлений фільм
- Отримати всю необхідну інформацію про фільм
- Вибрати фільм та почати перегляд.

Друга сцена . Пошук конкретних фільмів за жанрами

- Обрати категорію пошук фільмів
- Обрати жанр який до вподоби
- Обрати зацікавлений фільм
- Отримати всю необхідну інформацію про фільм
- Вибрати фільм та почати перегляд.

Третя сцена . Пошук мультфільмів

- Обрати категорію пошук мультфільмів

- Обрати зацікавлений мультфільм
- Отримати всю необхідну інформацію про мультфільм
- Вибрати фільм та почати перегляд.

3.6 Опис функціонування чат-боту

Для того щоб знайти бота потрібно зайти у Telegram та у пошуку вбити його назву «Cinema_Bot » або телеграм-адресу «@CinemaBot». Після знаходження та запуску бота автоматично виконається команда «/start», бот привітає нового користувача та запропонує обрати фільм (рис. 3.7).



Рисунок 3.7 – Сторінка привітання

На рисунку 3.8 зображено форма знімку екрану, на якій нам чат бот пропонує обрати жанр фільму який користувач бажає для перегляду.

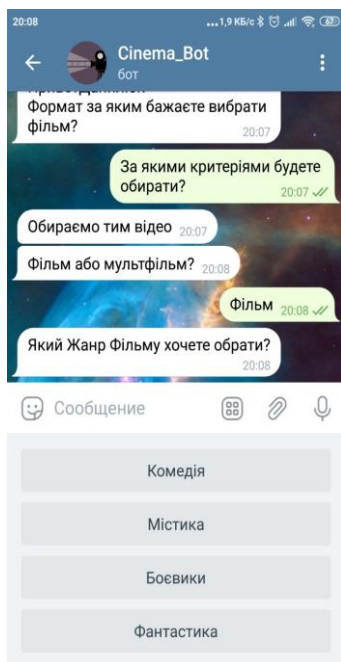


Рисунок 3.8 – Сторінка вибору жанру

На рисунку 3.9 зображено форма знімку екрану, на якій нам чат бот пропонує обрати сам фільм, який більше хоче користувач.



Рисунок 3.9 – Сторінка вибору фільму

На рисунку 3.10 зображено форма знімку екрану зображено інформаційне поле фільму де користувач може прочитати про сюжет та головних героїв фільму

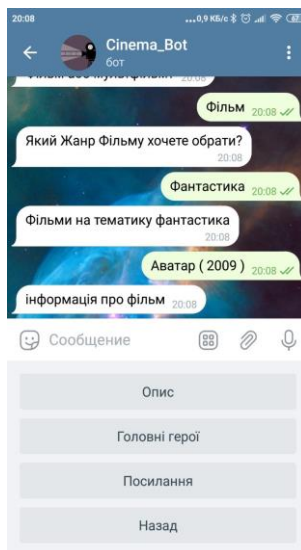


Рисунок 3.10 – Сторінка інформаційного поля фільму

На рисунку 3.11 зображено форма знімку екрану, де можна побачити, як подається інформаційне поле.

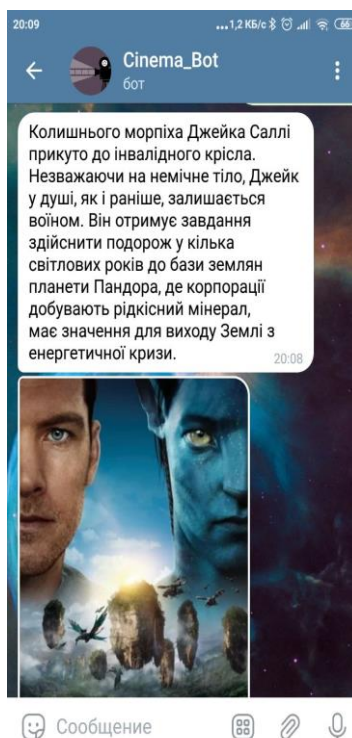


Рисунок 3.11 – Сторінка детальної інформації про фільм

На рисунку 3.12 зображено форма знімку екрану посилання на фільм. Перевагою використання додатку є те, що посилання формується безпосередньо в середовищі месенджеру і користувачеві немає потреби в переході на інші ресурси для отримання посилання.

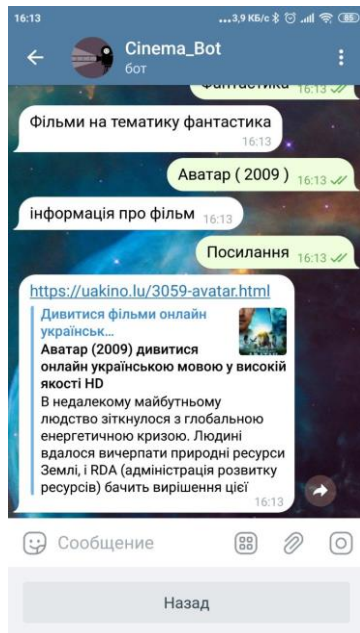


Рисунок 3.12 – Сторінка посилання на ресурс

На рисунку 3.13 зображено форма знімку екрану де можна побачити як подається інформаційне поле мультфільму.

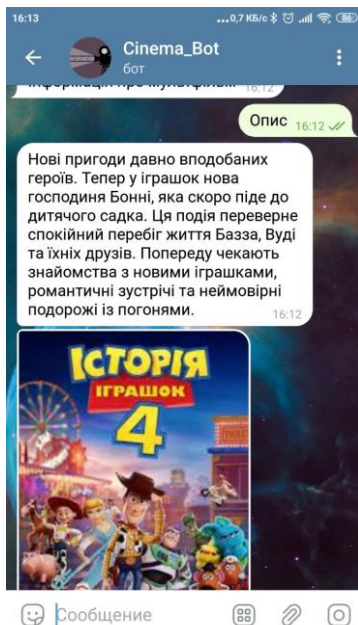


Рисунок 3.13 – Сторінка інформаційного поля мультфільму

На рисунку 3.14 зображено форму знімку екрану, де зображено інформаційне поле фільму, де користувач може прочитати про сюжет та головних героїв фільму.

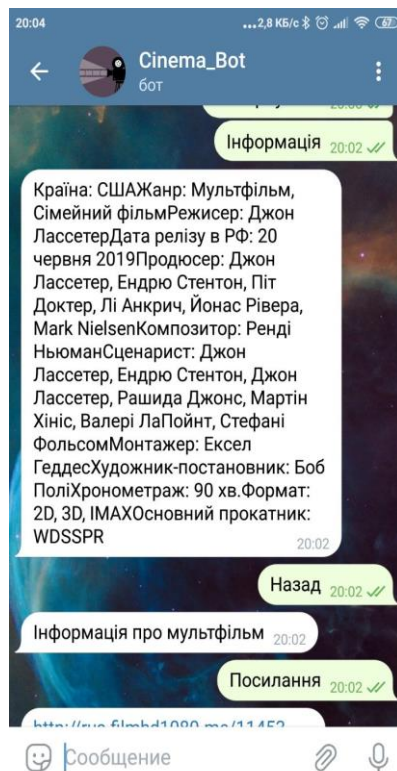


Рисунок 3.14 – Сторінка з описом сюжету та головних героїв фільму

3.7 Тестування чат-боту

Тестування програмного забезпечення – це метод перевірки того, чи фактичний програмний продукт відповідає очікуваним вимогам, і переконатися, що програмний продукт без дефектів. Він передбачає виконання компонентів програмного забезпечення/системи за допомогою ручних або автоматизованих інструментів для оцінки однієї чи кількох цікавих властивостей. Метою тестування програмного забезпечення є виявлення помилок, прогалин або відсутніх вимог на відміну від фактичних вимог.

Тестування №1 Пошук фільму Аватар

Test Case ID	Test Prior	Test Case Title	Step	Test Steps	Test Data	Expected Result	Status
Ф - 1	Середній	Фільм Аватар	1	Відкрити Telegram	Додаток Телеграм	Відкрився Телеграм	Pass
			2	Вибрати телеграм бот	Сінема_Bot		Pass
			3	Обрати необхідну категорію кнопки	За якими критеріями будете обирати?	у полі емейл відображаються категорії	Pass
			4	Обрати необхідну кнопку	Фільми	у полі пароль відображається кнопки фі	Pass
			5	Обрати необхідний жанр фільму.	Фантастика	у полі прізвище відображається жанри с	Pass
			6	Обрати фільм	Аватар	у формі ім'я відображається фільми	Pass
			7	Потрапляємо в інформаційне поле. Де обираємо необхідну інформацію		у полі відображається кнопки Опис, Рей	Pass
			8	Опис фільму		Читаємо опис фільму	Pass
			9	Рейтинг фільму		Читаємо рейтинг фільму	Pass
			10	Посилання на фільм.		Переходимо по посиланню	Pass

Рисунок 3.10

Тестування №2 Обираємо іншу категорію

Test Case ID	Test Prior	Test Case Title	Step	Test Steps	Test Data	Expected Result	Status
Ф-2	Середній	Обираємо іншу категорію	1	Відкрити Telegram	Додаток Телеграм	Відкрився Телеграм	Pass
			2	Вибрати телеграм бот	Сінема_Bot		Pass
			3	Обрати необхідну категорію кнопки	За якими критеріями будете обирати?	у полі емейл відображаються категорії	Pass
			4	Обрати необхідну кнопку	Фільми	у полі пароль відображається кнопки фі	Pass
			5	Обрати необхідний жанр фільму.	Босвики	у полі прізвище відображається жанри с	Pass
			6	Обрати фільм	Форсак	у формі ім'я відображається фільми	Pass
			7	Потрапляємо в інформаційне поле. Де обираємо необхідну інформацію		у полі відображається кнопки Опис, Рей	Pass
			8	Опис фільму		Читаємо опис фільму	Pass
			9	Рейтинг фільму		Читаємо рейтинг фільму	Pass
			10	Посилання на фільм.		Переходимо по посиланню	Pass

Рисунок 3.11

Тестування №3 Обираємо мультфільм

Test Case ID	Test Prior	Test Case Title	Step	Test Steps	Test Data	Expected Result	Status
МФ - 3	Середній	Холодне серце	1	Відкрити Telegram	Додаток Телеграм	Відкрився Телеграм	Pass
			2	Вибрати телеграм бот	Сінема_Bot		Pass
			3	Обрати необхідну категорію кнопки	За якими критеріями будете обирати?	у полі емейл відображаються категорії	Pass
			4	Обрати необхідну кнопку мультфільм	мультфільми	у полі пароль відображається кнопки фі	Pass
			5	Обрати фільм		у полі прізвище відображається жанри м	Pass
			6	Потрапляємо в інформаційне поле. Де обираємо необхідну інформацію	Холодне серце	у формі ім'я відображається мультфільм	Pass
			7	Опис мультфільму		у полі відображається кнопки Опис, Рей	Pass
			8	Рейтинг мультфільму		Читаємо опис мультфільму	Pass
			9	Посилання на мультфільм.		Читаємо рейтинг мультфільму	Pass
			10			Переходимо по посиланню	Pass

Рисунок 3.12

ВИСНОВКИ

1. Проведено огляд предметної галузі пошуку цікавого відео-контенту. Визначено характеристики цільової аудиторії.

2. Проведено огляд та аналіз наявних аналогів Telegram чат-ботів для пошуку цікавого відео-контенту. Визначено їх ключові переваги та недоліки.

3. Визначено вимоги до структури та алгоритму роботи чат-боту, сформовано технічне завдання. Для моделювання вимог, проектування архітектури та даних використано діаграми UML.

4. Розроблено Telegram чат-бот для пошуку цікавого відео-контенту. Для розробки використано Telegram API та мову Python.

5. Розроблено тест-кейси та проведено тестування чат-боту.

ПЕРЕЛІК ПОСИЛАНЬ

1. The Movie Bot [Електронний ресурс] – Режим доступу до ресурсу: <https://bestoftelegram.com/bots/MovieDatabaseBot>.
2. acutebot [Електронний ресурс] – Режим доступу до ресурсу: <https://bestoftelegram.com/bots/acutebot>.
3. Переваги та недоліки використання чат-ботів [Електронний ресурс] – Режим доступу до ресурсу: <https://internetdevels.ua/blog/pros-and-cons-of-using-chatbots-for-business>.
4. FilmAffinity Bot [Електронний ресурс] – Режим доступу до ресурсу: <https://bestoftelegram.com/bots/faffinitybot>.
5. Кращі мови програмування - 10 кращих мов програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.education-wiki.com/4077268-best-programming-languages>.
6. Що таке чат-бот? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oracle.com/chatbots/what-is-a-chatbot/>.
7. Гібридний чат-бот: як змусити людей і роботів працювати разом [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ringcentral.com/us/en/blog/integration-agents-chatbots-botmind>
8. Що таке чат-бот? Різновиди чат-ботів для бізнесу. [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/article/sho_take_chatbot_riznovidi_chatbotiv_dlya_biznesu.
9. Переваги розробки чат-ботів без коду [Електронний ресурс] – Режим доступу до ресурсу: <https://landbot.io/blog/no-code-chatbot-development#:~:text=No%2Dcode%20tools%20work%20by,chatbots%20according%20to%20their%20needs..>
10. Приголомшлива статистика чат-ботів у 2023 році: переваги, демографія, тенденції, маркетинг [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bloggersideas.com/uk/chatbot-statistics/>.

11. Що таке Python? [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/ru/what-is/python/>.

12. Кращі бібліотеки Python у 2022 [Електронний ресурс] – Режим доступу до ресурсу: <https://merehead.com/ru/blog/best-python-libraries-know-in-2022/>.

13. PyCharm. [Електронний ресурс] – Режим доступу до ресурсу: <https://wikipedia.org/wiki/PyCharm>.

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка Telegram чат-боту для пошуку цікавого відео-контенту мовою Python

Виконав студент4 курсу
групи ПД - 43
Данилюк Олег Сергійович
Керівник роботи

К.т.н, доц, доцент кафедри ІПЗ Золотухіна Оксана Анатоліївна
Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- Мета роботи - спрощення процесу пошуку фільмів , мультфільмів та інших відео за рахунок Telegram чат боту мовою Python.
- Об'єкт дослідження - процес автоматизації процесу пошуку цікавого відео контенту.
- Предмет дослідження - програмне забезпечення для допомоги автоматизованого пошуку інформації про цікавий відео контент.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

Завдання кваліфікаційної (бакалаврської) роботи є розробка чат бота для пошуку цікавого відео контенту у месенджері Telegram. Виходячи з поставленої мети, були поставлені такі завдання:

1. Розглянути визначення та поняття чат-ботів та коротку історію їх появи;
2. Пошук та порівняння наявних аналогів чат-ботів;
3. Виявлення вимог, якими має володіти програмне забезпечення;
4. Розробка та впровадження розробленого програмного забезпечення;
5. Тестування розробленого чат боту ;

3

ТАБЛИЦЯ ПОРІВНЯННЯ АНАЛОГІВ

Показник	The Movie Bot	FilmAffinity.bot	Cinema_Bot (Розроблений бот)
Локалізація українською мовою	–	–	+
Наявність пошуку по категорії мультфільм	–	–	+
Відображення рейтингу фільмів	–	–	+
Наявність пошуку по жанрам фільмів	+	+	+
Наявність пошуку по акторами фільмів	+	–	–
Загальна інформація про фільм (короткий сюжет, головні герої)	+	+	+
Посилання на фільм	+	+	+

4

ВИМОГИ ДО ДОДАТКУ

Функціональні вимоги

1. Розробити додаток на українській мові для загального вітчизняного користування.
2. Користувач повинен мати можливість пошуку фільмів за жанрами.
3. Користувач повинен мати можливість пошуку фільмів за найпоширеніших фільмів світу.
4. Користувач повинен отримувати необхідну інформацію про фільм.
 - Короткий опис сюжету фільму
 - Головні герої
5. Користувач повинен мати можливість отримати посилання на фільм не виходячи з додатку.

Нефункціональні вимоги

1. Простий інтерфейс користувача.
2. Надійність додатку.
3. Продуктивність додатку.

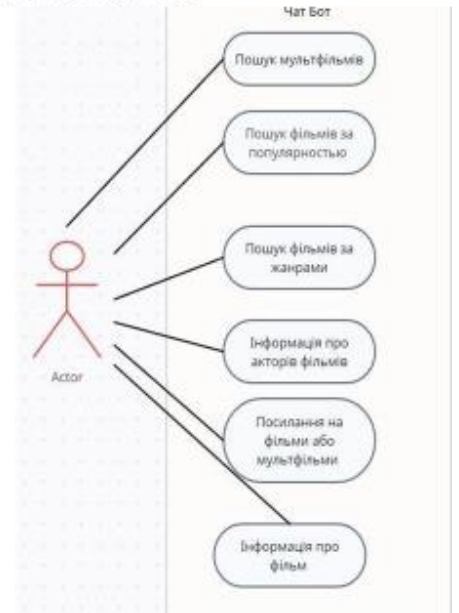
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

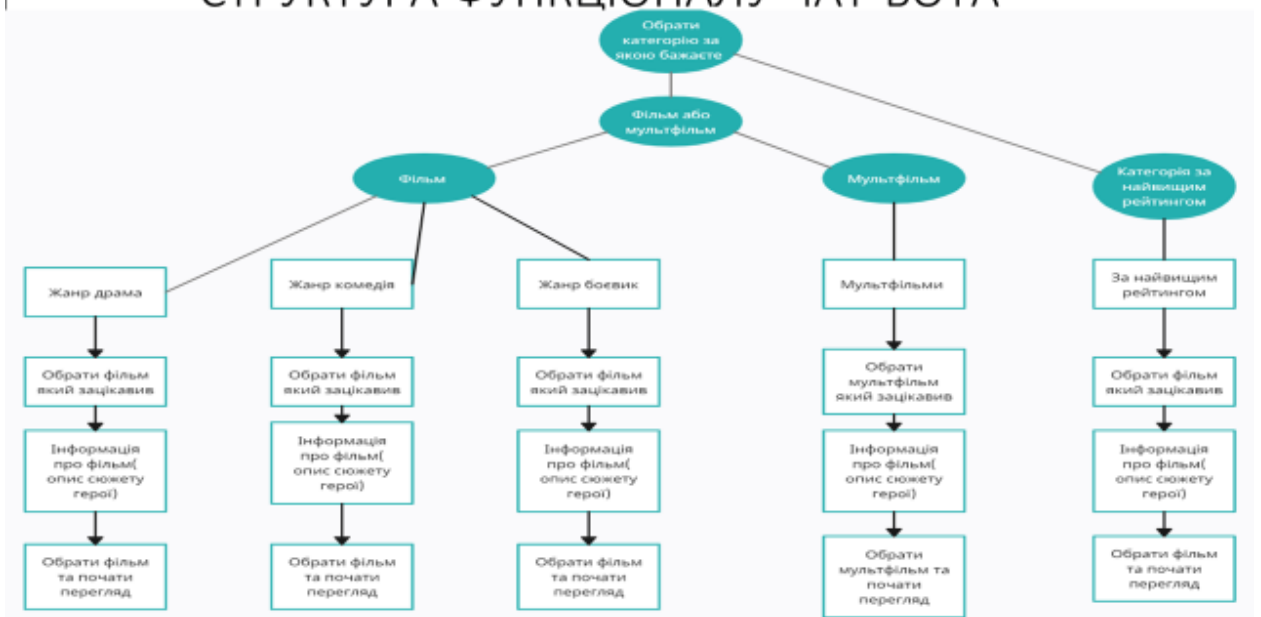


6

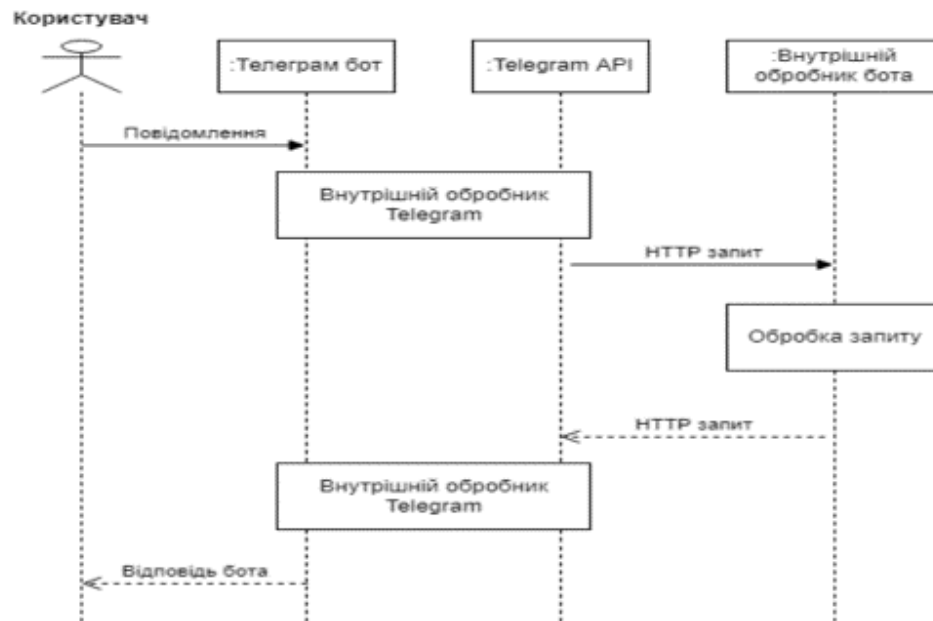
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



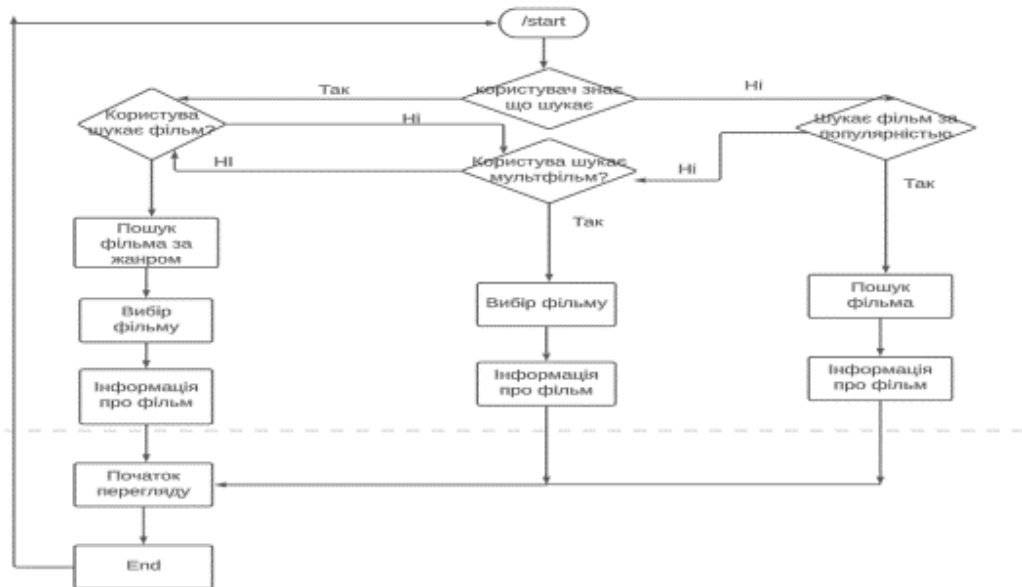
СТРУКТУРА ФУНКЦІОНАЛУ ЧАТ-БОТА



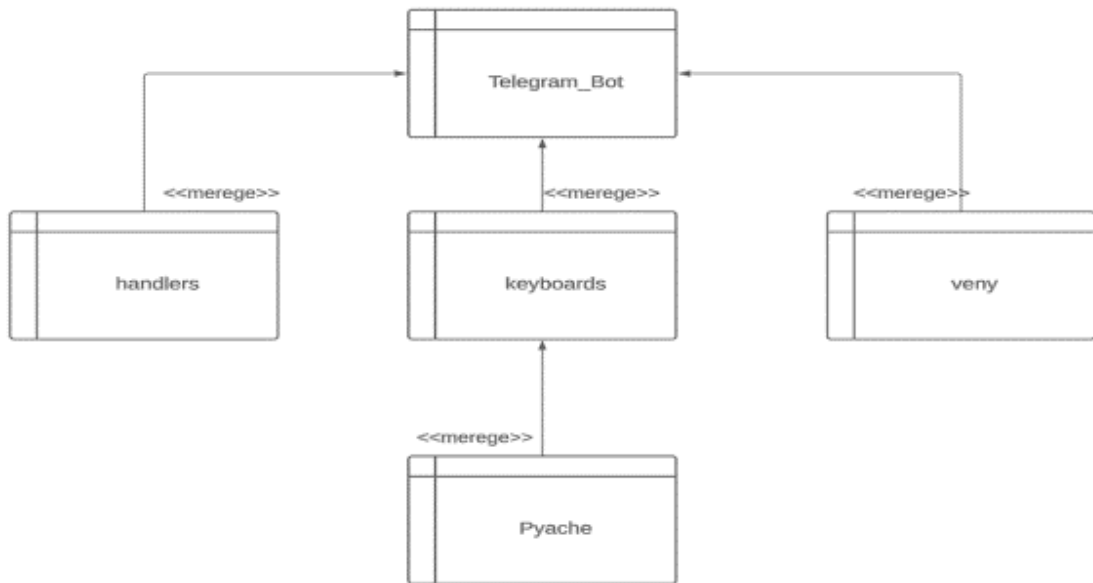
ДІАГРАМА ПОСЛІДОВНОСТЕЙ



АЛГОРИТМ ПІДБОРУ КОНТЕНТУ



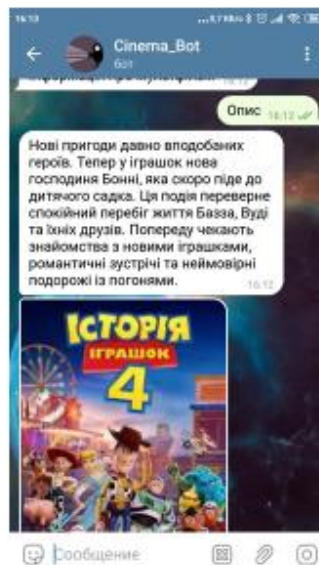
ДІАГРАМА ПАКЕТІВ



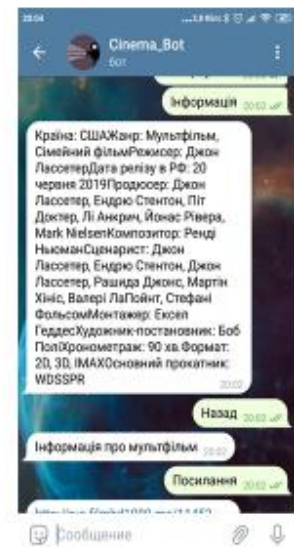
ЕКРАНИ ФОРМИ РОЗРОБЛЕНОГО ТЕЛЕГРАМ ЧАТ БОТУ



Стартовий екран чат боту

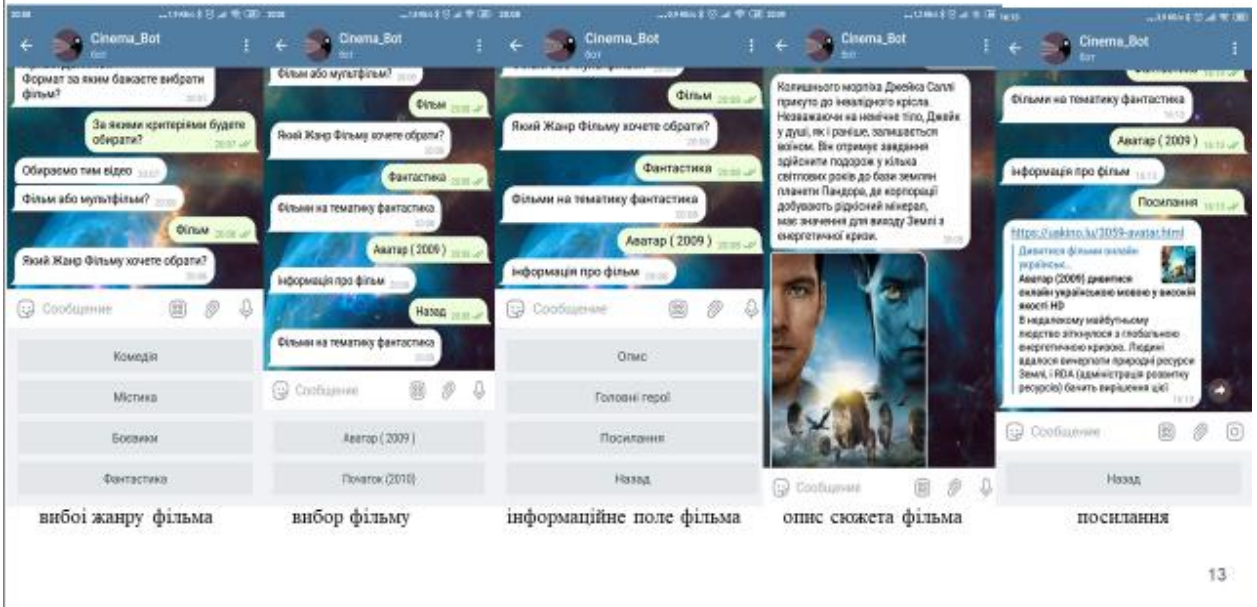


Екран короткого опису сюжету



Інформація про створення мультфільма

ЕКРАНИ ФОРМИ РОЗРОБЛЕНОГО ТЕЛЕГРАМ ЧАТ БОТУ



АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Данилюк О.С. Дослідження автоматизації Бізнесу за допомогою Telegram та Telegramчат - боту / Данилюк О.С., Золотухіна О.А.//Автоматизація та управління бізнес процесами Всеукраїнська науково-технічна конференція Застосування програмного забезпечення в інфокомунікаційних технологіях. Збірник тез. 20.04.2023, ДУТ, м. Київ — К.: ДУТ, 2023. — С. 141.

2. Данилюк О.С. Розробка автоматизованої програмного забезпечення для пошуку цікавого відео – контенту за допомогою діалогового штучного інтелекту / Данилюк О.С., Золотухіна О.А.//ІоТ та штучний інтелект; «СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ». Збірник тез 7.04.2023 ДУТ, м. Київ — К.: ДУТ, 2023. — С. 135.

ВИСНОВКИ

1. Проведено огляд предметної галузі пошуку цікавого відео-контенту. Визначено характеристики цільової аудиторії.
2. Проведено огляд та аналіз наявних аналогів Telegram чат-ботів для пошуку цікавого відео-контенту. Визначено їх ключові переваги та недоліки.
3. Визначено вимоги до структури та алгоритму роботи чат-боту, сформовано технічне завдання. Для моделювання вимог, проектування архітектури та даних використано діаграми UML.
4. Розроблено Telegram чат-бот для пошуку цікавого відео-контенту. Для розробки використано Telegram API та мову Python.
5. Розроблено тест-кейси та проведено тестування чат-боту.

ДОДАТОК Б

ЛІСТИНГИ ПРОГРАМНОГО КОДУ

```
from flask import Flask
from flask import request
from flask import Response
import requests

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        msg = request.get_json()
        print(msg)

        return Response('ok', status=200)
    else:
        return "<h1>Welcome!</h1>"

if __name__ == '__main__':
    app.run(debug=True)

TOKEN = "<5627458810:AAHqK2A-V3W98DVXy1bQZV2wmhRYccBi1Q4>"
app = Flask(Cinema_Bot )

def parse_message(message):
    print("message-->",message)
    chat_id = message['message']['chat']['id']
```

```

txt = message['message']['text']
print("chat_id-->", chat_id)
print("txt-->", txt)
return chat_id,txt

def tel_send_message(chat_id, text):
    url = f'https://api.telegram.org/bot{5627458810:AAHqK2A-
V3W98DVXy1bQZV2wmhRYccBi1Q4}/sendMessage'
    payload = {
        'chat_id': chat_id,
        'text': text
    }

    r = requests.post(url,json=payload)
    return r

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        msg = request.get_json()

        chat_id,txt = parse_message(msg)
        if txt == "hi":
            tel_send_message(chat_id,"start!!")
        else:
            tel_send_message(chat_id,'from webhook')

    return Response('ok', status=200)
else:
    return "<h1>start!</h1>"

```

```
if __name__ == '__main__':
    app.run(debug=True)

TOKEN = "<5627458810:AAHqK2A-V3W98DVXy1bQZV2wmhRYccBi1Q4>"

app = Flask (Cinema_Bot)

def tel_parse_message(message):
    print("message-->",message)
    try:
        chat_id = message['message']['chat']['id']
        txt = message['message']['text']
        print("chat_id-->", chat_id)
        print("txt-->", txt)

        return chat_id,txt
    except:
        print("NO text found-->>")

def tel_send_message(chat_id, text):
    url = f'https://api.telegram.org/bot{TOKEN}/sendMessage'
    payload = {
        'chat_id': chat_id,
        'text': text
    }

    r = requests.post(url,json=payload)

    return r
```



```

def tel_send_image(chat_id):
    url = f'https://api.telegram.org/bot{5627458810:AAHqK2A-
V3W98DVXy1bQZV2wmhRYccBi1Q4}/sendPhoto'

    payload = {
        'chat_id': chat_id,
        'photo': "",
        'caption': "This is a sample image"
    }

    r = requests.post(url, json=payload)
    return r

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        msg = request.get_json()
        try:
            chat_id, txt = tel_parse_message(msg)
            if txt == "hi":
                tel_send_message(chat_id, "Hello, world!")
            elif txt == "image":
                tel_send_image(chat_id)

        else:
            tel_send_message(chat_id, 'from webhook')
    except:
        print("from index-->")

    return Response('ok', status=200)
else:

```

```

    return "<h1>Welcome!</h1>"

if __name__ == '__main__':
    TOKEN = "<5627458810:AAHqK2A-V3W98DVXy1bQZV2wmhRYccBi1Q4>"

    app = Flask (Cinema_Bot)

def tel_parse_message(message):
    print("message-->",message)
    try:
        chat_id = message['message']['chat']['id']
        txt = message['message']['text']
        print("chat_id-->", chat_id)
        print("txt-->", txt)

        return chat_id,txt
    except:
        print("NO text found-->>")

def tel_send_message(chat_id, text):
    url = f'https://api.telegram.org/bot{TOKEN}/sendMessage'
    payload = {
        'chat_id': chat_id,
        'text': text
    }

    r = requests.post(url,json=payload)

    return r

def tel_send_image(chat_id):

```

```
TOKEN = "<5627458810:AAHqK2A-V3W98DVXy1bQZV2wmhRYccBi1Q4>"
```

```
app = Flask (Cinema_Bot)
```

```
def tel_parse_message(message):
```

```
    print("message-->",message)
```

```
    try:
```

```
        chat_id = message['message']['chat']['id']
```

```
        txt = message['message']['text']
```

```
        print("chat_id-->", chat_id)
```

```
        print("txt-->", txt)
```

```
        return chat_id,txt
```

```
    except:
```

```
        print("NO text found-->>")
```

```
def tel_send_message(chat_id, text):
```

```
    url = f'https://api.telegram.org/bot{TOKEN}/sendMessage'
```

```
    payload = {
```

```
        'chat_id': chat_id,
```

```
        'text': text
```

```
    }
```

```
    r = requests.post(url,json=payload)
```

```
    return r
```

```
def tel_send_image(chat_id):
```

```
    app.run(threaded=True)
```

```
def tel_send_button(chat_id):  
    url = f'https://api.telegram.org/bot{TOKEN}/sendMessage'  
  
    payload = {  
        'chat_id': chat_id,  
        'text': "Привет",  
        'reply_markup': {'keyboard': [[{"За якими критеріями будете обирати?": 'Обираємо  
тим відео"Мультфільм"Фільм"Опис"Рейтинг"посилання "посилання "головні герої "За  
найвищим рейтингом '}, {"За якими критеріями будете обирати?": 'Обираємо тим  
відео"Мультфільм"Фільм"Опис"Рейтинг"посилання "посилання "головні герої "За  
найвищим рейтингом '}]]}  
    }  
  
    r = requests.post(url, json=payload)  
  
    return r
```