

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

« » червня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційно-аналітична система визначення цільової аудиторії
бізнесу у соціальних мережах»
здобувача групи ІН-03-1 Стеценка Станіслава Ігоровича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

Станіслав СТЕЦЕНКО

_____ (підпис)

Керівник,
старший викладач кафедри
комп'ютерних наук, к.т.н., доцент

Дмитро ВЕЛИКОДНИЙ

_____ (підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми
«Інформатика»
здобувача групи ІН-03-1 Стеценка С. І.

1. Тема роботи: «Інформаційно-аналітична система визначення цільової аудиторії бізнесу у соціальних мережах» затверджую наказом по СумДУ від 22 квітня 2024 р. No 0414-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 29 травня 2024 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз предметної області, постановка задач дослідження.
2) Огляд існуючих рішень 3) Моделювання процесу використання та архітектури інформаційно-аналітичної системи. 4) Розробка програмного забезпечення 5) Тестування програмного забезпечення 6) Аналіз результатів роботи.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____

_____ (підпис)

Керівник _____

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз предметної області, постановка задач дослідження</i>	06.05.24-08.05.24	
2	<i>Огляд існуючих рішень</i>	09.05.24-10.05.24	
3	<i>Моделювання процесу використання та архітектури інформаційно-аналітичної системи</i>	10.05.24-14.05.24	
4	<i>Розробка програмного забезпечення</i>	14.05.24-16.05.24	
5	<i>Тестування програмного забезпечення</i>	16.05.24-17.05.24	
6	<i>Аналіз результатів роботи</i>	17.05.24 - 18.05.24	
7	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	18.05.24- 20.05.24	

Здобувач вищої освіти

_____ (підпис)

Керівник

_____ (підпис)

АНОТАЦІЯ

Записка: 72 стор., 32 рис., 1 додаток, 30 використаних джерел.

Обґрунтування актуальності теми роботи – тема роботи є актуальною, оскільки клієнти все частіше шукають надійні джерела професійних бізнес-порад в умовах зростаючої ролі соціальних мереж.

Об’єкт дослідження — процес надання бізнес-порад клієнтам за допомогою інтерактивного Telegram-бота, що базується на технологіях штучного інтелекту.

Мета роботи — розробка інтерактивного Telegram-бота, який надаватиме клієнтам професійні бізнес-поради, використовуючи технології штучного інтелекту.

Методи дослідження — аналіз літературних джерел та існуючих систем, створення архітектури бази даних та розробка програмного забезпечення, тестування системи.

Результати — розроблено інтерактивний Telegram-бот, який функціонує на основі штучного інтелекту і надає користувачам можливість отримувати бізнес-поради за допомогою чату.

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА, ТЕЛЕГРАМ БОТ, JAVA, JSON,
НЕЙРОМЕРЕЖА CHATGPT 3.5 TURBO.

Зміст

ВСТУП.....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
1.1 Актуальність та аналіз існуючої проблеми.....	7
1.2 Telegram-боти	11
1.3 Огляд існуючих систем	16
1.4 Постановка задачі.....	24
2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ	25
2.1 Вибір мови програмування	25
2.2 Штучний інтелект.....	28
2.3 Моделювання процесу використання Телеграм боту	29
2.4 Архітектура чат-боту.....	40
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	43
3.1 Реєстрація нового чат бота засобами Телеграм API.....	43
3.2 Результати реалізації чат боту	44
3.3 Тестування	49
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
Додаток А. Програмна реалізація	58

ВСТУП

Актуальність. У сучасному світі все більше зростає використання новітніх технологій у сфері бізнесу. Одними з таких технологій є інтерактивні чат-боти. Вони набувають все більшої популярності, стають не тільки зручними інструментами комунікації, але й ефективними помічниками у різних сферах діяльності, зокрема в сфері бізнесу.

Метою даного дослідження є створення телеграм боту, який надаватиме корисні та професійні поради в сфері бізнесу. Цей бот буде спрямований на надання підтримки та консультацій підприємцям, малому бізнесу та стартапам, які шукають експертні поради щодо стратегій розвитку, маркетингу, фінансів та інших аспектів підприємницької діяльності.

Об'єкт дослідження. Процес надання бізнес-порад клієнтам за допомогою інтерактивного Telegram-бота, що базується на технологіях штучного інтелекту.

Гіпотеза. Розроблений телеграм бот, надаючи корисні поради в сфері бізнесу, значно полегшить процес прийняття рішень та розвитку підприємницької діяльності для своїх користувачів.

Новизна. Новизна роботи полягає в розробці телеграм боту, що використовує штучний інтелект для надання професійних порад у сфері бізнесу, на відміну від існуючих інформаційно-аналітичних систем, які як правило обмежені статичними аналітичними звітами або не надають інтерактивного спілкування з користувачем.

Структура. Робота складається із вступу, аналізу існуючих систем, постановки задачі, моделювання процесу використання телегра боту, архітектури телеграм боту, програмної реалізації, тестування, висновків, списку використаних джерел, додатків.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Актуальність та аналіз існуючої проблеми

В сучасних умовах розвитку бізнесу інформація та консалтинг відіграють ключову роль у досягненні успіху підприємств [1]. Підприємства постійно шукають нові методи для підвищення своєї конкурентоспроможності, ефективного управління ресурсами та прийняття обґрунтованих рішень. Інформація щодо нових ринкових тенденцій, стратегій розвитку, маркетингових кампаній, фінансових прогнозів та інших важливих аспектів бізнесу стає необхідною для виживання та зростання у все більш конкурентному середовищі [2].

Чат-боти, створені для надання корисної інформації у сфері бізнесу, мають багато переваг для компаній та підприємств. Вони можуть значно підвищити ефективність, продуктивність та загальний успіх бізнесу [3].

Інтеграція чат-ботів у бізнес-процеси надає компаніям значні переваги, включаючи підвищення ефективності, зниження витрат, поліпшення обслуговування клієнтів та можливість надання персоналізованих послуг. Чат-боти стають незамінним інструментом для сучасних компаній, які прагнуть залишатися конкурентоспроможними на ринку та забезпечувати високий рівень задоволеності клієнтів. Завдяки своїй гнучкості та масштабованості, чат-боти можуть адаптуватися до будь-яких потреб бізнесу, що робить їх важливим елементом стратегії розвитку та успіху компаній.

Чат-боти мають багато переваг, зокрема:

- Цілодобова доступність
- Швидкість відповіді
- Точність та якість інформації
- Гнучкість та масштабованість
- Підтримка процесів навчання та розвитку

Новітні чат-боти радикально змінюють сферу інтернет-торгівлі, підвищують продуктивність бізнесу і сприяють його масштабуванню.

Статистика ChatInsight показує, що на початку 2024 року понад 91% американських інтернет-магазинів використовували інтелектуальні чат-боти, які можуть автоматично обробляти понад 70% запитів клієнтів [4]. А в Україні згідно з останніми статистичними даними 24,9% споживачів використовують чат-боти для взаємодії з бізнесом, порівняно з 13% у 2018 році. За даними GoogleTrends можна побачити, що інтерес до чат ботів зростає по всьому світу(рис.1.1).

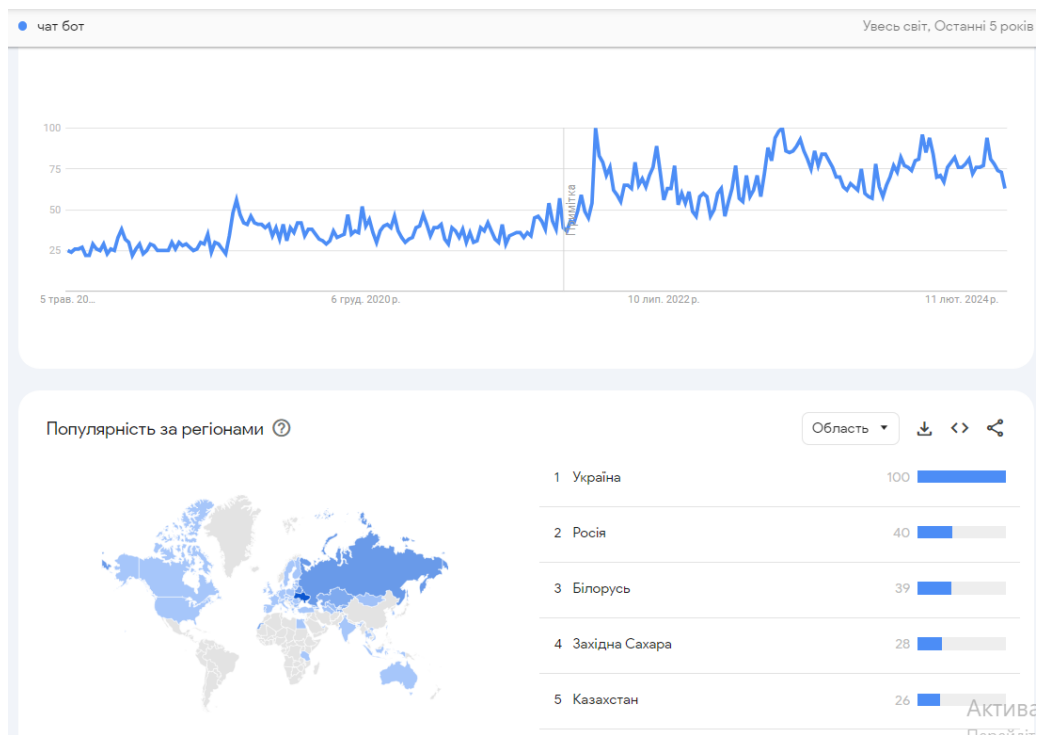


Рисунок 1.1 – Статистика інтересу до чат ботів у світі за останні 5 років

Найбільш популярні чат-боти станом на 2022 рік у IT, стартапів та SaaS-сервісів: за даними SendPulse, на цю нішу припадає більше 25 % усіх чат-ботів. В топ сфер, де активно використовуються чат-боти, потрапили бізнес, маркетинг, онлайн-освіта, та медіа [5]. Цей інструмент впроваджують у свої бізнес-процеси навіть консервативні банки, консалтингові та юридичні компанії (рис.1.2).

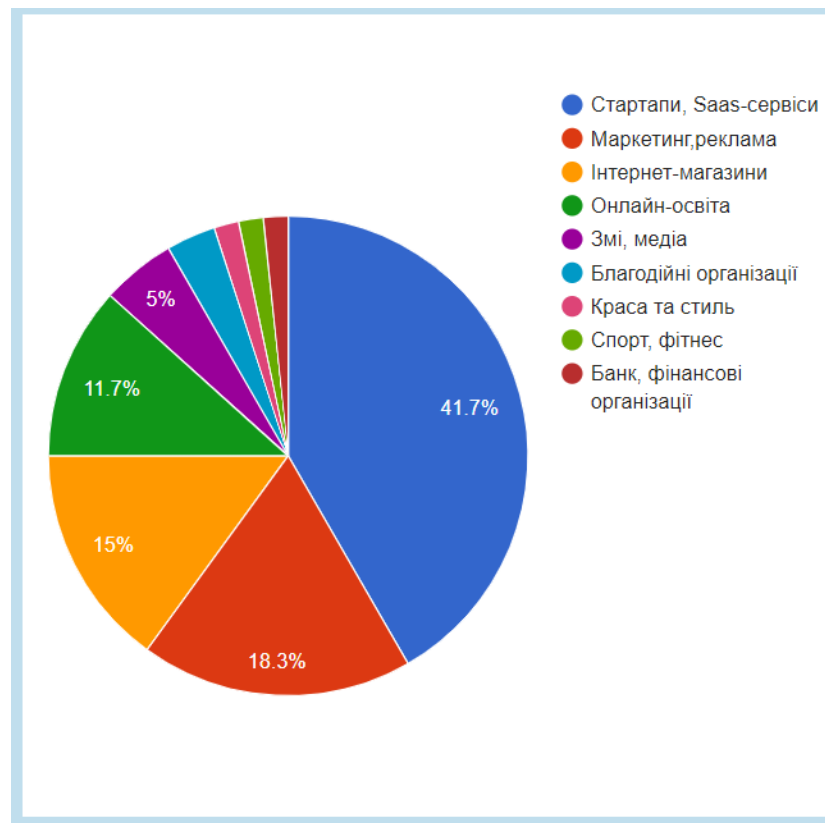


Рисунок 1.2 – Статистика використання чат ботів у різних сферах

Telegram-боти є дуже популярними видами чат-ботів, які використовуються у бізнесі. Вони мають великий ряд переваг: широке охоплення аудиторії завдяки популярності месенджера Telegram, можливість автоматизувати взаємодію з клієнтами, а також інтеграція з іншими сервісами та додатками [6].

Такі боти можуть надавати різноманітні послуги, від відповідей на питання клієнтів до прийому замовлень і надання консультацій. Чат-боти для надання бізнес-порад є дуже потужним інструментом, який допомагає підприємствам отримувати оперативну та персоналізовану інформацію, підвищувати продуктивність, знижувати витрати та покращувати обслуговування клієнтів. [7].

Чат-боти у бізнесі можна класифікувати за декількома критеріями, такими як функціональність, тип взаємодії з користувачем, спосіб реалізації та інші. В таблиці 1.1 приведені основні категорії чат-ботів у бізнесі.

Таблиця 1.1 Основні категорії чат-ботів у бізнесі

Критерії	Тип	Опис
Функціональність	Інформаційні	Надають інформацію про компанію, продукти та послуги, акції, розклад.
	Транзакційні	Дозволяють користувачам здійснювати покупки, бронювати послуги або робити інші операції.
	Консультаційні	Надають консультації щодо продуктів або послуг.
	Підтримка клієнтів	Надають підтримку користувачам, допомагають вирішувати проблеми.
Тип взаємодії з користувачем	Текстові чат-боти	Взаємодіють з користувачем через текстові повідомлення.
	Голосові чат-боти	Взаємодіють з користувачем за допомогою голосового введення.
Спосіб реалізації	Правила та шаблони	Бот використовує заздалегідь визначені правила та шаблони для відповідей
	Машинне навчання	Бот використовує алгоритми машинного навчання для аналізу запитань та надання відповідей.
	Гібридний чат бот	Поєднують штучний інтелект та простий функціонал
Цільова аудиторія	Внутрішні боти	Призначені для внутрішнього використання у компанії, для спілкування з персоналом.
	Зовнішні боти	Призначені для спілкування з клієнтами.

1.2 Telegram-боти

Чат-бот може бути інтегрований в будь-який веб сервіс. Але, чат боти стали найбільш популярними завдяки месенджерам. Це можна пояснити тим, що все більше людей з кожним роком надають перевагу смартфонам [8]. Світова інтернет-аудиторія продовжує зростати— за рік вона виросла на 0,9 %. Найбільше за минулий рік збільшилася кількість користувачів соцмереж — на 5,6 % (рис. 1.3)

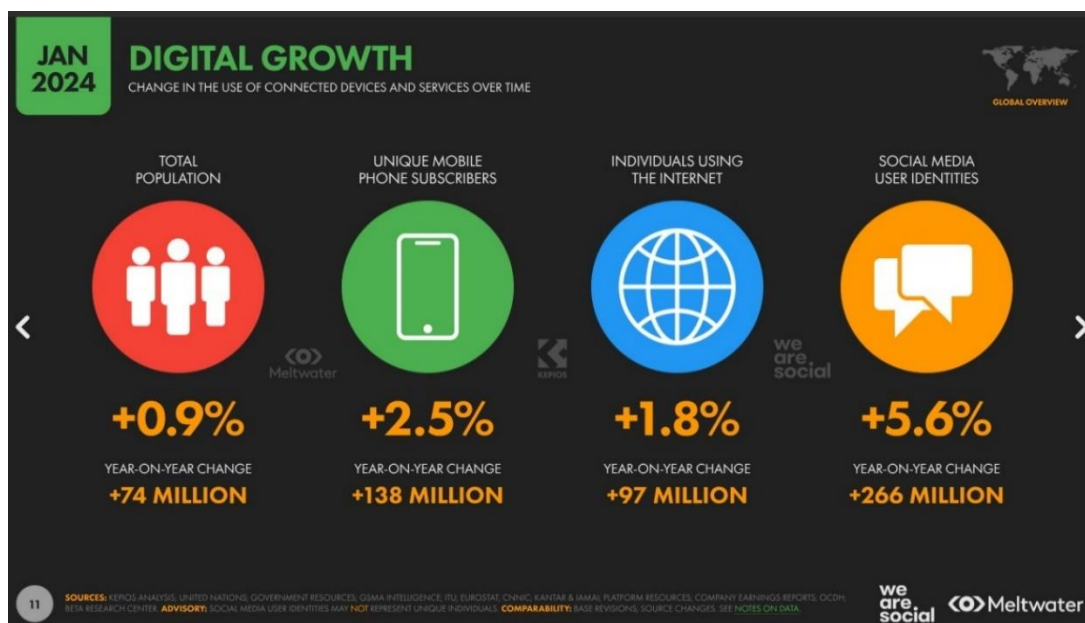


Рисунок 1.3 – Статистика використання смартфонів, інтернету, медіа

Також, зібравши статистичні дані щодо найбільш популярних месенджерів в Україні, за даними дослідження Telemetry, 80 % користувачів з України повідомили, що використовують Telegram для читання новин станом на 2024 рік

За даними досліджень GoogleTrends можна побачити, що статистика запитів щодо Телеграму значно зросла [9] (рис. 1.4)

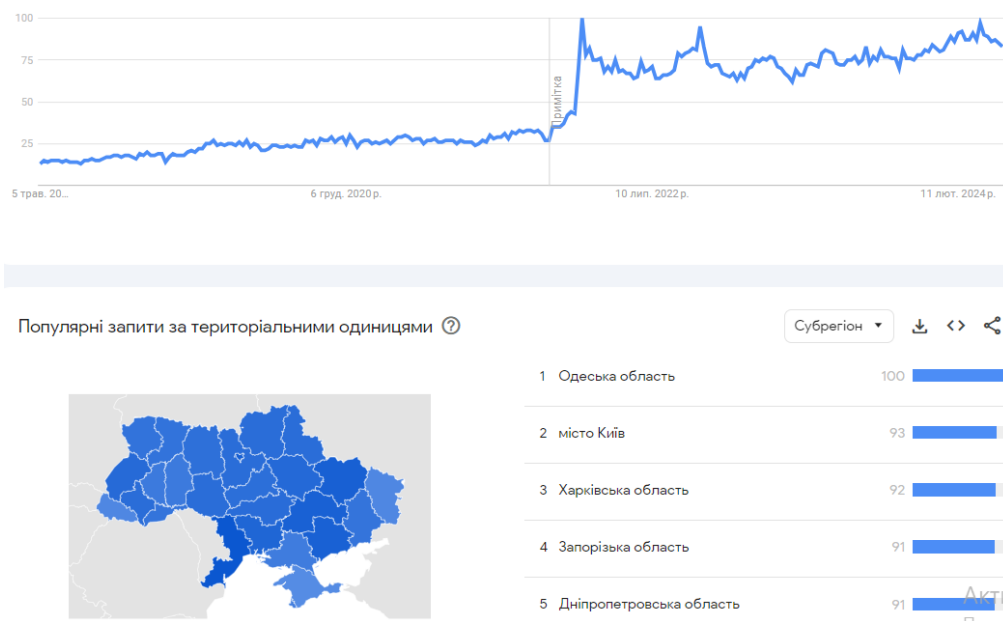


Рисунок 1.4 – Запити на тему Telegram

Telegram - це хмарний месенджер. Він дозволяє користувачам обмінюватися повідомленнями, фотографіями, відео, аудіофайлами та іншими файлами. Telegram відомий своєю швидкістю, безпекою та широкими можливостями для користувачів і розробників [10].

Телеграм набув великої популярності завдяки ряду переваг. З основних це:

- Безпека і конфіденційність (шифрування: Telegram використовує шифрування для захисту повідомлень, а також пропонує опцію секретних чатів, де використовується кінцеве шифрування).
- Швидкість і надійність (хмарний сервіс).
- Висока швидкість: (Повідомлення дуже швидко доставляються, завдяки оптимізації роботи серверів).
- Розширені можливості: (боти: надання API для створення ботів, які можуть автоматизувати різні завдання, надавати інформацію; канали: можливість створювати публічні або приватні канали для широкої аудиторії. Групи: Підтримка груп, і це робить Telegram ідеальним сервісом для спільнот і організацій).

- Доступність (Доступний на різних платформах, включаючи iOS, Android, Windows, macOS, Linux, через веб-інтерфейс).

Боти, створені у Telegram, є одним із різновидів чат-ботів і можуть бути як публічними, так і приватними. Публічні боти доступні для будь-якого користувача та можуть бути відкриті через посилання або знайдені за допомогою пошуку. Приватні чат-боти створюються для обслуговування клієнтів конкретної компанії, таких як відвідувачі кафе або пацієнти клініки. Особливістю приватних ботів є те, що їх не можна знайти у відкритому доступі – підключення до них здійснюється самою компанією для своїх клієнтів [11].

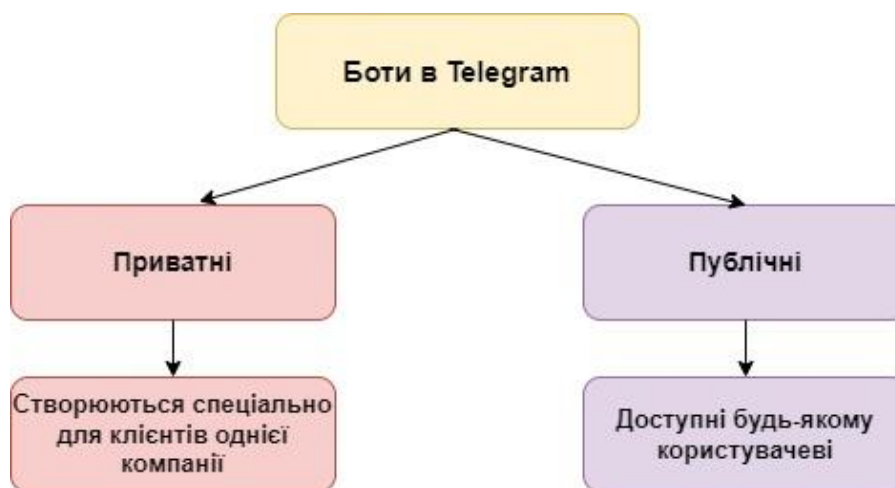


Рисунок 1.5 – Приватні та публічні боти

Чат-бот у Telegram - це зовнішня програма, яку можна використовувати в месенджері Telegram. Користувач взаємодіє з ботом, відправляючи йому повідомлення, команди та вбудовані запити. Для управління ботом використовується спеціальний API, який працює на протоколі HTTPS (рис. 1.6).

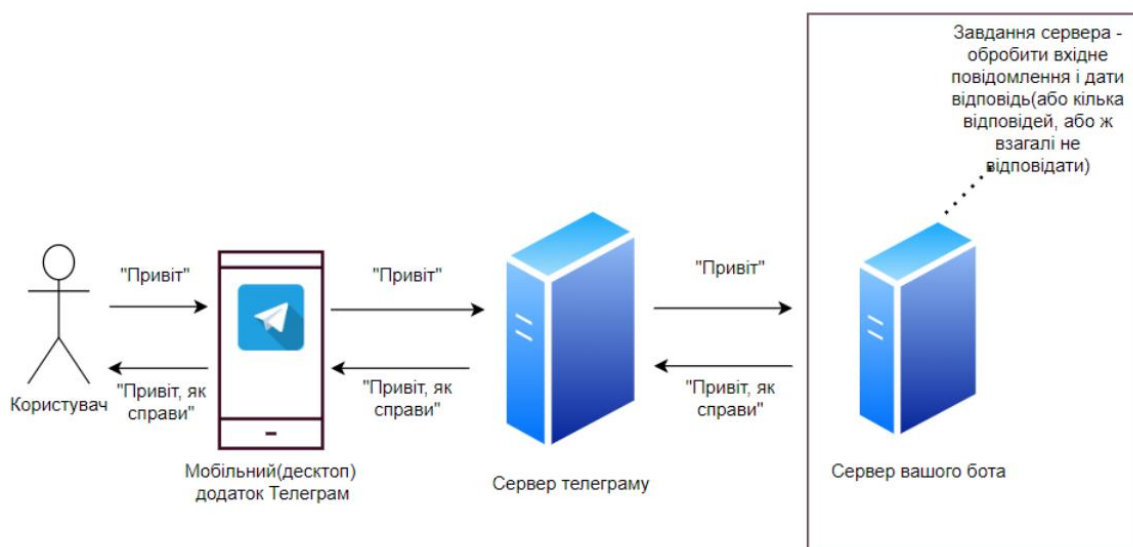


Рисунок 1.6 – Схема роботи чат боту в Телеграмі

Загальна схема роботи телеграм-боту виглядає наступним чином:

- Реєстрація та створення бота: Створення бота та отримання токена для доступу до API.
- Налаштування середовища розробки: Обрання мови програмування та налаштування середовища для розробки чат-бота.
- Програмування функціоналу бота: Програмування функціоналу бота, включаючи обробку повідомлень від користувачів та генерацію відповідей.
- Підключення до Телеграм API: Використання отриманого токена для підключення бота до API Телеграма.
- Розгортання бота: Розгортання бота на обраному сервері або хмарному сервісі.
- Тестування та вдосконалення: Тестування бота, виявлення та виправлення помилок, вдосконалення функціоналу.

Використання бота користувачами: Користувачі можуть звертатися до бота через Телеграм для отримання інформації або виконання певних дій.

Взаємодія телеграм-бота з користувачами відбувається так (рис. 1.7):

- Прийом повідомлення від користувача: Користувач надсилає повідомлення боту через чат у Телеграмі.
- Аналіз повідомлення: Бот отримує повідомлення і аналізує його.
- Обробка запиту: Якщо повідомлення є запитом на інформацію або дію, бот обробляє його, виконуючи відповідні дії.
- Генерація відповіді: Бот генерує відповідь на основі отриманого запиту або виконує необхідну дію.
- Надсилання відповіді користувачеві: Бот надсилає відповідь користувачеві через чат у Телеграмі.
- Продовження або завершення взаємодії: Користувач може продовжити взаємодію з ботом, надсилаючи додаткові запити, або завершити взаємодію.

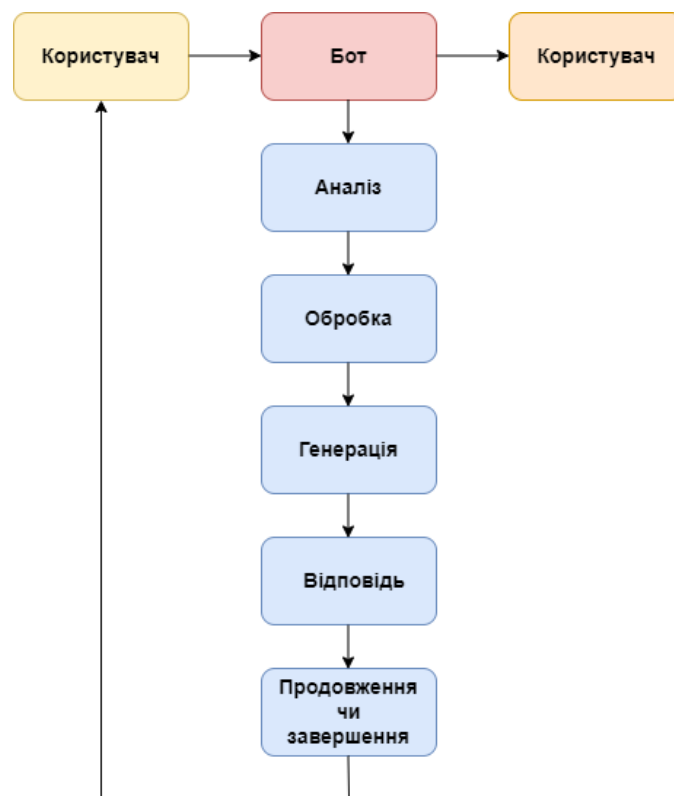


Рисунок 1.7 – Взаємодія Telegram-бота з користувачем.

1.3 Огляд існуючих систем

Огляд існуючих телеграм-ботів. В даному підрозділі розглядаються відомі телеграм-боти, що забезпечують спілкування з клієнтами та оптимізацію бізнес-процесів [12]:

TGStat Bot використовується бізнесом, зокрема маркетологами компаній, для аналізу статистики їхніх каналів у Telegram. Цей бот безперервно сканує месенджер, шукаючи нові канали та працюючи зі старими. Підключивши його до каналу компанії або організації, що надає послуги, бот збиратиме статистичні дані про цільову аудиторію каналу. Це дозволить вивчати спільноти конкурентів.

Щоб розпочати роботу з TGStat Bot (рис. 1.8), потрібно зареєструватися і пройти авторизацію. Потім додати його до адміністраторів каналу, після чого бот автоматично збиратиме статистику (рис. 1.9). Потрібно тільки вказати дні, коли дані мають бути відправлені на електронну пошту.

Функції бота включають:

- Вивантаження даних про підписників каналу.
- Збір та аналіз статистики залучення цільової аудиторії.
- Дослідження індексу цитування.
- Розрахунок охоплення публікацій.

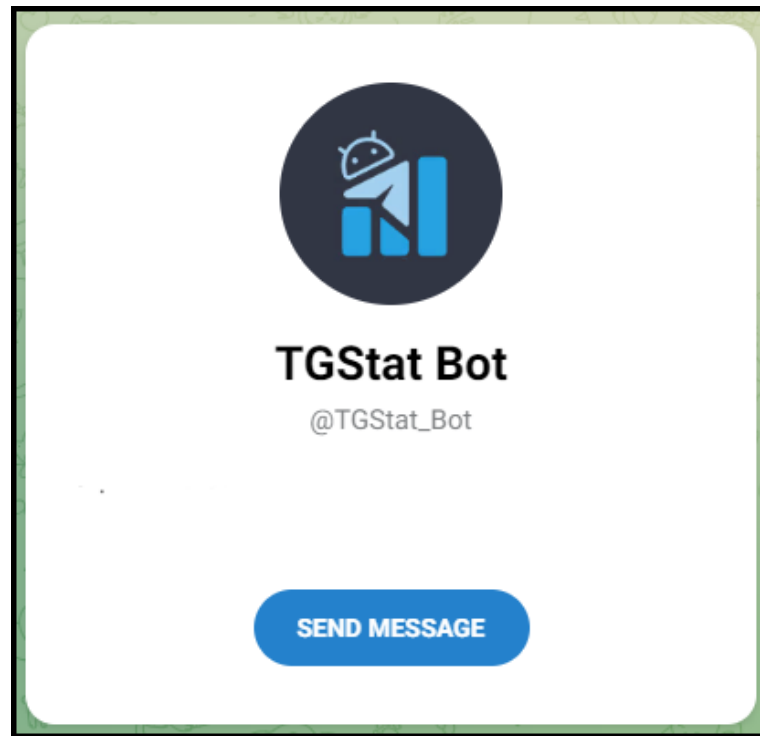


Рисунок 1.8 - інтерфейс TGStat Bot.

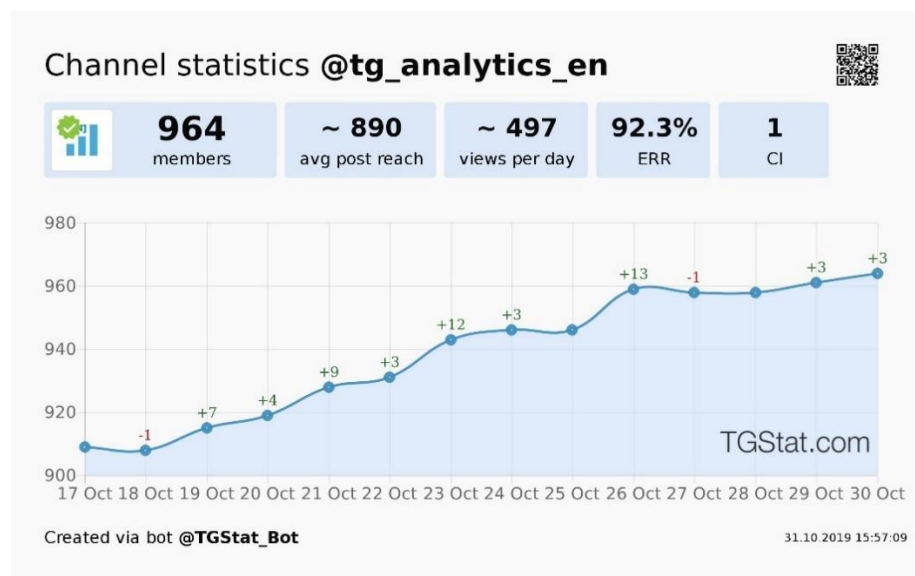


Рисунок 1.9 – Збирання статистики чат-боту

Переваги боту:

- Аналітика каналу: TGStat Bot дозволяє отримувати різноманітну аналітичну інформацію про ваш канал, таку як статистика

підписників, перегляди повідомлень, активність користувачів та інше.

- Зручний інтерфейс: Бот має зручний інтерфейс для отримання аналітичних даних, що дозволяє швидко та легко отримувати необхідну інформацію.
- Широкі можливості: TGStat Bot підтримує багато функцій, які дозволяють детально аналізувати ваш канал і приймати обґрунтовані рішення щодо його просування та управління.

Недоліки:

- Обмежені можливості безкоштовної версії: Безкоштовна версія TGStat Bot має обмежені можливості порівняно з платною версією, що може обмежити доступ до певних аналітичних даних.
- Відсутність деяких функцій: У порівнянні з іншими аналітичними сервісами для Telegram, TGStat Bot не має деяких додаткових функцій та інтеграцій.

Pomodoro Bot – Це функціональний таймер Pomodoro для роботи в спринті та мінімізації відволікань [13]. Допомагає користувачам вимірювати час для кращої продуктивності. Користувачі можуть встановити його на 5, 15 або 25 хвилин(рис.1.10)

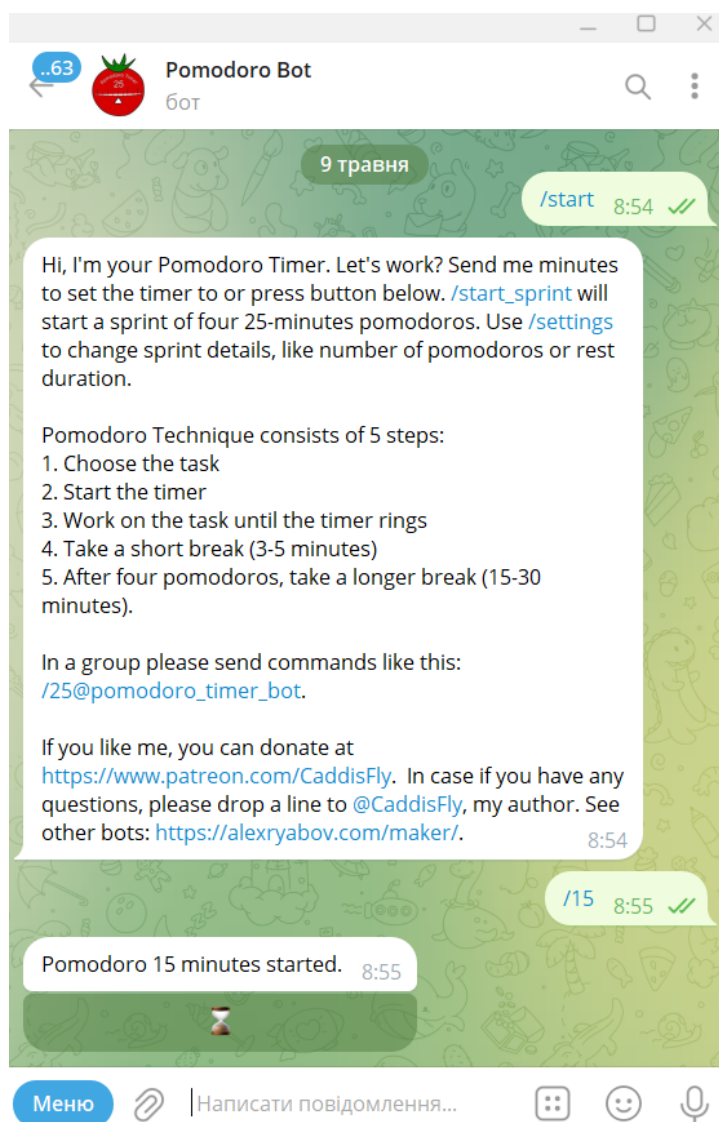


Рисунок 1.10 – Pomodoro Bot

Для користувачів зручність полягає в тому, що таймер працює безпосередньо в месенджері, що робить його важчим для пропуску або ігнорування, у порівнянні зі звичайним таймером [14]. Щодо переваг для бізнесу, надання користувачам інтуїтивно зрозумілих інструментів продуктивності, таких як цей, допомагає компаніям заслужити довіру та лояльність своїх клієнтів.

Недоліки:

- Обмежена функціональність: даний бот обмежений у функціях порівняно з іншими ботами, які пропонують більше різноманітних функцій і можливостей.

- Неадаптивність: Pomodoro бот не є досить адаптивним до унікальних потреб користувачів або ситуацій, що може стати перешкодою для деяких користувачів.

Asos Helper - Asos є одним із найбільших світових роздрібних торговців. Залежно від країни та валюти, які ви оберете під час оформлення замовлення, ціни будуть відрізнятися [15]. Бот допомагає порівнювати ціни на всіх веб-сайтах Asos і вибирати найнижчу ціну на певний продукт (рис. 1.11).

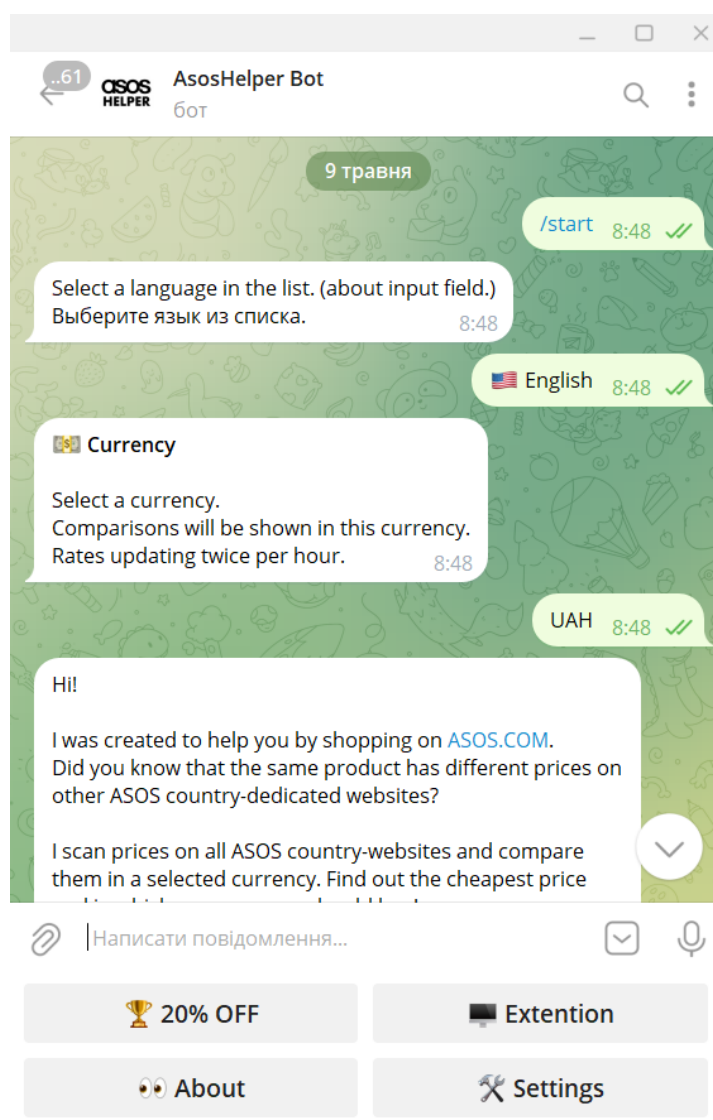


Рисунок 1.11 – інтерфейс Asos Helper Bot

Для користувачів цей чат-бот дарує можливість економити гроші або приймати більш обдумані рішення. Щодо бізнесу, компанії електронної комерції

можуть використовувати цю механіку чат-бота, щоб допомогти своїм клієнтам порівнювати різні аспекти, такі як ціни, бренди, місця або інгредієнти, полегшуючи процес покупок та сприяючи ухваленню обґрунтованих рішень.

Недоліки:

- Обмежена функціональність: Хоча Asos Helper Bot може допомагати з деякими аспектами покупок на Asos, його функціонал може бути обмежений порівняно з офіційним мобільним додатком або веб-сайтом Asos, де доступні більш широкі можливості.
- Недостатня точність інформації: Бот може надавати неточну або застарілу інформацію про товари, акції або наявність товарів на складі, що може призвести до незадоволення користувачів.
- Відсутність можливості огляду товару: Оскільки Asos Helper Bot працює у текстовому форматі, користувачам може бути складно отримати повноцінне враження про товар без можливості перегляду фотографій або відео.

Eddy Travels - це високоякісний бот Telegram для подорожуючих, який співпрацює з Skyscanner, щоб допомогти користувачам знаходити найвигідніші пропозиції на авіаквитки, готелі та місця для відвідування. Просто виберіть пункт призначення або скористайтеся опцією «Візьміть мене куди завгодно», щоб отримати натхнення.

Бот зберігає місцезнаходження, щоб уникнути повторення процесів, і має вбудовані команди для зручного отримання інформації [16]. Користувачі можуть отримати повний обсяг інформації про будь-яке місце призначення безпосередньо в своєму месенджері.

Щодо бізнесу, цей чат-бот Telegram допомагає не лише надавати користувачам цінність, але і залучає безкоштовний реферальний трафік на Skyscanner та інші веб-сайти партнерів [17].

Недоліки:

- Обмежена географія: Eddy Travels може мати обмежену покриття та не охоплювати всі регіони або напрямки подорожей, що може обмежити вибір користувачів.
- Нестача персоналізації: Eddy Travels може не завжди надавати персоналізовані рекомендації або пропозиції, що може знизити задоволення користувачів від використання бота (рис.1.12).

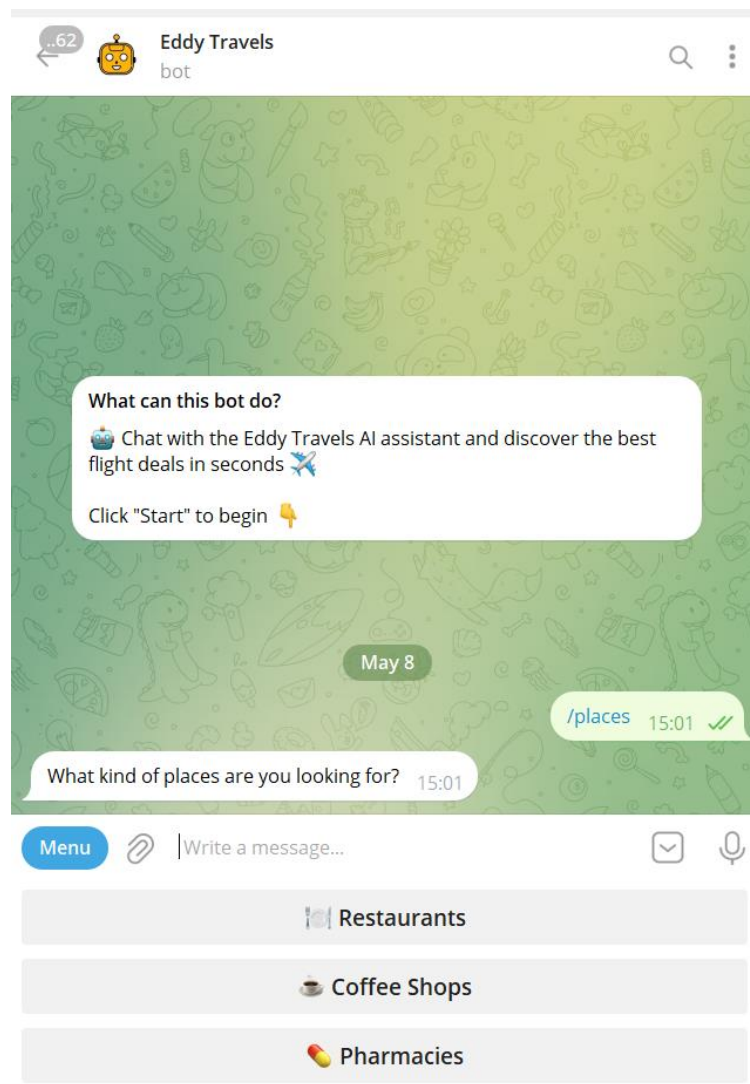


Рисунок 1.12 – інтерфейс EddyTravelsBot

Отже, можна зробити висновок, що існуючі чат-боти можуть бути обмежені у здатності надавати точні та персоналізовані рекомендації через

обмежений функціонал. Не можуть відповісти на всі потреби користувачів, через нестабільну роботу та неправильну обробку повідомлень від користувачів.

Новий чат-бот буде використовувати аналітику соціальних мереж для кращого розуміння потреб користувачів та надання їм найбільш вигідних пропозицій.

Щодо переваг чат-боту:

- Персоналізовані бізнес-поради: На відміну від TGStat Bot, мій чат-бот не лише надає статистичні дані, а також пропонує конкретні поради та рекомендації, адаптовані до потреб користувачів.
- Широкий спектр консультацій: Мій бот надає рекомендації з різних аспектів бізнесу, включаючи маркетинг, управління, фінанси, що значно розширює можливості порівняно з Romodoro Bot та Asos Helper, які обмежені у своїй функціональності.
- Інтерактивна взаємодія: Бот підтримує активну взаємодію з користувачами, що дозволяє оперативно отримувати необхідні консультації та відповіді на запитання, чого не вистачає в Eddy Travels, де рекомендації можуть бути менш персоналізованими.
- Зручність використання: Користувачі можуть легко отримувати поради без необхідності реєстрації або складних налаштувань, що робить бот доступним для широкого кола користувачів.
- Адаптивність до потреб користувачів: Бот здатний адаптувати свої поради в залежності від конкретних потреб та запитів користувачів, що робить його більш універсальним інструментом для бізнес-консультацій порівняно з іншими аналізованими ботами.

Отже, даний телеграм бот вирішує важливу проблему надання персоналізованих та професійних бізнес-порад, що робить його цінним інструментом для підприємців та бізнесменів, які прагнуть покращити свої бізнес-процеси та досягти успіху [18].

1.4 Постановка задачі

Метою кваліфікаційної роботи на здобуття освітнього ступеня бакалавр є створення та розгортання телеграм боту, який надаватиме корисні та професійні поради в сфері бізнесу. Цей бот буде спрямований на надання підтримки та консультацій підприємцям, малому бізнесу та стартапам, які шукають експертні поради щодо стратегій розвитку, маркетингу, фінансів та інших аспектів підприємницької діяльності.

Для досягнення даної мети, були сформульовані такі задачі дослідження:

- 1) Вибір методів розв'язання задачі
- 2) Моделювання процесу використання інформаційно-аналітичної системи
- 3) Розробка архітектури телеграм боту
- 4) Програмна реалізація
- 5) Тестування системи
- 6) Висновки

Дані задачі, які представлені, охоплюють всі етапи від дослідження та планування до реалізації, тестування, запуску телеграм бота, забезпечуючи його ефективну роботу та корисність.

2. ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Вибір мови програмування

Java є однією з найбільш поширених мов програмування, особливо для створення великих систем і платформ, таких як веб-додатки, мобільні додатки та корпоративні платформи. Java відома своєю надійністю, масштабованістю та широкими можливостями застосування [19].

Python також є дуже популярною завдяки своїй легкості у вивченні, багатофункціональності та великій кількості доступних бібліотек і фреймворків.

C++ і C, з іншого боку, використовуються переважно в сферах, де потрібна висока продуктивність та ефективне використання ресурсів, таких як вбудовані системи, операційні системи та графічні додатки [20]. Проте, у порівнянні з Python та Java, вони можуть бути складнішими для освоєння та розробки через свою складну синтаксичну структуру.

Переваги використання Java для розробки чат-бота включають:

- Кросплатформенність: Java дозволяє запускати один і той самий код на різних операційних системах без змін.
- Велика кількість бібліотек: Java надає велику кількість бібліотек і фреймворків, які спрощують розробку чат-ботів, зокрема для роботи з природною мовою, обробки тексту та інших завдань.
- Масштабованість: Java добре підходить для розробки масштабованих систем, що є важливим для чат-ботів, які можуть обслуговувати велику кількість користувачів.
- Безпека: Java має вбудовані механізми безпеки, які допомагають захистити чат-бота від вразливостей та атак.
- Активна спільнота: Java має велику та активну спільноту розробників, що є корисним для отримання допомоги та підтримки.
- Широке застосування: Java використовується в багатьох галузях, що свідчить про її універсальність та ефективність.

Ці переваги роблять Java оптимальним вибором для створення чат-бота, зокрема завдяки її масштабованості, безпеці та великому вибору інструментів для розробки

Ось порівняльна таблиця 2.2 за кількома критеріями для мов програмування Java, Python, C++ та C:

Таблиця 2.1 Порівняльна таблиця мов програмування

Критерій	Java	Python	C++	C
Простота вивчення мови	Середня	Висока	Висока	Висока
Швидкість обміну інформації	Середня	Низька	Висока	Висока
Бібліотеки, фреймворки	Великий вибір	Великий вибір	Великий вибір	Мало
Масштабованість	Висока	Середня	Висока	Висока
Підтримка	Висока	Висока	Висока	Висока

Java є найкращим вибором для написання чат-боту через його високу масштабованість, широкий вибір бібліотек та фреймворків для роботи з текстом та обробки природної мови, а також велику спільноту розробників, що забезпечує активну підтримку та розвиток [21].

Для реалізації чат-боту на Java була обрана бібліотека TelegramBots. Вона є досить популярною серед розробників чат-ботів на Java і має декілька переваг:

- Простота використання: Бібліотека має зрозумілий та простий інтерфейс, що дозволяє швидко розпочати роботу над створенням бота.
- Повна функціональність Telegram API: TelegramBots надає доступ до всіх можливостей Telegram Bot API, що дозволяє реалізувати різноманітні функції в чат-боті.
- Активна підтримка та оновлення: Бібліотека постійно оновлюється та підтримується розробниками, що забезпечує її стабільну та надійну роботу.
- Документація: TelegramBots має добре структуровану та зрозумілу документацію, яка допоможе вам швидко зрозуміти принципи роботи бібліотеки та використати їх у вашому проєкті.

TelegramBots - гарний вибір для створення чат-боту на Java, оскільки вона дозволяє швидко та ефективно реалізувати всі необхідні функції для взаємодії з Telegram API [22]. На рисунку 2.1 можна побачити як правильно додати бібліотеку TelegramBots до проєкту.

```
<dependency>
  <groupId>org.telegram</groupId>
  <artifactId>telegrambots-spring-boot-starter</artifactId>
  <version>5.4.0</version>
</dependency>
```

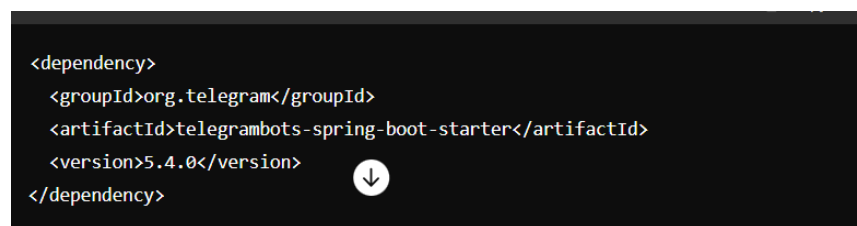
A screenshot of a code editor showing XML dependency code for TelegramBots. The code is enclosed in a dark background with light text. A white circular icon with a downward arrow is positioned to the right of the code block.

Рисунок 2.1 – Додавання бібліотеки TelegramBots до проєкту

Для розробки коду на Java було обрано середовище розробки IntelliJ IDEA. IntelliJ IDEA є потужним інструментом для розробки на Java, який надає широкий спектр функціональності для покращення продуктивності розробників [23].

IntelliJ IDEA має багато корисних функцій, таких як:

- автоматичне завершення коду,

- підтримка рефакторингу,
- відладка

IntelliJ IDEA також підтримує роботу з різними фреймворками та технологіями, що є особливо корисним для написання роботи.

2.2 Штучний інтелект

Прості боти зазвичай базуються на кнопках, що робить їх ідеальними для проведення опитувань, підтримки продажів і виконання будь-яких простих завдань автоматизації процесів, де сценарії взаємодії чітко визначені[24].

Натомість розумні чат-боти, які використовують штучний інтелект, призначені для імітації майже людської взаємодії з клієнтами. Вони здатні вести вільні розмови, розуміти наміри, мову та емоції завдяки технологіям обробки природної мови (NLP, NLU тощо). Natural Language Processing (NLP) – це галузь штучного інтелекту, яка дозволяє комп'ютерам аналізувати та розуміти людську мову [24]. На рисунку 2.5 показано загальну схему роботи Telegram-бота з підтримкою штучного інтелекту.

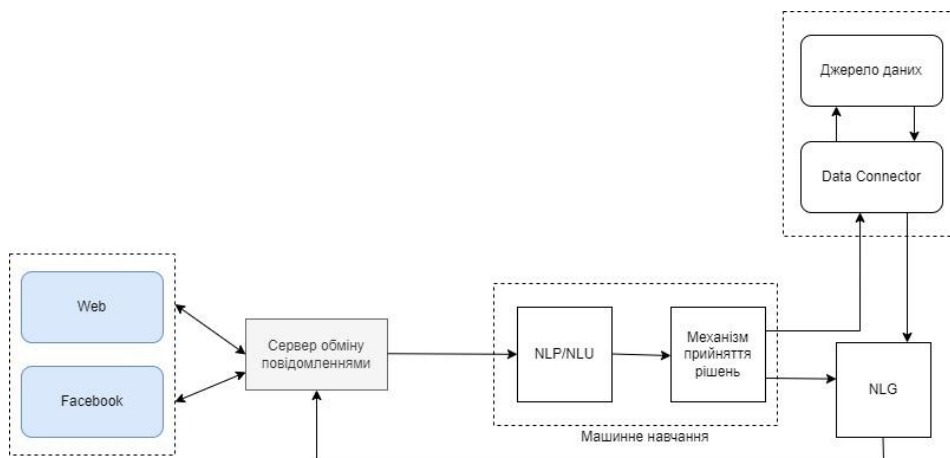


Рисунок 2.5 - Схема роботи Телеграм бота з підтримкою штучного інтелекту

Найбільша різниця між простим чат-ботом і розумним чат-ботом полягає у використанні моделей машинного навчання, що значно розширює можливості бота, дозволяючи йому розпізнавати сотні різних запитів від користувачів [26].

Штучний інтелект (ШІ) відіграє ключову роль у розвитку чат-ботів. Основні способи застосування ШІ в чат-ботах включають:

- Аналіз та розпізнавання природної мови (NLP): Використовуючи NLP, чат-боти можуть аналізувати текстові повідомлення, щоб зрозуміти сенс і наміри користувача.
- Машинне навчання: Чат-боти використовують алгоритми машинного навчання для аналізу історичних даних та оптимізації своїх відповідей, що дозволяє їм навчатися на основі минулих взаємодій і вдосконалювати свої навички.
- Контекстуальна обробка: Чат-боти можуть зберігати контекст попередніх розмов, що дозволяє їм надавати більш релевантні та персоналізовані відповіді, враховуючи історію взаємодії з користувачем.
- Автоматизація завдань: Здійснення певних завдань безпосередньо в чаті, таких як бронювання квитків або запис на прийом.
- Аналіз настроїв: Визначення настрою користувача на основі його поведінки для відповідної реакції на його потреби.

У даній роботі штучний інтелект (ШІ) буде застосовано для створення інтелектуального чат-бота, який надаватиме корисні та професійні поради в сфері бізнесу [26]. Використання ШІ дозволить забезпечити високу якість обслуговування користувачів, надаючи їм інформацію та рекомендації на основі їхніх запитів.

2.3 Моделювання процесу використання Телеграм боту

У цьому розділі буде описано моделювання процесу використання чат-бота, розробленого для надання рекомендацій щодо бізнесу через платформу Telegram. Буде представлено контекстну діаграму IDEF0, діаграму декомпозиції IDEF1, UML діаграму варіантів використання [28].

IDEF0 (Integration Definition for Function Modeling) — це методологія моделювання, яка використовується для опису процесів і систем з точки зору функціональної взаємодії між їхніми компонентами. Вона допомагає визначити і проаналізувати основні функції, а також потоки даних і матеріалів, які проходять через систему [28].

Метою IDEF0 є чітке зображення і розуміння структури процесів або системи, визначення їх основних функцій, входів, виходів, обмежень і механізмів. Це дозволяє виявити взаємодії між різними компонентами системи, що допомагає в аналізі, оптимізації і розробці ефективних процесів.

Основний процес даної системи полягає в наданні рекомендацій користувачам щодо створення власного бізнесу через Телеграм бот, який використовує неймережу для генерації цих рекомендацій. На рисунку 2.6 зображено декомпозицію функціональної моделі нульового рівня

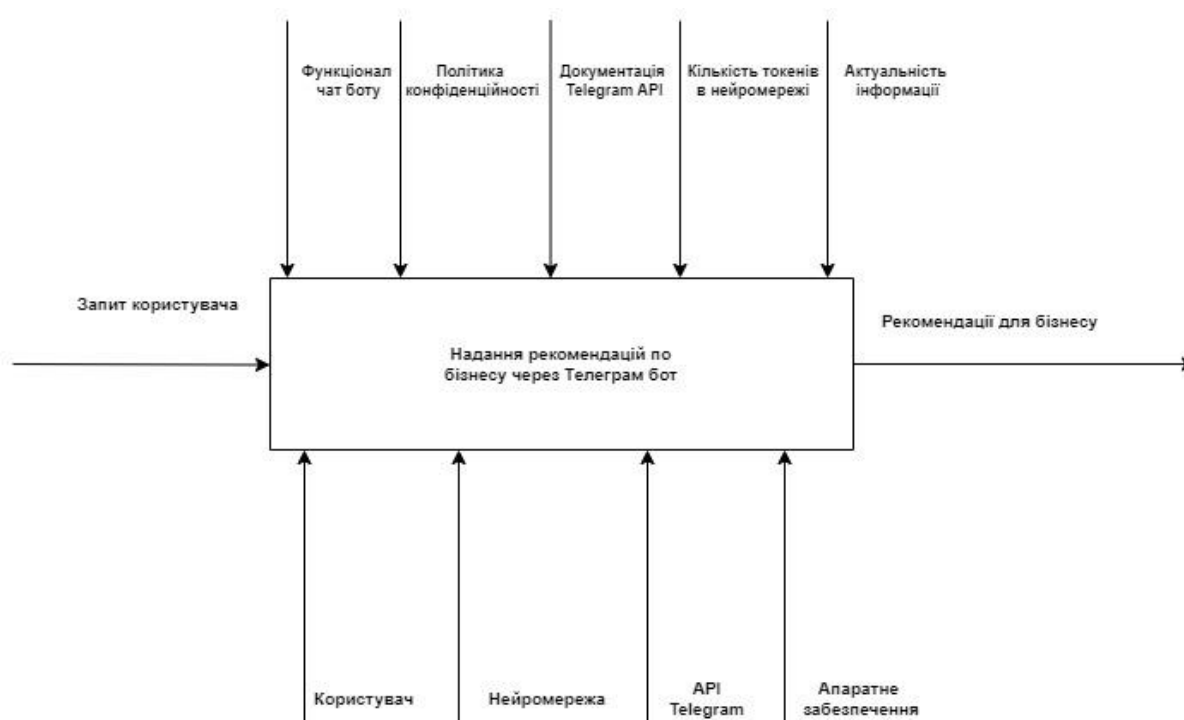


Рисунок 2.6 - Декомпозиція функціональної моделі нульового рівня

Компоненти діаграми:

Вхідні дані:

- Запит користувача: Це запит від користувача, який надсилається через Телеграм бот. Це текстове повідомлення або питання, яке користувач надсилає через чат-бот у Телеграмі. Запит містить інформацію про бізнес, який цікавить користувача, або конкретні питання щодо бізнесу. Наприклад, на вхід до системи потрапляє такий запит від користувача: "Як почати бізнес з продажу одягу онлайн?" Запит користувача є тригером для всієї системи, яка починає обробку даних для формування відповідей.

Вихідні дані:

- Рекомендації для бізнесу: Це результати, які отримує користувач після обробки його запиту - Інформація, яку користувач отримує у відповідь на свій запит. Це можуть бути поради, кроки або аналітичні дані, що допоможуть користувачу в бізнесі. Рекомендації містять поради та інформацію, яка допоможе користувачеві у створенні або розвитку власного бізнесу. Наприклад: "Щоб почати бізнес з продажу одягу онлайн, вам потрібно обрати платформу для продажів (наприклад, Shopify), налаштувати інтернет-магазин, забезпечити якісний контент і маркетинг, а також вибрати надійного постачальника." Роллю у процесі є надання користувачу необхідної інформації для прийняття рішень щодо бізнесу. Це кінцевий результат роботи чат-бота.

Обмеження:

- Функціонал чат-бота: Це набір можливостей та функцій, які повинен мати чат-бот для обробки запитів і надання відповідей. До них

входять можливості розпізнавання запитів, інтеграції з нейромережею та інші функції. Функціонал визначає, які завдання бот може виконувати, а які ні. Це впливає на можливості та якість обробки запитів.

- Політика конфіденційності: Цей набір правил і принципів визначає, як збираються, використовуються та захищаються дані користувачів. Політика конфіденційності забезпечує захист особистої інформації користувачів. Наприклад: Не зберігати особисту інформацію користувачів без їх згоди, використовувати дані лише для надання запитаної послуги. Роль у процесі: Захист конфіденційності користувачів, дотримання правових норм і підвищення довіри до чат-бота.
- Документація Telegram API: Це офіційна документація, яка описує, як використовувати API Telegram для інтеграції з ботом. Вона містить технічні вимоги, обмеження та приклади використання. Приклад: Методи для надсилання повідомлень, отримання оновлень, обробки команд. Роль у процесі: Забезпечує технічну основу для інтеграції чат-бота з платформою Telegram, дозволяє використовувати всі можливості Telegram.
- Кількість токенів у нейромережі (не більше 350): Це обмеження на максимальну кількість токенів (слів або символів), які нейромережа може обробити за один запит. Це обмеження визначає розмір даних, які можуть бути оброблені нейромережею.
- Актуальність інформації (версія нейромережі 3.5 Turbo): Це версія нейромережі, яка використовується для генерації рекомендацій. Вона визначає точність і актуальність даних, які можуть бути використані для відповіді на запит користувача. Отже нейромережа, в даній роботі заснована на моделі GPT-3.5 Turbo. Роль у процесі: Визначає точність і актуальність відповідей, які може надати

нейромережа. Обмеження по актуальності впливають на якість і точність рекомендацій.

Механізми:

- Користувач: Це особа, яка взаємодіє з чат-ботом через Telegram для отримання рекомендацій. Користувач надсилає запити та отримує відповіді від бота.
- Нейромережа: Це штучна інтелектуальна система, яка використовується для обробки запитів та генерації рекомендацій. Нейромережа аналізує вхідні дані та формує відповідь на основі алгоритмів машинного навчання.
- Telegram API: Це інтерфейс програмування, який дозволяє чат-боту взаємодіяти з платформою Telegram. API забезпечує передачу запитів від користувача до сервера бота і повернення відповідей назад користувачу.
- Апаратне забезпечення: Це фізичні ресурси (сервери, процесори, пам'ять), які використовуються для роботи чат-бота та нейромережі. Апаратне забезпечення забезпечує обчислювальну потужність, необхідну для обробки запитів і генерації відповідей.

IDEF1 (Integration Definition for Information Modeling) — це методологія моделювання інформаційних систем, яка використовується для детального опису та аналізу структури інформаційних потоків у процесах. Вона дозволяє побудувати модель даних, що відображає всі етапи обробки інформації в системі.

На рисунку 2.7 зображено діаграму декомпозиції IDEF1

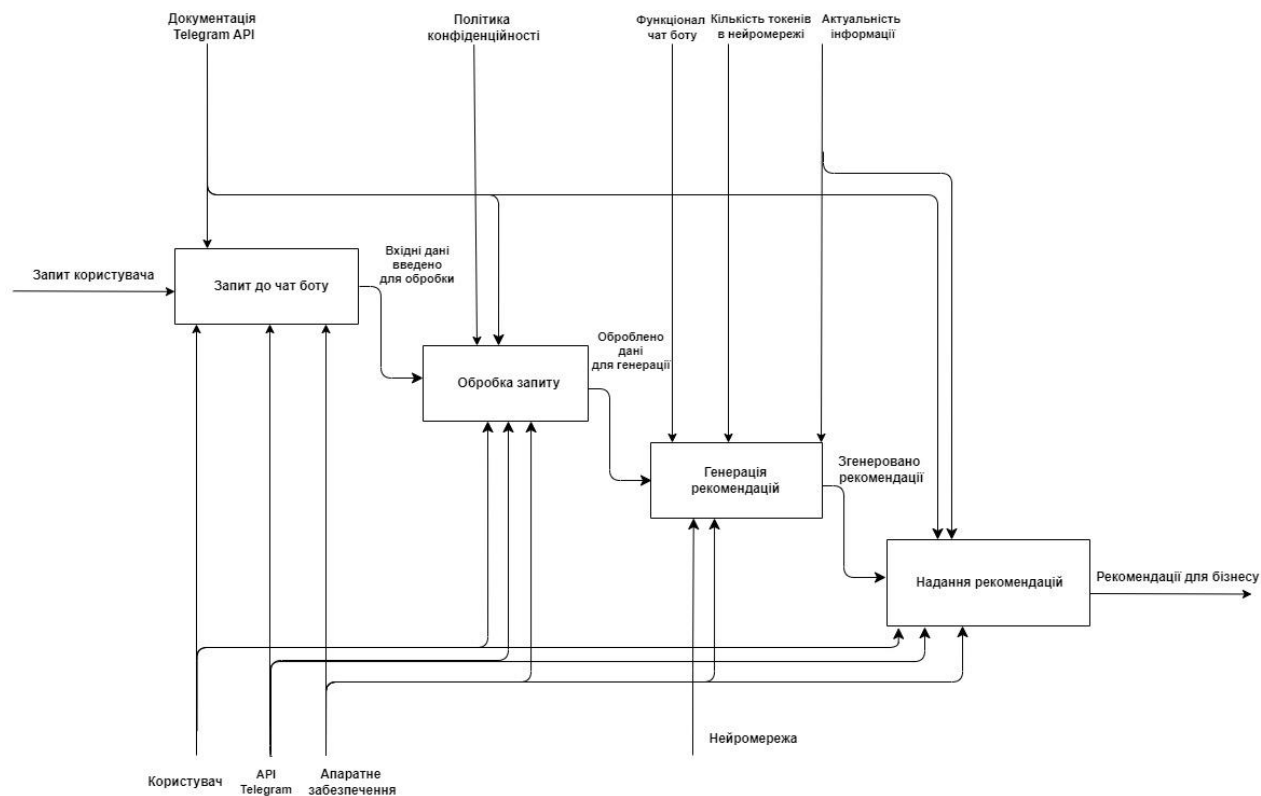


Рисунок 2.7 - діаграма декомпозиції IDEF1

Основні процеси:

1. Запит до чат бота (A1)

- Вхід: Запит користувача: Це повідомлення, яке користувач надсилає через Телеграм бот, включаючи питання або запит на рекомендацію.
- Вихід: Вхідні дані введено для обробки: Це перетворений запит користувача, який готовий до подальшої обробки системою.
- Обмеження:
 - Документація Telegram API: Це офіційна документація, яка регламентує, як використовувати API для взаємодії з ботом. Вона визначає технічні можливості та обмеження.
- Механізми:
 - Користувач: Ініціатор процесу, який надсилає запит.
 - Telegram API: Інтерфейс, що дозволяє прийняти запит користувача та передати його далі.

- Апаратне забезпечення: Фізичні ресурси (сервери, мережеве обладнання), що забезпечують функціонування системи.
- Чому саме такі механізми та обмеження:
 - Користувач необхідний для ініціації запиту.
 - Telegram API потрібен для передачі даних від користувача до системи.
 - Апаратне забезпечення забезпечує технічну підтримку процесу.
 - Документація Telegram API визначає, як правильно використовувати API для обробки запитів.

2. Обробка запиту (в Телеграмі) (A2)

- Вхід: Вхідні дані введено для обробки: Це підготовлений запит, який потрібно обробити перед відправкою до нейромережі.
- Вихід: Оброблено дані для генерації: Це оброблений запит, який готовий для передачі нейромережі для генерації рекомендацій.
- Обмеження:
 - Політика конфіденційності: Правила, що регламентують, як обробляються та захищаються дані користувачів.
 - Документація Telegram API: Визначає, як правильно обробляти та передавати дані через API.
- Механізми:
 - Користувач: Інформує про свої запити, які потребують обробки.
 - Telegram API: Забезпечує передачу оброблених даних до нейромережі.
 - Апаратне забезпечення: Забезпечує ресурси для обробки запитів.
- Чому саме такі механізми та обмеження:

- Користувач визначає початкові дані для обробки.
- Telegram API потрібен для передачі оброблених даних.
- Апаратне забезпечення забезпечує процес обробки даних.
- Політика конфіденційності забезпечує безпеку та відповідність правовим нормам.

3. Генерація рекомендацій (A3)

- Вхід: Оброблено дані для генерації: Це дані, підготовлені для передачі нейромережі для генерації рекомендацій.
- Вихід: Згенеровано рекомендації: Це результати обробки нейромережею, які містять рекомендації для користувача.
- Обмеження:
 - Функціонал чат-бота: Визначає можливості та функції бота.
 - Кількість токенів у нейромережі (не більше 350): Обмеження на максимальну кількість слів або символів для обробки.
 - Актуальність інформації (версія нейромережі 3.5 Turbo): Визначає точність і актуальність рекомендацій.
- Механізми:
 - Апаратне забезпечення: Забезпечує ресурси для роботи нейромережі.
 - Нейромережа: Виконує обробку даних та генерацію рекомендацій.
- Чому саме такі механізми та обмеження:
 - Апаратне забезпечення необхідне для обчислювальних процесів.
 - Нейромережа виконує ключову функцію генерації рекомендацій.
 - Функціонал чат-бота визначає можливості системи.
 - Кількість токенів обмежує обсяг даних для обробки.

- Актуальність інформації забезпечує точність рекомендацій.

4. Надання рекомендацій (A4)

- Вхід: Згенеровано рекомендації: Це результати роботи нейромережі, які потрібно передати користувачу.
- Вихід: Рекомендації для бізнесу: Це фінальні рекомендації, які отримує користувач.
- Обмеження:
 - Актуальність інформації (версія нейромережі 3.5 Turbo): Визначає точність рекомендацій.
 - Документація Telegram API: Визначає технічні аспекти передачі даних.
- Механізми:
 - Апаратне забезпечення: Забезпечує технічні ресурси для передачі даних.
 - Telegram API: Інтерфейс для передачі рекомендацій користувачу.
 - Користувач: Отримує кінцеві рекомендації.
- Чому саме такі механізми та обмеження:
 - Апаратне забезпечення забезпечує передачу даних.
 - Telegram API необхідний для технічної передачі інформації.
 - Користувач є кінцевим отримувачем рекомендацій.
 - Актуальність інформації забезпечує релевантність рекомендацій.
 - Документація Telegram API забезпечує правильність технічної реалізації.

UML (Unified Modeling Language) діаграма варіантів використання описує взаємодії між користувачами (акторами) та системою через різні функціональні

можливості (прецеденти) [29]. Це дозволяє візуалізувати функціональність системи з точки зору користувача, а також визначити, як різні компоненти системи взаємодіють один з одним. На рисунку 2.8 представлена uml діаграма варіантів використання:

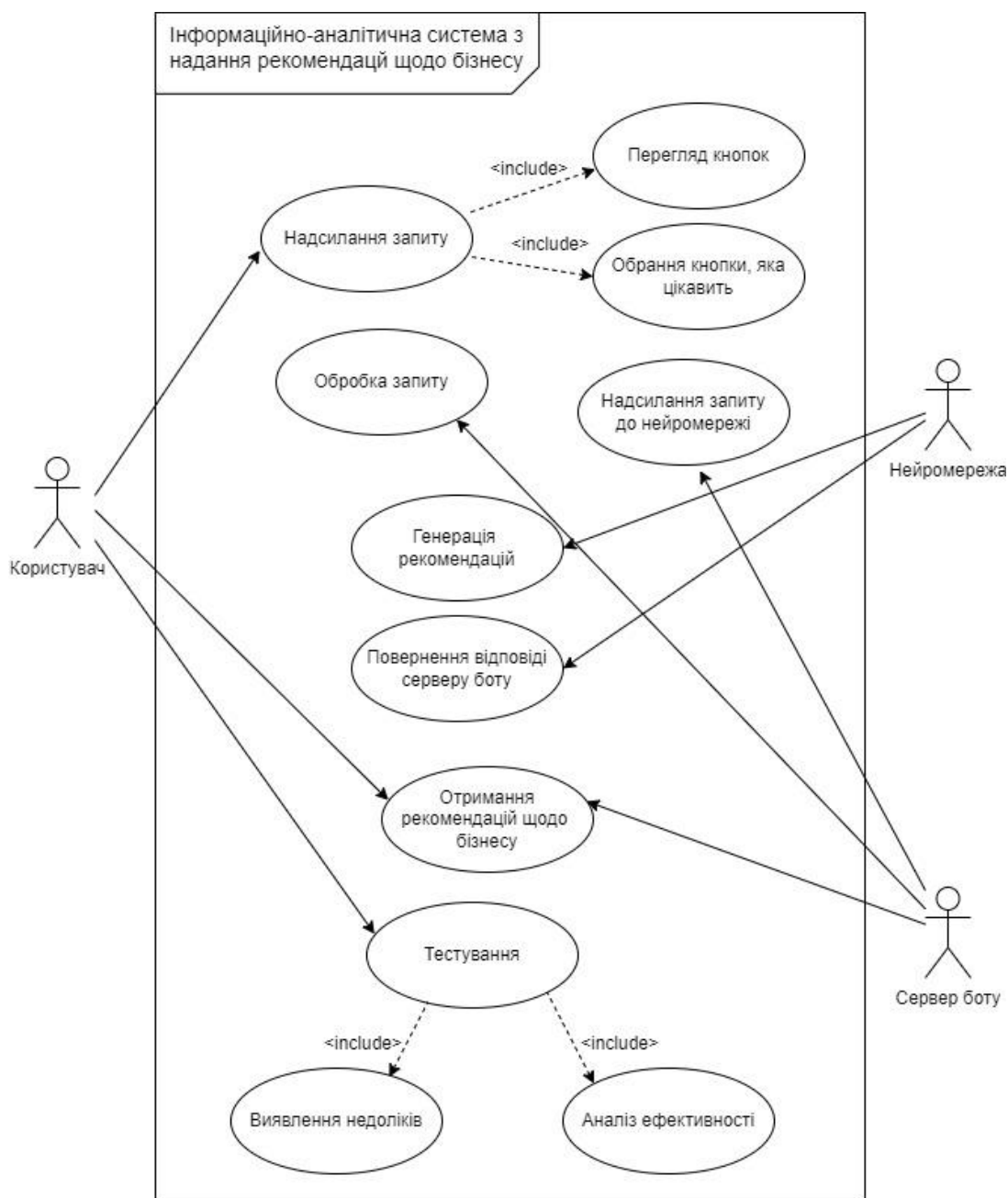


Рисунок 2.8 - UML діаграма варіантів використання

Опис акторів, прецедентів та стрілок

Актори:

- Користувач: Основний користувач системи, який взаємодіє з ботом для отримання рекомендацій щодо бізнесу.
- Нейромережа: Компонент, який відповідає за генерацію рекомендацій на основі запитів, отриманих від користувача через сервер бота.
- Сервер бота: Компонент, який обробляє запити користувачів, передає їх нейромережі і повертає отримані рекомендації користувачу.

Прецеденти:

- Користувач:
 - Надсилання запиту: Процес, під час якого користувач взаємодіє з ботом для відправлення запиту про рекомендації щодо бізнесу.
 - Перегляд кнопок (include): Додаткова функція, яка дозволяє користувачу переглядати доступні кнопки на інтерфейсі бота.
 - Обрання кнопки яка цікавить (include): Додаткова функція, яка дозволяє користувачу обрати конкретну кнопку для отримання необхідної інформації.
 - Отримання рекомендацій щодо бізнесу: Процес, при якому користувач отримує рекомендації від бота після обробки запиту нейромережею.
 - Тестування: Дія, яка включає в себе перевірку роботи бота, виявлення недоліків та аналіз ефективності.
 - Виявлення недоліків (include): Частина процесу тестування, яка включає в себе ідентифікацію проблем в роботі бота.
 - Аналіз ефективності (include): Частина процесу тестування, яка включає в себе оцінку ефективності роботи бота.

- Нейромережа:
 - Генерація рекомендацій: Процес створення рекомендацій на основі запиту, отриманого від серверу бота.
 - Повернення відповідей серверу бота: Процес повернення згенерованих рекомендацій серверу бота для подальшої передачі користувачу.
- Сервер бота:
 - Надсилання запиту до нейромережі: Процес передачі запиту, отриманого від користувача, до нейромережі для генерації рекомендацій.
 - Обробка запиту: Процес обробки запиту користувача перед його передачею нейромережі та після отримання рекомендацій.
 - Отримання рекомендацій щодо бізнесу: Процес отримання рекомендацій від нейромережі та їх передача користувачу.

2.4 Архітектура чат-боту

Архітектура дозволяє чат-боту ефективно взаємодіяти з користувачами, обробляти їхні запитання та надавати відповіді на них[30].



Рисунок 2.9 – Архітектура чат боту

Компоненти Архітектури та їх Завдання:

1. Користувач:

– Завдання:

- Взаємодіяти з чат-ботом через мобільний додаток Телеграм.
- Надсилати запити для отримання рекомендацій щодо бізнесу.
- Отримувати згенеровані рекомендації від чат-бота.

2. Мобільний додаток Телеграм:

– Завдання:

- Забезпечувати інтерфейс для користувача для взаємодії з чат-ботом.
- Надсилати запити користувача на сервер Телеграму.
- Отримувати відповіді від сервера Телеграму та відображати їх користувачу.

3.Сервер Телеграму:

– Завдання:

- Обробляти запити від мобільного додатку Телеграм.
- Пересилати запити до відповідного сервера бота.
- Отримувати відповіді від сервера бота та пересилати їх назад до мобільного додатку.

4.Сервер бота:

– Завдання:

- Отримувати запити від сервера Телеграму.
- Обробляти запити, формуючи їх для подальшої передачі нейромережі.
- Надсилати запити до нейромережі для генерації рекомендацій.
- Отримувати відповіді від нейромережі.
- Форматувати відповіді та надсилати їх назад до сервера Телеграму для передачі користувачу.

5. Нейромережа:

- Завдання:
 - Отримувати запити від сервера бота.
 - Аналізувати запити та генерувати рекомендації на основі отриманих даних.
 - Надсилати згенеровані рекомендації назад до сервера бота.

Взаємодія між компонентами:

- Користувач через мобільний додаток Телеграм надсилає запит на отримання рекомендацій.
- Мобільний додаток Телеграм передає цей запит до сервера Телеграму.
- Сервер Телеграму перенаправляє запит до сервера бота.
- Сервер бота обробляє запит і формує його для подальшої передачі до нейромережі.
- Сервер бота надсилає запит до нейромережі.
- Нейромережа аналізує запит і генерує рекомендації.
- Нейромережа надсилає згенеровані рекомендації назад до сервера бота.
- Сервер бота отримує відповідь від нейромережі, форматує її та надсилає назад до сервера Телеграму.
- Сервер Телеграму передає відповідь до мобільного додатку Телеграм.
- Мобільний додаток Телеграм відображає рекомендації користувачу.

Ця архітектура забезпечує чіткий потік даних від користувача до нейромережі та назад, дозволяючи користувачам легко отримувати необхідні рекомендації через Телеграм бот.

3. ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Реєстрація нового чат бота засобами Телеграм API

Реєстрація нового чат боту

Перший крок у створенні бота – реєстрація нового бота через BotFather. Для цього виконується команда /newbot і далі необхідно слідувати інструкціям для встановлення імені та юзернейму бота.

Після успішної реєстрації буде отримано токен доступу до API, який необхідно зберегти для подальшого використання у додатку.

Налаштування проекту Spring Boot

Для початку виконується створення нового проекту Spring Boot

У конфігураційному файлі application.properties виконується зберігання отриманого токена і юзернейму бота (рис. 3.1)

```
bot.token=YOUR_BOT_TOKEN_HERE
bot.username=YOUR_BOT_USERNAME_HERE
```

Рисунок 3.1 - Зберігання отриманого токена і юзернейму бота

Додавання залежностей для Telegram API:

На даному етапі відбувається додавання всіх необхідних залежностей у pom.xml для підтримки Telegram API (рис. 3.2):

```
<dependency>
  <groupId>org.telegram</groupId>
  <artifactId>telegrambots-spring-boot-starter</artifactId>
  <version>5.4.0</version>
</dependency>
```

Рисунок 3.2 - Додавання всіх необхідних залежностей

Створення основного класу бота:

На даному етапі відбувається створення класу бота MyTelegramBot, який успадковує TelegramLongPollingCommandBot. Цей клас буде відповідальний за обробку всіх повідомлень, що надходять від користувачів (рис. 3.3):

```

@Component
public class MyTelegramBot extends TelegramLongPollingCommandBot {

    private final String username;

    @Autowired
    private StartHandler startHandler;

    @Autowired
    private ChatHandler chatHandler;

    @Override
    public String getBotUsername() {
        return username;
    }

    public MyTelegramBot(@Value("${bot.token}") String botToken,
        @Value("${bot.username}") String username) {
        super(botToken);
        this.username = username;
    }

    @Override
    public void processNonCommandUpdate(Update update) {
        if (update.hasMessage() && update.getMessage().hasText()) {
            String messageText = update.getMessage().getText();
            if (messageText.equals("/start")) {
                startHandler.execute(this, update.getMessage().getFrom(), update.getMessage());
            } else {
                chatHandler.answerGeneration(this, update.getMessage().getFrom(), update.getMessage());
            }
        }
    }
}

```

Рисунок 3.3 - створення класу бота MyTelegramBot, який успадковує TelegramLongPollingCommandBot

3.2 Результати реалізації чат боту

Основні компоненти чат-бота:

1. Клас ChatGPT

Цей клас використовується для взаємодії з API OpenAI для отримання відповідей на запити. Він містить два основних методи: listTokens() і prompts().

listTokens() Цей метод відправляє GET-запит до OpenAI API для отримання списку моделей, доступних для використання (рис. 3.4).

```

1 public static void listTokens() {
2     try {
3         URL url = new URL("https://api.openai.com/v1/models");
4         HttpURLConnection con = (HttpURLConnection) url.openConnection();
5         con.setRequestMethod("GET");
6         con.setRequestProperty("Content-Type", "application/json");
7         con.setRequestProperty("Accept", "application/json");
8         con.setRequestProperty("Authorization", "Bearer " + API_KEY);
9         con.setRequestProperty("OpenAI-Organization", ORG_KEY);
10
11         int responseCode = con.getResponseCode();
12         System.out.println("Response Code : " + responseCode);
13
14         if (responseCode == HttpURLConnection.HTTP_OK) {
15             BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
16             String inputLine;
17             StringBuffer response = new StringBuffer();
18             while ((inputLine = in.readLine()) != null) {
19                 response.append(inputLine);
20             }
21             in.close();
22             System.out.println(response);
23         } else {
24             System.out.println("GET request failed");
25         }
26     } catch (Exception e) {
27         System.out.println(e.getMessage());
28     }
29 }

```

Рисунок 3.4 - Клас ChatGPT

2. Клас ChatHandler

Цей клас обробляє команди бота та генерує відповіді за допомогою методу answerGeneration (рис. 3.4)

```

1 @Component
2 public class ChatHandler extends BotCommand {
3
4     public ChatHandler(@Value("help") String commandIdentifier, @Value(" ") String description) {
5         super(commandIdentifier, description);
6     }
7
8     @Override
9     public void execute(AbsSender absSender, User user, Chat chat, String[] strings) {
10    }
11
12    public void answerGeneration(AbsSender absSender, User user, Chat chat, String strings) {
13        ChatGPT.listTokens();
14        String systemContent = strings + " Дай відповідь у межах 400 токенів";
15
16        String query = String.format("""
17        {
18            "model": "gpt-3.5-turbo",
19            "max_tokens": 500,
20            "messages": [
21                {
22                    "role": "system",
23                    "content": "%s"
24                }
25            ]
26        }
27        """, systemContent);
28        String answer = ChatGPT.prompts(query);
29        try {
30            absSender.execute(new SendMessage(chat.getId().toString(), answer));
31        } catch (TelegramApiException e) {
32            throw new RuntimeException(e);
33        }
34    }
35 }

```

Рисунок 3.5 - Клас ChatGPT

3. Клас StartHandler

Цей клас обробляє команду /start і відправляє користувачу привітальне повідомлення.

```

1  @Component
2  public class StartHandler extends BotCommand {
3
4  public StartHandler(@Value("start") String commandIdentifier, @Value("") String description) {
5      super(commandIdentifier, description);
6  }
7
8  @Override
9  public void execute(AbsSender absSender, User user, Chat chat, String[] strings) {
10     File file = new File("C:\\Users\\lesja\\Desktop\\Annona\\ChartBot\\src\\main\\resources\\generate_a_video_for_me_where_the_robot_is_very_rich_and_has_a_lot.mp4");
11     InputFile inputFile = new InputFile(file);
12     SendAnimation sendAnimation = new SendAnimation(chat.getId().toString(), inputFile);
13
14     String message7 = "Давай ми з тобою трошки познайомимося і я тобі коротенько розповім тобі про те що я жму\u003D\u00E99?!\n"
15     + "\n"
16     + "Я думаю такіж найкращим помічником в сучасному бізнесі\u003D\u00E88!\n"
17     + "\n"
18     + "За допомогою мене ти зможеш знайти абсолютно будь-яку інформацію і матеріал\u003E\u00E1!\n"
19     + "\n"
20     + "Ти зможеш навіть разом зі мною створити власний бізнес план, ну звісно якщо ти в цій справі новачок \u003E\u00D7? Але пам'ятайте ніколи не потрібно боятися і експериментувати \u003E\u00E4!\n"
21     + "\n"
22     + "Я жму пропракувати всі витрати і надходження до твого бізнесу \u003D\u00E4!\n"
23     + "\n"
24     + ".....Повір разом ми зможемо змінити цей світ.....\u003D\u00E88\u003D\u00E88\u003D\u00E88!\n"
25     + "\n"
26     + "Мені вже не терпиться розпочати\u003D\u00E9F!\n"
27     + "\n"
28     + "Вибери просто один із готових варіантів, якщо не знаєш з чого почати \u003D\u00E99";
29
30     SendMessage sendMessage7 = new SendMessage(chat.getId().toString(), message7);
31
32     try {
33         absSender.execute(sendAnimation);
34     } catch (TelegramApiException e) {
35         throw new RuntimeException(e);
36     }
37
38     ReplyKeyboardMarkup keyboardMarkup = new ReplyKeyboardMarkup();
39     List<KeyboardRow> keyboard = new ArrayList<>();
40
41     KeyboardRow row1 = new KeyboardRow();
42     KeyboardRow row2 = new KeyboardRow();
43     KeyboardRow row3 = new KeyboardRow();
44     KeyboardRow row4 = new KeyboardRow();
45     KeyboardRow row5 = new KeyboardRow();
46     KeyboardRow row6 = new KeyboardRow();
47     KeyboardRow row7 = new KeyboardRow();
48     KeyboardRow row8 = new KeyboardRow();
49
50     KeyboardButton button1 = new KeyboardButton("Які показники існують при виборі цінової аудиторії?");
51     KeyboardButton button2 = new KeyboardButton("Які існують неефективні канали комунікації з цільовою аудиторією?");
52     KeyboardButton button3 = new KeyboardButton("Як закрити проблеми цільової аудиторії?");
53     KeyboardButton button4 = new KeyboardButton("Як визначити вашу цільову аудиторію?");
54     KeyboardButton button5 = new KeyboardButton("Як створити власний бізнес-план?");
55     KeyboardButton button6 = new KeyboardButton("Які онлайн інструменти та налаштування потрібні для ефективної роботи бізнес-плану?");
56     KeyboardButton button7 = new KeyboardButton("Яким є бізнес-план для нового старту?");
57     KeyboardButton button8 = new KeyboardButton("Найефективніші бізнес-плани");
58
59     row1.add(button1);
60     row2.add(button2);
61     row3.add(button3);
62     row4.add(button4);
63     row5.add(button5);
64     row6.add(button6);
65     row7.add(button7);
66     row8.add(button8);

```

Рисунок 3.6 - Клас StartHandler

Обробка команд:

Команда /start обробляється в StartHandler, де відправляється привітальне повідомлення і набором кнопок для вибору різних запитів(рис. 3.7). :

```

@Component
public class StartHandler extends BotCommand {

    public StartHandler(@Value("start") String commandIdentifier, @Value("") String description) {
        super(commandIdentifier, description);
    }

    @Override
    public void execute(AbsSender absSender, User user, Chat chat, String[] strings) {
        // Відправка відео
        File file = new File("C:\\Users\\lesja\\Desktop\\Диплом\\ChartBot\\src\\main\\resources\\generate_a_video_for_me_where_the_robot_is_very_rich_and_has_a_lot.mp4");
        InputFile inputFile = new InputFile(file);
        SendAnimation sendAnimation = new SendAnimation(chat.getId().toString(), inputFile);

        try {
            absSender.execute(sendAnimation);
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }

        // Привітальне повідомлення
        String messageText = "Давай ми з тобою трішки познайомимося і я тобі коротенько розповім тобі про те що я можу 🤖\n"
            + "\n"
            + "Я буду твоїм найкращим помічником в сучасному бізнесі 🐱\n"
            + "\n"
            + "За допомогою мене ти зможеш знайти абсолютно будь-яку інформацію і матеріали 📄\n"
            + "\n"
            + "Ти зможеш навіть разом зі мною створити власний бізнес план, ну звісно якщо ти в цій справі новачок 🤔 Але пам'ятайте ніколи не потрібно боятися і експериментувати 🤖\n"
            + "\n"
            + "Я зможу прорахувати всі витрати і надходження до твого бізнесу 📊\n"
            + "\n"
            + ".....Повір разом ми зможемо змінити цей світ..... 🤖🤖🤖\n"
            + "\n"
            + "Мені вже не терпиться розпочати 🤔\n"
            + "\n"
            + "Вибери просто один із готових варіантів, якщо не знаєш з чого почати 🤖";

        SendMessage sendMessage = new SendMessage(chat.getId().toString(), messageText);

        // Налаштування клавіатури
        ReplyKeyboardMarkup keyboardMarkup = new ReplyKeyboardMarkup();
        List<KeyboardRow> keyboard = new ArrayList<>();

        KeyboardRow row1 = new KeyboardRow();
        row1.add(new KeyboardButton("Які показники існують при виборі цільової аудиторії"));
        keyboard.add(row1);
    }
}

```

Рисунок 3.7 - Клас StartHandler

На рисунку 3.8 можна бачити як дане текстове повідомлення відображається у користувача



Рисунок 3.8 – Привітальне повідомлення

Обробка запитів користувачів:

Щоб обробити запити користувачів, виконується налаштування клавіатури з різними варіантами вибору запитів для користувача.:

```
// Налаштування клавіатури
ReplyKeyboardMarkup keyboardMarkup = new ReplyKeyboardMarkup();
List<KeyboardRow> keyboard = new ArrayList<>();

KeyboardRow row1 = new KeyboardRow();
row1.add(new KeyboardButton("Які показники існують при виборі цільової аудиторії?"));
keyboard.add(row1);

KeyboardRow row2 = new KeyboardRow();
row2.add(new KeyboardButton("Які існують найефективніші канали комунікації з цільовою аудиторією?"));
keyboard.add(row2);

KeyboardRow row3 = new KeyboardRow();
row3.add(new KeyboardButton("Як закрити проблеми цільової аудиторії?"));
keyboard.add(row3);

KeyboardRow row4 = new KeyboardRow();
row4.add(new KeyboardButton("Як визначити вашу цільову аудиторію?"));
keyboard.add(row4);

KeyboardRow row5 = new KeyboardRow();
row5.add(new KeyboardButton("Як створити власний бізнес-план?"));
keyboard.add(row5);

KeyboardRow row6 = new KeyboardRow();
row6.add(new KeyboardButton("Які онлайн інструменти та налаштування потрібні для ефективної роботи бізнес плану?"));
keyboard.add(row6);

KeyboardRow row7 = new KeyboardRow();
row7.add(new KeyboardButton("Яким є бізнес-план для нового стартапу?"));
keyboard.add(row7);

KeyboardRow row8 = new KeyboardRow();
row8.add(new KeyboardButton("Найефективніші бізнес-плани"));
keyboard.add(row8);
```

Рисунок 3.9 – Обробка запитів користувачів

Інтеграція з ChatGPT:

Клас ChatGPT відповідає за взаємодію з OpenAI API, включаючи відправку запитів і отримання відповідей. Приклад методу для відправки запиту:


```

public class ChatGPT {
    // ...
    public static String prompts(String inputJson) {
        try {
            URL url = new URL("https://api.openai.com/v1/chat/completions");
            HttpURLConnection con = (HttpURLConnection) url.openConnection();
            con.setRequestMethod("POST");
            con.setRequestProperty("Content-Type", "application/json");
            con.setRequestProperty("Authorization", "Bearer " + API_KEY);
            con.setDoOutput(true);

            try (OutputStream os = con.getOutputStream()) {
                byte[] input = inputJson.getBytes(StandardCharsets.UTF_8);
                os.write(input, 0, input.length);
            }

            if (con.getResponseCode() == HttpURLConnection.HTTP_OK) {
                BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
                StringBuilder response = new StringBuilder();
                String inputLine;
                while ((inputLine = in.readLine()) != null) {
                    response.append(inputLine);
                }
                in.close();
                JSONObject jsonResponse = new JSONObject(response.toString());
                return jsonResponse.getJSONArray("choices").getJSONObject(0).getJSONObject("message").getText();
            } else {
                // Error handling
            }
        } catch (Exception e) {
            // Error handling
        }
        return ("Помилка. Дай відповідь пізніше");
    }
}

```

Рисунок 3.10 – Інтеграція з ChatGPT

3.3 Тестування

Функціональне тестування:

Команда /start:

Перевірка, чи бот правильно відповідає на команду /start, відправляючи привітальне повідомлення і пропонуючи користувачам вибір кнопок для різних запитів (рис. 3.11).



Рисунок 3.11 – Відповідь на команду /start

Запити користувачів:

Перевірка, чи бот правильно обробляє запити користувачів, такі як "Як визначити вашу цільову аудиторію?", "Яка зараз сама мажоринальна ніша?", "Що потрібно для старту бізнесу?" і "Які маркетингові стратегії підходять для стартапів?". Всі ці запити повинні генерувати правильні відповіді за допомогою ChatGPT (рис. 3.12, рис. 3.13).

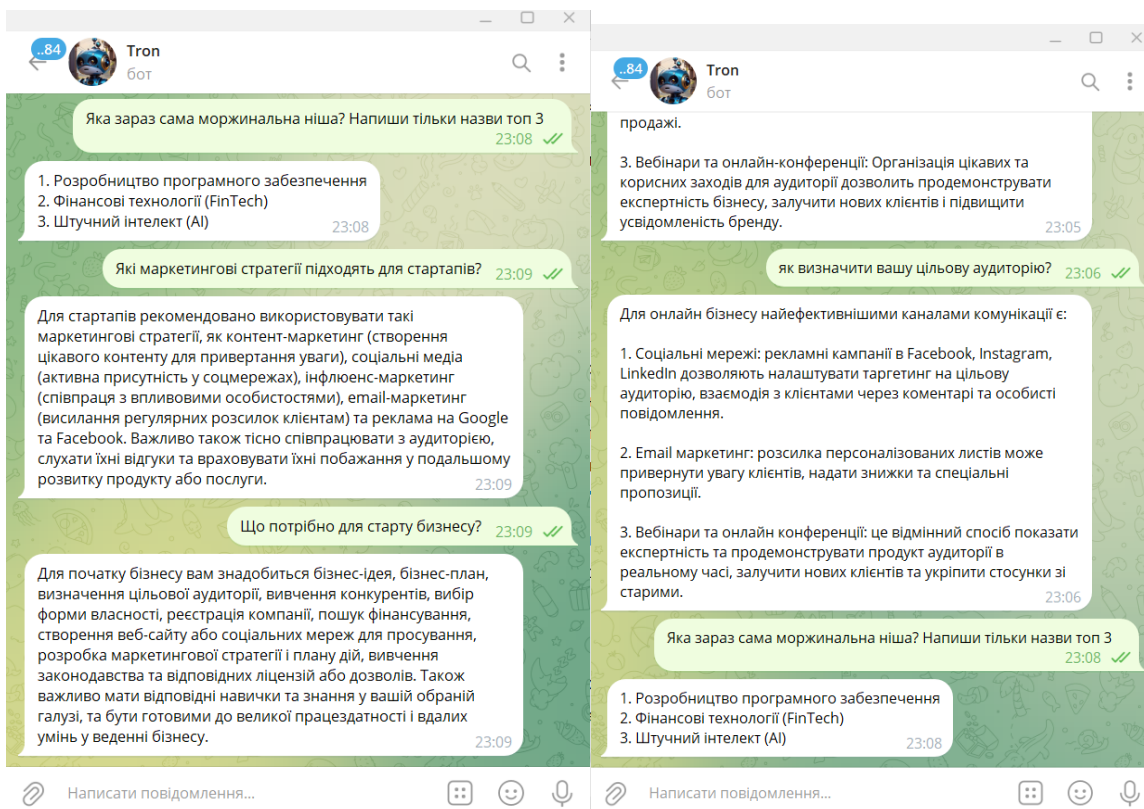


Рисунок 3.12 – Відповідь телеграм боту на запит користувача

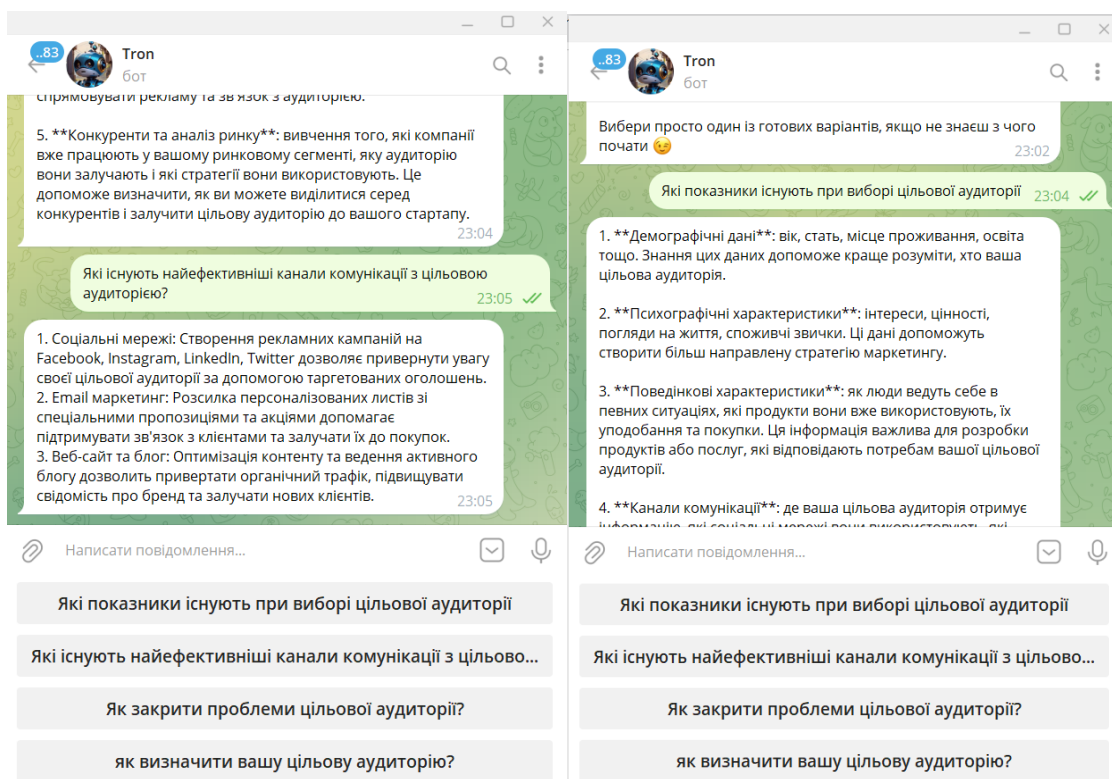


Рисунок 3.13 – Відповідь телеграм боту на запит користувача

Перевірка на конфіденційність даних користувачів:

Забезпечення конфіденційності даних користувачів є важливим аспектом під час розробки та тестування чат-ботів. У даній роботі конфіденційність даних користувачів забезпечена.

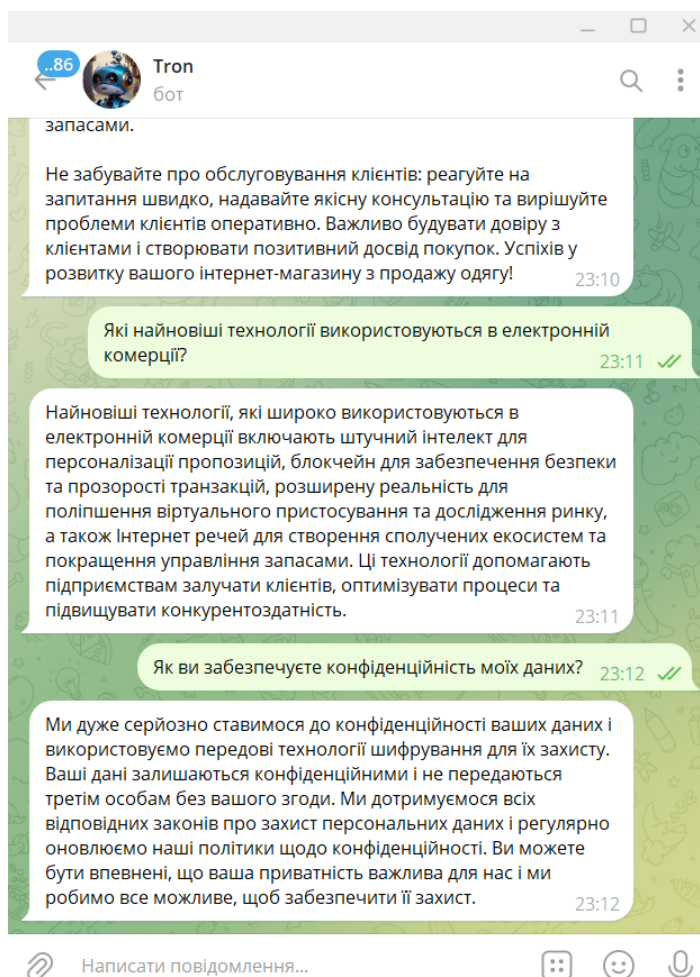


Рисунок 3.14 – Забезпечення конфіденційності даних користувачів

ВИСНОВКИ

В результаті проведеної роботи було створено чат-бот на базі Telegram, який інтегрується з ChatGPT для генерації відповідей на запити користувачів. Основними перевагами бота є його інтерактивність, інтеграція з ChatGPT, автоматизація, зручність використання: Бот може автоматично надавати консультації та рекомендації користувачам, що дозволяє зекономити час і ресурси підприємств та організацій. Завдяки інтеграції з ChatGPT, відповіді бота є більш точними та релевантними, що підвищує задоволеність користувачів. Такий бот може бути використаний у різних сферах, включаючи маркетинг, підтримку клієнтів, навчання та консалтинг. Використання автоматизованого бота дозволяє скоротити витрати на людські ресурси, необхідні для надання консультацій і підтримки.

Відповідно до постановки задачі було виконано:

1. Аналіз предметної області, постановка задач дослідження.
2. Огляд та аналіз існуючих рішень.
3. Моделювання процесу використання та архітектури інформаційно-аналітичної системи.
4. Програмна реалізація чат боту.
5. Тестування програмного забезпечення.
6. Аналіз результатів роботи.

Загалом, створений чат-бот є потужним інструментом, який покращує взаємодію з користувачами, надає релевантні відповіді на їхні запити і сприяє підвищенню ефективності роботи організацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке чат-бот: секрети використання та основні переваги. Блог HelpCrunch. [Електронний ресурс] Режим доступу до ресурсу: <https://helpcrunch.com/blog/uk/shcho-take-chat-bot/> (дата звернення: 01.05.2024).
2. Reach of mobile messenger apps among Android users in Ukraine in June 2020, by platform. [Електронний ресурс] // [statista.com](https://www.statista.com/statistics/1188579/most-popular-messengers-in-ukraine/) – 2020. – Режим доступу: <https://www.statista.com/statistics/1188579/most-popular-messengers-in-ukraine/>(дата звернення: 10.05.2024).
3. Korchevskiy. 7 найкращих ШІ чат-ботів у 2024 році – Claspo.io. Claspo.io. [Електронний ресурс] Режим доступу до ресурсу: <https://claspo.io/ua/blog/7-best-ai-chatbots-in-2024/> (дата звернення: 01.05.2024).
4. Best Telegram Bot Examples to Get Inspired by in 2024. Email and Internet Marketing Blog. [Електронний ресурс] Режим доступу до ресурсу: <https://sendpulse.com/blog/telegram-bot-examples> (date of access: 01.05.2024).
5. Можливості Телеграм для бізнесу [Електронний ресурс] Режим доступу до ресурсу: <https://www.promodo.ua/blog/yak-onlayn-magazinu-prodavati-cherez-telegram> (дата звернення: 01.05.2024).
6. Поліковська Ю. Україна на шостому місці у світі за кількістю телеграм-каналів з понад мільйоном підписників, – Telemetrio. [ms.detector.media](https://ms.detector.media/sotsmerezhi/post/33786/2023-12-19-ukraina-na-shostomu-mistsi-u-sviti-za-kilkisty-telegram-kanaliv-z-ponad-milyonom-pidpysnykiv-telemetrio/). [Електронний ресурс] Режим доступу до ресурсу: <https://ms.detector.media/sotsmerezhi/post/33786/2023-12-19-ukraina-na-shostomu-mistsi-u-sviti-za-kilkisty-telegram-kanaliv-z-ponad-milyonom-pidpysnykiv-telemetrio/> (дата звернення: 03.05.2024).
7. Five Different Types of Chatbot. [Електронний ресурс] // [medium.com](https://medium.com/voiceui/five-different-types-of-chatbot17bb255b23b4) – 2021. Режим доступу до ресурсу: <https://medium.com/voiceui/five-different-types-of-chatbot17bb255b23b4> (дата звернення: 06.05.2024).

8. ТОП 10 Найбільш популярних месенджерів України та світу. [Електронний ресурс] // eo-marketing.com.ua – 2020. – Режим доступу до ресурсу: <https://seomarketing.com.ua/top-10-naibilsh-populiarnykh-mesendzheriv-ukrainy-ta-svitu/> (дата звернення: 03.05.2024).
9. Що таке бот для бізнесу? [Електронний ресурс] Режим доступу до ресурсу: <https://hub.kyivstar.ua/articles/shho-take-chat-bot-dlya-biznesu-use-shho-potribno-znaty-kompaniyam> (дата звернення: 03.05.2024).
10. Тренди масових комунікацій 2022: як бізнес використовує email, SMS, Viber-розсилки та чат-ботів - Блог про email та інтернет-маркетинг. Блог про email та інтернет-маркетинг. [Електронний ресурс] Режим доступу до ресурсу: <https://sendpulse.ua/blog/trends-in-mass-communication> (дата звернення: 04.05.2024)
11. Чат-бот для бізнесу: плюси й мінуси впровадження роботів [Електронний ресурс]. Режим доступу до ресурсу: <https://nikopolnews.net/chat-bot-dlya-biznesupljusi-j-minusi-vprovadzheniya-robotiv/> (дата звернення: 04.05.2024)
12. Кейс: як розробити конструктор Telegram чат-ботів та вирости до 100 000 активних ботів у 2022 [Електронний ресурс]. Режим доступу до ресурсу: <https://sendpulse.ua/blog/how-sendpulse-created-a-telegram-chatbot-builder> (дата звернення: 04.05.2024)
13. 2020 State of Conversational Marketing Drift. [Електронний ресурс] Режим доступу до ресурсу: https://www.drift.com/blog/state-of-conversational-marketing/?utm_source=salesforce&utm_medium=blog (date of access: 01.05.2024).
14. Документація по створенню Telegram-ботів [Електронний ресурс]. Режим доступу до ресурсу: <https://core.telegram.org/bots> (дата звернення: 04.05.2024)
15. About SQL [Електронний ресурс] – 2021. - Режим доступу до ресурсу: <https://www.sqlite.org/about.html> (date of access: 06.05.2024).
16. Телеграм бот. Нотатки. Стаді плани. Архітектура. Вебсокети. [Електронний ресурс] Режим доступу до ресурсу:

<https://drukarnia.com.ua/articles/telegram-bot-notatki-stadi-plani-arkhitektura-chast-3-lodap> (дата звернення: 04.05.2024)

17. JSON. JSON. [Електронний ресурс] Режим доступу до ресурсу: <https://www.json.org/json-uk.html> (date of access: 09.05.2024).

18. GitHub - rubenlagus/TelegramBots: Java library to create bots using Telegram Bots API. GitHub. [Електронний ресурс] Режим доступу до ресурсу: <https://github.com/rubenlagus/TelegramBots> (date of access: 09.05.2024).

19. Hrushak S., Pavlenko C. Advantages of DNS-over-HTTPS over DNS // Computer and information systems and technologies (Kharkiv, Ukraine, 2020). – 2020. (date of access: 09.05.2024).

20. Протоколи HTTP і HTTPS: що це таке, як вони працюють і чим відрізняються [Електронний ресурс] Режим доступу до ресурсу: https://besthosting.ua/ua/http_and_https_protocols.php (дата звернення: 08.05.2024)

21. ChatLab - GPT chatbot for your website - Home. ChatLab - GPT chatbot for your website - Home. [Електронний ресурс] Режим доступу до ресурсу: https://www.chatlab.com/?gad_source=1&gclid=CjwKCAjwgdAyBhBQEiwAXhMxtggtMyKFQ7EvY0xEtEF84pWL2u36kx5sCK4utFBxJhi5fzLrFNvfhhoCiG8QAvD_BwE (date of access: 15.05.2024) .

22. How to Build a Chatbot in 2023 - A Comprehensive Guide. Owlab Group. [Електронний ресурс] Режим доступу до ресурсу: <https://owlab.group/our-blog/post/how-to-build-a-chatbot-in-2023-a-comprehensive-guide> (date of access: 17.05.2024) .

23. How Chatbots For Education Are Being Used In 2023 – AirDroid. AirDroid | Delight Your Multi-Screen Life. [Електронний ресурс] Режим доступу до ресурсу: <https://www.airdroid.com/ai-insights/chatbot-for-education/> (date of access: 18.05.2024)

24. Apps for Good Impact. Apps for Good. [Електронний ресурс] Режим доступу до ресурсу: https://www.appsforgood.org/impact?gad_source=1&gclid=CjwKCAjwgdAyBh

BQEiwAXhMxTkU7tNCO2LPvF9TAWJq5lrZcW_MuhbrJvJoElC_j6GRQ4egABUS
k1hoC4twQAvD_BwE (date of access: 07.05.2024) .

25. SHS Web of Conferences. [Электронный ресурс] Режим доступа до ресурсу: https://www.shsconferences.org/articles/shsconf/pdf/2023/05/shsconf_ictl2023_01002.pdf (date of access: 07.05.2024) .

26. Artificial Intelligence Chatbot Platform: AI Chatbot Platform for Educational Recommendations in Higher Education [Электронный ресурс] Режим доступа до ресурсу: <https://www.ijiet.org/vol14/IJIEET-V14N1-2021.pdf> (date of access: 07.05.2024)

27. University of Tartu guidelines for using AI chatbots for teaching and studies [Электронный ресурс] Режим доступа до ресурсу: https://ut.ee/sites/default/files/2023-05/university_of_tartu_guidelines_for_using_ai_chatbots_for_teaching_and_studies_28_april_2023_pdf.pdf (date of access: 07.05.2024)

28. Chatbot Positive Design to Facilitate Referencing Skills and Improve Digital Well-Being [Электронный ресурс] Режим доступа до ресурсу: <https://online-journals.org/index.php/i-jim/article/view/38395/13177> (date of access: 07.05.2024)

29. ChatEd: A Chatbot Leveraging ChatGPT for an Enhanced Learning Experience in Higher Education [Электронный ресурс] Режим доступа до ресурсу: <https://arxiv.org/pdf/2401.00052> (date of access: 07.05.2024)

30. Exploring the Trend and Potential Distribution of Chatbot in Education: A Systematic Review [Электронный ресурс] Режим доступа до ресурсу: <https://www.ijiet.org/vol13/IJIEET-V13N3-1834.pdf> (date of access: 07.05.2024)

Додаток А. Програмна реалізація

Логіка чат боту

```
package com.example.chartbot.handler;

import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;

public class ChatGPT {

    // Встановіть ваші ключі API та організаційні ключі тут
    private static final String API_KEY = "sk-proj-
vXZTymX1iKpvJZjw7WAXt3BlbkFJcrestKnzoeGorhMZkeO5";
    private static final String ORG_KEY = "org-YMeITlqudAQLUTSARjXzOXsJ";

    public static void listTokens() {
        try {
            // URL для отримання моделей
            URL url = new URL("https://api.openai.com/v1/models");
            HttpURLConnection con = (HttpURLConnection) url.openConnection();
            con.setRequestMethod("GET");
            con.setRequestProperty("Content-Type", "application/json");
            con.setRequestProperty("Accept", "application/json");
            con.setRequestProperty("Authorization", "Bearer " + API_KEY);
            con.setRequestProperty("OpenAI-Organization", ORG_KEY);

            int responseCode = con.getResponseCode();
            System.out.println("Response Code : " + responseCode);

            if (responseCode == HttpURLConnection.HTTP_OK) {
```

```

        BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();
        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();
        System.out.println(response);
    } else {
        System.out.println("GET request failed");
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}
}

```

```

public static String prompts(String inputJson) {
    try {
        URL url = new URL("https://api.openai.com/v1/chat/completions");
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("POST");
        con.setRequestProperty("Content-Type", "application/json");
        con.setRequestProperty("Accept", "application/json");
        con.setRequestProperty("Authorization", "Bearer " + API_KEY);
        con.setRequestProperty("OpenAI-Organization", ORG_KEY);
        con.setDoOutput(true);

        String jsonString = inputJson;

        try (OutputStream os = con.getOutputStream()) {
            byte[] input = jsonString.getBytes(StandardCharsets.UTF_8);
            os.write(input, 0, input.length);
        }

        int responseCode = con.getResponseCode();
        //System.out.println("Response Code : " + responseCode);

        if (responseCode == HttpURLConnection.HTTP_OK) {

```

```

        BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuilder response = new StringBuilder();
        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        // Розпарсити JSON відповідь
        JSONObject jsonResponse = new JSONObject(response.toString());
        String messageContent = jsonResponse
            .getJSONArray("choices")
            .getJSONObject(0)
            .getJSONObject("message")
            .getString("content");
        return (messageContent.trim());
    } else {
        System.out.println("POST request failed");
        BufferedReader in = new BufferedReader(new
InputStreamReader(con.getErrorStream()));
        String inputLine;
        StringBuilder response = new StringBuilder();
        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();
        System.out.println(response);
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
return ("Помилка. Запитай пізніше");
}
}

```

Взаємодія з чатом

```
package com.example.chartbot.handler;
```

```

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import
org.telegram.telegrambots.extensions.bots.commandbot.commands.BotCommand;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Chat;
import org.telegram.telegrambots.meta.api.objects.User;
import org.telegram.telegrambots.meta.bots.AbsSender;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;

@Component
public class ChatHandler extends BotCommand {

    public ChatHandler(@Value("help") String commandIdentifier, @Value(" ")
String description) {
        super(commandIdentifier, description);
    }

    @Override
    public void execute(AbsSender absSender, User user, Chat chat, String[]
strings) {
    }

    public void answerGeneration(AbsSender absSender, User user, Chat chat,
String strings) {
        ChatGPT.listTokens();
        String systemContent = strings + "Дай відповідь у менше між 400 токенів";

        String query = String.format("""
            {
                "model": "gpt-3.5-turbo",
                "max_tokens": 500,
                "messages": [
                    {
                        "role": "system",
                        "content": "%s"
                    }
                ]
            }
        """, systemContent);
    }
}

```

```

    }
    "", systemContent);
String answer = ChatGPT.prompts(query);
try {
    absSender.execute(new SendMessage(chat.getId().toString(), answer));
} catch (TelegramApiException e) {
    throw new RuntimeException(e);
}
}
}
}

```

Розмова з чатом:

```

package com.example.chartbot.handler;
import java.io.File;
import java.util.ArrayList;
import java.util.List;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import
org.telegram.telegrambots.extensions.bots.commandbot.commands.BotCommand;
import org.telegram.telegrambots.meta.api.methods.send.SendAnimation;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Chat;
import org.telegram.telegrambots.meta.api.objects.InputFile;
import org.telegram.telegrambots.meta.api.objects.User;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardButton;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;
import org.telegram.telegrambots.meta.bots.AbsSender;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;

@Component
public class StartHandler extends BotCommand {

```

```

    public StartHandler(@Value("start") String commandIdentifier, @Value("")
String description) {
        super(commandIdentifier, description);
    }

    @Override
    public void execute(AbsSender absSender, User user, Chat chat, String[]
strings) {

        File file = new
File("C:\\Users\\lesja\\Desktop\\Диплом\\ChartBot\\src\\main\\resources\\generate_a
_video_for_me_where_the_robot_is_very_rich_and_has_a_lot.mp4");
        InputFile inputFile = new InputFile(file);
        SendAnimation sendAnimation = new SendAnimation(chat.getId().toString(),
inputFile);

        String message7 = "Давай ми з тобою трішки познайомимося і я тобі коротенько
розповім тобі про те що я можу\uD83D\uDE09?\n"
            + "\n"
            + "Я буду твоїм найкращим помічником в сучасному бізнесі\uD83D\uDE08\n"
            + "\n"
            + "За допомогою мене ти зможеш знайти абсолютно будь-яку інформацію і
матеріали\uD83E\uDEE1\n"
            + "\n"
            + "Ти зможеш навіть разом зі мною створити власний бізнес план, ну
звісно якщо ти в цій справі новачок \uD83E\uDD72 Але пам'ятайте ніколи не потрібно
боятися і експериментувати \uD83D\uDE43\n"
            + "\n"
            + "Я зможу прорахувати всі витрати і надходження до твого бізнесу
\uD83D\uDE43\n"
            + "\n"
            + ".....Повір разом ми зможемо змінити цей
світ.....\uD83D\uDE08\uD83D\uDE08\uD83D\uDE08\n"
            + "\n"
            + "Мені вже не терпиться розпочати\uD83D\uDE0F\n"
            + "\n"
            + "Вибери просто один із готових варіантів, якщо не знаєш з чого почати
\uD83D\uDE09";

```

```
SendMessage sendMessage7 = new SendMessage(chat.getId().toString(),
message7);

try {
    absSender.execute(sendAnimation);
} catch (TelegramApiException e) {
    throw new RuntimeException(e);
}

ReplyKeyboardMarkup keyboardMarkup = new ReplyKeyboardMarkup();
List<KeyboardRow> keyboard = new ArrayList<>();

KeyboardRow row1 = new KeyboardRow();
KeyboardRow row2 = new KeyboardRow();
KeyboardRow row3 = new KeyboardRow();
KeyboardRow row4 = new KeyboardRow();
KeyboardRow row5 = new KeyboardRow();
KeyboardRow row6 = new KeyboardRow();
KeyboardRow row7 = new KeyboardRow();
KeyboardRow row8 = new KeyboardRow();

KeyboardButton button1 = new KeyboardButton("Які показники існують при
виборі цільової аудиторії");
KeyboardButton button2 = new KeyboardButton("Які існують найефективніші
канали комунікації з цільовою аудиторією?");
KeyboardButton button3 = new KeyboardButton("Як закрити проблеми цільової
аудиторії?");
KeyboardButton button4 = new KeyboardButton("Як визначити вашу цільову
аудиторію?");
KeyboardButton button5 = new KeyboardButton(
    "Як створити власний бізнес-план?");
KeyboardButton button6 = new KeyboardButton(
    "Які онлайн інструменти та налаштування потрібні для ефективної роботи
бізнес плану");
KeyboardButton button7 = new KeyboardButton(
    "Яким є бізнес-план для нового стартапу?");
KeyboardButton button8 = new KeyboardButton(
    "Найефективніші бізнес-плани");
```



```

keyboardMarkup.setResizeKeyboard(true);

    row1.add(button1);
    row2.add(button2);
    row3.add(button3);
    row4.add(button4);
    row5.add(button5);
    row6.add(button6);
    row7.add(button7);
    row8.add(button8);

    keyboard.add(row1);
    keyboard.add(row2);
    keyboard.add(row3);
    keyboard.add(row4);
    keyboard.add(row5);
    keyboard.add(row6);
    keyboard.add(row7);
    keyboard.add(row8);

    keyboardMarkup.setKeyboard(keyboard);
    sendMessage7.setReplyMarkup(keyboardMarkup);

    try {
        absSender.execute(sendMessage7);
    } catch (TelegramApiException e) {
        throw new RuntimeException(e);
    }
}
}
}

```

Кнопки та відповідь на питання:

```

package com.example.chartbot.query;
public class QueryMessage {

    public static String query1 = ""

```

```

    {
        "model": "gpt-3.5-turbo",
        "max_tokens": 600,
        "messages": [
            {
                "role": "system",
                "content": "При відкритті нового стартапу, якими показниками
варто керуватися при визначенні цільової аудиторії? Топ 5"
            }
        ]
    }
    """;
    public static String query2 = """
    {
        "model": "gpt-3.5-turbo",
        "max_tokens": 500,
        "messages": [
            {
                "role": "system",
                "content": "Які канали комунікації найефективніше дозволять
зв'язатися з цільовою аудиторією, якщо це онлайн бізнес ? З основних. Опишіть
стратегію коротко у 350 токенах "
            }
        ]
    }
    """;
    public static String query3 = """
    {
        "model": "gpt-3.5-turbo",
        "max_tokens": 400,
        "messages": [
            {
                "role": "system",
                "content": "Які основні проблеми потрібно вирішити новому
бізнесу , щоб закрити потреби цільової аудиторії? З основних .Опишіть коротко у 350
токенах"
            }
        ]
    }
    """;

```

```
""";
public static String query4 = ""
{
    "model": "gpt-3.5-turbo",
    "max_tokens": 400,
    "messages": [
        {
            "role": "system",
            "content": "Загальні рекомендації як визначити цільову
аудиторію онлайн бізнесу. Топ 5. Дай коротку відповідь у 350 токенах"
        }
    ]
}
""";

public static String query5 = ""
{
    "model": "gpt-3.5-turbo",
    "max_tokens": 400,
    "messages": [
        {
            "role": "system",
            "content": "Як створити власний бізнес план для нового онлайн
бізнесу? Задай план з 5 пунктів. Дай коротку відповідь у 350 токенах"
        }
    ]
}
""";

public static String query6 = ""
{
    "model": "gpt-3.5-turbo",
    "max_tokens": 400,
    "messages": [
        {
            "role": "system",
            "content": "Які загальні онлайн інструменти та налаштування
потрібні для ефективної роботи бізнес плану? Дай коротку відповідь у 350 токенах"
        }
    ]
}
```

```

    ]
  }
  """;

public static String query7 = """
{
  "model": "gpt-3.5-turbo",
  "max_tokens": 400,
  "messages": [
    {
      "role": "system",
      "content": "який загальний бізнес-план для нового онлайн
стартапу? опиши план з 6 пунктів. дай коротку відповідь у 350 токенах"
    }
  ]
}
""";

public static String query8 = """
{
  "model": "gpt-3.5-turbo",
  "max_tokens": 400,
  "messages": [
    {
      "role": "system",
      "content": " які існують бізнес - плани для успішного
просування продукту на ринку? топ 3 дай коротку відповідь у 350 токенах"
    }
  ]
}
""";
}

```

Підключення штучного інтелекту

```
package com.example.chartbot;
```

```

import com.example.chartbot.handler.ChatGPT;
import com.example.chartbot.handler.ChatHandler;
import com.example.chartbot.handler.StartHandler;
import com.example.chartbot.query.QueryMessage;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import
org.telegram.telegrambots.extensions.bots.commandbot.TelegramLongPollingCommandBot;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Update;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;

@Component
public class MyTelegramBot extends TelegramLongPollingCommandBot {

    private final String username;
    @Autowired
    private StartHandler startHandler;

    @Autowired
    private ChatHandler chatHandler;

    @Override
    public String getBotUsername() {
        return username;
    }

    public MyTelegramBot(@Value("${bot.token}") String botToken,
        @Value("${bot.username}") String username) {
        super(botToken);
        this.username = username;
    }

    @Override
    public void processNonCommandUpdate(Update update) {
        if (update.hasMessage() && update.getMessage().hasText()) {
            String messageText = update.getMessage().getText();
            if (messageText.equals("/start")) {

```

```

        startHandler.execute(this, update.getMessage().getFrom(),
update.getMessage().getChat(),
        null);
    } else if (messageText.equals("Які показники існують при виборі цільової
аудиторії")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query1);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    } else if (messageText.equals("Які існують найефективніші канали
комунікації з цільовою аудиторією?")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query2);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    } else if (messageText.equals("Як закрити проблеми цільової аудиторії?"))
{
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query2);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    } else if (messageText.equals("Як визначити вашу цільову аудиторію?")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query2);

```

```

        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    } else if (messageText.equals("Як створити власний бізнес-план?")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query2);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    } else if (messageText.equals("Які онлайн інструменти та налаштування
потрібні для ефективної роботи бізнес плану?")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query3);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    }

    }else if (messageText.equals("Яким є бізнес-план для нового стартапу?")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query3);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    }
}

```

```
    } else if (messageText.equals(
        "Найефективніші бізнес-плани")) {
        ChatGPT.listTokens();
        String answer = ChatGPT.prompts(QueryMessage.query4);
        System.out.println(answer);
        try {
            execute(new
SendMessage(update.getMessage().getChat().getId().toString(), answer));
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    } else {
        chatHandler.answerGeneration(this, update.getMessage().getFrom(),
            update.getMessage().getChat(),
            messageText);
    }
}
}
}
}
0
```