

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра прикладної математики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Застосування технології чат-ботів для управління обробкою замовлень  
(тема)

Виконав:  
здобувач 2 року початку, групи ПМм-23-2  
Светличний О.В.  
(прізвище, ініціали)

Спеціальність 113 Прикладна математика  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика  
(повна назва освітньої програми)

Керівник доц. Бринза Н.О.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ \_\_\_\_\_ Сидоров М.В.  
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна математика

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ \_\_\_\_\_

(підпис)

“ 25 ” листопада 2024 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Світличному Олександр Валерійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Застосування технології чат-ботів для управління обробкою замовлень

затверджена наказом по університету від 22 листопада 2024 р. № 1223 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи технічні, аналітичні та емпіричні дані, які слугують основою для розробки, впровадження та оцінки ефективності чат-ботів для управління обробкою замовлень

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз предметної області

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій \_\_\_\_\_

1. Актуальність теми роботи \_\_\_\_\_

2. Постановка задачі \_\_\_\_\_

3. Аналіз предметної області \_\_\_\_\_

4. Метод чисельного аналізу \_\_\_\_\_

5. Результати обчислювального експерименту \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	25 листопада – 1 грудня 2024 р.	виконано
2	Вибір та обґрунтування методу	2 – 8 грудня 2024 р.	виконано
3	Розробка алгоритму і програми	9 – 22 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	23 – 29 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	30 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання 25 листопада 2024 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Бринза Н.О.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 127 с., 5 табл., 57 рис., 1 дод., 47 джерел.

АВТОМАТИЗАЦІЯ БІЗНЕС ПРОЦЕСІВ, ТЕЛЕГРАМ, ТЕЛЕГРАМ БОТ,  
УПРАВЛІННЯ ЗАМОВЛЕННЯМИ, ЧАТ БОТ.

Об'єкт дослідження – процес створення Telegram-бота для управління замовленнями.

Мета роботи – впровадження технології чат-ботів для автоматизації процесів управління замовленнями.

Методи дослідження – методи та технології, що застосовуються під час розробки Telegram-бота для управління замовленнями.

У кваліфікаційній роботі розглядається потенціал використання чат-ботів для автоматизації та покращення клієнтського обслуговування. Проаналізовано Telegram як платформу для розміщення ботів. Для вибору відповідних технологій проведено порівняння різних типів баз даних і мов програмування.

У результаті виконання роботи було розроблено та протестовано програму, яка забезпечує основні функції Telegram-бота для управління замовленнями.

Telegram-бот створений із використанням таких інструментів і технологій: MongoDB, MongoDB Shell, Compass, IntelliJ IDEA, Java 11, бібліотека org.telegram.telegrambots, Spring Boot, Project Lombok, Apache Maven, Git, Docker.

## ABSTRACT

Introductory note: 127 pages, 5 tables, 57 figures, 1 appendixes, 47 sources.

BUSINESS PROCESS AUTOMATION, CHAT BOT, ORDER MANAGEMENT, TELEGRAM, TELEGRAM BOT.

Object of research – process of creating a Telegram bot for order management.

Purpose of work – implement chatbot technology to automate order management processes.

Methods of research - methods and technologies used in the development of a Telegram bot for order management.

The study examines the potential of using chatbots to automate and improve customer service. Telegram as a platform for hosting bots is analyzed. Different types of databases and programming languages were compared to select the appropriate technologies.

As a result of the work, a program was developed and tested that provides the main functions of a Telegram bot for order management.

The Telegram bot was created using the following tools and technologies: MongoDB, MongoDB Shell, Compass, IntelliJ IDEA, Java 11, org.telegram.telegrambots library, Spring Boot, Project Lombok, Apache Maven, Git, Docker.

## ЗМІСТ

	С.
Вступ .....	8
1. Аналіз предметної області та постановка задач дослідження .....	10
1.1 Актуальність застосування чат-ботів.....	10
1.2 Telegram як платформа розміщення ботів.....	20
1.3 Змістовна та формальна постановка задачі .....	24
1.4 Постановка задач дослідження .....	25
2. Засоби та технології розроблення Telegram-ботів для управління обробкою замовлень .....	37
2.1 Вибір бази даних для створення Telegram-боту .....	37
2.2 Обґрунтування вибору мови програмування та відповідної програмної платформи для розробки Telegram-бот.....	39
2.3 Проектування алгоритму взаємодії користувача із проєктованим Telegram-ботом .....	48
Висновки за розділом 2 .....	51
3. Створення чат-боту для управління обробкою замовлень: аспекти реалізації та аналіз отриманих результатів .....	52
3.1 Створення бази даних для Telegram боту з використанням СКБД MongoDB.....	52
3.1.1. Створення колекцій.....	53
3.2 Розробка Telegram-бота з використанням Java .....	60
3.3 Впровадження постачання розробленого Telegram-боту .....	65
Висновки за розділом 3 .....	69
4. Результати обчислювального експерименту та їх аналіз .....	71
4.1 Тестування розробленого Telegram-боту .....	71
4.2 Аналіз використання Telegram-бота для управління обробкою замовлень .....	87
Висновки за розділом 4 .....	89

	7
Висновки .....	90
Перелік джерел посилання .....	91
Додаток А Лістинг програми .....	95

## ВСТУП

**Актуальність теми.** У сучасному бізнесі автоматизація процесів взаємодії з клієнтами є критично важливою для підвищення конкурентоспроможності підприємств. Розробка Telegram-ботів для управління замовленнями дозволяє вирішити низку актуальних проблем, таких як швидкість обробки запитів, доступність сервісу 24/7 та покращення зручності для користувачів.

Сучасний розвиток технологій у сфері автоматизації бізнес-процесів значно впливає на ефективність роботи підприємств, забезпечуючи зростання швидкості обробки даних, покращення якості обслуговування клієнтів та оптимізацію внутрішніх операцій. Використання чат-ботів, зокрема на платформі Telegram, є однією з ключових тенденцій у вирішенні завдань автоматизації управління замовленнями. Провідні компанії та наукові установи активно працюють над створенням та вдосконаленням технологій, що дозволяють інтегрувати чат-боти в існуючі бізнес-процеси.

Світові тенденції розвитку інформаційних технологій демонструють зростання популярності чат-ботів завдяки їхній простоті використання, багатofункціональності та широким можливостям інтеграції. Дослідження у цій сфері спрямовані на розробку інтелектуальних алгоритмів взаємодії, забезпечення масштабованості рішень та їхньої адаптації до конкретних потреб користувачів.

**Мета і завдання кваліфікаційної роботи.** Метою кваліфікаційної роботи є розробка Telegram-боту для автоматизації управління обробкою замовлень у малих та середніх підприємствах. Telegram-бот дозволить в автоматичному режимі приймати замовлення, без перерви та вихідних. За його допомогою можна створювати замовлення без допомоги людей. Що дозволить на невеликих підприємствах працювати з маленьким штатом працівників більш ефективно, не залучаючи більше працівників. Замовлення можна зробити навіть коли у магазині буде не робочий час. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести огляд і аналіз сучасного стану задачі автоматизації управління замовленнями;
- дослідити можливості використання платформи Telegram для створення чат-ботів;
- розробити алгоритми роботи чат-боту;
- проаналізувати актуальність застосування чат-ботів для бізнесу;
- вивчити можливості платформи Telegram для розміщення ботів;
- порівняти технології, які можуть бути використані для розробки Telegram-ботів;
- створити додаток, що реалізує функції Telegram-боту для управління замовленнями;
- провести тестування додатку та оцінити його ефективність.

*Об'єктом дослідження є процес створення Telegram-боту для управління обробкою замовлень.*

*Предметом дослідження є технології, що використовуються для реалізації такого боту.*

**Методи дослідження.** У кваліфікаційній роботі використовується метод аналізу існуючих технологій створення чат-ботів, розробка програмного забезпечення з використанням сучасних мов програмування, тестування функціоналу бота в умовах моделювання реальних бізнес-процесів.

Кваліфікаційна робота зосереджена на глибокому аналізі та розробці чат-бота для управління обробкою замовлень, з урахуванням критичного вивчення його сутності, ключових особливостей і переваг як для бізнесу, так і для клієнтів. Дослідження охоплює основні етапи розробки та впровадження чат-бота, включаючи його архітектуру та функціональні можливості.

У ході роботи застосовувалися сучасні інструменти та технології, зокрема MongoDB, MongoDB Shell, Compass, IntelliJ IDEA, Java, бібліотека org.telegram.telegrambots, Spring Boot, Project Lombok, Apache Maven, Git і Docker.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

## 1.1 Актуальність застосування чат-ботів

Сучасний світ глибоко інтегрований з інформаційними технологіями, а бізнес-середовище безперервно змінюється і вдосконалюється. В умовах цієї постійної динаміки застосування чат-ботів стає вкрай актуальним. На основі критичного огляду літератури були визначені основні фактори, що підкреслюють доцільність їх використання [1].

Фактор 1. Покращення обслуговування клієнтів завдяки чат-ботам: чат-боти здатні миттєво відповідати на запити клієнтів 24/7. Це особливо важливо в таких галузях, як онлайн-торгівля, готельний бізнес і сфера послуг, де оперативне надання інформації є критичним. Клієнти можуть одразу отримати відповіді без очікування зв'язку з оператором. Чат-боти також інформують клієнтів про статус замовлень, що дозволяє їм слідкувати за процесом обробки без необхідності дзвінків або написання звернень у службу підтримки [2].

Чат-боти можуть автоматично приймати нові замовлення, дозволяючи покупцям здійснювати покупки безпосередньо через бот, що робить процес швидким та зручним.

Сучасні чат-боти також використовують дані клієнтів для створення персоналізованих рекомендацій та пропозицій. Наприклад, вони можуть пропонувати індивідуальні знижки чи рекомендації на основі попередніх покупок, що покращує рівень задоволення обслуговуванням.

Чат-боти допомагають знизити час очікування на відповідь, що особливо важливо у сферах з високим навантаженням, де тривалі черги можуть негативно впливати на задоволення клієнтів.

Чат-боти здатні обробляти стандартні запити та прості завдання, які не потребують глибокого аналізу або залучення оператора, що дозволяє операторам зосередитися на більш складних завданнях [2].

Нижче представлена діаграма, яка ілюструє поліпшення обслуговування клієнтів за допомогою чат-ботів (рис. 1.1).

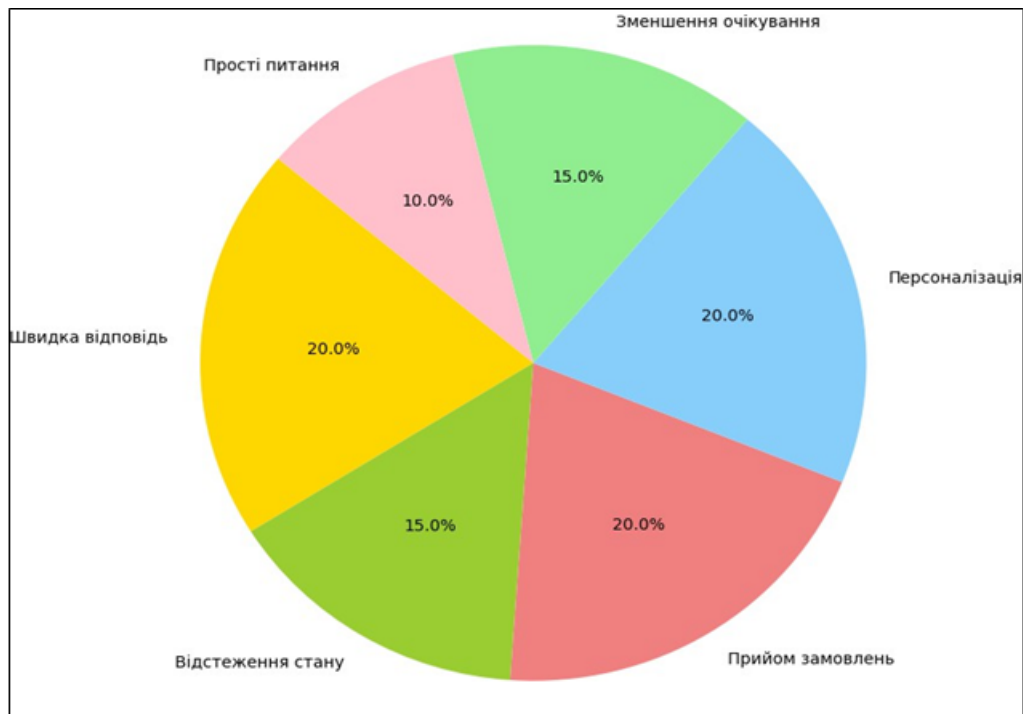


Рисунок 1.1 – Поліпшення обслуговування клієнтів за допомогою чат-ботів

Використання чат-ботів для підвищення якості обслуговування клієнтів стає особливо актуальним в умовах зростаючої конкуренції та зростаючих очікувань клієнтів щодо якості сервісу. Чат-боти допомагають компаніям забезпечувати швидке, ефективне та персоналізоване обслуговування, що може сприяти підвищенню лояльності клієнтів та зростанню обсягу продажів.

Фактор 2. Зниження витрат через автоматизацію завдяки чат-ботам: чат-боти надають можливість значно зменшити витрати, пов'язані з оплатою праці та іншими ресурсами [3]. На нашу думку, це відбувається завдяки наступним аспектам:

– автоматизація рутинних завдань: багато завдань, пов'язаних з обслуговуванням клієнтів і обробкою замовлень, є повторюваними і стандартними; це включає відповіді на поширені питання, заповнення форм, реєстрацію даних та

інші рутинні дії; замість того, щоб персонал виконував ці завдання вручну, чат-боти дозволяють автоматизувати процеси, що значно підвищує їх ефективність;

- цілодобова доступність: чат-боти працюють безперервно, надаючи клієнтам підтримку 24/7, що дозволяє знизити витрати на робочий час та заробітну плату персоналу, який в іншому випадку потребував би змінного графіка та перерв [4];

- здатність до масової обробки запитів: чат-боти можуть обслуговувати одночасно велику кількість клієнтських запитів, що дозволяє значно знизити кількість операторів та, відповідно, зменшити витрати на заробітну плату;

- зменшення кількості помилок: автоматизація процесів за допомогою чат-ботів сприяє зниженню помилок, які могли б допустити оператори; це допомагає уникнути витрат, пов'язаних з виправленням помилок і компенсаціями клієнтам;

- економія часу та коштів: висока ефективність роботи чат-ботів дозволяє знизити витрати на оплату праці та прискорити процеси обробки замовлень, що позитивно впливає на задоволеність клієнтів і їх лояльність.

Завдяки проаналізованим факторам, побудовано діаграму (рис. 1.2), яка відображає економію коштів завдяки впровадженню чат-ботів.

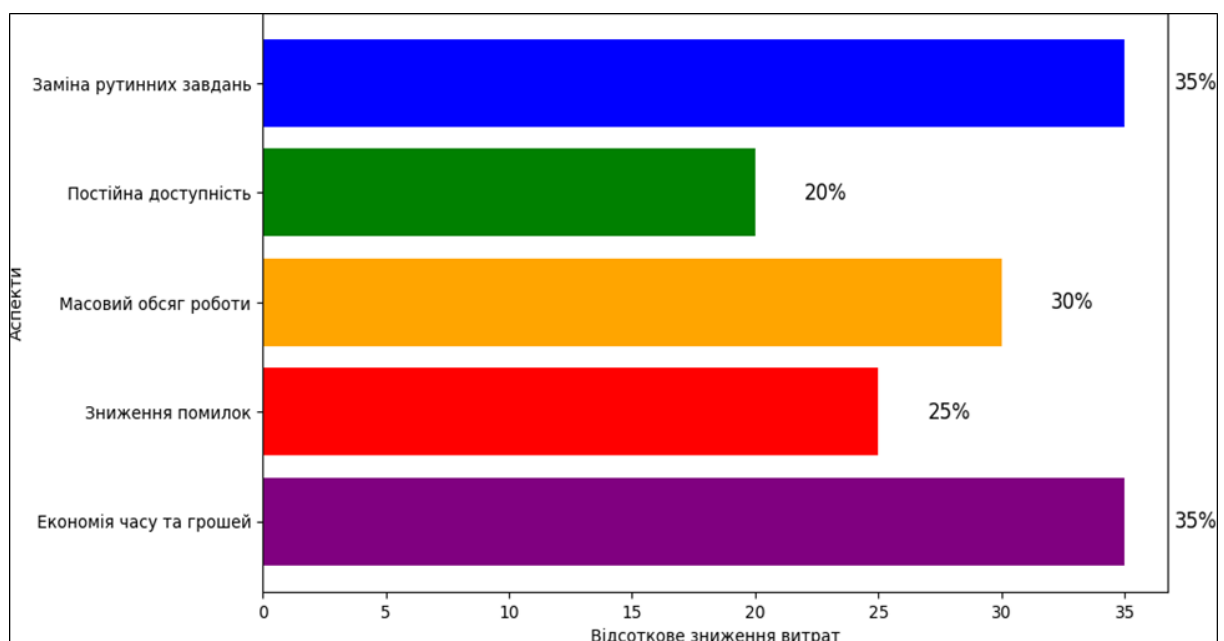


Рисунок 1.2 – Зниження витрат

Таким чином, використання чат-ботів сприяє оптимізації бізнес-процесів, зниженню витрат та підвищенню продуктивності, що є суттєвою конкурентною перевагою у сучасному бізнес-середовищі.

Фактор 3. Підвищення продуктивності та персоналізація обслуговування: чат-боти працюють цілодобово, що дозволяє підприємствам обробляти запити клієнтів у будь-який час, незалежно від робочих годин. На основі аналізу літературних джерел [1, 2, 4, 5] виділено наступні ключові переваги, що сприяють підвищенню продуктивності та персоналізації обслуговування за допомогою чат-ботів:

- швидкість реакції: чат-боти можуть миттєво відповідати на запити клієнтів, не потребуючи перерв, що дозволяє підприємствам швидше обробляти замовлення та запити, що позитивно впливає на задоволеність клієнтів і сприяє зростанню продажів;

- одночасна обробка запитів: чат-боти здатні одночасно вести численні розмови, що значно підвищує їх продуктивність; вони не схильні до втоми і не припускаються помилок через перевантаження;

- моніторинг та аналітика: чат-боти збирають дані про взаємодію з клієнтами, що дозволяє бізнесам аналізувати попит, оцінювати ефективність обслуговування та вдосконалювати свої послуги;

- навчання на основі даних: використовуючи машинне навчання та штучний інтелект, чат-боти здатні адаптуватися до потреб клієнтів, що дозволяє їм надавати більш персоналізовані відповіді;

- індивідуальні рекомендації: чат-боти можуть пропонувати товари або послуги, що найкраще відповідають інтересам клієнтів на основі історії покупок та попередніх запитів;

- персоналізовані акції та знижки: чат-боти здатні надавати індивідуальні пропозиції та знижки, що стимулює покупки та підвищує лояльність клієнтів;

- індивідуальний підхід: завдяки здатності слухати запити клієнтів, чат-боти забезпечують особистий підхід, який враховує індивідуальні потреби кожного клієнта.

Діаграма нижче (рис. 1.3) демонструє відносну вагу різних переваг чат-ботів [6].

Таким чином, використання чат-ботів дозволяє підприємствам не лише підвищити продуктивність і швидкість реагування на клієнтські запити, а й забезпечити більш персоналізований досвід, що сприяє зростанню лояльності клієнтів і покращенню конкурентоспроможності.

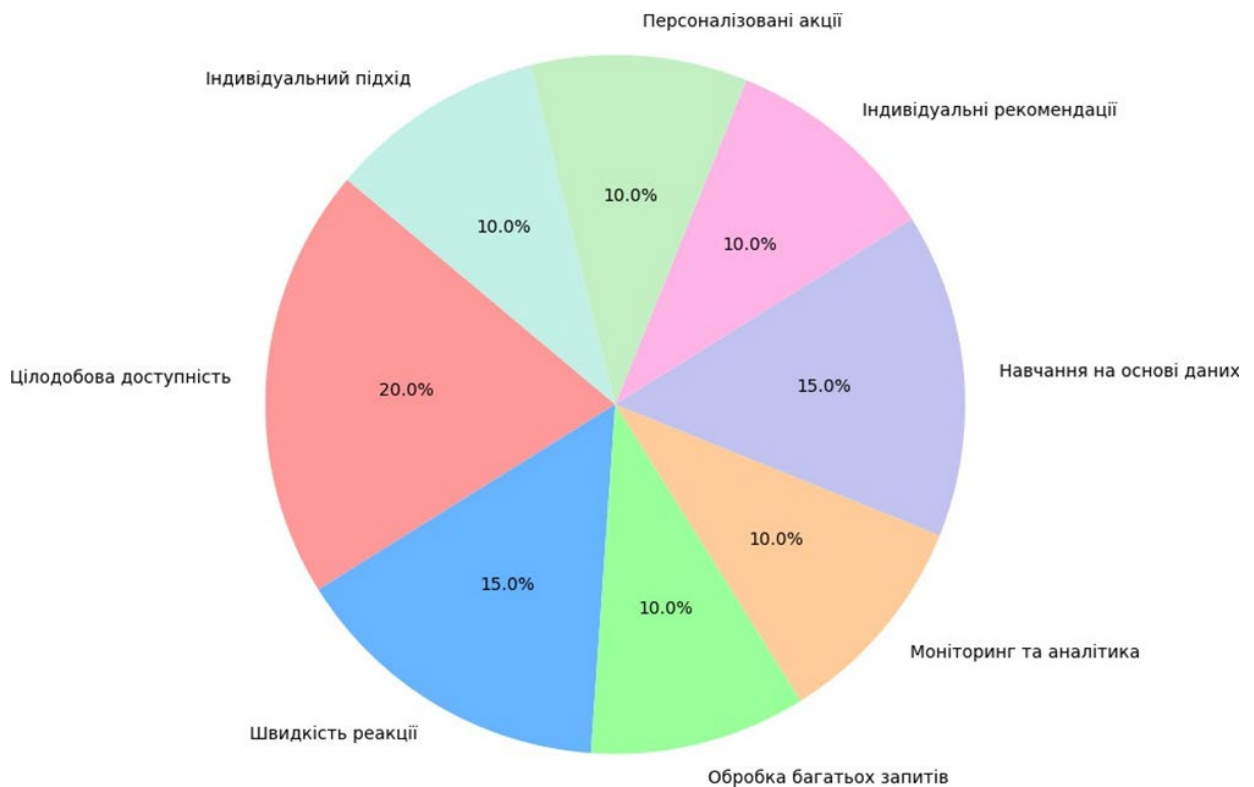


Рисунок 1.3 – Переваги чат-ботів

Фактор 4. Масштабованість бізнесу та підвищення конкурентоспроможності.

Застосування чат-ботів надає бізнесу значні переваги в контексті масштабування та підвищення конкурентоспроможності [7]. Переваги чат-ботів представлені на рис. 1.3.

Чат-боти здатні обробляти численні запити та замовлення одночасно, що особливо важливо в періоди високого попиту, як під час розпродажів чи святкових сезонів. Завдяки цьому більше клієнтів може отримати обслуговування без необхідності додаткового очікування.

Зазвичай масштабування обсягу обслуговування потребує додаткових витрат на інфраструктуру, персонал та інші ресурси. Чат-боти дозволяють компаніям обслуговувати більшу кількість клієнтів без значного збільшення витрат.

Чат-боти можуть обслуговувати клієнтів незалежно від їхнього місцезнаходження, що надає бізнесу можливість розширюватися глобально, залучаючи клієнтів із різних країн без необхідності фізичної присутності компанії у кожному регіоні.

Впровадження чат-ботів підвищує конкурентоспроможність бізнесу, оскільки ця технологія стає стандартом у багатьох галузях і відображає сучасний підхід до обслуговування клієнтів. Основні переваги конкурентоспроможності включають:

- покращення обслуговування клієнтів: компанії, що використовують чат-ботів, можуть швидко та зручно обслуговувати клієнтів, забезпечуючи миттєві відповіді на запити та допомогу у вирішенні проблем, що підвищує загальне задоволення клієнтів;

- зниження витрат: автоматизація рутинних завдань чат-ботами допомагає знизити витрати на оплату праці та інші операційні витрати, що дозволяє компаніям пропонувати конкурентоспроможні ціни та умови;

- відповідність сучасним тенденціям: використання чат-ботів підкреслює інноваційний підхід компанії, її готовність адаптуватися до сучасних технологічних змін, що покращує імідж бренду;

- залучення нових клієнтів: використання чат-ботів приваблює клієнтів, які цінують сучасні та технологічні підходи до обслуговування, забезпечуючи бізнесу перевагу в умовах конкуренції.

На рис. 1.4 зображено важливість кожної переваги використання чат-ботів у бізнесі.

У сучасному бізнес-середовищі застосування чат-ботів активно сприяє зростанню компаній, підвищенню продуктивності та покращенню конкурентоспроможності.



Рисунок 1.4 – Важливість кожної переваги використання чат-ботів у бізнесі

Очевидно, що впровадження чат-ботів для обробки замовлень є актуальним і важливим. Використання таких технологій здатне значно покращити ефективність компанії та задовольнити потреби клієнтів. Чат-боти, орієнтовані на обробку замовлень, можуть стати потужним інструментом автоматизації, що вирішує завдання від пошуку товарів до обробки замовлень, відстеження їхнього статусу та відповідей на запитання клієнтів.

Попри очевидні переваги, чат-боти ще не отримали широкого застосування в багатьох компаніях і галузях. Це може бути зумовлено як недостатнім розумінням їхніх можливостей, так і технічними складнощами, що виникають при інтеграції в наявні бізнес-процеси.

Мета цієї роботи полягає в дослідженні можливостей і переваг чат-ботів у контексті електронної комерції, аналізі їхнього потенціалу для управління обробкою замовлень, а також у розробці оптимальних методів впровадження цієї технології.

Чат-боти не лише покращують бізнес-процеси, але й підвищують рівень задоволеності клієнтів, що сприяє збільшенню обсягів продажів і доходів. Їхня роль у цьому процесі дуже значуща, адже вони виконують багато завдань, від

обробки запитів клієнтів до інформування про статус замовлення. Це полегшує життя споживачам та підвищує ефективність бізнесу. На рис. 1.5 представлена діаграма, що ілюструє галузі застосування чат-ботів. Незважаючи на їхні численні переваги, впровадження чат-ботів у сферу управління замовленнями все ще залишається обмеженим через низку технічних складнощів, потребу в інтеграції з існуючими системами та недостатню обізнаність про їхній потенціал.

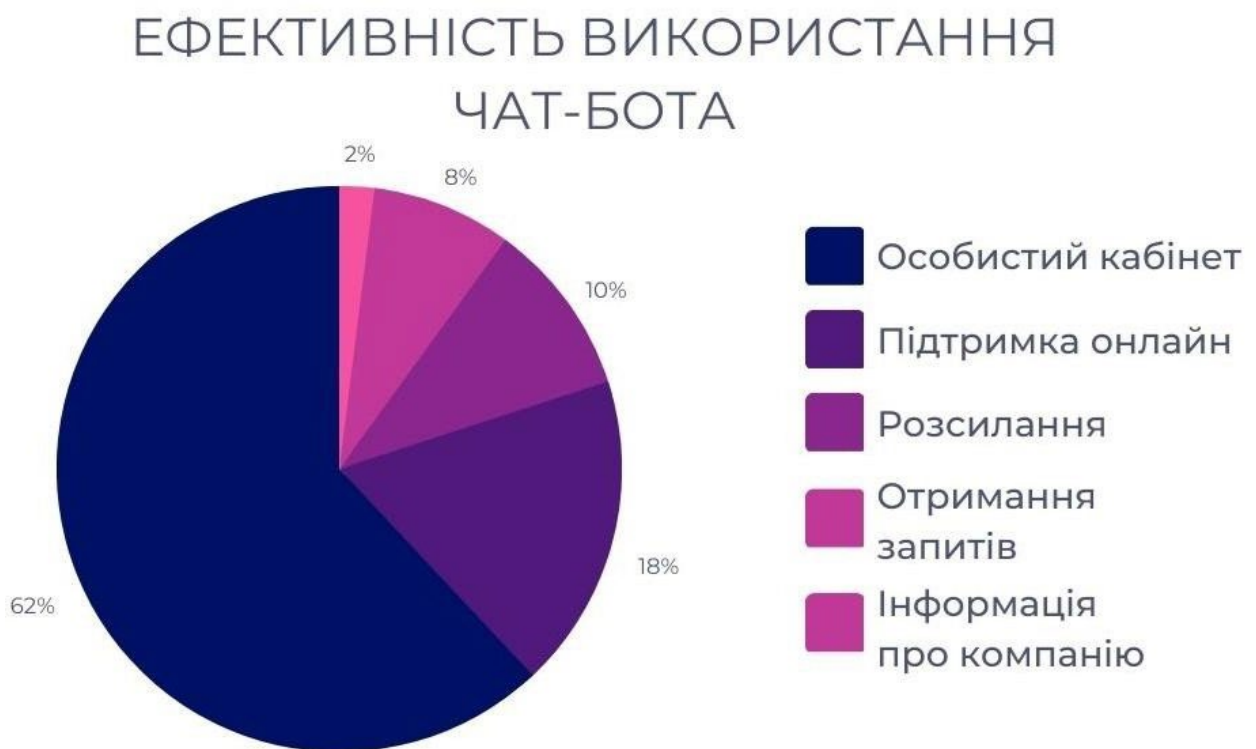


Рисунок 1.5 – Діаграма галузей використання чат ботів

Таким чином, ця кваліфікаційна робота відповідає актуальним викликам та може надати корисні результати для компаній, які займаються електронною комерцією.

Чат-боти стали важливим елементом сучасних інформаційних технологій та електронної комерції. Вони поділяються на кілька категорій за різними критеріями [3 – 6].

Критерій 1. За призначенням [4]: чат-боти для обслуговування клієнтів використовуються для вирішення різних завдань, пов'язаних із підтримкою

клієнтів. Їх функції включають: надання відповідей на запитання клієнтів, надання інформації про товари та послуги, роз'яснення деталей замовлення та технічної підтримки, прийом замовлень на товари та послуги, повідомлення про їхню ціну та доступність, прийом замовлень на товари та послуги, повідомлення про їхню ціну та доступність.

Переваги використання чат-ботів для підтримки клієнтів [3]:

- зниження витрат: автоматизація рутинних задач за допомогою чат-ботів дозволяє зменшити витрати компанії на клієнтське обслуговування;
- цілодобова підтримка: чат-боти забезпечують клієнтам доступ до інформації та підтримки 24/7;
- персоналізація: завдяки аналізу потреб та інтересів клієнтів, чат-боти можуть пропонувати індивідуальний підхід.

Чат-боти для розваг створені для забезпечення користувачів відпочинком та розвагами. Серед основних функцій можна виділити:

- проведення тестів, зокрема особистісних, інтелектуальних або тестів на знання;
- організація розважальних вікторин;
- створення цікавих історій для користувачів.

Чат-боти для навчання можуть використовуватися для надання користувачам доступу до освітніх матеріалів, допомоги у вивченні нових навичок або отримання інформації на певні теми.

Критерій 2. За способом комунікації [4]: текстові чат-боти є найбільш поширеними. Вони використовують текстовий формат для взаємодії, де користувачі вводять запитання або команди, а боти відповідають текстом. Аудіо чат-боти спілкуються через голосові повідомлення: вони розпізнають мову користувача і відповідають аудіо. Візуальні чат-боти залучають графічні елементи – зображення, відео або іншу візуальну інформацію для відповідей.

Критерій 3. За рівнем інтелектуальності [5]: прості чат-боти працюють за задалегідь встановленими правилами та сценаріями, відповідаючи на основі

заданих шаблонів. Модеровані чат-боти комбінують шаблони та можливість навчання: вони покращуються на основі досвіду, але їхні дії контролюються. Інтелектуальні чат-боти, що використовують штучний інтелект, здатні аналізувати текст, розуміти контекст та вести розмову, приймаючи самостійні рішення.

Критерій 4. За галуззю застосування [4]: медичні чат-боти надають консультації з приводу здоров'я, відповідають на запитання щодо симптомів, нагадують про прийом ліків та здійснюють моніторинг стану пацієнтів. Банківські чат-боти допомагають клієнтам з управлінням рахунками, переказами, отриманням інформації про фінансові послуги. Чат-боти для роздрібної торгівлі підтримують покупців у пошуку товарів, формуванні замовлень, інформуванні про знижки та акції. Чат-боти для управління персоналом та процесами забезпечують робітників інформацією для виконання завдань та автоматизують деякі виробничі процеси.

Перелік можливих сфер застосування чат-ботів може бути розширений.

Критерій 5. За платформою [6]:

– чат-боти для соціальних мереж інтегруються в месенджери та платформи соціальних мереж, такі як Facebook Messenger, WhatsApp або Telegram, їх можна використовувати для різноманітних завдань: надання інформації, продаж товарів і послуг або взаємодія з користувачами;

– чат-боти на веб-сайтах працюють на сайтах компаній, забезпечуючи підтримку клієнтів онлайн; вони відповідають на запитання, надають інформацію про продукти та послуги й можуть сприяти продажам;

– чат-боти в мобільних додатках інтегровані в застосунки для смартфонів, що дає можливість користувачам спілкуватися з ботами на своїх мобільних пристроях; вони підходять для надання інформації, продажу товарів та послуг або просто спілкування з клієнтами.

На рис. 1.6 представлено діаграму популярності чат-ботів за рівнем інтелектуальності [6], а загальну класифікацію чат-ботів наведено на рис. 1.7.

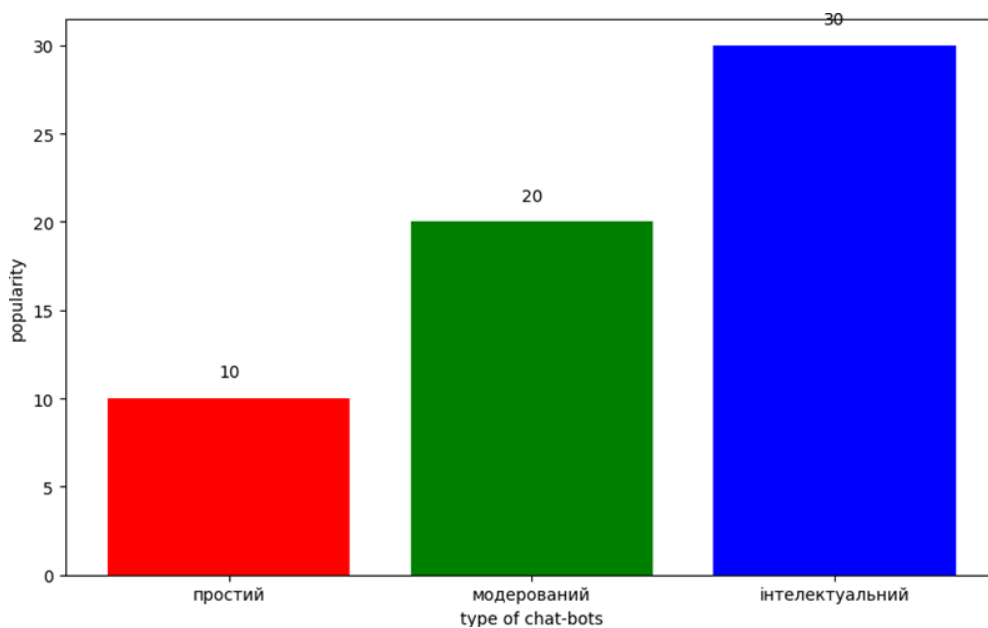


Рисунок 1.6 – Діаграма популярності чат-ботів за рівнем інтелектуальності

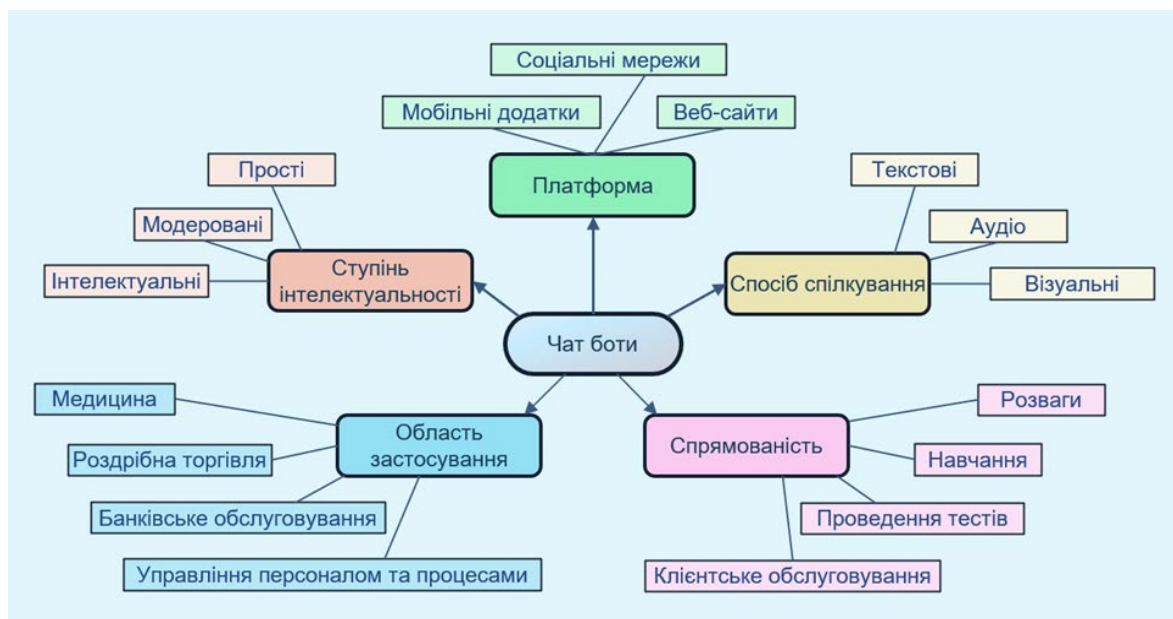


Рисунок 1.7 – Класифікація чат-ботів

## 1.2 Telegram як платформа розміщення ботів

Telegram є одним із найпопулярніших месенджерів та соціальних платформ у світі. Однією з основних його переваг є можливість створення та застосування чат-ботів - автоматизованих облікових записів, які взаємодіють із користувачами через чат.

Чат-бот у Telegram – це особливий вид облікових записів, здатний виконувати різноманітні завдання та реагувати на запити користувачів у формі повідомлень. Завдяки програмуванню їхня функціональність є гнучкою, що дозволяє налаштовувати боти для виконання різноманітних завдань: від відповідей на запитання до виконання автоматизованих дій.

Приклади застосування Telegram-ботів [1 – 3, 7, 8]:

- обслуговування клієнтів: багато компаній використовують Telegram-ботів для підтримки клієнтів, де користувачі можуть отримати інформацію про товари чи послуги, перевірити статус замовлення або вирішити питання чи скарги;

- нагадування та календарі: Telegram-боти можуть надсилати нагадування про важливі події, дедлайни та інші дати, а також допомагати додавати чи оновлення;

- аналітика та звітність: деякі компанії застосовують Telegram-ботів для отримання звітів та статистики, таких як моніторинг продажів або відвідуваності сайту створювати записи в календарі;

- розваги та ігри: Telegram-боти використовуються для інтеграції розважальних функцій, пропонуючи користувачам ігри, опитування, головоломки тощо;

- фінансові послуги: Telegram також підтримує боти для виконання фінансових операцій, зокрема переказу коштів, перевірки балансу, конвертації валют тощо.

На основі критичного аналізу літератури [1, 3, 7] та власного практичного досвіду розробника і користувача було визначено кілька переваг використання Telegram-ботів:

- зручність використання – спілкування з ботами відбувається через чат, що забезпечує зручний інтерфейс для користувачів;

- автоматизація – боти здатні автоматично виконувати багато завдань, що економить час та зусилля;

– глобальний доступ – Telegram-боти можна використовувати в будь-якій точці світу, що робить їх доступними для глобальної аудиторії;

– спільнота розробників – Telegram надає інструменти для розробників, що спрощує створення та розгортання ботів, а також має активну спільноту розробників.

На рис. 1.8 представлено діаграму, яка відображає поточне використання Telegram-ботів у різних категоріях [9].



Рисунок 1.8 – Використання Telegram-ботів за категоріями

Чат-боти є ефективним засобом для взаємодії з аудиторією, надання послуг та автоматизації рутинних завдань. Вони корисні як для бізнесу, так і для розважальних проєктів, забезпечуючи зручний спосіб спілкування з користувачами.

Для дослідження процесу розробки Telegram-ботів, призначених для управління обробкою замовлень, важливо виконати аналіз наявної літератури. У цьому розділі проведено огляд доступних джерел з метою визначення існуючих підходів, методологій, технологій та інструментів, які використовуються в цій галузі. Аналіз наукових джерел є ключовим етапом у дослідженні, оскільки дає уявлення про поточний стан розвитку обраної тематики. Основною метою цього огляду є виділення важливих аспектів, які визначають процес створення Telegram-ботів для управління замовленнями.

У процесі аналізу літератури розглядаються наукові статті, книги, журнальні публікації та інші матеріали, що стосуються розробки чат-ботів. Особлива увага приділяється вивченню різних мов програмування та фреймворків, які застосовуються для створення ботів, оскільки вибір правильної технології може суттєво вплинути на ефективність і функціональність рішення.

Крім того, аналізуються архітектурні підходи, що застосовуються в розробці чат-ботів, зокрема методи інтеграції з базами даних, сервісами для обробки платежів та іншими зовнішніми системами. Це має особливе значення для забезпечення належної роботи бота при обробці замовлень у Telegram.

Також у процесі аналізу виділяються аспекти безпеки і конфіденційності, що є особливо важливими при розробці ботів для обробки замовлень, оскільки обробка персональних і фінансових даних вимагає надійного захисту.

Таким чином, цей огляд допоможе сформуванню основи для подальших досліджень і створення бота, орієнтованого на управління обробкою замовлень, із використанням найкращих практик, знайдених у літературі.

Аналіз показує значущість створення таких чат-ботів у сучасному бізнес-середовищі. Telegram, як платформа, надає розробникам потужний інструментарій через відкритий API та активну спільноту. Це сприяє зростанню інтересу до Telegram-ботів у сферах, таких як електронна комерція, логістика та культура. При виборі мов програмування та технологій важливо враховувати специфічні вимоги проєкту. Аналіз також вказує, що більшість Telegram-ботів використовують архітектуру клієнт-сервер, що спрощує їх розробку та підтримку. Щодо безпеки, джерела акцентують на критичності цього аспекту, особливо при обробці персональних даних та фінансових операцій. Застосування шифрування та інших заходів безпеки є необхідним для захисту інформації користувачів.

Загалом, огляд літератури підтверджує активний інтерес до розробки Telegram-ботів для управління замовленнями та забезпечує підґрунтя для подальшого дослідження і розробки в цій сфері.

### 1.3 Змістовна та формальна постановка задачі

Метою розробки є створення Telegram-бота для автоматизації процесів управління обробкою замовлень. Основними завданнями бота є приймання замовлень від користувачів, надання інформації про товари, обробка даних клієнтів, вибір способу оплати та доставки, а також відстеження статусу замовлення. Цей бот повинен забезпечити інтуїтивну взаємодію з користувачами, підтримуючи швидкий і ефективний обмін інформацією.

Функціональні вимоги до бота:

- приймання замовлень, користувач повинен мати змогу вибирати товари з каталогу, вказувати їх кількість та параметри (наприклад, розмір або колір);
- збір даних клієнтів, бот повинен запитувати у користувача ім'я, контактний номер телефону та адресу доставки;
- вибір способу оплати та доставки, бот має надавати перелік доступних варіантів оплати та доставки, дозволяючи користувачеві зробити вибір;
- відстеження статусу замовлення, користувач повинен мати змогу отримувати актуальну інформацію про стан виконання замовлення;
- зворотний зв'язок, бот повинен надавати можливість користувачам залишати відгуки про виконання замовлення;
- інтеграція з базою даних, для збереження інформації про замовлення та клієнтів бот має бути інтегрований із базою даних.

Обмеження і припущення:

- бот працює на платформі Telegram і використовує Telegram API для взаємодії з користувачами;
- інформація про товари зберігається в базі даних і може оновлюватися адміністратором;
- сценарії роботи бота повинні підтримувати як прості замовлення (одиночний товар), так і складні (кілька товарів з різними характеристиками);

– система повинна забезпечувати конфіденційність даних користувачів та відповідати основним вимогам безпеки.

Формальна постановка задачі.

Необхідно зробити Telegram-бот  $B$ , який виконає наступні дії:

– приймає вхідні дані від користувача  $U$ :

$$D_U = \{T_i, C, P, D, F\},$$

де  $T_i$  – набір товарів з характеристиками (назва, кількість, параметри);

$C$  – контактні дані;

$P$  – спосіб оплати;

$D$  – спосіб доставки;

$F$  – зворотній зв'язок;

– зберігає отримані дані  $D_U$  в базі даних  $DB$ ;

– надає користувачу інформацію про статус виконання замовлення  $S$ .

Цільова функція: необхідна розробка алгоритмів, які забезпечують максимально швидку обробку запитів користувача, мінімізацію помилок при введенні та збереженні даних та оптимальне управління ресурсами бази даних для забезпечення високої продуктивності бота.

Таким чином, розробка Telegram-бота передбачає створення програмного продукту, який відповідає зазначеним вимогам і може бути ефективно використаний для автоматизації обробки замовлень.

#### 1.4. Постановка задач дослідження

Виходячи з актуальності проблеми, головним науковим завданням вважається розробка дієвого чат-бота для управління процесом обробки замовлень. Для вирішення цього завдання необхідно вивчити основні принципи роботи та можливості чат-ботів, дослідити потреби користувачів і компаній в сфері елек-

тронної комерції, а також розробити ефективні методи і технології для їх використання.

Вирішення цього завдання потребує комплексного підходу, який включає декілька основних етапів.

На першому етапі важливо провести дослідження, яке дозволить краще зрозуміти вимоги клієнтів і специфіку функціонування чат-ботів. Це передбачає аналіз різних типів ботів, їх функціональних можливостей, а також методів інтеграції у бізнес-процеси. Важливим є також дослідження досвіду користувачів для виявлення можливих недоліків і напрямів покращення.

На другому етапі, на основі зібраної інформації, буде розроблена концепція чат-бота. Вона повинна враховувати потреби цільової аудиторії, а також технічні обмеження та можливості Telegram. На третьому етапі буде реалізовано розробку та тестування прототипу бота.

Четвертий етап передбачає оцінку ефективності створеного чат-бота, включаючи аналіз його практичного використання, збирання зворотного зв'язку та аналіз показників роботи.

Подальші дослідницькі питання формуються з огляду на поставлене завдання та спрямовані на визначення основних напрямів дослідження. Особливість цієї роботи полягає в тому, що вона охоплює розробку конкретного продукту – чат-бота для управління замовленнями, що потребує інтеграції теоретичних знань та практичних навичок.

Основні питання, які буде досліджено в межах цієї роботи, включають такі аспекти:

- які основні можливості чат-ботів і як їх можна застосувати для управління обробкою замовлень;
- які потреби користувачів і компаній, пов'язані з електронною комерцією, може задовольнити чат-бот;
- які технічні виклики і проблеми виникають при розробці та інтеграції чат-бота у бізнес-процеси;

– які методи і технології найефективніше використовувати для створення чат-бота, що полегшує управління обробкою замовлень.

Ці питання допоможуть окреслити межі дослідження і визначити його основні напрями, а також зосередити зусилля на отриманні відповідей, що будуть корисними для вирішення наукової задачі. У наступних розділах кожне з цих питань розглядатиметься детально.

Перше питання дослідження стосується можливостей чат-ботів у сфері управління обробкою замовлень. Чат-боти – це програми, здатні автоматизувати різноманітні види взаємодії в Telegram. Вони можуть відповідати на повідомлення, починати діалог, виконувати команди та здійснювати операції в інших програмах чи сервісах. У контексті управління замовленнями, чат-боти можуть виконувати низку важливих функцій.

Наприклад, бот може відповідати на запити користувачів щодо продукції чи послуг, пропонованих компанією, включаючи інформацію про ціну, наявність товару, його характеристики тощо. Бот може також допомогти оформити замовлення, прийняти кількість товару, вибрати спосіб доставки. Після оформлення замовлення бот здатний відслідковувати його статус і повідомляти користувачів про будь-які зміни.

При цьому можливості чат-ботів можуть бути значно ширшими, адже вони здатні виконувати складніші завдання, такі як аналіз даних користувачів, персоналізація, інтеграція з іншими системами, тощо.

Для повного використання потенціалу чат-ботів у сфері управління обробкою замовлень необхідно ретельно вивчити їх функціонал і обмеження, а також застосувати найкращі методи та підходи для розробки.

Друге питання стосується потреб користувачів і бізнесу у сфері електронної комерції, які можна задовольнити за допомогою чат-бота. Розуміння специфічних вимог як користувачів, так і компаній є важливим чинником під час розробки бота.

З погляду користувачів, ключовими потребами, які чат-бот може задовольнити, є зручність і ефективність. Вони очікують зручний інтерфейс для

пошуку та замовлення товарів, а також можливість відслідковувати статус замовлень. Крім того, користувачі цінують можливість оперативно отримувати відповіді на свої запитання та вирішувати проблеми, що можуть виникнути під час оформлення замовлення [10].

Зі сторони бізнесу основні потреби зводяться до підвищення продажів, утримання клієнтів та оптимізації внутрішніх процесів. Чат-боти можуть задовольнити ці вимоги через автоматизацію процесів продажу і клієнтської підтримки, що сприяє швидшому вирішенню проблем клієнтів, покращенню комунікації та взаємодії з ними, а також збору даних для подальшого аналізу та вдосконалення бізнес-процесів.

Під час розробки чат-бота важливо врахувати ці потреби і забезпечити, щоб бот не лише відповідав їм, але й надавав додаткові переваги, які забезпечать йому конкурентоспроможність на ринку.

Третє дослідницьке питання стосується основних технічних викликів і труднощів, які виникають під час розробки та інтеграції чат-бота в існуючі бізнес-процеси. Розробка бота для управління обробкою замовлень пов'язана з низкою специфічних технічних труднощів, що потребують глибокого розуміння та аналізу.

Серед основних викликів:

- створення надійної та гнучкої архітектури бота, здатної обробляти велику кількість запитів і даних;
- інтеграція бота з наявними ІТ-системами компанії, такими як системи управління базами даних, CRM, обробка замовлень, платіжні системи тощо;
- забезпечення захисту даних користувачів і відповідність нормативним вимогам;
- розробка ефективних алгоритмів обробки природної мови для взаємодії з користувачами;
- підтримка багатомовності та врахування культурних особливостей для глобального використання.

Для створення високоякісного та ефективного чат-бота необхідно розробити відповідні підходи для подолання цих викликів, що включає вивчення найкращих практик розробки ботів, застосування сучасних технологій і тісну співпрацю з ІТ-командою компанії [11].

Варто також врахувати, що інтеграція бота може вимагати значних змін у поточних бізнес-процесах, таких як перегляд і оптимізація процесів обробки замовлень, зміни в обслуговуванні клієнтів, налаштування ІТ-систем для взаємодії з ботом і навчання персоналу для роботи з новим інструментом.

Створення та інтеграція чат-бота для обробки замовлень – це комплексний та ресурсозатратний процес, що потребує ретельного планування, дослідження та підготовки. Проте, якщо підхід буде успішним, результати можуть включати підвищення ефективності, збільшення продажів, поліпшення задоволеності клієнтів, що в кінцевому підсумку призведе до зростання прибутковості.

Це підводить нас до четвертого дослідницького питання: які підходи та стратегії можна застосувати для розробки і впровадження чат-бота, здатного ефективно управляти обробкою замовлень.

При створенні та впровадженні бота слід врахувати кілька важливих чинників. Перш за все, бот повинен бути корисним і відповідати очікуванням користувачів, мати простий та зручний інтерфейс, швидко відповідати на запити та вирішувати проблеми клієнтів.

Далі, бот має бути інтегрований у бізнес-процеси компанії з метою підвищення їхньої ефективності та продуктивності, що може включати автоматизацію обробки замовлень, управління комунікацією з клієнтами та збір даних для вдосконалення стратегії бізнесу.

Також важливо реалізувати стратегію захисту даних користувачів і забезпечення відповідності правовим нормам. Це передбачає використання надійних механізмів захисту даних, дотримання відповідних законодавчих вимог та регулярні перевірки відповідності.

Останній, але не менш важливий аспект – це оцінка ефективності та результативності бота. Необхідно мати інструменти для моніторингу роботи бота, щоб оперативно виявляти і вирішувати проблеми та постійно покращувати його функціонал.

Таким чином, виклики, що виникають під час розробки та впровадження чат-бота для управління обробкою замовлень, формують наукову задачу, яка полягає в оптимальному виборі стратегій і методів для досягнення максимальної ефективності та результативності бота.

Загалом, розробка та впровадження чат-бота для управління обробкою замовлень у компанії має значний потенціал для покращення бізнес-процесів, обслуговування клієнтів та збільшення продажів. Водночас, компанія стикається з важливими викликами, які потребують ґрунтовного аналізу та рішень.

Ці виклики охоплюють технічні аспекти розробки бота, його інтеграцію з існуючими ІТ-системами та бізнес-процесами, забезпечення безпеки даних користувачів та відповідність правовим нормам. Крім того, компанії необхідно визначити ефективні стратегії впровадження, удосконалення та оцінки результативності бота.

При аналізі підходів, методів та моделей для досягнення цілей майбутнього чат-бота "Управління обробкою замовлень" слід сконцентрувати увагу на трьох основних аспектах: проектуванні бота, обробці даних та інтерфейсі користувача.

На перший погляд, розробка чат-бота може здаватися простою задачею, але при детальному аналізі стає зрозуміло, що це складний процес, що вимагає глибоких знань у галузях комп'ютерних наук, програмування, алгоритмів та мережевих технологій. Детальне вивчення цих аспектів дозволить краще зрозуміти процес проектування та функціонування ботів у Telegram.

На рис. 1.9 наведена спрощена схема роботи чат-бота.

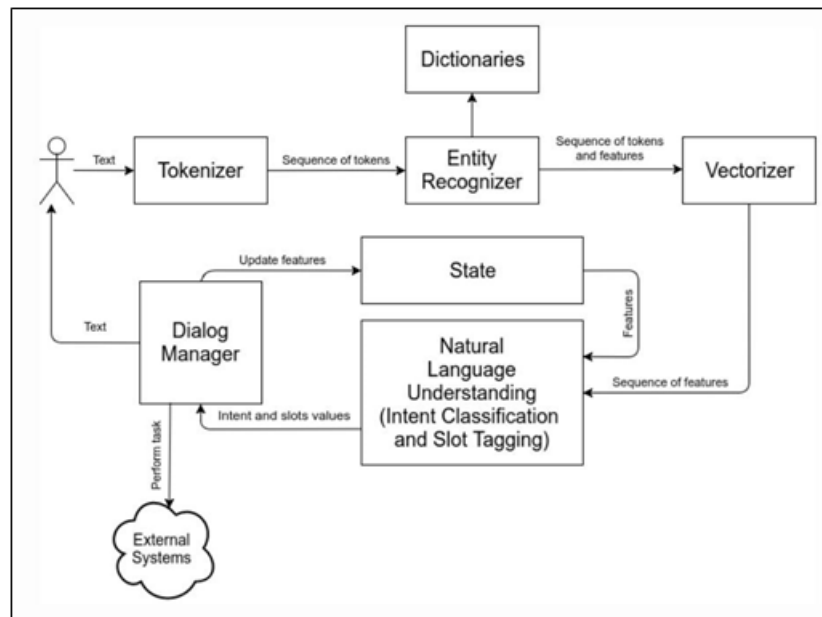


Рисунок 1.9 – Структура роботи чат-боту

Процес проектування чат-бота, як і будь-якого іншого програмного продукту, потребує ретельного планування перед переходом до розробки. При цьому важливо врахувати цілі бота, його можливості та обмеження, які накладаються платформою Telegram. Зокрема, бот повинен бути здатний обробляти запити користувачів і надавати відповіді в реальному часі, що вимагає застосування асинхронного програмування для паралельної обробки кількох запитів одночасно [12].

Другим важливим аспектом є алгоритми обробки даних. Оскільки бот має працювати з великими обсягами інформації, вибір оптимальних алгоритмів і структур даних є критично важливим для забезпечення його ефективності та надійності.

Ще одним ключовим елементом є інтерфейс користувача. Успішність бота значною мірою залежить від його зручності та інтуїтивності. Тому дослідження найкращих практик створення користувацьких інтерфейсів та аналіз відгуків користувачів є важливими етапами в процесі проектування.

Не менш важливим є вибір інструментів для розробки. Потрібно врахувати безліч чинників, зокрема здатність інструментів відповідати вимогам

бота та їх сумісність із іншими технологіями, що використовуються в проекті [13].

Загалом, дослідження в цій галузі повинно сприяти виявленню та оцінці різних підходів, методів і моделей, які допоможуть створити ефективного чат-бота для обробки замовлень. Результати таких досліджень послужать основою для розробки бота, що відповідатиме всім вимогам і надасть користувачам зручний та ефективний інструмент для обробки замовлень.

Однак для успішного проектування бота важливо розуміти його потреби і цілі, зокрема вимоги та очікування клієнтів і менеджерів. Аналіз демографічних, культурних і технологічних особливостей цільової аудиторії, а також їхніх потреб є важливими етапами.

Інтеграція системи управління базами даних є важливим фактором для успішного впровадження бота. Зокрема, бот повинен мати можливість обробляти та відстежувати замовлення, що потребує вибору відповідної моделі управління базами даних [14].

Забезпечення безпеки, особливо в контексті обробки замовлень, є критичним, і тому важливо використовувати ефективні механізми аутентифікації, авторизації, шифрування даних, а також надійні підходи до зберігання та передачі даних.

Також слід враховувати сумісність бота з різними версіями Telegram і платформами, оскільки різні версії можуть мати відмінності в функціональності.

Процес розробки чат-бота має бути гнучким і здатним швидко реагувати на зміни, виправляючи помилки та адаптуючись до нових вимог і технологічних змін. Створення ефективного дизайну інтерфейсу користувача, що задовольняє потреби клієнтів та забезпечує високу продуктивність і надійність системи, є важливим етапом цього процесу.

Не менш значущим є взаємодія з користувачами. Розробники повинні активно збирати відгуки та пропозиції від користувачів і використовувати їх для покращення функціональності продукту. Крім того, необхідно співпрацювати з

іншими фахівцями та розробниками для обміну досвідом, отримання підтримки та вдосконалення проекту.

Успіх проекту багато в чому залежить від відданості та мотивації команди. Розробники повинні бути готові до інтенсивної роботи, зберігати високу мотивацію і прагнути до створення якісного продукту. Важливо також створювати умови для інновацій та творчого розвитку, даючи можливість експериментувати з новими ідеями, вивчати нові технології та вдосконалювати свої навички.

Таким чином, розробка чат-бота "Управління обробкою замовлень" є важливим і перспективним проектом, здатним принести значні переваги користувачам, менеджерам, розробникам і всій галузі. Однак, як і в будь-якому інноваційному проекті, існують певні ризики.

Перший ризик пов'язаний з технічними труднощами. Якщо команда не має достатнього досвіду або ресурсів, це може призвести до проблем, затримок або навіть невдач у розробці. Тому важливо забезпечити належну підготовку команди, фінансову та технічну підтримку, а також ефективне управління проектом.

Другим ризиком є захист даних. Оскільки чат-бот буде працювати з особистою та фінансовою інформацією користувачів, потрібно вжити необхідних заходів для її безпеки. Це включає застосування надійних методів шифрування, регулярне тестування безпеки та дотримання законодавчих норм і стандартів.

Третій ризик пов'язаний із можливою низькою прийнятністю бота серед користувачів. Якщо користувачі виявлять, що бот складний у використанні або не виконує свої функції, вони можуть відмовитись від його використання. Щоб уникнути цього, важливо постійно взаємодіяти з користувачами, стежити за їхніми відгуками та вчасно вносити необхідні корективи.

Врахування цих ризиків є важливою частиною успішного впровадження проекту. Розробка чат-бота може бути складною, але надзвичайно результативною, якщо до процесу підходити з професіоналізмом, обережністю та відкритістю до нових можливостей [12].

Крім того, важливо звернути увагу на адаптацію чат-бота до конкретних бізнес-моделей і контекстів використання. Оскільки різні компанії мають свої унікальні потреби, бот повинен бути здатний адаптуватися до різних сценаріїв. Наприклад, невелика компанія може використовувати бота для автоматизації замовлень, в той час як більша корпорація – для управління складними логістичними процесами.

У контексті цього дослідження основна мета – проаналізувати переваги розробки чат-бота "Управління обробкою замовлень". Для цього перед нами стоять кілька завдань. Перше з них – детальний аналіз ринку і технологічних трендів, зокрема у сфері чат-ботів. Важливо зрозуміти потенціал цього напрямку і можливості його використання для бізнесу, дослідити основних гравців на ринку та їх стратегії, а також вивчити успішні приклади впровадження подібних технологій для обробки замовлень.

Наступним завданням є розробка концепції чат-бота для управління обробкою замовлень, що включає створення детального плану з функціональності, принципів роботи, інтерфейсу та вимог до безпеки даних користувачів, з урахуванням отриманих знань з аналізу ринку [15].

Для того, щоб зрозуміти, як зручність використання чат-бота впливає на його прийнятність серед користувачів, необхідно дослідити, які конкретно характеристики бота покращують його зручність та задоволення від використання. Важливо визначити, які функції та елементи дизайну роблять бот легким і приємним у використанні.

Крім того, важливо оцінити економічну ефективність впровадження чат-бота. Потрібно прорахувати витрати на його розробку і подальшу підтримку, а також спрогнозувати доходи від його використання, з урахуванням витрат на програмування, серверне обладнання, маркетинг і підтримку користувачів.

Після цього слід узагальнити результати аналізу і сформулювати рекомендації. Потрібно виділити основні переваги та потенційні труднощі, пов'язані з розробкою і запуском чат-бота для управління замовленнями.

Узагальнення даних і висновків дозволить розробити стратегію для подальшого розвитку цього напрямку, що буде корисною для організацій, які мають намір оптимізувати свою діяльність і підвищити конкурентоспроможність за допомогою цієї технології.

Також важливо звернути увагу на етичні питання, зокрема на захист персональних даних і забезпечення безпеки. Необхідно ретельно проаналізувати аспекти зберігання та обробки даних і забезпечити відповідність чинним законодавчим нормам і стандартам.

Іншим важливим аспектом є оцінка впливу чат-бота на робочі процеси в організації. Потрібно проаналізувати, як зміниться роль учасників процесу, включаючи менеджерів і співробітників служби підтримки. Може виникнути потреба в коригуванні організаційної структури або способів взаємодії між учасниками процесу.

Останнім етапом буде розробка стратегії тестування та впровадження бота. Потрібно вибрати методи тестування, визначити етапи впровадження і встановити критерії успіху. Кожна нова технологія потребує ретельного тестування, щоб забезпечити її ефективність і надійність.

З широкої соціокультурної точки зору, слід розглянути, як впровадження чат-бота вплине на комунікацію між користувачами та менеджерами. Важливо зрозуміти, як зміняться взаємовідносини між учасниками процесу і який ефект це може мати на їхню співпрацю [16].

З практичної точки зору важливо вивчити стратегії впровадження бота в різних сферах, таких як комерційні й некомерційні організації, великі і малі підприємства, а також урядові та приватні структури. Потрібно аналізувати потреби різних груп користувачів і підбирати найбільш ефективні методи впровадження для кожної категорії.

Вивчення всіх аспектів впровадження чат-бота дозволить сформулювати комплексні рекомендації та стратегії для його використання. Оскільки технології швидко змінюються, важливо передбачити тенденції розвитку та адаптувати стратегії до майбутніх змін.

Таким чином, це дослідження не лише оцінює переваги і недоліки чат-бота для управління замовленнями, але й аналізує його можливий вплив на ширший соціокультурний контекст. Важливо також оцінити ризики, пов'язані з впровадженням нових технологій, включаючи технічні складнощі, питання безпеки, сумісність з іншими системами та зміни в організаційних процесах.

Завдяки детальному і систематичному аналізу цих аспектів, можна розробити чіткі й обґрунтовані рекомендації щодо ефективного використання чат-бота для управління замовленнями.

## 2 ЗАСОБИ ТА ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ TELEGRAM-БОТІВ ДЛЯ УПРАВЛІННЯ ОБРОБКОЮ ЗАМОВЛЕНЬ

### 2.1 Вибір бази даних для створення Telegram-боту

Для розробки Telegram-бота можна застосовувати різні типи баз даних [14, 17, 18]. Реляційні бази даних, як-от MySQL, PostgreSQL, Oracle, є популярним вибором для веб-додатків, оскільки вони добре підходять для зберігання структурованих даних у вигляді таблиць із рядками та стовпцями. Проте реляційні бази даних можуть бути менш ефективними при зберіганні неструктурованих даних, таких як текстові повідомлення, зображення чи відео.

Нереляційні бази даних, як-от MongoDB, Neo4j і Cassandra, краще підходять для роботи з неструктурованими даними. Вони не вимагають жорсткої структури, що робить їх більш гнучкими та масштабованими [18].

Результати порівняння реляційних та нереляційних баз даних наведені у табл. 2.1.

Таблиця 2.1 – Порівняння реляційних та нереляційних баз даних.

Характеристика	Реляційні бази даних	Нереляційні бази даних
Структура даних	Строго структурована	Неструктурована або напівструктурована
Гнучкість	Мене гнучка	Більш гнучка
Масштабування	Мене масштабоване	Більш масштабоване
Ефективність	Більш ефективна для структурованих даних	Більш ефективна для неструктурованих даних

Для розробки Telegram-бота для управління обробкою замовлень оптимальним варіантом є використання нереляційної бази даних, такої як MongoDB.

Це зумовлено тим, що дані, які потрібно зберігати, є неструктурованими або напівструктурованими. Наприклад, база даних може містити відомості про замовлення, такі як номер і дата замовлення, статус, контактні дані клієнта тощо. Гнучка структура MongoDB дозволяє адаптуватися до різних типів інформації без жорстких обмежень [19].

MongoDB є популярною нереляційною базою даних, що чудово підходить для обробки великих обсягів неструктурованих даних, а також є масштабованою та гнучкою, що робить її чудовим вибором для розробки чат-бота, який займатиметься управлінням обробкою замовлень.

Ключові переваги MongoDB для створення Telegram-бота [18]:

- швидкість і ефективність, що критично важливо для чат-ботів, які працюють з великими обсягами запитів;
- масштабованість, яка дозволяє боту легко адаптуватися до збільшення кількості користувачів;
- гнучкість, що забезпечує легке налаштування під потреби користувачів.

База даних MongoDB буде використовуватися для зберігання інформації про замовлення, включно з номером і датою замовлення, його статусом, контактними даними клієнта, а також деталями замовлення, такими як перелік продуктів, кількість, ціна тощо.

Telegram-бот зможе звертатися до MongoDB для відстеження статусу замовлень. Наприклад, якщо клієнт запитає про статус свого замовлення, бот зможе миттєво надати актуальну інформацію, отриману з бази даних. Крім того, Telegram-бот може відправляти клієнтам сповіщення про зміни статусу їхніх замовлень, наприклад, повідомлення про доставку.

Основні переваги MongoDB у контексті управління обробкою замовлень [20]:

- гнучкість: MongoDB дозволяє зберігати неструктуровані та напівструктуровані дані, що робить її ідеальною для таких задач;
- масштабованість: база даних легко підтримує зростання обсягів замовлень;

– ефективність: висока швидкість обробки, важлива для чат-бота, що працює з великим потоком запитів.

Отже, MongoDB є оптимальним вибором для створення чат-бота для управління обробкою замовлень, адже ця база даних швидка, масштабована та гнучка, що робить її придатною для зберігання неструктурованої інформації.

## 2.2 Обґрунтування вибору мови програмування та відповідної програмної платформи для розробки Telegram-бот

Чат-бот – це комп'ютерна програма, що відповідає на вхідні повідомлення, імітуючи людську мову. Чат-боти широко застосовуються в бізнесі для підтримки клієнтів, продажів та маркетингових завдань.

Для створення чат-бота доступні різні мови програмування та платформи, а вибір залежить від конкретних цілей та вимог проєкту.

Нижче описані деякі з найпопулярніших мов програмування для розробки Telegram-ботів.

Python є однією з провідних мов програмування для створення веб-додатків та ботів. Завдяки простому синтаксису та великій спільноті розробників, Python залишається дуже популярним вибором [21]. Це високорівнева, інтерпретована мова, яка має низку переваг для створення Telegram-ботів:

– має легкий для розуміння синтаксис, що сприяє швидкому освоєнню та розробці;

– підтримується великою спільнотою розробників і пропонує багато готових бібліотек та фреймворків, які полегшують процес створення та підтримки ботів.

Для розробки Telegram-ботів на Python існують спеціальні бібліотеки, які спрощують роботу з Telegram API. Наприклад, бібліотека `python-telegram-bot` надає розробникам зручний інтерфейс для створення та налаштування ботів [22].

Python також дозволяє використовувати сторонні бібліотеки та фреймворки для розширення можливостей бота, включаючи обробку мови, аналіз даних, роботу з базами даних тощо.

Основні переваги Python [21]:

- простота синтаксису, що сприяє швидкому навчанню;
- величезна спільнота розробників, яка надає підтримку та ресурси;
- велика кількість бібліотек і фреймворків для розробки чат-ботів;
- можливість підтримки штучного інтелекту.

Недоліки Python [21]:

- може бути повільним для роботи з великими ботами;
- не має такої високої масштабованості, як Java.

Приклади застосування:

- Google Allo;
- Slack;
- Facebook Messenger;
- Amazon Alexa;
- Microsoft Teams.

Недоліками Python для розробки Telegram-ботів є вища потреба у ресурсах порівняно з іншими мовами, а також необхідність у встановленні та налаштуванні залежностей для роботи з Telegram API.

Java є популярною мовою програмування для створення корпоративних застосунків та ботів. Завдяки великій бібліотеці готових компонентів, Java спрощує процес розробки.

Java – це об'єктно-орієнтована мова з широкою спільнотою розробників та багатофункціональною платформою [23]. Її переносимість дозволяє запуском програмам працювати на різних операційних системах без потреби в адаптації коду, що є перевагою для створення чат-ботів на різних платформах.

Java підтримує широкий набір сторонніх бібліотек і фреймворків, як-от Telegram Bots API, що полегшує розробку Telegram-ботів і забезпечує взаємодію з Telegram API.

Ця мова широко використовується у корпоративному програмуванні та веб-сервісах, забезпечуючи обробку великих обсягів даних та роботу у високонавантажених системах. Це особливо корисно, коли Telegram-бот обробляє багато замовлень або запитів одночасно.

Активна спільнота Java і велика база знань роблять її зручною для швидкого вирішення проблем під час розробки [24]. Java також забезпечує високий рівень безпеки, що є важливим для захисту даних користувачів Telegram-бота, завдяки вбудованим засобам безпеки, які допомагають уникнути вразливостей.

Переваги Java [23]:

- велика бібліотека готових компонентів для полегшення розробки;
- масштабованість;
- безпека;
- портативність.

Недоліки Java [23]:

- складна для початківців;
- може бути повільною для великих ботів.

Приклади застосування:

- Google Assistant;
- Amazon Alexa;
- Microsoft Cortana;
- Line;
- WeChat.

Враховуючи ці переваги, використання Java для розробки Telegram-ботів є обґрунтованим рішенням, особливо коли потрібен надійний і потужний бот для керування обробкою замовлень [25].

PHP – одна з найпопулярніших мов програмування для створення веб-додатків і ботів [26, 27]. Вона вирізняється простим синтаксисом і підтримкою великої спільноти розробників.

PHP славиться легкістю та зрозумілістю синтаксису, що робить її чудовим вибором для новачків. Немає необхідності витратити багато часу на освоєння складних мовних конструкцій.

Велика й активна спільнота розробників PHP у всьому світі забезпечує доступ до численних порад, допомоги та готових рішень. Крім того, численні бібліотеки й фреймворки полегшують розробку чат-ботів.

Для PHP також існують спеціальні бібліотеки та SDK, що спрощують інтеграцію з Telegram Bot API, полегшуючи процес створення та налаштування бота для Telegram.

PHP відома високою продуктивністю, що важливо для ботів, особливо при обробці великої кількості запитів у реальному часі.

Якщо ваш Telegram-бот пов'язаний з веб-сайтом або веб-додатком, PHP є відмінним вибором, адже це одна з провідних мов веб-розробки. Це дозволяє легко інтегрувати бота з вашим сайтом і підтримувати цілісність функціоналу.

Переваги PHP [26]:

- проста структура синтаксису, що полегшує навчання;
- велика спільнота розробників, яка надає підтримку і безліч ресурсів;
- різноманіття бібліотек і фреймворків для розробки чат-ботів.

Недоліки PHP [26]:

- може бути повільною для складних ботів;
- менша масштабованість порівняно з Java.

Приклади використання:

- Google Allo;
- Slack;
- Facebook Messenger;
- Amazon Alexa;
- Microsoft Teams.

Таким чином, обираючи PHP для розробки Telegram-бота, отримуємо простий та функціональний інструмент, що дозволяє легко створювати і підтримувати чат-бот для ефективного керування обробкою замовлень.

JavaScript є мовою програмування, широко застосовуваною для створення веб-сторінок та чат-ботів [28]. Вона має зручний синтаксис і інтегрована у веб-браузери.

JavaScript став популярним завдяки своїй внутрішній підтримці в браузерах, що надає йому низку переваг у розробці Telegram-ботів.

Можливість використовувати JavaScript безпосередньо в браузерах робить його оптимальним вибором для створення веб-орієнтованих ботів, таких як Telegram-боти. Це дозволяє користувачам взаємодіяти з ботом без встановлення додаткових програм.

Великий вибір бібліотек і фреймворків спрощує розробку ботів. Наприклад, бібліотека "node-telegram-bot-api" полегшує взаємодію з Telegram API [29].

Проста структура синтаксису дозволяє швидко розібратися з кодом, що пришвидшує процес розробки.

Розвинена спільнота розробників, що означає доступ до відповідей на питання, вирішення проблем, прикладів коду та навчальних ресурсів.

Переваги JavaScript [28]:

- простота вбудовування в веб-сторінки;
- зручний синтаксис;
- широкий вибір бібліотек і фреймворків для розробки чат-ботів.

Недоліки JavaScript [28]:

- може бути складним для розробки більш комплексних ботів;
- менш масштабований порівняно з Java.

Приклади використання:

- Google Allo;
- Slack;
- Facebook Messenger;
- Amazon Alexa;
- Microsoft Teams.

Отже, вибір JavaScript для створення Telegram-ботів є доцільним завдяки його підтримці в браузерах, великому вибору бібліотек, зрозумілому синтаксису та активній спільноті, яка надає підтримку. Ця мова дозволяє розробити зручного та функціонального Telegram-бота, що відповідатиме потребам як користувачів, так і розробників проекту [30].

C++ – мова програмування, яка зазвичай застосовується для створення високопродуктивних додатків і ботів [31, 32]. Її відрізняє потужний синтаксис та можливість прямого керування апаратним забезпеченням.

Розробка на C++ може виявитися більш складною і вимагати більше часу та зусиль для реалізації основних функцій бота в порівнянні з іншими мовами.

Переваги C++ [32]:

- висока продуктивність;
- можливість прямого керування апаратним забезпеченням;
- підтримка штучного інтелекту.

Недоліки C++ [32]:

- складність у вивченні та використанні;
- може виявитися недостатньо швидкою для великих ботів.

Приклади використання:

- Google Allo;
- Slack;
- Facebook Messenger;
- Amazon Alexa;
- Microsoft Teams.

Вибір C++ для створення Telegram-бота може бути виправданим, якщо потрібна висока продуктивність і можливість безпосереднього керування апаратним забезпеченням. Проте, слід враховувати складність розробки на цій мові та необхідність високої компетентності у програмуванні.

Вибір мови програмування для розробки чат-бота залежить від цілей і вимог конкретного проекту. Основні фактори, які варто враховувати:

- легкість у вивченні – певні мови програмування легше засвоюються, що важливо для новачків;
- доступність бібліотек та фреймворків – обрання мови з великою спільнотою розробників відкриває доступ до корисних ресурсів і рішень;
- продуктивність – для ботів, які обробляють великі обсяги даних чи запитів, продуктивність є критичним фактором;
- можливість керування апаратним забезпеченням – деякі мови дозволяють безпосередньо взаємодіяти з апаратною частиною, що необхідно для промислових застосунків або швидких і точних операцій.

Результати порівняння мов для створення чат-ботів представлені в таблиці 2.2.

Для Telegram-бота для управління замовленнями ми обрали Java, яка має такі переваги:

- масштабованість – Java дозволяє з легкістю підтримувати обробку зростаючої кількості запитів;
- надійність – важлива для роботи з критичними даними;
- безпека – забезпечує захист конфіденційної інформації.

Java також пропонує значні переваги для розробників, зокрема:

- велику бібліотеку готових компонентів, що значно спрощує процес розробки;
- широку спільноту розробників – Java має велику та активну спільноту, яка надає підтримку та необхідні ресурси для розробників.

Ось приклад чат-бота, створеного за допомогою Java, з назвою BotName.

BotName – це чат-бот, створений для допомоги клієнтам у замовленні товарів в інтернет-магазині. Його основна функція полягає в тому, щоб клієнти могли вибирати продукти з каталогу, додавати їх до кошика та здійснювати оформлення замовлення. BotName був розроблений за допомогою мови програмування Java. Бот інтегрується з Telegram API для взаємодії з користувачами та з API інтернет-магазину для обробки замовлень.

Таблиця 2.2 – Порівняння мов програмування

Мова програмування	Переваги	Недоліки	Сфера застосування	Популярність	Вартість
Python	Простий синтаксис, велика спільнота розробників	Може бути повільним для великих ботів	Загальна, чат-боти	Популярна	Низька
Java	Велика бібліотека готових компонентів, масштабована	Може бути складним для початківців	Корпоративні застосунки, чат-боти	Дуже популярна	Середня
PHP	Простий синтаксис, велика спільнота розробників	Може бути повільним для великих ботів	Загальна, чат-боти	Популярна	Низька
JavaScript	Просто вбудовувати в веб-сторінки, простий синтаксис	Може бути складним для розробки складних ботів	Веб-сторінки, чат-боти	Дуже популярна	Низька
C++	Високо потужний, дозволяє безпосередньо керувати апаратним забезпеченням	Може бути складним для вивчення і використання	Високопродуктивні застосунки, чат-боти	Популярна	Середня

Основні переваги використання Java для створення чат-бота для управління обробкою замовлень:

а) безпека: Java є однією з найбезпечніших мов програмування, що особливо важливо для ботів, які працюють з конфіденційними даними, такими як інформація про кредитні картки;

б) масштабованість: Java підтримує масштабованість, що дозволяє зручно адаптувати систему для обробки великої кількості замовлень;

3) якість коду: мова програмування Java має багатий набір готових компонентів, що полегшує процес розробки та підвищує ефективність створення ботів.

Також була створена діаграма, що ілюструє популярність різних мов програмування для розробки Telegram-ботів (рис. 2.1).

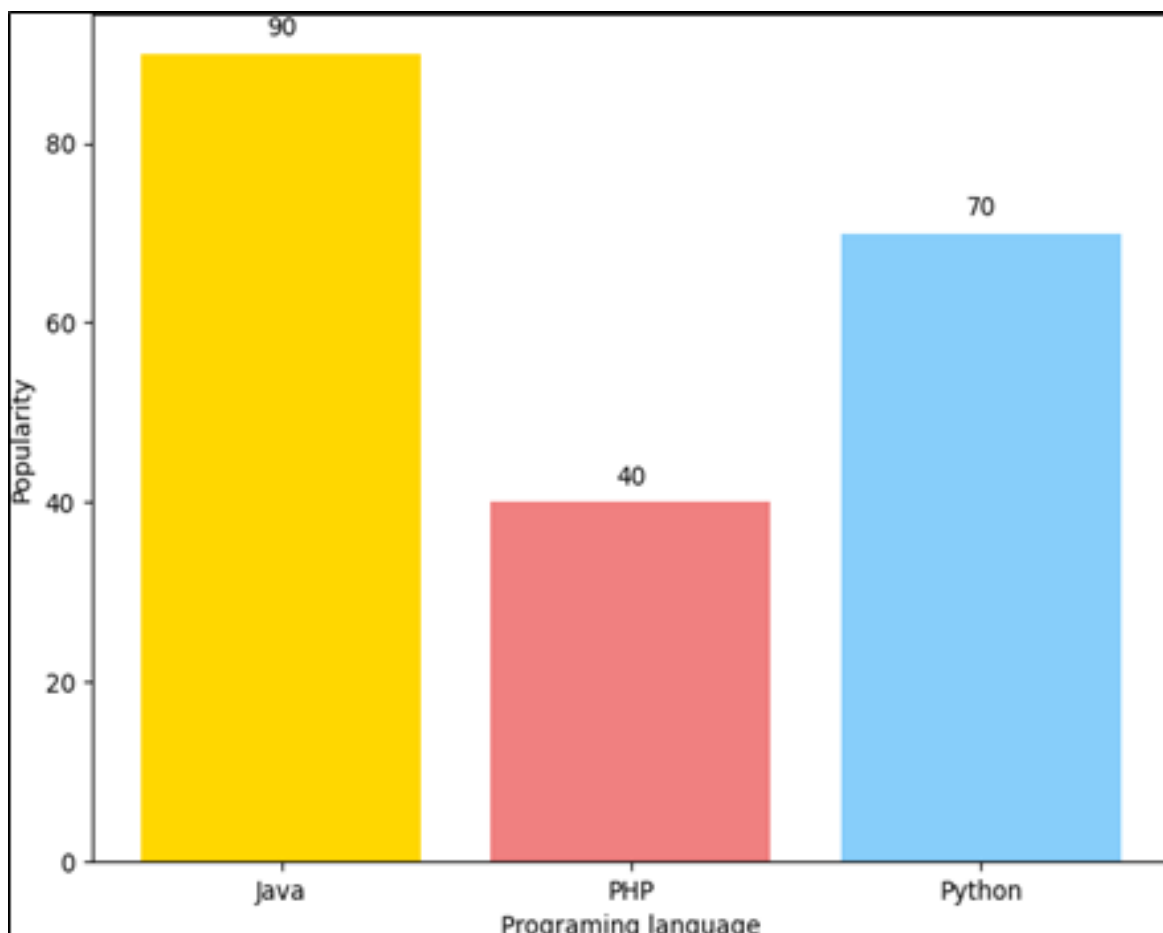


Рисунок 2.1 – Популярність Telegram-ботів і мов програмування

Java є відмінним вибором для створення Telegram-бота для управління обробкою замовлень завдяки своїй масштабованості, надійності, безпеці, великій кількості готових компонентів та активній спільноті розробників.

### 2.3 Проектування алгоритму взаємодії користувача із проєктованим Telegram-ботом

Складний сценарій для ідеального Telegram-бота передбачає обробку замовлень будь-якої складності. Користувач може вибирати товари з каталогу, надавати контактні дані, вибирати спосіб оплати та доставки, а також відслідковувати статус замовлення [10].

Алгоритм взаємодії користувача з складним Telegram-ботом:

- користувач звертається до бота з бажанням оформити замовлення;
- бот запитує у користувача інформацію про товар, який він хоче замовити;
- користувач вказує назву товару, його кількість та розмір;
- бот запитує контактні дані користувача;
- користувач надає своє ім'я, телефон та адресу доставки;
- бот запитує спосіб оплати;
- користувач вказує метод оплати;
- бот підтверджує замовлення та надає користувачу номер;
- бот слідкує за статусом замовлення та повідомляє про зміни.

Спрощений сценарій для Telegram-бота включає тільки базові функції обробки замовлень, такі як вибір товару, вказання кількості та розміру, а також підтвердження замовлення.

Алгоритм взаємодії користувача з спрощеним Telegram-ботом:

- користувач звертається до бота для оформлення замовлення;
- бот запитує інформацію про товар;
- користувач вказує товар та його кількість;
- бот запитує розмір товару;

- користувач вказує розмір;
- бот підтверджує замовлення і надає номер.

Складний сценарій є більш функціональним, дозволяючи користувачам замовляти товари різної складності та відслідковувати статус. Однак для початкового проекту спрощений сценарій виглядає більш реалістичним, даючи змогу зосередитись на основних функціях.

Для реалізації функцій обробки замовлень доцільно використовувати Telegram API, що дозволяє ботам взаємодіяти з сервером, отримувати повідомлення, надсилати відповіді та створювати чати [9]:

- бот отримує повідомлення з даними про замовлення;
- перевіряє коректність інформації;
- створює замовлення в базі даних;
- надсилає користувачу номер замовлення.

Спрощений сценарій є доцільним для початкового етапу, оскільки він дозволяє зосередитись на основних функціях і уникнути витрат на додаткові можливості.

Реалізація відстеження статусу замовлень потребує додаткових технологій, таких як веб-сервіси або API, що може вимагати більше часу і ресурсів. Тому для початкового етапу доцільно реалізувати лише основні функції, як обробка замовлень і надання інформації про їх статус [33].

Спрощений сценарій є доцільним для початкового етапу, оскільки він дозволяє зосередитись на основних функціях і уникнути витрат на додаткові можливості.

Реалізація відстеження статусу замовлень потребує додаткових технологій, таких як веб-сервіси або API, що може вимагати більше часу і ресурсів. Тому для початкового етапу доцільно реалізувати лише основні функції, як обробка замовлень і надання інформації про їх статус [33].

Аргументи на користь спрощеного сценарію:

- спрощений сценарій дозволяє розробити бота швидше та зосередитись на важливих функціях;

– це зменшує витрати часу та ресурсів, дозволяючи зосередитись на основних завданнях.

Діаграма, що пояснює взаємодію користувача з ботом, при відправленні "Привіт" і відповіді "Привіт, як справи", представлена на рис. 2.2.

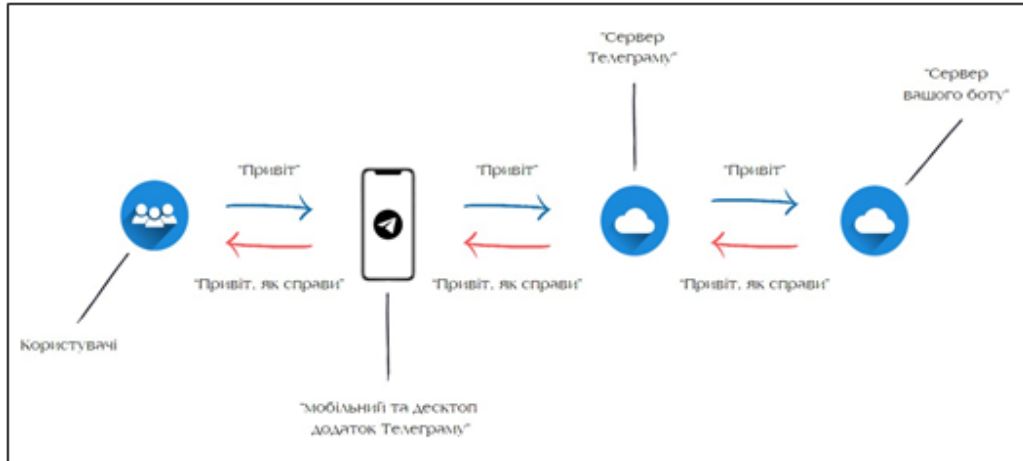


Рисунок 2.2 – Взаємодія між ботом та користувачем

Нижче представлена спрощена схема алгоритму реалізації чат-бота (рис. 2.3). Більш детальну схему роботи Telegram-бота буде представлено в третьому розділі, згідно з реалізованою функціональністю.

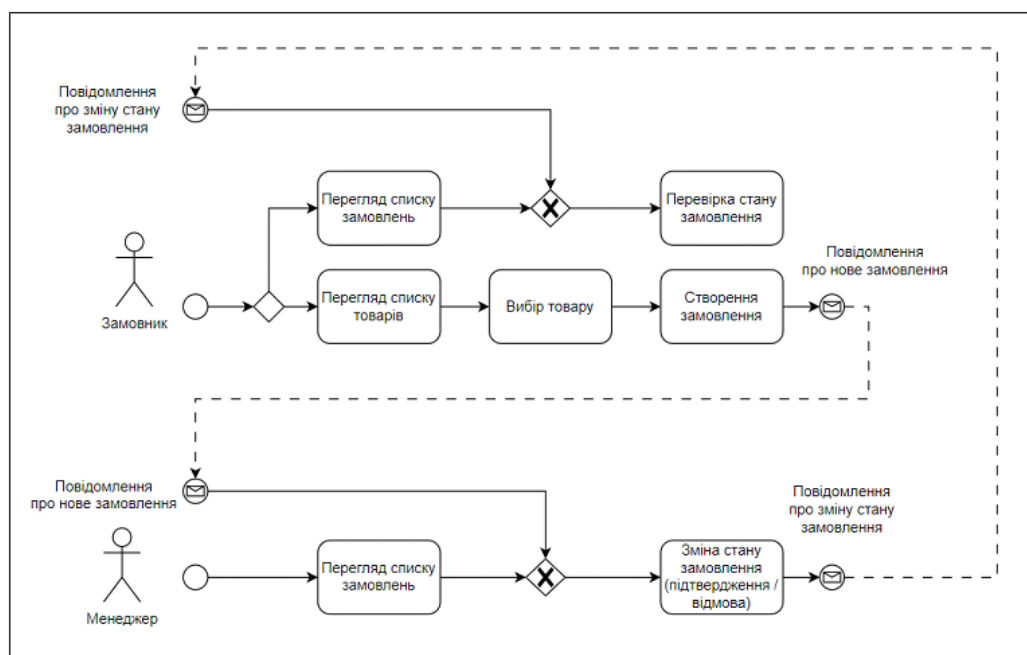


Рисунок 2.3 – Алгоритм реалізації чат-бота

## Висновки за розділом 2

У цьому розділі було охарактеризовано основні інструменти та технології, що застосовуються при розробці Telegram-ботів для управління обробкою замовлень. Розглянуто різноманітні типи баз даних для зберігання даних бота. Реляційні бази даних є оптимальними для зберігання структурованої інформації, такої як таблиці з рядками та стовпцями, в той час як нереляційні бази краще підходять для зберігання неструктурованих даних, таких як текстові повідомлення, зображення і відео.

Також було проведено огляд найбільш популярних мов програмування для розробки чат-ботів, серед яких Python, Java, PHP, JavaScript та C++. Кожна з них має свої переваги та підходить для певних завдань у розробці бота.

Описано алгоритм взаємодії користувача з чат-ботом для обробки замовлень. Складний сценарій передбачає можливість обробки замовлень різного рівня складності, де користувач може замовляти товари з каталогу, надавати свої контактні дані, обирати спосіб оплати та доставки. Крім того, бот повинен забезпечувати можливість відстеження статусу замовлення.

Отже, можна зробити висновок, що чат-боти є ефективним інструментом для автоматизації процесів та підвищення якості обслуговування клієнтів. Вони здатні виконувати різні функції, включаючи приймання замовлень, відповідь на запитання клієнтів і відстеження стану замовлень.

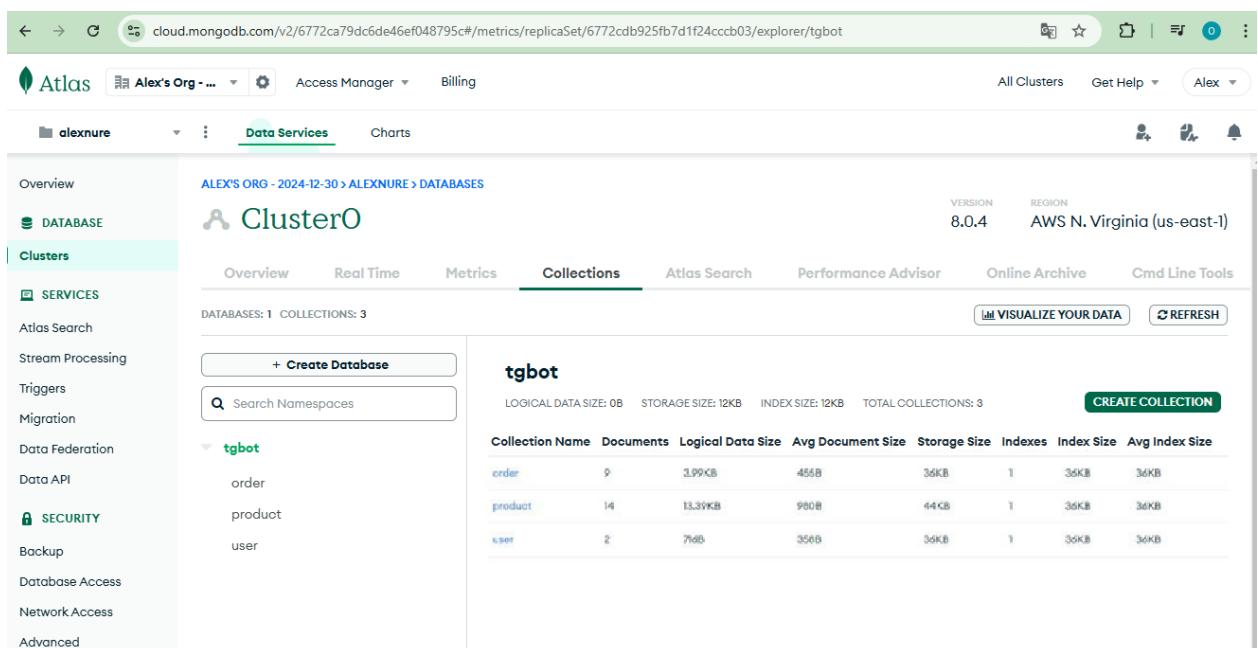
Для реалізації чат-бота, який займається обробкою замовлень, необхідно обрати відповідну базу даних, мову програмування та платформу, враховуючи специфіку проекту та вимоги до його розробки.

## 3 СТВОРЕННЯ ЧАТ-БОТУ ДЛЯ УПРАВЛІННЯ ОБРОБКОЮ ЗАМОВЛЕНЬ: АСПЕКТИ РЕАЛІЗАЦІЇ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

### 3.1 Створення бази даних для Telegram боту з використанням СКБД MongoDB

Для зберігання даних, необхідних для функціонування бота, обрано сервіс MongoDB Cloud Services [34]. Цей сервіс надає безкоштовний доступ до повного спектра можливостей бази даних MongoDB, включаючи створення колекцій, керування правами доступу, створення резервних копій, реплікацію, веб-доступ до даних, а також моніторинг системи.

Після реєстрації у сервісі був створений проєкт із назвою "alexnure" і база даних "tgbot" (рис. 3.1).



The screenshot displays the MongoDB Atlas web interface. The browser address bar shows the URL: `cloud.mongodb.com/v2/6772ca79dc6de46ef048795c#/metrics/replicaSet/6772cdb925fb7d1f24cccb03/explorer/tgbot`. The interface includes a navigation menu on the left with sections for Overview, DATABASE, Clusters, SERVICES, and SECURITY. The main content area shows the 'Cluster0' overview for 'ALEX'S ORG - 2024-12-30 > ALEXNURE > DATABASES'. It indicates 'VERSION 8.0.4' and 'REGION AWS N. Virginia (us-east-1)'. Below this, there are tabs for Overview, Real Time, Metrics, Collections, Atlas Search, Performance Advisor, Online Archive, and Cmd Line Tools. The 'Collections' tab is active, showing 'DATABASES: 1' and 'COLLECTIONS: 3'. A search bar for namespaces is present. The 'tgbot' database is expanded, showing a list of collections: 'order', 'product', and 'user'. A table below the list provides details for each collection:

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
order	9	3.99KB	455B	36KB	1	36KB	36KB
product	14	13.39KB	980B	44KB	1	36KB	36KB
user	2	716B	358B	36KB	1	36KB	36KB

Рисунок 3.1 – База даних tgbot створена на сервері Atlas

Сторінка моніторингу ресурсів бази даних наведена на рис. 3.2.



Рисунок 3.2. – Сторінка моніторингу ресурсів бази даних

### 3.1.1. Створення колекцій

Для зберігання даних, необхідних для роботи бота, планується створити такі колекції:

- user – містить інформацію про користувачів, а також додаткові дані, пов'язані з режимом обробки їхніх повідомлень;
- product – включає інформацію про доступні товари, зокрема назву підрозділу (department), за яким визначається менеджер, відповідальний за обробку замовлення певного товару;
- order – містить дані про замовлення, включаючи посилання на користувача, який його зробив, і менеджера, що займається його обробкою.

Детальний опис кожної з колекцій наведено в табл. 2.3 – 2.5. Усі дані зберігаються у текстовому форматі.

Таблиця 2.3 – Структура колекції user

Поле	Опис	Значення
_id	Ідентифікаційний номер документу в колекції	UUID
telegramId	Ідентифікаційний номер користувача в телеграм	Число
chatId	Ідентифікаційний номер чату	Число
firstName	Ім'я користувача під яким він зареєстрований в Telegram	
lastName	Додаткове ім'я користувача під яким він зареєстрований в Telegram	
userName	Логін користувача під яким він зареєстрований в Telegram	
roles	Список ролей	“admin”, “manager”, “customer”
delivery	Адреса доставки	Може бути пустим
departments	Список підрозділів (груп товарів) за які відповідає даний користувач якщо він є менеджером.	Для менеджера: “Electronics”, “Appliances”, “Cloth”, “Furniture”.
messageMode	Режим обробки повідомлень (команда)	“/search”, “/apply”
messageData	Вміст повідомлення переданого в режимі “/apply”	

Таблиця 2.4 – Структура колекції product

Поле	Опис	Значення
_id	Ідентифікаційний номер документу в колекції	UUID
productCode	Код товару в магазині	Число
productName	Назва товару	
description	Короткий опис товару	
details	Детальний опис товару	
department	Підрозділ (група товарів)	“Electronics”, “Appliances”, “Cloth”, “Furniture”
managerId	Ідентифікаційний номер менеджера (“_id”) відповідального за дану групу товарів	UUID
price	Ціна товару	Число з двома знаками після коми
available	Наявність товару	“true”, “false”
images	Список посилань на файли зображення товару	Приклад : [“https://resource.com/photo1”]
	Вміст повідомлення передано-го в режимі “/apply”	

Таблиця 2.5 – Структура колекції order

Поле	Опис	Значення
_id	Ідентифікаційний номер документу в колекції	UUID
productCode	Код товару в магазині	Число
productName	Назва товару	
amount	Кількість	Число
price	Ціна товару	Число з двома знаками після коми
sum	Сума	Число з двома знаками після коми
delivery	Адреса доставки	Може бути пустим
status	Стан замовлення	"created" - підготовлено, "confirmed" - підтверджено замовником (очікує обробки менеджером), "approved" - підтверджено менеджером, "declined" - відхилено менеджером, "delivery" - доставляється замовнику, "completed" - завершено,
manager	Ідентифікаційний номер менеджера ("_id") відповідального за дану групу товарів	UUID

Продовження таблиці 2.5

Поле	Опис	Значення
customer	Ідентифікаційний номер користувача в телеграм (“telegramId”)	Число
created	Дата та час створення замовлення (з використанням часової зони)	Приклад: "2024-12-30T12:26:40.4209116+03:00"
updated	Дата та час зміни стану замовлення (з використанням часової зони)	Приклад: "2024-12-30T14:41:34.1783047+03:00"

Кожна колекція буде створюватися автоматично під час виконання першої команди додавання документа. Однак деякі дані потребують попередньої підготовки, зокрема інформація про товари для колекції product.

Для того щоб Telegram-бот міг надавати більш детальну інформацію про доступні товари, крім текстового опису, необхідно додати графічні зображення. Telegram пропонує різні способи додавання зображень до повідомлень, зокрема найпростіший – використання посилань на зовнішні ресурси, де зберігаються файли зображень.

Для зберігання зображень товарів було обрано сервіс Shutterstock [35]. Після реєстрації на платформі Shutterstock були завантажені файли із зображеннями товарів, а також отриманий список посилань на кожне зображення (рис. 3.3).

Для додавання чи редагування даних в БД використовуються такі інструменти:

- Compass: графічний інтерфейс доступу до бази даних MongoDB;
- MongoDB Shell : інтерфейс командної строки доступу до бази даних MongoDB (mongosh).

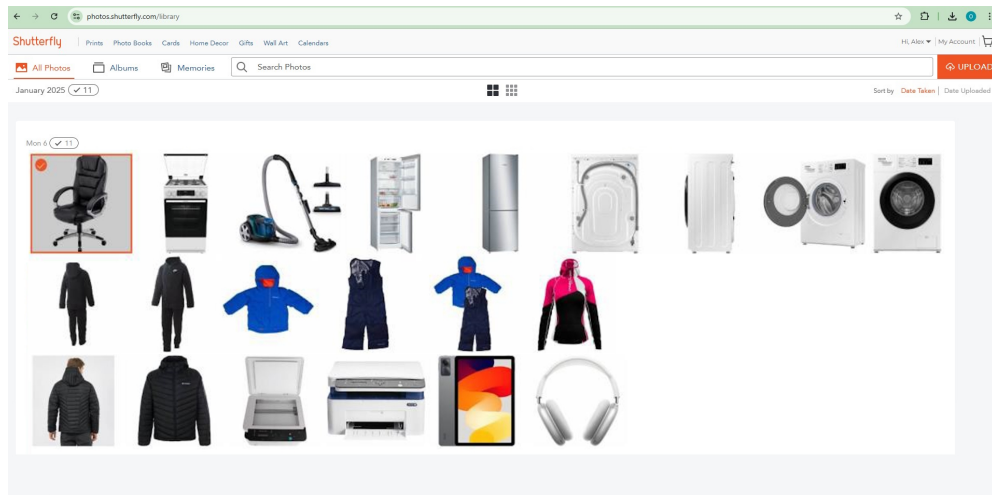


Рисунок 3.3 – Архів зображень товарів на сервері Shutterfly

Використання додатку Compass наведено на рис. 3.4.

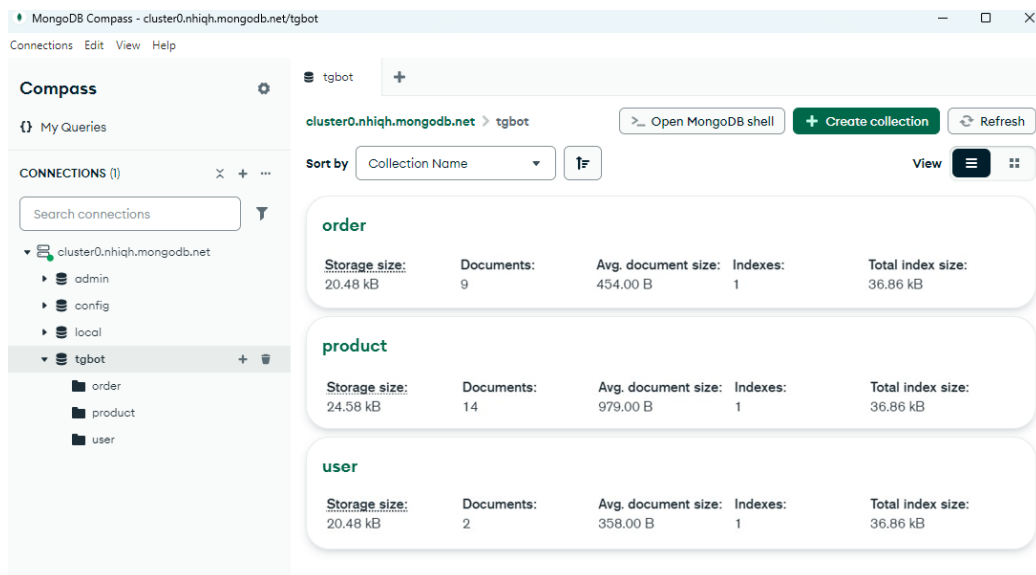


Рисунок 3.4 – Використання додатку Compass

Приклад використання інтерфейсу командної строки наведено на рис. 3.5.

```

> use tgbot
< 'switched to db tgbot'
> db.getCollectionNames()
< [ 'order', 'product', 'user' ]
Atlas atlas-srwlyj-shard-0 [primary] tgbot >
  
```

Рисунок 3.5 – Приклад використання MongoDB Shell

Детальну інформацію про налаштування доступу до бази даних можна знайти на сервері в розділі Data Services - Database Deployments - Connect (рис.3.6, рис.3.7).

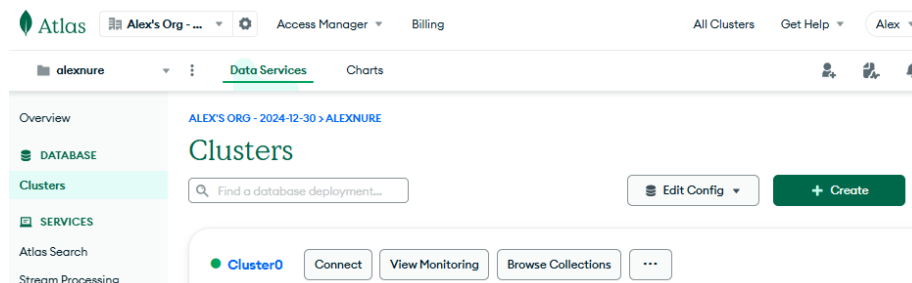


Рисунок 3.6 – Доступ до інформації про підключення

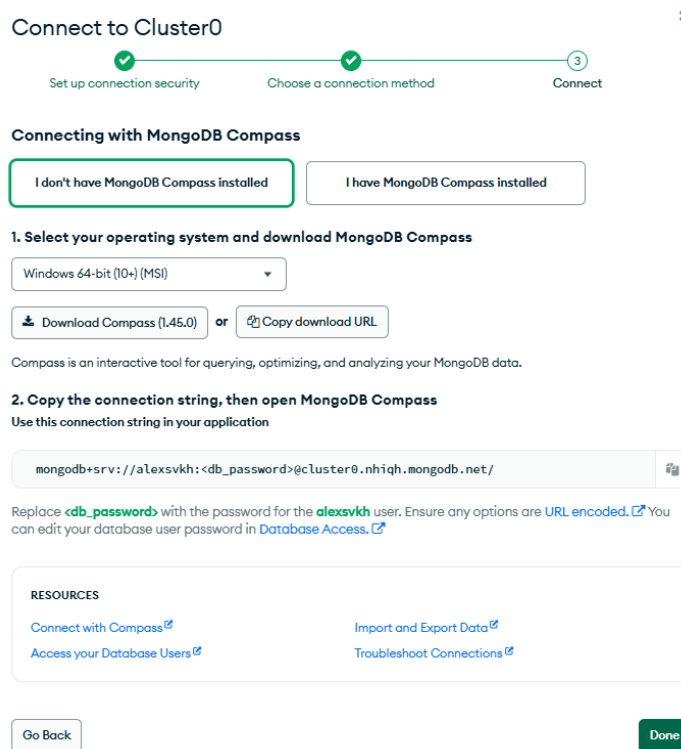


Рисунок 3.7 – Інформація про налаштування доступу

Щоб наповнити колекцію product даними про товари, створюється скрипт, який додає до бази даних окремий документ для кожного товару. Цей скрипт зберігається у файлі fill\_data.sh, розташованому в директорії data репозиторію з кодом програми. До кожного документа також додається список посилань на зображення товарів, підготовлених на попередньому етапі. Для дода-

вання документів у базу даних потрібно виконати скрипт за допомогою наступної команди:

```
Bash ./fill_data.sh mongodb+srv://alexdiplom:<password>  
@alexnure.cbqv0wk.mongodb.net/admin?authSource=admin.
```

Після завершення попередніх налаштувань база даних готова до використання в розробці та експлуатації Telegram-бота, за винятком необхідності подальшого редагування інформації про менеджерів (деталі цього процесу будуть розглянуті у наступних розділах).

### 3.2. Розробка Telegram-бота з використанням Java

Для створення Telegram-бота було обрано наступні технології, інструменти та бібліотеки:

- IntelliJ IDEA – інтегроване середовище розробки (IntelliJ IDEA 2024.3.1.1) [56];

- Java 11 – об'єктно-орієнтована мова програмування (OpenJDK 11.0.15 2022-04-19 LTS) [37];

- org.telegram.telegrambots – бібліотека для створення телеграм ботів на Java (версія 6.5.0) [11];

- Spring Boot – розширення Spring, що використовується для створення Java програм на основі мікросервісів (версія 3.4.1) [38];

- Lombok – Java бібліотека і процесор анотацій який яка полегшує написання програмного коду (версія 1.18.28) [39];

- Apache Maven – засіб автоматизації роботи з програмними проектами (версія 3.9.9) [40];

- Git – розподілена система керування версіями файлів та спільної роботи (версія 2.47.1.windows.1) [41];

– Docker – інструментарій для управління ізольованими Linux-контейнерами [42].

У процесі розробки додатка алгоритм роботи Telegram-бота було оновлено для врахування ключових функцій. Оновлений алгоритм подано на рис. 3.8. Під час створення Telegram-бота використовувалася система керування версіями Git, яка забезпечує контроль змін програмного коду. Усі зміни завантажувалися на платформу Bitbucket. Програмний код можна завантажити за посиланням “<https://bitbucket.org/nure-diplom-o/tgbot/src/main/>”.

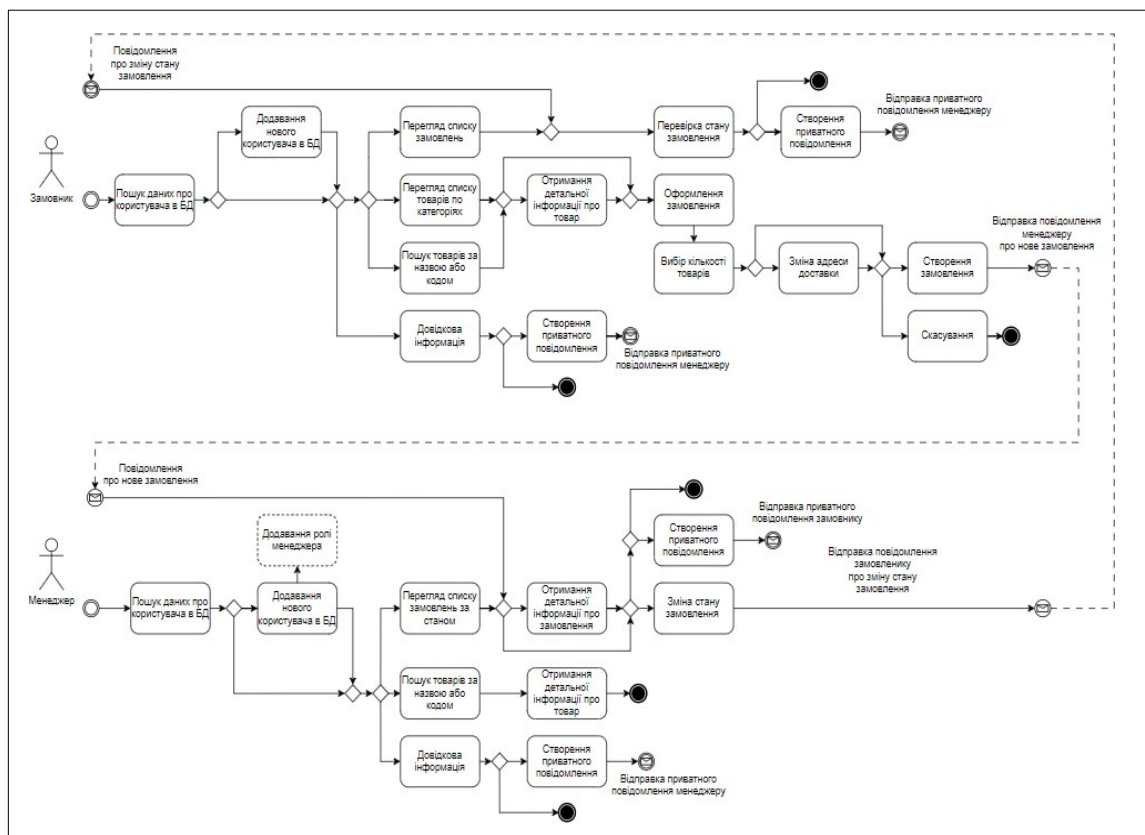


Рисунок 3.8 – Алгоритм роботи Телеграм боту

Код додатку згруповано по наступних пакетах:

- model – класи які описують колекції в БД (user, product, order);
- repository – інтерфейси доступу к БД з використанням бібліотеки spring-boot-starter-data-mongodb;
- service.telegram.bot – базовий клас який реалізує інтеграцію з Telegram а також контейнер репозиторіїв;

- `service.telegram.config` – класи для конфігурації використання додатку у якості телеграм-боту;
- `service.telegram.processor` – окремі класи для обробки кожної команди або повідомлення;
- `service.telegram.util`, `service.telegram.processor.util` – допоміжні класи які містять статичні дані – коди емоджі, найменування станів замовлень та груп товарів, методи для створення замовлень;
- `service.telegram.time` – класи які використовуються для формування значення поточного часу в замовленнях.

Загальну структуру проекту наведено на рис. 3.9.

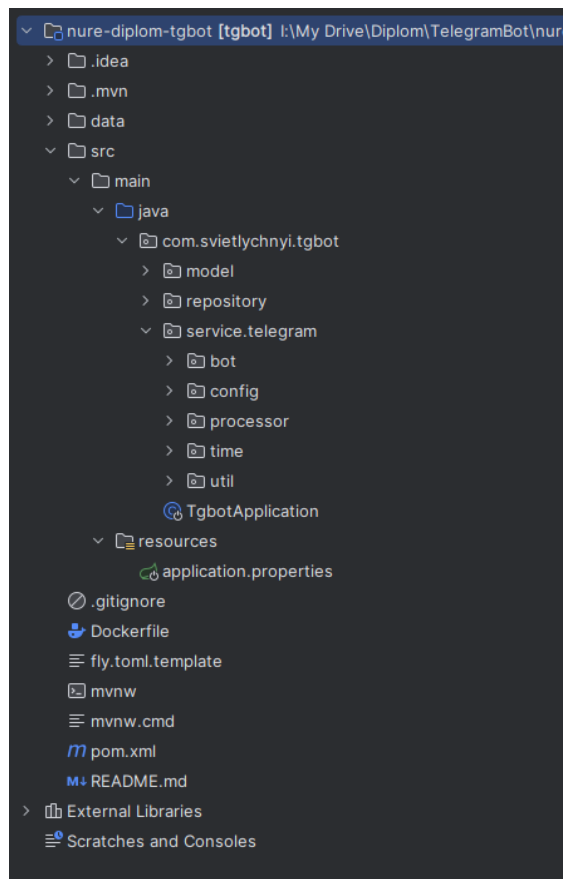


Рисунок 3.9 – Структура проекту

Для запуску додатка у ролі Telegram-боту було виконано реєстрацію згідно рекомендацій на офіційному сервері телеграм “<https://core.telegram.org/bots/tutorial>” [11, 43] після чого отримано токен доступу.

У файлі README.md містяться вказівки щодо запуску програми як стандартного Java-додатка на локальному комп'ютері або у вигляді Docker-контейнера. Для налаштування запуску необхідно вказати ім'я та токен, отримані під час реєстрації бота, а також ім'я користувача і пароль для доступу до бази даних, які були встановлені під час її створення.

Для тестування функціоналу бота з можливістю покрокового контролю виконання програми потрібно налаштувати параметри запуску в інтегрованому середовищі розробки IntelliJ IDEA (див. рис. 3.10).

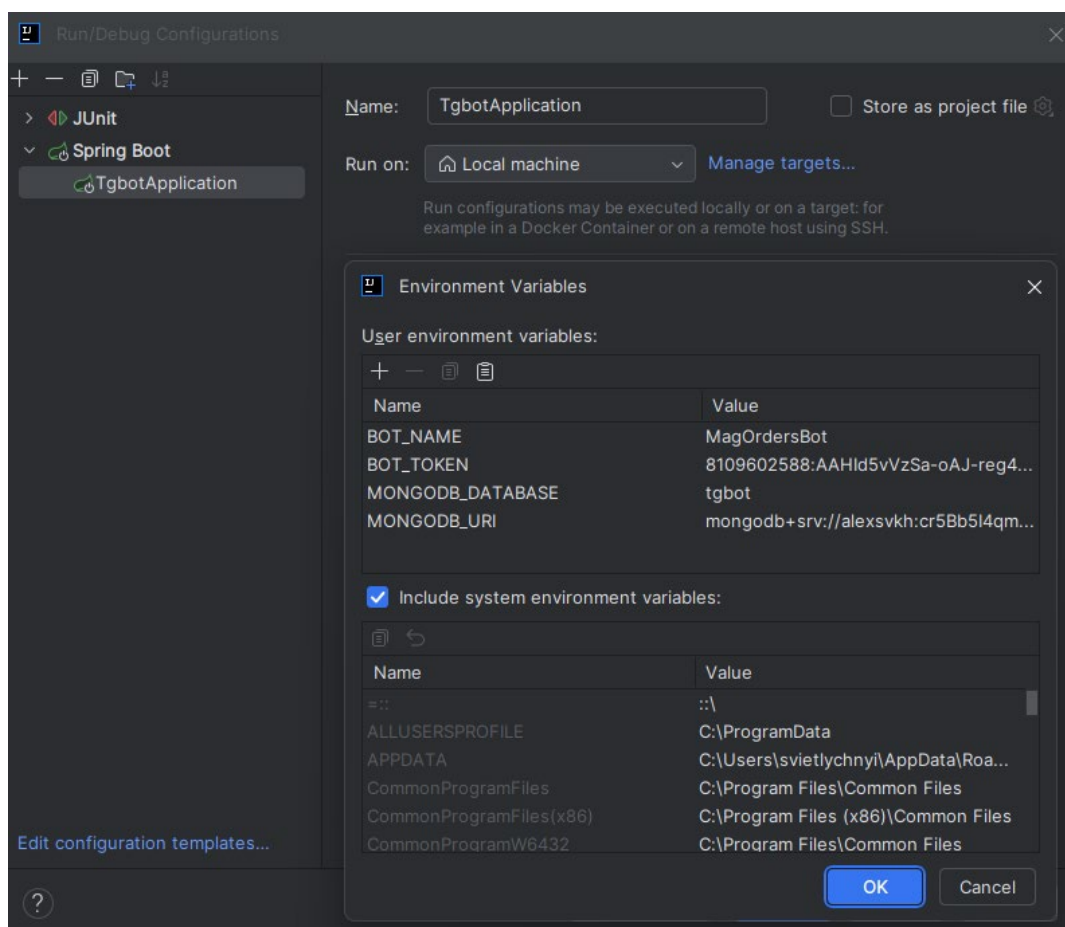


Рисунок 3.10 – Налаштування запуску в середовищі розробки

Під час розробки Telegram-бота було впроваджено механізм автоматичної реєстрації нових користувачів, які за замовчуванням додаються в базу даних із роллю "замовник". Проте для повноцінної роботи необхідно зареєструвати від

одного до чотирьох користувачів, які виконуватимуть роль менеджерів. Один менеджер може відповідати за кілька підрозділів.

Кроки для реєстрації менеджерів:

а) запустити телеграм бот;  
 б) користувач (менеджер) повинен знайти бота в Telegram за назвою shops\_orders\_bot і підключитися до нього, натиснувши кнопку "Старт". Новий користувач буде автоматично доданий до бази даних;

в) адміністратор Telegram-бота повинен знайти запис нового користувача в базі даних та внести зміни:

- 1) змінити роль ("roles") з "customer" на "manager";
- 2) додати від одного до чотирьох назв підрозділів з списку: "Electronics", "Appliances", "Cloth", "Furniture".

Приклад внесення змін для реєстрації менеджера за допомогою інструменту Compass показано на рис. 3.11.

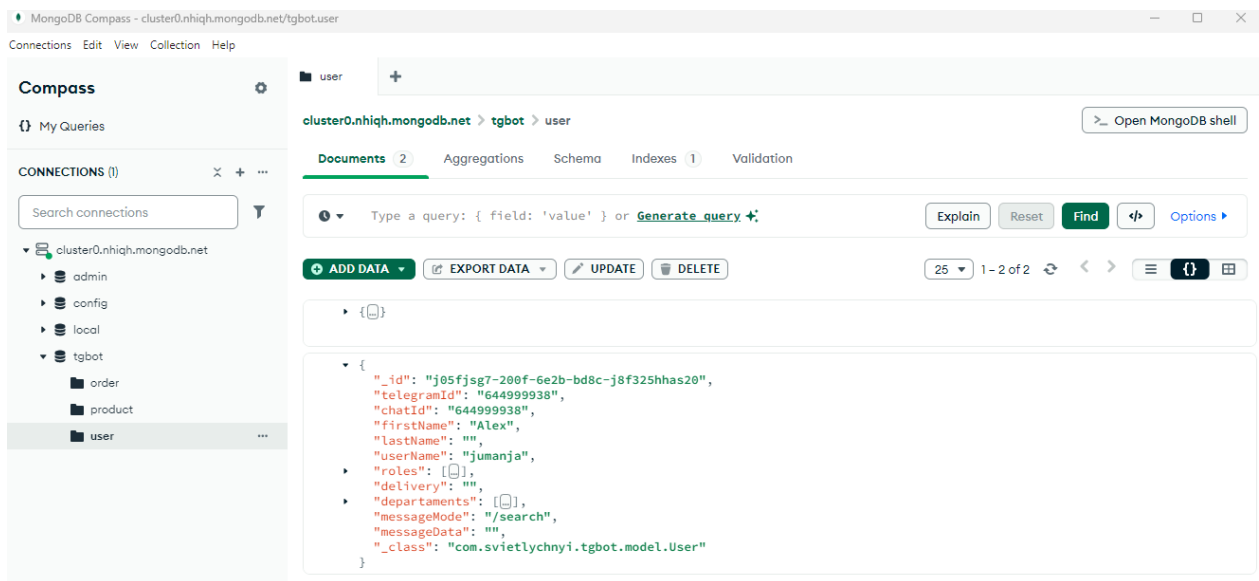


Рисунок 3.11 – Реєстрація менеджера

Також Telegram-бот підтримує роль адміністратора ("admin"), яку можна призначити одному з менеджерів. Функції адміністратора реалізовані в мінімальному обсязі для цілей тестування.

Після завершення реєстрації менеджерів Telegram-бот готовий до роботи. Нові користувачі можуть підключатися до бота, шукати товари, оформлювати замовлення, переглядати їхній статус і отримувати сповіщення про зміни. Менеджери можуть отримувати повідомлення про нові замовлення, переглядати їх та змінювати статуси.

### 3.3. Впровадження постачання розробленого Telegram-боту

Для забезпечення стабільної роботи Телеграм боту додаток необхідно використати хмарні сервіси. Існує багато хмарних сервісів які забезпечують надійну роботу додатків та надають безліч додаткових послуг- моніторинг, збір статистики, масштабування, налаштування доступу, архівування, автоматичне розгортання та інше. Найбільш відомі з них: Amazon Web Services [44], Microsoft Azure [45], Google Cloud [46].

У рамках цієї кваліфікаційної роботи для розгортання бота обрано сервіс Fly.io [47]. Хоча цей сервіс має обмежені можливості автоматизації та моніторингу, його можна використовувати безкоштовно, якщо додаток не потребує значних ресурсів. У межах безкоштовного плану доступно до трьох віртуальних машин із 256 МБ оперативної пам'яті. Це робить Fly.io придатним для невеликих проєктів із обмеженою кількістю користувачів.

Для розгортання на Fly.io в проєкті було створено файл Dockerfile для підготовки Docker-контейнера додатка. Також додано файл fly.toml, що містить параметри підключення до сервісу. Через конфіденційність даних, пов'язаних із доступом до бази даних і Telegram, цей файл не зберігається в репозиторії. Однак надано шаблон fly.toml.template для створення конфігурації.

Для роботи з сервісом Fly.io використовується інтерфейс командної строки- утиліта flyctl яку можна завантажити з сервера “<https://fly.io/docs/hands-on/install-flyctl/>”. Також на сервері є розділ з докладною документацією і рекомендаціями по роботі з сервісом “<https://fly.io/docs/>”.

Перед розгортанням додатку необхідно виконати компіляцію за допомогою команди Maven:

```
mvn clean install.
```

Результат компіляції коду програми за допомогою команди Maven наведено на рис. 3.12.

```
[INFO] --- install:2.5.2:install (default-install) @ tgbot ---
[INFO] Installing C:\nure\program\tgbot\target\tgbot-0.0.1.jar to C:\Users\svietlychnyi\.m2\repository\com\svietlychnyi\tgbot\0.0.1\tgbot-0.0.1.jar
[INFO] Installing C:\nure\program\tgbot\pom.xml to C:\Users\svietlychnyi\.m2\repository\com\svietlychnyi\tgbot\0.0.1\tgbot-0.0.1.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.992 s
[INFO] Finished at: 2025-11-05T23:45:59+02:00
[INFO] -----
```

Рисунок 3.12 – Результат компіляції коду програми

Далі запускаємо збірку та розгортання додатку на сервері за допомогою утиліти flyctl:

```
fly launch --now --build-arg JAR_FILE=target/tgbot-0.0.1.jar.
```

Приклад запуску та розгортання додатку наведено на рис. 3.13 та рис. 3.14.

```
$ fly launch --now --build-arg JAR_FILE=target/tgbot-0.0.1.jar
Creating app in C:\nure\program\tgbot
An existing fly.toml file was found for app orders-tg-bot
? Would you like to copy its configuration to the new app? Yes
Scanning source code
Detected a Dockerfile app
? Choose an app name (leaving blank will default to 'orders-tg-bot')
? Choose an app name (leaving blank will default to 'orders-tg-bot')
automatically selected personal organization: Andrii
App will use 'waw' region as primary

Created app 'orders-tg-bot' in organization 'personal'
Admin URL: https://fly.io/apps/orders-tg-bot
Hostname: orders-tg-bot.fly.dev
? Create .dockerignore from 2 .gitignore files? No
Wrote config file fly.toml
Validating C:\nure\program\tgbot\fly.toml
Platform: machines
✓ Configuration is valid
```

Рисунок 3.13 – Запуск та розгортання додатку на сервері

```

The push refers to repository [registry.fly.io/orders-tg-bot]
866e133cb81c: Pushed
772ca119eb87: Pushed
cc2447e1835a: Pushed
deployment-01HEGQ8E9K07Z0GZ6541W8K450: digest: sha256:7ba1eb918637095e56c57adf1461a16066e8220dc5946e6c6983e74721934342 size: 953
-> Pushing image done
image: registry.fly.io/orders-tg-bot:deployment-01HEGQ8E9K07Z0GZ6541W8K450
image size: 303 MB

Watch your deployment at https://fly.io/apps/orders-tg-bot/monitoring

Provisioning ips for orders-tg-bot
Dedicated ipv6: 2a09:8280:11:4e:9ea4
Shared ipv4: 66.241.125.157
Add a dedicated ipv4 with: fly ips allocate-v4

This deployment will:
  create 2 "app" machines

No machines in group app, launching a new machine

-----
✓ Machine 3d8d544c4d3089 [app] update finished: success
WARNING The app is not listening on the expected address and will not be reachable by fly-proxy.
You can fix this by configuring your app to listen on the following addresses:
  - 0.0.0.0:8080
Found these processes inside the machine with open listening sockets:
-----
PROCESS | ADDRESSES
-----
/.fly/hallpass | [fdaa:2:7c8a:a7b:84:17a8:9938:2]:22

Creating a second machine to increase service availability
Finished launching new machines

-----
✓ Machine 918571e0f69e08 [app] update finished: success
-----

Visit your newly deployed app at https://orders-tg-bot.fly.dev/

```

Рисунок 3.14 – Завершення розгортання додатку на сервері

Залежно від конфігурації сервера додаток може бути розгорнутий на двох віртуальних машинах. Однак на цей момент додаток не підтримує масштабування, що може спричинити помилки через доступ до Telegram двох екземплярів із однаковим токеном. У такій ситуації необхідно тимчасово вимкнути всі віртуальні машини, а потім увімкнути лише одну, використовуючи команди:

```

fly scale count 0,
fly scale count 1.

```

Після завершення розгортання можна переглянути ресурси сервера (рис. 3.15), перевірити стан віртуальних машин (рис. 3.16) та переглянути логи програми (рис. 3.17).

Після успішного розгортання Telegram-бот готовий до використання. У наступному розділі будуть наведені приклади застосування розробленого бота для управління обробкою замовлень.

The screenshot shows the 'Overview' page for the application 'orders-tg-bot'. At the top, there are four performance metrics: Average Data In (1 kB/s), Average Data Out (297 B/s), Average Memory Used (194 MB/228 MB), and Average Load (0.11). Below these are sections for Application Information (Hostname, App Type, Process groups, Machine Sizes, IP addresses), Activity (Release by Oleksandr), Fly Platform Status, and App Status (Deployed).

Рисунок 3.15 – Загальна інформація про ресурси

The screenshot shows the 'Machines' page for the application 'orders-tg-bot'. A warning message states: 'Your app is currently running on a single machine. To ensure high availability, especially for production apps, we'd strongly recommend running at least 2 instances. You can scale up the number of machines using the fly machine clone command in the CLI. Check out the documentation for more details.' Below this is a table of machines:

Name	ID	Region	State	Image	Size/CPU
rough-darkness-4827	918571e0f69e08	waw	destroyed	orders-tg-bot:deployment-01HEGQ8E9K07Z0G26541W8K450	shared-cpu-1x
black-pond-8679	4d89799a3d5087	waw	started	orders-tg-bot:	shared-cpu-1x
sparkling-glitter-4055	3d8d544c4d3089	waw	destroyed	orders-tg-bot:deployment-01HEGQ8E9K07Z0G26541W8K450	shared-cpu-1x

Рисунок 3.16 – Інформація про стан віртуальних машин

The screenshot shows the 'Monitoring' page for the application 'orders-tg-bot'. It displays a log stream with MongoDB and application logs. Below the logs is a table of machines:

ID	Name	Region	Process Group	Image	State	Checks	Created
4d89799a3d5087	black-pond-8679	waw	app	registry.fly.io/orders-tg-bot	started	Setup checks	Nov 5, 2023 21

Рисунок 3.17 – Доступ до логів програми

### Висновки за розділом 3

У процесі створення Telegram-бота для управління обробкою замовлень було акцентовано увагу на виборі технологій з точки зору їхньої гнучкості, надійності та перспективності. Важливим критерієм також стала безкоштовність використання технологій, що дозволяє мінімізувати витрати, спрямовуючи кошти лише на розробку. Telegram як платформа знижує загальну собівартість проекту, оскільки більшість функцій інтерфейсу користувача надаються безпосередньо платформою.

Розроблений Telegram-бот реалізує обмежений, але достатній набір функцій для ефективного управління обробкою замовлень. Обрані технології залишають простір для подальшого вдосконалення та розширення функціональності програми.

Використання нереляційної бази даних MongoDB забезпечує високу гнучкість у роботі зі структурою даних, що дозволяє легко додавати нові групи даних. Крім того, нереляційні бази даних краще адаптуються до масштабування, що може стати критичним за умови збільшення кількості користувачів, товарів чи функцій.

Мова програмування Java також надає значні переваги у плані розвитку та удосконалення програми. Заздалегідь складно спрогнозувати, які саме функції можуть стати необхідними в майбутньому, однак використання Java гарантує, що бот можна легко адаптувати до нових вимог та інтегрувати з іншими системами.

Під час розробки Telegram-бота були ідентифіковані окремі обмеження самої платформи Telegram. Проте доступних функцій цілком достатньо для реалізації не лише системи обробки замовлень, але й базового функціоналу інтернет-магазину.

Бот також демонструє потенціал для застосування в управлінні бізнес-процесами. Його функціонал можна адаптувати не тільки для великих інтернет-магазинів, а й для управління будь-якими підприємствами. Telegram-бот дозво-

ляє автоматизувати процеси, пов'язані з надсиланням повідомлень учасникам складних бізнес-процесів, зокрема виконавцям проміжних завдань. Така автоматизація сприяє зниженню собівартості процесів та підвищенню їхньої надійності.

## 4. РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

### 4.1 Тестування розробленого Telegram-боту

Для тестування роботи Телеграм боту використовувалось два аккаунта:

- “kandriiko” – користувач який виконує роль замовника;
- “zzmij” – користувач який виконує роль менеджерів для всіх чотирьох підрозділів.

Під час створення Telegram-бота тестування та налагодження проводилися на комп’ютері розробника. Кожна нова частина додатка проходила окремий етап тестування. Після завершення розробки кожного блоку функцій вони запускалися та перевірялися на сервері, використовуючи хмарний сервіс Fly.io [64].

Закінчивши розробку Telegram-бота, його було зареєстровано в Telegram під назвою “shops\_orders\_bot”. Додаток розгорнули на сервері Fly.io, де було виконано тестування всіх функцій, доступних як для замовників, так і для менеджерів. Нижче наведено перелік протестованих функцій із відповідними ілюстраціями. На представлених зображеннях користувач із роллю замовника позначений зеленим фоном, а менеджер – блакитним.

Замовник:

- реєстрація нового користувача (рис. 4.1);
- отримання довідкової інформації (рис. 4.2);
- вибір категорії товарів (рис. 4.3);
- перегляд товарів обраної категорії, відображається по три одиниці товарів за один раз (рис. 4.4);
- зміна сторінки відображення списку товарів (рис. 4.5);
- перегляд детальної інформації про товар (рис. 4.6);
- створення замовлення (рис. 4.7);
- зміна кількості одиниць товару (рис. 4.8);

- оформлення замовлення (рис. 4.9);
- зміна адреси доставки (рис. 4.10);
- скасування оформлення замовлення (рис. 4.11);
- пошук замовлення за станом, відображається до трьох замовлень за один раз (рис. 4.12);
- перегляд інформації про замовлення (рис. 4.13);
- підтвердження створення замовлення (рис. 4.14);
- автоматична відправка повідомлення менеджеру про створене замовлення (рис. 4.14);
- отримання повідомлення про зміну стану замовлення (рис. 4.20, 4.21, 4.22, 4.23, 4.24);
- пошук товару за назвою (рис. 4.15);
- пошук товару за кодом (рис. 4.16);
- відправка приватного повідомлення менеджеру (Рис. 4.17).

#### Менеджер:

- реєстрація нового користувача (рис. 4.18);
- перегляд інформації про замовлення при отриманні повідомлення (рис. 4.19);
- скасування замовлення яке обробляється, виконується або доставляється (рис. 4.20, 4.22);
- підтвердження замовлення (рис. 4.21);
- відправка замовлення (рис. 4.23);
- завершення виконання замовлення (рис. 4.24);
- автоматична відправка повідомлення замовнику про зміну стану замовлення (рис. 4.20, 4.21, 4.22, 4.23, 4.24);
- пошук замовлення за станом (рис. 4.22);
- пошук товару за назвою (рис. 4.26);
- пошук товару за кодом (рис. 4.27);
- відправка приватного повідомлення замовнику (рис. 4.28);

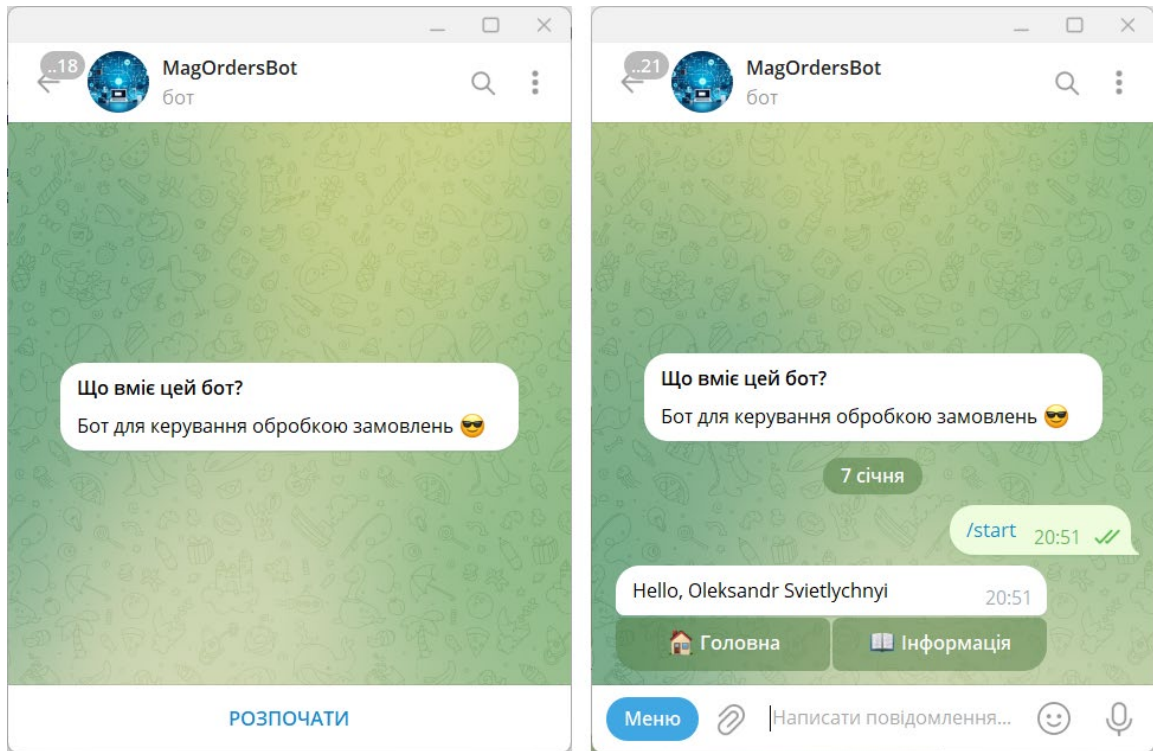


Рисунок 4.1 – Реєстрація нового користувача – замовника



Рисунок 4.2 – Отримання довідкової інформації

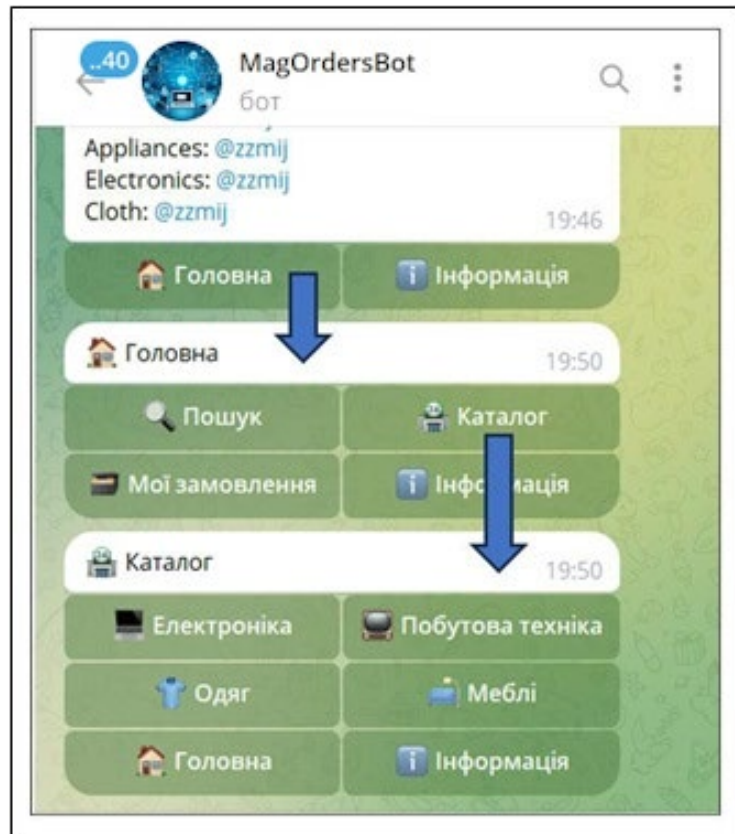


Рисунок 4.3 – Вибір категорії товарів

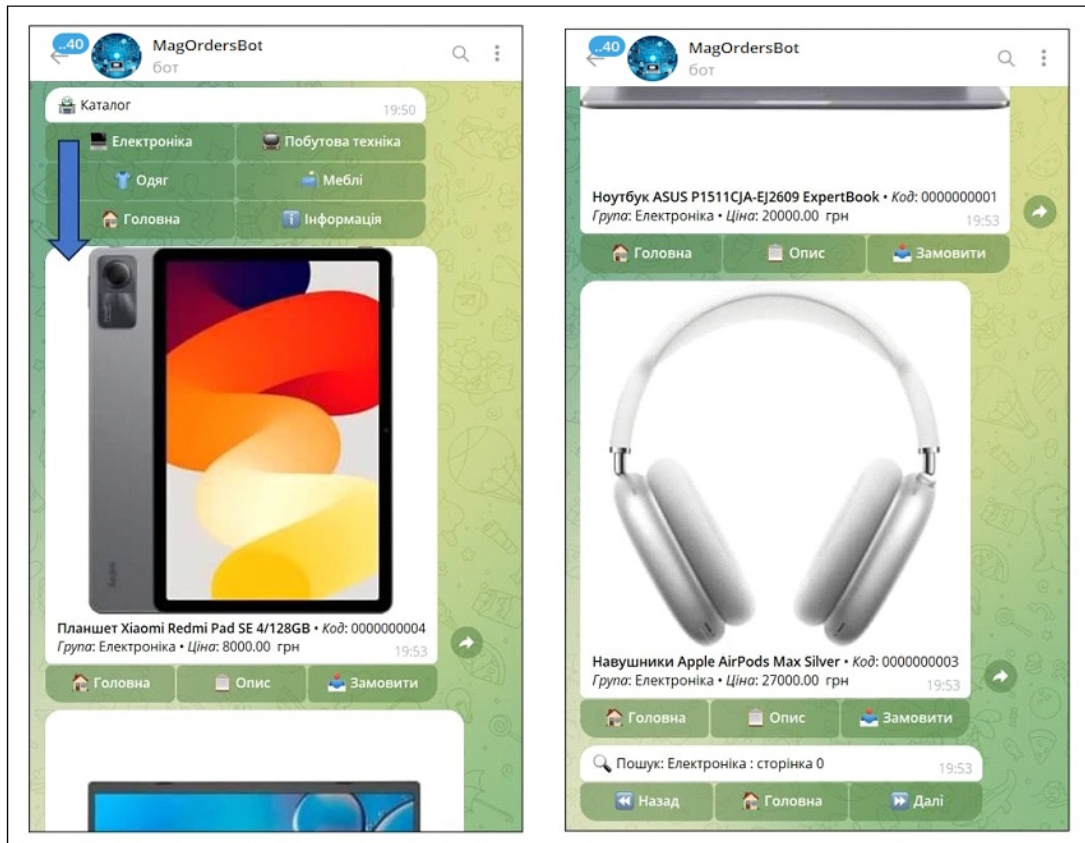


Рисунок 4.4 – Перегляд товарів обраної категорії

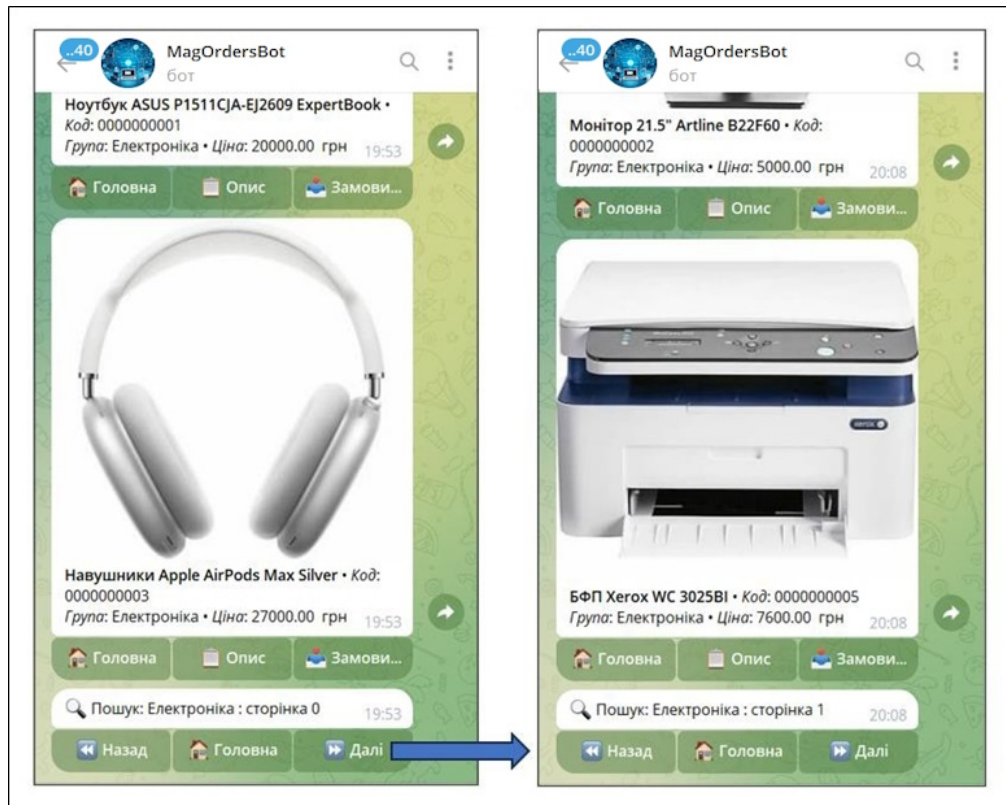


Рисунок 4.5 – Зміна сторінки відображення списку товарів

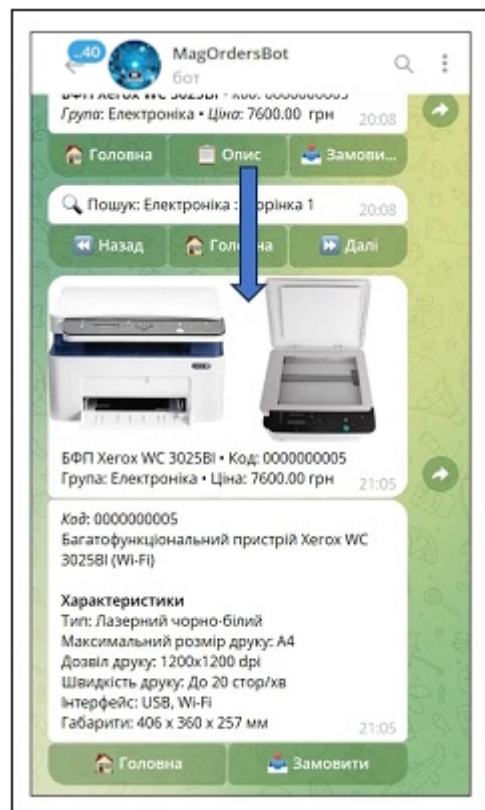


Рисунок 4.6 – Перегляд детальної інформації про товар

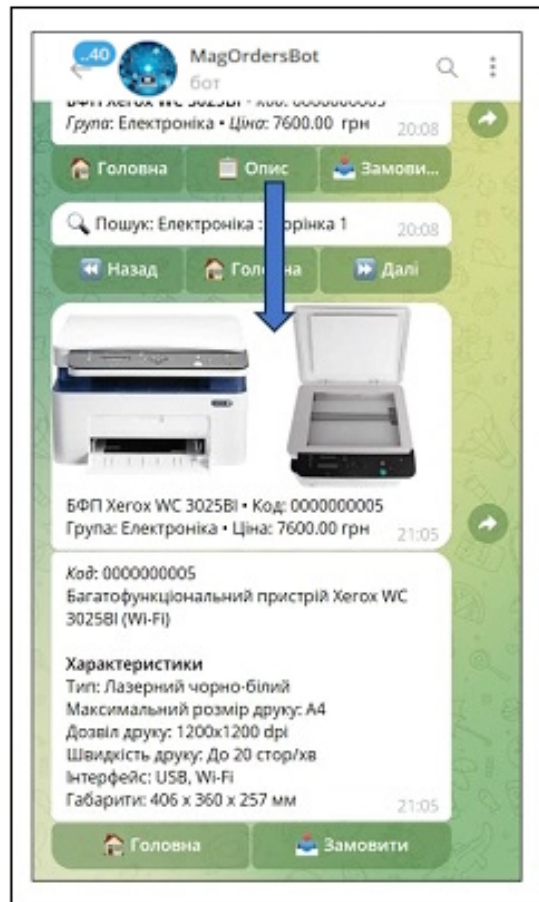


Рисунок 4.7 – Створення замовлення



Рисунок 4.8 – Зміна кількості одиниць товару



Рисунок 4.9 – Оформлення замовлення



Рисунок 4.10 – Зміна адреси доставки



Рисунок 4.11 – Скасування оформлення замовлення

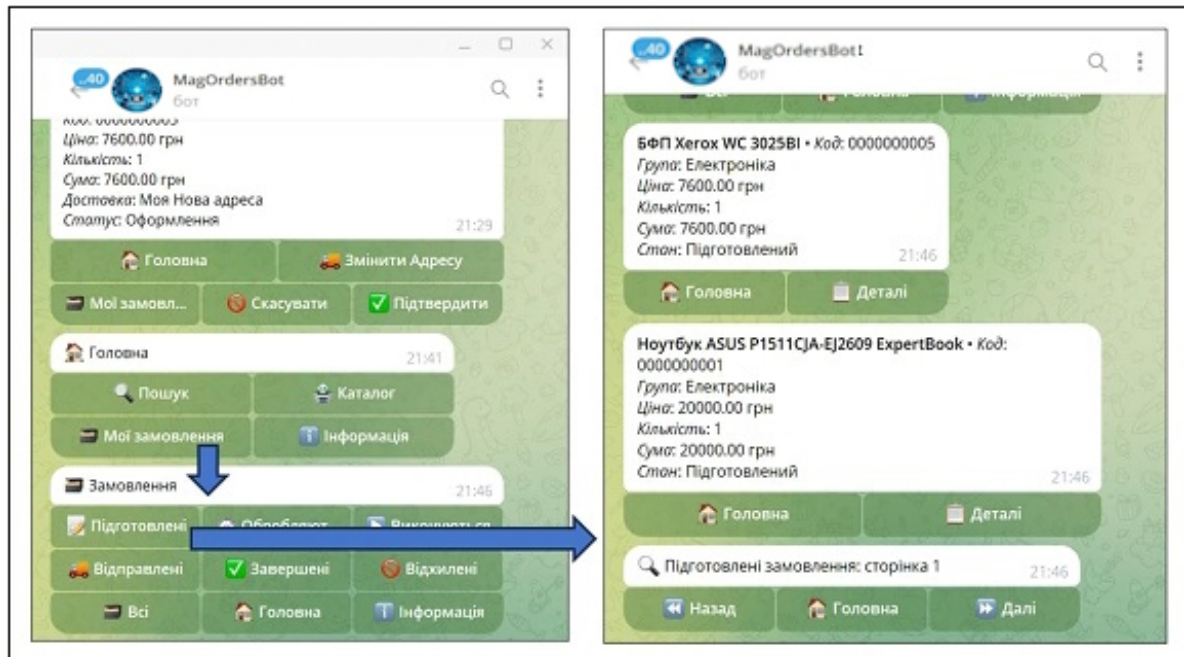


Рисунок 4.12 – Пошук замовлення за станом

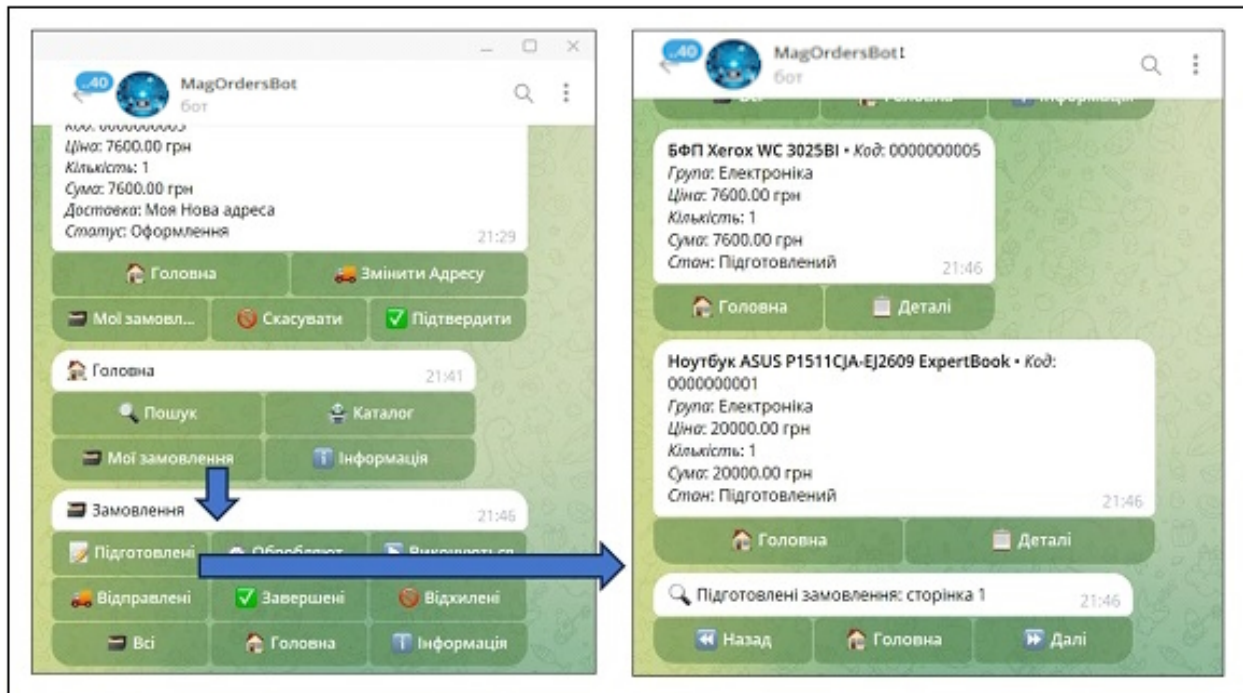


Рисунок 4.13 – Перегляд інформації про замовлення

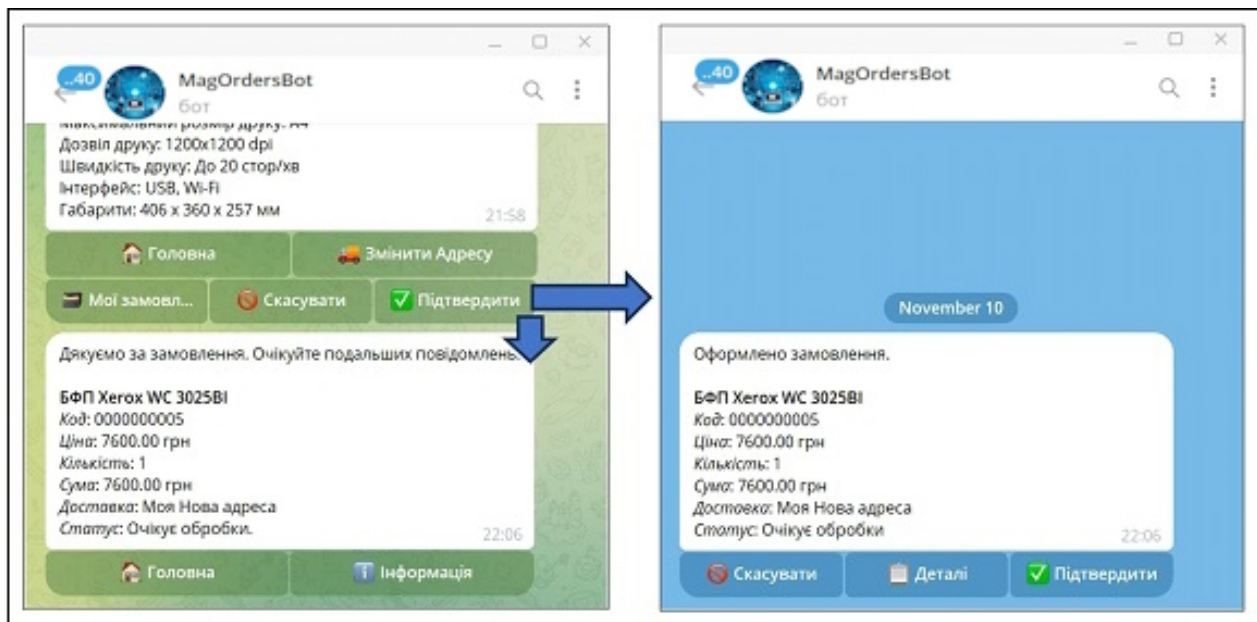


Рисунок 4.14 – Підтвердження створення замовлення та автоматична відправка повідомлення менеджеру



Рисунок 4.15 – Пошук замовником товару за назвою



Рисунок 4.16 – Пошук замовником товару за кодом



Рисунок 4.17 – Відправка приватного повідомлення менеджера



Рисунок 4.18 – Реєстрація нового користувача – менеджера

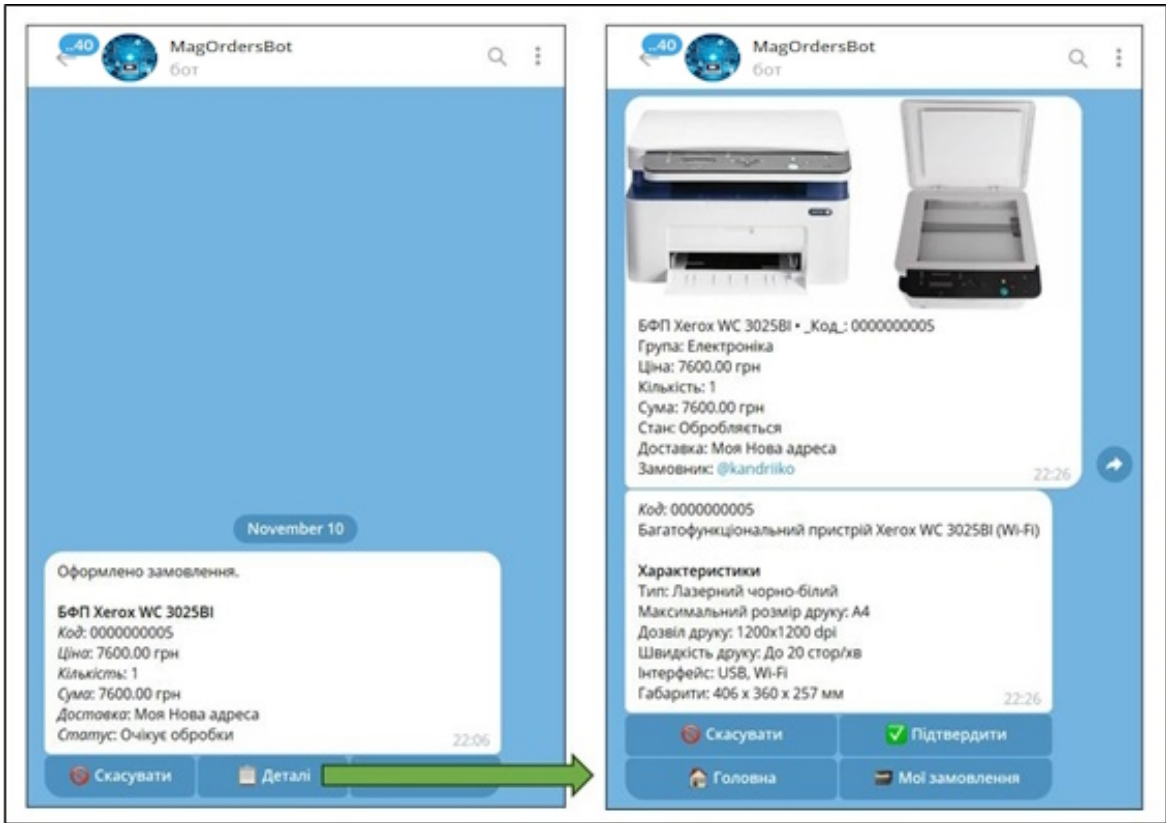


Рисунок 4.19 – Перегляд інформації про замовлення при отриманні повідомлення

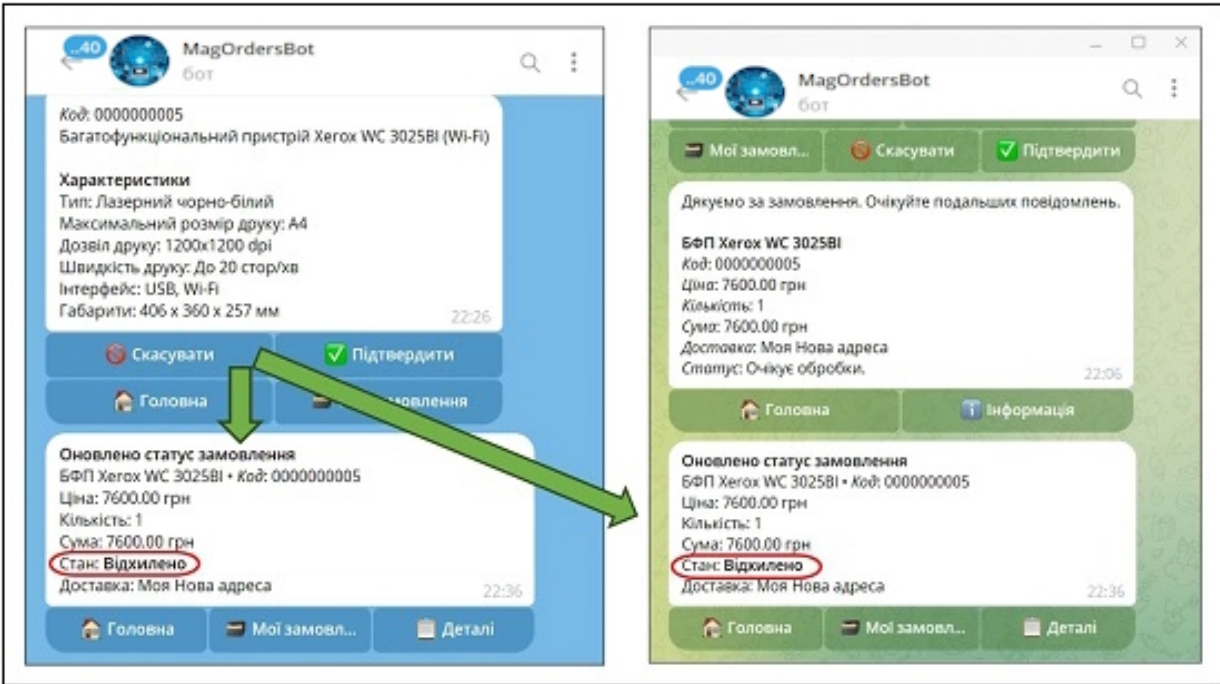


Рисунок 4.20 – Скасування замовлення менеджером та автоматична відправка повідомлення замовнику

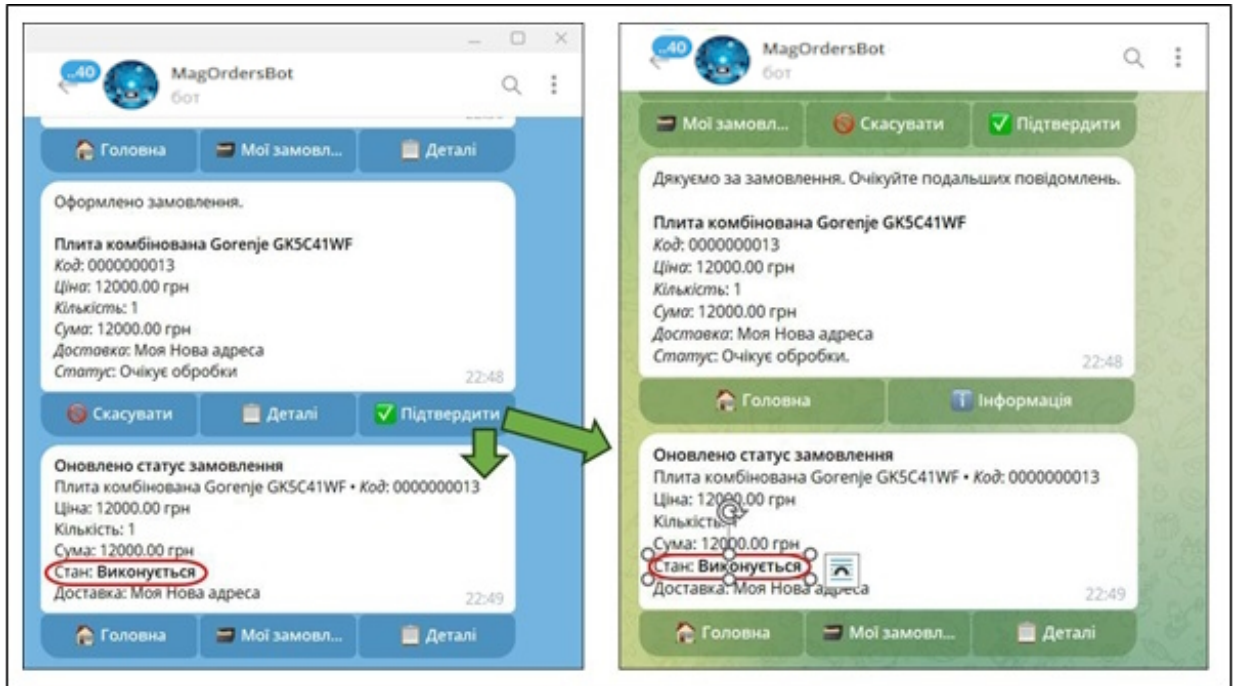


Рисунок 4.21 – Підтвердження замовлення менеджером та автоматична відправка повідомлення замовнику

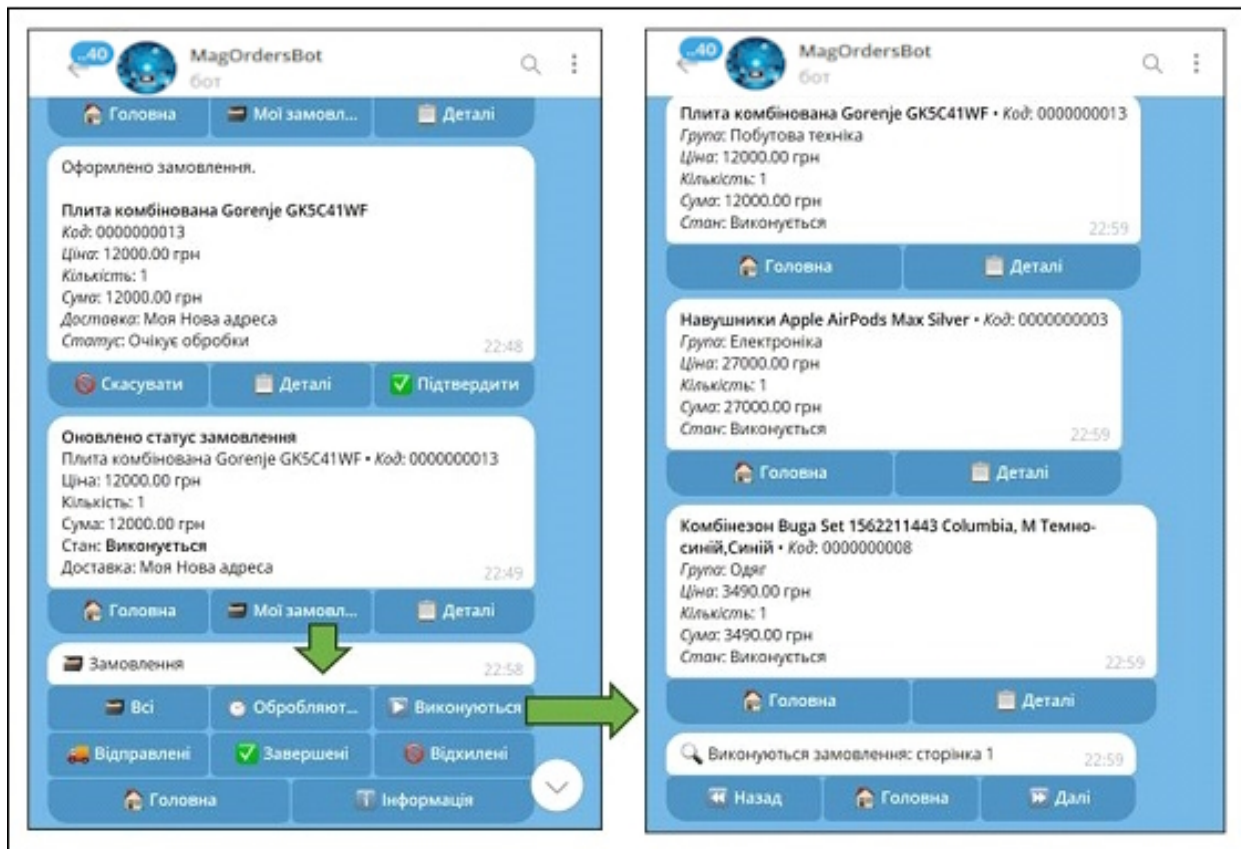


Рисунок 4.22 – Пошук замовлення за станом

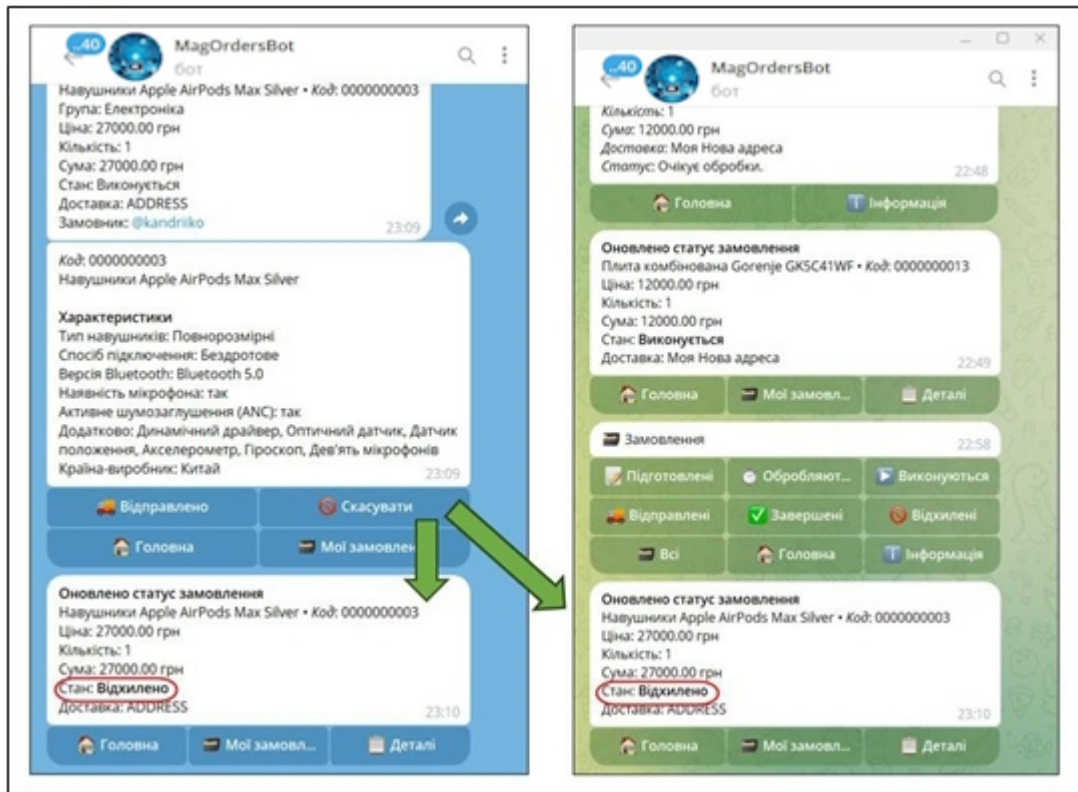


Рисунок 4.23 – Скасування замовлення яке виконувалось менеджером та автоматична відправка повідомлення замовнику

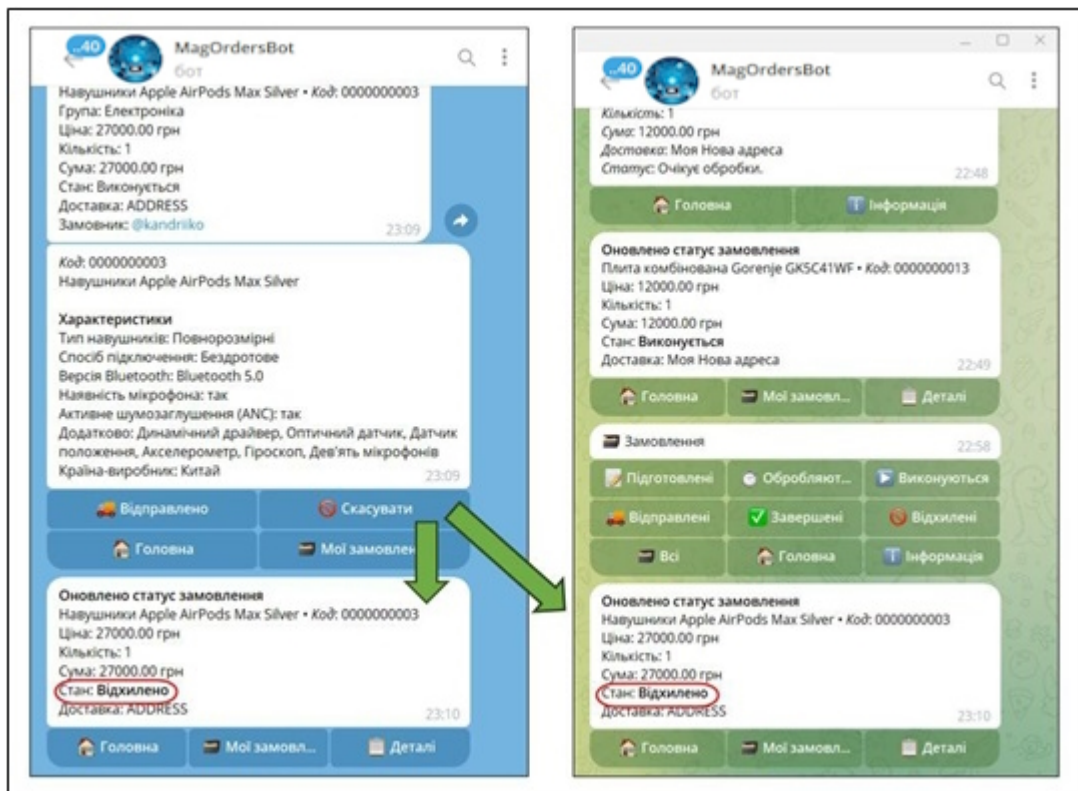


Рисунок 4.24 – Відправка замовлення та автоматична відправка повідомлення замовнику

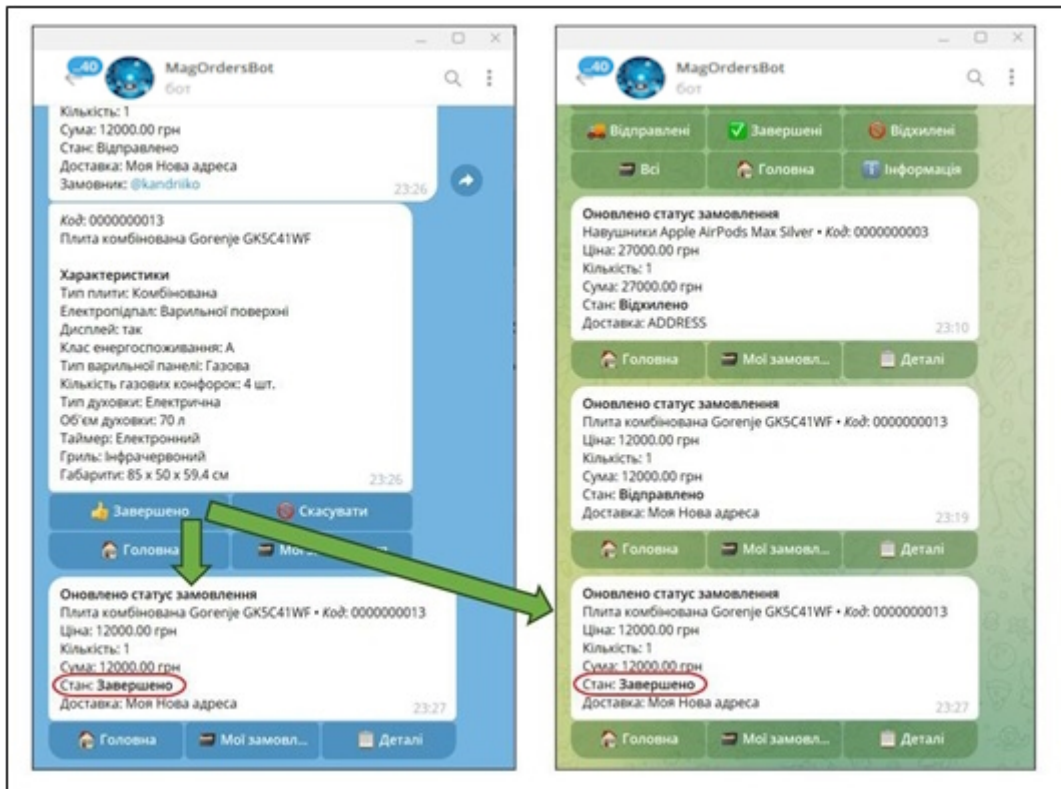


Рисунок 4.25 – Завершення виконання замовлення та автоматична відправка повідомлення замовнику



Рисунок 4.26 – Пошук менеджером товару за назвою



Рисунок 4.27 – Пошук менеджером товару за кодом

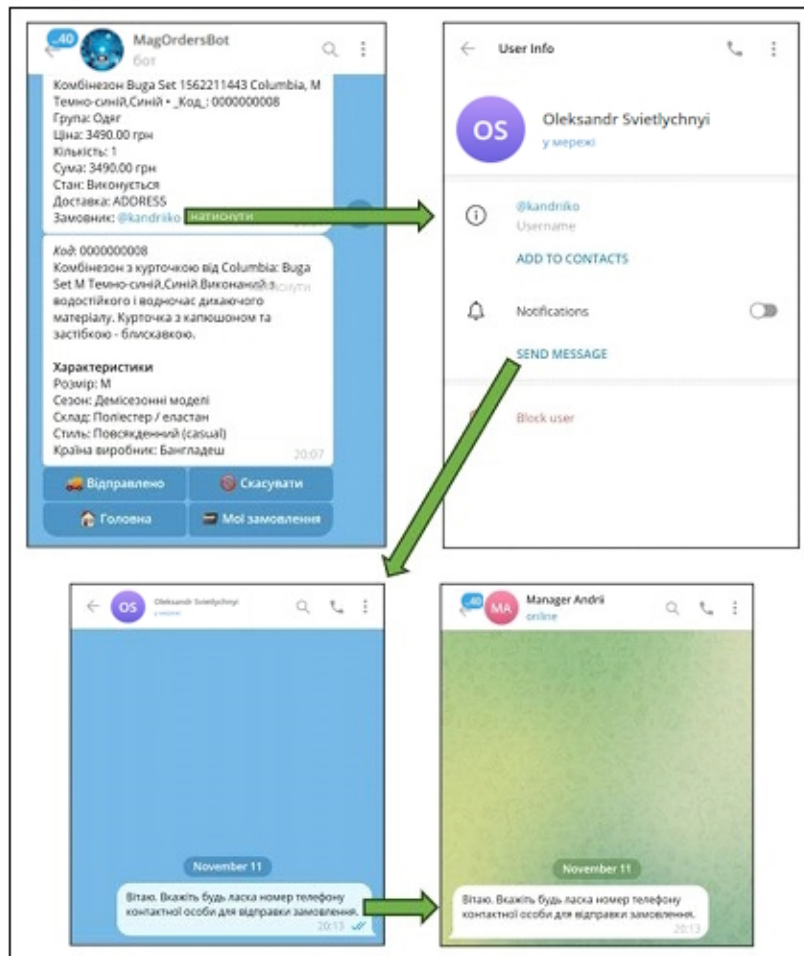


Рисунок 4.28 – Відправка приватного повідомлення замовнику

Під час тестування було виявлено одну особливість: нумерація сторінок при пошуку товарів за назвою або кодом починається з «0». Для зручності користувачів доцільно було б починати нумерацію з «1». Усі реалізовані функції працюють згідно із запланованим.

Загалом, тестування підтвердило, що розроблений Telegram-бот реалізує основний набір функцій, необхідний для роботи простого інтернет-магазину:

- пошук товарів;
- перегляд інформації про товар;
- створення замовлення та перегляд замовлень і їх стану;
- автоматичне відправлення повідомлення менеджеру при створенні замовлення;
- зміна стану замовлення менеджером;
- автоматичне відправлення повідомлення замовнику при зміні стану замовлення;
- відправка приватних повідомлень менеджеру і замовнику.

#### 4.2 Аналіз використання Telegram-бота для управління обробкою замовлень

Метою цієї кваліфікаційної роботи було впровадження технологій чат-ботів для управління обробкою замовлень. У процесі розробки стало очевидно, що для досягнення поставленої мети необхідно реалізувати значно більше функцій, ніж ті, які безпосередньо стосуються обробки замовлень. Для управління замовленнями довелося реалізувати такі можливості:

- доступ до бази даних товарів;
- пошук товарів;
- перегляд інформації про товар;
- створення замовлення.

Хоча це вимагало значного часу, у результаті вдалося створити додаток, який забезпечує повноцінну взаємодію клієнтів і менеджерів із системою.

Проте для використання Telegram-бота в реальних умовах, можливо, знадобиться його вдосконалення. Під час тестування були виявлені функції, які могли б покращити його роботу, але не були реалізовані через обмежені терміни:

- а) реєстрація менеджерів без необхідності додаткових змін в базі даних;
- б) підтвердження зміни адреси доставки;
- в) підтвердження отримання замовлення замовником;
- г) відображення номеру замовлення для замовника і менеджера;
- г) пошук замовлень за номером;
- д) можливість налаштування профіля користувачем;
  - 1) кількість товарів на сторінці при пошуку;
  - 2) вибір типу сортування;
  - 3) використання фільтрів (наприклад: всі товари або тільки ті що є в наявності);
- е) можливість для менеджера змінювати і контролювати наявну кількість товарів;
- є) інтеграція з платіжними системами.

Незважаючи на ці обмеження, Telegram-бот у його поточному вигляді здатен виконувати базові функції для обробки замовлень. Він автоматизує відправлення сповіщень та підтримує інтерфейси для клієнтів і менеджерів. Такий бот можна використовувати для невеликого інтернет-магазину, хоча можуть знадобитися певні доопрацювання відповідно до асортименту товарів.

У процесі розробки та тестування були також ідентифіковані певні обмеження самої платформи Telegram. Зокрема, платформа надає обмежені можливості для налаштування інтерфейсу, що ускладнює відображення всієї необхідної інформації. Крім того, у деяких випадках було б корисно використовувати структуровані запити, наприклад, для заповнення форм клієнтами. На жаль, такої функції в Telegram не передбачено.

З іншого боку, Telegram пропонує багато інших можливостей, які не були задіяні під час розробки. Подальше дослідження цих функцій може допомогти створити більш зручний і функціональний бот.

#### Висновки за розділом 4

У результаті тестування Telegram-бота було підтверджено його функціональність відповідно до запланованих вимог. Для перевірки роботи використовувалися два облікові записи, які моделювали ролі замовника та менеджера. Усі функції бота успішно протестовано на сервері забезпечивши стабільну роботу програми.

Окремі недоліки, такі як початок нумерації сторінок із «0» під час пошуку товарів, незначно впливають на загальну функціональність. Усі основні функції працюють коректно, що робить бот придатним для використання в невеликому інтернет-магазині.

Розробка Telegram-бота підтвердила доцільність використання чат-ботів для автоматизації обробки замовлень. Для забезпечення повноцінної взаємодії клієнтів та менеджерів із системою були реалізовані функції, що виходять за межі базових: доступ до бази даних товарів, пошук, перегляд інформації та створення замовлень.

Попри виявлені обмеження платформи Telegram (обмеження у налаштуванні інтерфейсу та відсутність підтримки структурованих запитів), можливості бота можна значно розширити, досліджуючи додаткові функції Telegram.

Розроблений бот успішно автоматизує сповіщення та забезпечує базовий інтерфейс для взаємодії клієнтів і менеджерів. Цей інструмент може бути ефективно використаний у невеликих інтернет-магазинах із потенціалом подальшого вдосконалення.

## ВИСНОВКИ

Під час написання кваліфікаційної роботи було проаналізовано Telegram як платформу для чат-ботів і визначено, що цей месенджер пропонує унікальні можливості для створення і функціонування чат-ботів. Вони можуть бути використані для різноманітних завдань, таких як підтримка клієнтів, надсилання нагадувань та організація календарних функцій.

Також було охарактеризовано основні інструменти та технології, що застосовуються при розробці Telegram-ботів для управління обробкою замовлень. Розглянуто різноманітні типи баз даних для зберігання даних бота. Реляційні бази даних є оптимальними для зберігання структурованої інформації, такої як таблиці з рядками та стовпцями, в той час як нереляційні бази краще підходять для зберігання неструктурованих даних, таких як текстові повідомлення, зображення і відео.

Результатом розробки телеграм бота, можна застосовувати для автоматизації процесів та підвищення якості обслуговування клієнтів. Вони здатні виконувати різні функції, включаючи приймання замовлень, відповідь на запитання клієнтів і відстеження стану замовлень.

Для реалізації чат-бота, який займається обробкою замовлень, необхідно обрати відповідну базу даних, мову програмування та платформу, враховуючи специфіку проекту та вимоги до його розробки.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. How chatbots influence marketing. URL: [https://www.researchgate.net/publication/333871373\\_How\\_chatbots\\_influence\\_marketing](https://www.researchgate.net/publication/333871373_How_chatbots_influence_marketing) (дата звернення: 10.10.2024).
2. Чат-боти як нове покоління каналів комунікації. URL: <https://core.ac.uk/download/pdf/197267488.pdf> (дата звернення: 10.10.2024).
3. Advantages of Chatbots in E-commerce. URL: <https://ochatbot.com/advantages-of-chatbots-in-e-commerce/> (дата звернення: 24.10.2024).
4. Чат-бот – ефективний інструмент оптимізації роботи контакт-центру. URL: <https://areon.ua/crm-blogs/koryagin/chat-bot-call-center/> (дата звернення: 12.10.2024).
5. Покращення бізнесу за допомогою чат-ботів: що варто знати. URL: <https://webpromoeexperts.net/ua/blog/pokrashchennya-biznesu-za-dopomogoyu-chat-botiv-shcho-var-to-znati/> (дата звернення: 12.10.2024).
6. Тренди чат-ботів 2022. URL: [https://gerabot.com/article/trendi\\_chatbotiv\\_2022](https://gerabot.com/article/trendi_chatbotiv_2022) (дата звернення: 14.10.2024).
7. How Chatbot Change Everything? URL: <https://glovorytech.medium.com/how-chatbot-change-everything-2a557d989373> (дата звернення: 19.10.2024).
8. Як створити чат-бот для Telegram для покращення підтримки клієнтів? URL: <https://helpcrunch.com/blog/uk/chat-bot-dla-telegram/> (дата звернення: 15.10.2024).
9. Chatbots In Customer Service – Statistics and Trends. URL: <https://www.invespcro.com/blog/chatbots-customer-service/> (дата звернення: 23.10.2024).
10. Designing Chatbots: Creating Conversational Experiences. URL: <https://books.google.com.ua/books?hl=en&lr=&id=vdskDwAAQBAJ&oi=fnd&pg=PR11&dq=Designing+Chatbots:+Creating+Conversational+Experiences&ots=7YjwvIFPPR&sig=c-9ALwRtbg->

e7WdRZuXaFr3v9nw&redir\_esc=y#v=onepage&q=Designing%20Chatbots%3A%20Creating%20Conversational%20Experiences&f=false( дата звернення: 19.10.2024).

11. Telegram Bots Book URL: <https://telegrambots.github.io/book/> (дата звернення: 25.10.2024).

12. Як створити Телеграм-Бота правильно. URL: <https://www.servicedesk.site/uk/2019/08/09/як-створити-телеграм-бота-правильно/> (дата звернення: 21.10.2024).

13. Будуємо телеграм чат-бот на Java: від ідеї до деплою. URL: <https://dou.ua/forums/topic/38358/> (дата звернення: 24.10.2024).

14. Types of Databases: Choose the Right Database for Your Needs. URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення: 17.10.2024).

15. Як створити автоматизований ланцюжок для Telegram чат-бота. URL: <https://sendpulse.ua/knowledge-base/chatbot/telegram/create-flow> (дата звернення: 21.10.2024).

16. 19 Best UX Practices for Building Chatbots. URL: <https://chatbotsmagazine.com/19-best-practices-for-building-chatbots-3c46274501b2> (дата звернення: 26.10.2024)

17. Pros and Cons of Using SQL vs NoSQL Databases. URL: <https://www.codingninjas.com/studio/library/pros-and-cons-of-using-sql-vs-nosql-databases> (дата звернення: 25.10.2024).

18. Introduction to NoSQL. URL: <https://www.mongodb.com/nosql-explained> (дата звернення: 31.10.2024).

19. MongoDB. URL: <https://www.mongodb.com/> (дата звернення: 29.10.2024).

20. Choosing the Right Database for Your Application. URL: <https://www.digitalocean.com/community/tutorials/a-comparison-of-nosql-database-management-systems-and-models> (дата звернення: 05.10.2024).

21. Introduction to Python. URL: <https://docs.python.org/3/tutorial/index.html> (дата звернення: 26.10.2024).

22. Python Telegram Bot Library. URL: <https://python-telegram-bot.readthedocs.io/en/stable/> (дата звернення: 21.10.2024).
23. Java Documentation. URL: <https://docs.oracle.com/en/java/> (дата звернення: 22.10.2024).
24. Java Programming Tutorial. URL: <https://docs.oracle.com/javase/tutorial/> (дата звернення: 19.10.2024).
25. Java Telegram bot tutorial. URL: <https://monsterdeveloper.gitbook.io/java-telegram-bot-tutorial/> (дата звернення: 18.10.2024).
26. PHP Manual. URL: <https://www.php.net/manual/en/> (дата звернення: 20.10.2024).
27. PHP Programming Tutorial. URL: <https://www.w3schools.com/php/> (дата звернення: 07.10.2024).
28. Node.js Documentation. URL: <https://nodejs.org/en/docs/> (дата звернення: 20.10.2024).
29. Node.js Telegram Bot. API URL: <https://github.com/yagop/node-telegram-bot-api> (дата звернення: 26.10.2024).
30. How to create Telegram Chatbot with Node.js? URL: <https://www.geeksforgeeks.org/how-to-create-telegram-chatbot-with-node-js/> (дата звернення: 21.10.2024).
31. C++ Telegram Bot API Wrapper. URL: <https://github.com/reo7sp/tgbot-cpp> (дата звернення: 20.10.2024).
32. C++ Documentation. URL: <https://en.cppreference.com/w/cpp> (дата звернення: 24.10.2024).
33. Build a Simple Chatbot with Tensorflow, Python and MongoDB. URL: <https://kheartig.wordpress.com/2017/12/30/build-a-simple-chatbot-with-tensorflow-python-and-mongodb/> (дата звернення: 21.10.2024).
- 34 MongoDB Cloud Services. [Електронний ресурс] - Режим доступу: <https://www.mongodb.com/products/platform/cloud> (дата звернення: 30.12.2024).
- 35 Shutterfly [Електронний ресурс] - Режим доступу: <https://www.shutterfly.com/> (дата звернення: 30.12.24).

36 IntelliJ IDEA – the Leading Java and Kotlin IDE. [Электронный ресурс] - Режим доступа: <https://www.jetbrains.com/idea/> (дата звернения: 20.12.2024).

37 Java SE 11 Archive Downloads [Электронный ресурс] - Режим доступа: <https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html> (дата звернения: 21.12.2024).

38 Spring Boot. [Электронный ресурс] - Режим доступа: <https://spring.io/projects/spring-boot> (дата звернения: 24.12.2024).

39. Project Lombok. [Электронный ресурс] - Режим доступа: <https://projectlombok.org> (дата звернения: 24.12.2024).

40. Apache Maven Project [Электронный ресурс] - Режим доступа: <https://maven.apache.org> (дата звернения: 20.12.2024).

41. Git [Электронный ресурс] - Режим доступа: <https://git-scm.com> (дата звернения: 25.12.2024).

42 Docker [Электронный ресурс] - Режим доступа: <https://www.docker.com> (дата звернения: 25.12.2024).

43. Telegram Bot API Overview [Электронный ресурс] - Режим доступа: <https://core.telegram.org/bots> (дата звернения: 15.12.2024).

44. Amazon Web Services [Электронный ресурс] - Режим доступа: <https://aws.amazon.com> (дата звернения: 24.12.2024).

45. Microsoft Azure [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/en-us> (дата звернения: 23.12.2024).

46. Google Cloud [Электронный ресурс] - Режим доступа: <https://cloud.google.com> (дата звернения: 15.12.2024).

47. Fly.io [Электронный ресурс] - Режим доступа: <https://fly.io/> (дата звернения: 30.12.2024).