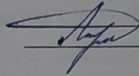


Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматики
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

Бакалаврська дипломна робота на тему:

РОЗРОБКА ЧАТ-БОТУ ДЛЯ ПРОДАЖУ ТОВАРІВ

Виконав: студент 4 курсу, групи КН-22мс
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Лебєдев Д.І.

(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри КН

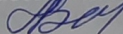


Озеранський В.С.

(прізвище та ініціали)

« » 2024 р.

Рецензент: к.т.н., доцент кафедри АІТ



Барабан М.В.

(прізвище та ініціали)

«07» 06 2024 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)



«07» 06 2024 р.

Зінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Навчальний модуль – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А.А.

“ 07 ” 06 2024 року

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Лебедєву Дмитру Івановичу

Тема роботи: Розробка чат-боту для продажу товарів.
Рівень роботи: Озеранський Володимир Сергійович, к.т.н., доцент
Підтверджені наказом вищого навчального закладу “ 11 ” 05 2024 року № 80

Строк подання студентом роботи 27 05 2024 р.

Вихідні дані до роботи :

Кількість видів товарів – не менше 10шт;
Кількість користувачів для одночасного сеансу – не менше 100чол;
Мова програмування – об'єктно-орієнтована;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

обґрунтування доцільності розробки чат-боту для продажу товарів;
аналіз переваг та недоліків існуючих чат-ботів для продажу товарів;
створення структури чат-боту для продажу товарів;
розробка алгоритму роботи чат-боту для продажу товарів;
визначити предмет, об'єкт, мету та задачі подальшого дослідження.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

Схема алгоритму роботи чат-боту для продажу товарів;
Діаграми компонентів чат-боту для продажу товарів;
Загальний вигляд інтерфейсу користувача чат-боту для продажу товарів.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання визначено	завдання прийнято
1-3	Озеранський В.С., к.т.н., доцент		

6. Дата видачі завдання 07.03.2024

КАЛЕНДАРНИЙ ПЛАН

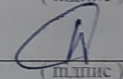
№ з / п	Назва та зміст етапу	Термін виконання		Прим.
		початок	закінчення	
1	Обґрунтування доцільності розробки чат-боту для продажу товарів	06.05.24	07.05.24	
2	Розробка чат-боту для продажу товарів	08.05.24	09.05.24	
3	Програмна реалізація чат-боту для продажу товарів	13.05.24	14.05.24	
4	Аналіз функціонування чат-боту для продажу товарів	19.05.24	19.05.24	
5	Оформлення додатків	23.05.24	26.05.24	
6	Розробка інструкції користувача	26.05.24	27.05.24	
7	Оформлення матеріалів до захисту БДР	04.06.24	06.06.24	

Студент


(підпис)

Лебедєв Д.
(прізвище та ініціали)

Керівник проекту (роботи)


(підпис)

Озеранський
(прізвище та ініціали)

АНОТАЦІЯ

УДК 621.374.415

Лебедєв Д.І. Розробка чат-боту для продажу товарів. Бакалаврська дипломна робота складається з 76 сторінок формату А4, на яких є 38 рисунків, список використаних джерел містить 19 найменувань.

У даній бакалаврській дипломній роботі досліджено процес розробки чат-бота для продажу товарів. В роботі проаналізовано сучасні аналоги, що використовуються для продажу товарів з використанням телеграм-ботів. Наведено переваги і недоліки існуючих аналогів. Досліджено способи вирішення даної задачі. Запропоновано використовувати підхід, що ґрунтується на застосуваннях UML-діаграм та середовища розробки Microsoft Visual Code. Результатом дипломного дослідження є система у вигляді телеграм-боту, що реалізовано на мові програмування Python.

Ключові слова: месенджер, телеграм-бот, продаж товарів, Python.

ABSTRACT

Lebedev D.I. Development of a chatbot for selling goods. The bachelor thesis consists of 76 pages of A4 format, on which there are 38 figures, the list of used sources contains 19 names.

In this bachelor's thesis, the process of developing a telegram bot for selling goods is investigated. The paper analyzes modern analogs used to sell goods using Telegram bots. The advantages and disadvantages of existing analogs are presented. The ways of solving this problem are investigated. It is proposed to use an approach based on the use of UML diagrams and the Microsoft Visual Code development environment. The result of the thesis research is a system in the form of a telegram bot implemented in the Python programming language.

Keywords: messenger, telegram bot, sales of goods, Python.

ЗМІСТ

ВСТУП.....	7
1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ЧАТ-БОТА ДЛЯ ПРОДАЖУ ТОВАРІВ	9
1.1 Обґрунтування актуальності розробки чат-бота для продажу товарів	9
1.2 Аналіз існуючих чат-ботів для продажу товарів	12
1.3 Постановка задачі.....	16
1.4 Висновок	18
2 РОЗРОБКА ТА АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ЧАТ-БОТА ДЛЯ ПРОДАЖУ ТОВАРІВ	19
2.1 Варіантний аналіз шляхів розв’язання поставленої задачі	19
2.2 Розробка UML-діаграм варіантів використання чат-бота для користувача та адміністратора	22
2.3 Розробка загального алгоритму роботи чат-бота для продажу товарів	23
2.4 Розробка алгоритму функціонування головного меню чат-бота для продажу товарів.....	25
2.5 Алгоритм запуску чат-бота для продажу товарів.....	27
2.6 Висновок	28
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТА ДЛЯ ПРОДАЖУ ТОВАРІВ	30
3.1 Обґрунтування вибору мови програмування.....	30
3.2 Вибір стеку технологій для реалізації чат-бота	31
3.3 Обґрунтування вибору платформи для розробки чат-бота.....	32
3.4 Аналіз процесу створення оболонки чат-бота з використанням «BotFather».....	34
3.5 Створення інтерфейсу чат-бота для продажу товарів	38
3.6 Аналіз та обґрунтування вибору бази даних для розроблюваного чат-боту .	42
3.7 Розробка бази даних для чат-бота для продажу товарів	44
3.8 Тестування роботи чат-бота для продажу товарів	49
3.9 Висновок	60
ВИСНОВКИ	61

ДОДАТОК А. Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	64
ДОДАТОК Б. Інструкція користувача	65
ДОДАТОК В. Лістинг програми	67
ДОДАТОК Г. Ілюстративна частина	71

ВСТУП

Актуальність теми. У сучасному світі електронна комерція та онлайн-торгівля стають все більш популярними та перспективними напрямками бізнесу. Зростаюча залежність від мобільних пристроїв та поширення месенджерів [2] створюють нові можливості для ефективної комунікації та здійснення операцій з покупцями. У цьому контексті розробка чат-бота для продажу товарів має великий потенціал.

На сьогоднішній день багато підприємств та підприємців вже використовують соціальні медіа та веб-платформи для реалізації своїх товарів та послуг. Однак, використання телеграм-ботів для цих цілей є новаторським підходом, який надає численні переваги, такі як простоту та зручність взаємодії з користувачами, автоматизацію процесу продажу, розширені можливості маркетингу та інші [3].

Оскільки телеграм-боти для продажу товарів є досить новим явищем, у роботі буде проведений аналіз сучасного стану розвитку цієї технології. Будуть досліджені вже існуючі рішення та принципи їхньої роботи, а також визначені їхні переваги та недоліки. Це допоможе визначити найбільш ефективні та перспективні підходи для розробки необхідного чат-бота для продажу товарів.

На сьогоднішній день створити телеграм-бота можливо кількома способами, такими як:

- використання Telegram Bot API [4]. Telegram надає інтерфейс програмування додатків (API) для створення та управління телеграм-ботами. Цей метод надає різні методи для керування повідомленнями, клавіатурами, зображеннями та іншими функціями;

- використання платформ для створення ботів (конструкторів). Усе більше платформ надають можливості для створення телеграм-ботів без програмування. Наприклад, Chatfuel, ManyChat і Flow XO – це сервіси, що дозволяють

створювати ботів за допомогою візуального інтерфейсу [5]. Вони надають шаблони повідомлень, правила для обробки вхідних запитів та інші функції;

- ручне програмування. Використовуючи середовище розробки та мову програмування, самотійно написати телеграм-бота в ручну. Цей метод дозволяє максимально точно виконати поставлене завдання.

Незважаючи на всю різноманітність для створення телеграм-ботів, лише ручне програмування може надати найширший спектр можливостей для розвитку такої роботи. Індивідуально розроблений інтерфейс з необхідним функціоналом дозволяють створити такого телеграм-бота, який повністю відповідає її цілям і завданням, пропонуючи можливість подальшого розвитку.

Таким чином, розробка чат-бота для продажу товарів є доцільною, оскільки вона дозволяє підприємствам ефективно використовувати популярний месенджер Телеграм для залучення та обслуговування клієнтів, автоматизації процесу продажу та покращення маркетингових стратегій.

Метою дослідження є розширення функціональних можливостей програмного забезпечення для продажу товарів.

Для досягнення поставленої мети потрібно виконати наступні завдання

1. Здійснити аналіз предметної області створення чат-ботів;
2. Виконати огляд існуючих чат-ботів для продажу товарів та здійснити аналіз їх переваг та недоліків;
3. Розробити загальний алгоритм роботи чат-бота для продажу товарів;
4. Виконати програмну реалізацію чат-бота для продажу товарів;
5. Здійснити тестування розробленого програмного забезпечення.

Об'єкт дослідження – процес продажу товарів за допомогою чат-ботів.

Предмет дослідження – програмні засоби для продажу товарів за допомогою чат-ботів

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ЧАТ-БОТА ДЛЯ ПРОДАЖУ ТОВАРІВ

1.1 Обґрунтування актуальності розробки чат-бота для продажу товарів

Чат-боти є програмними агентами, що взаємодіють з користувачами за допомогою текстових повідомлень [6]. Вони здатні автоматично обробляти запити, розуміти мову користувача та надавати відповіді чи виконувати дії на основі заданої функціональності. Чат-боти стали популярними завдяки зростанню інтересу до штучного інтелекту та розвитку природно-мовних інтерфейсів.

Згодом, з появою соціальних медіа та месенджерів, чат-боти стали використовуватися для автоматизації розмов з користувачами, надання послуг та вирішення різноманітних завдань. Завдяки технологічному розвитку та доступності штучного інтелекту, сучасні чат-боти стали більш розумними та ефективними.

Зручність та доступність для користувачів. Телеграм-боти забезпечують простий та зручний спосіб взаємодії між бізнесом і користувачами. Користувачі можуть швидко переглядати товари, отримувати інформацію про ціни та замовляти продукти безпосередньо в месенджері.

Використання телеграм-бота дозволяє бізнесу залучати клієнтів з різних країн та місцезнаходжень, оскільки Telegram є доступним практично для будьякого користувача з підключенням до Інтернету.

Далі детальніше розглянемо області використання чат-ботів [7].

Можливими нішами для чат-ботів є сервіси інформування учасників певних груп і колективів, рутині інформаційні процеси виборчих кампаній та закладів вищої освіти, особливо на початку навчального року, служби бронювання і купівлі квитків, служби підтримки з доставки товарів, їжі, квітів тощо. Сфери і способи застосування ботів найрізноманітніші і залежать від креативності розробників. Чат-боти є незамінними помічниками у будь-якій сфері, де наявні великі обсяги комунікації з клієнтами. До того ж, при розробці чат-ботів можна

інтегрувати платіжні системи для онлайн-оплати при прийомі замовлень, що дозволяє суттєво покращити, спростити та пришвидшити взаємодію з клієнтами, позбавляє від потреби телефонувати їм. Застосування чат-ботів суттєво економить час, позаяк клієнт, який звернувся за підтримкою, отримує відповідь миттєво в режимі 24/7, без очікування на підключення оператора. При цьому він не стикається зі спамом, зайвими балачками, нав'язливими закликами, а отримує лише корисну інформацію [6].

Нині однією з основних сфер застосування чат-ботів є розсилка рекламних оголошень, акційних пропозицій тощо. Тобто чат-боти використовують як потужний інструмент маркетингу. Дослідження свідчить, що 90% повідомлень у месенджерах відкривають протягом 90 секунд після отримання, а у п'ять разів перевищує показники ефективності e-mail та SMS. При цьому витрати на розсилку набагато нижчі. Тому компанії зацікавлені в розширенні бази контактів і збільшенні обсягів продажів засобами відповідних чат-ботів. Крім того, застосування чат-бота дозволяє суттєво знизити навантаження і витрати на службу підтримки, позаяк автоматизує обробку простих, часто повторюваних запитів від клієнтів. При цьому відбувається паралельне опрацювання необмеженої кількості заявок. Це дозволяє розвантажити команду і залучати менеджерів тільки за потреби вирішення складних завдань, тим самим оптимізувати витрати на персонал [6].

У сфері статистики боти корисні тим, що можуть автоматично відслідковувати ціни на акції, перегляди сторінок веб-сайту компанії або кількість контактів, які було створено за попередній день, формувати статистичні дані у зручний користувачу формат тощо [6].

У медицині чат-боти можуть використовуватись, насамперед, для швидкої дистанційної попередньої діагностики. Крім того, засоби штучного інтелекту можуть використовуватись для аналізу даних здоров'я як окремих пацієнтів, так і прогнозування тенденції поширення вірусних захворювань на період сезонних коливань та можливих епідемій. Майбутнє в області охорони здоров'я залежить

від здатності впровадження таких технологій, адже збір даних при опитуванні пацієнтів відбувається віддалено, що знижує навантаження на лікарів та зменшує ймовірність поширення інфекційних захворювань. Крім того, успішно зарекомендували себе чат-боти в загальній та клінічній психології як додаткові засоби медичної допомоги. Такі чат-бот для психічного здоров'я дозволяють користувачам говорити про все, що вони хочуть, і є достатньо розумними, щоб задавати змістовні питання та відповідати на них. Їх метою є психологічна допомога людям, які борються з депресією, особливо якщо їм не вистачає уваги і нема з ким поговорити про свою проблему [6].

У навчальному процесі існують різні види чат-ботів. Одні з них здатні миттєво відповісти на численні питання абітурієнтів під час вступу до закладу вищої освіти (запитання про вступну кампанію, спеціальності та їх різницю, гуртожитки, історію факультету тощо). Інші мають за мету полегшити та пришвидшити студентам будь-якого курсу пошук інформації під час навчання (про розклад занять, навчально-методичні матеріали, розміщення аудиторій, навчальний процес тощо) та допоможуть швидко вирішити будь-які організаційні питання, адже швидке отримання відповідей на запитання – одна з найбільш бажаних функцій у процесі навчання студента. Дослідження свідчать про високу довіру студентів до цієї технології та задоволеність її використанням. Успішним є використання чат-бота у галузі професійної орієнтації для пошуку майбутньої роботи, здатного обробляти будь-які запити природною мовою та підвищуючи мотивацію учнів до отримання певної професії у віртуальному інформаційному середовищі [6].

У галузі рекрутингу чат-бот вважається ефективною комунікаційною системою, яка успішно спрощує роботу менеджерів по персоналу та рекрутерів, автоматично збираючи та систематизуючи відповідні компетенції, навички та досвід під час автоматичного онлайн опитування кандидатів. Адже людські ресурси (HR) будь-якої організації відіграють важливу роль у будівництві та зростанні компанії, визначаючи ефективних та компетентних працівників. Технологія, керована ШІ, перетворює складні проблеми на спрощені рішення.

Такого роду чат-боти вдало визначають найкращих кандидатів, збільшують кількість заявок, адже працюють у режимі 24/7, автоматично планують проведення співбесід і відповідають на запитання кандидатів на роботу. З винаходом чат-ботів, які є важливою сферою в галузі ІІІ та обробки природних мов, організації стали ширше їх залучати для виконання робіт, орієнтованих на спілкування без будь-якого людського втручання. Чат-боти налагоджують стосунки, відповідають на запитання і сприяють кращому порозумінню серед працівників організації з урахуванням відповідних проблем, впливають на процес прийняття рішень в організації [6].

Отже чат-боти можуть мати різне призначення і мати різний функціонал, і саме від нього залежить вибір платформи, в якій він буде працювати, і засобів розробки. Після визначення вимог до створюваного чат-бота, залежно предметної області, треба розробити комунікаційну архітектуру, яка дозволить вибудувати ієрархію контенту для спілкування. Для цього варто провести опитування серед групи людей цільової аудиторії, структурувати контент і змодельовати схему діалогу бота, зібрати інформацію, необхідну для можливого подальшого навчання бота з метою повноцінного ведення ним діалогу і розуміння ним різних варіантів одного й того самого поняття для формування коректних відповідей [6].

1.2 Аналіз існуючих чат-ботів для продажу товарів

У цьому розділі будуть розглянуті три різні чат-боти, які можна використати для продажу товарів. Для того, щоб почати розробку програмного продукту, необхідно було ознайомитись з ринком онлайн помічників та асистентів для продажу товарів. Кожен помічник виконує певну задачу і працює в своїй предметній області. Оскільки майже всі помічники по своїй структурі є досить простими, наявність інструкції не є необхідністю [3]. Зараз будуть наведені аналоги технологій онлайн помічника для продажу товарів у мережі інтернет.

Чат-боти. Дані боти можуть бути використаними у двох випадках: у

месенджерах та безпосередньо на сайті. При цьому данні боти можуть бути як на базі штучного інтелекту, так і готовими відповідями на повідомлення. Даний вид є найпопулярнішим серед онлайн помічників.

Приклади:

1. Чат-бот поштової служби «Justin». Бот допомагає знайти інформацію про відділення, дізнатись про поштові послуги, розрахувати вартість доставки конкретної посилки та відслідкувати її.

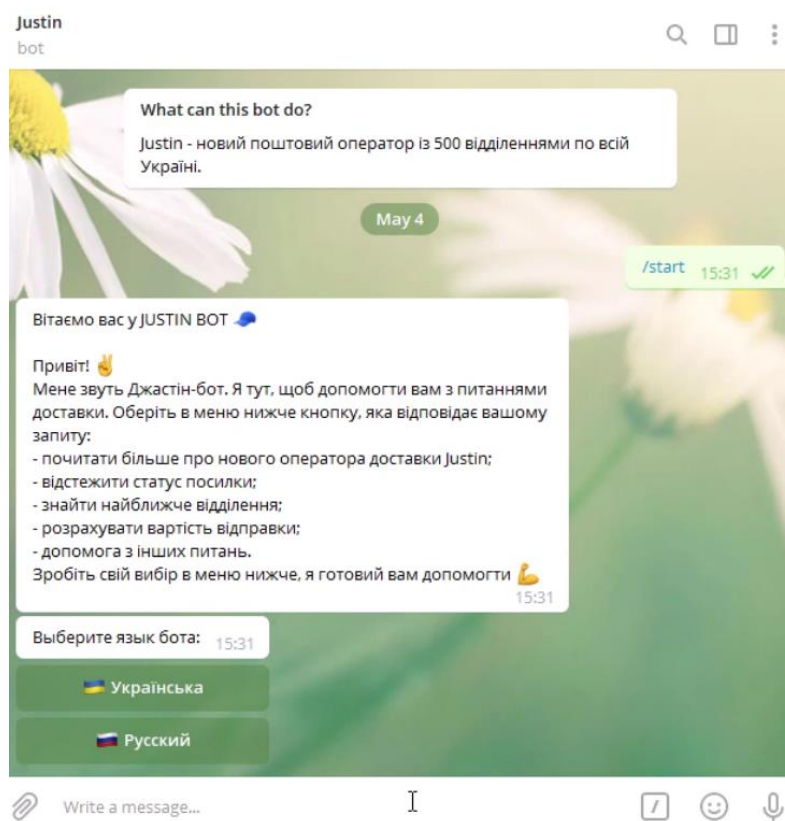


Рисунок 1.1 – Чат-бот служби «Justin»

Голосові помічники. Даний вид онлайн помічників, а точніше асистентів, є досить цікавим. Він реалізований наступним чином: Або напряду на сайті магазину, або окремим додатком, з яким можливо спілкуватись голосом [6]. Така система є складною завдяки тому, що використовується система розпізнавання голосу, система штучного інтелекту.

Приклади:

1. Магазин електроніки «Electronic Store». На даному сайті представлений голосовий асистент, який допомагає у здійсненні покупок. Він може знайти

певний розділ, чи товар. Також відкривши картку товару, даний бот може розповісти його характеристики.

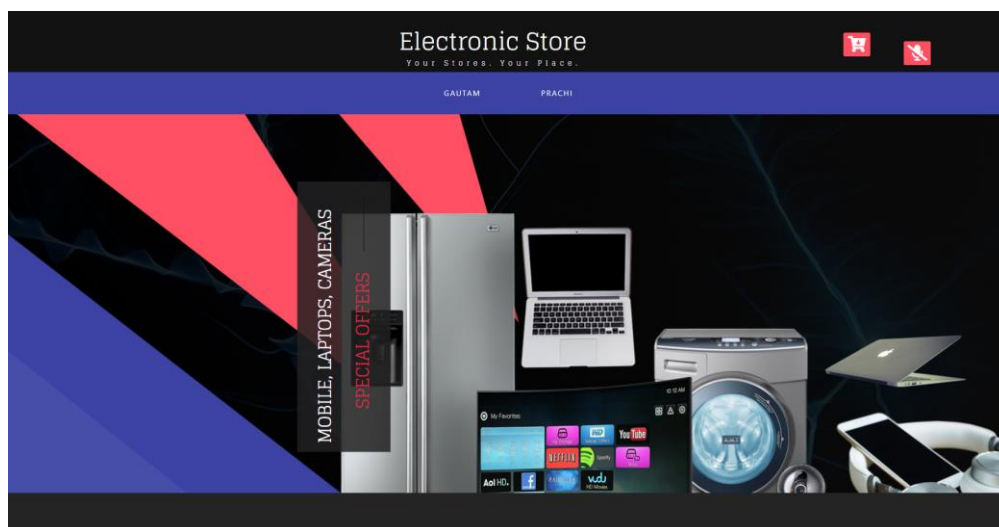


Рисунок 1.2 – Інтернет магазин «Electronic Store»

2. Бот онлайн-асистент «Leon». Це онлайн помічник який спілкується за допомогою голосу та тексту. Він виконує багато функцій, серед яких спілкування, консультація, допомога. Він зможе сказати вам прогноз погоди, новини у вашому місті, чи допомогти у покупках. Даний асистент знайде потрібний вам товар на різних сайтах, чим спростить вам роботу.

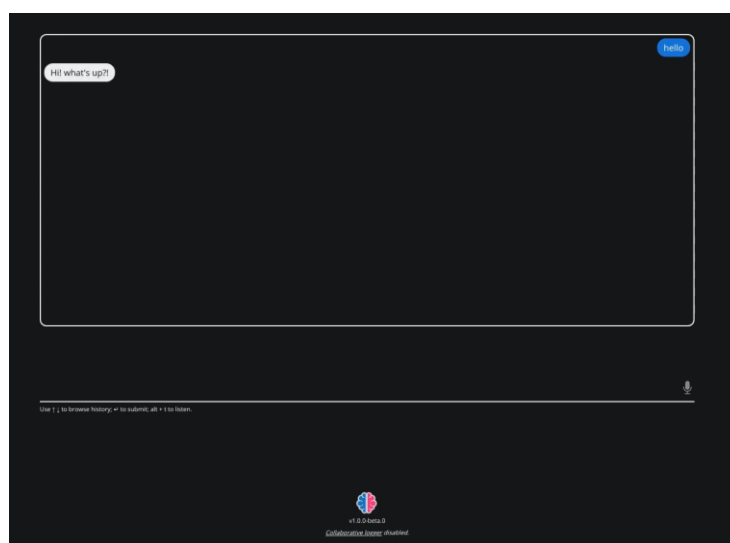


Рисунок 1.3 – Онлайн-асистент «Leon»

Системи рекомендацій. Такі система використовуються на багатьох сайтах онлайн покупок, але вони знаходяться на досить примітивному рівні. Ці системи вмонтовуються у магазин, збирають данні користувачів, їх відгуки, покупки, товари що вони продивляються, і на основі цього рекомендують їм те, що вони з більшою вірогідністю куплять [5].

1. Система рекомендацій інтернет-магазину «Rozetka».

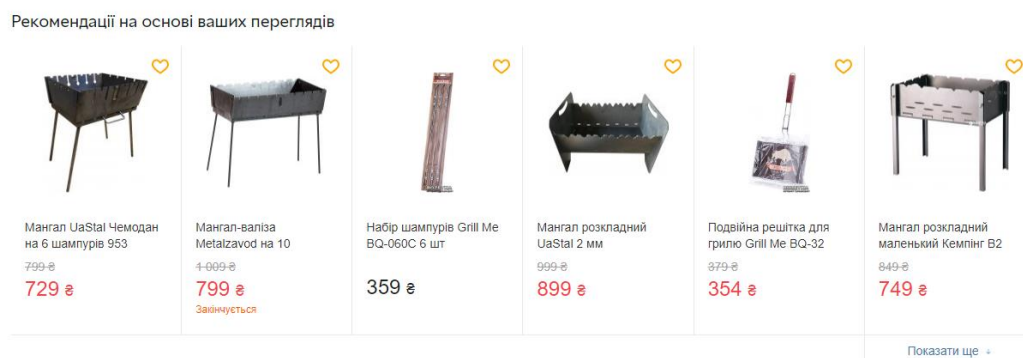


Рисунок 1.4 – Система рекомендацій «Rozetka»

Дана система збирає дані користувачів, включно з моїми. Аналізує їх, дивиться на минулі покупки, ціни і багато іншого, та рекомендує мені якісь товари, особливо акційні. Окрім даної рубрики рекомендацій є і інші.



Рисунок 1.5 – Система рекомендацій «Rozetka»

Ми не маємо змогу напряму спілкуватись з даним помічником, але користь від нього така сама як і він чат-ботів та голосових помічників.

Незважаючи на переваги, існуючі чат-боти для продажу товарів було також виявлено наступні недоліки:

- повністю платні рішення. Деякі чат-боти вимагають платну підписку або комісію з кожного продажу. Це може бути обтяжливим для бізнесу, особливо для малих підприємств з обмеженими фінансовими ресурсами;
- реклама. Частина чат-ботів можуть мати вбудовану рекламу, що може роздратовувати користувачів та негативно впливати на їхнє користувацьке досвід;
- обмежений функціонал. Інша частина чат-боти можуть бути обмежені у функціоналі, не надаючи можливості інтеграції з платіжними системами, аналітикою продажів та іншими додатковими можливостями, які можуть бути важливими для ефективного управління бізнесом;
- відсутність захисту від спаму повідомленнями користувачем;
- відсутність статистики використання телеграм-бота.

З урахуванням недоліків існуючих чат-ботів, розробка власного програмного модуля для телеграм-бота для продажу товарів може мати переваги, такі як більша контрольованість над функціоналом, відсутність реклами, український інтерфейс, зворотній зв'язок, гнучкість у розробці, відсутність постійної залежності від сторонніх розробників, захист від спаму, коротка статистика використання бота тощо.

1.3 Постановка задачі

Розроблюваний програмний модуль призначений для полегшення процесу покупки та забезпечення зручності та ефективності взаємодії між користувачем та бізнесом.

Після запуску бота, користувач повинен мати можливість переглянути каталог товарів, свій профіль, наявні товари, зв'язатись з підтримкою, а також отримати Frequently Asked Question (FAQ) [8].

У свою чергу запустивши бота від імені адміністратора, користувач отримає доступ до додаткової панелі для здійснення модерування товарів. А саме додавання, видалення та зміну як одного так і усіх розділів, товарів, позицій.

Додати можливість видалення усіх категорій, товарів чи позицій. Передбачити отримання короткої статистики використання бота. Статистика включатиме: кількість користувачів за день, тиждень, весь час; кошти за продажі за день, тиждень, весь час; кошти, які наявні в системі; кількість товарів, позицій, категорій, виставлених в даний час.

Необхідно розробити можливість автоматизовано в самому ботові змінити значення «FAQ» та контактної інформації підтримки.

Створити перемикачі роботи бота, такі як:

- «технічні роботи» – доступ до бота надається тільки адміністрації;
- «покупки» – користувач зможе запустити бота, але доступ буде лише до «FAQ» та «підтримки» (не стосується адміністрації).

Реалізувати можливість адміністраторам змінювати баланс, збільшувати баланс, переглядати покупки користувачів, відправляти особисті повідомлення.

Створити функцію розсилки. Ввести підтвердження на відправку розсилки. Після здійснення функції, визначити та відобразити адміністрації: час, який витрачено на розсилку; кількість користувачів всього; користувачі які отримали чи не отримали повідомлення.

Для вирішення наведеної задачі, слід вирішити такі завдання:

1. Здійснити аналіз предметної області;
2. Виконати огляд існуючих телеграм-ботів для продажу товарів та здійснити аналіз їх переваг та недоліків;
3. Розробити загальний алгоритм роботи телеграм-бота для продажу
4. товарів;
5. Виконати програмну реалізацію телеграм-бота для продажу товарів;
6. Здійснити тестування розробленого програмного забезпечення.

1.4 Висновок

Виконано аналіз предметної області, де розглянуто необхідність використання чат-ботів, як критично-необхідного процесу автоматизації у сучасному світі. Детально розглянуто області використання чат-ботів та проаналізовано основні задачі, які виконують телеграм-боти.

Обґрунтовано актуальність та доцільність розробки телеграм-бота для продажу товарів.

Здійснено огляд та проаналізовано популярні програми-аналоги чат-ботів для продажу товарів. Огляд показав, що розглянуті аналоги мають ряд недоліків, зокрема: відсутність внутрішнього балансу, який можна поповнювати та витрачати, відсутність вимикачів по типу «технічні роботи», «покупка товарів», відсутність статистики роботи бота, відсутність обробника помилок при різних ситуаціях при використанні бота, захист від спама. Тому доцільно створити власну вдосконалену версію телеграм-бота для продажу товарів.

Здійснено постановку задачі розробки телеграм-бота для продажу товарів, а саме визначено основні завдання, напрямки розвитку та необхідний функціонал програми, який полегшить взаємодію людина та бота.

2 РОЗРОБКА ТА АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ЧАТ-БОТА ДЛЯ ПРОДАЖУ ТОВАРІВ

2.1 Варіантний аналіз шляхів розв'язання поставленої задачі

Варіантний аналіз шляхів розв'язання поставленої задачі є важливим етапом проектування. У цьому розділі ми розглянемо два можливих шляхи реалізації: ручне програмування та використання конструкторів. Кожен з цих підходів має свої переваги та недоліки, тому проведемо детальний аналіз кожного з них.

Ручне програмування телеграм-бота вимагає глибоких знань і навичок у програмуванні. Для реалізації бота можна використати мови програмування, такі як Python, JavaScript тощо, та відповідні бібліотеки для взаємодії з Телеграм API. Основні переваги ручного програмування включають:

- Гнучкість і можливість повного контролю над функціоналом та поведінкою бота.
- Нестандартні можливості та можливість розширення за потреби.
- Глибоке розуміння архітектури бота і його внутрішньої логіки.
- Однак, ручне програмування також має свої недоліки:
- Великі затрати часу та ресурсів на розробку, особливо при складних функціональних вимогах.
- Висока складність підтримки і розширення бота в майбутньому.
- Вимоги до вмінь програмування та досвіду в розробці ботів.

Використання конструкторів телеграмботів надає можливість швидко створити прототип бота без необхідності в глибоких знаннях програмування. Деякі популярні конструктори, такі як Chatfuel [9] (рис. 2.1), ManyChat [10] (рис. 2.2), BotPress [11] (рис. 2.3), надають інтуїтивно зрозумілий інтерфейс для створення ботів.

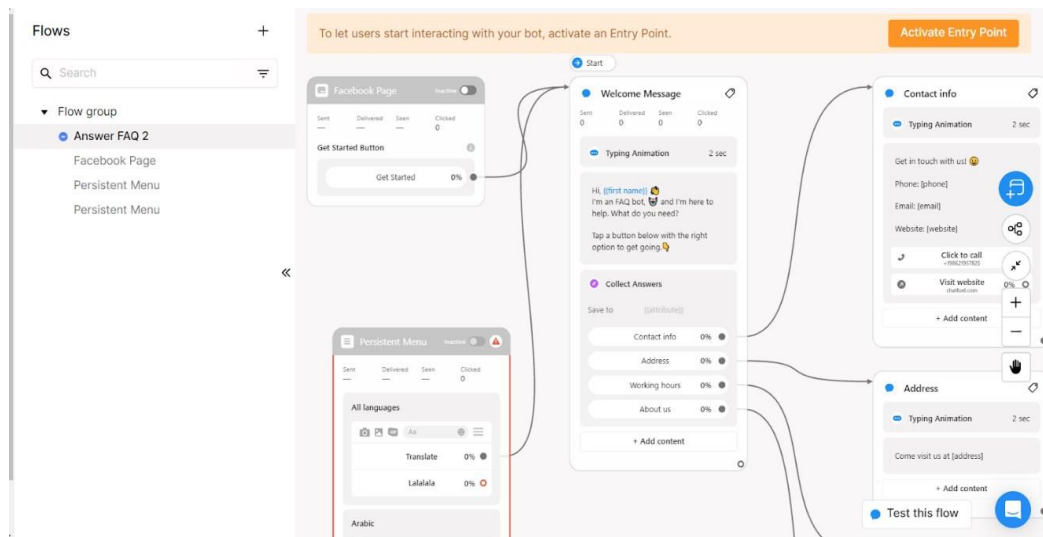


Рисунок 2.1 – Зовнішній вигляд конструктора Chatfuel

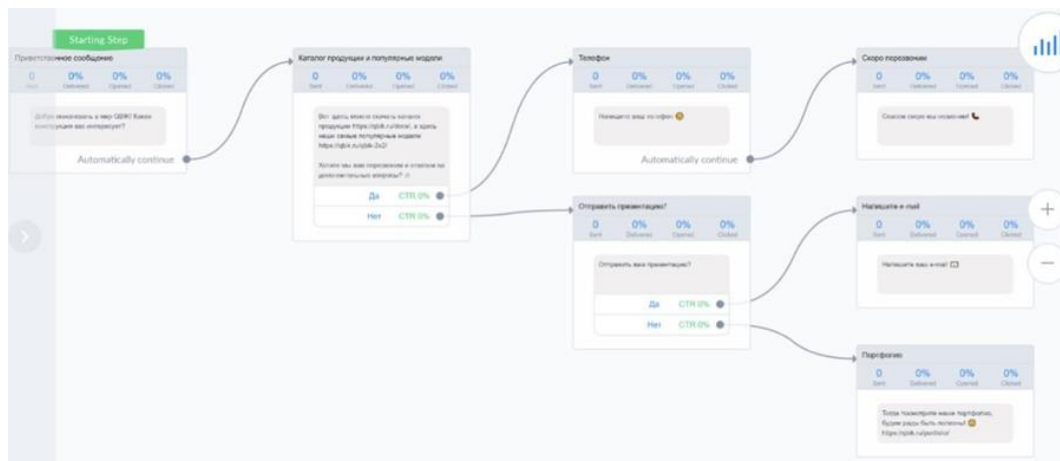


Рисунок 2.2 – Зовнішній вигляд конструктора ManyChat

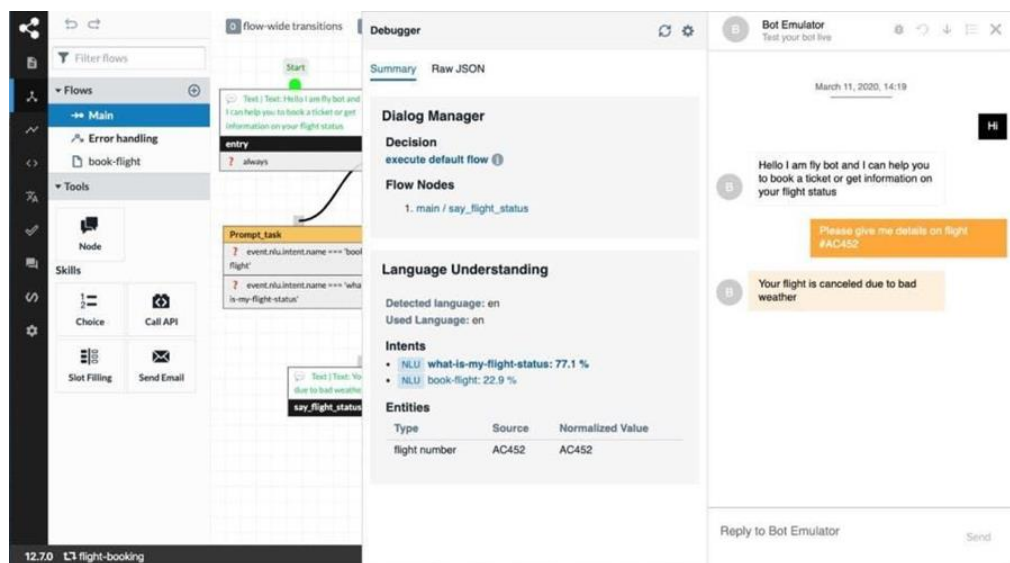


Рисунок 2.3 – Зовнішній вигляд конструктора BotPress

Основні переваги використання конструкторів включають:

- Швидкість розробки і простоту використання без необхідності в програмуванні.
- Наявність вбудованих інструментів для створення і керування ботом.
- Можливість використання готових шаблонів та інтеграцію з іншими сервісами.
- Проте, використання конструкторів також має свої обмеження та недоліки:
 - Обмежені можливості налаштування та гнучкості функціоналу бота.
 - Залежність від функціональності та обмежень конкретного конструктора.
 - Вартість підписки на деякі конструктори та можливі обмеження на кількість користувачів або повідомлень.

У результаті проведеного варіантного аналізу ми можемо зробити наступні висновки.

Ручне програмування дозволяє отримати повний контроль та гнучкість, але вимагає значних затрат часу та навичок програмування.

Використання конструкторів дозволяє швидко створити прототип бота без програмування, але має обмежені можливості та залежність від конструктора.

Саме тому, для реалізації роботи було вибране ручне програмування, оскільки враховуючи необхідність в розширеному функціоналі та для виконання окремих технічного завдання цей спосіб є кращим. Це дозволить забезпечити успішну реалізацію проекту з мінімальними затратами та найкращим функціоналом.

2.2 Розробка UML-діаграм варіантів використання чат-бота для користувача та адміністратора

На цьому етапі буде створено діаграми варіантів використання чат-бота користувачем (рис. 2.4) та адміністратором (рис. 2.5) [12]. Саме ці діаграма дозволять відобразити основні можливості кожного з зазначених класів.

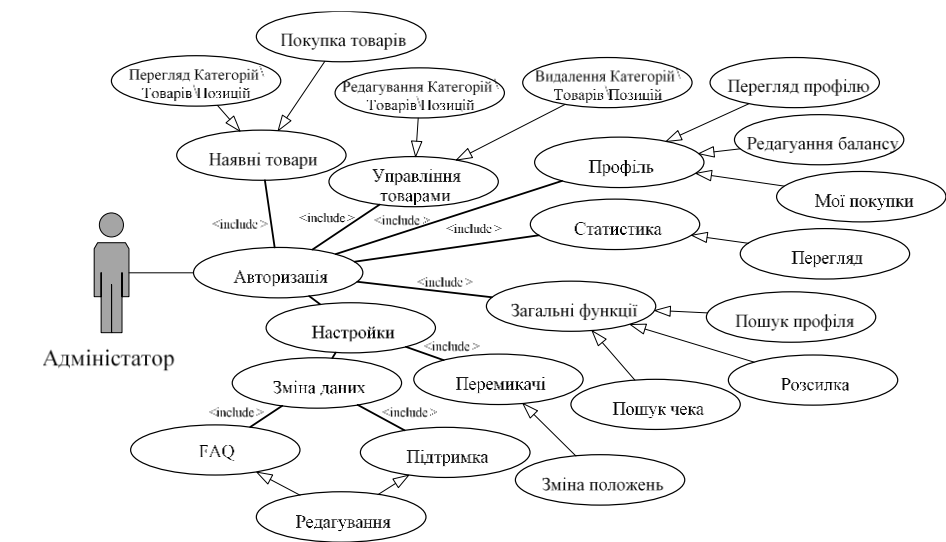


Рисунок 2.4 – UML-діаграма варіантів використання чат-бота для продажу товарів від імені адміністратора

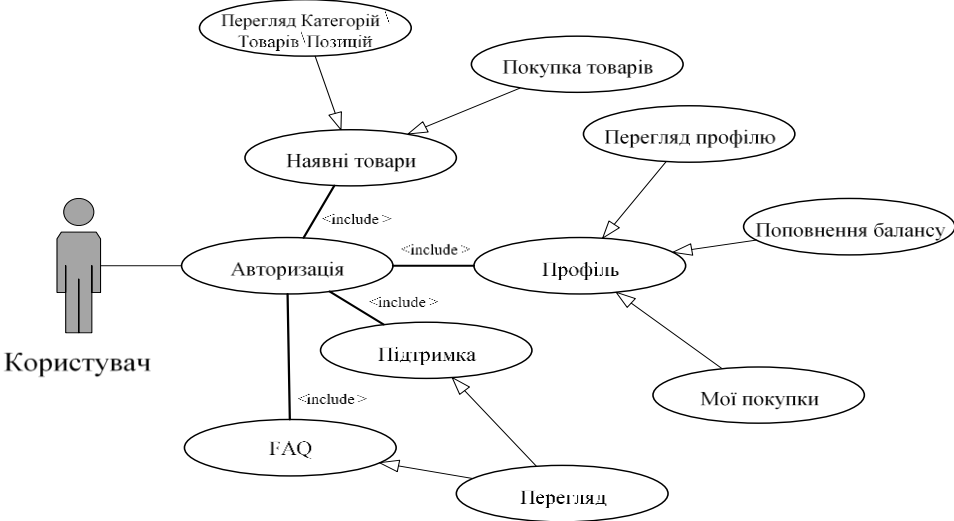


Рисунок 2.5 – UML-діаграма варіантів використання чат-бота для продажу товарів від імені користувача

2.3 Розробка загального алгоритму роботи чат-бота для продажу товарів

Алгоритм роботи бота визначає послідовність дій, які бот виконує при отриманні повідомлень від користувачів (рис 2.6).

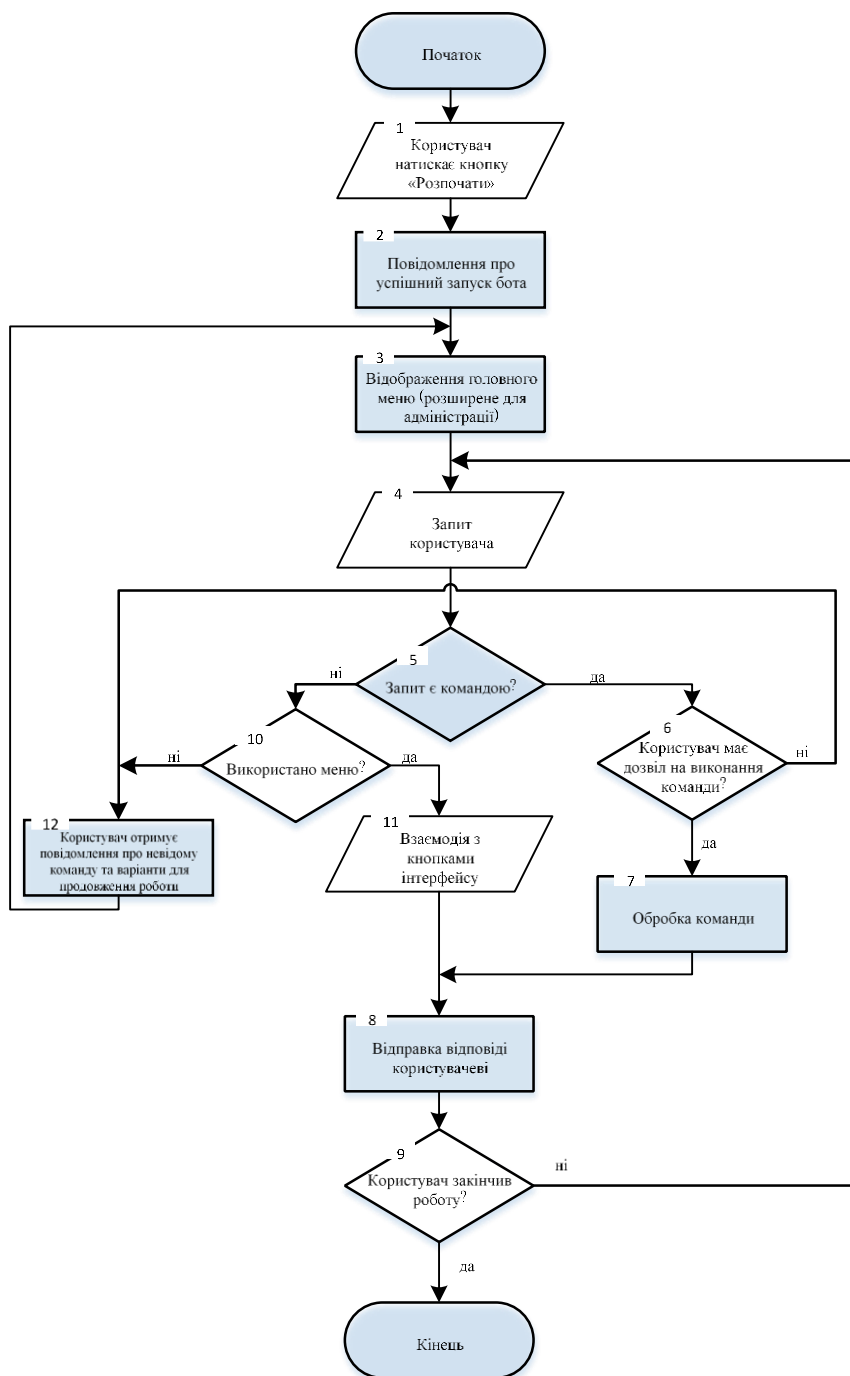


Рисунок 2.6 – Загальний алгоритм роботи чат-бота для продажу товарів

Основні кроки алгоритму роботи телеграм-бота включають наступне:

Крок 1-3: користувач натискає на кнопку «Розпочати». Бот запускається та надсилає привітальне повідомлення у відповідь. Відображається головне меню (розширене для користувачів з статусом адміністратор);

Крок 4: користувач відправляє запит (повідомлення або взаємодіє з головним меню);

Крок 5: бот приймає та аналізує запит користувача. Перевіряє чи заданий запит є командою. Якщо відповідь на попереднє запитання так, то бот переходить до кроку 6, якщо ні – крок 10;

Крок 6: бот перевіряє чи доступна команда введена в запиті користувачу з відповідним статусом (адміністратор\звичайний користувач). Якщо відповідь на попереднє запитання так, то бот переходить до кроку 7, якщо ні – крок 12;

Крок 7: відбувається обробка команди. Якщо отримане повідомлення є командою (наприклад, /start, /support, /faq), бот виконує відповідні дії відповідно до команди.

Крок 8: бот відправляє відповідь користувачеві. Відповідь може бути як у вигляді повідомлення так і у зміні кнопок інтерфейсу меню.

Крок 9: якщо користувач закінчив роботу з ботом, то він переходить до кінця виконання алгоритму, який передбачає закінчення роботи з ним. Якщо користувач продовжує відправляти запити, бот переходить циклічно до крока 4;

Крок 10: якщо вказаний запит користувачем – це використання меню, то бот переходить до кроку 11, якщо ні – крок 12;

Крок 11: бот має вмонтований механізм обробки невідомих команд, та відповідного повідомляє користувача про це. Припинення роботи: бот може бути зупинений за вимогою адміністратора (вимикачі) або при критичних помилках.

2.4 Розробка алгоритму функціонування головного меню чат-бота для продажу товарів

Розробка алгоритму функціонування головного меню чат-бота для продажу товарів (рис 2.7) є важливим етапом у створенні функціонального та ефективного бота. Першим кроком у розробці алгоритму є визначення основних функцій, які має виконувати телеграм-бот для продажу товарів. Це можуть бути функції, такі як перегляд каталогу товарів, звернення в підтримку, оформлення замовлення, обробка платежів тощо. При визначенні функцій необхідно враховувати потреби та очікування цільової аудиторії.

Після визначення функцій потрібно встановити логіку взаємодії користувача з ботом. Це включає в себе послідовність кроків, які користувач буде здійснювати для використання функцій бота. Наприклад, користувач може починати з головного меню, потім перегляд категорій/товарів, покупка товару або щось подібне.

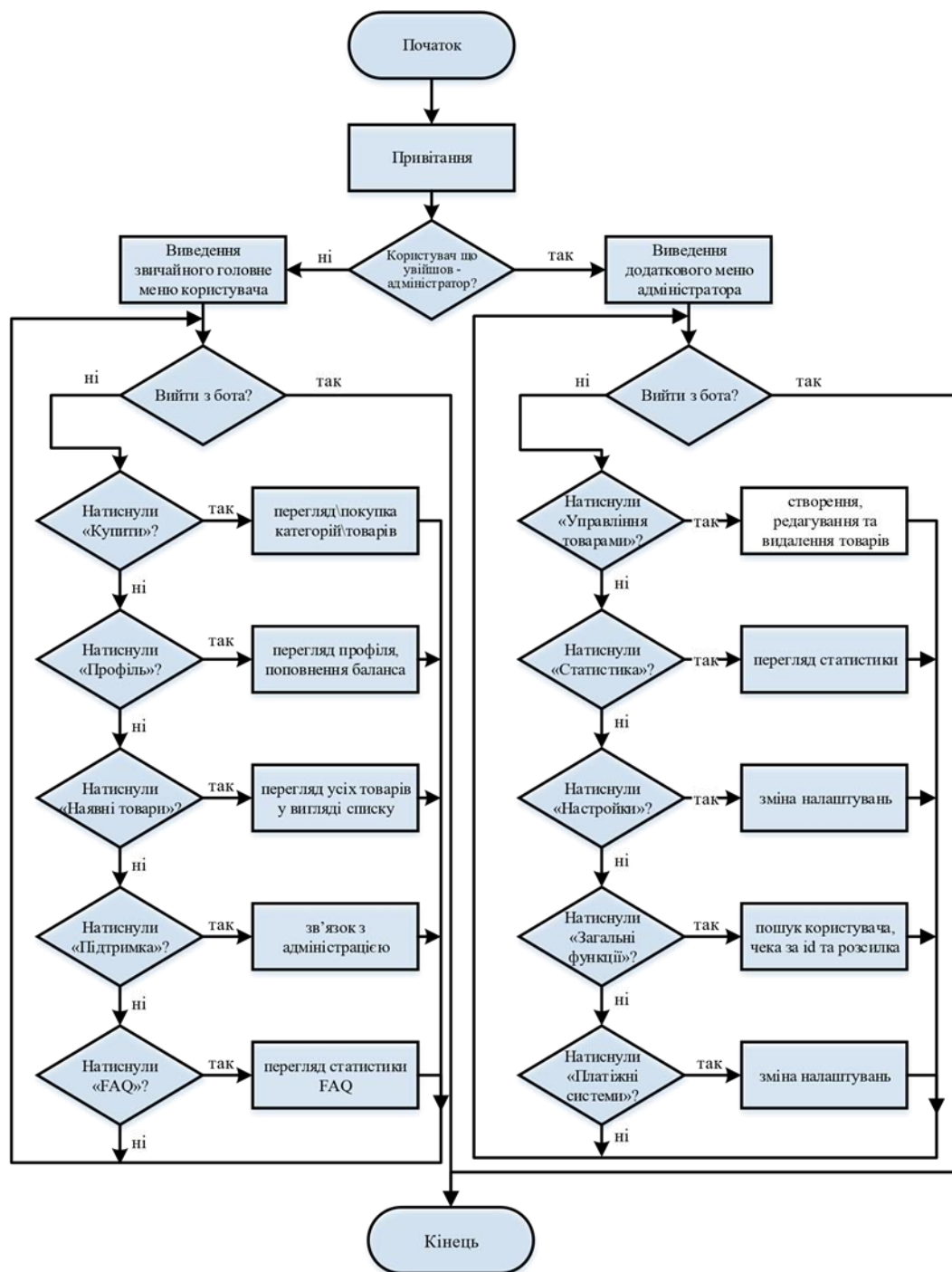


Рисунок 2.7 – Алгоритм роботи головного меню чат-бота для продажу товарів

Після визначення функцій та взаємодії необхідно реалізувати алгоритм роботи головного меню телеграм-бота для продажу товарів. Опираючись на вищесказане загальний алгоритм роботи меню буде виглядати наступним чином.

2.5 Алгоритм запуску чат-бота для продажу товарів

Розробимо алгоритм який буде використовуватись при запуску телеграмбота, розглянемо основні моменти та вигляд в середовищі розробки. Алгоритм запуску телеграм-бота для продажу товарів зображено на рисунку 2.8.

Коротко опишемо що відбувається:

1. Імпорт необхідних модулів і пакетів.
2. Ініціалізація **colorama** для підтримки кольорів у консолі.
3. Імпорт функцій та об'єктів з інших модулів та пакетів.
4. Оголошення функцій **scheduler_start** і **on_startup**, які будуть виконуватись пізніше.

Функція **scheduler_start** налаштовує планувальник (scheduler) для виконання різних завдань у певні моменти часу. Зокрема:

- **update_profit_day** буде виконуватись щоденно о 00:00.
- **update_profit_week** буде виконуватись щопонеділка о 00:01.
- **autobackup_admin** буде виконуватись щоденно о 00:00.

Функція **on_startup** виконується після запуску бота і виконує такі дії:

- створюється асинхронна сесія rSession.
- записується сесія rSession в поле rSession об'єкта bot.
- встановлюються команди бота.
- перевіряються дані бота.
- запускається планувальник за допомогою функції **scheduler_start**.
- відправляється повідомлення про початок роботи бота.
- виводяться повідомлення у консоль.

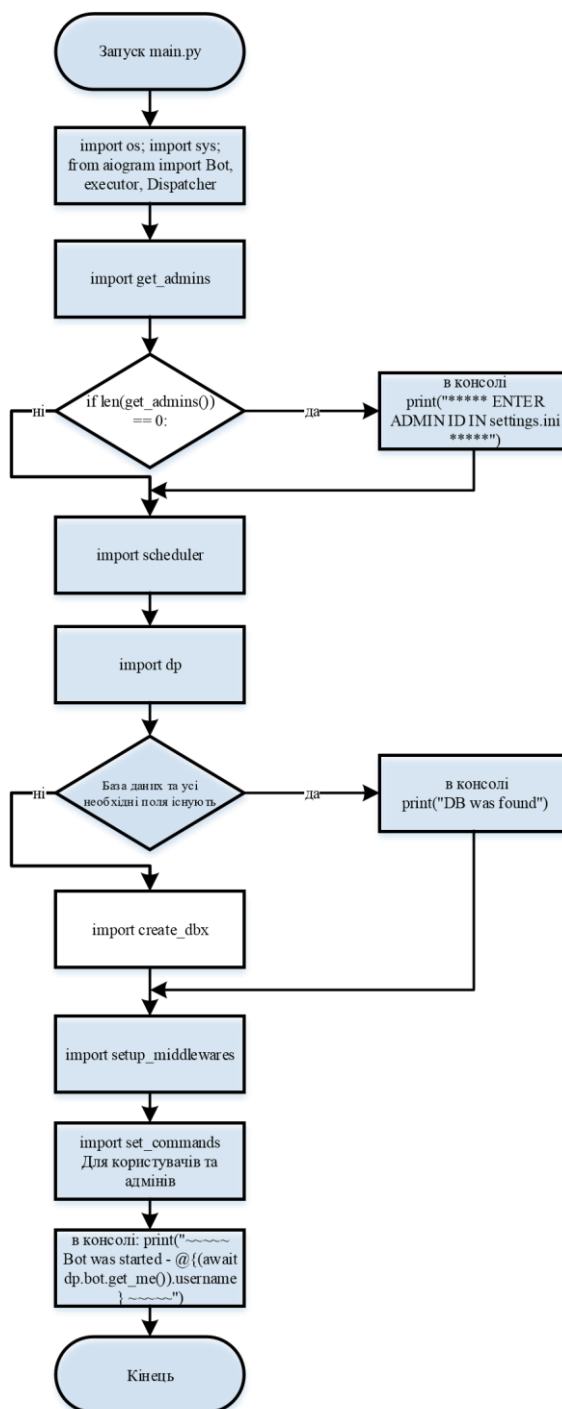


Рисунок 2.8 – Алгоритм запуску чат-бота для продажу товарів

2.6 Висновок

Виконано варіантний аналіз шляхів розв'язання поставленої задачі. Розглянуто два основні варіантами реалізації чат-бота: ручне програмування та використання конструкторів. Конструктори надають швидшу і простішу

реалізацію, але можуть бути обмежені у можливостях налаштування. Ручне програмування вимагає більшого обсягу роботи і знань програмування, але забезпечує більшу гнучкість і контроль над функціональністю, тому обираємо саме його.

Розроблено UML-діаграми варіантів використання чат-бота для користувача та адміністратора, що дозволило зрозуміти різні можливості та функціональність бота. Це дало чітке уявлення про взаємодію користувачів з ботом та потрібні функціональні модулі.

Розроблено загальний алгоритм роботи чат-бота для продажу товарів.

Розроблено алгоритм функціонування головного меню чат-бота для продажу товарів, що включає послідовність кроків та операцій, необхідних для забезпечення зручного та ефективного використання системи навігації користувачем.

Розроблено та проаналізовано алгоритм запуску телеграм-бота для продажу товарів. Алгоритм передбачає імпортування необхідних бібліотек та директив, а також оголошення та виконання запуску планувальника.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТА ДЛЯ ПРОДАЖУ ТОВАРІВ

3.1 Обґрунтування вибору мови програмування

У цьому пункті ми розглянемо різні мови програмування, які можуть бути використані для реалізації телеграм-бота для продажу товарів. З метою вибору найкращого варіанту, порівняємо кілька актуальних мов програмування та врахуємо їх характеристики та можливості.

Python:

Python є однією з найпопулярніших мов програмування для розробки телеграм-ботів. Вона відрізняється легкістю вивчення, зрозумілим синтаксисом та великою кількістю бібліотек та фреймворків. Зокрема, для роботи з Telegram API існує потужна бібліотека `aioogram`, яка спрощує взаємодію з платформою Telegram та надає широкі можливості для реалізації функціональності телеграмбота.

JavaScript:

JavaScript є ще однією популярною мовою програмування, яка може бути використана для розробки телеграм-бота. Вона використовується для реалізації фронтенд-розробки веб-додатків, але також може бути використана для програмування телеграм-бота за допомогою бібліотеки `Telegraf` або використанням `Node.js` з Telegram API. Java:

Java є мовою програмування, яка широко використовується у розробці серверних додатків та має багато бібліотек та фреймворків. Для реалізації телеграм-бота в Java можна використовувати бібліотеку `TelegramBots`, яка надає зручний доступ до Telegram API та спрощує розробку функціональності бота.

C#:

C# є мовою програмування, яка використовується головним чином у розробці програм для платформи .NET. Для реалізації телеграм-бота в C# можна

використовувати бібліотеку Telegram.Bot, яка надає зручний доступ до Telegram API та дозволяє легко реалізувати функціональність бота.

Ruby:

Ruby є мовою програмування з простим синтаксисом та акцентом на зручність розробки. Для реалізації телеграм-бота в Ruby можна використовувати бібліотеку telegram-bot-ruby, яка надає можливості для роботи з Telegram API та реалізації бота.

Враховуючи вищезазначені мови програмування, ми зробили висновок, що Python є найбільш підходящою мовою для реалізації телеграм-бота для продажу товарів. Його простота вивчення, велика кількість доступних бібліотек та фреймворків, а також підтримка aiogram роблять його ефективним вибором для нашого проекту.

3.2 Вибір стеку технологій для реалізації чат-бота

При розробці чат-бота для продажу товарів був використаний певний стек технологій, що дозволив ефективно реалізувати функціональність бота та забезпечити зручний та швидкий процес взаємодії з користувачами. Нижче наведений перелік бібліотек, які були задіяні в процесі розробки:

- ***aiogram***: Ця бібліотека є потужним інструментом для розробки телеграм-ботів на мові програмування Python. Вона надає зручний API для взаємодії з Telegram API та дозволяє реалізувати різноманітну функціональність, включаючи обробку повідомлень, створення кнопок, роботу з клавіатурами тощо.
- ***APScheduler***: Ця бібліотека дозволяє розкладати та виконувати задачі в задані моменти часу або з певною періодичністю. Вона була використана для реалізації автоматичного відправлення повідомлень або оновлення даних в телеграм-боті.

- **colorlog** та **colorama**: Ці бібліотеки надають можливості для забарвлення та налаштування кольорів виводу логів в консолі. Вони дозволили зробити логування більш зручним та інформативним.
- **beautifulsoup4** та **bs4**: Ці бібліотеки були використані для парсингу та аналізу HTML-сторінок. Вони дозволяють отримувати необхідну інформацію з веб-сторінок та виконувати необхідні обробки даних.
- **aiohttp**: Ця бібліотека забезпечує зручний інтерфейс для здійснення асинхронних HTTP-запитів. Вона була використана для взаємодії з зовнішніми сервісами, такими як система платежів або база даних.
- **typing**: Ця бібліотека дозволяє використовувати анотації типів в мові програмування Python. Вона сприяє покращенню читабельності коду та робить розробку більш надійною.
- **pytz**: Ця бібліотека надає функціональність для роботи з часовими зонами. Вона була використана для коректного відображення та обробки дат і часу в телеграм-боті.

Цей перелік бібліотек представляє лише деякі складові стеку технологій, використаних при розробці телеграм-бота для продажу товарів.

3.3 Обґрунтування вибору платформи для розробки чат-бота

У цьому розділі ми розглянемо вибір платформи для розробки нашого чат-бота для продажу товарів. Наш вибір падає на Microsoft VS Code, яке є чудовою платформою для розробки на різних мовах, включаючи Python.

Microsoft VS Code (VS Code) – це зручне та потужне середовище розробки з відкритим вихідним кодом, яке надає багато можливостей для роботи з різними мовами програмування. Воно підтримує розширення для багатьох мов, включаючи Python, що робить його ідеальним вибором для розробки нашого телеграм-бота.

Деякі переваги Microsoft VS Code:

- Мультиплатформеність: VS Code підтримує операційні системи Windows, macOS та Linux, що дозволяє розробникам працювати на своїх улюблених платформах.
- Розширення для Python: VS Code має широкий вибір розширень для роботи з Python. Наприклад, розширення Python є офіційним розширенням від Microsoft, яке надає інтеграцію з інтерпретатором Python та багато інших корисних функцій.
- Легкість використання: VS Code має інтуїтивний і дружній для користувача інтерфейс, що робить його простим у використанні навіть для початківців. Він надає можливість швидко налаштувати проект, керувати версіями за допомогою систем контролю версій та виконувати різні операції з файлами та теками.
- Розширені можливості редагування: VS Code має багато корисних функцій редагування, таких як автодоповнення коду, підказки, перехід до визначення функцій та багато іншого. Це допомагає покращити продуктивність розробника та зменшити кількість помилок.
- Інтеграція з Git: VS Code має вбудовану підтримку системи контролю версій Git, що дозволяє розробникам легко керувати своїми проектами, використовуючи Git команди безпосередньо з інтерфейсу VS Code.

Однією з переваг використання Microsoft VS Code для розробки нашого телеграм-бота є те, що розробник (у нашому випадку) вже має досвід використання цього середовища. Це дозволяє зменшити час на ознайомлення з новими інструментами та більш ефективно використовувати існуючі навички.

Microsoft VS Code є потужним та гнучким середовищем розробки, яке надає широкий спектр можливостей для розробки на різних мовах програмування. Використання VS Code для розробки нашого телеграм-бота дозволить нам зосередитись на розробці функціональності бота та забезпечити ефективне та

зручне середовище для нашої роботи. На рисунку 3.1 зображено інтерфейс платформи VS Code (рис. 3.1).

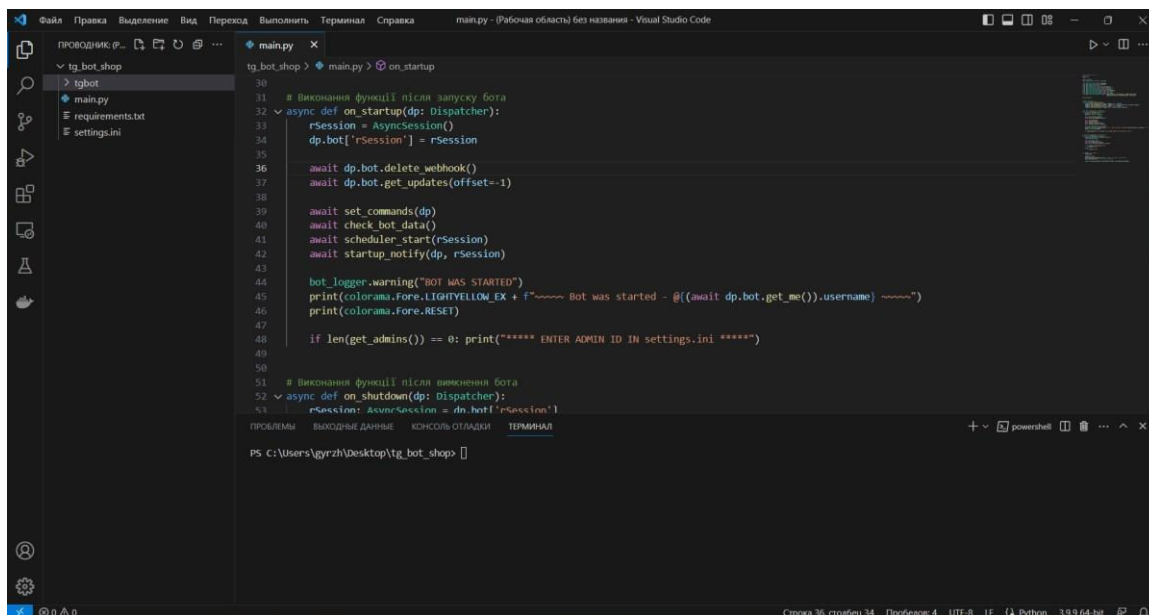


Рисунок 3.1 – Зовнішній вигляд інтерфейсу Microsoft VS Code

3.4 Аналіз процесу створення оболонки чат-бота з використанням «BotFather»

BotFather – найпростіший спосіб для реєстрації, налаштування та управління іншими telegram-ботами. Робота з ним проста і не вимагає специфічних навичок. За допомогою BotFather можна зареєструвати необмежену кількість нових чат-ботів. Єдиною умовою для реєстрації нового бота є його унікальний username. На рисунку 3.2 зображено команди чат-бота «BotFather».

Процес створення телеграм-бота наведено на рисунку 3.3. Взаємодія з BotFather здійснюється за допомогою простих команд. Наприклад, для того, щоб зареєструвати нового бота, досить відправити в чат команду /newbot і слідувати простим інструкціям:

1. **придумати ім'я бота**, яке буде відображатися в чатах і контактах. Надалі його можна буде змінити. Тут все залежить тільки від вашої фантазії і вимог;

2. **придумати username.** Ім'я повинно бути унікальним і закінчуватися на «bot». Допускаються букви латинського алфавіту, цифри і символ підкреслення (приклад – «MySuper_wow_bot»). Загальна кількість символів не менше 5 і не більше 32;

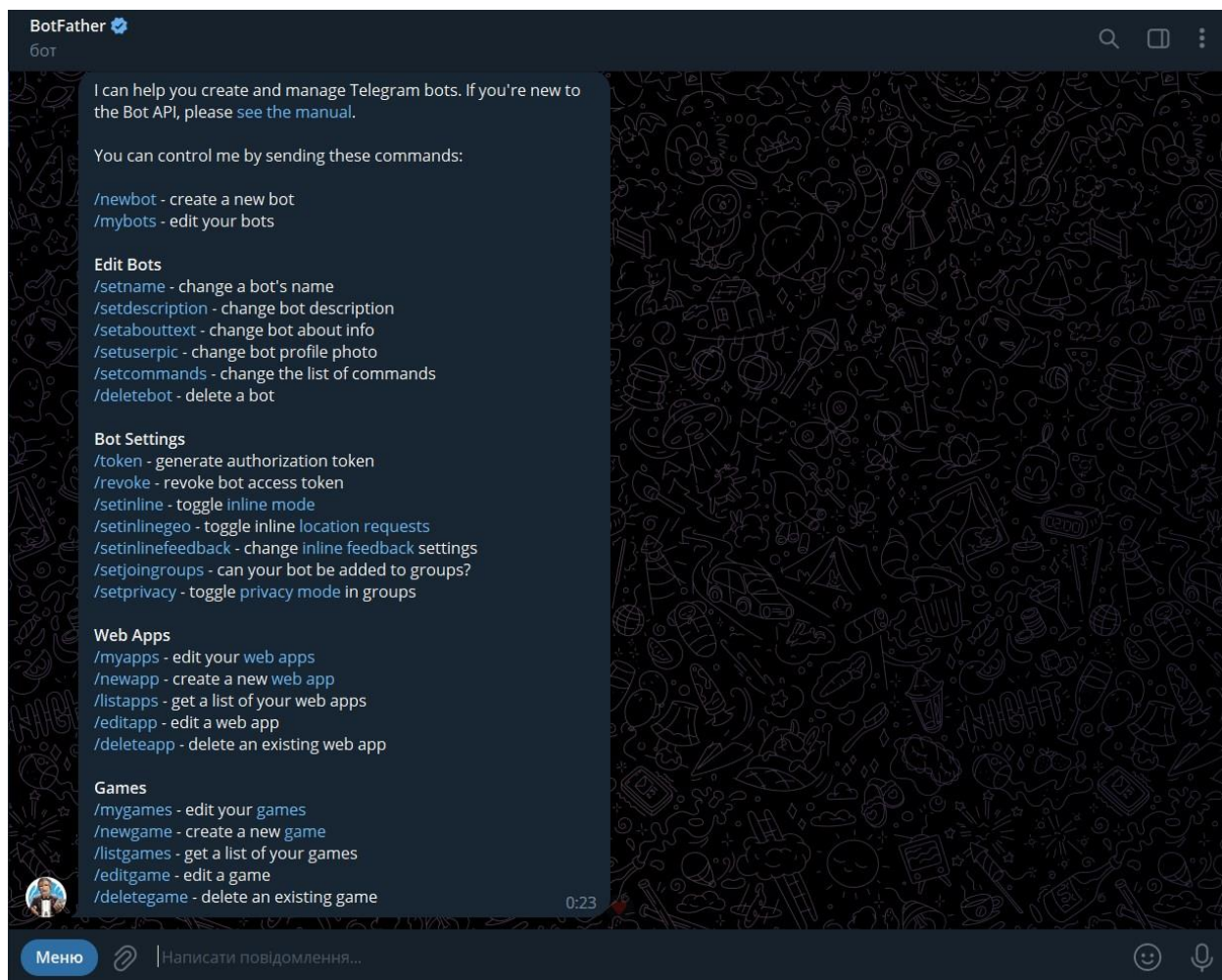


Рисунок 3.2 – Команди чат-бота «BotFather»

3. Якщо все виконано правильно, то у відповідь ми отримаємо **повідомлення з токеном** (рис. 3.3). Токен необхідний для роботи з Bot API за допомогою http-протоколу. Не можна передавати його іншим і бажано не втрачати. Хорошим рішенням буде скопіювати його: зберегти в текстовий файл і перемістити в надійне, завжди доступне місце – наприклад, хмарне сховище.

Після реєстрації можна приступати до облагороджування бота: встановити аватар, додати опис та інше.

Крім реєстрації бота, за допомогою BotFather можна здійснювати його налаштування і управління.

За потребою можна змінити ім'я бота, використавши команду `/setname` (рисунок 3.4).

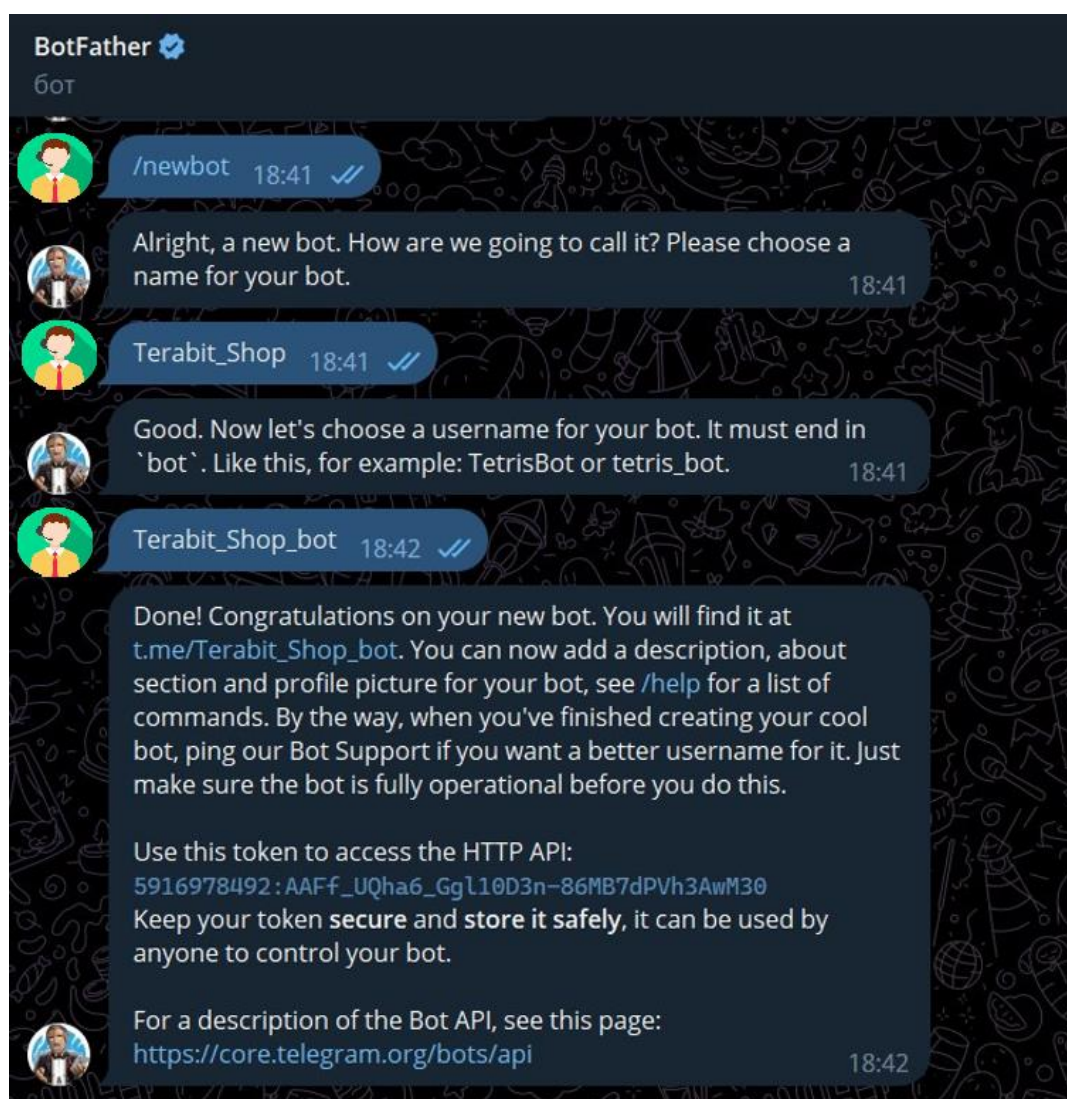


Рисунок 3.3 – Отримання повідомлення з токеном

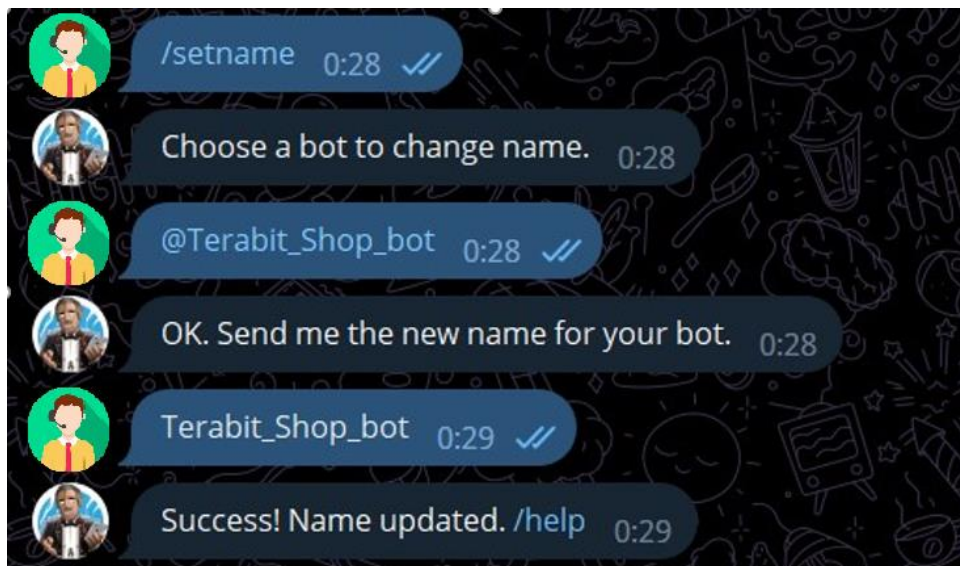


Рисунок 3.4 – Зміна імені бота

Встановити аватарку можна командою `/setuserpic`, а змінити або додати короткий опис – командою `/ setdescription`. Зміна інформації в профілі виконується командою `/ setabouttext`. На цьому кастомізацію зовнішнього вигляду бота можна вважати закінченою, і можна ознайомитися його функціональними можливостями. Рисунок 3.5 зображено процес зміни аватарки з використання можливостей BotFather.

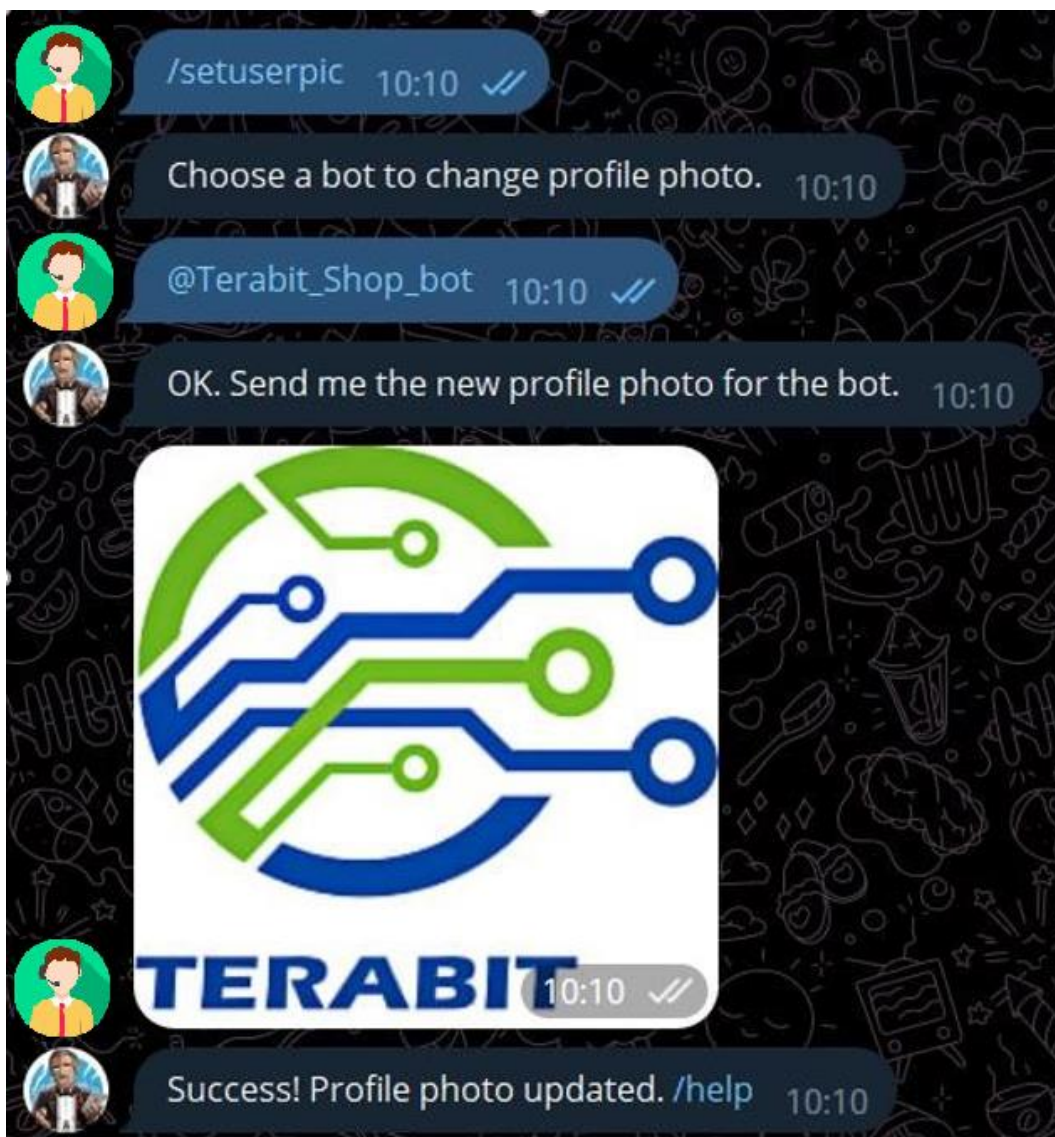


Рисунок 3.5 – Зміна аватарки бота з використання можливостей BotFather

Інші команди наведено на рисунку 3.2.

3.5 Створення інтерфейсу чат-бота для продажу товарів

Визначення основних модулів телеграм-бота є важливим етапом розробки, оскільки від цього залежить організація програмного забезпечення та структура бази даних.

Структура та функції основних модулів телеграм-бота наведені у таблиці 3.1.

Таблиця 3.1 – Функції модулів

Назва модуля	Основні функції
main.py	Імпорт бібліотек, бази даних, перевірка наявності адміністраторів; Запуск планувальника; Виконання функцій після запуску бота; Виконання функцій після вимкнення бота.
config.py	Читання токена бота; Назначення адміністрації бота;
main_start.py	Статус технічних робіт; Статус покупок; Статус поповнень; Привітальне повідомлення при запускові бота.
admin_functions.py	Розсилка; Пошук профіля\чека; Управління профілем
admin_products.py	Створення, редагування та видалення категорій\товарів\позицій
keyboards	Редагування кнопок меню для користувачів та адміністрації
api_sqlite.py	Запити до БД; Створення усіх таблиць для БД

На рисунку 3.6 наведено структуру модулів чат-бота.

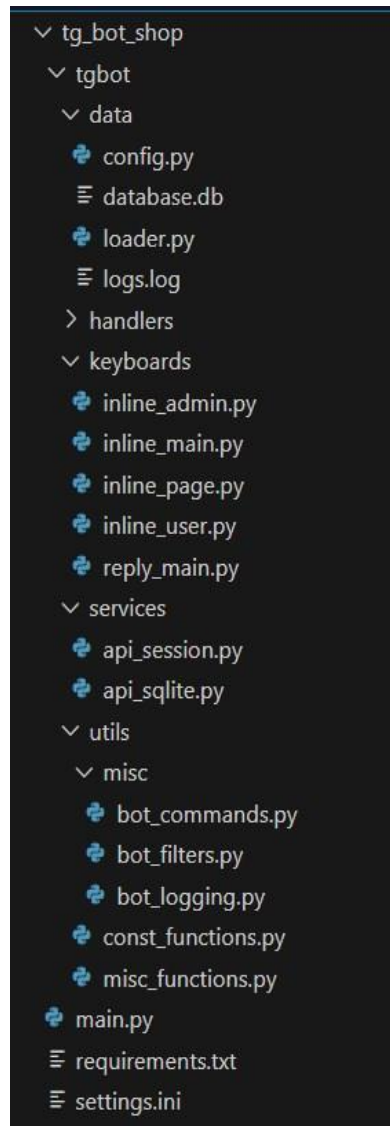


Рисунок 3.6 – Структуру програмних модулів чат-бота для продажу товарів

Якщо не було вказано жодного адміністратора (`get_admins ()` повертає порожній список), виводиться повідомлення `***** ENTER ADMIN ID IN settings.ini *****` для попередження (рисунок 3.7).

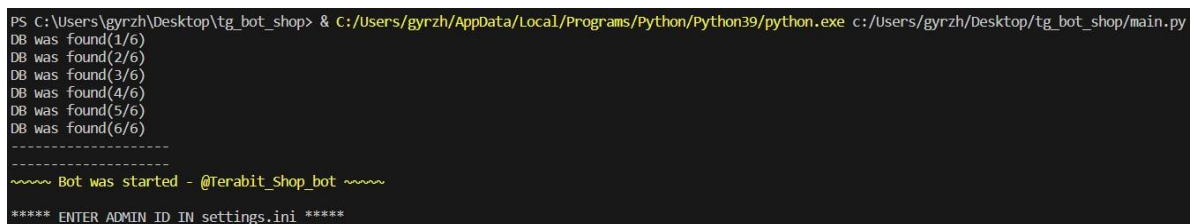


Рисунок 3.7 – Вигляд консолі під час запуску чат-бота для продажу товарів та повідомлення попередження про відсутність id адміністрації

Частина коду яка реалізує наведений вищевказаний алгоритм наведена далі:

```
# - *- coding: utf- 8 - *-
import os import sys

import colorama
from aiogram import executor, Dispatcher

from tgbot.data.config import get_admins from
tgbot.data.loader import scheduler from
tgbot.handlers import dp
from tgbot.middlewares import setup_middlewares from
tgbot.services.api_session import AsyncSession from
tgbot.services.api_sqlite import create_dbx from
tgbot.utils.misc.bot_commands import set_commands from
tgbot.utils.misc.bot_filters import IsPrivate from
tgbot.utils.misc.bot_logging import bot_logger from
tgbot.utils.misc_functions import (check_bot_data,
startup_notify, update_profit_day, update_profit_week,
autobackup_admin) colorama.init ()

# Запуск планувальника async def
scheduler_start (rSession):
    scheduler.add_job (update_profit_day, trigger="cron", hour=00)
    scheduler.add_job (update_profit_week, trigger="cron",
    day_of_week="mon", hour=00, minute=1)
    scheduler.add_job (autobackup_admin, trigger="cron", hour=00)

# Виконання функції після запуску бота async
def on_startup (dp: Dispatcher):
    rSession = AsyncSession ()
    dp.bot['rSession'] = rSession

    await dp.bot.delete_webhook ()
    await dp.bot.get_updates (offset=-1)

    await set_commands (dp)      await
    check_bot_data ()          await
    scheduler_start (rSession)  await
    startup_notify (dp, rSession)

    bot_logger.warning ("BOT WAS STARTED")
    print (colorama.Fore.LIGHTYELLOW_EX + f"~~~~~ Bot was started
- @{ (await dp.bot.get_me ()) .username} ~~~~~")
    print (colorama.Fore.RESET)

    if len (get_admins ()) == 0: print ("***** ENTER ADMIN ID IN
settings.ini *****")
```

Розроблено інтерфейс відповідно до алгоритму роботи головного меню (рис. 2.7) для користувачів та адміністрації зображено на рисунках 3.8 – 3.9.

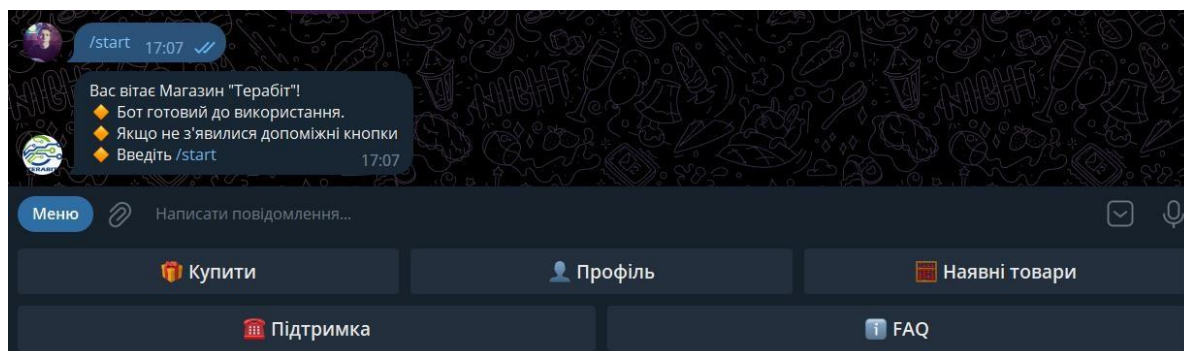


Рисунок 3.8. – Інтерфейс чат-бота для продажу товарів від імені звичайного користувача

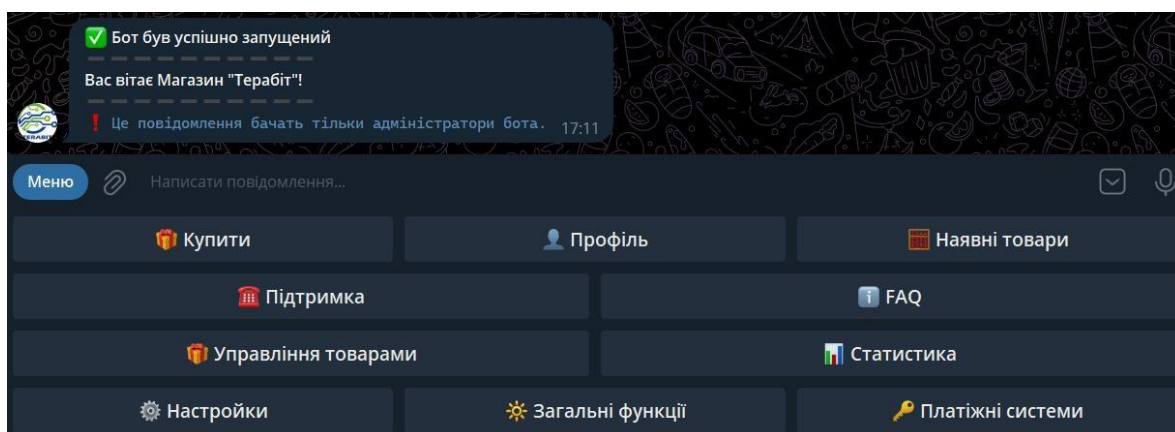


Рисунок 3.9. – Інтерфейс чат-бота для продажу товарів від імені адміністратора

3.6 Аналіз та обґрунтування вибору бази даних для розроблюваного чат-боту

В процесі обґрунтування вибору бази даних для телеграм-бота для продажу товарів було проведено порівняльний аналіз трьох СУБД: SQLite3, MySQL та PostgreSQL. Кожна з них має свої переваги та недоліки:

Переваги SQLite3:

- легкість використання та розгортання.
- низькі вимоги до ресурсів.
- інтеграція на рівні файлу.
- підтримка sql.
- надійність та цілісність даних.
- Недоліки SQLite3:
- менша продуктивність при роботі з великим обсягом даних порівняно

з серверними СУБД.

- обмежені можливості масштабування в розподіленому середовищі.
- Переваги MySQL:
- серверна СУБД з високими можливостями масштабування.
- висока продуктивність при обробці великого обсягу даних.
- розширені можливості для роботи з реляційними базами даних.
- Недоліки MySQL:
- вищі вимоги до ресурсів.
- більш складний процес налаштування та розгортання. Переваги

PostgreSQL:

- масштабована та потужна СУБД.
- підтримка розширень та додаткових функцій.
- повна підтримка стандартів sql. Недоліки PostgreSQL:
- вищі вимоги до ресурсів.
- більш складний процес налаштування та розгортання.

Порівнюючи характеристики СУБД, можна зробити висновок, що SQLite3 є найкращим варіантом для розробки телеграм-бота для продажу товарів, особливо з урахуванням наступних факторів:

- легкість використання та розгортання дозволяють швидко розпочати роботу з базою даних.
- низькі вимоги до ресурсів дозволяють економити на обчислювальних ресурсах.

- інтеграція на рівні файлу спрощує збереження та передачу бази даних.
- підтримка SQL забезпечує зручність у роботі з даними та запитами.
- надійність та цілісність даних гарантують правильну та безпечну роботу з базою.

3.7 Розробка бази даних для чат-бота для продажу товарів

Розробка бази даних є важливою складовою частиною телеграм-бота для продажу товарів. У цьому пункті буде описано процес розробки бази даних та важливі елементи, що входять до її складу.

Вибір системи управління базами даних (СУБД):

В попередніх розділах було вибрано СУБД. У даному проекті ми використовуємо SQLite3 – легку та компактну СУБД, яка ідеально підходить для невеликих проектів, таких як телеграм-бот. Як зазначалось однією з переваг SQLite3 є те, що база даних зберігається у звичайному файлі, що дозволяє легко переносити та резервувати дані.

Автоматичне створення бази даних:

SQLite3 має вбудовану функціональність, яка дозволяє автоматично створювати базу даних при запуску бота. Тобто база даних буде автоматично створена, якщо її не існує, і готова для використання.

Роботу над базою даних виконує модуль «api_sqlite.py». Безпосередньо перед форматування запитів, необхідно імпортувати бібліотеки, вказати шлях до файлу самої БД (рис. 3.10).

```

1  # -*- coding: utf- 8 - *-
2  import math
3  import random
4  import sqlite3
5  💡
6  from tgbot.data.config import PATH_DATABASE
7  from tgbot.utils.const_functions import get_unix, get_date, clear_html

```

Рисунок 3.10 – Імпортування бібліотек та файлу БД в модулі «api_sqlite.py»

Структура таблиць:

Після створення бази даних потрібно визначити структуру таблиць, що будуть використовуватися для збереження даних. Кожна таблиця має свою структуру, включаючи назви полів, їх типи даних та обмеження.

Розробка бази даних є важливим етапом реалізації телеграм-бота для продажу товарів. Використання SQLite3 дозволяє легко створювати та керувати базою даних, а автоматичне створення бази даних спрощує процес розгортання бота. Правильна структура бази даних допомагає забезпечити ефективну роботу бота та збереження важливої інформації.

Структура та короткий опис існуючих таблиць бази даних наведена далі. Таблиця для категорій – «storage_category» таблиці 3.2.

Таблиця 3.2 – Категорії

Назва поля	Тип
increment	INTEGER
category_id	INTEGER
category_name	TEXT

В таблиці «storage_category» зберігається інформація про існуючі категорії товарів. А саме в неї додаються, видаляються або редагуються уже існуючі категорії.

Таблиця для товарів – «storage_item» таблиці 3.3.

Таблиця 3.3 – Товари

Назва поля	Тип
increment	INTEGER
item_id	INTEGER
item_data	TEXT
position_id	INTEGER
category_id	INTEGER
creator_name	TEXT
creator_id	INTEGER
add_date	TIMESTAMP

В таблиці «Товари» зберігається інформація про id товару, id позиції, id категорії, ім'я створювача, id створювача та зазначається дата створення.

Наступна таблиця зберігає дані створених позицій – «storage_position».

Таблиця 3.4 – Позиції

Назва поля	Тип
increment	INTEGER
position_id	INTEGER
position_name	TEXT
position_price	INTEGER
position_description	TEXT
position_photo	TEXT
position_date	TIMESTAMP
category_id	INTEGER

В таблиці «Товари» зберігається інформація про id позиції, її ім'я, ціна, опис, фото чат створення та id категорії в якій знаходиться.

Наступна таблиця зберігає дані виконаних покупок – «storage_purchases».

Таблиця 3.4 «Покупки»

Назва поля	Тип
increment	INTEGER
user_id	INTEGER
purchase_receipt	TEXT
purchase_count	INTEGER
purchase_price	INTEGER
purchase_price_one	INTEGER
purchase_position_id	INTEGER
purchase_position_name	TEXT
purchase_item	TEXT
purchase_date	TIMESTAMP
balance_before	INTEGER
balance_after	INTEGER

В таблиці «Покупки» зберігається інформація про уже здійснені покупки. А саме: id користувача, назва позиції, кількість позиції, яка була придбана, ціна покупки, баланс до і після виконаної операції, id чека та інше.

В таблиці «storage_settings» зберігаються налаштування бота.

Таблиця 3.5 – Налаштування

Назва поля	Тип
status_work	TEXT
status_refill	TEXT
status_buy	TEXT
misc_faq	TEXT
misc_support	TEXT
misc_update	TEXT
misc_profit_day	INTEGER
misc_profit_week	INTEGER

В таблиці «Налаштування» зберігається інформація про стан вимикачів, текст FAQ, контакти підтримки, прибуток в день та в тиждень.

В таблиці «storage_users» зберігаються інформація про користувачів.

Таблиця 3.6 «Користувачі»

Назва поля	Тип
increment	INTEGER
user_id	INTEGER
user_login	TEXT
user_name	TEXT
user_balance	INTEGER
user_refill	INTEGER
user_date	TIMESTAMP

В таблиці «Користувачі» зберігається інформація про користувачів, які користуються телеграм-ботом. Вказується id користувача, його ім'я та логін, поточний баланс, поповнення та дата початку використання бота.

Приклади запитів до БД наведено на рисунках 3.11-3.13.

```
if len(con.execute("PRAGMA table_info(storage_users)").fetchall()) == 8:
    print("DB was found(1/6)")
else:
    con.execute("CREATE TABLE storage_users("
                "increment INTEGER PRIMARY KEY AUTOINCREMENT,"
                "user_id INTEGER,"
                "user_login TEXT,"
                "user_name TEXT,"
                "user_balance INTEGER,"
                "user_refill INTEGER,"
                "user_date TIMESTAMP,"
                ")")
    print("DB was not found(1/6) | Creating...")
```

Рисунок 3.11 – Створення таблиці «storage_users» зі зберіганням даних користувачів

```

# Додавання категорій
def add_categoryx(category_id, category_name):
    with sqlite3.connect(PATH_DATABASE) as con:
        con.row_factory = dict_factory
        con.execute("INSERT INTO storage_category (category_id, category_name) VALUES (?, ?)",
                    [category_id, category_name])

# Зміна категорій
def update_categoryx(category_id, **kwargs):
    with sqlite3.connect(PATH_DATABASE) as con:
        con.row_factory = dict_factory
        sql = "UPDATE storage_category SET"
        sql, parameters = update_format(sql, kwargs)
        parameters.append(category_id)
        con.execute(sql + "WHERE category_id = ?", parameters)

# Видалення категорій
def delete_categoryx(**kwargs):
    with sqlite3.connect(PATH_DATABASE) as con:
        con.row_factory = dict_factory
        sql = "DELETE FROM storage_category"
        sql, parameters = update_format_where(sql, kwargs)
        con.execute(sql, parameters)

# Видалення всіх категорій
def clear_categoryx():
    with sqlite3.connect(PATH_DATABASE) as con:
        con.row_factory = dict_factory
        sql = "DELETE FROM storage_category"
        con.execute(sql)

```

Рисунок 3.12 – Запити для виконання дій над категоріями товарів

```

def add_itemx(category_id, position_id, get_all_items, user_id, user_name):
    with sqlite3.connect(PATH_DATABASE) as con:
        con.row_factory = dict_factory

        for item_data in get_all_items:
            if not item_data.isspace() and item_data != "":
                con.execute("INSERT INTO storage_item "
                            "(item_id, item_data, position_id, category_id, creator_id, creator_name, add_date) "
                            "VALUES (?, ?, ?, ?, ?, ?, ?)",
                            [random.randint(1000000000, 9999999999), clear_html(item_data.strip()), position_id,
                             category_id,
                             user_id, user_name, get_date()])

```

Рисунок 3.13 – Додавання товару в таблицю «storage_item»

3.8 Тестування роботи чат-бота для продажу товарів

Функціональне тестування є одним з важливих етапів тестування роботи телеграм-бота для продажу товарів. Його основна мета – перевірити правильність реакції бота на повідомлення користувачів, коректність обробки команд і запитів, а також впевненість у відповідях та поведінці бота в різних сценаріях використання.

На початку роботи нас зустрічає інтерфейс з єдиною кнопкою «РОЗПОЧАТИ».

Після її натискання боту автоматично відправляється команда «/start». В свою чергу відображається привітальне повідомлення, меню (рис. 3.15) та кнопки інтерфейсу для користування розробленим чат-ботом.

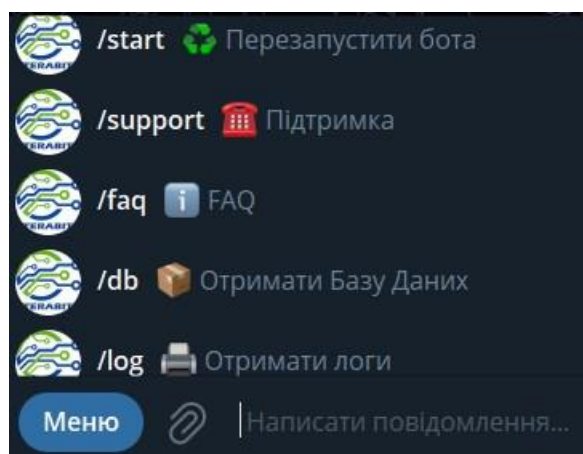


Рисунок 3.15 – Меню чат-бота для продажу товарів

Спроба отримати файли логів або бази даних не маючи статусу адміністратор зображено на рисунку 3.16.

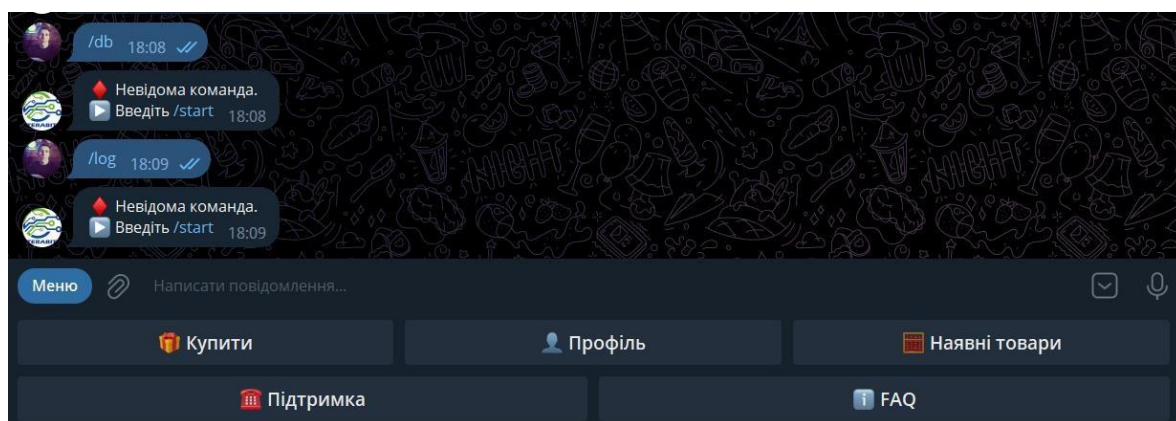


Рисунок 3.16 – Спроба отримати базу даних/лог не маючи статусу адміністратор

Оскільки у нас немає прав адміністратора, ми отримали повідомлення «невідома команда».

Використаємо можливості, що надають кнопки «Підтримка» та «FAQ» (рис. 3.17):

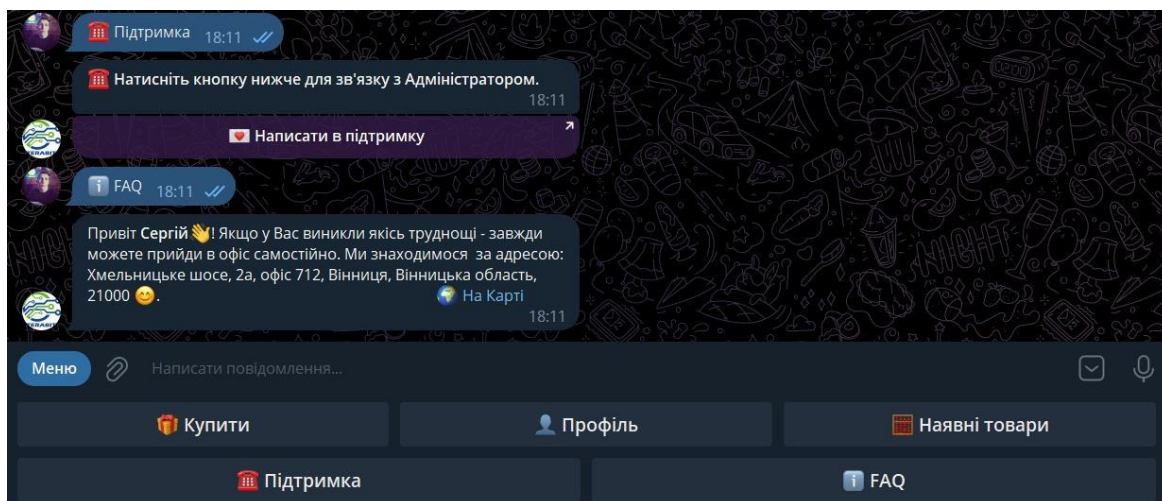


Рисунок 3.17 – Використання кнопок «Підтримка» та «FAQ»

В результаті нам надали контакт адміністратора, який зможе нам допомогти в складних ситуаціях. Також в FAQ є коротка інформація, яка може стати у нагоді коли вимикач «Покупки» буде вимкнтий і можливості перейти до товарів та категорій буде вимкнута. Або ж комусь комфортніше буде можливість візуально побачити товар в живу відвідавши магазин-майстерню самостійно.

Далі переглянемо профіль (рисунок 3.18). Використаємо кнопку «Профіль»:

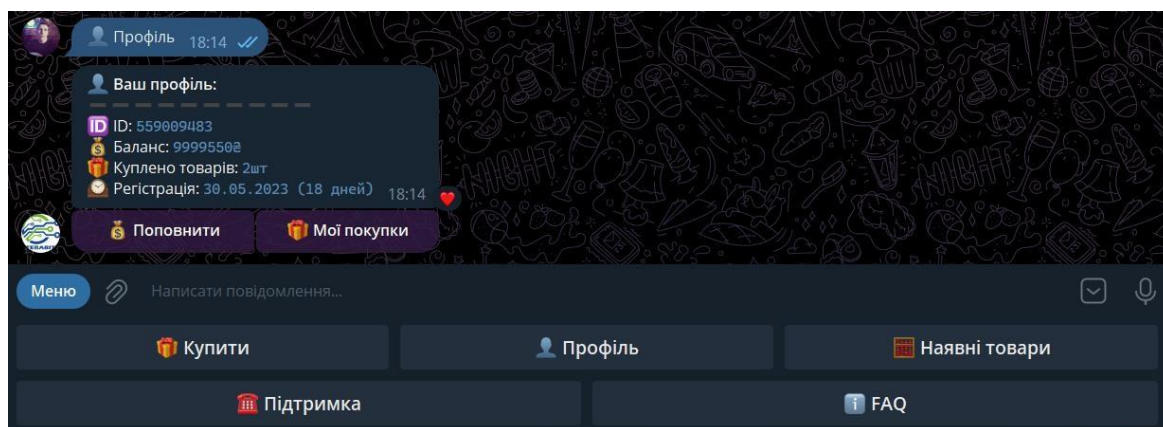


Рисунок 3.18 – Вигляд профіля від імені користувача, використавши кнопку «Профіль»

Відкрився профіль у якому ми можемо побачити поточний баланс, кількість куплених товарів, id та дату реєстрації. Також є можливість поповнити баланс та переглянути «Мої покупки» (рис. 3.19).

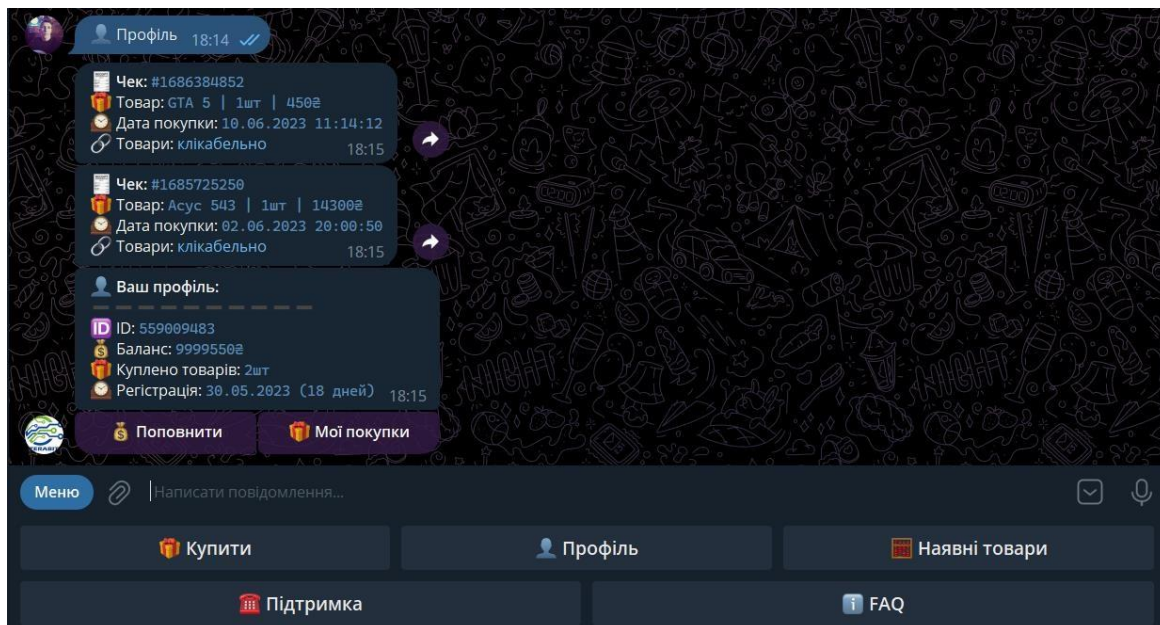


Рисунок 3.19 – Перегляд «Мої покупки» в меню «Профіль»

Результат виконання кнопки «наявні товари» наведено на рисунку 3.20.

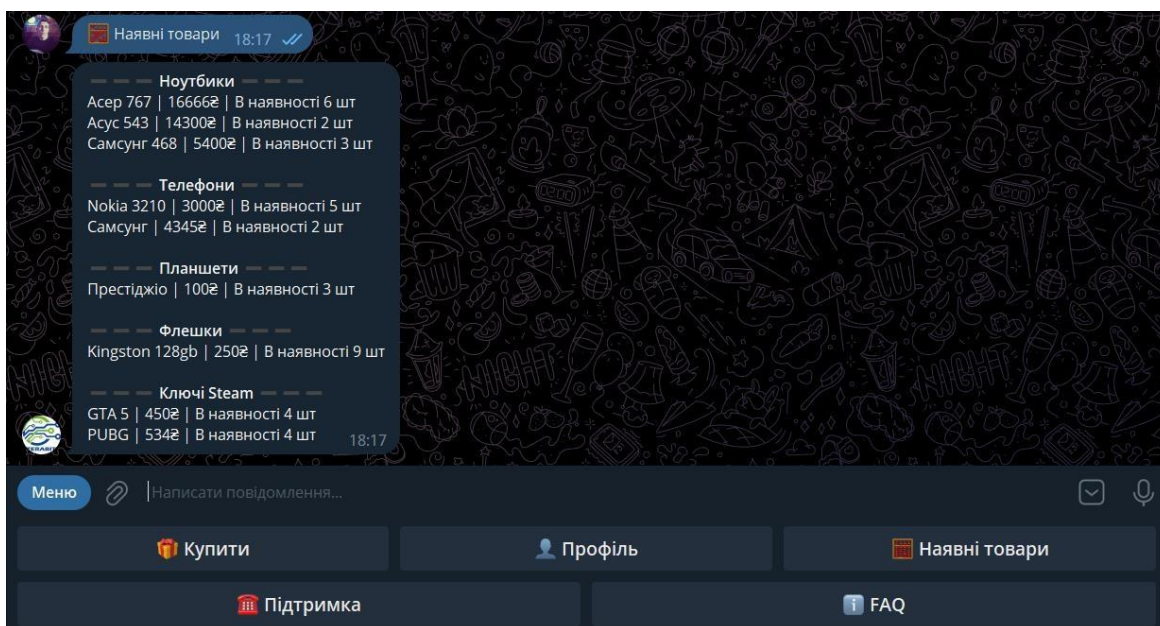


Рисунок 3.20 – Пункт меню «Наявні товари»

Використавши даний пункт ми отримуємо список усіх наявних товарів, які можна придбати, даний список ігнорує товари, які відсутні.

Спробуємо переглянути та придбати товар (рис. 3.21):

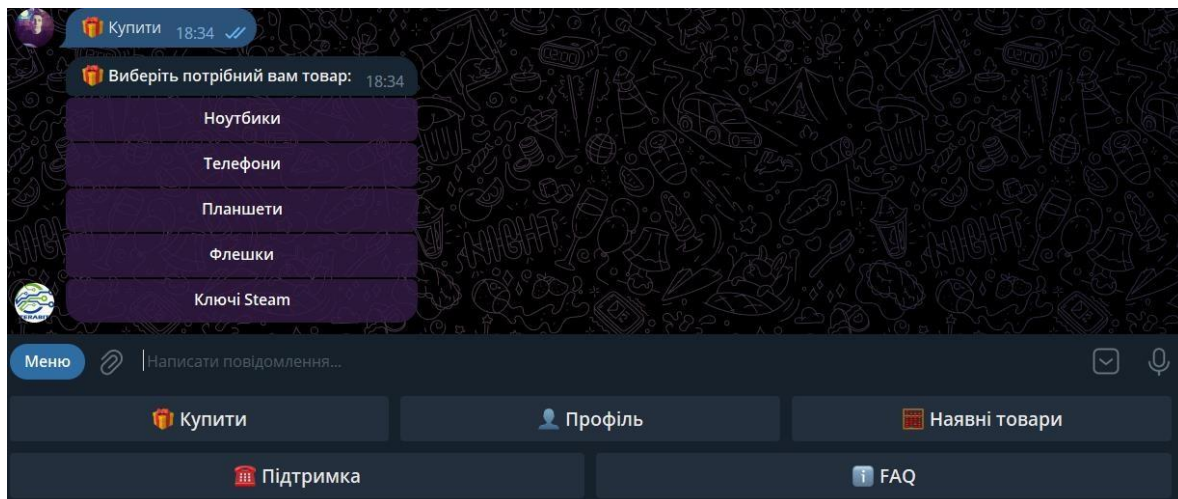


Рисунок 3.21 – Вибір категорії товару

Спробуємо придбати ключ GTA 5 (рис 3.22).

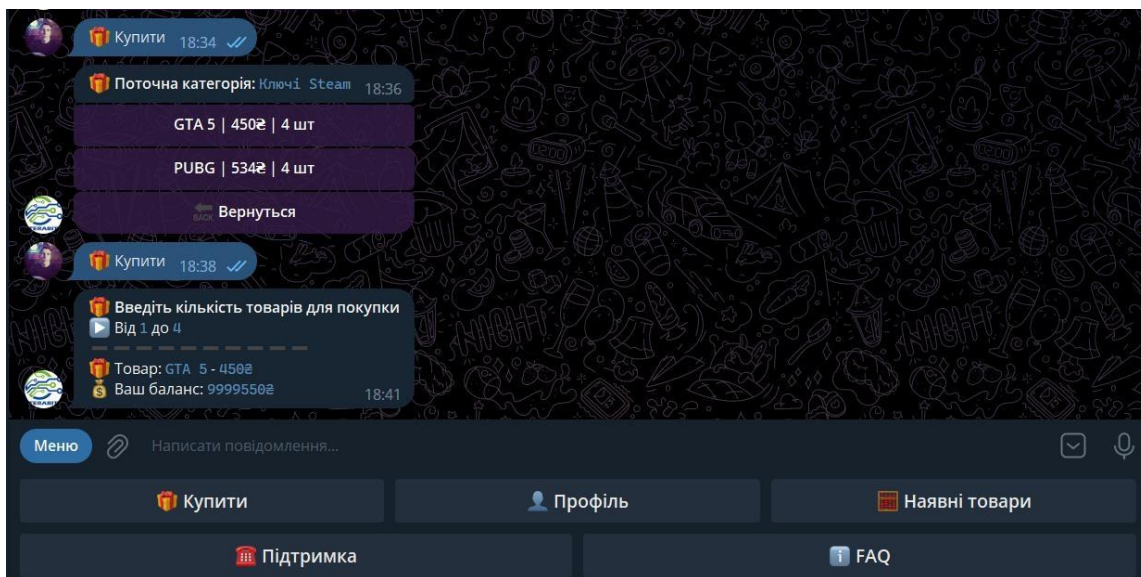


Рисунок 3.22 – Вибір товару обраної категорії

Введемо значення, яке не задовільняє умову і перевіримо, що станеться (рис. 3.23)

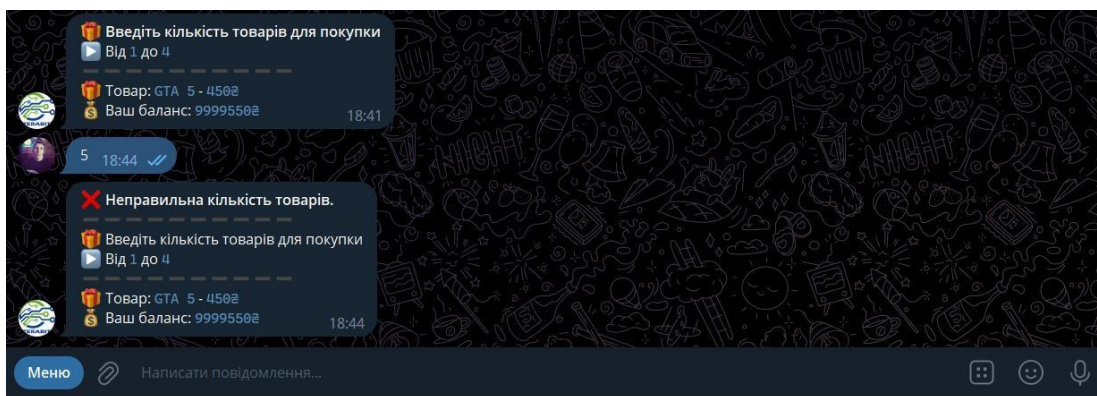


Рисунок 3.23 – Спроба придбати товар в більшість кількості від наявних

Отримали повідомлення, яке попереджує, що ми ввели невірну кількість товарів. Повторимо спробу використавши вірне значення (рис. 3.24, 3.25).

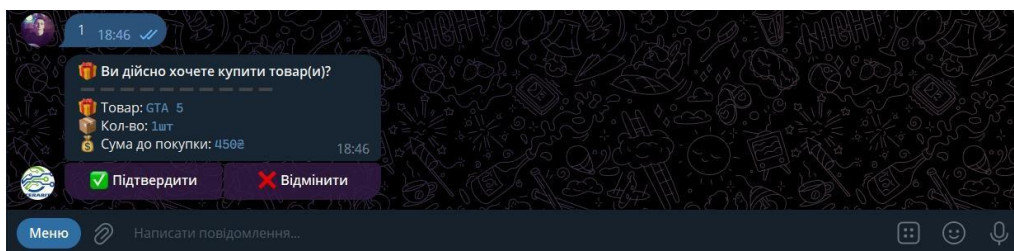


Рисунок 3.24 – Вікно підтвердження покупки

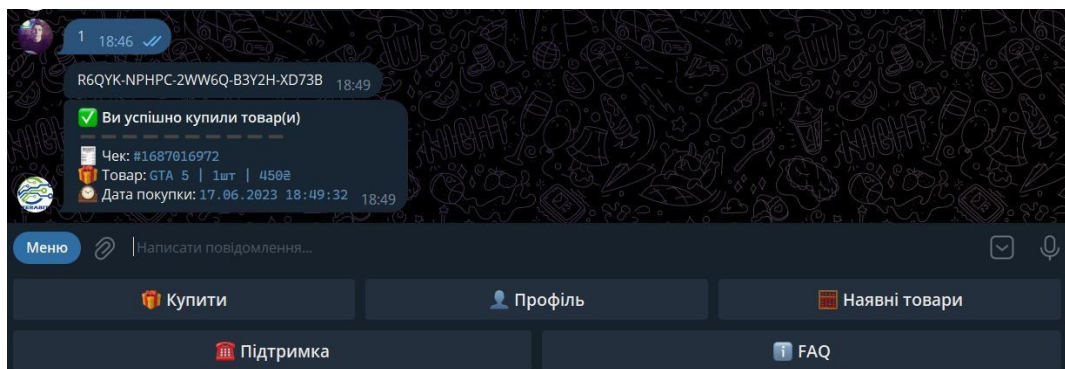


Рисунок 3.25 – Вдала покупка ключа Steam «GTA 5»

Після підтвердження покупки, ми отримали сам ключ, та чек. В чекові зазначено його номер, найменування товару, кількість та час покупки.

Після зазначення свого телеграм-id в файлі конфігурацій бота та його перезапускові ми отримуємо повідомлення про успішний запуск телеграм-бота та меню з додатковими кнопками інтерфейсу порівняно з звичайним користувачем.

Використаємо меню (рис. 3.26) та отримаємо файл бази даних та логів:

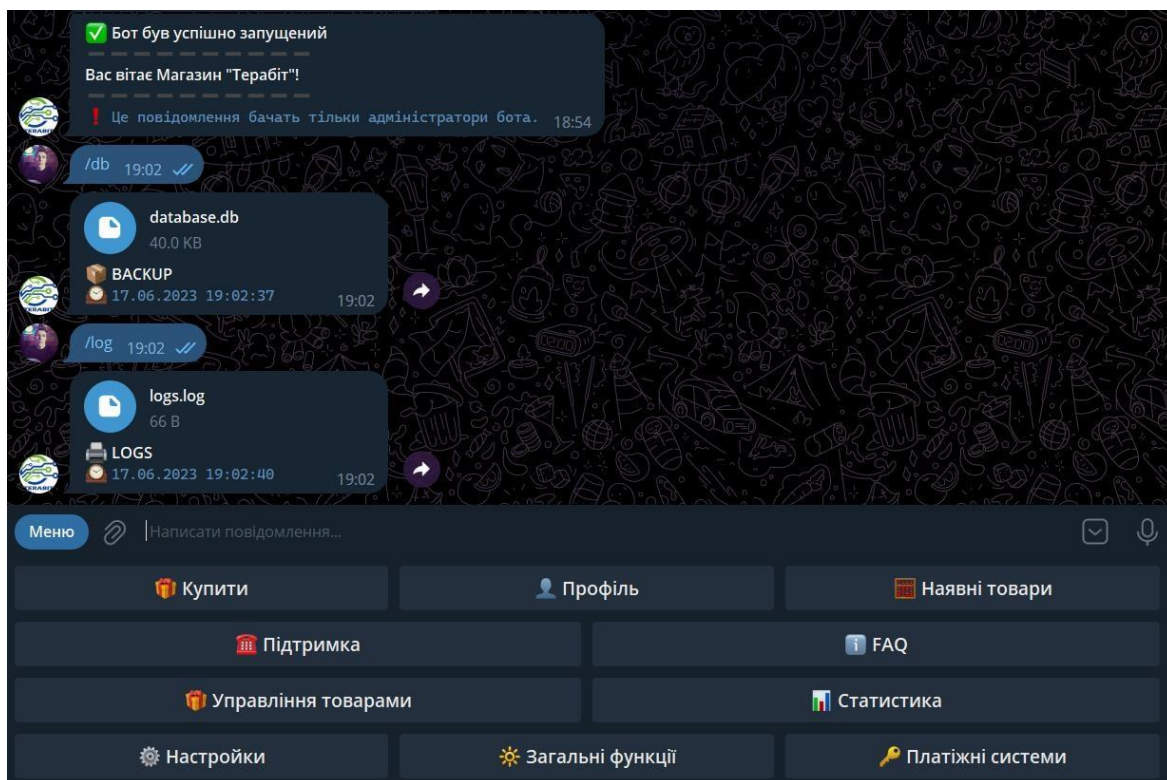


Рисунок 3.26 – Виконання команд «/db» та «/log» з статусом адміністратора

Отримали два файли, які можливо завантажити та переглянути використавши відповідні редактори.

Використаємо меню «Управління товарами» та створемо нову позицію. Меню редагування товарів зображено на рисунку 3.27

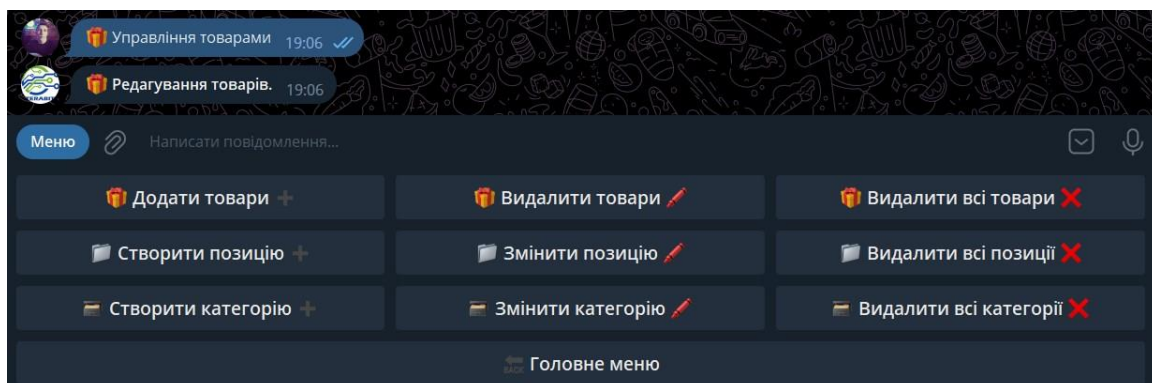


Рисунок 3.27 – Меню редагування товарів

Є два варіанти: пункт наявна категорія або натиснемо на кнопку «створити категорію» в нижній частині інтерфейсу. Виберемо наявну категорію Флешки (рис 3.28).

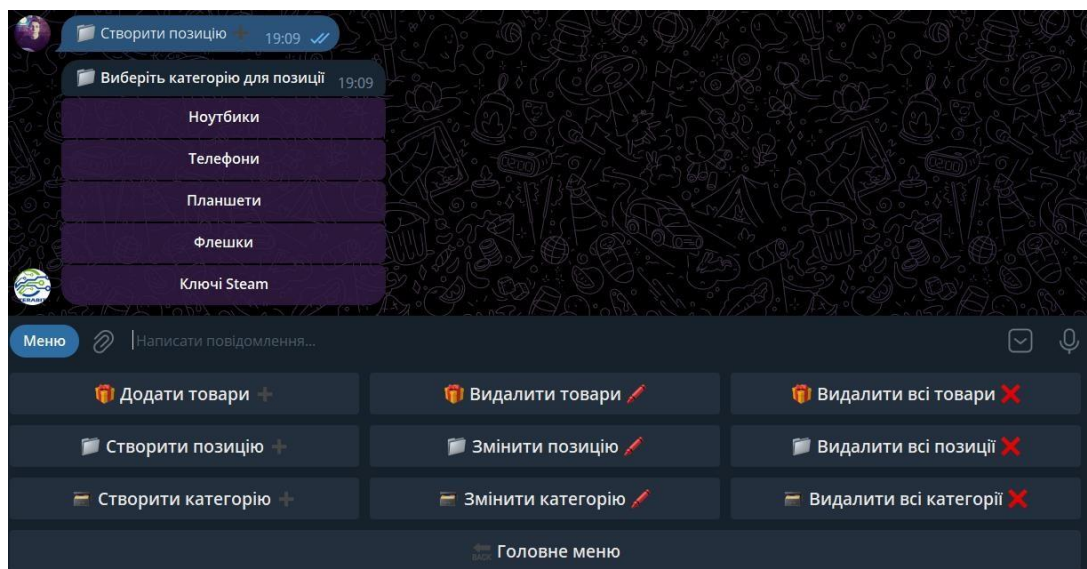


Рисунок 3.28 – Вибір категорії створюваної позиції

Вводимо дані, які від нас вимагає чат-бот (рис. 3.29).

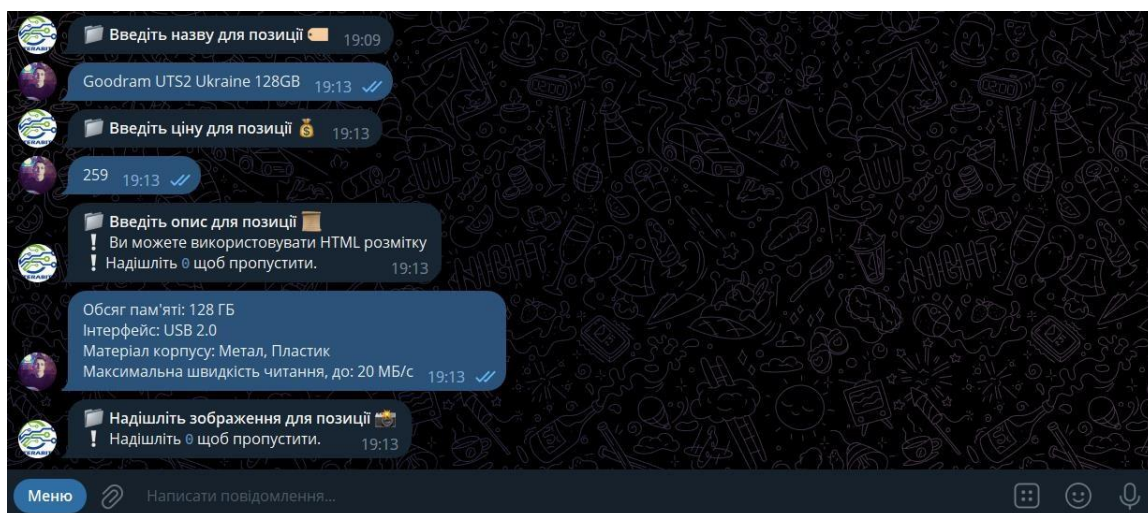


Рисунок 3.29 – Заповнення необхідних даних позиції

Готова позицію зображена на рисунку 3.30.



Рисунок 3.30 – Вигляд створеної позиції після заповнення усіх необхідних даних

Після заповнення потрібних даних ми отримали готову позицію та можемо відразу виконати над нею додаткові дії:

- змінити назву;
- змінити ціну;
- змінити опис;
- змінити фото;
- видалити усі товари позиції;
- додати товари;
- переглянути наявні товари за позицією;
- видалити позицію та усі товари які її використовували.

На рисунку 3.31 наведено описане вище меню.

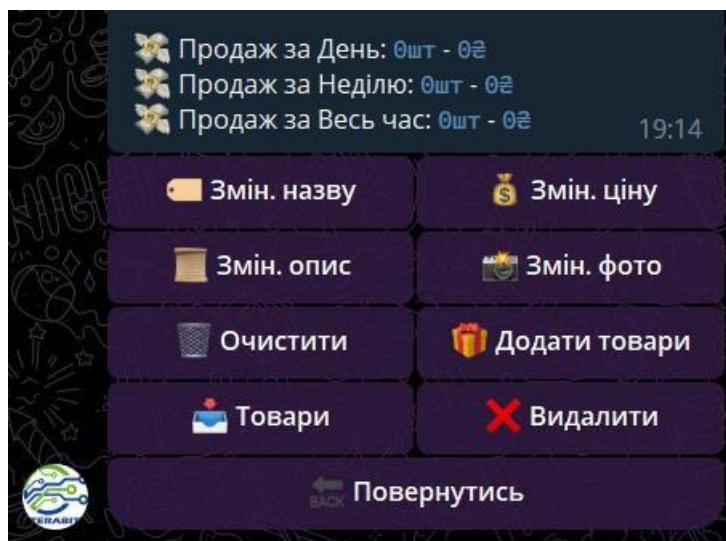


Рисунок 3.31 – Меню редагування створеної позиції

Перейшовши в «Налаштування» нас зустрічає два налаштування, а саме можливість змінити контакти, які зазначені в «Підтримка» та текст «FAQ», а також керування вимикачами телеграм-бота (рисунок 3.32).

Наступним кроком ми переглянемо статистику телеграм-бота для продажу товарів (рисунок 3.33).

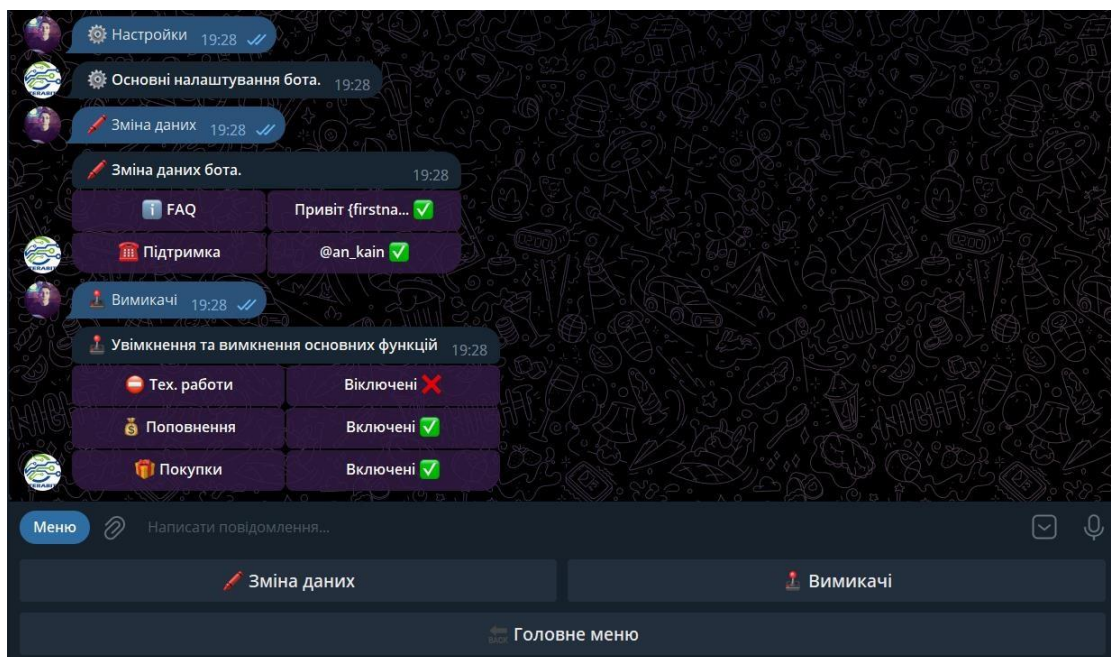


Рисунок 3.32 - Зміна налаштувань телеграм-бота для продажу товарів

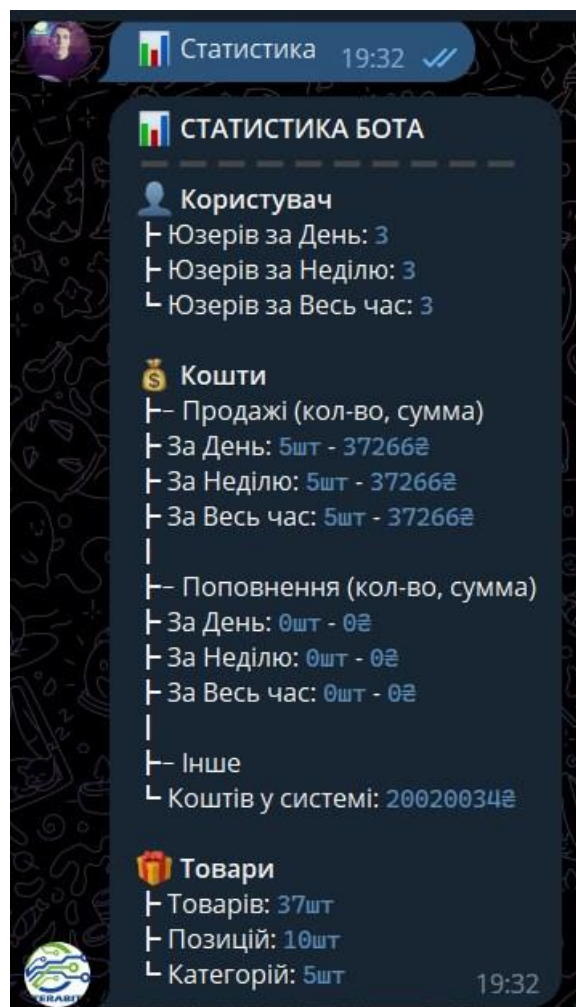


Рисунок 3.33 – Статистика телеграм-бота

Порівняємо розроблений чат-бот з раніше розглянутими ботами для продажу товарів. Порівняння наведено у таблиці 3.7.

Таблиця 3.7 – Порівняння чат-ботів для продажу товарів

Назва програми	Зміна налаштувань бота	Редагування замовлення	відображення статистики для адміністратора	Онлайн редагування товарів	виведення опису замовлених товарів
Чат-бот «Justin»	+	+	-	-	-
Чат-бот «Electronic Store»	+	+	-	-	+
Розроблений чат-бот	+	+	+	+	+

Таким чином в результаті аналізу роботи створеного чат-боту та відповідно

до порівняльної таблиці, можна зробити висновок про те що було розширено функціонал: додано функцію відображення статистики для адміністратора, та онлайн редагування товарів, що свідчить про те що мету дослідження було досягнуто.

3.9 Висновок

Обґрунтовано вибір мови програмування. В результаті було обрано Python як основну мову програмування для реалізації телеграм-бота. Обґрунтовано доцільність застосування ключових бібліотек та технологій, таких як aiogram, APScheduler, colorlog, beautifulsoup4, aiohttp, typing, colorama, bs4, pytz.

Microsoft VS Code було обрано як платформу для розробки телеграм-бота, оскільки вона кросплатформна, має досить зручний та багатифункціональний інтерфейс.

Здійснено аналіз процесу створення оболонки телеграм-бота з використанням «BotFather».

Обґрунтовано вибір СУБД для телеграм-бота з продажу товарів. Після порівняння різних СУБД було доведено, що SQLite3 є доцільним варіантом для телеграм-бота.

Реалізація телеграм-бота передбачає розробку структури бота та бази даних, реалізацію основних функцій, обробку повідомлень користувачів.

Здійснено тестування роботи телеграм-бота де перевірено правильність його роботи, включаючи функціональне тестування та взаємодію з користувачем.

ВИСНОВКИ

У ході виконання бакалаврської дипломної роботи було розроблено чат-бота для продажу товарів.

У ході аналізу предметної області розглянуто необхідність використання телеграм-ботів, як необхідного процесу автоматизації у сучасному світі.

Проведено докладний аналіз процесу створення оболонки чат-бота з використанням «BotFather».

Розроблено UML-діаграми варіантів використання телеграм-бота для користувача та адміністратора. Розроблено загальний алгоритм роботи телеграм-бота для продажу товарів. Розроблено алгоритм функціонування головного меню телеграм-бота для продажу товарів, що включає послідовність кроків та операцій, необхідних для забезпечення зручної та ефективної роботи користувача.

В результаті аналізу роботи створеного чат-боту та відповідно до порівняльної таблиці, можна зробити висновок про те що було розширено функціонал: додано функцію відображення статистики для адміністратора, та онлайн редагування товарів, що свідчить про те що мету дослідження було досягнуто.

Отже всі поставлені задачі виконано, мету роботи досягнуто.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Месенджер — що це таке, як встановити і як користуватися [Електронний ресурс] – Режим доступу до ресурсу: <http://radka.in.ua/poradi/mesendjer-sho-ce-take-iak-vstanovit.html>.
2. Telegram для бізнесу: як розкрутити інтернет магазин в месенджері з нуля [Електронний ресурс] – Режим доступу до ресурсу: <https://torgsoft.ua/articles/stati/sozdat-biznes-v-telegram/>.
3. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>.
4. Дев'ять платформ для створення чат-бот магазину в Telegram [Електронний ресурс] – Режим доступу до ресурсу: <https://vc.ua/services/182007devyat-platform-dlya-sozdaniya-chat-bot-magazina-v-telegram>.
5. Все про чат-боти: типи і приклади, якому бізнесу підійде, список конструкторів для створення [Електронний ресурс] – Режим доступу до ресурсу: <https://web-promo.ua/ua/blog/vse-o-chat-botah-tipy-i-primery-kakomu-biznesupodojdet-spisok-konstruktorov-dlya-sozdaniya>.
6. Трофименко О. Г. Сфери застосування чат ботів [Електронний ресурс] / О. Г. Трофименко – Режим доступу до ресурсу: <http://dspace.onua.edu.ua/bitstream/handle/11300/18203/Трофименко%20О.%20Г.%20Сфери%20застосування%20чат-ботів.pdf?sequence=1&isAllowed=y>.
7. Словник frequently asked questions [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.glosbe.com/en/uk/frequently%20asked%20questions>
8. Chatfuel Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.chatfuel.com/>.
9. ManyChat Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://manychat.com/>.
10. BotPress Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://botpress.com/docs/>.

11. Martin F. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd edition / Fowler Martin., 2018.
12. Основи UML - діаграми використання (use-case) [Електронний ресурс] – Режим доступу до ресурсу: <https://pro-prof.com/archives/2594>.
13. python-telegram-bot [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python-telegram-bot.org/en/stable/>.
14. Features Of SQLite [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sqlite.org/features.html>.
15. Raj S. Building Chatbots with Python: Using Natural Language Processing and Machine Learning. Apress, 2018. 211 p.
16. Learn to build your first bot in Telegram with Python n [Електронний ресурс] - Режим доступу: <https://medium.freecodecamp.org/learn-to-buildyourfirstbot-in-telegram-with-python-4c99526765e4>.
17. Створюємо Telegram бота на Python. Частина 1[Електронний ресурс] - Режим доступу: <https://codeguida.com/post/410>.
18. Visual Studio Code - Code Editing. Redefined. URL: <https://code.visualstudio.com/>.
19. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. – Вінниця : ВНТУ, 2023. – (PDF, 54 с.).

ДОДАТОК А
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка чат-боту для продажу товарів

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

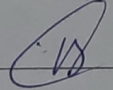
Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність 82,1% Схожість 17,9%

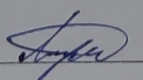
Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Озеранський В.С.

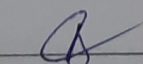
Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи



Лебедєв Д.І.

Керівник роботи



Озеранський В.С.

ДОДАТОК Б

Інструкція користувача

На початку роботи нас зустрічає інтерфейс з єдиною кнопкою «РОЗПОЧАТИ».

Після її натискання боту автоматично відправляється команда «/start». В свою чергу відображається привітальне повідомлення, меню (рис. Б.1) та кнопки інтерфейсу для користування розробленим чат-ботом.

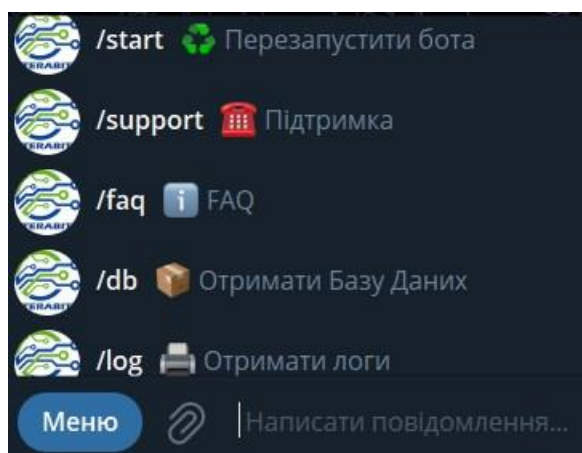


Рисунок Б.1 – Меню чат-бота для продажу товарів

Спроба отримати файли логів або бази даних не маючи статусу адміністратор зображено на рисунку Б.2.

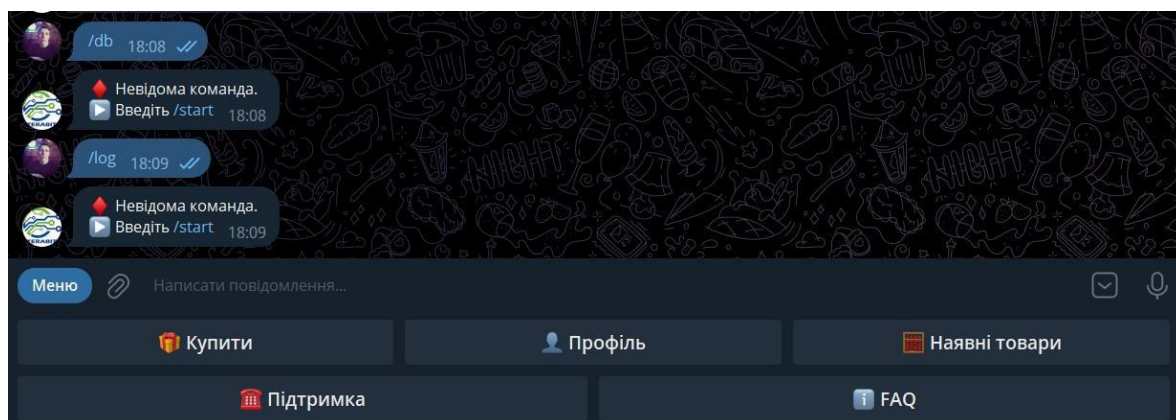


Рисунок Б.2 – Спроба отримати базу даних/лог не маючи статусу адміністратор

Оскільки у нас немає прав адміністратора, ми отримали повідомлення «невідома команда».

Використаємо можливості, що надають кнопки «Підтримка» та «FAQ» (рис. Б.3):

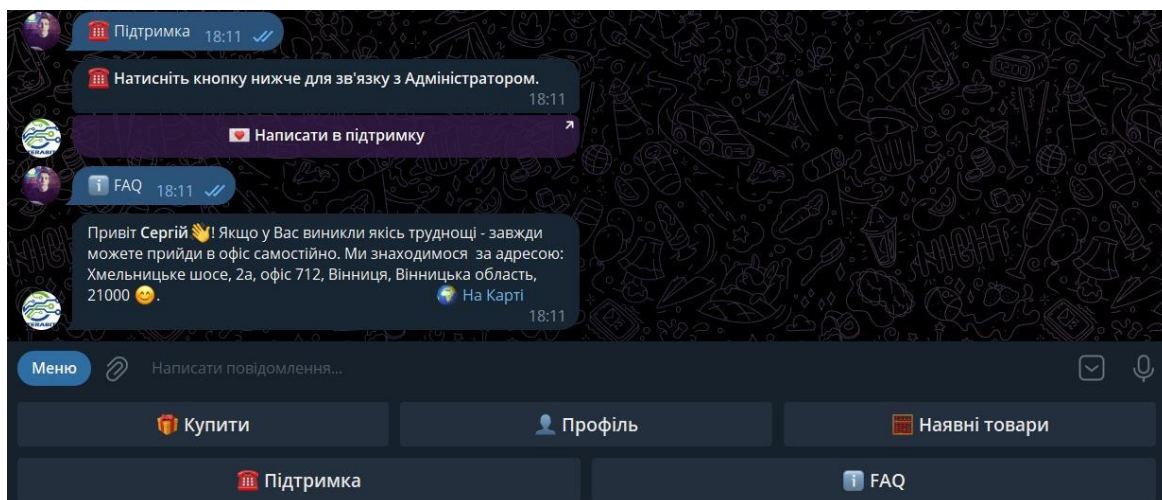


Рисунок Б.3 – Використання кнопок «Підтримка» та «FAQ»

В результаті нам надали контакт адміністратора, який зможе нам допомогти в складних ситуаціях. Також в FAQ є коротка інформація, яка може стати у нагоді коли вимикач «Покупки» буде вимкнтий і можливості перейти до товарів та категорій буде вимкнута. Або ж комусь комфортніше буде можливість візуально побачити товар в живу відвідавши магазин-майстерню самостійно.

Далі переглянемо профіль (рисунок Б.4). Використаємо кнопку «Профіль»:

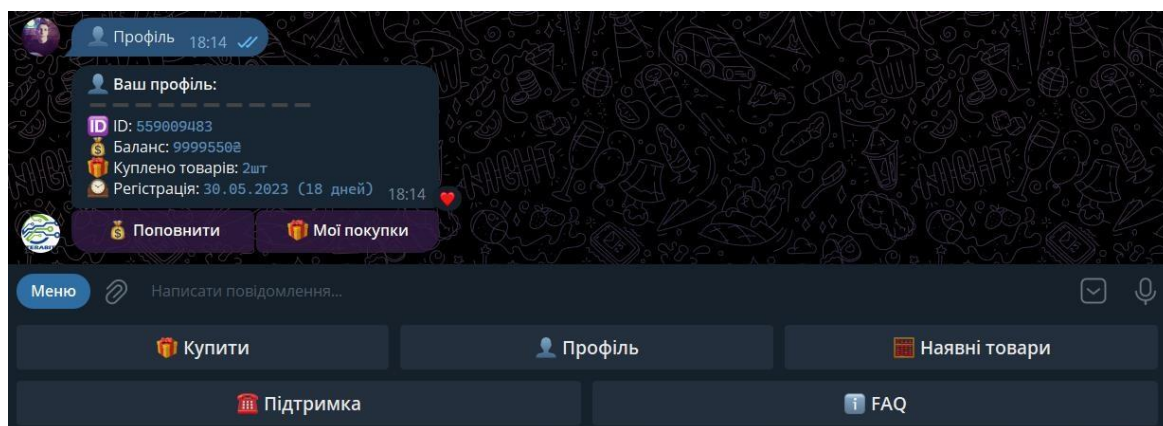


Рисунок Б.4 – Вигляд профіля від імені користувача, використавши кнопку «Профіль»

ДОДАТОК В

Лістинг програми

Лістинг коду файла main.py

```
# - *- coding: utf- 8 - *-
import os import sys

import colorama
from aiogram import executor, Dispatcher
    from tgbot.data.config import get_admins
from tgbot.data.loader import scheduler
from tgbot.handlers import dp
from tgbot.middlewares import setup_middlewares from
tgbot.services.api_session import AsyncSession from
tgbot.services.api_sqlite import create_dbx from
tgbot.utils.misc.bot_commands import set_commands from
tgbot.utils.misc.bot_filters import IsPrivate from
tgbot.utils.misc.bot_logging import bot_logger
from tgbot.utils.misc_functions import (check_bot_data, startup_notify,
update_profit_day,
                                update_profit_week,
autobackup_admin)

colorama.init()

# Запуск планувальника async def
scheduler_start(rSession):
    scheduler.add_job(update_profit_day, trigger="cron", hour=00)
        scheduler.add_job(update_profit_week,            trigger="cron",
day_of_week="mon", hour=00, minute=1)
        scheduler.add_job(autobackup_admin, trigger="cron", hour=00)

# Виконання функції після запуску бота async
def on_startup(dp: Dispatcher):
    rSession = AsyncSession()
    dp.bot['rSession'] = rSession
        await dp.bot.delete_webhook()
await dp.bot.get_updates(offset=-1)
        await set_commands(dp)        await
check_bot_data()        await
scheduler_start(rSession)        await
startup_notify(dp, rSession)

        bot_logger.warning("BOT                                WAS                                STARTED")
print(colorama.Fore.LIGHTYELLOW_EX + f"~~~~~ Bot was started - @{(await
dp.bot.get_me()).username} ~~~~~")        print(colorama.Fore.RESET)

        if len(get_admins()) == 0: print("***** ENTER ADMIN ID IN settings.ini
*****")
```

```

# Виконання функції після вимкнення бота async
def on_shutdown(dp: Dispatcher):      rSession:
AsyncSession = dp.bot['rSession']      await
rSession.close()
    await dp.storage.close()
await dp.storage.wait_closed()
    await (await dp.bot.get_session()).close()
    if sys.platform.startswith("win"):
        os.system("cls")
else:
    os.system("clear")
if __name__ == "__main__":
    create_dbx()

    scheduler.start()
    dp.filters_factory.bind(IsPrivate) # Підключення фільтра приватності
setup_middlewares(dp) # Підключення міddlварей

    executor.start_polling(dp, on_startup=on_startup,
on_shutdown=on_shutdown)

```

Лістинг коду файла settings.ini

```

# - *- coding: utf- 8 - *-
[settings]
token=5916978492:AAff_UQha6_Ggl10D3n-86MB7dPVh3AwM30 admin_id=559009483

```

Лістинг коду файла bot_filters.py

```

# - *- coding: utf- 8 - *- from
aiogram import types
from aiogram.dispatcher.filters import BoundFilter

from tgbot.data.config import get_admins
from tgbot.services.api_sqlite import get_settingsx

# Перевірте діалог в ЛС бота class
IsPrivate(BoundFilter):
    async def check(self, message):
        if "id" in message:
            return message.message.chat.type == types.ChatType.PRIVATE
else:
    return message.chat.type == types.ChatType.PRIVATE

```

```

# Перевірка на адміністратора class
IsAdmin(BoundFilter):      async def check(self,
message: types.Message):      if
message.from_user.id in get_admins():
        return True
else:
        return False

# Перевірка на технічні роботи class
IsWork(BoundFilter):      async def check(self,
message: types.Message):      get_settings =
get_settingsx()
        if get_settings['status_work'] == "False" or message.from_user.id
in get_admins():
        return False
else:
        return True

# Перевірка на можливість поповнення class
IsRefill(BoundFilter):      async def check(self,
message: types.Message):
        get_settings = get_settingsx()
        if get_settings['status_refill'] == "True" or
message.from_user.id in get_admins():
        return False
else:
        return True

# Перевірка на можливість купівлі товару class
IsBuy(BoundFilter):      async def check(self,
message: types.Message):
        get_settings = get_settingsx()
        if get_settings['status_buy'] == "True" or message.from_user.id
in get_admins():
        return False
else:
        return
True

```

Лістинг коду файла reply_main.py

```

# - *- coding: utf- 8 - *-
from aiogram.types import ReplyKeyboardMarkup

from tgbot.data.config import get_admins

# Кнопки главного меню def
menu_frep(user_id):
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.row("🛒 Купити", "👤 Профіль", "📦 Наявні товари")
    keyboard.row("🗨 Підтримка", "❓ FAQ")
    if user_id in get_admins():

```

```

        keyboard.row("🛒 Управління товарами", "📊 Статистика")
    keyboard.row("⚙️ Настройки", "☀️ Загальні функції", "🔑 Платіжні системи")
    return keyboard

# Кнопки платіжних систем def
payments_frep():
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)

    keyboard.row("Виберіть спосіб поповнення")
    keyboard.row("⬅️ Головне меню")

    return keyboard

# Кнопки загальних функцій def
functions_frep(user_id):
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.row("👤 Пошук профіля 🔍", "🔊 Розсилка", "📄 Пошук чеків 🔍")
    keyboard.row("⬅️ Головне меню")

    return keyboard

# Кнопки налаштувань def
settings_frep():
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.row("✎ Зміна даних", "🔇 Вимикачі")
    keyboard.row("⬅️ Головне меню")

    return keyboard

# Кнопки зміни товарів def
items_frep():
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)
    keyboard.row("🛒 Додати товари +", "🛒 Видалити товари ✎", "🛒 Видалити всі товари ✕")
    keyboard.row("📦 Створити позицію +", "📦 Змінити позицію ✎", "📦 Видалити всі позиції ✕")
    keyboard.row("👉 Створити категорію +", "👉 Змінити категорію ✎", "👉 Видалити всі категорії ✕")
    keyboard.row("⬅️ Головне меню")

    return keyboard

# Завершення завантаження товарів
finish_load_rep = ReplyKeyboardMarkup(resize_keyboard=True)
finish_load_rep.row("🏁 Закінчити завантаження товарів")

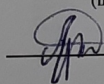
```

ДОДАТОК Г

ГРАФІЧНА ЧАСТИНА

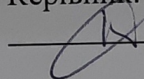
«РОЗРОБКА ЧАТ-БОТУ ДЛЯ ПРОДАЖУ ТОВАРІВ»

Виконав: студент 4 курсу, групи КН-22мс
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Лебедєв Д.І.
(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри КН



Озеранський В.С.
(прізвище та ініціали)

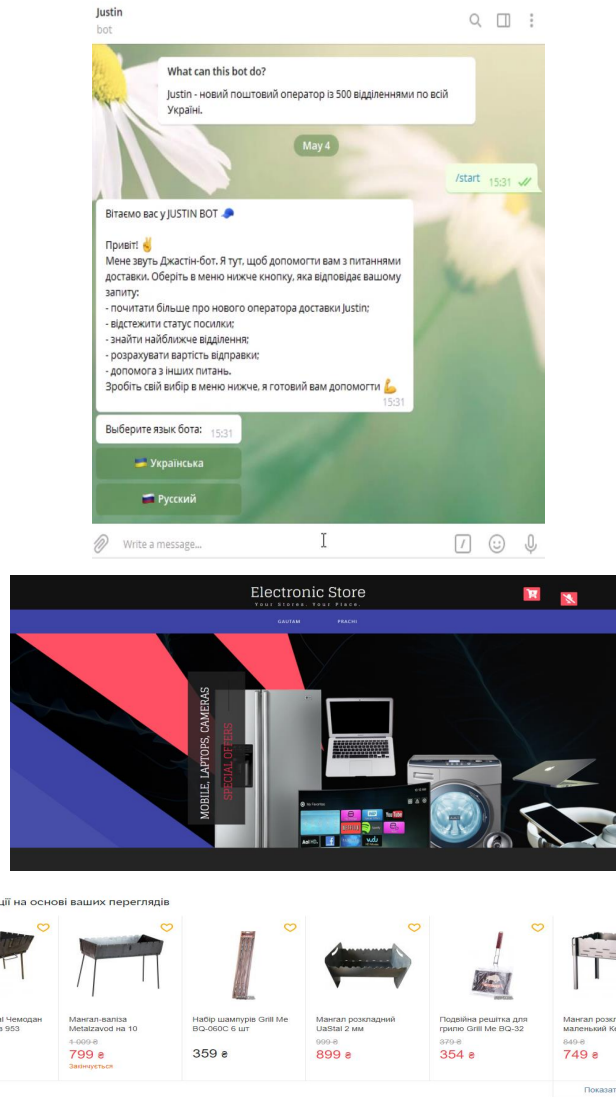


Рисунок Г.1 – Приклади існуючих чат-ботів для продажу товарів



Рисунок Г.2 – UML-діаграма варіантів використання чат-бота для продажу товарів від імені адміністратора

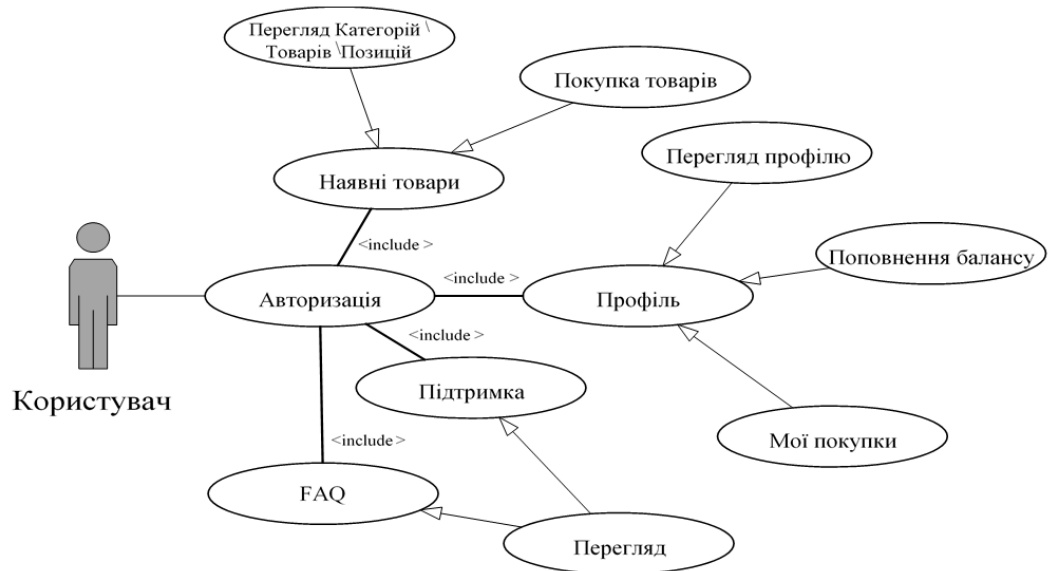


Рисунок Г.3 – UML-діаграма варіантів використання чат-бота для продажу товарів від імені користувача

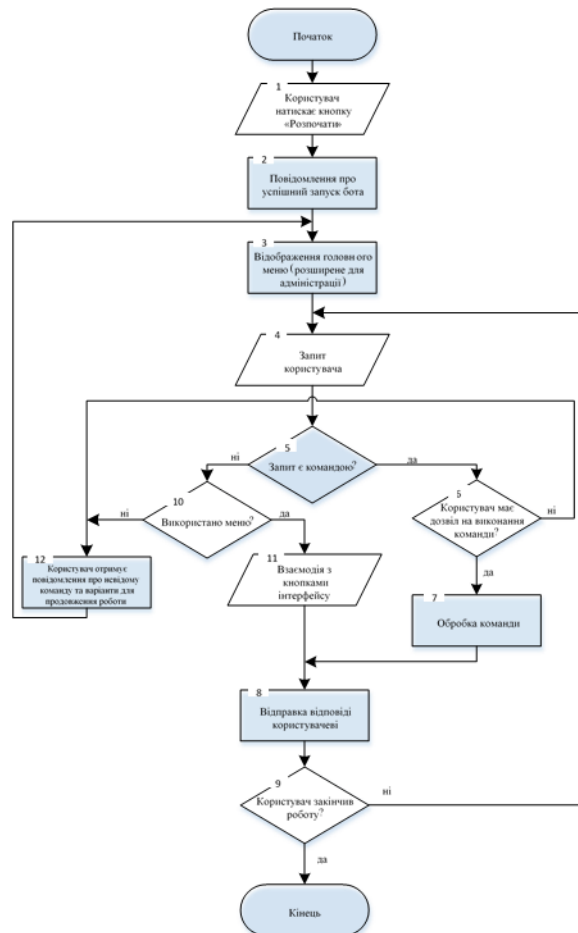


Рисунок Г.4 – Схема алгоритму роботи чат-бота для продажу товарів

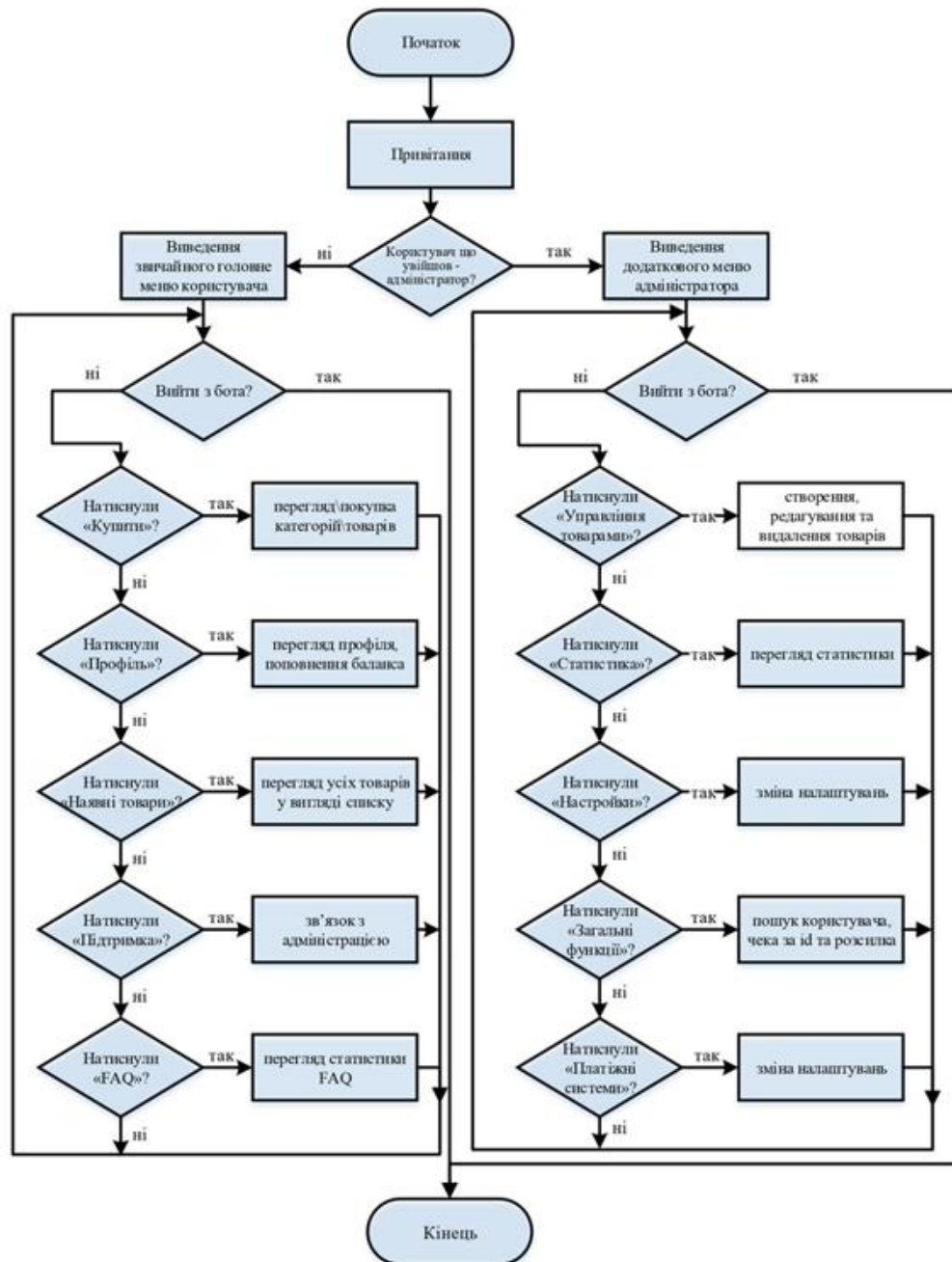


Рисунок Г.5 – Схема алгоритму роботи головного меню чат-бота для продажу товарів

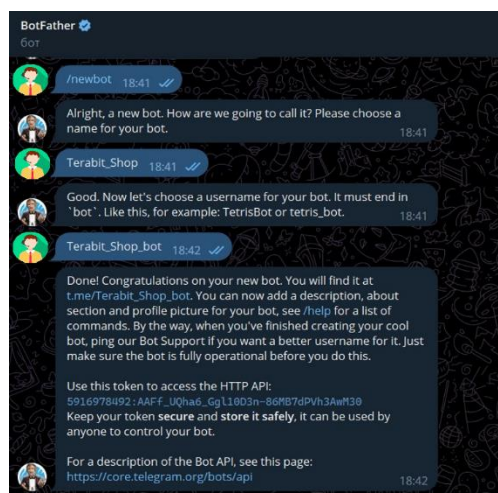
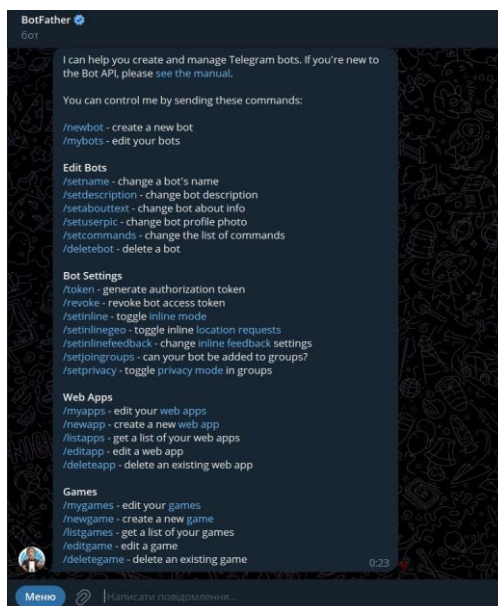


Рисунок Г.6 – Процес створення чат-бота за допомогою «BotFather»

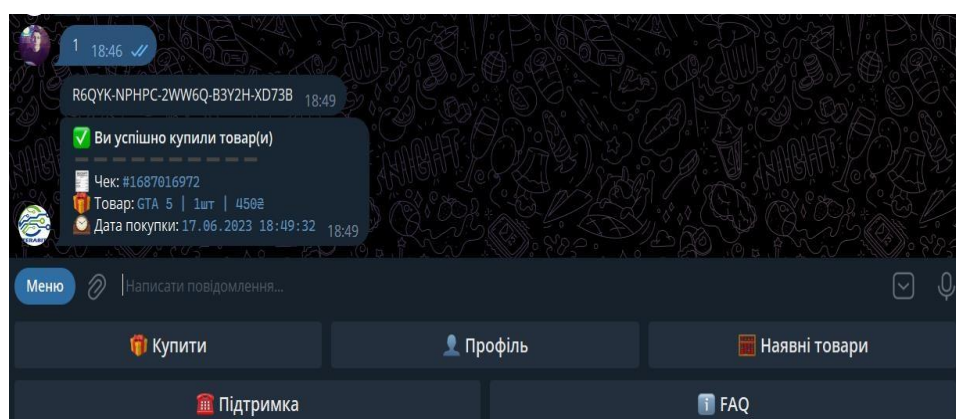
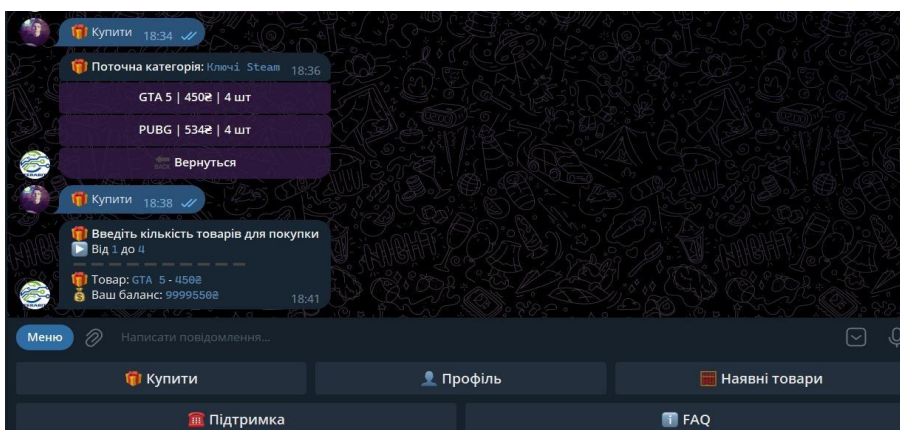
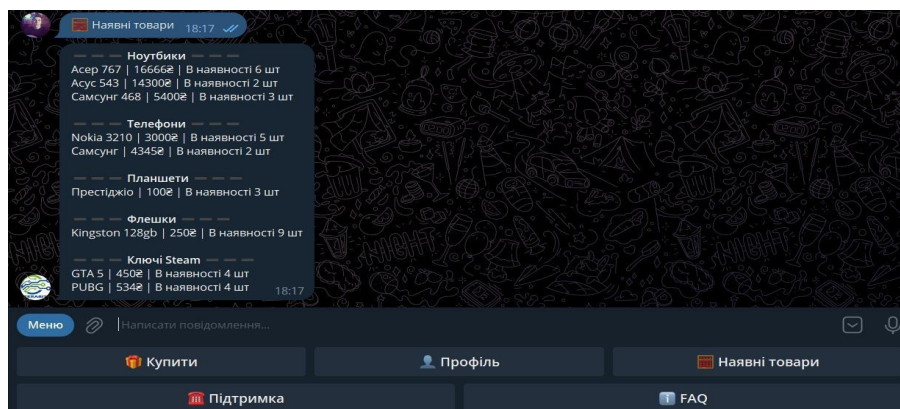


Рисунок Г.7 – Інтерфейс розробленого чат-боту в месенджері Telegram

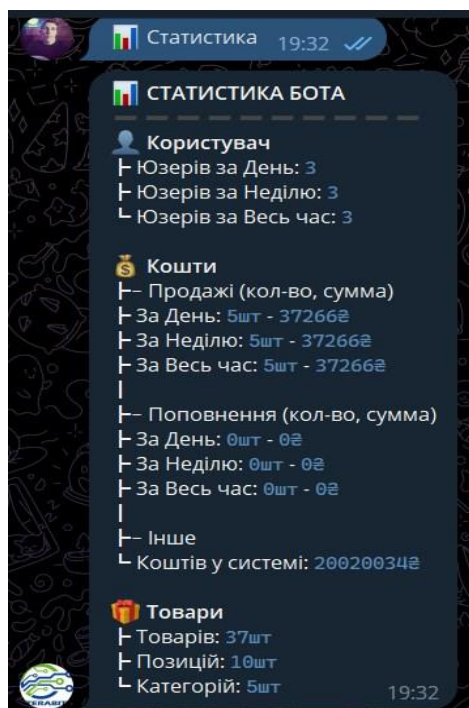


Рисунок Г.8 – Інтерфейс розробленого чат-боту в месенджері Telegram