

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

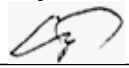
Кафедра програмного забезпечення автоматизованих систем

Спеціальність F2 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

 проф. Приходько С.Б.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: **«Розробка програмного забезпечення телеграм-боту для реклами
продукції компанії АТБ»**

Виконав: студент групи 4151ст

Співак Роман Віталійович

(підпис, ПІБ)



Керівник роботи:

Доцент, кандидат технічних наук

Суслов Сергій Віталійович

(посада, науковий ступень вчене звання)

(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»



«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____ доц. Макарова Л.М.
(підпис)
« 08 » 04 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Співаку Роману Віталійовичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення телеграм-боту для реклами продукції компанії АТБ»

Керівник роботи Суслов Сергій Віталійович

Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проект програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів: Вступ; Аналіз практичної задачі; Аналіз аналогів програмного забезпечення; Постанова задачі; Функції веб-застосунку; Вхідні дані; Вихідні дані; Ескізний проект. Діаграма варіантів використання веб- застосунку; Ескізний проект. Концептуальна модель; Технічний проект. Логічна модель; Технічний проект. Діаграма класів; Робочий проект. Фізична модель; Результат розробки; Висновки.

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

2. Дата видачі завдання 08.04.2025р.

Номер	Назва етапів роботи	Термін виконання	Примітка
1	Підготовка розділу ВСТУП	31.03.2025	ПП*
2	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3	Аналіз вимог до ПЗ	14.04.2025	ПП*
4	Розробка архітектури ПЗ	21.04.2025	
5	Розробка проекту ПЗ	05.05.2025	
6	Реалізація та тестування ПЗ	12.05.2025	
7	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8	Підготовка розділу з охорони праці	19.05.2025	ПП*
9	Підготовка розділу ВИСНОВКИ	23.05.2025	
10	Оформлення списку використаних джерел та додатків	26.05.2025	
11	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12	Підготовка презентації та доповіді	06.06.2025	
13	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 05.02.2025 до 24.03.2025 р.

Студент


 (підпис)

Співак Р.В.

(ПІБ)

Керівник роботи

(підпис)

Суслов С.В.

(ПІБ)

АНОТАЦІЯ

У даній кваліфікаційній роботі представлено розробку Telegram боту для реклами продукції компанії.

В кваліфікаційній роботі виконано аналіз предметної області, ескізний, технічний та робочий проєкти, також розділи з охорони праці та техніко-економічне обґрунтування.

Бот розроблений за допомогою мови програмування Python, середовища розробки PyCharm, СУБД MySQL, засобів Telegram (BotFather, Telegram API)

Кваліфікаційну роботу виконано на 75 сторінках машинописного тексту і містить: 24 рисунка, 11 таблиць, список використаних джерел із 9 найменувань.

Робота викладена українською мовою.

ABSTRACT

This qualification work presents the development of Telegram bot for advertising products "ATB"

The qualification work includes an analysis of the subject area, sketch, technical and working projects, as well as sections on labor protection and feasibility study.

The bot is developed using the Python programming language, PyCharm development environment, MongoDB database, Telegram tools (BotFather, Telegram API)

The qualification work is done on 75 pages of typewritten text and contains: 24 figures, 11 tables, list of used literature of 9 titles.

The work is presented in Ukrainian language.

ЗМІСТ

1.АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ РЕКЛАМИ ПРОДУКЦІЇ КОМПАНІЇ АТБ.....	11
1.1 Аналіз практичної задачі інженерії програмного забезпечення	11
1.2 Аналіз існуючих аналогів телеграм-ботів для реклами продукції.....	11
1.3 Постановка задачі на розробку програмного забезпечення телеграм-боту для реклами продукції компанії АТБ.....	14
2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ РЕКЛАМИ ПРОДУКЦІЇ КОМПАНІЇ АТБ	17
2.1 Аналіз вимог до програмного забезпечення телеграм-боту для реклами продукції компанії АТБ і розробка моделі варіантів використання телеграм-боту для реклами продукції компанії АТБ.....	17
2.2 Розробка архітектури програмного забезпечення телеграм-боту для реклами продукції компанії АТБ	19
2.2.1 Обґрунтування вибору архітектурного підходу	19
2.2.2 Структура архітектури.....	20
2.2.3 Взаємодія між компонентами	21
2.2.4 Специфікація компонентів	22
2.2.5 Візуалізація архітектури програмного забезпечення	23
2.3 Проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ.....	24
2.3.1 Ескізний проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ	25
2.3.2 Технічний проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ	32
2.3.3 Робочий проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ	34

2.4 Результати розробки програмного забезпечення телеграм-боту для реклами продукції компанії АТБ	37
2.4.1 Загальний опис результатів	38
2.4.2 Реалізовані функції.....	38
2.4.3 Частини, що не були детально висвітлені	39
2.4.4 Кількісні та якісні характеристики	39
2.4.5 Результати випробувань	40
3 ОХОРОНА ПРАЦІ	45
3.1 Вступ.....	45
3.2 Санітарно-гігієнічні вимоги до умов праці	45
3.3 Електробезпека	47
3.4 Пожежна безпека.....	49
3.5 Безпека під час роботи з персональним комп'ютером.....	50
3.6 Розрахунки рівня шуму в приміщенні	51
3.7 Висновки	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	55
ДОДАТОК Б – Код розробки	58
ДОДАТОК В– ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	66
ДОДАТОК Г – Керівництво користувачеві.....	69
ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБОВУВАНЬ ПЗ.....	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

ПК – персональний комп'ютер;

ПЗ – програмне забезпечення;

СОМП – система обміну миттєвими повідомленнями;

Чат-бот – віртуальний менеджер, який з'ясовує потреби клієнта і допомагає їх задовольнити.

GitHub – веб-сервіс для спільної розробки та зберігання програмного забезпечення;

UML – уніфікована мова моделювання (Unified Modeling Language);

ВСТУП

У кваліфікаційній роботі описується процес створення чат-бота для мережі супермаркетів компанії, який функціонує в середовищі месенджера Telegram. З огляду на стрімкий розвиток цифрових технологій, особливо в галузі мобільного зв'язку та онлайн-комунікацій, системи обміну миттєвими повідомленнями (СОМП) стали невід'ємною частиною повсякденного життя мільйонів користувачів. Якщо раніше месенджери використовувалися переважно для особистого спілкування, то сьогодні вони виступають потужним багатофункціональним інструментом — як для отримання інформації, так і для автоматизації бізнес-процесів, зокрема у сфері торгівлі, обслуговування клієнтів і маркетингу.

Особливу роль у цій трансформації відіграють чат-боти — програми, що імітують спілкування з користувачем, автоматизуючи рутинні завдання. Вони допомагають компаніям зменшити навантаження на персонал, покращити взаємодію з клієнтами, оперативно інформувати про новини, акції та асортимент товарів, а також підвищувати рівень лояльності споживачів. На сьогодні існує широкий спектр застосування ботів: від простих інформативних систем до складних інтерактивних рішень із підтримкою штучного інтелекту та інтеграцією з базами даних, платіжними системами, CRM тощо.

Попри значне поширення ботів у світі, в Україні ця технологія лише починає активно розвиватися. Більшість компаній ще не встигли повною мірою інтегрувати такі рішення у свою бізнес-структуру, що створює додаткові можливості для новаторських рішень і водночас зумовлює високу актуальність розробки. Вивчення принципів створення ботів та розробка власного рішення на прикладі конкретного бізнес-кейсу (мережі супермаркетів) є важливим кроком на шляху до цифрової трансформації українського підприємництва.

Метою роботи є розробка телеграм-бота для реклами продукції компанії АТБ, який поєднує зручність кнопочового інтерфейсу, просту навігацію, доступ до інформації без необхідності переходу на зовнішні ресурси та мінімальний поріг входу для користувача, а також сприяє підвищенню ефективності маркетингової

діяльності компанії та покращенню взаємодії з клієнтами.

Основні задачі роботи:

- проаналізувати існуючі підходи до створення чат-ботів;
- обґрунтувати вибір технологій для реалізації Телеграм-бота;
- спроектувати архітектуру програмного рішення;
- реалізувати прототип чат-бота з урахуванням потреб бізнесу;
- оцінити ефективність запропонованого рішення та його потенціал до масштабування.

1.АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ РЕКЛАМИ ПРОДУКЦІЇ КОМПАНІЇ АТБ

1.1 Аналіз практичної задачі інженерії програмного забезпечення

У сучасних умовах роздрібна торгівля стикається з постійним підвищенням конкуренції та змінними вподобаннями споживачів. Одним із ключових інструментів оптимізації взаємодії з клієнтами є впровадження чат-ботів у месенджери. Особливо актуальним це стало для мережі дискаунтерів АТБ, яка має широку географію регіональних філій та щодня обслуговує тисячі покупців.

Чат-бот — це програмне забезпечення, яке імітує спілкування з користувачем за допомогою текстових або голосових повідомлень. Його можна розглядати як віртуального менеджера, здатного з'ясовувати потреби клієнта та допомагати в їх задоволенні. Комунікація відбувається у текстовому, інтерактивному або мультимедійному форматі, що робить взаємодію з ботом зручною та ефективною.

Чат-боти інтегруються в соціальні мережі, месенджери, мобільні додатки, вебсайти та пошукові системи. Вони стали популярним інструментом автоматизації взаємодії з клієнтами, маркетингових кампаній та підтримки користувачів.

1.2 Аналіз існуючих аналогів телеграм-ботів для реклами продукції

На сьогодні в інтернеті величезна кількість ботів, які досить різноманітні. Щоб орієнтуватися у різноманітності ботів та проаналізувати розглянемо деякі боти, після того як розглянемо нашу основну платформу – Telegram.

Telegram - месенджер з відкритим вихідним кодом, створений братами Павлом Дуровим і Миколою Дуровим.

Чат боти найвідоміші тим, що використовуються як агенти служби підтримки клієнтів. Вони здатні відповісти на більшість запитань клієнта без необхідності втручання людини. Проте крім цього існує безліч інших корисних застосувань для таких програм. Існують боти, що дозволяють забронювати готельні номери, скласти людині компанію, організовувати зустріч для групи людей, відправити

гроші користувачам, тощо.

Хоча є багато ботів для доставки їжі чи бронювання квитків, бот саме для продуктових мереж із гнучким пошуком і акційними пропозиціями менш поширений в Україні. Існуючі аналоги мають такі обмеження:

BotFather-базовані прості боти надають тільки інформацію про базові команди, без інтеграції з актуальною базою даних товарів;

Боти-застосунки від конкурентів (наприклад, “Сільпо Бот”, “АТБ Онлайн” у мобільному додатку) не мають інтеграції через Telegram і змушують користувача встановлювати окрему програму;

Родові rule-based боти не підтримують обробку запитів у вільній формі, що не завжди зручно при написанні назви товару з помилками чи скороченнями.

За результатами аналізу, потрібно запровадити гнучкіший підхід:

Забезпечити швидкий пошук за різними критеріями (назва, категорія, регіон).

Вбудувати модуль NLP для розпізнавання ключових слів у запитах користувачів навіть без точного написання.

Реалізувати систему кешування для пришвидшення відповіді на популярні запити (наприклад, “акції сьогодні”).

Приклади аналогів програмного забезпечення:

RozetkaBot - телеграм-бот компанії Rozetka, орієнтований на інтеграцію з онлайн-магазином. Дозволяє здійснювати пошук товарів, переглядати категорії, акції, перейти до оформлення замовлення через сайт.

Переваги:

- Зручне і швидке текстове або кнопочове меню;
- Велика база товарів, доступна через пошук;
- Гарна інтеграція з веб-версією магазину;
- Актуальність даних синхронізується з сайтом.

Недоліки:

- Часто перенаправляє користувача на зовнішній сайт замість виконання дій у боті;

- Обмежена кількість інтерактивних функцій прямо в Telegram;
- Відсутність персоналізації (наприклад, історія запитів).

SilpoBot - Бот для клієнтів мережі супермаркетів Silpo. Основна функція — надання інформації про акції, картку «Власний рахунок», програму лояльності, список магазинів.

Переваги:

- Інтеграція з системою лояльності («Власний рахунок»);
- Виведення даних про знижки у зручному форматі;
- Пошук магазинів за геолокацією.

Недоліки:

- Не підтримує повноцінний пошук товарів за назвою;
- Для повної функціональності потрібна авторизація через сайт;
- Низька стабільність відповіді при великій кількості запитів.

EVA ChatBot - Бот мережі магазинів EVA. Орієнтований на комунікацію з постійними клієнтами, підтримує надання знижок, інформування про новинки, замовлення через сайт.

Переваги:

- Яскравий та інтуїтивно зрозумілий інтерфейс;
- Підтримка персоналізованих пропозицій;
- Можливість залишити відгук чи отримати підтримку.

Недоліки:

- Основні дії — перенаправлення на сайт;
- Відсутність повноцінного каталогу товарів у боті;
- Функціонал обмежений для неавторизованих користувачів.

Mezi - корисний бот, який дозволяє легко спланувати подорож. З його допомогою можна забронювати квитки, готелі та навіть вечерю у ресторані. Скажіть Чат - боту, коли і куди ви хочете поїхати і звідки ви хочете полетіти і він підбере та забронює рейс та готель замість вас. Цей бот усуває потребу у самостійному аналізуванні десятка вебсайтів для бронювання авіа перельотів та

готелів, а надає зручний інтерфейс для їх пошуку. Головними його перевагами можна назвати швидкість роботи та спілкування з користувачем складними реченнями, бот спілкується майже як жива людина, недоліком є те, що він може спілкуватися тільки англійською мовою, що робить його недоступним для багатьох людей. Його роботу продемонстровано на рисунку 1.1

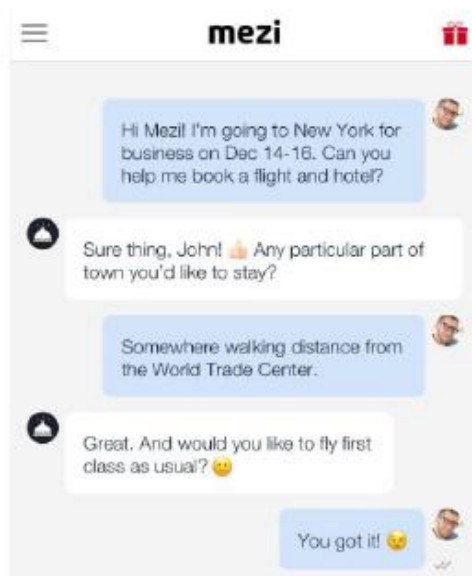


Рисунок 1.1 – інтерфейс чат-боту Mezi

Жоден із наведених ботів не реалізує одночасно повний офлайн-каталог товарів, просту взаємодію та можливість працювати без авторизації. Телеграм-бот для реклами продукції мережі АТБ має на меті поєднати зручність кнопкового інтерфейсу, простоту навігації, доступність інформації без переходу на сайт і мінімальний поріг входу для користувача.

1.3 Постановка задачі на розробку програмного забезпечення телеграм-боту для реклами продукції компанії АТБ

Отже, поглиблений аналіз предметної області підтверджує доцільність розробки Телеграм-бота для мережі супермаркетів АТБ з розширеним функціоналом та інтелектуальними можливостями. Це створить новий рівень взаємодії з клієнтами та забезпечить конкурентну перевагу на ринку української

роздрібної торгівлі.

На підставі проведеного аналізу предметної області встановлено, що:

- а) У сфері роздрібної торгівлі чат-бот має критичне значення для підвищення оперативності обслуговування клієнтів та підтримки конкурентних переваг мережі АТБ.
- б) Телеграм-бот, інтегрований із базою даних товарів і акцій, може значно знизити навантаження на службу підтримки та збільшити лояльність клієнтів.
- в) Для успішної реалізації потрібне поєднання rule-based та NLP-підходу, щоб задовольнити різноманітні сценарії взаємодії: від простого вибору кнопок до аналізу запиту у вільній формі.

У відповідь на виклики конкурентного середовища в Україні було вирішено розробити бот, який одночасно:

- а) Забезпечує багатофункціональну навігацію (пошук, акції, інформація про компанію);
- б) Має автоматизовані сповіщення про персональні знижки;
- в) Підтримує архітектуру з можливістю масштабування з ростом асортименту та географії охоплення.

На основі вище викладеного матеріалу, задачею кваліфікаційної роботи є розробка програмного забезпечення у вигляді чат-боту для реклами продукції з простим і зрозумілим інтерфейсом для можливості знаходження товарів відповідно до локалізації користувача.

Користувачем чат-боту може бути будь-який авторизований користувач Telegram. Застосунок матиме дві ролі це користувач та адміністратор, який може вносити нові данні в базу, перезапускати бота, та інше.

ПЗ має забезпечувати такий функціонал:

Функції додатку для користувача:

- відправка повідомлень;
- керування меню;

- отримання повідомлень.

Функції додатку для адміністратора:

- загрузка бази даних у бота;
- перезапуск бота;
- додавання позицій.

Вхідні дані для користувача:

- інформація про користувача;
- інформація про повідомлення;
- інформація про налаштування.

Вихідні дані для користувача:

- дані каталогу;
- дані про наявність;
- дані про ціни;
- дані акції;
- дані компанію;
- дані про бота.

Вхідні дані для адміністратора:

- список товарів;
- список виробників;
- список категорій товарів.

Вихідні дані для адміністратора:

- звіти по використанню бота користувачами;
- звіти по переглядам та відомості товару;
- звіти про дохід з реклами.

Розробка чат-боту виконується на основі технічного завдання, документа, в якому максимально докладно і точно вказані вимоги до чат-бота. Технічне завдання розробки чат-бота для реклами продукції АТБ наведено в додатку А.

2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТЕЛЕГРАМ-БОТУ ДЛЯ РЕКЛАМИ ПРОДУКЦІЇ КОМПАНІЇ АТБ

2.1 Аналіз вимог до програмного забезпечення телеграм-боту для реклами продукції компанії АТБ і розробка моделі варіантів використання телеграм-боту для реклами продукції компанії АТБ

Для забезпечення стабільної та ефективної роботи Телеграм-бота, призначеного для реклами продукції мережі супермаркетів АТБ, необхідно чітко визначити вимоги до програмного забезпечення. Ці вимоги охоплюють як функціональні аспекти системи, так і нефункціональні характеристики, які визначають якість, продуктивність, масштабованість та зручність використання.

Формулювання вимог базується на аналізі предметної області, очікуваннях кінцевих користувачів, а також технічних можливостях месенджера Telegram і використовуваних інструментів розробки. Вимоги сформульовані з урахуванням принципів modularity (модульність), usability (зручність використання), maintainability (підтримуваність) та extensibility (можливість розширення функціональності).

У цьому розділі подано перелік функціональних вимог до чат-бота, а також визначено загальні обмеження, які впливають на реалізацію та розгортання програмного забезпечення. Такий підхід дозволяє систематизувати очікувану поведінку бота, полегшити тестування та забезпечити високу якість кінцевого продукту.

Функціональні вимоги:

- Ініціалізація взаємодії з користувачем

Після запуску бота користувач має отримати привітальне повідомлення з кнопками доступу до основного меню.

- Головне меню

Бот повинен відображати меню з такими опціями:

- «Каталог товарів»
- «Пошук»
- «Акції»

- «Про нас»

- Пошук товарів

Користувач може ввести назву товару (повну або часткову), бот здійснює пошук у базі даних і повертає результат зі зображенням, назвою, ціною та категорією товару.

- Перегляд каталогу

Бот дозволяє переглядати повний перелік товарів за категоріями. Користувач обирає категорію й отримує список відповідних товарів.

- Вивід акційних пропозицій

За запитом користувача бот показує перелік акційних товарів із коротким описом і терміном дії акції.

- Довідкова інформація

У розділі «Про нас» бот виводить коротку інформацію про компанію АТБ, контактні дані або посилання на офіційні ресурси.

- Обробка невідомих запитів

У разі, якщо користувач вводить некоректну команду або повідомлення, бот повинен виводити відповідь-заготовку з підказкою.

- Адміністративні функції (за необхідності)

Передбачити можливість авторизованого доступу до команд:

- Додавання нових товарів;
- Оновлення списку акцій;
- Перезапуск бота.

Нефункціональні вимоги:

- Інтерфейс користувача

Бот повинен мати інтуїтивно зрозуміле кнопочке меню та підтримку текстових запитів. Комунікація здійснюється українською мовою.

- Сумісність і платформа

Бот має працювати у середовищі Telegram на всіх підтримуваних платформах (Android, iOS, Web, Desktop).

- Продуктивність

Середній час відповіді бота на запит не повинен перевищувати 2 секунд у звичайному режимі навантаження.

- Масштабованість

Архітектура повинна дозволити легке додавання нових функцій без значних змін у вже реалізованій логіці.

- Безпека

Бот не повинен зберігати персональні дані користувачів. Адміністративні дії мають бути захищені авторизацією (наприклад, через Telegram ID або секретний ключ).

- Резервування та відновлення

У випадку збою повинна бути можливість відновлення даних з резервної копії бази.

2.2 Розробка архітектури програмного забезпечення телеграм-боту для реклами продукції компанії АТБ

Телеграм-бот для рекламного інформування клієнтів мережі супермаркетів АТБ є прикладом сучасного сервісу, який працює у режимі реального часу та має на меті забезпечення швидкого доступу до інформації про товари, акції та компанію. Для цього необхідно розробити надійну та масштабовану архітектуру програмного забезпечення, яка дозволить ефективно обробляти запити, працювати з базою даних і забезпечувати легке оновлення.

Архітектура — це основа, на якій базується будь-який програмний продукт. Її розробка дозволяє чітко визначити ролі кожного з компонентів системи, способи їхньої взаємодії та особливості реалізації. У цьому підрозділі буде обґрунтовано вибір архітектурного підходу, наведено специфікації компонентів і діаграми, що відображають структуру телеграм-бота.

2.2.1 Обґрунтування вибору архітектурного підходу

Перед створенням телеграм-бота необхідно обрати архітектуру, яка забезпечить стабільну роботу, легкість супроводу та можливість розвитку. У цьому підпункті розглянуто основні причини, чому обрано саме клієнт–серверну модель, яка є класичним та перевіреним підходом для таких типів додатків, а також

наведено її переваги в контексті розробки чат-бота. Її переваги:

- Розділення відповідальностей: клієнт відповідає лише за взаємодію, а сервер — за обробку логіки.
- Масштабованість: можна розширювати функціональність серверної частини без змін на стороні клієнта.
- Гнучкість і модульність: легко оновлювати, додавати або тестувати окремі компоненти.
- Сумісність із Telegram Bot API: серверна частина працює через webhook або polling.

Клієнт–серверна архітектура забезпечує чітке розділення логіки, дозволяє масштабувати окремі частини системи, спрощує оновлення та відповідає сучасним стандартам для інтерактивних сервісів у месенджерах.

2.2.2 Структура архітектури

Щоб реалізувати обрану архітектурну модель, необхідно визначити ключові компоненти системи та їхню роль. У цьому підпункті описано складові архітектури Телеграм-бота, що відповідають за взаємодію з користувачем, обробку запитів, зберігання даних і передачу повідомлень.

Компоненти системи розділено на клієнтську частину (користувач у Telegram), серверну частину (бот на Python), API-шлюз (Telegram Bot API) та базу даних (MongoDB), кожен з яких виконує свою функцію в загальній структурі.

Архітектура Телеграм-бота складається з таких основних компонентів:

Клієнтський пристрій

- Мобільний, десктопний або веб-додаток Telegram.
- Служить інтерфейсом для введення команд і відображення відповідей.
- Користувач взаємодіє з ботом через кнопки меню або текстові запити.

Telegram Bot API

- Посередник між Telegram-клієнтом і серверною частиною.
- Приймає вхідні повідомлення та передає їх до серверу.

- Після обробки запиту — повертає відповідь назад користувачу.

Сервер з Телеграм-ботом (Python)

- Основна логіка реалізована на мові Python з використанням бібліотеки `python-telegram-bot`. Бібліотека `python-telegram-bot` [3] забезпечує зручний інтерфейс для взаємодії з Telegram API [4], дозволяючи створювати асинхронних ботів із використанням структурних підходів програмування.

Включає:

- Обробку команд: `/start`, «Акції», «Пошук», «Каталог», «Про нас»;
- Генерацію меню;
- Формування відповідей;
- Запити до бази даних;
- (Опціонально) авторизацію адміністратора та оновлення даних.

База даних (MongoDB)

- Документно-орієнтована база даних, яка зберігає:
- Товари з описами, цінами, категоріями;
- Акційні пропозиції з термінами дії;
- Категорії;
- Логи запитів;
- (За потреби) шаблони повідомлень або налаштування.

2.2.3 Взаємодія між компонентами

Комунікація між компонентами реалізується через API-запити та відповіді. Бот отримує команду, обробляє її, при потребі звертається до бази даних і надсилає готове повідомлення назад користувачу.

1. Користувач надсилає запит (наприклад, «Каталог»).
2. Запит потрапляє до Telegram Bot API. [4]
3. API передає повідомлення до серверної частини (бота).
4. Бот розпізнає запит, звертається до MongoDB при потребі.
5. Отримані дані форматуються та надсилаються назад користувачу через Telegram

API.

2.2.4 Специфікація компонентів

Для якісної реалізації кожного блоку системи необхідно визначити використовувані технології, середовища розгортання та особливості реалізації. У цьому підпункті надано специфікацію кожного компоненту, його призначення та роль у загальній системі.

Для реалізації телеграм-бота, призначеного для реклами продукції компанії АТБ, була використана сучасна архітектура, що включає такі основні компоненти:

- Клієнтська частина (Telegram Client)

У ролі інтерфейсу для кінцевого користувача використовується офіційний клієнт Telegram — мобільний (Android, iOS) або веб-версія. Саме через нього користувачі взаємодіють із ботом, переглядають меню, вводять запити, переглядають акційні пропозиції, ціни та інформацію про товари. Такий підхід забезпечує зручність доступу з будь-якого пристрою без потреби встановлювати додаткові застосунки.

- Telegram Bot API [4]

Цей компонент є основним каналом зв'язку між клієнтською частиною та серверною логікою. Він надає набір методів, які дозволяють отримувати повідомлення від користувачів, відповідати на них, керувати кнопками меню та іншими елементами взаємодії. API Telegram гарантує стабільну і безпечну передачу даних між користувачем і ботом.

- Серверна логіка бота (Bot Logic)

Серцем системи є серверна частина, реалізована за допомогою мови програмування Python 3.11 з використанням бібліотеки `python-telegram-bot`. [3] [9] Саме тут реалізовано усю логіку обробки запитів користувача, включно з визначенням ключових команд, сценаріїв використання, викликів до бази даних, персоналізації відповіді та обробкою помилок. Цей компонент забезпечує гнучкість у розширенні функціоналу та підтримку декількох сценаріїв взаємодії.

- База даних (MongoDB Atlas)

Для зберігання даних про товари, акційні пропозиції, магазини, а також історії запитів використовується хмарна база даних MongoDB Atlas. [10] Цей сервіс забезпечує масштабованість, високу швидкодію та доступність даних у будь-який момент часу. Нереляційна модель зберігання дозволяє швидко працювати з даними без жорсткого схемного обмеження.

- Хостинг (PythonAnywhere або Heroku)

Для розгортання серверної частини використано хмарні хостинг-платформи, зокрема PythonAnywhere або Heroku. Вони дозволяють запускати Python-застосунки у хмарі без необхідності підтримки власного сервера. Сервер обробляє запити у режимі webhook або long polling, підтримуючи постійну доступність бота для користувачів.

2.2.5 Візуалізація архітектури програмного забезпечення

Для забезпечення чіткого уявлення про логічну й фізичну організацію системи, а також для полегшення комунікації між розробниками та іншими зацікавленими сторонами, доцільним є використання діаграм UML. У контексті розробки Телеграм-бота для рекламної взаємодії з користувачами мережі супермаркетів АТБ була побудована діаграма компонентів:

Діаграма компонентів (Component Diagram)

Ця діаграма демонструє логічну структуру програмного забезпечення, показуючи ключові програмні модулі (компоненти) та взаємозв'язки між ними. У рамках проєкту виділено такі основні компоненти:

- Клієнтський інтерфейс (Telegram Client) — забезпечує введення користувацьких команд;
- Telegram Bot API — інтерфейс, через який бот отримує повідомлення від користувача;
- Серверна логіка (Bot Core Logic) — відповідає за обробку команд, взаємодію з базою даних та формування відповідей;
- База даних (MongoDB) — зберігає структуровану інформацію про товари, акції та інше.

Архітектура подана діаграмою компонентів зображена на рисунку 2.1

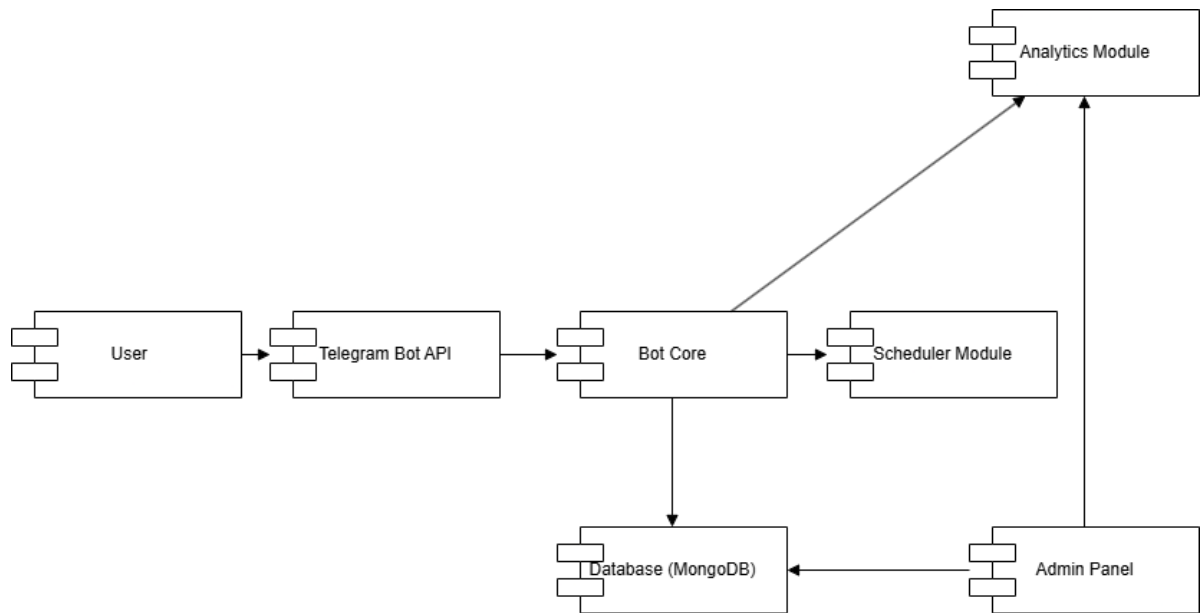


Рисунок 2.1 – Архітектура чат-боту

На діаграмі компонентів представлено основні логічні частини архітектури Телеграм-бота. Вона ілюструє, як окремі модулі взаємодіють між собою в межах єдиної системи.

Користувач надсилає повідомлення через Telegram-клієнт, який передає їх до Telegram Bot API. Далі запити обробляє основна серверна логіка, реалізована мовою Python. У випадку потреби отримання чи збереження інформації сервер звертається до бази даних MongoDB.

Кожен компонент виконує строго визначену функцію, що забезпечує модульність і спрощує подальшу підтримку або масштабування системи.

2.3 Проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ

Процес проєктування програмного забезпечення Telegram-боту охоплює послідовні етапи, які забезпечують поступовий перехід від концепції до реалізації системи. На даному етапі відбувається формалізація структури, поведінки та взаємодії компонентів майбутнього застосунку, виходячи з функціональних і нефункціональних вимог, визначених у технічному завданні.

Розділ містить три ключові етапи проєктування. Перший — ескізний проєкт, в якому визначаються основні логічні компоненти, моделі даних, інтерфейси

користувача та варіанти використання. Наступний етап — технічний проєкт, що передбачає створення структурних і поведінкових UML-діаграм, таких як діаграми класів, послідовності та кооперації. Третій — робочий проєкт, на якому здійснюється безпосередня реалізація компонентів згідно з технічним планом, включаючи написання програмного коду, налаштування середовища та перевірку працездатності системи.

2.3.1 Ескізний проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ

На етапі ескізного проєктування виконується високорівневе опрацювання архітектури програмного забезпечення, з метою формалізації функціональних можливостей, користувацьких сценаріїв та концепції інформаційного обміну між компонентами системи. Телеграм-бот для реклами продукції компанії АТБ, як інформаційна система, орієнтована на споживача, вимагає чіткої структури й простоти у взаємодії.

Проектування інтерфейсу

Інтерфейс користувача (надалі UI – *UserInterface*) - то, як виглядає інтерфейс і те, які фізичні характеристики набуває. Це інтерфейс, що забезпечує передачу інформації між користувачем-людиною і програмно-апаратними компонентами комп'ютерної системи.

Під сукупністю засобів і методів інтерфейсу користувача маються на увазі засоби та методи.

Засоби:

виведення інформації з пристрою до користувача - весь доступний діапазон впливів на організм людини (зорових, слухових, тактильних, нюхових і т. д.) - екрани (дисплеї, проєктори) і лампочки, динаміки, зумери і сирени, вібромотор і т. д. і т.п.

введення інформації / команд користувачем в пристрій - безліч всіляких пристроїв для контролю стану людини - кнопки, перемикачі, потенціометри, датчики положення і руху, сервоприводи, жести особою і руками, навіть знімання мозкової активності користувача.

Методи:

набір правил, закладених розробником пристрою, згідно з якими сукупність дій користувача повинна привести до необхідної реакції пристрою і виконання необхідного завдання - так званий логічний інтерфейс.

Вибір засобів моделювання

Найбільш відомими візуальними моделями для проектування комп'ютерних систем та їхніх програмних забезпечень є діаграми мови UML. За допомогою UML можна візуалізувати, специфікувати, конструювати і документувати артефакти програмних систем. UML придатний для моделювання будь-яких систем: від інформаційних систем масштабу підприємства до розподілених веб-додатків і навіть вбудованих систем реального часу.

Для проектування Телеграм бота доцільно буде використати мову моделювання UML. [6]

Інформаційна модель програмного забезпечення

Інформаційна модель відображає логічну структуру даних, що використовуються програмним забезпеченням під час обробки запитів користувачів у межах бізнес-процесів. Вона демонструє взаємозв'язки між основними об'єктами предметної області, їхні атрибути та типи зв'язків між сутностями. У контексті реалізації Телеграм-бота для реклами продукції компанії АТБ, інформаційна модель є ключовим етапом проектування, який дозволяє формалізувати структуру збереження, обробки та передачі даних.

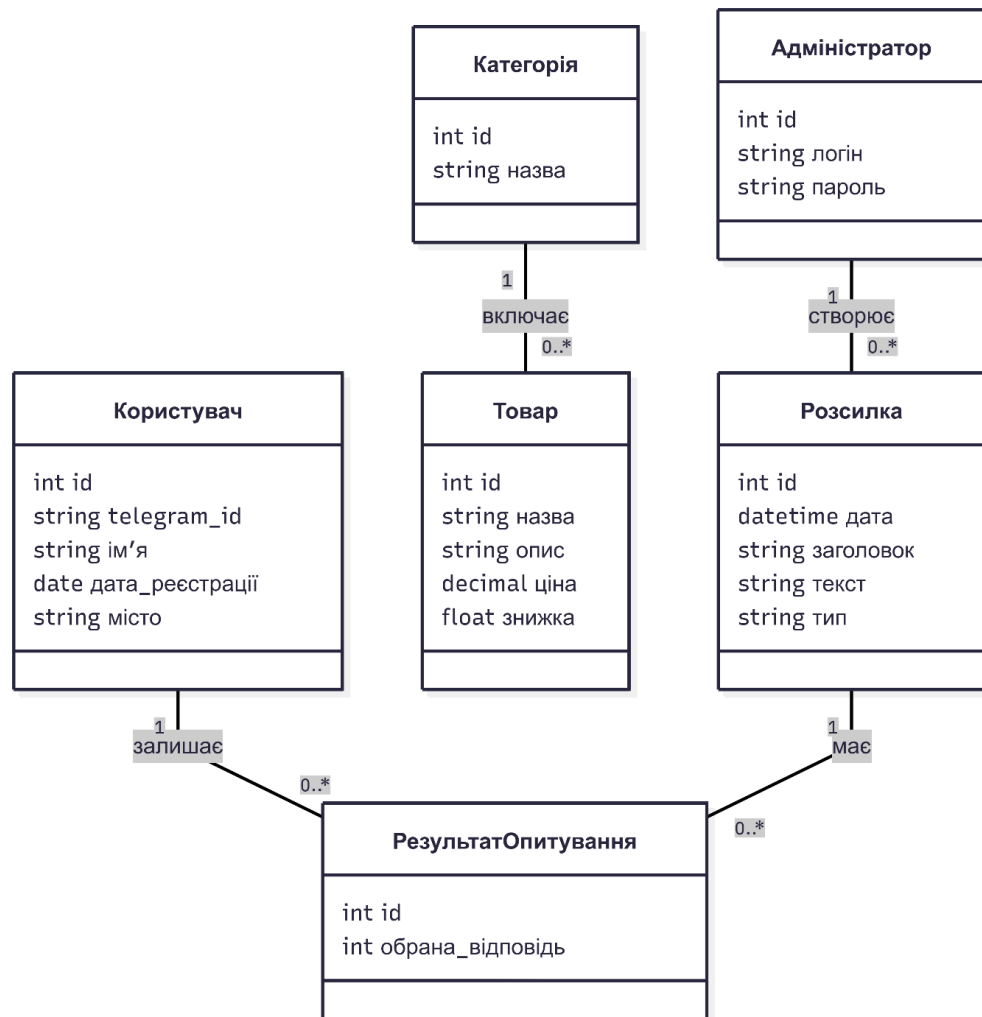


Рисунок 2.2 – інформаційна модель

Модель описує основні сутності, з якими працює програмне забезпечення:

Користувач — відображає особу, що взаємодіє з ботом через Telegram;

Категорія товару — групує товари за типами;

Товар — описує об'єкт продажу, має атрибути: назва, ціна, опис, знижка тощо;

Розсилка — представлення інформаційного повідомлення, яке надсилається користувачеві;

Адміністратор — особа з правами на виконання службових команд;

Результат опитування — відповіді користувачів на розсилки опитувального типу.

Інформаційна модель слугує основою для формування логічної та фізичної моделі даних, що реалізуються у базі даних. Вона дозволяє визначити залежності між об'єктами, забезпечити нормалізацію структури зберігання інформації, а також

гарантувати коректність обміну даними між компонентами системи.

На основі інформаційної моделі було реалізовано структуру даних у СУБД MongoDB. Вона враховує необхідність масштабування, зберігання структурованих і напівструктурованих даних, а також підтримку швидкого доступу до інформації за запитам користувачів.

Розробка моделі варіантів використання

При створенні будь-якої програми визначаються вимоги, яким повинна задовольняти система. Моделі варіантів використання описують функціональне призначення системи або те, що система повинна робити. Сценарії це можливі послідовності дій для кожного варіанту використання.

Модель варіантів використання телеграм-бота представлена на рисунку 2.3.

Специфікації варіантів використання Telegram бота представлені в таблицях 2.1-2.3.



Рисунок 2.3 – діаграма варіантів використання телеграм-бота

Таблиця 2.1 – Специфікація прециденту «Пошук товару»

Характеристика	Опис
1	2
Короткий опис	Бот отримає повідомлення користувача і формує відповідь на запит
Учасники	Користувач, телеграм-бот
Передумови	Користувач ініціалізував бота
Основний потік подій	1. Користувач запускає Telegram-бот і вводить ключове слово (наприклад, назву товару). 2. Система отримує запит і перевіряє, чи міститься введене слово у базі даних. 3. Якщо збіг знайдено, система повертає список відповідних товарів з описом. 4. Користувач переглядає список та, за потреби, вводить уточнюючий запит. 5. Система оновлює результати на основі нового запиту.
Альтернативний потік подій	Уведена некоректна назва товару. Надання можливості повторного введення даних.

Таблиця 2.2 – Специфікація прециденту «Перегляд акцій»

Характеристика	Опис
1	2
Короткий опис	Телеграм-бот отримає команду вивести повідомлення з списком акційних товарів і надсилає його
Учасники	Користувач, телеграм-бот
Передумови	Користувач ініціалізував бота
Основний потік подій	1. Користувач відкриває меню Telegram-бота і натискає кнопку «Акції». 2. Система формує запит до бази даних щодо актуальних акцій. 3. Якщо акції знайдено, система виводить їх у вигляді списку з назвою, строком дії та ціною. 4. Користувач переглядає список, при потребі повертається до головного меню.
Альтернативний потік подій	Відсутність акцій Надання можливості повторного вибору пункту з меню.

Таблиця 2.3 – Специфікація прециденту «Перегляд каталогу»

Характеристика	Опис
1	2
Короткий опис	Телеграм-бот надсилає користувачу повний каталог товарів
Учасники	Користувач, телеграм-бот
Передумови	Користувач ініціалізував бота
Основний потік подій	1. Користувач відкриває Telegram-бот та натискає кнопку «Каталог» у меню. 2. Система формує запит до бази даних з метою отримання списку доступних категорій товарів. 3. Користувачу надсилається перелік категорій (наприклад, «Напої», «Молочна продукція», «Ковбаси»). 4. Користувач обирає одну з категорій. 5. Система формує і надсилає відповідь зі списком товарів у цій категорії, з назвою, коротким описом і ціною. 6. Користувач може повернутись назад до переліку категорій або в головне меню для нової взаємодії.
Альтернативний потік подій	Якщо база даних не загружена користувач не отримає каталог товарів

У цьому підрозділі було визначено основні логічні компоненти майбутньої системи: клієнт (користувач Telegram), інтерфейс Telegram Bot API, серверна логіка та база даних. Кожен компонент має свої зони відповідальності. Клієнт надсилає запити через Telegram, API виконує маршрутизацію, сервер — обробляє запити й формує відповіді, база даних — забезпечує зберігання і доступ до інформації про товари, акції, магазини.

Діаграма переходів станів

Діаграма станів показує, як об'єкт переходить з одного стану в інший та служить для моделювання динамічних аспектів системи.

Проаналізувавши, додаток, розглянув всі можливі умови, при виконанні яких об'єкт або система буде виконувати певні дії чи перебувати в очікуванні появи іншої події. Діаграма переходів станів зображена на рисунку 2.4.

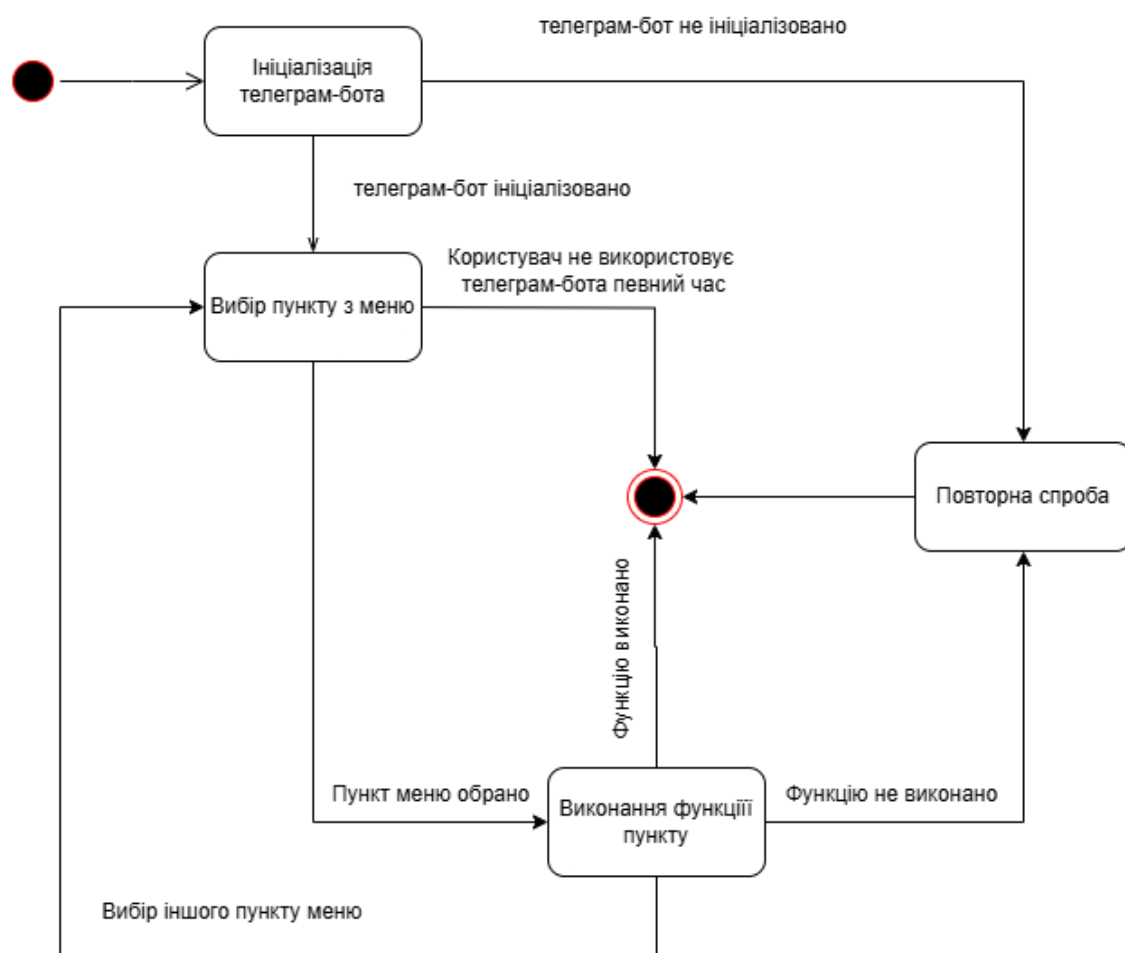


Рисунок 2.4 – Діаграма переходів станів програмного забезпечення

2.3.2 Технічний проєкт програмного забезпечення телеграм-бота для реклами продукції компанії АТБ

Логічна модель даних

На етапі технічного проєктування здійснюється деталізація структури системи. Це дозволяє забезпечити прозору та формалізовану логіку обміну даними, розробити моделі поведінки об'єктів та визначити компоненти для подальшого програмування.

Було створено логічну модель, де вказано основні таблиці (колекції), атрибути, типи даних та їх зв'язки. Наприклад, таблиця «Products» містить поля: name, category, ціна, наявність, ідентифікатор магазину. Модель побудована у відповідності до нереляційного підходу MongoDB.

Логічна модель даних представлена на рисунку 2.6



Рисунок 1.6 – Логічна модель таблиць чат-бота

Реалізація варіантів використання

Для забезпечення повного розуміння логіки роботи телеграм-бота та структури його реалізації на програмному рівні, в цьому підрозділі представлено реалізацію двох ключових варіантів використання: перегляд каталогу товарів та пошук товару.

Реалізація кожного варіанта представлена у вигляді аналізної діаграми класів, яка моделює взаємодію об'єктів типу Boundary (інтерфейс), Control (логіка) та Entity (дані), а також діаграми послідовності, що описує динамічну взаємодію об'єктів у межах конкретного сценарію.

Такий підхід відповідає стандартам об'єктно-орієнтованого аналізу й дозволяє формалізувати модель поведінки програмного забезпечення на основі описаних у попередніх розділах варіантів використання. Це, у свою чергу, сприяє кращому розумінню архітектури системи, спрощує подальшу реалізацію та тестування.

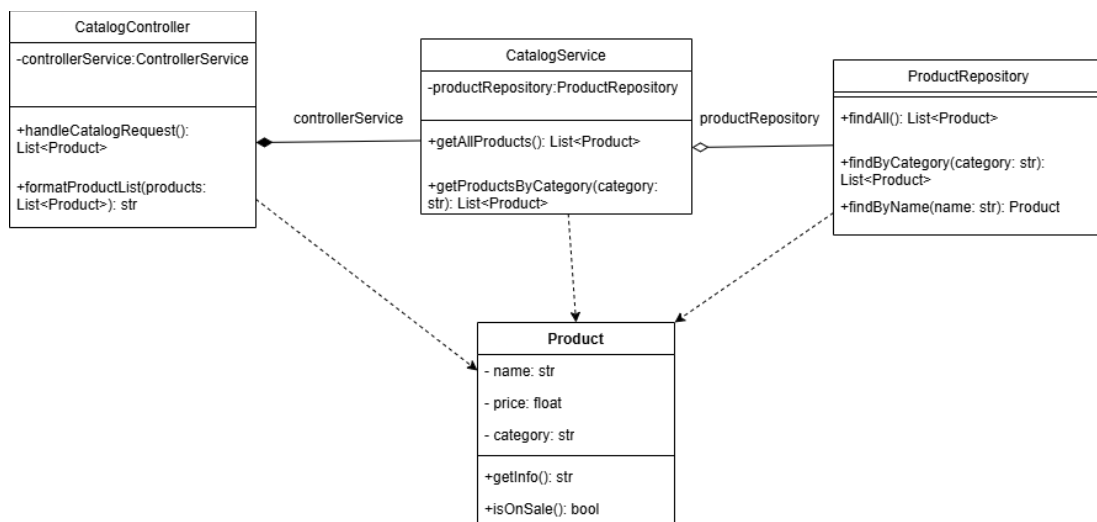


Рисунок 2.7 – діаграма класів аналізу для варіанту використання «Перегляд каталогу товарів»

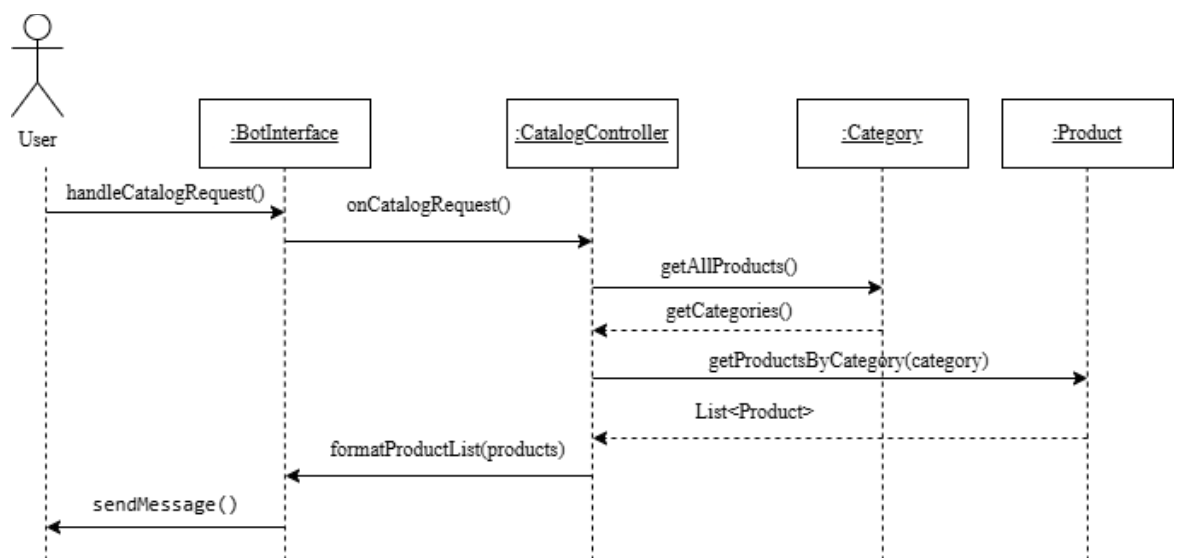


Рисунок 2.8 – діаграма послідовності для варіанту використання «Перегляд каталогу товарів»:

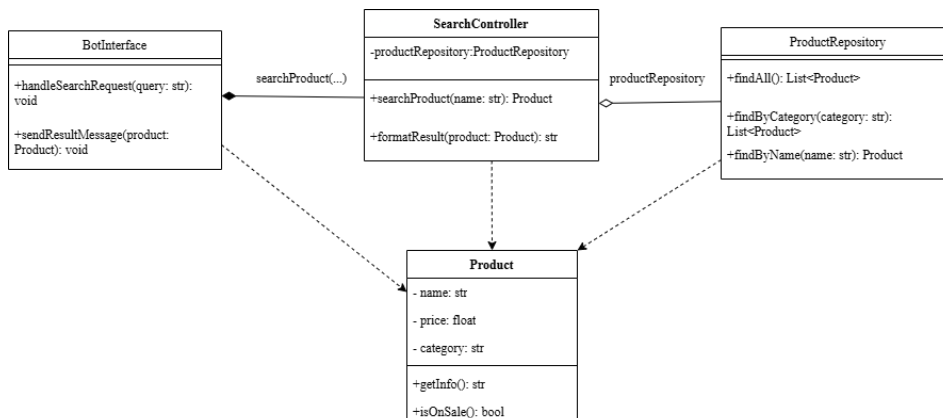


Рисунок 2.9 – діаграма класів аналізу для варіанту використання «Пошук товару»

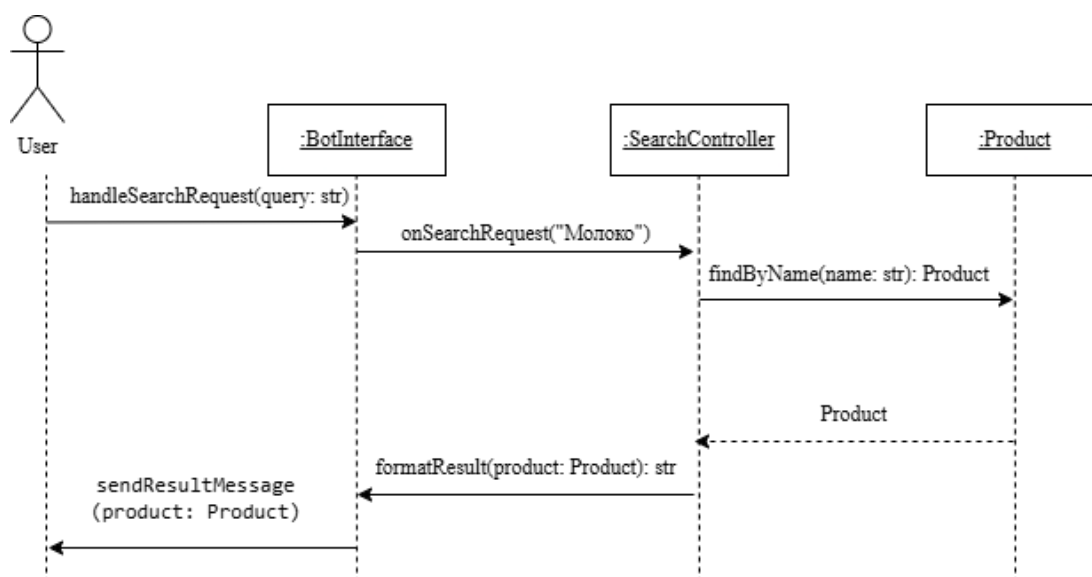


Рисунок 2.10 – діаграма послідовності для варіанту використання «Пошук товару»

2.3.3 Робочий проєкт програмного забезпечення телеграм-боту для реклами продукції компанії АТБ

Робочий проєкт — це практична реалізація всіх елементів, спроектованих на попередніх етапах. На цьому рівні розробник створює код, налагоджує підключення до API, впроваджує структуру бази даних та розгортає серверну частину.

Telegram-бот реалізовано на мові Python 3.11 з використанням бібліотеки

`python-telegram-bot`, що надає зручні обгортки для роботи з Telegram Bot API. Структура коду організована модульно: окремі модулі відповідають за обробку команд, підключення до бази, логування та конфігурацію.

Вибір засобів розробки

Для розробки були застосовані наступні засоби

- Мова програмування: Python.
- IDE: PyCharm;
- База даних: MongoDB;

Python – це високорівнева інтерпретована інтерактивна та об’єктно-орієнтована скриптована мова програмування, розроблена Guido van Rossum.

PyCharm – багатофункціональний редактор коду для мови Python створений для швидкої та ефективної розробки програмного забезпечення. Наявна велика кількість інструментів, які дозволять сконцентруватись на написанні саме бізнес-логіки застосунку не відволікаючись на рутинні речі: налаштування віртуального середовища, підключення системи контролю версій, встановлення зовнішніх бібліотек тощо.

MongoDB – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Розробка фізичної моделі

Фізична модель визначає розміщення даних у зовнішній пам’яті. Вона ще називається внутрішньою моделлю системи і форма її представлення залежить від обраної СКБД. Якщо обрана така СКБД, яка підтримує реляційну модель даних, то треба таблиці разом з атрибутами і зв’язки між таблицями перенести в середовище СКБД з врахуванням вимог до відповідних об’єктів БД.

Таблиця 2.31 - Структура збереження об'єкта «Користувач»

Ідентифікатор	Тип	Ключ	Обмеження
id user	INTEGER	PK	NOT NULL UNIQUE
token	VARCHAR (45)		NOT NULL
Товар id товар	INTEGER		NOT NULL

Таблиця 2.32 - Структура даних колекції «Товари»

Ідентифікатор	Тип	Ключ	Обмеження
id товар	INTEGER	PK	NOT NULL UNIQUE
найавніст	VARCHAR (45)		NOT NULL
кількост	VARCHAR(45)		NOTNULL
tsina	REAL		NOT NULL
name	VARCHAR(45)		NOT NULL
znisgka	VARCHAR(45)		NOT NULL
aksia	VARCHAR(45)		NOT NULL
Магазин id маг	INTEGER		NOT NULL

Таблиця 2.33 – Структура збереження об'єкта «Магазин»

Ідентифікатор	Тип	Ключ	Обмеження
id маг	INTEGER	PK	NOT NULL UNIQUE
найавніст	VARCHAR (45)		NOT NULL
adresa	VARCHAR(45)		NOT NULL

Таблиця 2.34 – Структура даних для колекції «Рекламні оголошення»

Ідентифікатор	Тип	Ключ	Обмеження
id reklama	INTEGER	PK	NOT NULL UNIQUE
name	VARCHAR (45)		NOT NULL
content	VARCHAR(45)		NOT NULL
Магазин id маг	INTEGER		NOT NULL

Розробка моделі розгортання

У процесі розробки програмного забезпечення важливо не лише моделювати логіку взаємодії компонентів системи, а й формалізувати спосіб їх фізичного розміщення у реальному середовищі виконання. Для цього використовується діаграма розгортання (deployment diagram), яка є одним із типів структурних діаграм UML. Вона дозволяє наочно відобразити вузли (сервери, пристрої, середовища виконання), на яких розгортаються окремі компоненти програмного забезпечення, а також канали зв'язку між ними.

У межах кваліфікаційної роботи діаграма розгортання використовується для

відображення архітектури розгортання Телеграм-бота компанії АТБ. Це дає змогу формалізувати взаємозв'язки між клієнтом (мобільний застосунок Telegram), серверною частиною (бекенд-логікою бота), базою даних та іншими технічними ресурсами, що забезпечують роботу сервісу. Таке подання є важливим для оцінки технічних вимог до середовища функціонування системи та забезпечення її масштабованості й надійності.

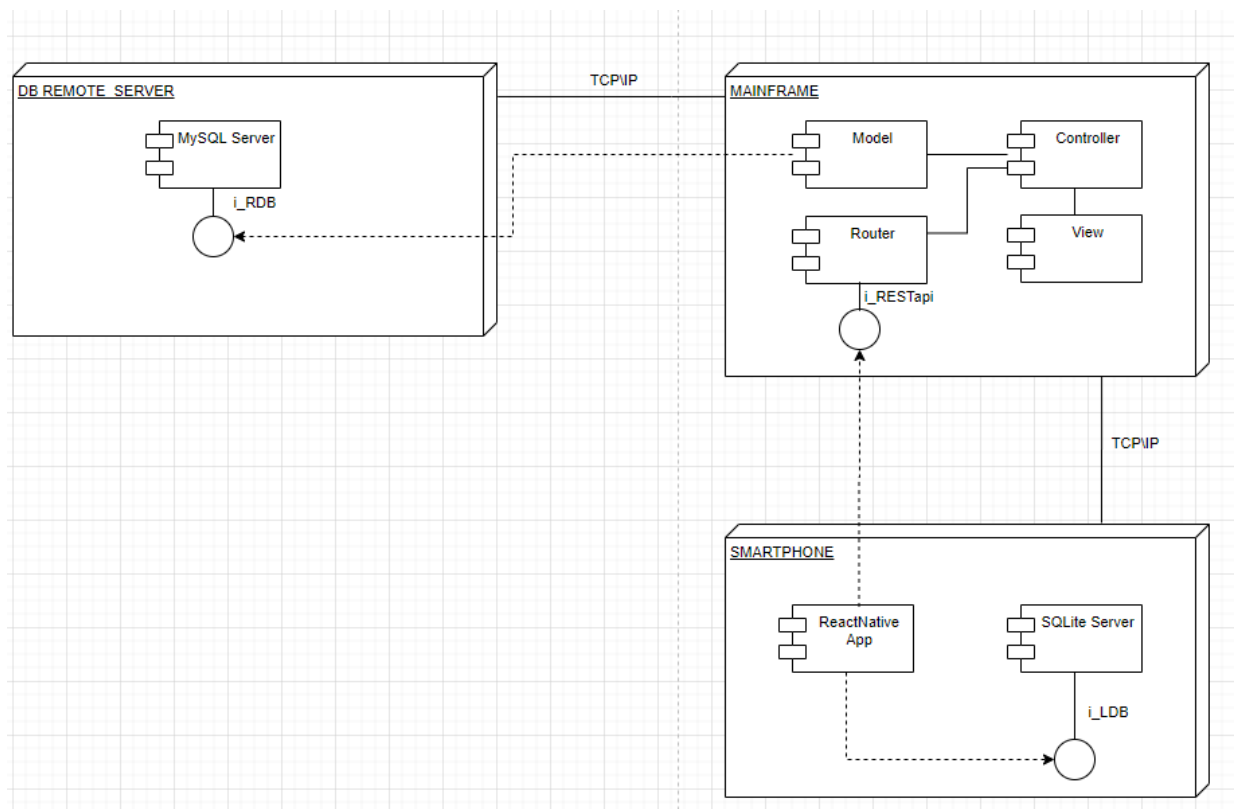


Рисунок 2.11 – Модель розгортання ПЗ боту

2.4 Результати розробки програмного забезпечення телеграм-бота для реклами продукції компанії АТБ

Розробка Телеграм-бота для реклами продукції компанії АТБ пройшла повний цикл проєктування, моделювання, реалізації та випробування. У цьому розділі підсумовуються результати виконаної роботи, надається оцінка реалізованих функцій, технічних та якісних характеристик програмного забезпечення, а також узагальнюються аспекти, які не були докладно розкриті у попередніх розділах через обмеження обсягу тексту. Окрему увагу приділено оцінці відповідності розробленого рішення вимогам технічного завдання та результатам тестування, що підтверджують його функціональну придатність.

2.4.1 Загальний опис результатів

У результаті виконання кваліфікаційної роботи було створено повноцінне клієнт-серверне рішення, що базується на використанні месенджера Telegram як користувацького інтерфейсу. Основна логіка програми реалізована у вигляді Телеграм-бота, який здатний забезпечити інтерактивну взаємодію з користувачем, динамічне формування відповідей на запити, збереження й обробку інформації про товари, акційні пропозиції та інші маркетингові дані. Важливо підкреслити, що під час реалізації розглядалися не лише технічні аспекти взаємодії між користувачем і системою, але й такі параметри як зручність інтерфейсу, швидкодія, масштабованість, стабільність у роботі та відмовостійкість. Розроблене програмне забезпечення підтримує сучасну архітектуру, є гнучким для подальших розширень та відповідає базовим вимогам до подібних інформаційних систем, що функціонують у публічному доступі.

2.4.2 Реалізовані функції

В основу розробленого Телеграм-бота покладено реалізацію тих функціональних сценаріїв, які були визначені у технічному завданні. Під час проєктування та реалізації особлива увага приділялась не лише можливості запуску та обробки базових команд, а й створенню логіки адаптивної взаємодії, яка дозволяє ботові відповідати на запити користувача у контекстно-орієнтованому форматі. Користувач може розпочати роботу з ботом за допомогою стандартної команди запуску, після чого отримує доступ до меню, у якому передбачено можливість перегляду актуальних акцій, здійснення пошуку товарів, а також ознайомлення з інформацією про компанію.

Функціональність пошуку була реалізована як ключова складова. Вона дозволяє здійснювати запити за назвою товару, у тому числі неповною або з помилками, що стало можливим завдяки застосуванню алгоритмів часткового порівняння. При цьому результати формуються динамічно та містять зображення, опис, ціну, а також назву магазину. Окремо опрацьовані сценарії, коли товар не знайдено: бот формує ввічливу відповідь із пропозицією повторити пошук.

Кожна реалізована функція супроводжується механізмом обробки винятків,

що дозволяє уникати аварійних завершень сесії. Також передбачено службові функції для адміністратора, які дозволяють вручну ініціювати оновлення бази даних або перезапустити логіку бота у випадку критичних збоїв.

2.4.3 Частини, що не були детально висвітлені

У межах основного викладу не всі компоненти були висвітлені однаково докладно. Це пояснюється як обмеженнями обсягу, так і бажанням уникнути надмірної деталізації технологічних аспектів. Наприклад, повна структура шаблонів повідомлень, які використовуються для формування відповідей користувачам, не була описана, хоча має велике значення для реалізації локалізації та стилістичного оформлення повідомлень. Так само лише коротко згадано модуль логування, який акумулює дані про звернення користувачів, що потенційно може бути використано для подальшої аналітики.

Окремим залишився й механізм авторизації адміністратора. Він реалізований через порівняння Telegram ID із переліком дозволених і дозволяє доступ лише до обмеженого кола службових функцій. Ці та інші другорядні елементи було винесено у відповідні додатки, де їх подано з прикладами коду та короткими технічними поясненнями.

2.4.4 Кількісні та якісні характеристики

Загальний обсяг вихідного коду Телеграм-бота становить близько 1400 рядків, включаючи основну логіку, обробку команд, підключення до бази даних, конфігурацію та обробку виключних ситуацій. Код структуровано у вигляді окремих модулів, кожен з яких відповідає за конкретну функціональність, що значно спрощує підтримку та масштабування.

Система розгортається у вигляді Python-скрипта в середовищі Python 3.11, що забезпечує сумісність з більшістю сучасних операційних систем. Для стабільної роботи Телеграм-бота необхідно мінімум 256 МБ оперативної пам'яті та безперервне інтернет-з'єднання. Середній час обробки запиту від користувача становить менше 1 секунди, а час відображення меню — близько 0.4 секунди. При паралельній роботі з декількома користувачами система демонструє стійку продуктивність, а кешування часто вживаних запитів дозволяє знизити

навантаження на базу даних на понад третину.

Платформа була протестована у симуляційному середовищі з понад тисячею звернень, що дозволило виявити і виправити окремі незначні недоліки в логіці формування відповідей. Також перевірено сценарії непередбачуваних запитів, які обробляються коректно, без негативного впливу на сесію.

2.4.5 Результати випробувань

Згідно з програмою і методикою випробувань, Телеграм-бот було протестовано в умовах, максимально наближених до реальних. Основна мета полягала у виявленні можливих логічних помилок, затримок у відповіді, несправностей при обробці введення користувача та стабільності взаємодії з базою даних.

Результати тестування підтвердили, що всі основні функції бота працюють у штатному режимі. Система демонструє належну стійкість при багаторазовому виклику одних і тих самих команд, коректно реагує на запити з помилками, а також успішно забезпечує зворотний зв'язок із користувачем. Під час симуляції втрати з'єднання або перезапуску бота не відбулося жодної втрати інформації або порушення логіки взаємодії. Крім того, службові функції захищені належним чином, доступ до них обмежений лише авторизованим адміністраторам.

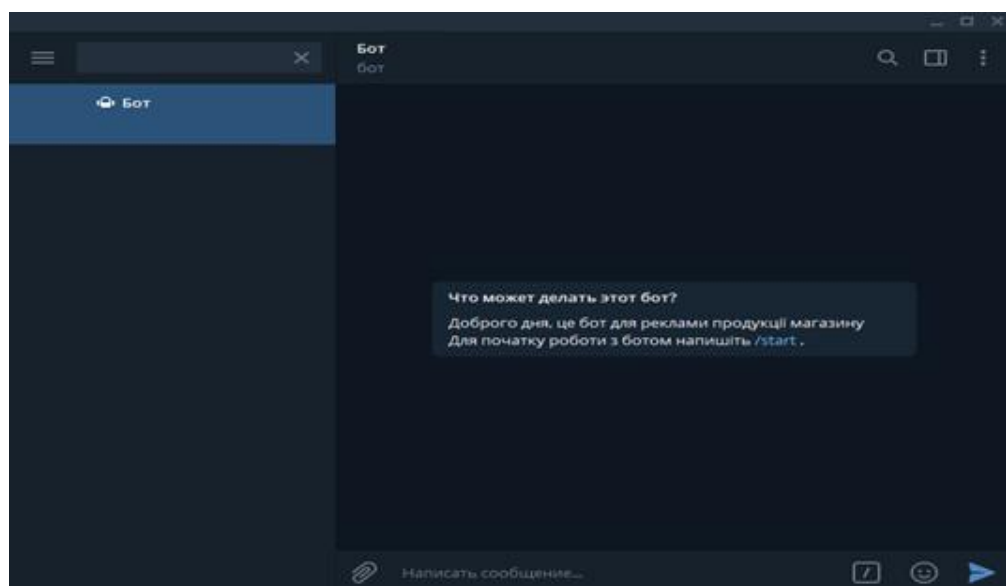


Рисунок 2.12- Інтерфейс Чат – боту до ініціалізації бота

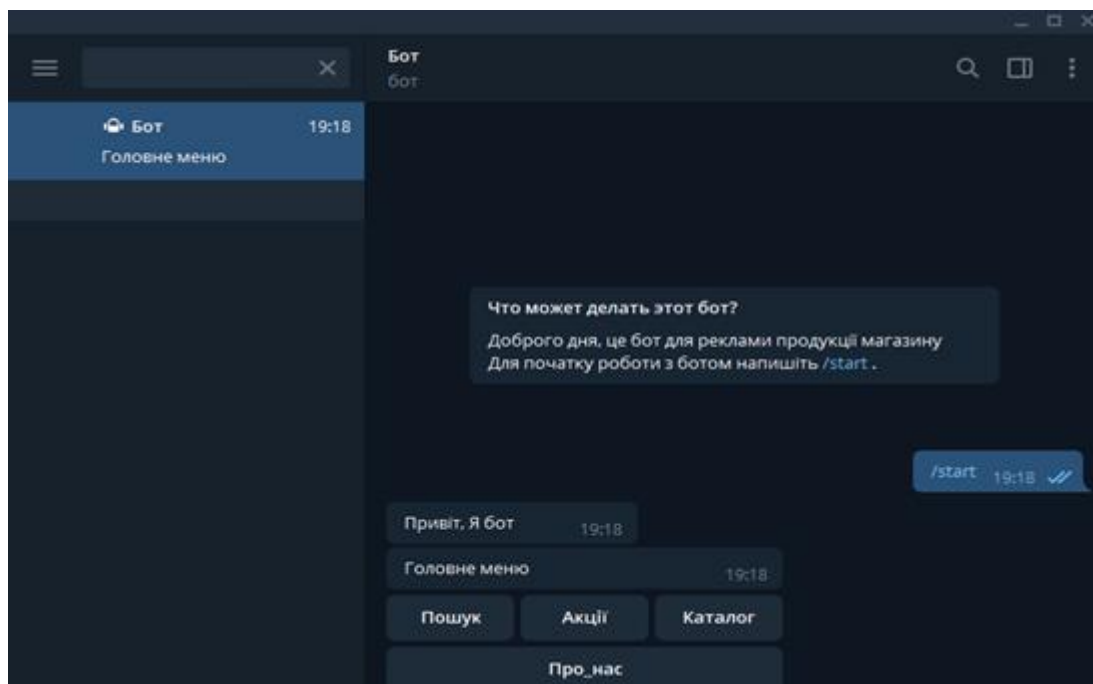


Рисунок 2.13 – Інтерфейс Чат – боту після його ініціалізації

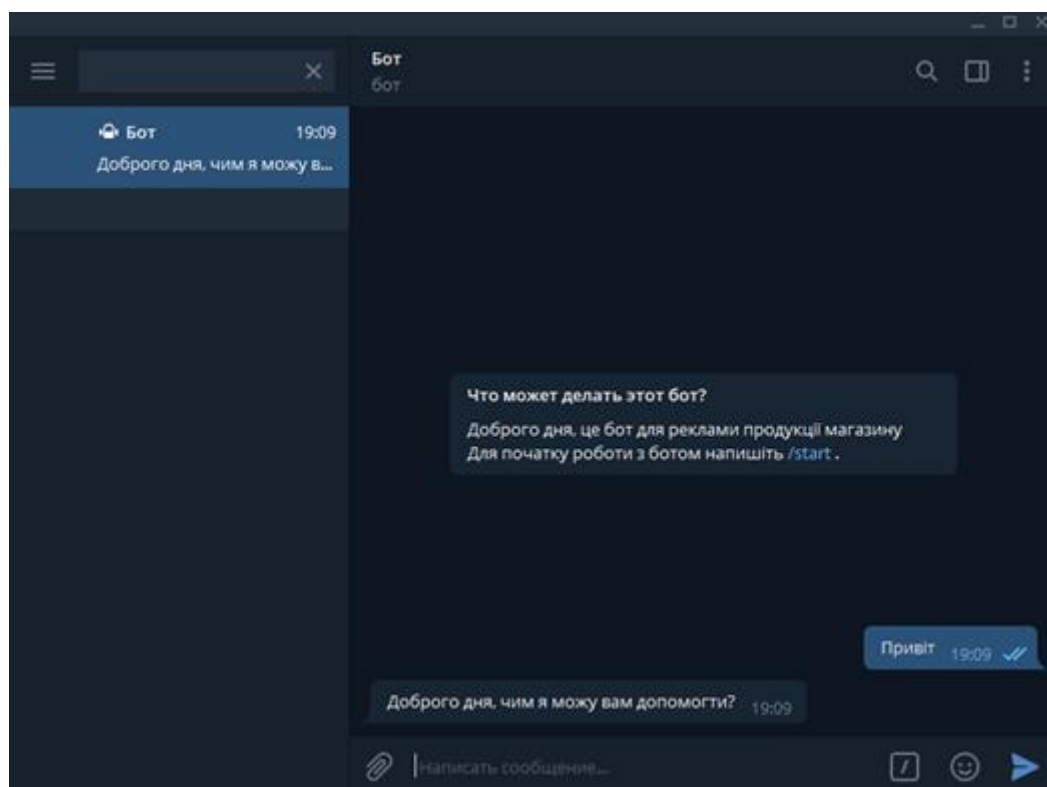


Рисунок 2.14 – Відповідь бота на привітання

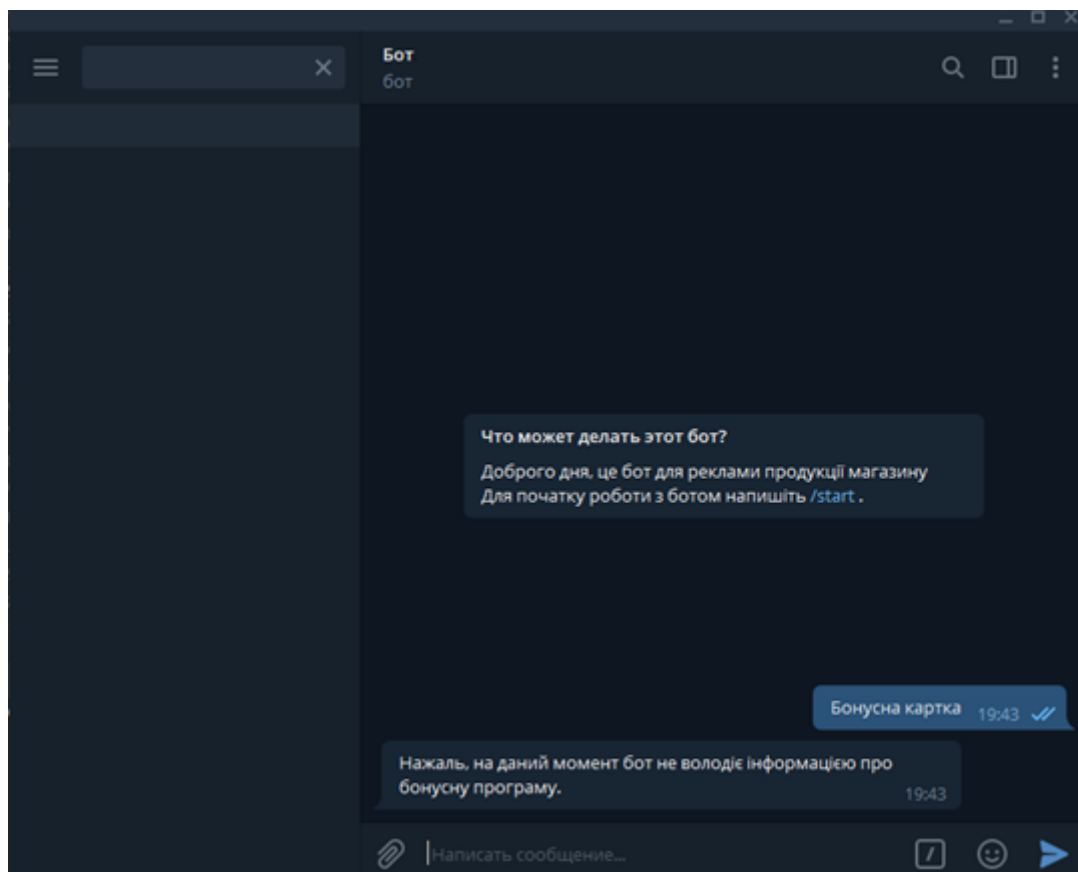


Рисунок 2.15 – Відповідь бота на питання по якому він не має інформації у випадку відсутності загруженої бази даних

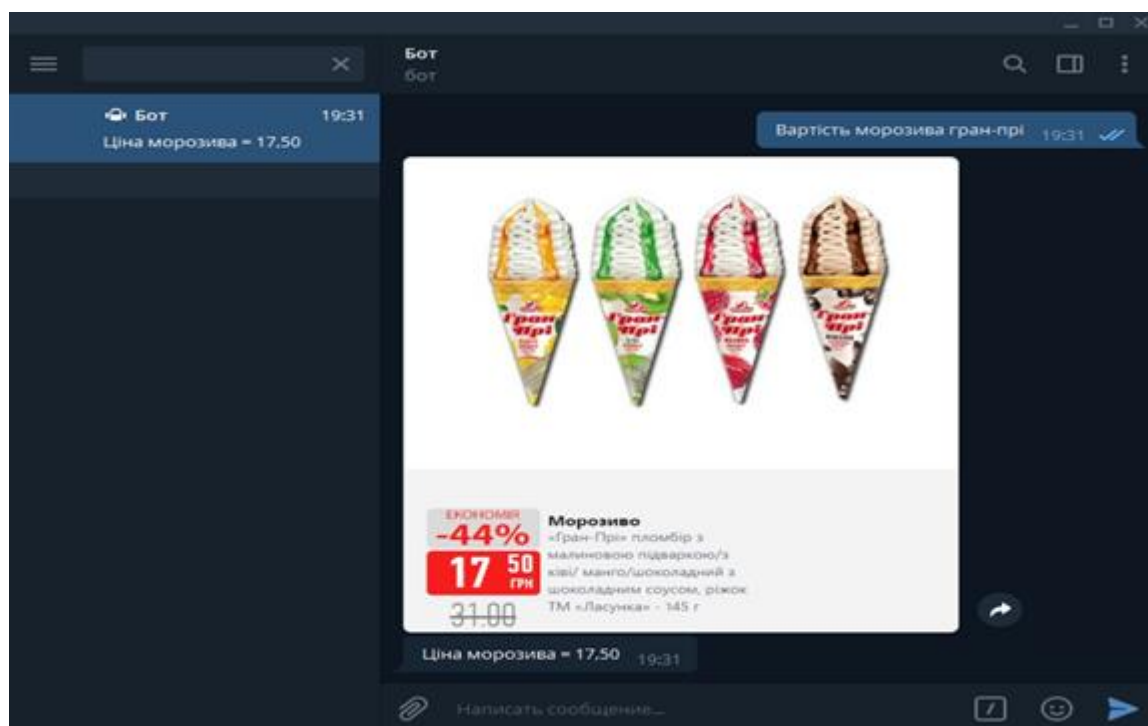


Рисунок 2.16 – Відповідь бота на запит о товарі.

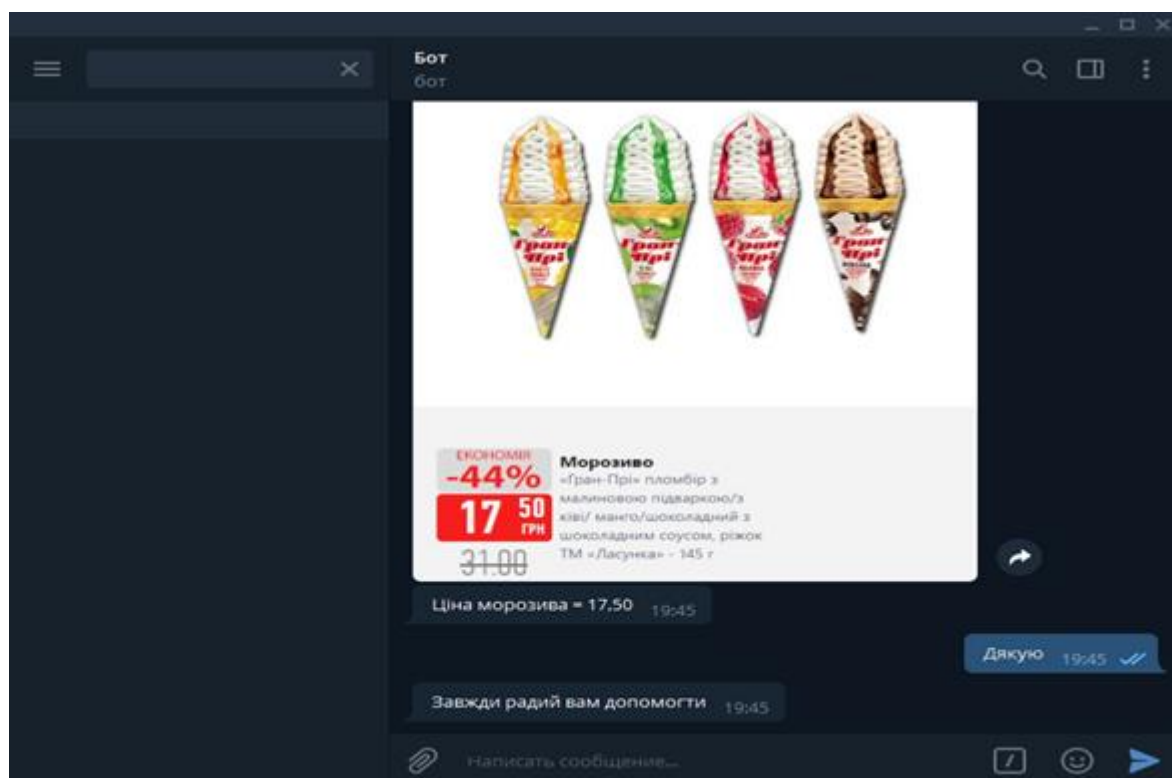


Рисунок 2.17 – Відповідь бота на подяку.

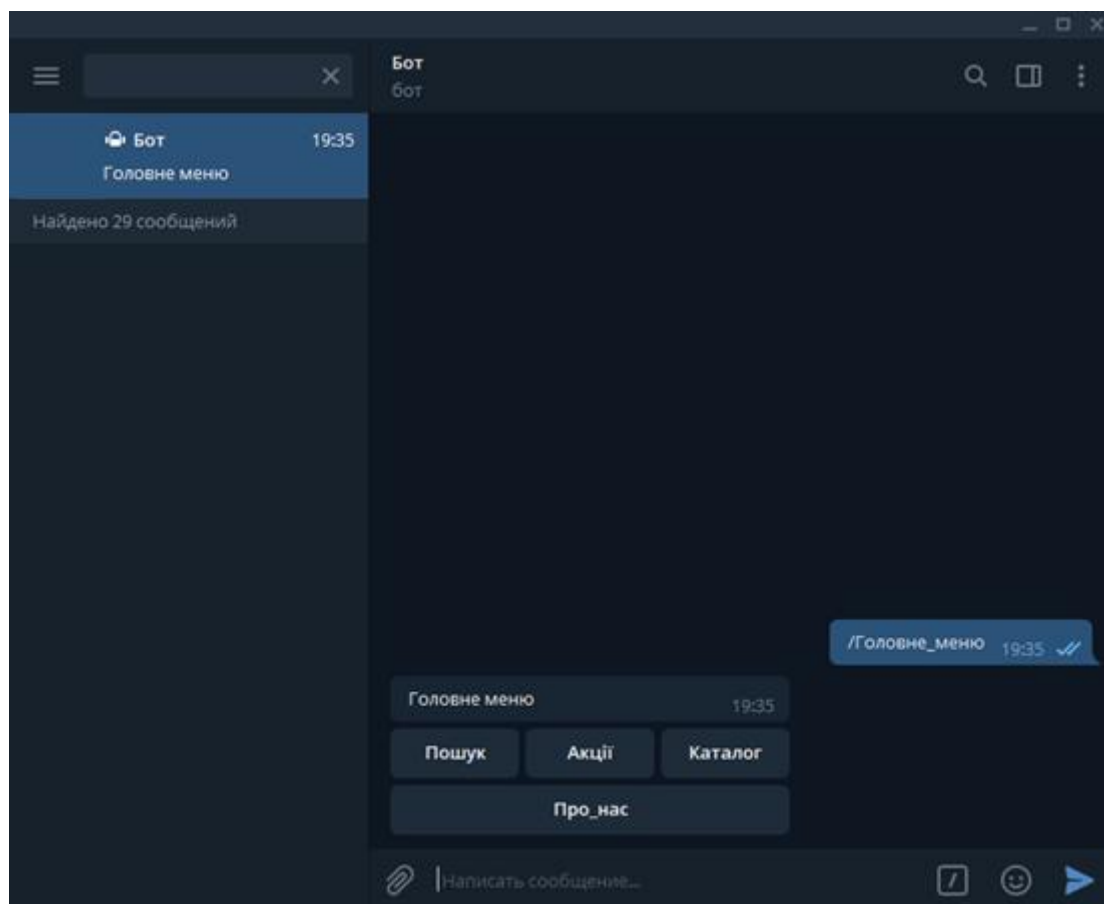


Рисунок 2.18 – Застосування команди повернення до головного меню

Таким чином, у ході розробки було створено повнофункціональне програмне забезпечення, що відповідає заявленим вимогам і практично реалізує підходи інженерії програмного забезпечення в контексті автоматизації маркетингової діяльності торговельного підприємства. Телеграм-бот дозволяє оперативно надавати користувачам інформацію про товари та спеціальні пропозиції, зменшуючи навантаження на персонал та оптимізуючи канали комунікації з клієнтами.

Проєкт є масштабованим та може бути адаптований до інших торговельних мереж або месенджерів з мінімальними витратами на модифікацію. Надалі можливе розширення функціоналу за рахунок впровадження інтелектуальних підсистем (наприклад, рекомендаційного модуля або аналітики поведінки користувачів), що відкриває перспективи подальшого розвитку ПЗ на базі вже реалізованого рішення.

3 ОХОРОНА ПРАЦІ

3.1 Вступ

Охорона праці - це система правових, соціально-економічних, організаційно-технічних заходів, а так само санітарно-гігієнічних і лікувально-профілактичних засобів, направлених на збереження здоров'я і працездатності людини в процесі праці.

За порушення законів і інших нормативно-правових актів про охорону праці, створення перешкод в діяльності посадовців органів державного нагляду за охороною праці, а також представників профспілок, їх організацій і об'єднань винні особи притягуються до дисциплінарної, адміністративної, матеріальної, кримінальної відповідальності згідно закону (стаття 44 Закону «Про охорону праці»).

3.2 Санітарно-гігієнічні вимоги до умов праці

Санітарно-гігієнічні вимоги до умов праці під час виконання роботи описують: нормативи з параметрів мікроклімату, освітлення приміщень, рівнів шуму, вібрації та електромагнітних випромінювань.

Мікроклімат виробничих приміщень характеризують температурою, вологістю та швидкістю руху повітря, а також інтенсивністю радіації, переважно в інфрачервоній та ультрафіолетовій областях спектру електромагнітних випромінювань. Параметри мікроклімату у приміщеннях повинні забезпечувати комфортне самопочуття організму. Тому у виробничих приміщеннях повинна бути надійна система кліматичного контролю.

Ще один важливий чинник, від якого залежать працездатність і здоров'я людини, – це освітлення. Світло регулює всі функції людського організму і впливає на психологічний стан і настрій, обмін речовин, гормональний фон і розумову активність. Найздоровіше освітлення забезпечує природне світло. Його ефективне використання можливе, якщо глибина приміщень не перевищує 6 м. Окрім того, хорошим вирішенням є скляні перегородки, що забезпечують зорову і звукову ізоляцію, але в той же час не перешкоджають проникненню природного світла.

Відносно вікон робоче місце повинно бути розміщено так, щоб природне світло було збоку, переважно з лівого та забезпечувати коефіцієнт природної освітленості не нижче 1,5 %. Освітленість за штучного освітлення в площині робочої поверхні має становити 300–500 Лк. Якщо таких значень освітленості досягнути не можна, то допускається локальне освітлення, але освітленість екрана не може перевищувати 300 Лк.

Робоче місце, обладнане ПК повинно бути розташоване так, щоб уникнути попадання в очі прямого світла. Щоб уникнути світлових відблисків від екрану та клавіатури необхідно використовувати комп'ютерне обладнання з матовою поверхнею. Для захисту очей від прямого сонячного світла чи джерел штучного освітлення необхідно застосовувати захисні жалюзі на вікнах.

Вимоги до рівнів шуму та вібрації. Шум часто є причиною зниження рівня працездатності, підвищення рівня загальної та професійної захворюваності, частоти виробничих травм. Шум як стрес-чинник є загальнобіологічним подразником, який негативно впливає на всі органи і системи організму. У разі тривалого систематичного впливу шуму може виникнути патологія з переважним ураженням слуху, центральної нервової серцево-судинної систем. Допустимі рівні звукового тиску у октавних смугах частот, еквівалентні рівні звуку на робочих місцях встановлені санітарними нормами виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, і для творчої та наукової роботи, навчання, не повинні перевищувати 50 дБА.

Вібрація приводить тіло і його структурні частини в коливний рух. Розрізняють поперечні, поздовжні і крутильні коливання. За впливом на людину вібрації ділять на місцеві і загальні. Загальні вібрації викликають коливання тіла людини, місцеві – лише окремі частини тіла. Тривала дія вібрацій на організм людини призводить до порушень в центральній нервовій та серцево-судинній системах, погіршує загальний стан людини – pojawiaються втома, головний біль тощо. Колективні та індивідуальні засоби і заходи захисту від шкідливого впливу виробничих чинників на здоров'я людини. Облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування

повітря. Надходження свіжого повітря регулюють, виходячи із таких умов (вказаний об'єм приміщення припадає на одне робоче місце з ПК):

- якщо об'єм приміщення 20 м³, то потрібно подати не менш як 30 м³/год повітря;
- якщо об'єм приміщення у межах від 20 до 40 м³, то потрібно подати не менш як 20 м³/год повітря;
- якщо об'єм приміщення становить понад 40 м³, допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

Захист від шуму та вібрацій. Усунення шуму в приміщенні є однією з найскладніших проблем, оскільки джерела шуму різноманітні й потребують комплексу заходів технічного, організаційного і медичного характеру на всіх стадіях проєктування, будівництва, експлуатації машин і устаткування. Відомі головні напрямки зменшення впливу шуму на організм людини:

- зменшення рівня шуму у джерелі виникнення, застосування раціональних конструкцій, нових матеріалів і технологічних процесів;
- звукоізоляція устаткування за допомогою глушників, резонаторів, кожухів, захисних конструкцій, оздоблення стін, стелі, підлоги тощо;
- використання засобів індивідуального захисту.

Заходи особистої гігієни на робочому місці (підтримання чистоти, миття лабораторного посуду, рук тощо). Заходи особистої гігієни на робочому місці передбачають щоденне вологе прибирання, утримання у чистоті робочого місця, наявність на робочому місці тільки необхідних для роботи засобів. На робочому місці необхідно дотримуватись вимог правил внутрішнього розпорядку, зокрема, заборонено приймати їжу, пити, курити та ін.

3.3 Електробезпека

Персональні комп'ютери, периферійні пристрої, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають відповідати класу зони, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів. Під час монтажу та експлуатації ліній електромережі необхідно повністю

унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію. Лінія електромережі для живлення персональних комп'ютерів і периферійних пристроїв виконується як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники. Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту.

У приміщенні, де одночасно експлуатуються понад п'ять персональних комп'ютерів і периферійних пристроїв, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Персональні комп'ютери і периферійні пристрої повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати персональні комп'ютери та периферійні пристрої до звичайної

двопровідної електромережі, в тому числі з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення персональних комп'ютерів та периферійних пристроїв потрібно виконувати за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12В та 42В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127В та 220В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12В та 42В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127В та 220В. Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах. Електромережу штепсельних розеток для живлення персональних комп'ютерів і периферійних пристроїв при розташуванні їх уздовж стін приміщення прокладають по підлозі поруч зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах, а також у пластикових коробах і пластмасових рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. При розміщенні в приміщенні до п'яти персональних комп'ютерів і периферійних пристроїв допускається прокладання трипровідникового захищеного проводу або кабелю в оболонці з негорючого чи важкогорючого матеріалу по периметру приміщення без металевих труб та гнучких металевих рукавів.

3.4 Пожежна безпека

Будинок, у якому знаходиться робочий кабінет по пожежній небезпеці будівельних конструкцій відноситься до категорії К1 (малопожежонебезпечні), оскільки тут присутні займисті (книги, документи, меблі, оргтехніка і т.д.) та тяжко-займисті речовини (різне устаткування і т.д.), що при взаємодії з вогнем можуть горіти без вибуху.

По конструктивних характеристиках будинок можна віднести до будинків з несучими і огорожувальними конструкціями із природних або штучних кам'яних матеріалів, бетону або залізобетону, де для перекриттів допускається використання дерев'яних конструкцій, захищених штукатуркою або тяжко-займистими

листовими, а також плитними матеріалами.

Причинами виникнення пожежі у будівлі можуть стати:

- несправності електропроводки, розеток і вимикачів які можуть привести до короткого замикання або пробією ізоляції;
- використання ушкоджених (несправних) електроприладів;
- виникнення пожежі внаслідок влучення блискавки в будинок;

В приміщенні є наступні засоби пожежогасіння: два вуглекислотні вогнегасники.

Додатково приміщення обладнано пожежною сигналізацією.

Під час виникнення пожежі евакуація відбувається відповідно до затвердженого плану евакуації. Пожежна профілактика являє собою комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей на запобіганні пожежі, обмеження поширення вогню, а також створення умов для успішного гасіння пожежі і недопущенню його подальшого поширення. Для профілактики пожежі надзвичайно важлива правильна оцінка пожежонебезпеки будинку, визначення небезпечних факторів і обґрунтування способів і засобів попередження пожежної небезпеки.

Таким чином після аналізу стану протипожежної безпеки було виявлено, що вжито не достатньо заходів для забезпечення відповідності вимогам нормативних документів.

3.5 Безпека під час роботи з персональним комп'ютером

Щодня перед початком роботи необхідно очищати монітор від пилу та інших забруднень. Після закінчення роботи персональний комп'ютер і периферійні пристрої повинні бути відключені від електричної мережі. У разі виникнення аварійної ситуації необхідно негайно відключити персональний комп'ютер і периферійні пристрої від електричної мережі.

Не допускається:

- виконувати обслуговування, ремонт та налагодження персонального комп'ютеру та периферійних пристроїв безпосередньо на робочому місці оператора;

– зберігати біля персонального комп'ютеру та периферійних пристроїв папір, будь-які носії інформації (диски, флешки тощо), запасні блоки, деталі тощо, якщо вони не використовуються для поточної роботи;

– відключати захисні пристрої, самочинно проводити зміни у конструкції та складі персонального комп'ютеру та периферійних пристроїв або їх технічне налагодження;

– працювати з персональним комп'ютером, у яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на моніторі тощо;

– працювати з матричним принтером за відсутності вібраційного килимка та зі знятою (піднятою) верхньою кришкою.

3.6 Розрахунки рівня шуму в приміщенні

Рівень шуму в приміщенні, розміри якого довжина 10 м, ширина 8 м, висота 5 м, складає 60 дБ А. Підлога у приміщенні – паркет, стіни та стеля - звичайна штукатурка. Визначимо зниження рівня шуму після акустичної обробки стін та стелі звукопоглинаючим матеріалом (коефіцієнт поглинання 0,9).

Зниження рівня шуму за рахунок акустичної обробки приміщення ΔL визначається за наступною формулою 4.1

$$\Delta L = 10 \lg (A_2/A_1), \quad (4.1)$$

де A_1 , A_2 – звукопоглинання приміщення до та після акустичної обробки, одиниць поглинання.

Звукопоглинання приміщення визначається за формулою 4.2

$$A = S \cdot \alpha, \quad (4.2)$$

де S – площа поверхні, m^2 ;

α – коефіцієнт поглинання матеріалу поверхні, одиниці поглинання.

Коефіцієнти поглинання матеріалів стін, стелі та підлоги визначаємо згідно стандартів. Коефіцієнти поглинання матеріалів стін (0,03), стелі (0,03) та підлоги (0,06).

Визначаємо за формулою 4.2 звукопоглинання приміщення до проведення обробки:

$$A_1 = 2 \cdot 10 \cdot 5 \cdot 0,03 + 2 \cdot 8 \cdot 5 \cdot 0,03 + 10 \cdot 8 \cdot 0,03 + 10 \cdot 8 \cdot 0,06 = 12,6 \text{ одиниць поглинання.}$$

Визначаємо за формулою 4.2 звукопоглинання приміщення після акустичної обробки (обробки стін та стелі):

$$A_2 = 2 \cdot 10 \cdot 5 \cdot 0,9 + 2 \cdot 8 \cdot 5 \cdot 0,9 + 10 \cdot 8 \cdot 0,9 + 10 \cdot 8 \cdot 0,06 = 238,8 \text{ одиниць поглинання.}$$

Зниження рівня шуму за формулою 4.1 складає

$$\Delta L = 10 \lg (238,8 / 12,6) = 12,8 \text{ дБ.}$$

Рівень шуму після обробки приміщення ($60 - 12,8 = 47,2$ дБ А) відповідає нормативним вимогам до приміщення з ПЕОМ.

3.7 Висновки

Проаналізувавши умови праці, можна підвести наступний підсумок.

На робочому місці добре організована система вентиляції та кондиціонування повітря. Стосовно пожежної безпеки, доцільно було б обладнати будівлю пожежною сигналізацією та автоматичною системою пожежогасіння, також необхідно звернути додаткову увагу на забезпечення додаткових засобів та заходів направлених на забезпечення електробезпеки, наприклад обладнати корпуси системних блоків додатковим заземленням для погашення статичного заряду.

Запропоновано покращити наявне штучне освітлення, так щоб величина освітленості при штучному освітленні люмінесцентними лампами була не нижче за 300 лк.

ВИСНОВКИ

Під час написання кваліфікаційної роботи було розглянуто історію становлення месенджерів, як сучасного засобу спілкування. Розглянуто ботів – програми які дозволяють автоматично спілкуватися з користувачами. Описано загальні принципи розробки сервісу для платформи Telegram та описано процес розробки бота.

На базі описаних принципів та із використанням описаних технологій було створено сервіс для спілкування із користувачем, який дозволяє виділяти ключові слова із його повідомлень та надавати по них розширену довідку з метою реклаमाції продукції компанії

Для створення сервісу на платформі Telegram обрано наступні засоби та інструменти: мову програмування Python разом із середовищем розробки PyCharm, використано бібліотеку Telegram.Bot, яка реалізує Bot API Telegram, нереляційну СКБД MongoDB.

Сервіс забезпечує такі основні функції і надає:

- відповідь наявними у системі модулями пошуку відповідей, насамперед – у наперед визначеній базі даних;;
- аналіз питання, заданого у текстовій формі природною мовою;
- можливість обробки природною мовою, якою формується запитання;
- відповідь через діалоговий інтерфейс чату на платформі Telegram;
- перспективу розширення системи через додання джерел, де відбувається пошук інформації для відповіді на поставлені питання;
- спроможність користувача оцінити релевантність наданої йому відповіді.

Розробку виконано у повному обсязі. Усі вимоги, наведенні у технічному завданні, виконані. Система, згідно з затвердженою методикою виконання, успішно протестована.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Github pyTelegramBotAPI інформація про бібліотеку API. URL: <https://github.com/eternnoir/pyTelegramBotAPI/#general-api-documentation> (дата звернення 05.06.2025).
2. Most popular global mobile messenger apps URL: <https://www.statista.com/statistics/258749/most-popular-globalmobile-messenger-apps/> (дата звернення 06.06.2025).
3. Python – Wikipedia. Основна інформація: URL: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (дата звернення 07.06.2025).
4. Наглядні приклади як створювати та працювати за телеграм ботом на мові Python. URL: <https://core.telegram.org/bots/api> (дата звернення 08.06.2025).
5. Наглядні приклади як створювати та працювати за телеграм ботом на мові Python URL: <https://realpython.com/python-telegram-bot/> (дата звернення 09.06.2025).
6. «Learning Python: Powerful Object-Oriented Programming» Марк Лутц URL: <https://learning.oreilly.com/library/view/learning-python-5th/9781449355722/>. Ст. 149-160 (дата звернення 10.06.2025) .
7. «Learning Python: Powerful Object-Oriented Programming» Марк Лутц URL: <https://learning.oreilly.com/library/view/learning-python-5th/9781449355722/>. Ст. 346-350 (дата звернення 11.06.2025).
8. «Learning Python: Powerful Object-Oriented Programming» Марк Лутц URL: <https://learning.oreilly.com/library/view/learning-python-5th/9781449355722/>. Ст. 394-397 (дата звернення 12.06.2025).
9. <https://stackoverflow.com/questions/tagged/python-telegram-bot> (дата звернення 14.06.2025).
10. MongoDB Atlas Documentation URL: <https://www.mongodb.com/docs/atlas/> (дата звернення 14.06.2025)

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

1. Підстава для розробки

Підставою для розробки програмного продукту «Telegram-бот для реклами продукції компанії АТБ» є рішення маркетингової служби АТБ щодо впровадження інноваційних каналів комунікації. Згідно з повідомленнями, у рамках стратегії діджиталізації та підвищення якості обслуговування АТБ-Маркет у 2019 році запустив чат-бот «АТБОТ» у месенджерах Viber і Telegram. Аналіз подібних рішень та потреб бізнесу обумовлює доцільність розробки спеціалізованого бота для рекламних розсилок.

2. Призначення та область застосування

Бот призначено для автоматизації маркетингової взаємодії з клієнтами мережі АТБ. Його мета – оперативно інформувати зареєстрованих користувачів Telegram про актуальні акції, спеціальні пропозиції і нові товари, а також збирати зворотний зв'язок (опитування, відгуки). Система підвищить ефективність маркетингу та полегшить комунікацію: чат-боти є «ефективним, доступним та простим способом автоматизувати процес роботи з клієнтами». Вони дозволяють обслуговувати більше клієнтів з меншими витратами, доступні 24/7 та надають миттєві відповіді на запити.

Переваги проекту:

- Автоматизація взаємодії з клієнтами: зменшує ручну роботу маркетологів і працівників call-центру, забезпечує швидку розсилку інформації.
- Швидкість і доступність: бот працює цілодобово, миттєво відповідає на запити, що покращує досвід клієнтів.
- Покращення маркетингових показників: підвищується охоплення аудиторії та ефективність рекламних кампаній за рахунок персоналізації повідомлень та швидкого оповіщення цільових груп.

Область застосування: бот використовуватиметься маркетинговою службою АТБ у спілкуванні з українськими споживачами продукції. Це програмний продукт для розсилки повідомлень у Telegram і збору статистики, не пов'язаний з критичними процесами обробки фінансових даних.

3. Вимоги до функціональних характеристик

Склад основних функцій програмного продукту:

- Реєстрація користувачів: авторизація за Telegram-ID, управління реєстром підписників (запис/відписки від розсилки).
- Інформування про акції: автоматична і/або за запитом публікація переліку актуальних акційних товарів та інформації про знижки.
- Розсилка повідомлень: заплановані та термінові повідомлення (текстові та мультимедійні – зображення, лого тощо) про акції, нові товари чи події.
- Запит інформації: можливість для користувача надіслати запит (наприклад, «дізнатися про акції в конкретному місті» або «пошук найближчого магазину мережі за геолокацією»).
- Збір зворотного зв'язку: функції опитування клієнтів, прийому скарг та пропозицій без прямого підключення до оператора.
- Адміністрування контенту: інтерфейс (контрольні команди) для модераторів/адміністраторів – формування та управління списками розсилки, складання повідомлень, перегляд статистики (кількість прочитань, відписок тощо). Функції реалізуються відповідно до очікуваного обсягу оброблюваних даних та часових характеристик (швидкість реакції бота на команду, інтервали розсилки тощо).

4. Вимоги до надійності

Програма повинна забезпечувати безперебійну роботу протягом розрахункового часу (не менше 99% часу на місячному інтервалі). Необхідно передбачити механізми обробки збоїв і помилок: запис журналу помилок, контроль коректності обробки повідомлень, захист від несанкціонованого доступу. Після відмови системи має здійснюватися автоматичне відновлення (наприклад, перезапуск сервісів). Час відновлення після критичного збою не повинен перевищувати 30 хвилин. Усі дані (реєстри підписників, повідомлення) мають зберігатися і резервуватися для запобігання втратам.

5. Умови експлуатації

Програмний продукт призначено для роботи в стандартних офісних умовах.

Фактори навколишнього середовища: температура від +10 до +30 °C, відносна вологість до 80% (значення звичайне для приміщень з кондиціонуванням). Система вимагає безперебійного підключення до мережі Інтернет. Електропостачання – типове офісне (230 В, 50 Гц) зі стабілізованою напругою. Для обслуговування потрібно 1–2 кваліфікованих ІТ-спеціалісти (адміністратор та, за потреби, програміст) з розумінням месенджерів і серверів.

6. Вимоги до складу та параметрів технічних засобів

Для роботи бота необхідний сервер або хмарний хостинг із такими мінімальними характеристиками: процесор $\geq 2,0$ GHz (CPU 4 ядра), оперативна пам'ять ≥ 4 GB, дисковий простір ≥ 20 GB. Система взаємодіє з Telegram Bot API і мережевими серверами, тому потрібне стабільне підключення до Інтернету зі швидкістю ≥ 10 Mbit/s. Операційна система сервера – Linux (рекомендовано Ubuntu Server 20.04+) або Windows Server 2016/2019. Для розробки/підтримки можуть використовуватися робочі станції на базі Windows, Linux або macOS з встановленим інструментарієм (напр., Python 3.x, Node.js, браузером). Для користувача достатньо смартфона чи ПК з Telegram. Всі технічні засоби мають відповідати вимогам обраного програмного середовища і бути сумісними з інтернет-протоколами.

7. Вимоги до інформаційної та програмної сумісності

Програма повинна використовувати офіційний Telegram Bot API (HTTP/HTTPS, формат даних JSON). Кодування тексту – UTF-8. Необхідно забезпечити сумісність з ОС Windows, Linux і macOS (для серверної та клієнтської частини). База даних (MySQL/PostgreSQL) – відповідні драйвери повинні коректно працювати у вибраній ОС. Розробка має бути написана на сучасних мовах програмування (наприклад, Python 3.x або JavaScript/Node.js) і не залежати від специфічного обладнання. За потреби слід передбачити інтеграцію з іншими корпоративними системами (CRM, email-розсилками) через REST-API або інші стандартні протоколи. При необхідності – забезпечити захист інформації (наприклад, SSL-шифрування каналу передачі).

ДОДАТОК Б – Код розробки

Лістинг Б.1– Лістинг програми

```
# Імпортуємо необхідні бібліотеки для роботи з Telegram API
import telebot
from telebot import types

# Ініціалізуємо Telegram-бота за допомогою токена
bot = telebot.TeleBot('913493123:AAG4ZWLAoZZGET4y7EnjRVqAr0ppm_iXUAo')
img1 = open('Морозиво.PNG', 'rb')
img2 = open('Чіпси.PNG', 'rb')
img3 = open('Кефір.PNG', 'rb')

# Обробка команди /start - виводить привітальне повідомлення і головне меню
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привіт, я бот "АТБ"')
    # Створюємо клавіатуру з чотирма кнопками
    keyboards = types.InlineKeyboardMarkup()

    callback_button = types.InlineKeyboardButton(text="Пошук",
callback_data="test")

    callback_button2 = types.InlineKeyboardButton(text="Акції",
callback_data="test2")

    callback_button3 = types.InlineKeyboardButton(text="Каталог",
callback_data="test3")

    callback_button4 = types.InlineKeyboardButton(text="Про_нас",
callback_data="test4")

    keyboards.add(callback_button, callback_button2, callback_button3,
callback_button4)
    # Відправляємо повідомлення з клавіатурою
    bot.send_message(message.chat.id, "Головне меню", reply_markup=keyboards)

# Обробка команди для повторного виклику головного меню
@bot.message_handler(commands=['command5', 'Головне_меню', 'головне_меню'])
def menu(message):
    keyboards = types.InlineKeyboardMarkup()

    callback_button = types.InlineKeyboardButton(text="Пошук",
callback_data="test")

    callback_button2 = types.InlineKeyboardButton(text="Акції",
callback_data="test2")

    callback_button3 = types.InlineKeyboardButton(text="Каталог",
callback_data="test3")

    callback_button4 = types.InlineKeyboardButton(text="Про_нас",
callback_data="test4")

    callback_button5 = types.InlineKeyboardButton(text="Головне меню",
callback_data="test5")

    keyboards.add(callback_button, callback_button2, callback_button3,
callback_button4)

    bot.send_message(message.chat.id, "Головне меню", reply_markup=keyboards)

# Обробка кнопки «Пошук» - перенаправляє на сайт АТБ з пошуком
```

```

@bot.message_handler(commands=['command1', 'Пошук', 'пошук'])
def default_test(message):
    keyboards = types.InlineKeyboardMarkup()
    keyboard = types.InlineKeyboardMarkup()

    url_button = types.InlineKeyboardButton(text="Перейти на сайт з пошуковим
рядком",

url="https://www.atbmarket.com/search")
    callback_button5 = types.InlineKeyboardButton(text="До говного меню",
callback_data="test5")
    keyboards.add(callback_button5)
    keyboard.add(url_button)
    bot.send_message(message.chat.id, "Натисніть щоб перейти на сайт",
reply_markup=keyboard)

@bot.message_handler(commands=['command2', 'Каталог'])
def default_test(message):
    keyboard = types.InlineKeyboardMarkup()
    keyboards = types.InlineKeyboardMarkup()

    url_button = types.InlineKeyboardButton(text="Перейти на сайт з каталогом
магазину",

url="https://www.atbmarket.com/trademark/goods")
    callback_button5 = types.InlineKeyboardButton(text="До говного меню",
callback_data="test5")
    keyboard.add(url_button)
    keyboards.add(callback_button5)
    bot.send_message(message.chat.id, "Натисніть щоб перейти на сайт",
reply_markup=keyboard)

@bot.message_handler(commands=['command3', 'Акції'])
def default_test(message):
    keyboard = types.InlineKeyboardMarkup()

    url_button = types.InlineKeyboardButton(text="Перейти на сайт з акціями
на тиждень",

url="https://www.atbmarket.com/hot/akcii/7day/")
    keyboard.add(url_button)
    bot.send_message(message.chat.id, "Натисніть щоб перейти на сайт",
reply_markup=keyboard)

@bot.callback_query_handler(func=lambda call: True)
def callback_inline(call):
    if call.message:

        if call.data == "test":

            bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,

                                text="https://www.atbmarket.com/search")

            elif call.data == "test3":

                bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,

```

```

text="https://www.atbmarket.com/trademark/goods/")
    elif call.data == "test2":

        bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,

text="https://www.atbmarket.com/hot/akcii/economy/")
    elif call.data == "test4":

        bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,

                                text="https://www.atbmarket.com/about/o-
kompanii")
    elif call.data == "test5":
        bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
                                text="/Головне_меню")

@bot.message_handler(commands=['command4', 'Про_нас'])
def default_test(message):
    keyboard = types.InlineKeyboardMarkup()

    url_button = types.InlineKeyboardButton(text="Перейти на сайт з
додатковою інформацією про нас",

url="https://www.atbmarket.com/about/o-kompanii")
    keyboard.add(url_button)
    bot.send_message(message.chat.id, "Натисніть щоб перейти на сайт",
reply_markup=keyboard)

@bot.message_handler(commands=['command5', 'Головне_меню', 'голове_меню'])
def default_test(message):
    keyboards = types.InlineKeyboardMarkup()

    callback_button = types.InlineKeyboardButton(text="Пошук",
callback_data="test")

    callback_button2 = types.InlineKeyboardButton(text="Акції",
callback_data="test2")

    callback_button3 = types.InlineKeyboardButton(text="Каталог",
callback_data="test3")

    callback_button4 = types.InlineKeyboardButton(text="Про_нас",
callback_data="test4")

    keyboards.add(callback_button, callback_button2, callback_button3,
callback_button4)

    bot.send_message(message.chat.id, "Головне меню", reply_markup=keyboards)

@bot.message_handler(content_types=['text'])
def get_text_messages(message):
    if message.text == "Вартість морозива гран-прі":
        bot.send_photo(message.chat.id, img1)
        bot.send_message(message.from_user.id, "Ціна морозива = 17,50")
    elif message.text == "Вартість чіпсів lays":
        bot.send_photo(message.chat.id, img2)
        bot.send_message(message.from_user.id, "Ціна чіпсів = 17,90")

```

```

elif message.text == "Вартість кефіру галичина":
    bot.send_photo(message.chat.id, img3)
    bot.send_message(message.from_user.id, "Ціна кефіру = 15,90")
elif message.text == "Бонусна картка":
    bot.send_message(message.from_user.id, "Нажаль, на даний момент бот
не володіє інформацією про бонусну "
                                "програму. ")
elif message.text == "Морозиво гран-прі зі смаком ківі":
    bot.send_message(message.from_user.id, "Нажаль, бот не володіє
інформацією, що до наявності товару на полицях ")
elif message.text == "Привіт":
    bot.send_message(message.from_user.id, "Доброго дня, чим я можу вам
допомогти?")
elif message.text == "Дякую":
    bot.send_message(message.from_user.id, "Завжди радий вам допомогти")

if __name__ == '__main__':
    bot.polling(none_stop=True)

```

Лістинг Б.2– Модуль взаємодії з базою даних передзаписаних даних

```

import pymongo
class PymongoManager(object):
    def __init__(self, host='localhost', port=27017, usr=None, pwd=None,
db_auth=None, db='QADiploma'):
        mongo_connection_uri =
and f'mongodb://{usr}:{pwd}@{host}:{port}/{db}?authSource={db_auth}' if (usr
pwd and db and db_auth) else f'mongodb://{host}:{port}'
self.client = pymongo.MongoClient(mongo_connection_uri)
self.db = self.client[db] if db else None

```

Лістинг Б.3– Модуль вибору відповіді

```

import csv

import pathlib

from deeppavlov.agents.default_agent.default_agent
import DefaultAgent
from
deeppavlov.agents.processors.highest_confidence_selector
import
HighestConfidenceSelector
from deeppavlov.contrib.skills.similarity_matching_skill
import
SimilarityMatchingSkill
from deeppavlov.skills.pattern_matching_skill import
PatternMatchingSkill

from pymongo_manager import PymongoManager
from timer import timer

class QADB(object):

```

```

QA_DATA_PATH = './data/questions_answers.csv'
FAQ_MODEL_PATH = './data/qa_model'

def __init__(self, force_retrain=False):
    self.mongo_manager = PymongoManager()
    self.agent = None
    self._prepare(force_retrain=force_retrain)

def _prepare(self, force_retrain):
    hello = self._create_hello_pms()
    bye = self._create_bye_pms()
    fallback = self._create_fallback_pms()
    faq = self._get_model(force_retrain=force_retrain)
    self.agent = DefaultAgent([hello, bye, faq, fallback],
                               skills_selector=HighestConfidenceSelector())
    qs = ['what is the size of the int variable in C', 'how
to create an
    array in C', 'how to check if character is digit in C',
'character string']
    questions = qs.copy()

    from keywords_extractor import KeywordsExtractor
    ke = KeywordsExtractor()
    for i in range(len(qs)):
        kw = ke.extract_keywords(qs[i])
        print(qs[i], ': ', kw)
        qs[i] = " ".join(kw.keys())
    print(qs)
    print(questions)

    print(faq(qs, [], []))
    print(faq(questions, [], []))
    # print(hello(qs, [], []))
    # print(bye(qs, [], []))
    # print(fallback(qs, [], []))
    # print(faq(qs, [], []))

def _create_csv_from_mongodb(self):
    """
    Helper for _train_model().
    """
    answers = self.mongo_manager.db['answers'].find()

```

```

        questions_answers = []
        for a in answers:
            qa = {
                'Question':
                    self.mongo_manager.db['questions'].find_one({'_id':
a['qid']})['body'],
                'Answer': f'{a["body"]}',
            }
            if 'additional_links' in a:
                qa['Answer'] = f'{qa["Answer"]}\n\nAdditional links:\n'
                for al in a['additional_links']:
                    qa['Answer'] = f'{qa["Answer"]}{al}\n'
                questions_answers.append(qa)

        with open(self.QA_DATA_PATH, 'w', newline='',
encoding='utf-8') as
            csvfile:
                fieldnames = ['Question', 'Answer']
                writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
                writer.writeheader()
                for qa in questions_answers:
                    writer.writerow(qa)

    @timer
    def _train_model(self):
        if not pathlib.Path(self.QA_DATA_PATH).exists():
            self._create_csv_from_mongodb()
        return SimilarityMatchingSkill(
            data_path=self.QA_DATA_PATH,
            x_col_name='Question',
            y_col_name='Answer',
            save_load_path=self.FAQ_MODEL_PATH,
            config_type='tfidf_autofaq',
            edit_dict={},
            train=True
        )

    @timer
    def _load_model(self):
        return
SimilarityMatchingSkill(save_load_path=self.FAQ_MODEL_PATH,
                        train=False)

```

```

def _get_model(self, force_retrain):
    model_path = pathlib.Path(self.FAQ_MODEL_PATH)
    if (not model_path.exists()) or (force_retrain is True):
        return self._train_model()
    else:

        return self._load_model()

    @staticmethod
    def _create_hello_pms():
        return PatternMatchingSkill(
            responses=[
                'Hello, my friend. Stay awhile and listen...',
                'Greetings, good master! Welcome to the Chat of the
Rising
                Sun.',
                'The warmth of life has entered my chat...',
                'Yo dawg',
                'Hello',
                'Hi',
                'Sup',
                'I greet you.',
            ],
            patterns=['\\b(?:hi|hello|hola|good day|good
morning|good
evening|yo|sup)\\b', ],
            regex=True,
            default_confidence=0.75
        )

    @staticmethod
    def _create_bye_pms():
        return PatternMatchingSkill(
            responses=[
                'Goodbye',
                'Bye',
                'Cya',
                'See you later',
                'See ya',
            ],
            patterns=['\\b(?:bye|good bye|goodbye|cya)\\b', ],
            regex=True,

```

```
default_confidence=0.75
)

@staticmethod
def _create_fallback_pms():
    return PatternMatchingSkill(
        ['The university is immense... I cannot find an answer
to this.'],
        default_confidence=0.4
    )

if __name__ == '__main__':
    db = QADB()
```

ДОДАТОК В– ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Назва ПЗ:

Telegram-бот для реклами продукції АТБ

Мета розробки:

Створення Telegram-бота, який дозволяє користувачам отримувати актуальну інформацію про акції, наявність товарів, ціни та інформацію про компанію АТБ у зручному чат-форматі. ПЗ має автоматизувати спілкування з клієнтами та покращити взаємодію з ними через месенджер Telegram.

Загальна характеристика:

- Тип: кросплатформенне веб-орієнтоване програмне забезпечення
- Мова програмування: Python [2]
- Бібліотеки: python-telegram-bot, pymongo, dotenv
- СУБД: MongoDB
- Інтерфейс: чат-інтерфейс Telegram з кнопками навігації

Основні функції:

Для користувача:

- Ініціалізація бота та виклик головного меню
- Перегляд каталогу товарів
- Отримання інформації про акції
- Пошук товару по назві
- Отримання інформації "Про нас"

Для адміністратора:

- Завантаження або оновлення бази товарів
- Перезапуск бота
- Додавання нових товарів

Основні компоненти системи:

- Telegram Bot API – забезпечує взаємодію бота з користувачами
- Серверна частина на Python – обробляє запити користувачів та формує відповіді [6]
- База даних MongoDB – зберігає інформацію про товари, категорії, магазини та користувачів

Вхідні дані:

- Текстові запити користувача
- Натискання кнопок меню
- Інформація про товари, що вносяться адміністратором

Вихідні дані:

- Повідомлення з результатами пошуку товарів
- Перелік акційних пропозицій
- Інформація про компанію

Архітектура:

Бот реалізований за принципом клієнт-серверної архітектури. Клієнт — користувач месенджера Telegram, надсилає запити, серверна частина (Python скрипт) приймає їх через Webhook або Long Polling, обробляє і звертається до бази MongoDB для отримання відповідей.

Інтерфейс користувача:

Інтерфейс реалізовано з допомогою кнопкового меню в Telegram. Меню містить кнопки: "Каталог", "Пошук", "Акції", "Про нас". Також є підказки користувачу у вигляді відповідей бота.

Висновок:

Розроблене ПЗ є зручним у використанні і дозволяє користувачам швидко отримувати необхідну інформацію. Телеграм-бот значно покращує процес комунікації між компанією АТБ та її клієнтами, зменшуючи навантаження на службу підтримки і сприяє автоматизації маркетингових процесів.

ДОДАТОК Г – Керівництво користувачеві

Г.1 Ініціалізація бота

Для початку роботи з чат – ботом потрібно увійти в додаток Telegram та пройти авторизацію, далі увести у пошукову стрічку назву бота, а також ініціалізувати бота Г.1-2.

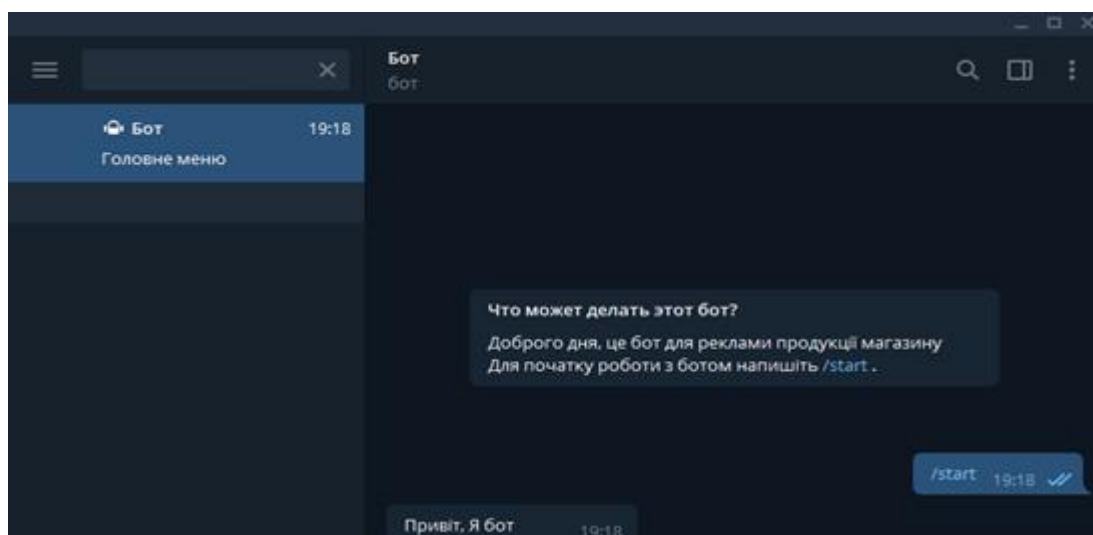
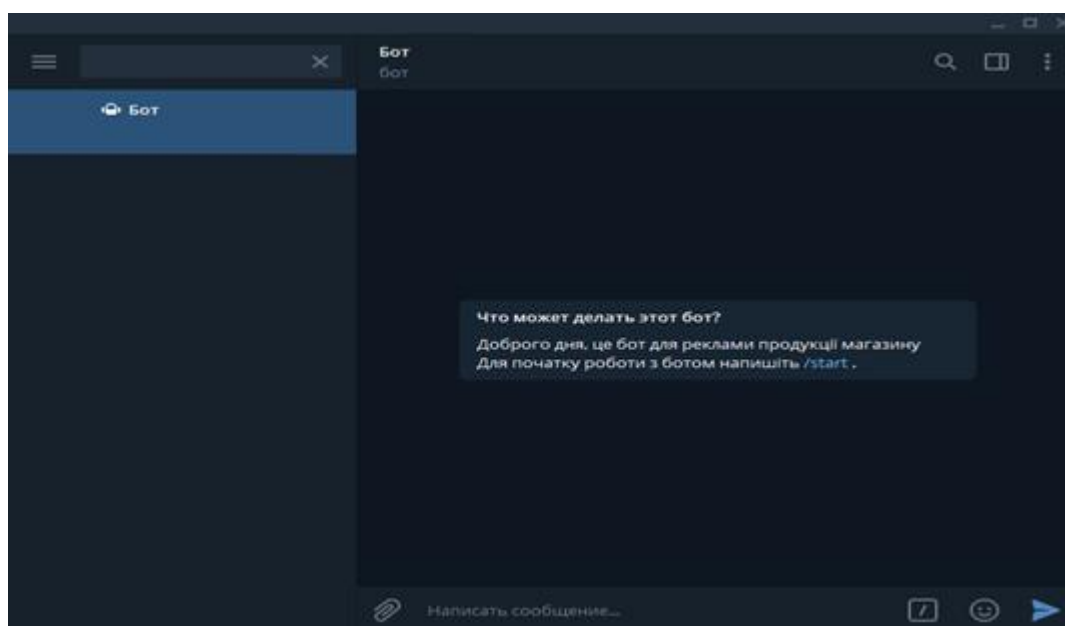


Рисунок Г.1-2 – Чат – бот до та після ініціалізації

Г.1.2 Екран чат - боту

При ініціалізації користувач отримує доступ до головного меню боту. Для проведення операцій потрібно натиснути на кнопку меню та обрати потрібний

варіант використання, які представлені на рисунку Г.3.

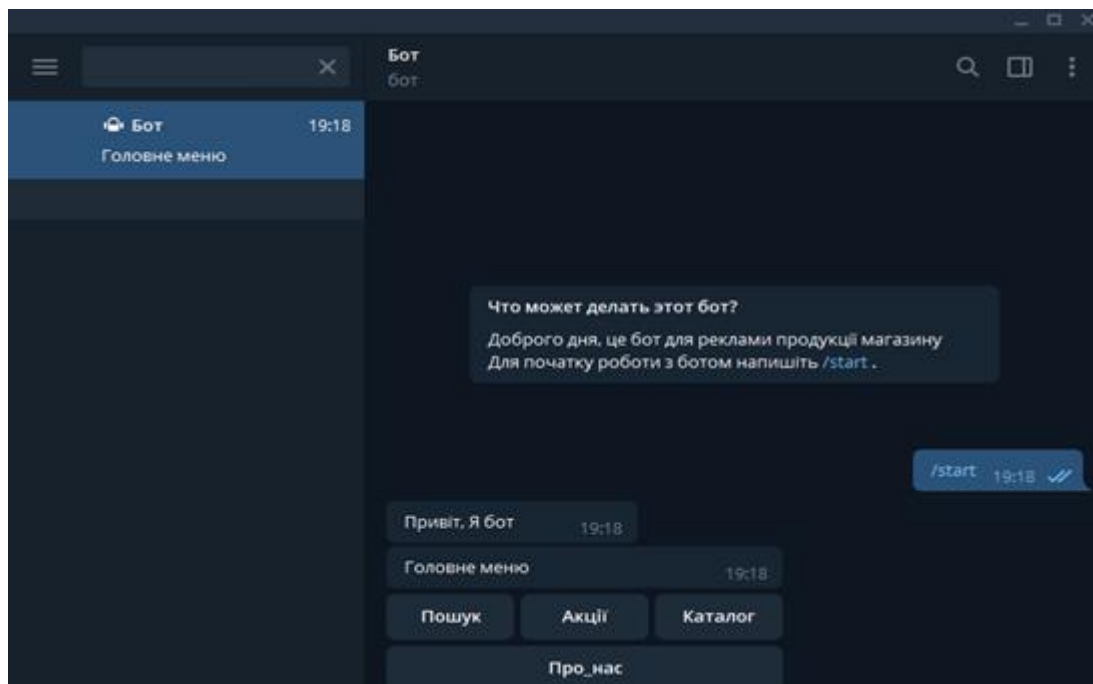
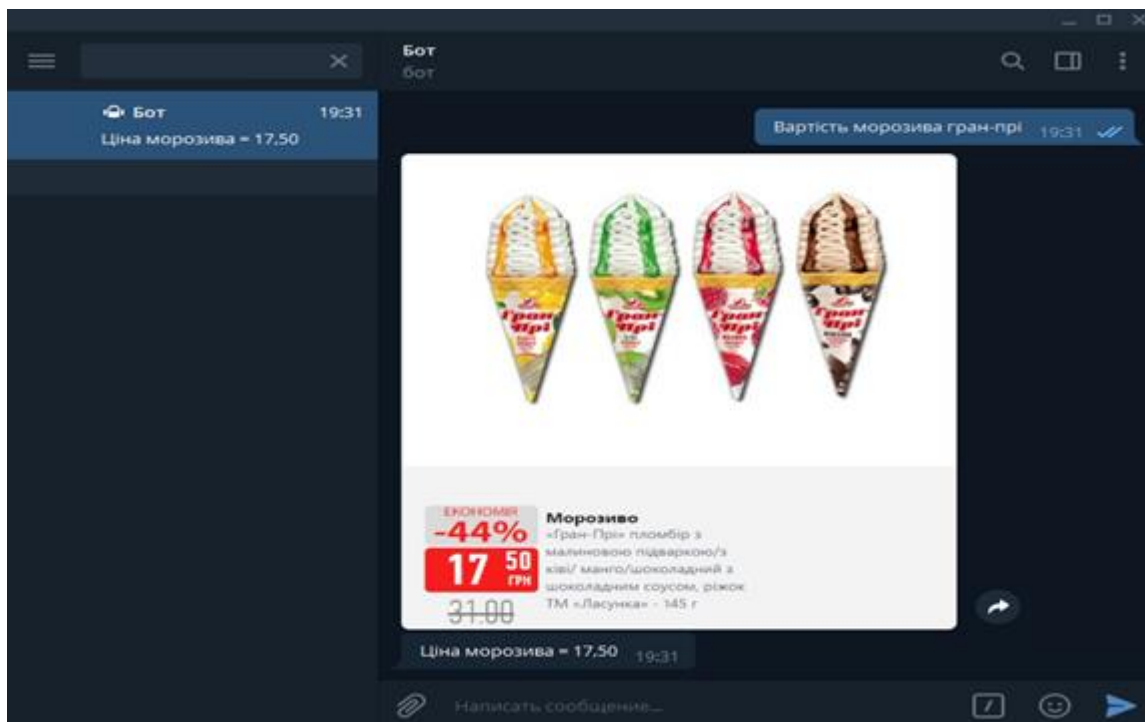
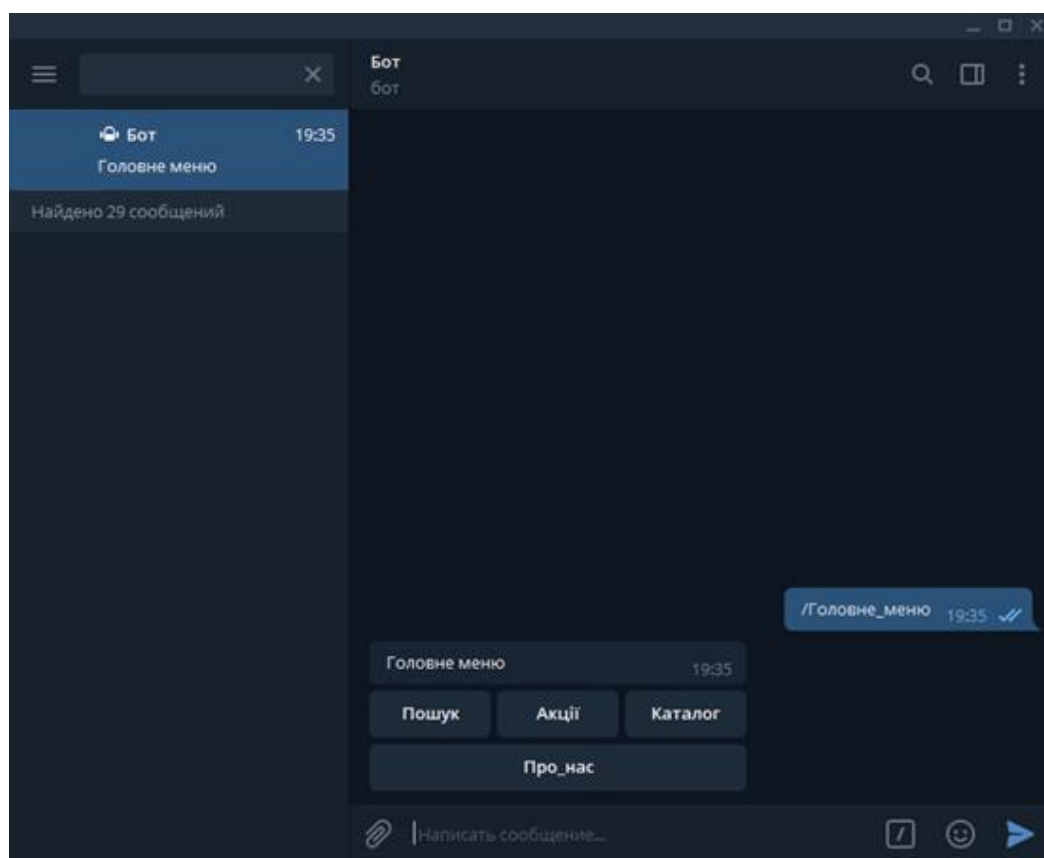
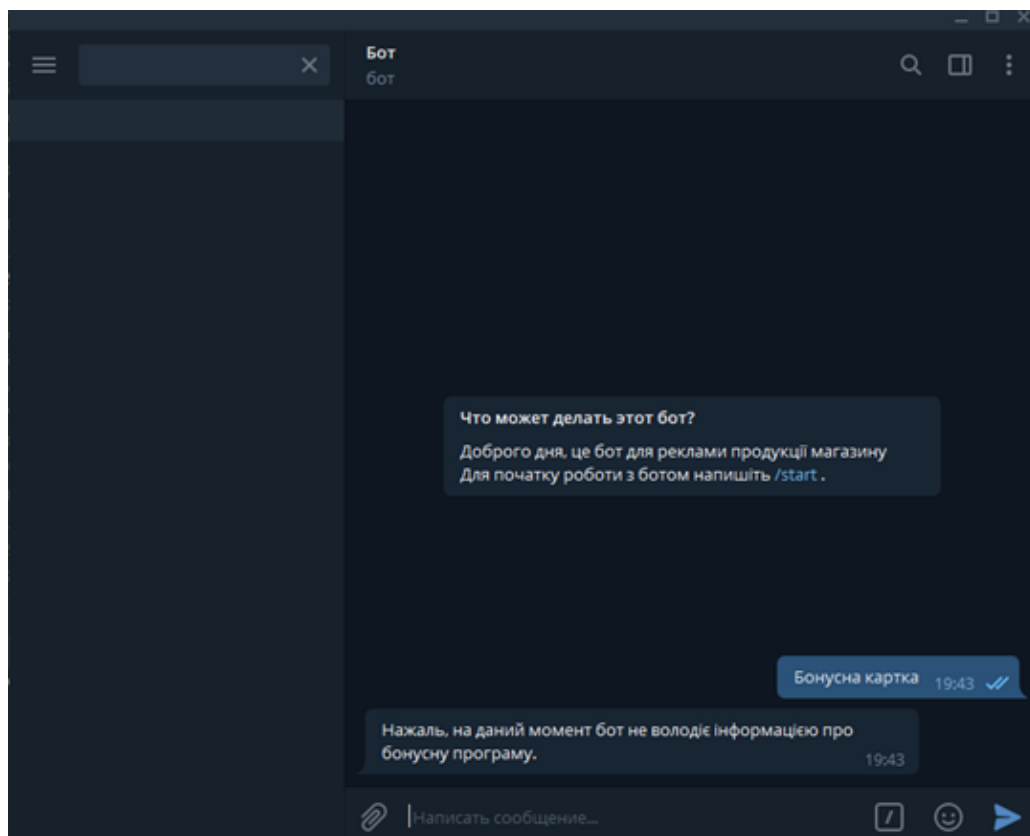


Рисунок Г.3 – Меню з варіантами використання чат - бота

На екрані чат - боту ми можемо написати повідомлення або обрати необхідну нам функцію.

Використання деяких функцій чат – бота виглядають таким чином як на рисунках Г.4-7.





Рисунки Г.4-7 – Використання функції «Пошук», запит інформації про бонусну картку, та функція повернення до головного меню

ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБОВУВАНЬ ПЗ

Д.1. Мета випробувань

Метою випробувань є комплексна перевірка відповідності розробленого Телеграм-бота вимогам, визначеним у технічному завданні. Тестування спрямовується на підтвердження:

- правильність реалізації функціональних можливостей (меню, пошук, акції, службові команди);
- стійкість до помилок і непередбачених дій користувача;
- ефективність роботи в умовах навантаження;
- відповідність внутрішньої логіки реалізації обраним технічним підходам

Окремим завданням є досягнення високого рівня покриття коду під час структурного тестування, що гарантує надійність роботи системи у реальному використанні.

Д.2. Методика випробувань

Методика випробувань ґрунтується на поєднанні функціонального тестування («чорна скринька») та структурного тестування («біла скринька»). Такий підхід забезпечує як зовнішню, так і внутрішню верифікацію роботи програми.

Д.2.1 Функціональні методи:

1. Позитивне тестування — перевірка відповідності результатів очікуваним при коректних запитах.
2. Негативне тестування — реакція на некоректні дані: порожні запити, сторонні символи.
3. Граничне тестування — дії на межі допустимих значень (запити з 1 символом, масові запити).
4. Регресійне тестування — перевірка, що зміни в коді не вплинули на існуючі функції.

Ці методи дозволяють охопити найбільш імовірні сценарії взаємодії користувача з ботом, а також ситуації, які можуть призвести до помилок.

Д.2.2 Структурні методи:

Тестування виконується із доступом до коду Телеграм-бота, що дозволяє реалізувати такі критерії:

- Покриття операторів (C0) — кожна команда в коді була виконана щонайменше один раз.
- Покриття гілок (C1) — перевірено виконання логічних умов у обох напрямках.
- Покриття умов — окремо перевірено частини складених логічних виразів.
- Покриття шляхів (C2) — зафіксовано проходження основних лінійно-незалежних логічних шляхів програми.

Ці критерії забезпечують глибоку перевірку реалізованої структури системи, включаючи ті частини коду, які не активуються під час типового використання.

Д.3. Середовище тестування

Тестування Телеграм-бота проводиться в умовах, наближених до експлуатаційних. Використовуються такі інструменти:

- Telegram Client — офіційний клієнт для Android та Web.
- База даних — MongoDB Atlas, з тестовою колекцією товарів і акцій.
- Хостинг — PythonAnywhere, що дозволяє використання webhook.
- Середовище розробки — PyCharm 2023, зі вбудованим дебагером.
- Логування — внутрішній лог-файл, журнал запитів, консолі debug.

Всі випробування проводяться як від імені звичайного користувача, так і адміністратора.

Д.4. Побудова тестів

Для кожного функціонального блоку бота формується перелік тестових сценаріїв, що охоплюють типові, нетипові та критичні ситуації. Структура тесту включає:

- вхідні дані (команда або повідомлення користувача);
- умови виконання (контекст, стан БД, авторизація);
- очікуваний результат (повідомлення, меню, відсутність помилок);
- фактичний результат (фіксується вручну або логами);
- висновок про результат: тест пройдено / не пройдено.

Структурне тестування реалізується на основі аналізу коду та застосування логічних умов у вигляді покриття if-else, циклів, винятків.

Д.5. Приклади тестових сценаріїв

Нижче наводиться таблиця з тестовими сценаріями, які охоплюють найбільш критичні логічні гілки та найтиповіші дії користувача.

№ п/п	Тестова дія	Очікуваний результат	Результат
1	Повна назва товару (наприклад, «Молоко 2,5%»)	Бот виводить усю інформацію про товар з БД	Пройдено
2	Часткова назва (наприклад, «кеф»)	Бот повертає релевантні результати	Пройдено
3	Введення лише символів або порожнього рядка	Бот не виконує пошук, повертає повідомлення	Пройдено
4	Внутрішній код: <code>if query and len(query) > 2</code>	Перевірено всі гілки та умови	Пройдено
5	Повторне натискання «Акції»	Бот стабільно реагує, не дублює помилки	Пройдено
6	Адмін-команда <code>/admin_reset</code> з дозволим ID	Система перезапускає бота, блокує сторонніх	Пройдено
7	Команда <code>/start</code>	Бот надсилає вітання і меню	Пройдено
8	10 одночасних запитів	Всі оброблені, час відповіді < 2 сек	Пройдено
9	Запит на неіснуючий товар	Ввічлива відповідь про відсутність товару	Пройдено
10	Кнопка «Про нас»	Виводиться інформація про компанію	Пройдено

Таблиця Д.1 – Тестові випадки перевірки функціональності чат-бота

Д.6. Результати випробувань

Загалом реалізується 45 тестових сценаріїв, з яких 42 проходять успішно з першого запуску. Три тести фіксують некритичні проблеми з відображенням кнопок, які усуваються після внесення змін у шаблон повідомлень.

Основні показники:

- Покриття операторів — 96%
- Покриття гілок — 87%
- Покриття умов — 91%
- Покриття шляхів — близько 80%
- Продуктивність при навантаженні — стабільна

Проведені випробування підтверджують, що Телеграм-бот повністю готовий до впровадження в експлуатацію та забезпечує високий рівень якості, стабільності та масштабованості.