

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації  
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук  
(повна назва кафедри (предметної, циклової комісії))

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія створення чат-бота для  
розміщення об'яв для товарів та послуг»

Виконав: студент 2-го курсу, групи 1КН-22м  
спеціальності 122 «Комп'ютерні науки»  
(шифр і назва напрямку підготовки, спеціальності)

Голоднюк В. О.  
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Паночішин Ю. М.  
(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., професор каф. КСУ

Биков М. М.  
(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 07 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та  
автоматизації  
Кафедра комп'ютерних наук  
Рівень вищої освіти II-й (магістерський)  
Галузь знань – 12 «Інформаційні технології»  
Спеціальність – 122 «Комп'ютерні науки»  
Освітньо-професійна програма – «Системи штучного інтелекту»

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри КН  
Д.т.н., проф. Яровий А.А.

29.08. 2023 року

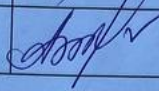
### ЗАВДАННЯ

#### НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Голоднюку Володимирі Олександровичу  
(прізвище, ім'я, по батькові)

- Тема роботи Інформаційна технологія створення чат-бота для розміщення об'яв для товарів та послуг
- керівник роботи к.т.н., доцент кафедри КН Паночишин Ю.М  
затверджені наказом вищого навчального закладу від "18" вересня 2023 року № 247
- Строк подання студентом роботи 13.11. 2023 року
- Вихідні дані до роботи:  
Вхідні дані – Мова програмування: Об'єктно орієнтована; Категорії товарів та послуг – 13 шт..
- Зміст текстової частини:  
Вступ, Аналіз предметної області. Розробка інформаційної технології телеграм бота для розміщення інформації про товари та послуги. Програмна реалізація телеграм бота для розміщення інформації про товари та послуги, тестування та аналіз результатів роботи телеграм бота для розміщення інформації про товари та послуги, висновки, список використаних джерел, додатки
- Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)  
Структура чат-бота для розміщення об'яв для товарів та послуг. Схема алгоритму функціонування телеграм бота для розміщення інформації про товари та послуги. Діаграма прецедентів взаємодії користувача із системою. Робочі вікна програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Паночишин Ю.М., к.т.н., доц. каф. КН		
5	Адлер О. О., к.т.н., доц. каф. ЕПВМ		

7. Дата видачі завдання 29.08 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного рівня інформаційних технологій створення чат-бота для розміщення об'яв для товарів та послуг. Постановка задач дослідження	1.09.23 - 07.09.23	Розділ 1
2	Побудова моделей створення чат-бота для розміщення об'яв для товарів та послуг	08.09.23 - 20.09.23	Розділ 2
3	Практичне застосування та оцінка ефективності розроблених моделей	21.09.23 - 20.10.23	Розділ 3
4	Підготовка економічної частини	21.10.23 - 29.10.23	Розділ 4
5	Апробація та/або впровадження результатів дослідження	30.10.23 - 05.11.23	Тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	Пояснювальна записка, графічний матеріал, презентація

Студент

Керівник роботи

  
(підпис)

  
(підпис)

Голоднюк В. О.

Паночишин Ю.М.

## АНОТАЦІЯ

УДК 004.8

Голоднюк В. О. Інформаційна технологія створення чат-бота для розміщення об'яв для товарів та послуг. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 92 с.

На укр. мові. Бібліогр.: 26 назв; рис.: 26; табл. 10.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології та програмного забезпечення створення чат-бота для розміщення об'яв для товарів та послуг. Розроблено інтелектуального обробника для чат-бота на основі згорткової нейромережі, обрано архітектуру згорткової нейронної мережі LeNet 5, розроблено схему бази даних та основні алгоритми для функціонування чат-бота для розміщення інформації про товари та послуги. Програмне забезпечення розроблено на мові програмування Python у середовищі Visual Studio Code з використанням бібліотеки Aiogram. Для створення згорткової нейромережі було використано бібліотеку Keras, а для створення бази даних було обрано SQLite. Аналіз роботи розробленого чат-бота на основі LeNet-5 показав підвищення достовірності видачі вірних відповідей на 6,2 % порівняно з аналогічним чат-ботом на основі ResNet50 (87,5 %).

Графічна частина складається з 6 плакатів.

У економічному розділі визначено, що згідно узагальненого коефіцієнту конкурентоспроможності, науково-технічна розробка переважає існуючі аналоги. Термін окупності становить 0,4 р., що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Ключові слова: чат-бот, згорткова нейронна мережа, товари, послуги.

## ABSTRACT

Golodniuk V. O. Information technology for creating a chatbot for placing ads for goods and services. Master's thesis in the specialty 122 – «Computer sciences», educational program – «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 92 p.

In Ukrainian language. Bibliogr. : 26 titles; fig . 26; table 10.

This master's qualification thesis is devoted to the development of information technology and software for creating a chatbot for placing ads for goods and services. An intelligent processor for a chatbot based on a convolutional neural network was developed, the LeNet 5 convolutional neural network architecture was chosen, a database scheme and basic algorithms for the operation of a chatbot for placing information about goods and services were developed. The software is developed in the Python programming language in the Visual Studio Code environment using the Aiogram library. The Keras library was used to create the convolutional neural network, and SQLite was chosen to create the database. Analysis of the work of the developed chatbot based on LeNet-5 showed an increase in the reliability of issuing correct answers by 6.2% compared to a similar chatbot based on ResNet50 (87.5%).

The graphic part consists of 6 posters.

In the economic section, it is determined that according to the generalized coefficient of competitiveness, scientific and technical development prevails over existing analogues. The payback period is 0.4 years, which indicates the commercial attractiveness of the scientific and technical development and may encourage a potential investor to finance the implementation of this development and its introduction to the market.

Keywords: chat bot, convolutional neural network, goods, services.

## ЗМІСТ

ВСТУП.....	4
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЧАТ-БОТІВ РОЗМІЩЕННЯ</b>	
<b>ІНФОРМАЦІЇ ДЛЯ ТОВАРІВ ТА ПОСЛУГ .....</b>	<b>7</b>
1.1 Поняття чат-бота та його особливості.....	7
1.2 Аналіз платформ, які підтримують роботу чат-ботів .....	9
1.3 Переваги чат-ботів над веб-сайтами та їх недоліки.....	13
1.4 Постановка задачі .....	15
1.5 Висновок до розділу 1 .....	16
<b>2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ЧАТ-</b>	
<b>БОТА ДЛЯ РОЗМІЩЕННЯ ОБ'ЯВ ДЛЯ ТОВАРІВ ТА ПОСЛУГ .....</b>	<b>17</b>
2.1 Особливості функціонування чат-бота .....	17
2.2 Розробка архітектури чат-бота для розміщення інформації про товари та послуги .....	19
2.3 Розробка інтелектуального обробника для чат-бота на основі згорткової нейромережі .....	21
2.4 Архітектура згорткової нейронної мережі LeNet 5.....	26
2.5 Розробка схеми бази даних.....	32
2.6 Розробка алгоритму роботи чат-бота для розміщення інформації про товари та послуги.....	35
2.7 Висновок до розділу 2 .....	39
<b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТА ДЛЯ РОЗМІЩЕННЯ</b>	
<b>ІНФОРМАЦІЇ ПРО ТОВАРИ ТА ПОСЛУГИ .....</b>	<b>40</b>
3.1 Способи створення чат-бота.....	40
3.2 Обґрунтування вибору програмних засобів .....	42
3.3 Обґрунтування вибору середовища програмування створення чат-бота для розміщення інформації про товари та послуги.....	47
3.4 Програмна реалізація чат-бота для розміщення інформації про товари та послуги .....	48
3.5 Висновок до розділу 3 .....	52

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ЧАТ-БОТА ДЛЯ РОЗМІЩЕННЯ ІНФОРМАЦІЇ ПРО ТОВАРИ ТА ПОСЛУГИ .....	53
4.1 Тестування чат-бота для розміщення інформації про товари та послуги.....	53
4.2 Аналіз результатів роботи чат-бота для розміщення інформації про товари та послуги.....	57
4.3 Висновок до розділу 4 .....	59
5 ЕКОНОМІЧНА ЧАСТИНА .....	60
5.1 Проведення комерційного та технологічного аудиту інформаційної технології створення чат-бота для розміщення об'яв для товарів та послуг .....	60
5.2 Розрахунок витрат на здійснення науково-дослідної роботи .....	61
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором .....	66
5.4 Висновок до розділу 5 .....	71
ВИСНОВКИ .....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	74
Додаток А (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ .....	77
Додаток Б (обов'язковий) Лістинг програми.....	78
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА .....	83
Додаток Г (довідниковий) Інструкція користувача .....	89

## ВСТУП

**Актуальність.** Продаж та придбання різних товарів є невід’ємною частиною кожної людини, від домогосподарок, студентів и навіть школярів до представників крупних компаній. Продають чи купляють все що завгодно від домашніх тварин до послуг різного роду. Всі верстви населення так чи інакше потребують зручного сервісу для виконання цих потреб. Використання месенджерів у сучасному світі набуває великого попиту. Одним із часто використовуваних месенджерів є Telegram. Інформаційна система цього месенджеру є засновником такої програми, як чатбот. Тому була поставлена актуальна задача по розробці системи для розміщення інформації про товари та послуги в месенджері Telegram, іншими словами можна сказати створення комп’ютерної програми чат-бот.

**Зв’язок роботи з науковими програмами, планами, темами.** Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп’ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

**Мета і завдання досліджень.** Метою магістерської кваліфікаційної роботи є підвищення достовірності видачі вірних відповідей чат-бота для розміщення інформації про товари та послуги.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз предметної області створення чат-ботів;
- розглянути існуючі методи створення чат-ботів та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити архітектуру чат-бота для розміщення інформації про товари та послуги;



- розробити інтелектуального обробника для чат-бота на основі згорткової нейромережі;
- розробити структуру та алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

**Об’єкт дослідження** – процес створення чат-боту розміщення інформації для товарів та послуг із застосуванням інтелектуальних технологій.

**Предмет дослідження** – інформаційна технологія та програмні засоби створення чат-боту розміщення інформації для товарів та послуг та достовірність видачі вірних відповідей.

**Методи дослідження.** У роботі використані наступні методи наукових досліджень: системного аналізу, теорії штучних нейромереж для реалізації інформаційної технології, методи математичної статистики для розробки процесу розв’язання задачі створення чат-боту та обрахунків результатів експериментів із програмним засобом, об’єктно-орієнтованого програмування.

**Наукова новизна одержаних результатів.**

1. Набула подальшого розвитку інформаційна технологія створення чат-боту розміщення інформації про товари та послуги, яка відрізняється використанням згорткової нейромережі, що дозволило підвищити достовірність видачі вірних відповідей чат-ботом.

**Практичне значення** одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення створення чат-боту розміщення інформації про товари та послуги.

Запропонована інформаційна технологія сприяє підвищенню достовірності видачі вірних відповідей чат-ботом, зокрема:

- розроблено алгоритм роботи програмного забезпечення створення чат-

бота розміщення інформації про товари та послуги;

- розроблено програмні засоби створення чат-боту розміщення інформації про товари та послуги.

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології створення чат-боту розміщення інформації про товари та послуги. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

**Особистий внесок здобувача.** Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу створення чат-боту розміщення інформації про товари та послуги та методів підвищення достовірності видачі вірних відповідей чат-ботом [1].

**Апробація результатів роботи.** Результати роботи були апробовані на Всеукраїнській науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)» (м. Вінниця, Україна, 2023-2024 рр.).

**Публікації.** За результатами досліджень опубліковано тези доповідей на науково-технічній конференції [1].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЧАТ-БОТІВ РОЗМІЩЕННЯ ІНФОРМАЦІЇ ДЛЯ ТОВАРІВ ТА ПОСЛУГ

## 1.1 Поняття чат-бота та його особливості

Чат-бот - це програма, яка функціонує всередині месенджера. Вона здатна відповідати на запитання, а також ставити їх самостійно. Чат-боти використовуються у різних сферах для вирішення типових завдань.

Створення чат-бота — це не тільки дотримання останніх тенденцій. Насамперед, йдеться про скорочення витрат, збільшення конверсії та покращення якості обслуговування клієнтів. Сотні компаній роблять все можливе, щоб досягти вищезазначених цілей. І їх досягнення необхідні боти як першорядний інструмент.

Згідно з поточною статистикою чат-ботів за 2022 рік, нерухомість є провідною галуззю з прибутків від чат-ботів (28%). Потім йдуть подорожі (16%), освіта (14%), охорона здоров'я (10%) та фінанси (5%). Причина такої популярності очевидна: агенти розуміють, що боти можуть ефективно справлятися з більшістю рутинних завдань, заощаджуючи їх час, ресурси і, зрештою, гроші. Програмне забезпечення чата може бути надзвичайно спеціалізованим, щоб працювати у певних галузях. Наприклад, найкраще програмне забезпечення чат-бота обслуговування клієнтів може відрізнитися від кращого варіанту чат-бота в маркетингових цілях.

Функціональні характеристики чат-ботів та суттєвий розвиток цієї галузі зробило її привабливою як з точки зору звичайних користувачів, так і з точки зору компаній. Одним з перших прикладів комерціалізації чат-ботів було їх вбудовування у вебсайти з метою допомоги користувачам у знаходженні відповідей на їх запитання та з метою оповіщення про події. В подальшому чат-боти ставали більш інтелектуальними, розвиток сфери штучних нейромереж та алгоритмів обробки природньої мови активно сприяли цьому. Вдало запрограмовані чат-боти почали розуміти частково

контекст діалогу та стали пропонувати свої варіанти у відповідь на неточні запитання користувачів. З'явилося таке поняття як «сервіс всередині чат-бота», виникли нові платформи, чат-боти стали невід'ємною частиною більшості державних та приватних бізнесів, наприклад NovaPoshtaBot Нової Пошти, RailwayBot Укрзалізниці тощо.

Чат-боти можуть використовуватись у службах підтримки, допомагаючи вирішити прості запитання, наприклад, такі як зміна пароля.

Чат-боти також можуть використовуватися для пошуку інформації. Наприклад, афіша заходів, прогноз погоди. У месенджері Telegram можна знайти десятки тисяч ботів [2], які можуть розповісти про погоду або допомогти вибрати подарунок.

У сфері подорожей стали одними з перших використовувати чат-боти. Чат-бот може запропонувати напрямки/рейси/ресторани — на підставі пошукових запитів та переваг користувача. Після покупки програма забезпечує клієнтську підтримку, надаючи відповіді на питання, що часто ставляться.

Чат-боти допомагають роботодавцям та здобувачам у процесі пошуку роботи та підбору кадрів. Влітку 2016 року агентство з пошуку роботи FirstJob випустило чат-бот Мія, який не тільки пропонує відповідні вакансії, а й проводить з кандидатом на вакансію співбесіду.

Маркетолог Естер Краудфорд створила чат-бота EtherBot, який розповідає представникам hr-департаментів про кар'єру господині, її здібності та хобі.

Чат-ботів можна використовувати як персональних помічників. Винахідник хештега Кріс Мессіна використовує бота-помічника, MessinaBot, який відповідає всім тим, хто пише йому в Facebook [3].

Свого часу йшли бурхливі дебати з приводу застосування чат-ботів для автоматизації операцій з купівлі-продажу товарів. Компанія-власник інтернет-аукціону eBay, в судовому порядку намагалася припинити діяльність третьої сторони по використанню чат-ботів для пошуку потрібних товарів та

угод. Однак, це спрацювало проти компанії eBay, повернувши увагу широкого кола осіб, які користуються чат-ботами. Інша компанія, яка є оператором британської біржі ставок Betfair стикнулася з настільки великим трафіком від чат-ботів, що була змушена розробляти та впроваджувати Webservice API для можливості контролю такого трафіку.

За кілька останніх років з'явилося дуже багато різноманітних чат-ботів.

Існує шість основних типів чат-ботів [4]:

1. На основі меню/кнопок
2. На основі лінгвістики (засновані на правилах)
3. На основі розпізнавання ключових слів
4. Машинного навчання
5. Гібридні моделі
6. Голосові чат-боти
7. Боти планування зустрічей або бронювання
8. Боти служби підтримки клієнтів

Чат-боти можуть знайомити потенційних клієнтів із послугами або товарами для створення автоворонки продажів.

В данній роботі буде розроблено бот для розміщення інформації про товари та послуги. Цільова аудиторія – від домогосподарок, студентів и навіть школярів до представників крупних компаній. Продаватись буде що завгодно від домашніх тварин до послуг різного роду. Кожна об'ява повинна відповідати певній категорії, містити інформацію про товар чи послугу та інформацію про продавця.

Існують аналоги подібних сервісів у вигляді веб-сайтів, тому потрібно дослідити переваги та недоліки бота в порівнянні з веб-сайтом.

## **1.2 Аналіз платформ підтримки роботи чат-ботів**

Бот-платформа - це додаток, на API якого є можливість створити віртуального «співрозмовника». На тепер чат-боти підтримуються більшістю

популярних месенджерів [5]. Серед яких Viber, Facebook Messenger, Skype, Whatsapp, Sender, Telegram, Instagram і навіть Wechat. Оскільки платформи у всіх месенджерів різняться, то і API у них різні. Тому не є можливим створити один чат-бот для усіх платформ одразу. Обираючи певну платформу для свого чат-бота, потрібно виходити з її можливостей. Розберемо функціональні можливості чат-ботів на найпопулярніших платформах.

Facebook Messenger. Чат-боти, побудовані на платформі Facebook Messenger-а, мають широкий набір функцій.

До нього входять:

- текстові повідомлення;
- кнопки з варіантами дій;
- структурні елементи (до 10 в одному повідомленні);
- рахунки на оплату.

Для створення чат-боту на Facebook, у компанії повинна бути обов'язково публічна сторінка в цій соцмережі.

Viber. Чат-боти у Viber не поступаються іншим платформам по функціоналу. Взаємодію з аудиторією можливо вести з публічного облікового запису в месенджері.

У нього існує набір стандартних можливостей і декілька додаткових, а саме:

- розсилки (відправка повідомлень усім контактам, не підписаним на канал включно);
- створення постів у виді «каруселі» з товарами.

Telegram. Один з перших месенджерів, що зробив можливим створювати ботів. Їх у Telegram можна розпізнати за приставкою «bot» у назві. Це є одною з обов'язкових вимог платформи до власників автоматизованих чатів. Логіка роботи чат-бота в Telegram контролюється за допомогою запитів HTTPS до API платформи [6].

Основні доступні функції:

- інтеграція з іншими сервісами;
- інлайн-режим (бот вбудовується в інші діалоги);
- рішення різноманітних завдань (передбачення погоди, переклад текстів і т. д.);
- інтерактивна взаємодія, гра з користувачем;
- відправка коментарів, текстових повідомлень, пошук інформації;
- виконання команд.

Також на цій платформі є можливості зміни інтерфейсу, зовнішнього зв'язування, кастомізації клавіатури.

Skype. На платформі Skype bot до створення доступні текстові чат-боти, що мало відмінні від конкурентів. Вони так само відповідають на запити користувача по прописаним заздалегідь алгоритмам. У повідомленнях є можливим додавання і зміна елементів.

Із цікавих функцій треба зазначити блок «Меню» з переліком можливостей конкретного чат-бота. Він демонструється користувачу при знайомстві з ботом. Значну увагу приділено здатності бота до вдосконалення. Так, Skype bot користується Cognitive Service Language API. Це передбачає збір з тексту ключових слів і побудова на їх основі відповідей, врахування досвіду користувача, самостійне генерування відповідей із напівструктурованої низки шаблонів.

Sender. Ця платформа активно застосовується в бізнесі, оскільки має гнучкі налаштування. Функціонал бота на Sender забезпечують API інтерфейси: для отримання і обробки платежів – Liqpay, для побудови логіки бізнес процесів - Corezoid.

Віртуальний співрозмовник має такі можливості:

- відправка повідомлень;
- видача рахунків;
- отримання оплат;
- розсилки;

- маркетингові дослідження;
- проведення різноманітних опитувань.

Чат-бот, розроблений у Sender, можна розмістити як у месенджері, так і на сайті.

Slack. Ця платформа найкраще підходить для створення корпоративних чат-ботів [7]. Наприклад, при проведенні опитувань співробітників, складанні звітів про поточні справи, плануванні графіків кожного із співробітників і спільних нарад.

Slack дозволяє створити два варіанти боті. Перший призначено для користувача. Він існує в межах месенджера і при необхідності додається в групи та діалоги співробітників. Другий – це мобільний додаток. Він інтегрується з месенджером, але розміщується за межами Slack.

Узагальнення порівняння особливостей двох найпопулярніших платформ для ботів наведено в таблиці 1.1.

Таблиця 1.1 – Особливості платформ Viber та Telegram

<b>Особливості</b>	<b>Viber</b>	<b>Telegram</b>
Базова концепція	Чат-бот для потреб оптимізації бізнес-процесів при можливості взаємодії з публічними акаунтами	Інтеграція із вибраними сервісами в межах платформи Telegram
Номінальна реалізація	За допомогою Viber API	Можливості Bot API
Атрибути ботів	Налаштування рівня публічного облікового запису	Адреса, ім'я, фото, короткий опис
Номінальні обмеження створення	Вимоги до ботів та вимоги до налаштувань публічного облікового запису	Відсутні



Продовження Таблиці 1.1

<b>Особливості</b>	<b>Viber</b>	<b>Telegram</b>
Підтримувані види вмісту	Текстові блоки, посилання, графічний контент, контакт, геолокація	Все, що підтримується на платформі Telegram (особисті дані, файли, опитування, місцезнаходження)
Рівень комунікації у чатах	Так, на основі повідомлень та за допомогою гнучких клавіатур	Так, за допомогою команд, клавіатур та текстових повідомлень
Можливість монетизації	На базі публічних облікових записів	Підтримується
Додавання ботів до інших чатів	Немає такої можливості	Підтримується
Дозвіл на бесіду	Немає підтримки	Підтримується
Розробка ботів	На базі Viber REST API	За допомогою боту @BotFather / Bot API

### **1.3 Переваги чат-ботів над веб-сайтами та їх недоліки**

Чат-бот однозначно виграє гонку розробки на дистанції з сайтом, навіть з урахуванням готових тем, шаблонів та блоків – у чат-бота теж все це є. З ботом основна кількість часу витрачається на продумування суті, а не візуала. Бот має такі переваги:

1. Відсутність дизайну. Уявіть сайт, для якого не потрібно продумувати розташування кнопок, їх колір, анімацію при наведенні, вибирати шрифт і його розмір для різних заголовків, думати, в який бік має відкриватися меню, яке тло має бути у першого блоку і його висоту... не можете? Правильно, тому що це робота. Сайт без дизайну не може існувати, а бот без дизайну -

типовий бот, якому немає необхідності боротися за увагу шляхом незвичайних градієнтів та курсору над різними елементами.

Вбудована адаптивність. Команда месенджера вже зробила версії під усі пристрої [8]. Не потрібно переживати про некоректне відображення контенту на частині смартфонів або витратити час на верстку з урахуванням особливостей браузерів.

2.Вартість. Середня ціна розробки бота суттєво нижча, ніж ціна розробки веб-сайту.

3.Якість. Людина без досвіду розробки в обох напрямках однозначно припуститься помилок, але для сайту варіантів припуститися помилки більше, як і їх негативний вплив на конверсії, ніж для чат-бота.

4.Робочий результат. Бот це залежний елемент програми, а значить, має обмеження в можливостях щодо сайту — саме цей фактор гарантує робочий результат самостійної розробки. Бот може бути неефективним, але користувач точно отримає інформацію, яку шукав. В руках новачка сайт вийде або набір однотипних нудних шаблонів, які не зможуть утримати увагу, або його індивідуальний зовнішній вигляд не дасть ознайомитися з інформацією.

5.Проблема навантаження. Як щодо додати відео на фон, зробити анімацію всіх елементів, змінити курсор, поставити кілька картинок у роздільній здатності, вклинити аналітику, онлайн чат, попап при закритті сторінки та ще пару цікавих плагінів? Із сайтом можна, а з ботом не можна. На таких веб-сайтах браузер перестає відповідати. Бот не споживатиме ресурсів більше, ніж вимагає месенджер, а значить однаково відображатиметься як на допотопних калькуляторах, так і на майнінговому обладнанні.

Також у ботів зустрічаються наступні недоліки:

1.Бот не пише першим - не всі знають, але бот може відповідати тільки після того, як користувач натиснув на ньому кнопку START, тобто почав діалог.

2. Інформація про користувача — Telegram надає таку інформацію про користувача: ім'я, прізвище, Telegram User ID, мову клієнта та юзернейм. Телефон користувача чат-бот не знає.

Також розробка чат-ботів - гідна альтернатива створенню мобільного додатка. Останній варіант має низку обмежень:

1. Висока вартість створення, що включає опрацювання дизайну та архітектури програми, послуги розробників, тестування.

2. Обмеження з боку провайдерів (App Store, Google Play тощо). Розробник не може впливати на швидкість публікації програми у провайдера, а також на ухвалення рішення про це.

3. Можливі складнощі при керуванні програмою. Швидкість оновлень програми також залежить від провайдера.

Найголовніше обмеження для програми - воно завжди потребує встановлення. Прийде переконати користувача завантажити новий додаток, встановити його на смартфон, розібратися в інтерфейсі, періодично повертатися до нього. Не всім користувачам це подобається.

Месенджери встановлені у більшості користувачів, а одного разу доданий чат-бот завжди залишається на увазі. Чат з роботом відображається в загальному списку з особистими листуваннями, які перевіряються кілька разів на день. Функціонал, розроблений у межах конкретного месенджера, здебільшого зрозумілий і вимагає додаткових витрат часу чи спеціальних навичок.

## **1.4 Постановка задачі**

Аналізуючи переваги та недоліки, можна зробити висновок, що розробка власного телеграм бота є актуальною. Для написання власного бота буде використано мову програмування Python та редактор коду Visual Studio Code.

Для коректної роботи чат-бота необхідно мати доступ до інтернету.

Отже створюваний бот має володіти рядом можливостей:

- можливість створення об'яви за категорією;
- можливість переглядати всі об'яви;
- можливість переглядати свої об'яви;
- можливість редагувати свої об'яви;

Цей бот може бути цікавий не тільки з точки зору його реалізації, але і з точки зору вдосконалень, які можуть бути в майбутньому. Цей бот має необмежені можливості для вдосконалення, наприклад:

- додати нові мови для бота;
- надати можливість відображення останніх об'яв;
- додати адмін панель для зручного керування об'явами;

### **1.5 Висновок до розділу 1**

В даному розділі проаналізовано предметну область створення чат-ботів для розміщення інформації про товари та послуги, перелічено основний функціонал, переваги та недоліки. В підсумку аналізу можна зробити висновок, що розглянуті недоліки незначні в порівнянні з перевагами. Отже, ринок потребує новий продукт, який зможе об'єднати переваги та позбутись недоліків. Проведено аналіз платформ, які підтримують роботу чат-ботів. Здійснено постановку задачі, в якій акцентовано увагу на функціях та можливостях, якими повинен володіти програмний продукт,

## 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ЧАТ-БОТА ДЛЯ РОЗМІЩЕННЯ ОБ'ЯВ ДЛЯ ТОВАРІВ ТА ПОСЛУГ

### 2.1 Особливості функціонування чат-бота

Чат-боти працюють за спеціальним алгоритмом. Використовується спеціальна система аналітики вхідних повідомлень та підбору відповідного варіанта відповіді із заданої бази [9].

Бот реагує на простий текст як команди і видає натомість певний контент. Запити, команди та повідомлення користувачів надсилаються на сервер розробника. Завдяки посередництву телеграм дані шифруються і здійснюється зворотний зв'язок між користувачем та софтом.

Взаємодія між користувачем та ботом виглядає так:

- 1.Користувач дає чат-боту команду.
- 2.Команда передається чат-ботом на сервер.
- 3.Отриманий запит обробляється програмою на сервері.
- 4.Сервер дає боту відповідь.
- 5.Бот виводить його на екран.

Такий цикл повторюється щоразу при натисканні на кнопки та взаємодію з будь-яким телеграм-ботом. При роботі з ним застосовується основна клавіатура та кнопки, що знаходяться безпосередньо у повідомленні робота. Основні кнопки використовуються для навігації, а кнопки в повідомленнях, коли потрібно провести користувача через певну послідовність кроків.

Telegram зберігає всі дані, тобто всі чати, а також ботів, зашифровані в хмарі. Тобто зовнішня резервна копія даних Telegram не є необхідною абсолютно, і всі особисті налаштування є доступними для користувачів, які увійшли до системи, у будь-який час, на різних платформах і скрізь. Однак конфіденційні команди та дані, до яких чат-бот звертається потім ззовні,

можуть бути доступними і за межами хмари, наприклад, на локальних серверах компаній і у власних БД.

Звичайні боти Telegram не є прихованими. Будь-який юзер може їх знайти та використати. Однак боти Telegram також можуть бути доступні лише для певних користувачів, якщо налаштовано окремий канал зв'язку з чат-ботом. Такий канал є «приватною» групою з ботом, до якої належить тільки основний користувач, схвалені ним юзери та фактичний бот. Тото Telegram Bot може використовуватися лише групою користувачів, яку контролює адміністратор.

Якщо розглядати бота Telegram як джерело даних, то фактичні дані залишаються зазвичай на локальних серверах джерела даних. Бот там отримує доступ до потрібних наборів даних, а потім готує їх для виведення повідомлення згідно запиту на кінцевому пристрої. Лише фактична структура програми постійно розташована в хмарі на серверах Telegram.

Фактична програма бота Telegram з відповідними даними зберігається на власних комп'ютерах і БД. Тут безпека програми залежить від адміністратора локального середовища комп'ютера на сайті. Бот Telegram доставляє лише вихідні дані, визначені в програмі, на відповідні кінцеві пристрої. Лише ці вихідні повідомлення розміщуються в хмарі Telegram.

Усі повідомлення і, таким чином, зв'язок з ботом захищені за допомогою шифрування клієнт-сервер/сервер-клієнт. У Telegram Cloud ці дані також розподіляються між рядом серверів і зберігаються там у зашифрованому вигляді.

Зазвичай, чат-бот має таку архітектуру:

а) Серверна частина (backend), де розпізнаються запити і формуються відповіді;

б) Клієнтська частина (frontend), сам месенджер (Facebook, Slack, Telegram та ін.).

Месенджер надає документацію та API для зв'язку платформи backend'a з чат-інтерфейсом бота через протокол HTTP.

## 2.2 Розробка архітектури чат-бота для розміщення інформації про товари та послуги

Архітектура програмного забезпечення — це процес структурування програмного забезпечення на слабо пов'язані та незалежні частини, формування зв'язків і опис процесів передачі даних між ними.

Архітектура програмного забезпечення побудована для того, щоб якнайкраще відповідати вимогам проекту.

Клієнт-серверна архітектура — це широко поширена концепція створення мережових розподілених застосунків, що встановлює правила обміну даними та взаємодії між ними.

Передбачено такі компоненти:

1. Набір серверів, які формують інформацію у відповідь на запит.
2. Набір клієнтів, що використовують сервери та інформацію, отриману від них.
3. Мережа, яка забезпечує обмін інформацією між серверами та клієнтами.

У системі керуванням чат-ботом є три сторони:

1. API Telegram Bot-a.
2. Сервер (основна логіка роботи чат-бота).
3. Клієнт.

На рис. 2.1 розглянуто узагальнений приклад серверної архітектури бота. Бот складається з чотирьох основних частин:

- а) Провайдер месенджера (Messenger Provider);
- б) Обробники команд (Handlers);
- в) Сервіси (Services);
- г) Контейнер стану (State storage).

Провайдер месенджера «обгортає» API Telegram Bot`a і надає універсальний інтерфейс для прийому, обробки та відправлення повідомлень.

Провайдер месенджера визначає що хоче користувач, і спрямовує його до відповідного обробника. Мета цієї частини - забезпечення шару абстракцій над API месенджерів.

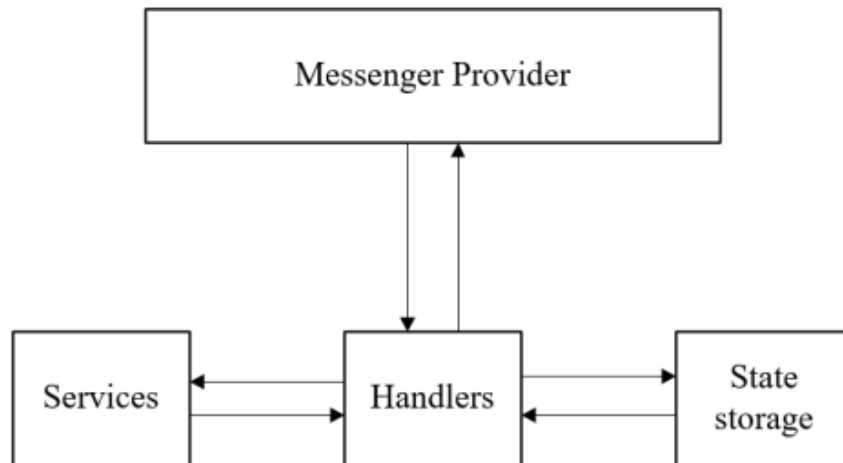


Рисунок 2.1 – Узагальнений приклад серверної архітектури бота

Обробники призначені для обробки повідомлення, яке передане від провайдера месенджера; для делегування завдань сервісам; для оновлення контейнеру стану, якщо це потрібно, та формування повідомлень для відправки їх назад до користувача.

```

@dp.message_handler(commands=['start'])
async def start(msg: types.Message):
    text = '''Доброго дня. Виберіть меню.'''
    keyboard = types.InlineKeyboardMarkup()
    keyboard.add(types.InlineKeyboardButton(text='Створити об\`яву', callback_data='create_ad'))
    keyboard.add(types.InlineKeyboardButton(text='Переглянути об\`яви', url='show_ads'))
    keyboard.add(types.InlineKeyboardButton(text='Мої об\`яви', url='my_ads'))

    await bot.send_message(msg.chat.id, text=text, reply_markup=keyboard)
  
```

Рисунок 2.2 - Приклад обробника

Розглянута архітектура повторюється і у інших чат-ботах із відкритим кодом, із модифікаціями, що пов'язані із використаною мовою програмування та стилем написання, тому вона оптимальна.



Описана вище загальна архітектура бота - частина основних принципів їх розробки та відіграє важливу роль у можливостях подальшого вдосконалення та масштабування. Також при розширенні функціональних можливостей необхідно враховувати можливе число користувачів та обирати коректні інструменти та засоби для розробки та розміщення бота у мережі.

### **2.3 Розробка інтелектуального обробника для чат-бота на основі згорткової нейромережі**

Чат-боти - це системи штучного інтелекту, які розуміють наміри, контекст і почуття користувача, належним чином взаємодіють з ними, що призводить до посиленого розвитку їх створення за останні кілька років. У цьому дослідженні згорткові нейронні мережі (CNN) використовуються як обробник (класифікатор), а деякі спеціальні інструменти для токенізації використовуються для створення чат-бота. Враховуючи те, що в тексті важко застосувати будь-який алгоритм, то використовується техніка під назвою «Вбудовування слів», яка перетворює текст на числа для запуску обробки тексту. Зокрема, було застосовано техніку Word Embedding «Word2Vec».

Машинне навчання (ML) зіграло важливу роль у розвитку чат-ботів. ML можна описати як підмножину штучного інтелекту, в якій математична модель на основі «навчальних даних» побудована для прийняття рішень або прогнозів без програмування. Структура чат-бота базується на попередньо оброблених даних, які легко сприймаються його двигуном. Цей процес складається з введення необробленого тексту, який розділено на окремі слова, які називаються токенами. Токенізовані рядки піддаються текстовій обробці, створенню регулярних виразів і видаленню будь-яких знаків пунктуації та стоп-слів. Ці сутності класифікуються в попередньо визначені класи, які або існують раніше, або створені програмістом системи. Потім відбувається класифікація намірів, під час якої чат-бот намагається зрозуміти, чого хоче користувач, щоб відповісти відповіддю або вирішити проблему. Є два основні

кроки, які хтось повинен зробити щодо чат-ботів, тобто спосіб створення токенів та їх класифікація.

Усі інтелектуальні чат-боти використовують обробку природної мови (natural language processing - NLP), яка забезпечує спілкування між машиною та користувачем із машиною за допомогою природної мови людини. NLP кодує інформацію, яку користувач вставляє у вектор [10].

У поточному дослідженні створення чат-бота базується на згорткових нейронних мережах (CNN). Згорткові нейронні мережі використовуються як класифікатор і використовуються деякі спеціальні інструменти для токенізації. Згорткова нейронна мережа — це алгоритм глибокого навчання, який зберігає ієрархічну структуру, у якій рівні навчаються один від одного. Враховуючи те, що в тексті важко застосувати будь-який алгоритм, було використано техніку під назвою «Word Embedding», яка перетворює текст на числа для запуску обробки тексту. Було застосовано техніку «Word2Vec», оскільки вона пропонує велику кількість попередньо підготовлених даних і дозволяє користувачам тренувати власний набір даних, припускаючи, що вони мають достатньо даних для вирішення проблеми. Крім того, унікальною особливістю Word2vec є те, що вектори можна отримати з інших векторів за допомогою векторних операцій. Це попередній процес, необхідний для вставлення даних у згорткову нейронну мережу. Використовувалися архітектури CNN AlexNet, LeNet5, ResNet і VGGNet.

Основною метою цієї роботи є використання згорткових нейронних мереж (CNN) в якості класифікатора при створенні чат-бота та дослідження того, як різні архітектури нейромереж впливають на час навчання та точність. Крім того, буде використовуватись спеціальний інструмент для токенізації.

Таким чином, можна припустити, що різні архітектури згорткових нейронних мереж можуть принести багатообіцяючі результати у створенні чат-бота.

Структурна схема етапів обробки тексту користувача у обробнику (класифікаторі) на основі згорткової нейронної мережі представлена на рис. 2.3,

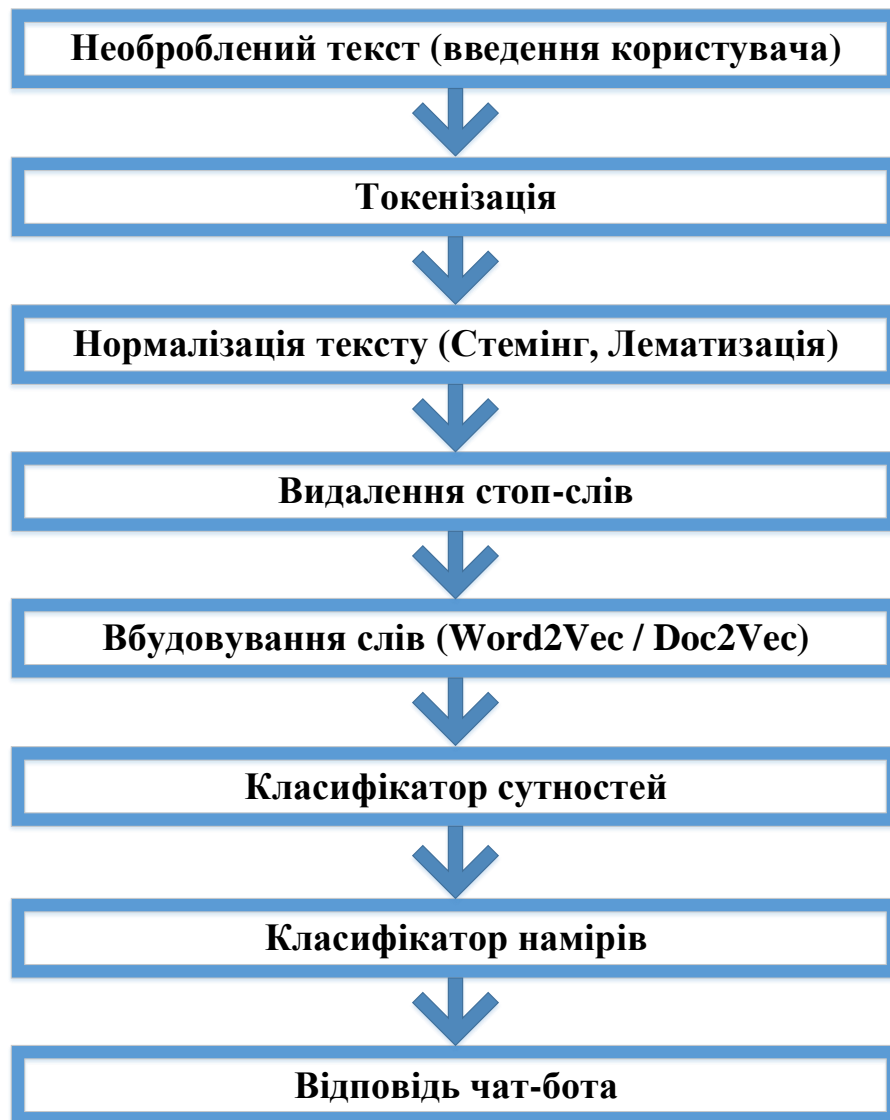


Рисунок 2.3 - Структурна схема етапів обробки тексту користувача у обробнику (класифікаторі) на основі згорткової нейромережі

Деяка складність існує щодо вибору певної методології для того, щоб змусити чат-бот розуміти введені користувачем дані, оскільки він потребує навчання на основі контексту розмови. Відповідно до схеми на рис. 2.3 необхідно виконати наступні кроки.

По-первах дані мають пройти процедуру попередньої обробки, яка вимагає поділ необробленого тексту, який введено користувачем, на окремі слова, що називають токенами. Крім видалення всіляких знаків пунктуації та стоп-слів, токенізовані рядки проходять через нормалізацію тексту, яка може бути досягнута двома техніками, виведенням та лематизацією. Виведення розділяє рядок на менші підрядки за допомогою набору правил (або моделі). Ідея полягає в тому, щоб видалити афікси слів (особливо суфікси), які змінюють значення. Лематизація, з іншого боку, шукає кожну лексему в словнику та повертає канонічний «корінь» слова в словнику, яке відоме як лема. Це може обробляти незвичайні випадки, а також лексеми з різними частинами мови, оскільки шукаються лексеми з основної істини. І виведення, і лематизація мають переваги та недоліки. Виведення є швидшим, оскільки воно просто просить користувачів з'єднати рядки слів. Лематизація, з іншого боку, диктує пошук у словнику чи базі даних і покладається на теги частини мови для визначення кореневої лемі слова, що робить її значно повільнішою, але ефективнішою, ніж виведення. В цьому дослідженні використовується лематизація [11].

У цій роботі використовувався Word2Vec, а точніше його розширення під назвою «Doc2Vec» [11]. Word2Vec приймає окремі слова як вхідні дані, тоді як Doc2Vec приймає ціле речення як вхідні дані. Word2Vec пропонує велику кількість попередньо підготовлених даних, які можна знайти в Інтернеті та використовувати в будь-якій системі.

Крім того, це дозволяє користувачам завантажувати власні дані для навчання мережі. Однак для успішного процесу навчання потрібно багато даних. Крім того, унікальною особливістю Word2vec є те, що вектори можна отримати з інших векторів за допомогою векторних операцій [12]. Ці сутності класифікуються в попередньо визначені класи, які або існують раніше, або створені програмістом системи. За вбудовуванням слів слідує класифікація об'єктів, процедура, у якій відбувається розпізнавання інформаційних одиниць, отриманих із неструктурованого тексту, таких як імена, включаючи

імена людей, організацій, числові вирази та назви місць. Ідентифікація посилань на ці сутності в тексті була визнана одним із важливих підзавдань вилучення інформації (information extracting - ІЕ) [13]. Потім слідує класифікація намірів, під час якої чат-бот намагається зрозуміти, чого хоче користувач, перш ніж нарешті відповісти чи вирішити проблему.

Згорткові нейронні мережі [14] CNN — це алгоритм глибокого навчання, який зберігає ієрархічну структуру, у якій шари навчаються один від одного. Згорткові нейронні мережі також є найпопулярнішими класифікаторами глибокого навчання, які ефективно працюють із класифікацією зображень.

CNN мають кілька шарів, таких як згортковий шар і повністю зв'язаний шар, які мають параметри, а також шар нелінійності та шар об'єднання, які не мають параметрів. Архітектури AlexNet, LeNet5, ResNet і VGGNet CNN були порівняні в цьому дослідженні, а відмінності між ними представлені в таблиці 2.1.

Таблиця 2.1 - Кількість основних шарів, які використовуються різними архітектурами згорткових нейронних мереж.

	Згорткові шари	Повністю зв'язані шари	Об'єднувальні шари	Функція активації виходу
AlexNet	5	3	3	Softmax
LeNet-5	2	3	2	Softmax
ResNet50	49	1	2	Softmax
VGGNet	13	3	5	Softmax

У розділі 4 буде показано, що кращою виявилася згорткова нейронна мережа LeNet-5. Саме тому вона обрана для реалізації інтелектуального чат-бота.

## 2.4 Архітектура згорткової нейронної мережі LeNet 5

Архітектура LeNet 5 — це «Hello World» у сфері згорткових нейронних мереж. Правило зворотного поширення вперше було застосовано до всіх резонних застосувань у 1989 році Янном ЛеКуном та його колегами з Bell Labs. Вони також стверджували, що шляхом накладення обмежень із області завдань гнучкість узагальнення мережі може бути значно посилена. ЛеКун встановив, що одношарові мережі справді мають тенденцію демонструвати слабкі навички узагальнення. Багатошарова, неприродна мережа може функціонувати надзвичайно добре, коли аномалію буде усунено за допомогою детекторів незмінних ознак. Він вважав, що ці відкриття доводять, що зменшення кількості вільних параметрів у нейронній мережі може покращити її здатність до узагальнення.

LeNet — це згорточна нейронна мережа, яку Ян ЛеКун представив у 1989 році. LeNet — загальний термін для LeNet-5, простої згорткової нейронної мережі.

LeNet-5 символізує появу згорточних нейронних мереж і окреслює їх ключові компоненти. Однак у той час він не був популярним через брак апаратного забезпечення, особливо GPU (блок графічного процесора) та альтернативних алгоритмів, таких як SVM, які могли б виконувати ефекти, подібні до ефектів LeNet або навіть кращі за них.

Особливості LeNet-5:

- 1) Кожен згортковий шар включає три частини: згортку, об'єднання та нелінійні функції активації.
- 2) Використання згортки для виділення просторових характеристик (спочатку згортка називалася рецептивними полями).
- 3) Середній шар об'єднання використовується для підвибірки. Як функція активації використовується гіперболічний тангенс 'tanh'.
- 4) Використання багатошарового перцептрона або повністю зв'язаних шарів як останнього класифікатора.

5) Рідкий зв'язок між шарами зменшує складність обчислень  
 Архітектура LeNet-5 CNN має сім рівнів. Три шари згортки, два шари підвибірки та два повнозв'язані шари складають структуру мережі (рис. 2.4).

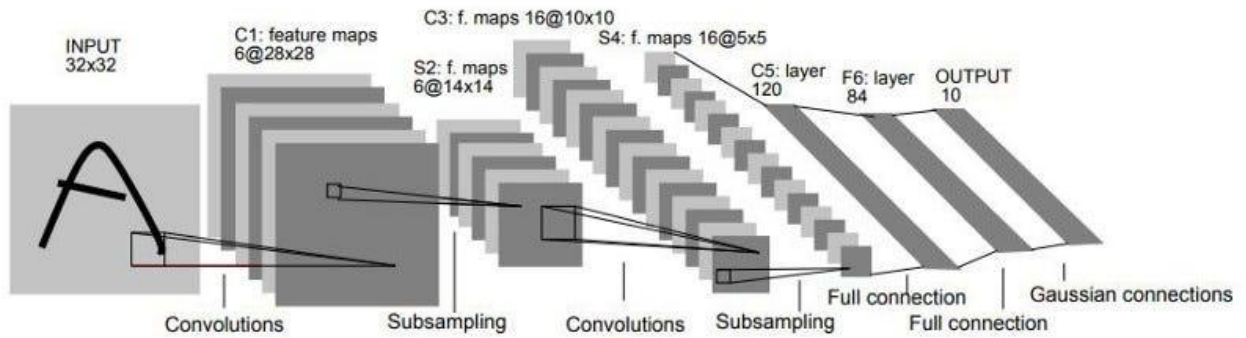


Рисунок 2.4 - Структура згорткової нейромережі LeNet-5

Перший шар. Зображення розміром 32x32 у відтінках сірого служить вхідним для LeNet-5 і обробляється першим згортковим шаром, що складається з шести карт або фільтрів із кроком один. Розміри зображення змінюються від 32x32x1 до 28x28x6 (рис. 2.5).

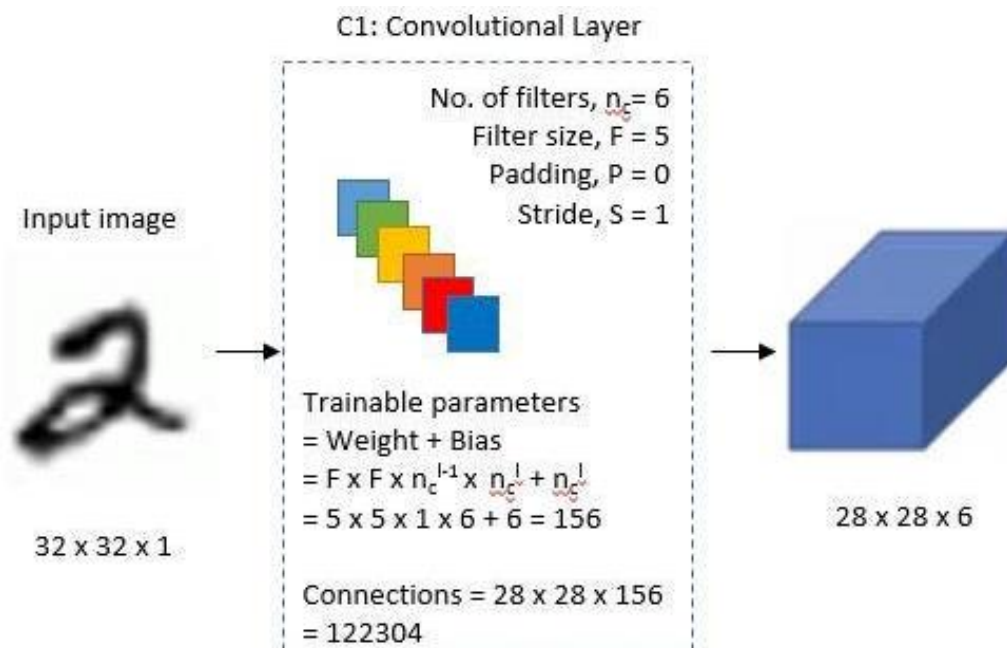


Рисунок 2.5 – Перший шар згорткової нейромережі LeNet-5

Другий шар. Потім, використовуючи розмір фільтра 22 і крок 2, LeNet-5 додає усереднювальний шар об'єднання або шар підвибірки.  $14 \times 14 \times 6$  буде зменшеним розміром остаточного зображення (рис. 2.6).

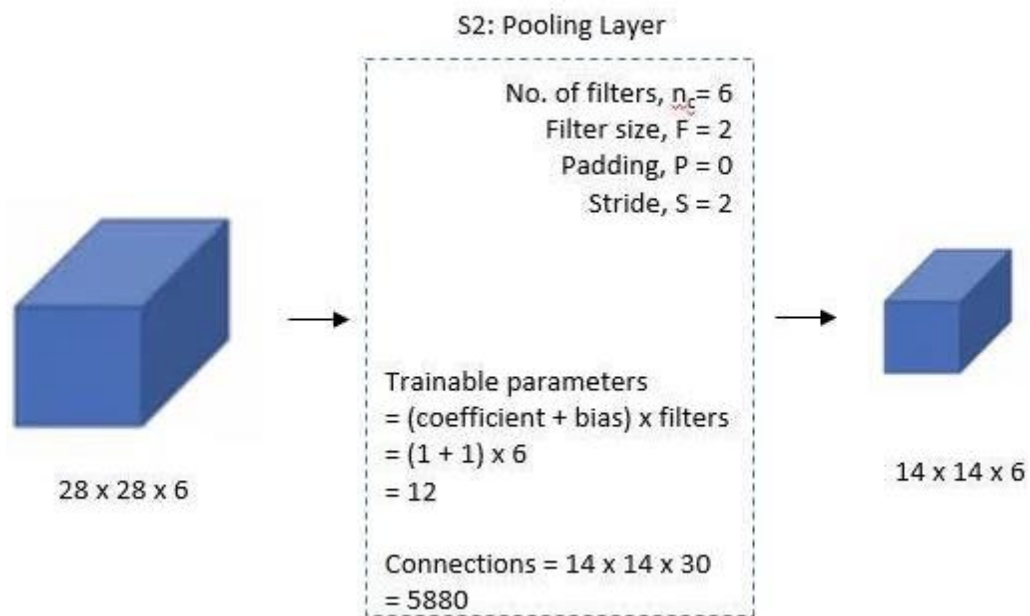


Рисунок 2.6 – Другий шар згорткової нейромережі LeNet-5

Третій шар. Потім присутній другий згортковий шар із 16 картами функцій розміром 55 і кроком 1. Лише 10 із 16 карт об'єктів у цьому шарі пов'язані з шістьма картами об'єктів у шарі нижче, як видно на рис. 2.7.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

Рисунок 2.7 – Таблиця карт ознак із шару S2 (кожен стовпчик), які комбінуються на основі блоків у відповідних картах ознак шару S3 нейромережі LeNet-5



Основна мета полягає в тому, щоб порушити симетрію мережі, зберігаючи керовану кількість зв'язків. Через це в цих шарах є 1516 параметрів навчання замість 2400, і так само є 151600 зв'язків замість 240000 (рис. 2.8).

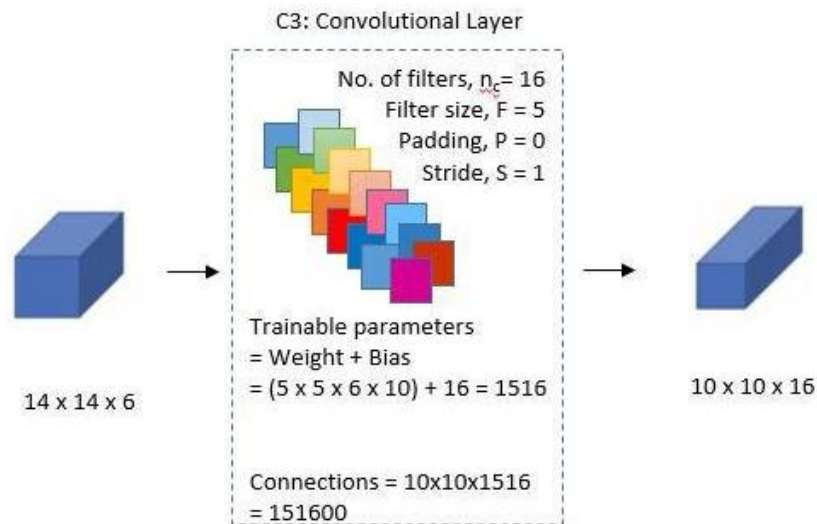


Рисунок 2.8 – Третій шар згорткової нейромережі LeNet-5

Четвертий шар. З розміром фільтра 22 і кроком 2 четвертий шар (S4) знову є усереднювальним шаром об'єднання. Результат буде зменшено до 5x5x16, оскільки цей шар ідентичний другому шару (S2), але має 16 карт функцій (рис. 2.9).

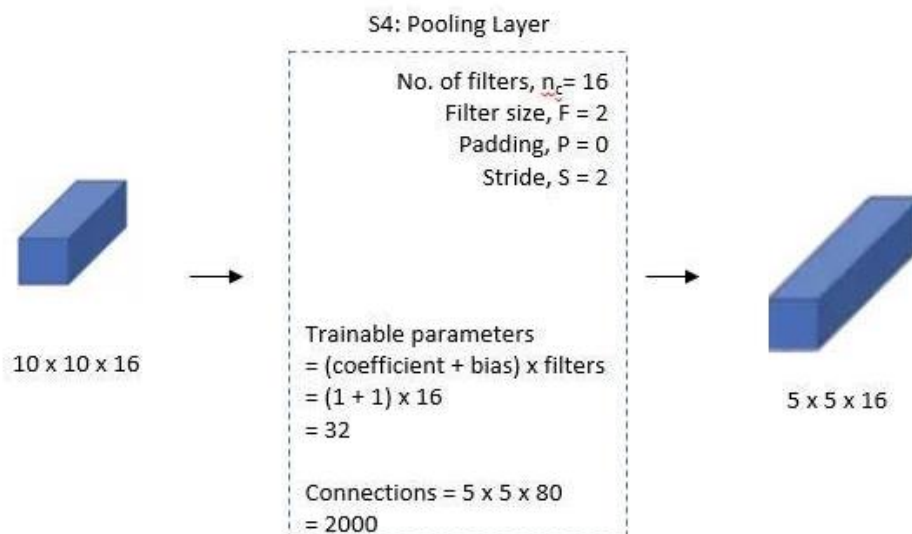


Рисунок 2.9 – Четвертий шар згорткової нейромережі LeNet-5

П'ятий шар. П'ятий шар (C5) містить 120 карт об'єктів, кожна розміром  $1 \times 1$ , і є повністю зв'язаним згортковим шаром. Усі 400 вузлів ( $5 \times 5 \times 16$ ) четвертого шару, S4, підключені до кожного з 120 одиниць у 120 одиницях шару C5 (рис. 2.10).

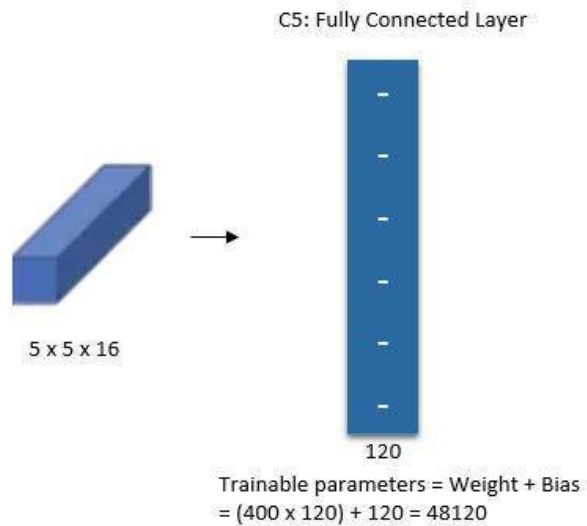


Рисунок 2.10 – П'ятий шар згорткової нейромережі LeNet-5

Шостий шар. Повнозв'язаний шар (F6) із 84 нейрона (рис.2.11).

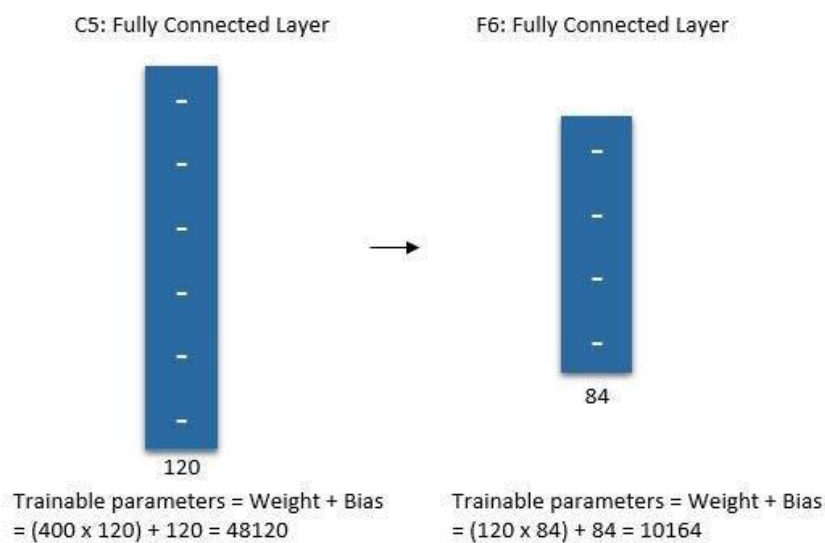


Рисунок 2.11 – Шостий шар згорткової нейромережі LeNet-5

Вихідний шар. Шар виходу SoftMax, який має кількість потенційнозначних виходів по кількості розпізнаваних класів, є останнім шаром (рис. 2.12).

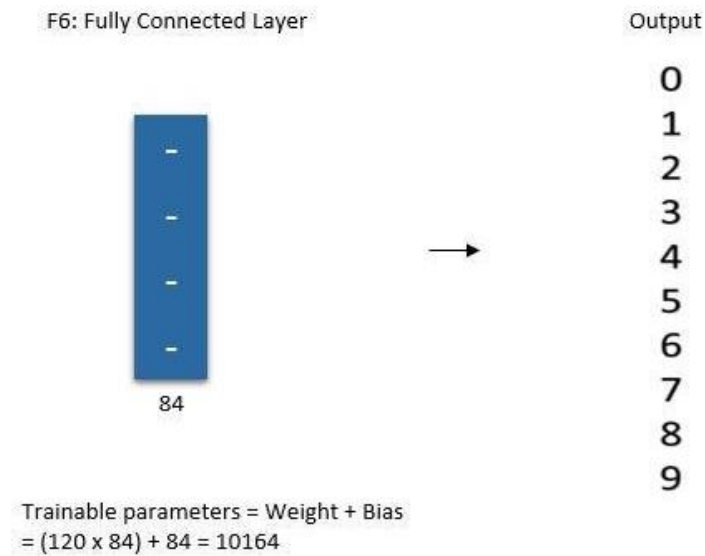


Рисунок 2.12 – Вихідний шар

Загальний опис архітектури LeNet-5 подано у табл. 2.2

Таблиця 2.2 – Загальний опис архітектури LeNet-5

Шар		Карта ознак	Розмір	Розмір ядра	Крок	Активаційна функція
Вхідний		1	32x32	-	-	-
1	Згортка	6	28x28	5x5	1	tanh
2	Усереднений пулінг	6	14x14	2x2	2	tanh
3	Згортка	16	10x10	5x5	1	tanh
4	Усереднений пулінг	16	5x5	2x2	2	tanh
5	Згортка	120	1x1	5x5	1	tanh
6	Повнозв'язний	-	84	-	-	tanh
Вихідний	Повнозв'язний	-	10	-	-	softmax

Згорткова нейронна мережа архітектури LeNet-5 обрана в цій роботі для побудови касифікатора запитів до чат бота. Тільки вхідний шар нейронної

мережі приймає не зображення а вектор ознак запиту, отриманий приперетворенні тексту за алгоритмом DocToVec.

## 2.5 Розробка схеми бази даних

База даних (БД) — організована структура, що призначена для зберігання, обробки та зміни взаємозалежної інформації великих обсягів. Об'єднання значної кількості даних у єдину базу дає змогу для формування множини варіацій групування інформації — у даному випадку інформації про об'яви, категорії, користувачів, фото та регіонів. Головною перевагою баз даних є швидкість внесення та використання необхідної інформації. Завдяки спеціальним алгоритмам, які застосовуються до баз даних, можна легко знаходити потрібні дані лише за кілька секунд. Також в БД існує певні взаємозв'язки інформації: зміни в одному із рядків можуть спричинити зміни в інших рядках. Це дозволяє працювати з інформацією швидше і простіше.

Для ефективного функціонування телеграм бота для розміщення інформації про товари та послуги необхідне сховище інформації про об'яви, користувачів, категорії, регіони та фото. Для вирішення цих потреб було створено реляційну базу даних з такими таблицями: Ads, Users, Regions, Categoryes, Photos.

Нормалізація – процес, у результаті якого можна позбутися дефектів у проектуванні БД. У процесі нормалізації одержують ряд нормальних форм, використовуючи при цьому набір правил, які описують те, що треба і що не треба робити із структурою таблиць. Процес нормалізації передбачає розбиття таблиць на менші, внаслідок чого отримується краща структура.

Нормалізація БД — це покроковий процес розбиття одного відношення згідно алгоритму нормалізації на кілька відношень на основі функціональних залежностей.

Нормалізація таблиць БД — це перший крок у проектуванні структури реляційної БД. Теорія реляційних БД була розроблена в кінці 70-х років 20

століття. Згідно з нею, виділяють шість нормальних форм, п'ять з яких називаються: перша нормальна форма, друга нормальна форма, третя нормальна форма, четверта нормальна форма, п'ята нормальна форма, а також нормальна форма Бойса-Кодда, яка лежить між третьою і четвертою. БД вважається нормалізованою, якщо її таблиці подано, як мінімум, у третій нормальній формі. Зачасту багато таблиць нормалізуються до четвертої нормальної форми. Очевидно, що в нормалізованій БД зменшується імовірність виникнення помилок і вона займає менше місця в пам'яті.

Мета нормалізації:

- Зменшення об'єму для збереження даних.
- Підвищення ефективності бази даних.
- Процедура нормалізації виконується поетапно.

При нормалізації в ідеалі треба домогтися, щоб довільне значення зберігалось в БД в одному екземплярі, причому це значення не має бути отримано шляхом розрахунку з інших даних, які зберігаються в БД. Таблиця вважається зведеною до першої нормальної форми (1НФ) якщо виконуються такі умови:

1. Усі значення таблиці є атомарними (неподільними). Це означає, що комірки таблиці повинні містити одиночні значення і всі записи в окремому стовпці таблиці (атрибуті) повинні мати один і той самий тип.

2. В якості атрибуту таблиці не допускаються групи або масиви даних, що повторюються.

Хаотично створені таблиці даних який завжди відповідають першій нормальної формі (1НФ). Як відомо, приведення таблиці до першої нормальної форми є необхідною умовою нормалізації.

Таблиця відповідає другій нормальній формі, якщо:

1. Таблиця наведена до першої нормальної форми 1НФ;
2. У таблиці відсутні часткові залежності. Це означає, що кожен неключовий атрибут (поле) таблиці неприводимо залежить від первинного ключа таблиці (ключа відношення).

Друга нормальна форма актуальна лише тих відносин, у яких є складові (комполитні) ключі.

Третя нормальна форма. Відношення знаходиться в 3НФ, коли знаходиться в 2НФ і кожен ключовий атрибут нетранзитивно залежить від первинного ключа. Іншими словами, друге правило вимагає виносити всі ключові поля, вміст яких може ставитися до кількох записів таблиці в окремій таблиці.

Відношення «Один-до-одного» є найпростішим між таблицями. Цей тип зв'язків зустрічається не часто. І тут об'єкту однієї сутності можна порівняти лише одне об'єкт іншій сутності.

Часто такий тип зв'язків передбачає розбиття однієї великої таблиці на декілька маленьких. Основна батьківська таблиця в цьому випадку продовжує містити дані, що часто використовуються, а залежна дочірня таблиця звичайно зберігає дані, що використовуються рідше. Щодо цього первинний ключ залежної таблиці одночасно є зовнішнім ключем, що посилається на первинний ключ із основної таблиці.

«Один-до-багатьох» - найпоширеніший тип зв'язків. У цьому типі зв'язків кілька рядків із дочірньої таблиці залежать від одного рядка в батьківській таблиці. Зовнішній ключ таблиці, що містить велику кількість зв'язків, може бути включений до складу складеного первинного ключа, але він є зовнішнім для базової таблиці.

На рисунку 2.13 зображені зв'язки таблиць «Users», «Categoryes», «Regions», «Photos», «Ads». В даному випадку використовується зв'язок «Один-до-Багатьох» для таких таблиць: «Regions-Users», «Ads-Photos», «Categoryes-Ads», «Regions-Ads», «Users-Ads». Зв'язки типу «Один до одного» не використовуються.

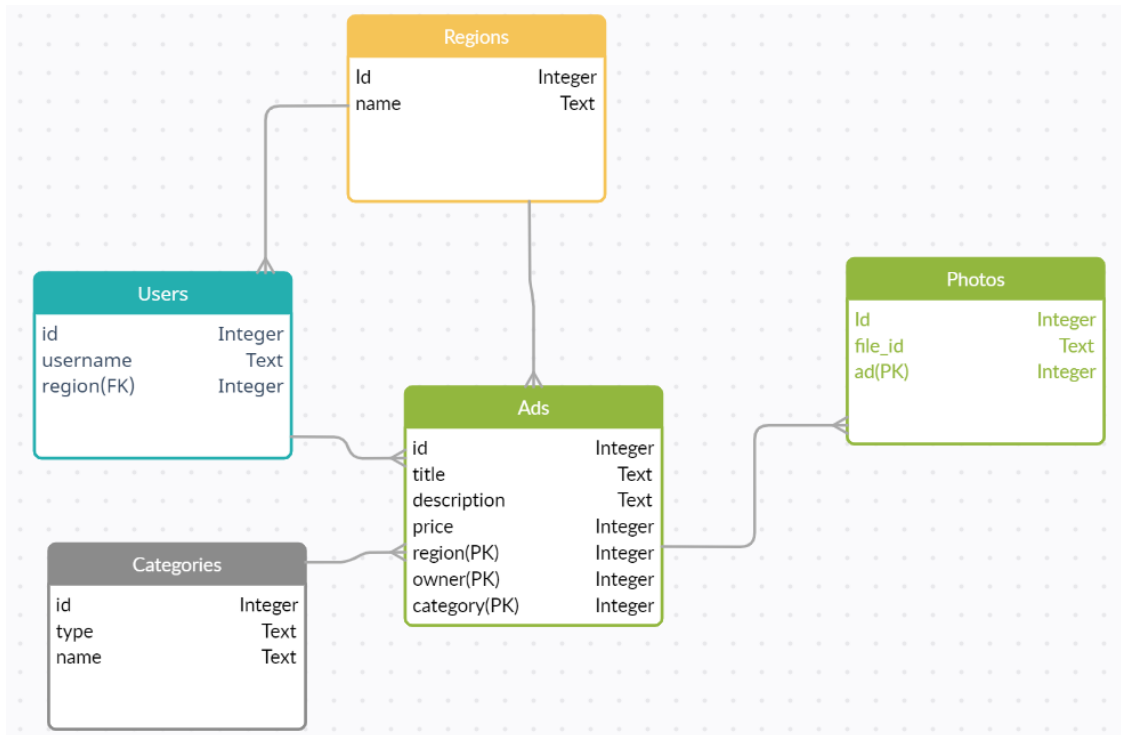


Рисунок 2.13 – Схема бази даних телеграм бота для розміщення інформації про товари та послуги

При типі зв'язків «Багато-до-Багатьох» один рядок з таблиці А може бути пов'язаний з безліччю рядків з таблиці В. У свою чергу один рядок з таблиці може бути пов'язаний з безліччю рядків з таблиці А. Для реалізації даного типу зв'язків необхідно створення допоміжної проміжної таблиці. Цей тип зв'язків також не використовується в даній схемі.

Таким чином, використовуючи різні типи зв'язків була побудована схема бази даних для телеграм бота для розміщення інформації про товари та послуги.

## 2.6 Розробка алгоритму роботи чат-бота для розміщення інформації про товари та послуги

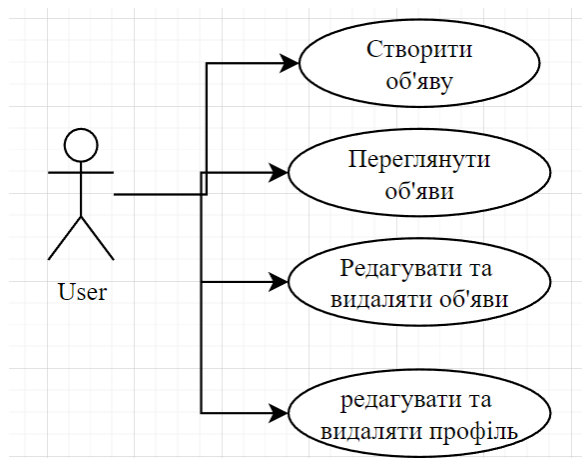
Зберігати інформацію про товари та послуги у вигляді об'яв у месенджері Telegram повинна інформаційна система за допомогою чат-бота. І звичайно бот повинен мати інтерфейс, зручний для роботи, без надбудов, що

ускладнює процес роботи. Особа, яка користується месенджером, повинна відразу зрозуміти, як шукати або створювати об'яви. Виходячи з цього, ми можемо сформулювати вимоги до нашої системи через бот.

Вимоги до системи:

- система має працювати в усіх версіях месенджера Telegram;
- можливість створювати об'яви за різними типами та категоріями;
- можливість переглядати об'яви за різними типами та категоріями;
- можливість редагувати та видаляти власні об'яви;
- можливість шукати об'яви за юзернеймом користувача;
- можливість редагувати та видаляти власний аккаунт.

Спираючись на сформовані дані про вимоги до боту та всі дії для, яких можна використовувати чат-бот, було створено Use Case діаграму. Вона продемонстрована нижче (рис 2.14.)



Рисунк 2.14 – Діаграма варіантів використання боту

Виклик головних функцій бота відбувається через головне меню за відповідними клавішами головної клавіатури. При першій взаємодії користувача з ботом необхідно перевірити, чи користувач існує, інакше викликати меню реєстрації з вибором регіону. В головному меню буде 4 клавіші для виклику меню з необхідними функціями.



Створення нової об'яви розділено на кілька етапів, на кожному з яких бот буде запитувати для вводу таку інформацію, як тип, категорія, заголовок, опис, ціна та фото об'яви. На кожному етапі можна скасувати створення та повернутись до головного меню. Схему алгоритму створення об'яви зображено на рисунку 2.15.

При пошуку об'яв, необхідно обрати регіон окрім типу та категорії.

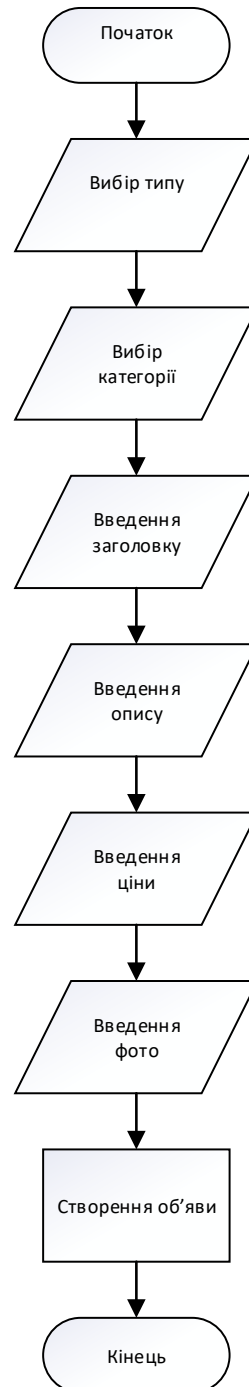


Рисунок 2.15 – Схема алгоритму створення об'яв

Схему алгоритму всього проекту зображено на рисунку 2.16.

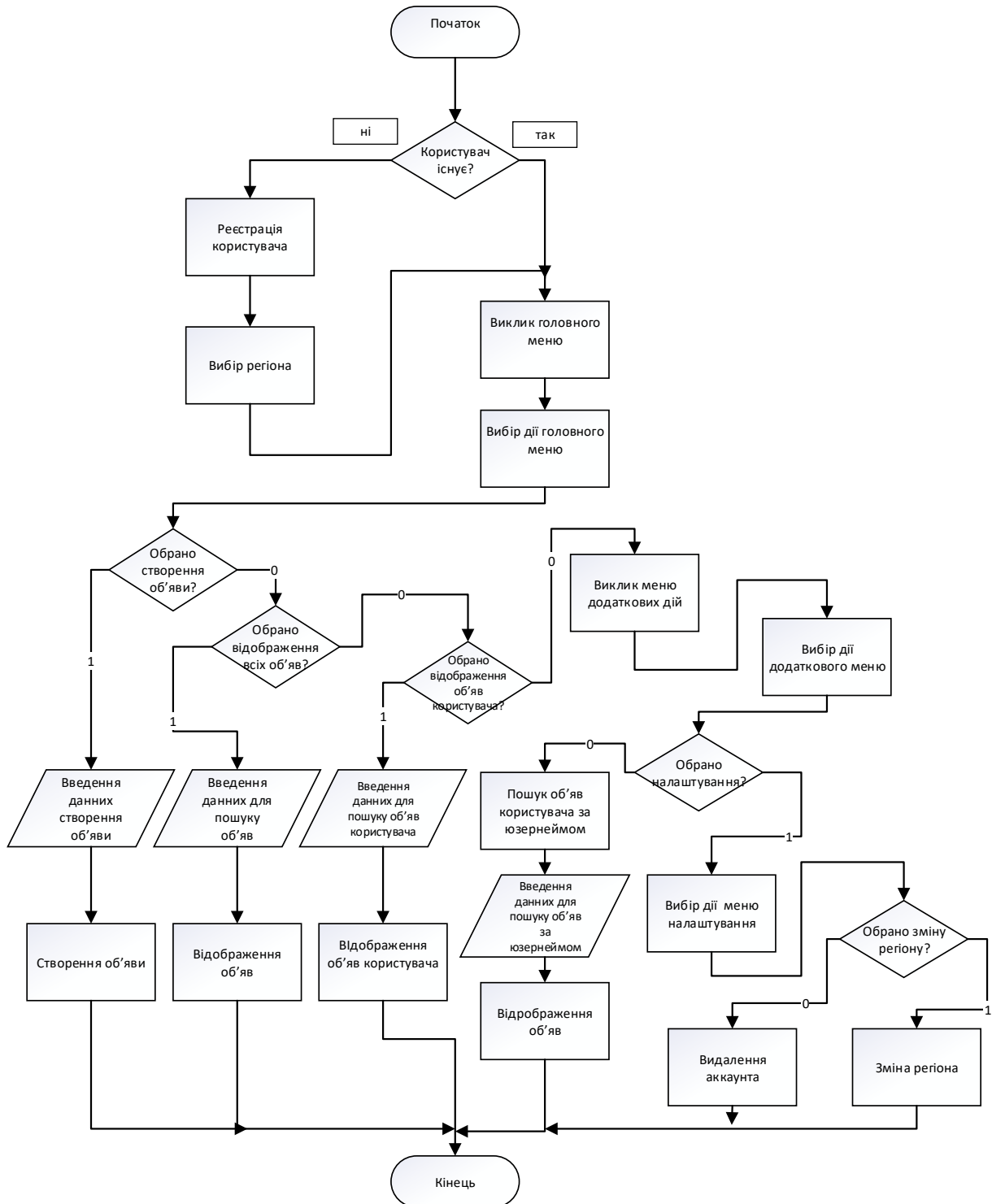


Рисунок 2.16 – Схема алгоритму проекту

## **2.7 Висновок до розділу 2**

У даному розділі визначено, якими будуть особливості розроблюваної інформаційної технології, розроблено архітектуру проекту. Розроблено інтелектуального обробника для чат-бота на основі згорткової нейромережі, обрано архітектуру згорткової нейронної мережі LeNet 5, розроблено схему бази даних та основні алгоритми для функціонування чат-бота для розміщення інформації про товари та послуги.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЧАТ-БОТА ДЛЯ РОЗМІЩЕННЯ ІНФОРМАЦІЇ ПРО ТОВАРИ ТА ПОСЛУГИ

### 3.1 Способи створення чат-бота

Існує два підходи до створення бота. Можна створити свою команду програмістів та розробити чат-бот з нуля, або піти простішим шляхом – застосовувати безкоштовні конструктори. Перший спосіб є більш дорогим та тривалим. Його використовують у середньому та великому бізнесі, коли необхідні більш складні алгоритми взаємодії з користувачем та докладний аналіз одержаної інформації.

Другий спосіб - створення чат-бота за допомогою безкоштовних конструкторів є оптимальним варіантом для тих, хто потребує швидких і нескладних рішень. Є багато доступних сервісів, які допомагають розробити бот для месенджера без допомоги фахових програмістів.

Наприклад, для того, щоб розробити віртуального помічника у Telegram, можна використати сервіс SendPulse. Потрібно зареєструвати обліковий запис і обрати месенджер для чат-бота: Telegram/Facebook. Після цього знайти у Telegram @BotFather з офіційним синім значком. Далі натиснути кнопку «Start» для активації і дати ім'я своєму боту. Наступний крок: скопіювати токен у свій обліковий запис на SendPulse і запустити бота. Коли Telegram-бот підключено, то залишається тільки налаштувати в обліковому записі ланцюжок повідомлень. SendPulse дозволяє створювати до 3-х чат-ботів з можливістю надсилання до 10 000 повідомлень щомісяця необмеженій кількості отримувачів. Такий конструктор дозволяє створювати ланцюжки повідомлень для Telegram та Facebook Messenger на одній платформі.

Для розробки чат-бота сайтів на WordPress, Drupal, Joomla або іншій CMS, існують готові плагіни. Встановити плагін можна самостійно: наявність особливих знань у програмуванні не є обов'язковою. На сайті бот спілкується

з відвідувачами через вікна з чатом. Чат-бот може автоматично відкривати вікно та сам починати діалог або чекати, поки користувач звернеться до нього.

Є ще один спосіб створення чат-бота на сайті – використання онлайн-конструкторів. Наприклад, конструктора Botsify. Завдяки цьому сервісу можна створити бота не тільки для веб-сайтів, але і для Slack і Facebook Messenger. Для створення тестової версії чат-бота потрібна реєстрація, а для повноцінної роботи - щомісячна плата 50 \$.

Щоб цілодобово отримувати повідомлення до директу Інстаграм та відповідати на користувацькі запити, інтернет-магазини створюють чат-боти, наприклад, за допомогою конструктора SMMBOT.net. Функціонал таких чат-ботів полегшує значно роботу бізнесу. Крім того, що вони закривають проблеми із частими запитаннями клієнтів, через таких чат-ботів є можливість робити тригерні розсилки. Також вони можуть інтегруватися з CRM, щоб автоматично передавати дані клієнтів у систему.

Спосіб створення ботів через онлайн-конструктори використовуються для реалізації простих ботів. Так як у даній роботі використовуються більш складні алгоритми взаємодії з користувачем, було обрано спосіб створення з нуля через мову програмування. Також перевагами такого способу є:

1. Можна реалізувати практично будь-який функціонал, за неможливості «зробити так» можна зробити схожу альтернативу.
2. Конфіденційність: всі дані під контролем і не потраплять до третіх осіб.
3. У майбутньому бот можна зробити мультимовним, що дозволить пропонувати свої послуги більш ніж в одній державі.
4. Великий простір інтеграцій: можна підключити будь-які сторонні сервіси для роботи з даними.

### 3.2 Обґрунтування вибору програмних засобів

Для створення підсистеми потрібно організувати базу даних, зробити зручний функціонал та забезпечити швидку роботу. Для написання проекту буде використовуватись мова програмування Python та бібліотека Aiogram. Для створення згорткової нейронної мережі знадобляться бібліотеки Keras [15], та TensorFlow [16].

Keras — це відкрита нейромережева спеціалізована бібліотека, написана мовою Python. Вона може працювати зверху TensorFlow, Theano та Microsoft Cognitive Toolkit. Її було спроектовано для реалізації швидких експериментів з мережами глибокого навчання, та зосереджено на зручності у використанні, модульності та розширюваності. Keras було створено як частину дослідницького проекту ONEIROS (англ. Open-ended Neuro-Electronic Intelligent Robot Operating System).

З 2017 року команда TensorFlow Google вирішила підтримувати Keras в межах основної бібліотеки TensorFlow. Було заявлено, що Keras замислювався скоріше як інтерфейс, ніж як самостійна система для машинного навчання.

Для швидкої та зручної взаємодії з БД було обрано систему керування базою даних SQLite [17]. Робота з базою даних, не виходячи з підсистеми, забезпечується розширенням sqlite3. Саме завдяки цьому адміністрація підсистеми може зручно та легко працювати з даними та інформацією в БД.

Мова Python проста у використанні. Python забезпечує набагато більше інструментів для створення та підтримки великих програм [18,19], ніж shell. Якщо порівнювати з іншими мовами, то Python краще для оброблення помилок ніж C і є дуже високорівневою мовою програмування. Також Python має вбудовані типи даних високого рівня, такі як гнучкі масиви та словники. Їх ефективне застосування у C вимагає великого часу. Python застосовується з більш загальними типами даних і до ширшого кола завдань, аніж Awk, і навіть, Perl, у той час як багато речей на Python робляться так само просто.

Python дозволяє розділяти додатки на модулі, які потім можливо використовувати в інших кодах. Python постачається з потужною бібліотекою стандартних модулів, які використовують як основу для нових програм або як приклади при вивченні мови програмування. За рахунок них можна одразу з «коробки» працювати із файлами, мережевими підключеннями, системними викликами та навіть інтерфейсами у різних графічних бібліотеках.

Python - це інтерпретована мова програмування, яка суттєво допомагає економити час, який зазвичай витрачають на компіляцію. Інтерпретатор також використовується інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестові функції при розробці “знизу-до-гори”. Це так само зручно, як настільний калькулятор. Python дозволяє писати компактно і читабельно будь які програми. Мова розширення Python: знання мови C дозволяє додавати нові вбудовані функції, модулі для виконання критичних операцій максимально швидко або для інтерфейсу в комерційних бібліотеках, які доступні тільки в цифровому форматі. Інтерпретатор мови Python може бути інтегрований в програму, що написана на C. Програми, які написано на Python, часто значно менші, ніж еквівалентні на C або C++ з декількох причин:

- Типи даних високого рівня дозволяють виражати складні функції в Директивах.
- Інструкції з групування виконуються за допомогою відступів (замість дужок).
- Відсутність необхідності рекламних змінних;

Мова Python у даний час використовується десятками тисяч розробників у всьому світі, і кількість людей, що використовують її стрімко зростає. Python залучає користувачів з багатьох причин. Вона використовується для розробки програмного забезпечення і дозволяє розвиватися набагато швидше, аніж традиційні мови Java, C або C++. Ця мова працює так само добре у Windows, як і у інших ОС, таких як Macintosh, UNIX

та OS/2, які можуть бути використані для зручного розгортання невеликих програм або скриптів і розробки великих програм. Python може надати доступ до потужного та легкого у використанні набору із 29 інструментів GUI.

Традиційний тип мови програмування C і Паскаль мають кілька характеристик, наприклад, базові формули, жорстка типізація, складні (і, звичайно, великі) цикли, потреба у великій кількості кодів для виконання відносно дрібних завдань. Візьмемо Java. Зараз вона є досить новою, але розділяє більшість функцій, залучених до цього переліку. Роботу з мовою Python значно полегшує суворі типізація, з чим погодяться усі програмісти, які знайомі з традиційними мовами.

Є багато відмінностей Python. Перераховуємо основні:

1. Керування пам'яттю є повністю автоматичним — не потрібно турбуватися про призначення або відпускання пам'яті. Немає загрози «небезпечних посилань». Єдиною мовою, яка пропонує таку ж концепцію, є Java.

2. Типи зв'язані з об'єктами, а не зі змінними. Тобто виходить, що значення будь-якого типу можна призначити змінній, і що (наприклад) таблиця може містити різні типи об'єктів. Традиційні мови не надають таку функцію.

3. Виконання операцій на абстрактнішому рівні.

Як бібліотека для розробки чат-бота було обрано Aiogram [8]. Вибір на користь Aiogram пояснюється тим, що він є простим та повністю асинхронним кодом. Це дозволяє робити ботів, які працюють швидше та простіше.

Як правило, програмний код виконується послідовно, лише одна конкретна операція відбувається на даний момент часу. Якщо функція залежить від результату виконання іншої функції, то вона повинна дочекатися, поки потрібна їй функція не завершить свою роботу і не поверне результат, і поки це не відбудеться, виконання програми, по суті, буде зупинено з точки зору користувача. Для користувачів така робота програми



не зручна, вона зависає. Щоб уникнути цього, використовують асинхронний код. Він прибирає блокуючу операцію з основного потоку програми, і вона продовжує працювати – для користувача така програма не зависає та працює швидше.

SQLite — це швидка і легка однофайлова СУБД, що вбудовується, на мові C, яка не має сервера і дозволяє зберігати всю базу локально на одному пристрої. Для роботи SQLite не потрібні сторонні бібліотеки або служби.

SQLite3 це консольна утиліта для роботи з SQLite від розробників СУБД. Вона запускається та працює в командному рядку, в консолі операційної системи. Ви можете завантажити версії для Windows, Mac OS та Linux.

За функціональністю SQLite3 - програма-клієнт для клієнт-серверних програм. З її допомогою можна вводити та передавати запити до бази даних: створювати, модифікувати, отримувати чи видаляти таблицю. Різниця в тому, що вона звертається не до окремого процесу-серверу, а до вбудованого в додаток двигуна SQLite.

У SQLite3 можна писати SQL-код: утиліта надішле запит до ядра, отримає та відобразить результат [18].

Переваги SQLite:

1. Висока швидкість. Завдяки особливостям архітектури SQLite працює швидко, особливо читання. Компоненти СУБД вбудовані у додаток і викликаються у процесі. Тому доступ до них швидший, аніж при взаємодії між різними процесами.

Збереження даних у одному файлі. БД складається із табличних записів, індексів, зв'язків між ними та інших компонентів. У SQLite вони зберігаються у єдиному файлі (database file), який знаходиться на тому ж таки пристрої, що і програма. Щоб під час роботи не виникало помилок, файл блокується для сторонніх процесів перед записом. Раніше це призводило до того, що записувати дані до бази міг лише один процес одноразово. Але у нових версіях це вирішується перенастроюванням режиму роботи СУБД.

2. Мінімалізм. Автори SQLite користуються принципом «мінімального

повного набору». Зі всіх можливостей SQL у ній є найбільш потрібні. Тому SQLite відрізняють малий розмір, простота рішень та легкість адміністрування. Для підвищення базової функціональності можна використовувати стороннє програмне забезпечення та розширення.

3.Надійність. Код на 100% покритий тестами. Це означає, що кожен компонент ПЗ протестований. Тому SQLite вважається надійною СУБД із мінімальним ризиком непередбачуваної поведінки.

4.Нульове конфігурування. Перед використанням СУБД не є потрібним складне налаштування чи тривале встановлення. Для вирішення більшості завдань їй можна скористатися «з коробки», без встановлення додаткових компонентів.

5.Мінімальний розмір. Повністю налаштований SQLite з усіма параметрами займає менше 400 Кб. Якщо використовувати СУБД без додаткових компонентів, можна зменшити розмір до 250 Кб. Він залежить лише від кількості завантаженої інформації. Незважаючи на мінімальний розмір, SQLite підтримує більшість функцій стандарту SQL2 і має ряд власних.

6.Доступність. SQLite є у публічному доступі. Немає правових обмежень на її використання, а власником вважається суспільство. Можна відкривати, переглядати та змінювати вихідний код встановленого програмного забезпечення.

7.Кросплатформність. СУБД підходить для UNIX-подібних систем, MacOS та Windows.

8.Автономність. Система є незалежною від стороннього програмного забезпечення, бібліотек або фреймворків. Для роботи додатку з базою SQLite не потрібні додаткові компоненти. Також не є обов'язковим наявність доступу до інтернету: вся БД зберігається на пристрої, отримувати дані можливо локально.

### 3.3 Обґрунтування вибору середовища програмування створення чат-бота для розміщення інформації про товари та послуги

Visual Studio Code – це один із самих популярних редакторів коду, які використовують програмісти [20,21]. Він легкий, швидкий і потужний. VSC був розроблений Microsoft як крос-платформний редактор коду для написання хмарних та веб-додатків. Вперше це було оголошено компанією Microsoft 29 квітня 2015 на конференції Build 2015, що проходила в Сан-Франциско. Через кілька місяців - 18 листопада 2015, VSC було випущено під ліцензією MIT, а вихідний код став доступним на GitHub. 14 квітня 2016 VSC було випущено в Інтернет.

Головна причина, завдяки якій VSC став таким популярним, полягає в наявності в ньому всього, що вимагається від редактора коду з деякими корисними та додатковими функціями (рис. 3.1). Він швидкий, легкий, з відкритим кодом та крос-платформний, а також має інші цікаві функції, які дають йому додаткові переваги перед іншими редакторами.

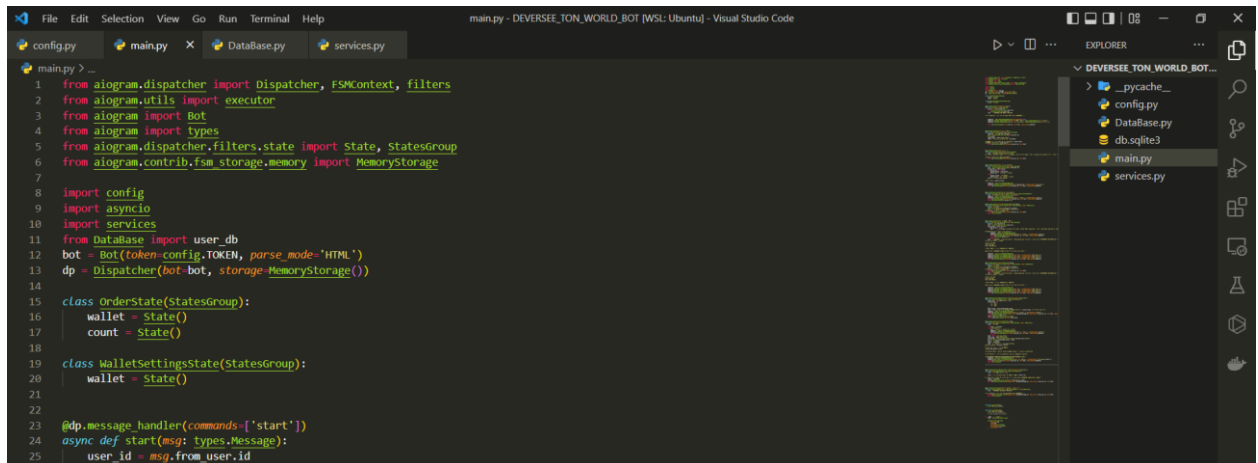


Рисунок 3.1 – Редактор кода VS Code.

Переваги VS Code:

1. Крос-платформність. Це безкоштовний, з відкритим вихідним кодом і крос-платформний редактор, який працює на Windows, Linux і MacOS, так що

ви можете працювати незалежно від платформи, на якій засновано ваш пристрій.

2. Підтримує багато мов програмування. VS Code підтримує майже всі основні мови програмування. Він підтримує Python, JavaScript, HTML, CSS, TypeScript, C++, Java, PHP, Go, C#, PHP, SQL, Ruby, Objective-C та багато іншого.

3. Кастомізація. Як і будь-який інший популярний редактор Visual Studio Code також забезпечує налаштування. Насправді, він забезпечує екстремальне налаштування завдяки своєму гнучкому налаштуванню переваг та безлічі розширень. VSC дає можливість змінити тему, змінити сполучення клавіш, налаштувати налаштування, створити фрагменти коду та багато іншого.

4. Налагодження. VSCode поставляється з вбудованим налагоджувачем, який також є однією з його ключових функцій. Це допомагає прискорити цикл редагування, компіляції та налагодження будь-якого програміста. Однак за замовчуванням він поставляється тільки з відладчиком, який підтримує NodeJS, який може налагоджувати все, що переноситься на JavaScript, але знову ж таки є можливість використовувати розширення для інших середовищ виконання.

### **3.4 Програмна реалізація чат-бота для розміщення інформації про товари та послуги**

Для розробки бота застосовується підхід клієнт-серверної архітектури. Для початку в головному меню розміщено клавіші, за допомогою яких можна викликати меню з потрібним функціоналом. В кожне меню буде вкладено інформацію і відповідний функціонал. Після визначення основних меню бота буде розроблено модуль service, який викликається в обробниках та взаємодіє з базою даних, реалізуючи основну логіку телеграм бота.

На рисунку 3.2 наведено приклад функції `create_ad` з модуля `services`, яка отримує необхідні дані та взаємодіє з базою даних для створення об'яви.

```
def create_ad(ad_data, user_id):
    user = user_db.get_user_by_id(user_id)

    photos = ad_data.pop('photos')

    ad_id = ads_db.create_ad(title=ad_data['title'], description=ad_data['description'],
                             price=ad_data['price'], category=ad_data['category'], user_id=user['id'], region=user['region'])

    for photo_id in photos:
        photos_db.create_photos_to_ad(ad_id, photo_id)
```

Рисунок 3.2 – Приклад функції з модуля `services`.

При створенні об'яви є можливість додавання кількох фото для опису, шляхом відправлення їх одним повідомленням. Так як в Aiogram відсутні фільтри для обробки всіх повідомлень MegiaGroup в одній функції, було розроблено Middleware для додавання даної функціональності.

Middleware - це програмне забезпечення [22] багаторазового використання, яке використовує шаблони та рамки для подолання розриву між функціональними вимогами додатків і базовими операційними системами, стеками мережевих протоколів і базами даних.

Aiogram - це посередник, який отримує доступ до оновлення до того, як воно потрапить до обробника. Middleware працює зі всіма типами подій телеграма, та дає можливість міняти дані події, та додавати нові для передачі їх в обробник [23]. На рисунку 3.3 наведена схема роботи Middleware.

Кожне повідомлення, яке потрапляє в бот від серверів Telegram проходить ланцюжок:

- `pre_process_update`
- `process_update`
- `pre_process_message`
- `filters`
- `process_message`

- handler
- post\_process\_message
- post\_process\_message

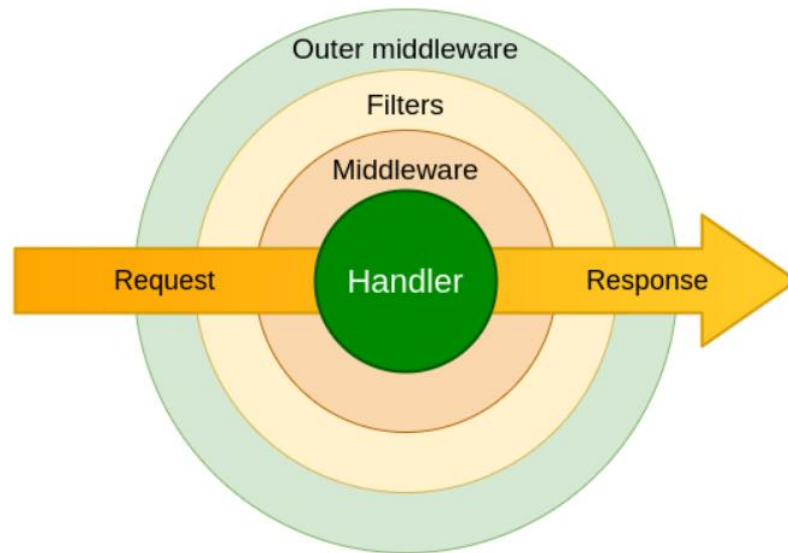


Рисунок 3.3 – Схема роботи Middlewares в aiogram

```

class AlbumMiddleware(BaseMiddleware):
    """This middleware is for capturing media groups."""

    album_data: dict = {}

    def __init__(self, latency: Union[int, float] = 0.01):
        self.latency = latency
        super().__init__()

    async def on_process_message(self, message: types.Message, data: dict):
        if not message.media_group_id:
            return

        try:
            self.album_data[message.media_group_id].append(message)
            raise CancelHandler()
        except KeyError:
            self.album_data[message.media_group_id] = [message]
            await asyncio.sleep(self.latency)

            message.conf["is_last"] = True
            data["album"] = self.album_data[message.media_group_id]

    async def on_post_process_message(self, message: types.Message, result: dict, data: dict):
        """Clean up after handling our album."""
        if message.media_group_id and message.conf.get("is_last"):
            del self.album_data[message.media_group_id]

```

Рисунок 3.4 – Приклад Middleware

Посередник реалізований у вигляді класу, який наслідується від BaseMiddleware. Метод on\_process\_message викликається після застосування фільтрів та перед виконанням обробника. Даний метод групує всі повідомлення MediaGroup в об'єкт album\_data за ідентифікатором та передає їх у обробник, реалізацію якого наведено на рисунку 3.5.

```
@dp.message_handler(is_media_group=True, content_types=types.ContentType.PHOTO, state=OrderState.photo )
async def order_photos_handler(msg: types.Message, album: List[types.Message], state: FSMContext):
    file_ids = list(map( lambda msg: msg['photo'][-1]['file_id'], album ))
    user_id = msg.from_user.id
    await state.update_data(photos=file_ids)
    data = await state.get_data()
    print(data)
    services.create_ad(data, user_id)
    text='<b>Об\`яву створено! Ви можете переглянути її в меню "Мої об\`яви"</b>'
    await bot.send_message( chat_id=user_id, text=text )
    await state.finish()
```

Рисунок 3.5 – Реалізація обробника для групи фото.

Внутрішній функціонал обробника побудовано на основі згорткової нейронної мережі LeNet 5. Фрагмент коду, що створює модель згорткової нейронної мережі LeNet 5 має такий вид:

```
def build_lenet(input_shape):
    # Define Sequential Model
    model = tf.keras.Sequential()

    # C1 Convolution Layer
    model.add(tf.keras.layers.Conv2D (filters=6, strides=(1,1),
kernel_size=(5,5), activation='tanh', input_shape=input_shape))

    # S2 SubSampling Layer
    model.add(tf.keras.layers.AveragePooling2D(pool_size=(2,2), strides=(2,2)))

    # C3 Convolution Layer
    model.add(tf.keras.layers.Conv2D (filters=6, strides=(1,1),
kernel_size=(5,5), activation='tanh'))

    # S4 SubSampling Layer
    model.add(tf.keras.layers.AveragePooling2D(pool_size=(2,2), strides=(2,2)))

    # C5 Fully Connected Layer
    model.add(tf.keras.layers.Dense(units=120, activation='tanh'))
```

```

# Flatten the output so that we can connect it with the fully connected
layers by converting it into a 1D Array
model.add(tf.keras.layers.Flatten())

# FC6 Fully Connected Layers
model.add(tf.keras.layers.Dense(units=84, activation='tanh'))

# Output Layer
model.add(tf.keras.layers.Dense(units=10, activation='softmax'))

# Compile the Model
model.compile(loss='categorical_crossentropy',
optimizer=tf.keras.optimizers.SGD(lr=0.1, momentum=0.0, decay=0.0),
metrics=['accuracy'])

return model

```

Цей фрагмент коду відповідає створенню згорткової нейронної мережі LeNet 5, архітектура шарів якої подана у Табл. 2.2.

### 3.5 Висновок до розділу 3

У третьому розділі було здійснено обґрунтування вибору мови програмування для реалізації інформаційної технології створення чат-бота для розміщення об'яв для товарів та послуг. Програмне забезпечення розроблено на об'єктно-орієнтованій мові програмування Python у середовищі програмування Visual Studio Code з використанням спеціалізованої бібліотеки Aiogram. Для створення згорткової нейронної мережі було використано бібліотеку Keras, а для створення бази даних було обрано SQLite. Детально описано програмну реалізацію основних модулів та алгоритмів системи, таких як Middleware та обробників запитів.



## 4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ЧАТ-БОТА ДЛЯ РОЗМІЩЕННЯ ІНФОРМАЦІЇ ПРО ТОВАРИ ТА ПОСЛУГИ

### 4.1 Тестування чат-бота для розміщення інформації про товари та послуги

Інструкція для користувача допоможе у використанні розробленої інформаційної технології чат-бота для розміщення інформації про товари та послуги. Користувач повин володіти такими навичками:

- володіння навичками роботи з операційними системами на яких можливі відтворення програмного забезпечення, або ж інших відворювальних засобів;
- володіння навичками роботи з месенджером Telegram на базовому рівні.

Після відкриття меню діалогу з ботом та натиснення кнопки “Start” почнеться процес реєстрації користувача з вибором регіону проживання, після чого в інтерфейсі з’явиться клавіатура для виклику меню, яке наведене на рисунку 4.1.

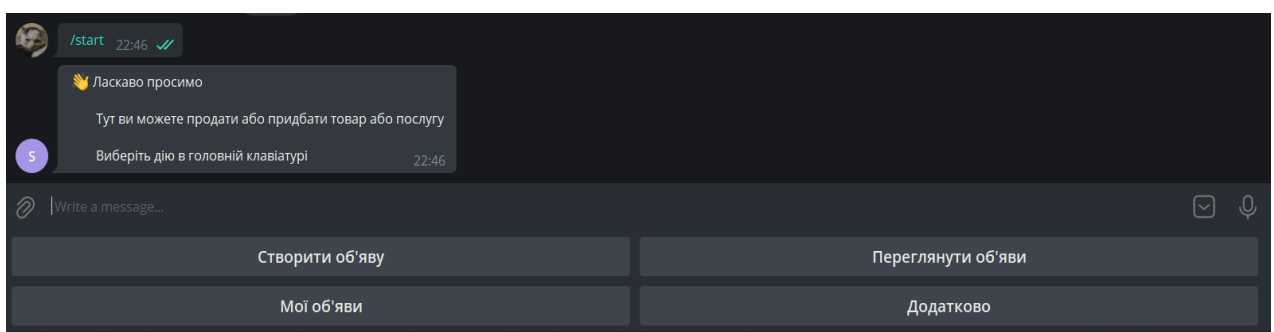


Рисунок 4.1 – Клавіатура головного меню та повідомлення з привітанням

Ми можемо створити об’яву в меню «Створити об’яву». Потрібно вибрати тип, категорію об’яви, та ввести заголовок, опис, ціну та відправити фото за бажанням.

Процес створення об'яви зображено на рисунку 4.2.

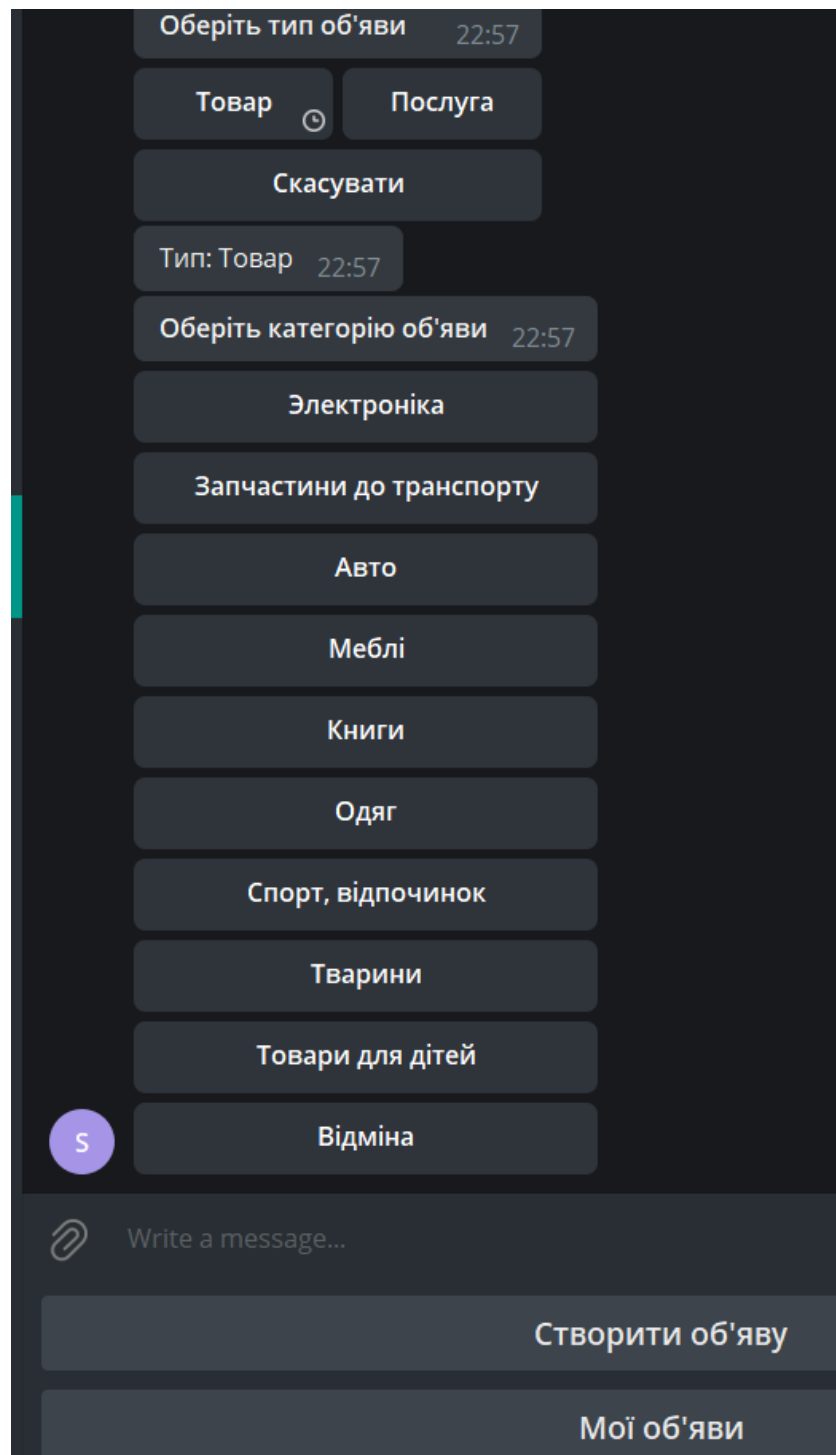


Рисунок 4.2 – Створення об'яви

В меню «Мої об'яви» можна переглянути свої об'яви за категорією та виконати над ними певні дії – редагувати або видалити (рис. 4.3)

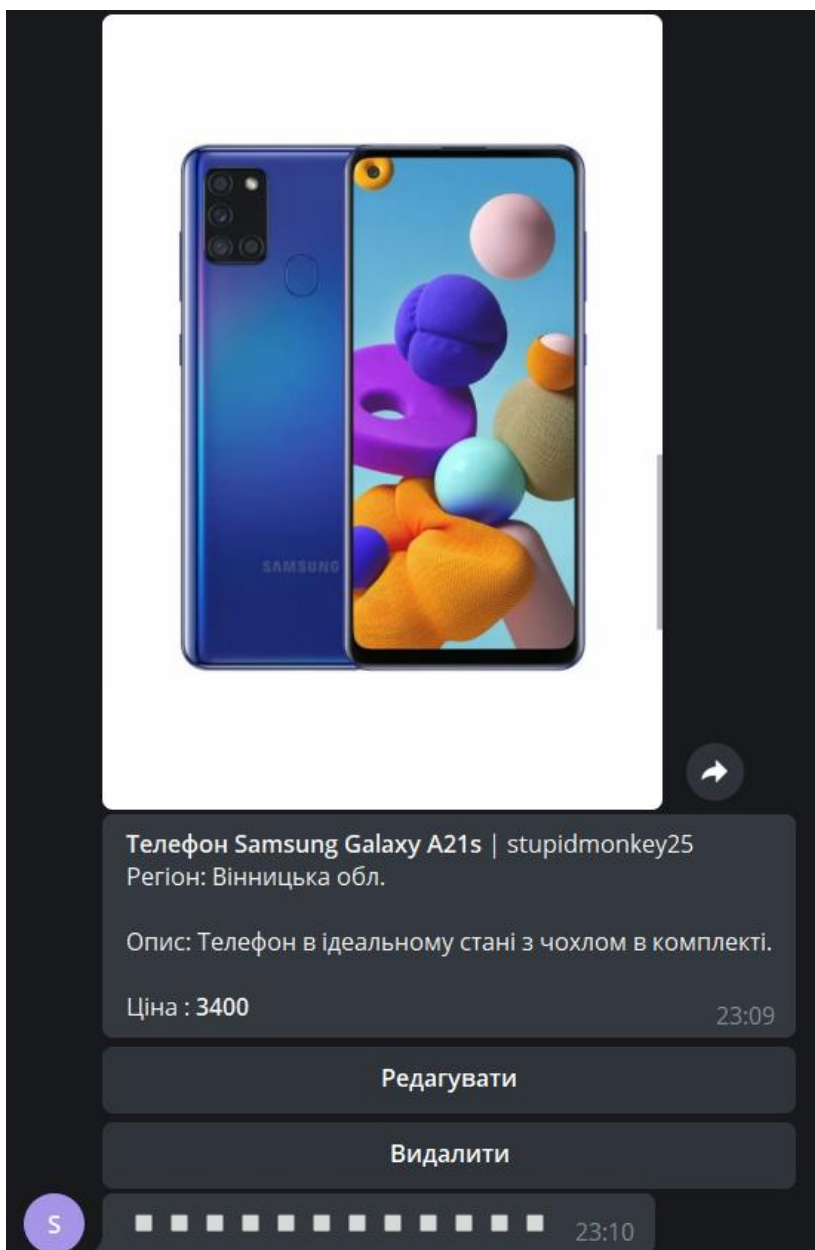


Рисунок 4.3 – Меню «Мої об’яви»

Переглянути всі об’яви можна в меню «Переглянути об’яви». Для цього потрібно обрати тип, категорію об’яви та регіон ( рис. 4.4).

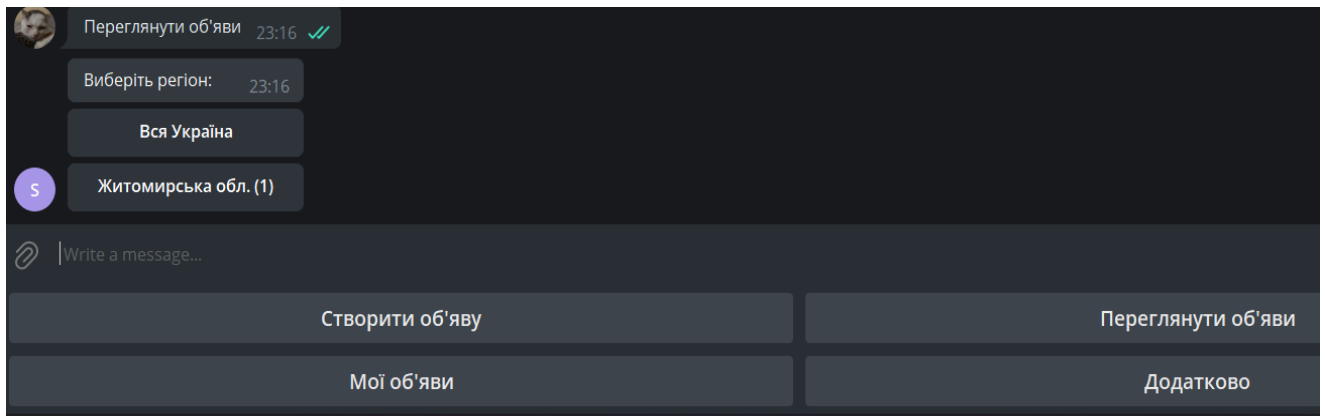


Рисунок 4.4 – Меню переглядання всіх об'яв

В меню «Додатково» знаходяться налаштування, в якому можна змінити регіон або видалити аккаунт, а також знайти об'яви за юзернеймом користувача.( рис. 4.5 )

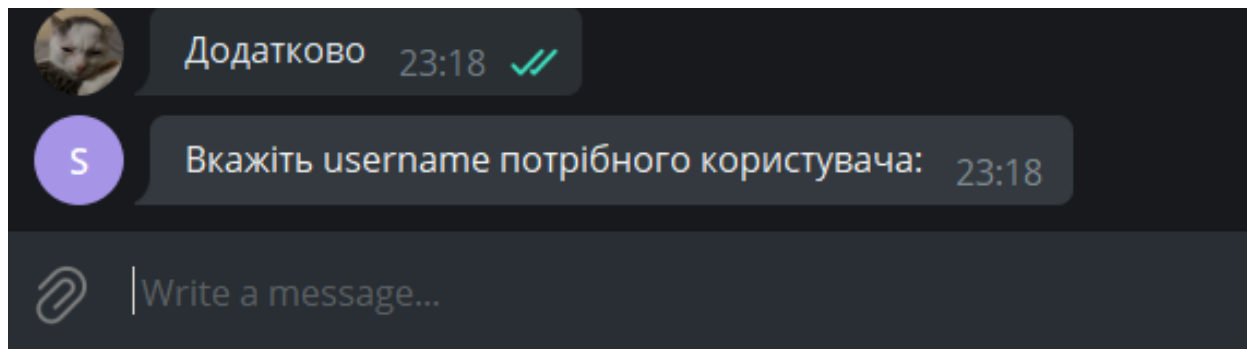


Рисунок 4.5 – Пошук об'яв за юзернеймом користувача.

З метою тестування чат-бота для розміщення інформації про товари та послуги було проведено реєстрацію для 5 користувачів, при цьому кожен користувач створив по 5 об'яв. Відповідно до проведених досліджень порівняємо розроблене програмне забезпечення з сервісами-аналогами за наступними критеріями: функція інтелектуального вибору відповіді, зручне редагування об'яви, зручне створення об'яв, пошук об'яв по імені користувача. Порівняння наведено у таблиці 4.1.

Таблиця 4.1 – Порівняння чат-ботів

Назва чат-бота	Ф-я інтелектуального вибору відповіді	Зручне редагування об'яв	Зручне створення об'яв	Пошук об'яв по імені користувача
OlxUaBot	-	+	+	-
RiaBot	-	-	+	-
Розроблений чат-бот	+	+	+	+

Таким чином, в результаті тестування чат-бота та відповідно до порівняльної таблиці сервісів для розміщення інформації про товари та послуги, можна зробити висновок про те, що було розширено функціонал чат-бота для розміщення інформації про товари та послуги.

#### **4.2 Аналіз результатів роботи чат-бота для розміщення інформації про товари та послуги**

Для порівняння якості роботи розробленого чат-бота на основі LeNet-5 з чат-ботом на основі згорткової нейромережі ResNet50, були проведені експериментальні дослідження його програмної реалізації.

Експериментальні дослідження були проведені на комп'ютері з процесором Intel Core i7-6700 (2,60 ГГц), 16 ГБ оперативної пам'яті, ОС Windows 10 Home (версія 20H2), відеокартою GeForce GTX 960M. Необхідний код був написаний на Python 3.8, а віртуальне середовище Anaconda також використовувалося для роботи з середовищем чат-бота, щоб уникнути локального запуску файлів на комп'ютері. Також була додана бібліотека NLTK.

Моделі згорткових нейронних мереж були навчені для впровадження чат-бота. Навчання проводилося для 200 епох на кожній моделі з використанням існуючого набору даних, який містив 826 питань разом із 352 унікальними відповідями. Як функція втрат використовувалась

«sparse\_categorical\_crossentropy». Як метод навчання нейронних мереж використовувався оптимізатор «Adam».

Для оцінки моделей використовувався тестовий набір даних. Тестовий набір містив 1/3 прикладів всього набору даних. Результати кожної архітектури після навчання для 200 епох подано у табл. 4.2.

Таблиця 4.2 – Результати роботи розробленого чат-бота та аналога

	Час навчання (на 200 епох)	Достовірність	F1- рахунок	Помилка
LeNet-5 (дана розробка)	22хв:00с	93,7 %	4,717	0,1548
ResNet50 (аналог)	8 год: 59 хв:	87,5 %	2,618	0,4388

У цьому дослідженні використовувалися згорткові нейронні мережі (CNN) як класифікатор для створення чат-бота та дослідження того, як різні архітектури впливають на різний час навчання та достовірність. Крім того, використовувався спеціальний інструмент для токенізації.

У таблиці 4.2 показано, що LeNet5 досяг найвищої точності та найменших втрат, вимагаючи мінімального часу навчання. Розроблений чат-бот на основі LeNet-5 має достовірність видачі вірних відповідей 93,7 %, а чат-бот аналог на основі ResNet50 – 87,5 %. Тобто мета роботи досягнута – достовірність видачі вірних відповідей підвищена на 6,2 %.

Слід зазначити, що достовірність чат-бота видачі вірних відповідей залежить від якості підготовки навчальних та тестових наборів даних.

### 4.3 Висновок до розділу 4

У результаті тестування розробленого чат-бота було доведено його повну працездатність та відповідність поставленому завданню. Розроблений бот володіє такими функціональними можливостями: створення об'яви за категорією; перегляд всіх об'яв; перегляд своїх об'яв; редагування своїх об'яв. Розроблений чат-бот на основі згорткової нейронної мережі LeNet5 має найвищу достовірність видачі вірних відповідей та найменших втрат, вимагаючи мінімального часу навчання. Розроблений чат-бот на основі LeNet-5 має достовірність видачі вірних відповідей 93,7 %, а чат-бот аналог на основі ResNet50 – 87,5 %. Тобто мета роботи досягнута – достовірність видачі вірних відповідей підвищена на 6,2 %.

## 5 ЕКОНОМІЧНА ЧАСТИНА

### 5.1 Проведення комерційного та технологічного аудиту інформаційної технології створення чат-бота для розміщення об'яв для товарів та послуг

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту [24,25] залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 5.1.

Таблиця 5.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології створення чат-бота для розміщення об'яв для товарів та послуг

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	1	2	2
Ринкові переваги (наявність аналогів)	2	3	2
Ринкові переваги (ціна продукту)	3	4	3
Ринкові переваги (технічні властивості)	2	3	3
Ринкові переваги (експлуатаційні витрати)	1	2	1
Ринкові перспективи (розмір ринку)	2	3	3
Ринкові перспективи (конкуренція)	3	4	3
Практична здійсненність (наявність фахівців)	2	3	2



Продовження табл. 5.1

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Практична здійсненність (наявність фінансів)	1	2	2
Практична здійсненність (необхідність нових матеріалів)	2	2	2
Практична здійсненність (термін реалізації)	3	2	3
Практична здійсненність (розробка документів)	1	2	1
Сума балів	23	32	27
Середньоарифметична сума балів, СБ	27		

За результатами розрахунків, наведених в таблиці 5.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології створення чат-бота для розміщення об'яв для товарів та послуг – середній.

## 5.2 Розрахунок витрат на здійснення науково-дослідної роботи

*Витрати на оплату праці.* Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми [24,25], обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

*Основна заробітна плата дослідників.* Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p},$$

де  $k$  – кількість посад дослідників, залучених до процесу дослідження;  $M_{ni}$  – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;  $T_p$  – число робочих днів в місяці; приблизно  $T_p = (21 \dots 23)$  дні;  $t_i$  – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 5.2.

Таблиця 5.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	39000	1773	7	12411
Розробник	34000	1545	22	33990
Всього:				46401

*Додаткова заробітна плата.* Додаткова заробітна плата  $Z_d$  всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,1 \cdot (Z_o + Z_p) = 0,1 \cdot (46401 + 0) = 4640,1 \text{ грн.}$$

*Відрахування на соціальні заходи.* Нарахування на заробітну плату  $N_{зп}$  розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned}
 H_{зп} &= \beta \cdot (З_о + З_p + З_д) = \\
 &= 0,22 \cdot (46401 + 4640,1 + 0) = 11229 \text{ грн.}
 \end{aligned}$$

де  $З_о$  – основна заробітна плата розробників, грн.;  $З_p$  – основна заробітна плата робітників, грн.;  $З_д$  – додаткова заробітна плата всіх розробників та робітників, грн.;  $\beta$  – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

*Програмне забезпечення.* До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i,$$

де  $C_{\text{іпрг}}$  – ціна придбання програмного забезпечення  $i$ -го виду, грн.;  $C_{\text{прг.і}}$  – кількість одиниць програмного забезпечення відповідного виду, шт.;  $K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення,  $K_i = (1,1 \dots 1,12)$ ;  $k$  – кількість видів програмного забезпечення.

Таблиця 5.3 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Visual Studio Code, мова програмування Python	-	-	-
Всього, з врахуванням коефіцієнта інсталяції та налагодження			0

*Амортизація обладнання.* Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування  $A$  в цілому бути розраховані за формулою:

$$A = \frac{Ц_б}{T_в} \cdot \frac{t}{12},$$

де  $Ц_б$  – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.;  $t$  – термін використання основного фонду, місяці;  $T_в$  – термін корисного використання основного фонду, роки.

Таблиця 5.4 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	39000	5	1	650
Всього			650	

*Витрати на електроенергію для науково-виробничих цілей.* Витрати на силову електроенергію  $B_e$ , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 5.5 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук		

$$Be = \sum \frac{W_i \cdot t_i \cdot Ce \cdot K_{впн}}{ККД} = \frac{0,17 \cdot 150 \cdot 7,5 \cdot 0,9}{0,98} = 632 \text{ грн.},$$

$W_i$  – встановлена потужність обладнання, кВт;  $t_i$  – тривалість роботи обладнання на етапі дослідження, год.;  $Ce$  – вартість 1 кВт електроенергії, грн.;  $K_{впн}$  – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

*Інші витрати.* До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{в} = (Z_o + Z_p) \cdot \frac{N_{I_{в}}}{100\%} = (46401 + 0) \cdot \frac{50}{100} = 23201 \text{ грн.},$$

де  $N_{I_{в}}$  – норма нарахування за статтею «Інші витрати».

*Накладні (загальновиробничі) витрати.* До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{НЗВ}} = (Z_o + Z_p) \cdot \frac{N_{\text{НЗВ}}}{100\%} = (46401 + 0) \cdot \frac{150}{100} = 69602 \text{ грн.},$$

де  $N_{\text{НЗВ}}$  – норма нарахування за статтею «Накладні (загально виробничі) витрати».

*Витрати на проведення науково-дослідної роботи.* Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} V_{\text{заг}} &= Z_o + Z_{\text{дод}} + Z_n + V_{\text{прг}} + A_{\text{обл}} + V_e + \\ &\quad + I_v + V_{\text{НЗВ}} = \\ &= 46401 + 4640,1 + 11229 + 0 + 650 + 632 + 23201 + 69602 \\ &= 156355 \text{ грн.} \end{aligned}$$

*Загальні витрати.* Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\eta} = \frac{156355}{0,5} = 312710 \text{ грн.},$$

де  $\eta$  – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії розробки дослідного зразка, то  $\eta = 0,5$ .

### **5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором**

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження

результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати:  $\Delta N$  – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу;  $N$  – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки;  $C_0$  – вартість послуги у році до впровадження інформаційної системи;  $\pm\Delta C_0$  – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta\Pi$  для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де  $\pm\Delta C$  – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки);  $\pm\Delta C_0$  може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни);  $N$  – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;  $C_0$  – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році;  $C_0$  – основний якісний показник, який визначає ціну реалізації існуючої (базової)

науково-технічної розробки у році до впровадження результатів;  $\Delta N$  – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки);  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт  $\lambda = 0,8333$ ;  $\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати  $\rho = 0,2 \dots 0,5$ ;  $\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році  $\vartheta = 18\%$ .

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta\Pi = ((3800 - 3500) \cdot 12000 - 3800 \cdot (12000 - 12000)) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = 73770480 \text{ грн.}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t} = \frac{73770480}{(1 + 0,1)^1} = 67064073 \text{ грн.,}$$

де  $\Delta\Pi$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.;  $T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо  $T=1$  рік);  $\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;  $t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.



Далі розраховують величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 10 \cdot 312710 = 3127100 \text{ грн.}$$

де  $k_{\text{інв}}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{\text{інв}}=2\dots5$ , але може бути і більшим;  $ЗВ$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект  $E_{\text{абс}}$  або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 67064073 - 3127100 = 63936973 \text{ грн.},$$

де  $\text{ПП}$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.;  $PV$  – теперішня вартість початкових інвестицій, грн.

Оскільки  $E_{\text{абс}} > 0$ , то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність  $E_{\text{в}}$  або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій  $E_B$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{Ea_{бс}}{PV}} = \sqrt{1 + \frac{63936973}{3127100}} = 2,7,$$

де  $T_{ж}$  – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування  $\tau_{\text{мін}}$ , тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{\text{мін}}$  визначається за формулою:

$$\tau_{\text{мін}} = d + f = 0,9 + 0,5 = 1,4,$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = 0,9 \dots 0,12$ ;  $f$  – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина  $f = 0,05 \dots 0,5$ , але може бути і значно вищою.

Оскільки  $E_B = 2,7 > \tau_{\text{мін}} = 1,4$ , то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій  $T_o$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_B} = \frac{1}{2,7} = 0,4 \text{ року.}$$

Оскільки  $T_o=0,4 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

#### **5.4 Висновок до розділу 5**

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія створення чат-бота для розміщення об'яв для товарів та послуг» становить 27,0 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 0,4 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія створення чат-бота для розміщення об'яв для товарів та послуг».

## ВИСНОВКИ

При виконанні даної магістерської кваліфікаційної роботи розв'язано задачу розробки інформаційної технології та програмного забезпечення для створення чат-ботів для розміщення інформації про товари та послуги з використанням згорткової нейронної мережі.

В першому розділі проаналізовано предметну область створення чат-ботів для розміщення інформації про товари та послуги, перелічено основний функціонал, переваги та недоліки. Зроблено висновок, що ринок потребує новий продукт, який зможе об'єднати переваги та позбутись недоліків. Проведено аналіз платформ, які підтримують роботу чат-ботів. Здійснено постановку задачі, в якій акцентовано увагу на функціях та можливостях, якими повинен володіти програмний продукт,

У другому розділі визначено, якими будуть особливості розроблюваної інформаційної технології, розроблено архітектуру проекту. Розроблено інтелектуального обробника для чат-бота на основі згорткової нейромережі, обрано архітектуру згорткової нейронної мережі LeNet 5, розроблено схему бази даних та основні алгоритми для функціонування чат-бота для розміщення інформації про товари та послуги

У третьому розділі було здійснено обґрунтування вибору мови програмування для реалізації інформаційної технології створення чат-бота для розміщення об'яв для товарів та послуг. Програмне забезпечення розроблено на об'єктно-орієнтованій мові програмування Python у середовищі програмування Visual Studio Code з використанням спеціалізованої бібліотеки Aiogram. Для створення згорткової нейронної мережі було використано бібліотеку Keras , а для створення бази даних було обрано SQLite. Детально описано програмну реалізацію основних модулів та алгоритмів системи, таких як Middleware та обробників запитів.

У четвертому розділі у результаті тестування розробленого чат-бота було доведено його повну працездатність та відповідність поставленому завданню. Розроблений бот володіє такими функціональними можливостями: створення об'яви за категорією; перегляд всіх об'яв; перегляд своїх об'яв; редагування своїх об'яв. Розроблений чат-бот на основі згорткової нейронної мережі LeNet5 має найвищу достовірність видачі вірних відповідей та найменших втрат, вимагаючи мінімального часу навчання. Розроблений чат-бот на основі LeNet-5 має достовірність видачі вірних відповідей 93,7 %, а чат-бот аналог на основі ResNet50 – 87,5 %. Тобто мета роботи досягнута – достовірність видачі вірних відповідей підвищена на 6,2 %.

У п'ятому розділі було визначено рівень комерційного потенціалу розробки, який становить 27,0 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 0,4 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Голоднюк В. О., Паночишин Ю. М. «Інформаційна технологія створення чат-бота для розміщення об'яв для товарів та послуг», в Матеріалах Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)», Вінниця, 2024, [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19742/16342>.
2. Telegram Bot API Documentation – [Електронний ресурс] .// Режим доступу: <https://core.telegram.org/bots/api>
3. Aiogram's documentation – [Електронний ресурс] .// Режим доступу: <https://docs.aiogram.dev/en/latest/>
4. Бот(програма)–[Електронний ресурс] .// Режим доступу: [https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D1%82\\_\(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0\)](https://ru.wikipedia.org/wiki/%D0%91%D0%BE%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)).
5. Physics is Simple – [Електронний ресурс] .// Режим доступу: <https://www.youtube.com/c/PhysicsisSimple>
6. Створення Python Telegram бота та його deploy на віртуальну машину – [Електронний ресурс] .// Режим доступу: <https://habr.com/post/549962/>
7. SQLite– [Електронний ресурс] .// Режим доступу: <https://metanit.com/sql/sqlite/>
8. Telegram bot на aiogram– [Електронний ресурс] .// Режим доступу: <https://habr.com/post/662632/>
9. Все, про що має знати розробник Телеграм-ботів– [Електронний ресурс] .// Режим доступу: <https://habr.com/post/543676/>
10. Baby, C.J.; Khan, F.A.; Swathi, J.N. Home Automation Using IoT and a Chatbot Using Natural Language Processing. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT); April 2017; pp. 1–6.

11. Bengfort, B.; Bilbro, R.; Ojeda, T. Applied Text Analysis with Python: Enabling LanguageAware Data Products with Machine Learning; O'Reilly Media, Inc., 2018; ISBN 978-1-4919-6299-2.

12. Manaswi, N.K.; Manaswi, N.K.; John, S. Deep Learning with Applications Using Python; Springer, 2018.

13. Nadeau, D.; Sekine, S. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes* 2007, 30, 3–26.

14. Muhammad Rizwan Khan LeNet-5 — A Classic CNN Architecture – [Електронний ресурс]. Режим доступу: <https://medium.datadriveninvestor.com/lenet-5-a-classic-cnn-architecture-c87d0b03560d/>

15. Tensorflow [Електронний ресурс]. – Режим доступу://www.tensorflow.org/.

16. Keras [Електронний ресурс]. – Режим доступу://keras.io/.

17. SQLite Python – [Електронний ресурс]// Режим доступу: <https://www.sqlitetutorial.net/sqlite-python/>

18. Кривництво по мові програмування Python– [Електронний ресурс] // Режим доступу: <https://metanit.com/python/tutorial/>

19. Асинхронність в Python– [Електронний ресурс] // Режим доступу: <https://highload.today/asinhronnost-v-python-kak-twitter-obrabatyvaet-milliardy-seansov-vden>.

20. Visual Studio Code: потужне керівництво користувача – [Електронний ресурс] // Режим доступу: <https://dev-gang.com/article/visual-studio-code-moscznoe-rukovodstvo-polzovatelja-dwedy9wjzg/>

21. Documentation for Visual Studio Code – [Електронний ресурс] // Режим доступу: <https://code.visualstudio.com/docs>

22. Middleware – [Електронний ресурс] - <https://en.wikipedia.org/wiki/Middleware>

23. Middlewares aiogram – [Електронний ресурс] // Режим доступу: <https://docs.aiogram.dev/en/dev-3.x/dispatcher/middlewares.html>

24.Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

25.Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

26.Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (PDF, 58 с.)



## Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія створення чат-бота для розміщення об'яв для товарів та послуг

Тип роботи: магістерська кваліфікаційна робота  
(БДР, МКР)

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

## Показники звіту подібності Unicheck

Оригінальність 90,47% Схожість 9,53%

## Аналіз звіту подібності (відмітити потрібне):

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

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

Автор роботи  Голоднюк В.О.

Керівник роботи  Паночишин Ю.М.

## Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку  Озеранський В.С.

**Додаток Б (обов'язковий)****Лістинг програми**

main.py

```
import config
from aiogram import Bot
from aiogram.utils import executor
from aiogram.dispatcher import Dispatcher
from aiogram.contrib.fsm_storage.memory import MemoryStorage

import asyncio
import Middlwares

bot = Bot(token=config.TOKEN, parse_mode='html')

dp = Dispatcher(bot=bot, storage=MemoryStorage())

async def on_startup(dp):
    print('BOT was started')

async def on_shutdown(dp: Dispatcher):
    print('BOT was stopped')

if __name__ == '__main__':
    dp.middleware.setup(Middlwares.AlbumMiddleware())
    from handlers import dp
    loop = asyncio.get_event_loop()

    executor.start_polling(
        dispatcher=dp,
        loop=loop,
        on_startup=on_startup,
        on_shutdown=on_shutdown,
        skip_updates=True
    )
```

```

create_ad_handlers.py
from aiogram.dispatcher import FSMContext
from aiogram import types
from typing import List
from main import dp, bot
from states import OrderState
import keyboards
import services
from DataBase import category_db

```

```

@dp.message_handler(text='Створити об\`яву')
async def create_ad_handler(msg: types.Message):
    text = '<b>Оберіть тип об\`яви</b>'
    await bot.send_message(chat_id=msg.from_user.id, text=text,
reply_markup=keyboards.create_order_types_keyboard())
    await OrderState.type.set()

```

```

@dp.callback_query_handler(state=OrderState.type)
async def create_ad_type_handler(call: types.CallbackQuery, state: FSMContext):
    type = call.data.split('_')[2]
    type_text = 'Товар' if type=='product' else 'Послуга'
    await state.update_data(type=type)
    categoryes = category_db.get_categoryes_by_type(type)
    text='<b>Оберіть категорію об\`яви</b>'
    keyboard = keyboards.create_order_categoryes_keyboard(categoryes)
    await bot.send_message(chat_id=call.from_user.id, text=f'Тип: {type_text}')
    await bot.send_message(chat_id=call.from_user.id, text=text,
reply_markup=keyboard )
    await OrderState.next()

```

```

@dp.callback_query_handler(state=OrderState.category)
async def create_ad_category_handler(call: types.CallbackQuery, state:
FSMContext):
    category_id = call.data.split('_')[3]
    category_name = category_db.get_category_by_id(category_id)['name']
    await state.update_data(category=category_id)

    text='<b>Укажіть заголовок об\`яви</b>'
    keyboard = keyboards.cancel_keyboard()

```

```

    await bot.send_message( chat_id=call.from_user.id, text=f'Категорія:
{category_name}')
    await bot.send_message( chat_id=call.from_user.id, text=text,
reply_markup=keyboard )
    await OrderState.next()

```

```

@dp.message_handler(state=OrderState.title)
async def create_ad_title_handler(msg: types.Message, state: FSMContext):
    title = msg.text
    await state.update_data(title=title)
    text='<b>Укажіть опис об\`яви</b>'
    keyboard = keyboards.cancel_keyboard()
    await bot.send_message( chat_id=msg.from_user.id, text=text,
reply_markup=keyboard )
    await OrderState.next()

```

```

@dp.message_handler(state=OrderState.description)
async def create_ad_description_handler(msg: types.Message, state: FSMContext):
    description = msg.text
    await state.update_data(description=description)
    text='<b>Укажіть ціну(грн), або натисніть кнопку "Договірна". </b>'
    keyboard = keyboards.create_ad_price_keyboard()
    await bot.send_message( chat_id=msg.from_user.id, text=text,
reply_markup=keyboard )
    await OrderState.next()

```

```

@dp.message_handler(state=OrderState.price)
async def create_ad_price_handler(msg: types.Message, state: FSMContext):
    price = msg.text
    try:
        price = int(price)
    except ValueError:
        await bot.send_message(chat_id=msg.from_user.id, text='Ціна повинна бути
невід\`ємним числом')
        return
    if price < 0:
        await bot.send_message(chat_id=msg.from_user.id, text='Ціна повинна бути
невід\`ємним числом')
        return
    await state.update_data(price=price)
    text='<b>Добавте одне або декілька фото, або ніжміть кнопку "Створити
об\`яву". </b>'

```

```

keyboard = keyboards.create_ad_photo_keyboard()
await bot.send_message( chat_id=msg.from_user.id, text=text,
reply_markup=keyboard )
await OrderState.next()

```

```

@dp.callback_query_handler(state=OrderState.price)
async def create_ad_price_handler(call: types.Message, state: FSMContext):
    price = -1
    await state.update_data(price=price)
    text='<b>Добавте одне або декілька фото, або ніжміть кнопку "Створити
об\`яву". </b>'
    keyboard = keyboards.create_ad_photo_keyboard()
    await bot.send_message( chat_id=call.from_user.id, text='Ціна: договірна.')
    await bot.send_message( chat_id=call.from_user.id, text=text,
reply_markup=keyboard )
    await OrderState.next()

```

```

@dp.callback_query_handler(state=OrderState.photo)
async def create_ad_none_photo_handler(call: types.Message, state: FSMContext):

    user_id = call.from_user.id
    await state.update_data(photos=[])
    data = await state.get_data()
    print(data)
    services.create_ad(data, user_id)
    text='<b>Об\`яву створено! Ви можете переглянути її в меню "Мої
об\`яви"</b>'
    await bot.edit_message_text( message_id=call.message.message_id,
chat_id=call.from_user.id, text=text )
    await state.finish()

```

```

@dp.message_handler(is_media_group=True,
content_types=types.ContentType.PHOTO, state=OrderState.photo )
async def order_photos_handler(msg: types.Message, album: List[types.Message],
state: FSMContext):
    file_ids = list(map( lambda msg: msg['photo'][-1]['file_id'], album ))
    user_id = msg.from_user.id
    await state.update_data(photos=file_ids)

```

```

data = await state.get_data()
services.create_ad(data, user_id)
text='<b>Об\`яву створено! Ви можете переглянути її в меню "Мої
об\`яви"</b>'
await bot.send_message( chat_id=user_id, text=text )
await state.finish()

```

```

@dp.message_handler(content_types=types.ContentType.PHOTO,
state=OrderState.photo )
async def order_photo_handler(msg: types.Message, state: FSMContext):
    file_id = msg['photo'][-1]['file_id']
    user_id = msg.from_user.id
    await state.update_data(photos=[file_id])
    data = await state.get_data()
    services.create_ad(data, user_id)
    text='<b>Об\`яву створено! Ви можете переглянути її в меню "Мої
об\`яви"</b>'
    await bot.send_message( chat_id=user_id, text=text )

    await state.finish()

}

```

## Додаток В (обов'язковий)

## ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТВОРЕННЯ ЧАТ-БОТА ДЛЯ  
РОЗМІЩЕННЯ ОБ'ЯВ ДЛЯ ТОВАРІВ ТА ПОСЛУГ

Виконав: студент 2-го курсу,  
групи 1КН-22м  
спеціальності 122 «Комп'ютерні науки»  
(шифр і назва напрямку підготовки, спеціальності)  
Голоднюк В. О.  
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН  
Паночишин Ю.М.  
(прізвище та ініціали)  
« 04 » 12 2023 р.

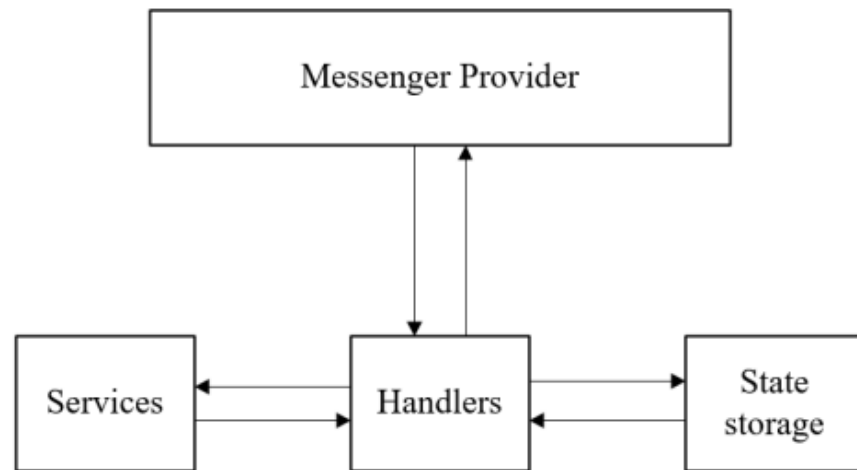


Рисунок В.1 – Узагальнений приклад серверної архітектури бота

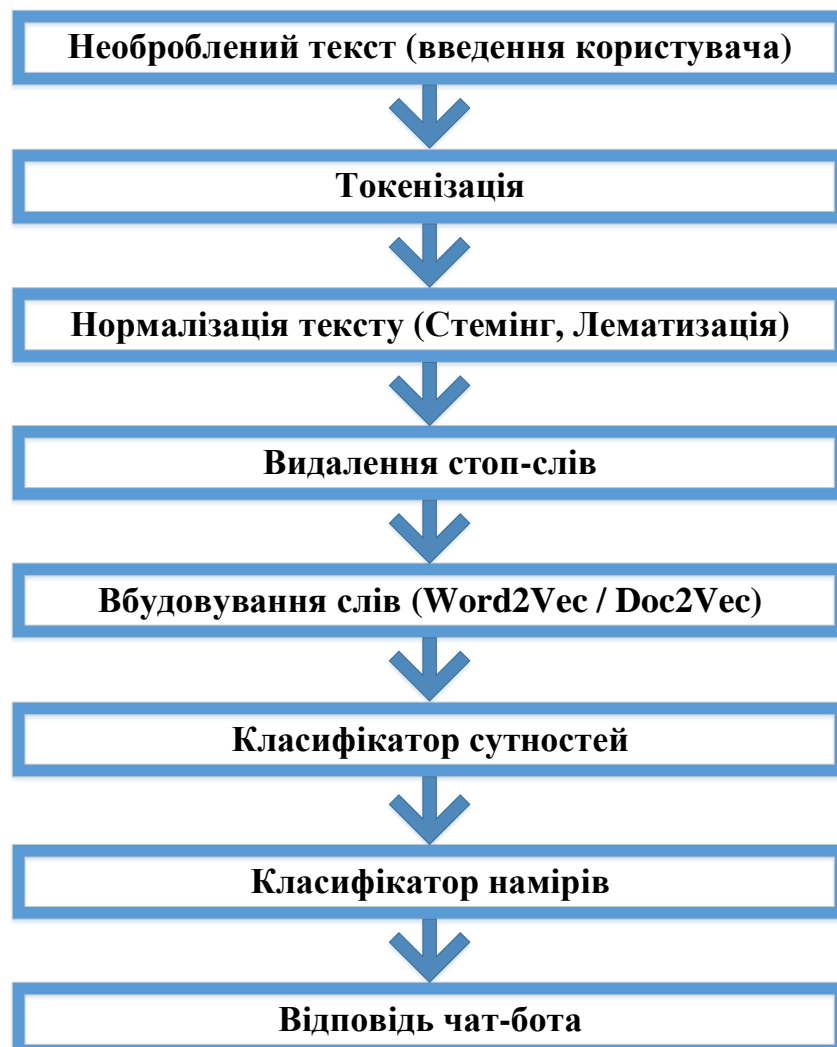
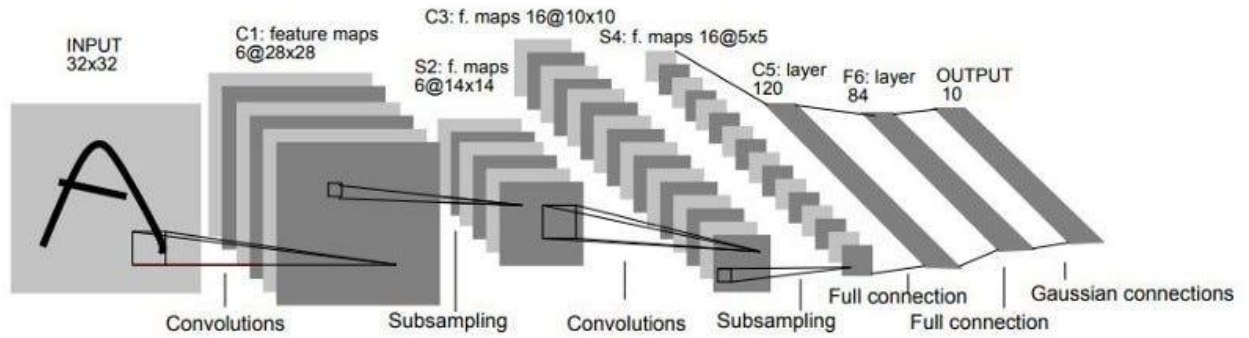


Рисунок В.2 – Структурна схема етапів обробки тексту користувача у обробнику (класифікаторі) на основі згорткової нейромережі





Шар		Карта ознак	Розмір	Розмір ядра	Крок	Активаційна функція
Вхідний		1	$32 \times 32$	-	-	-
1	Згортка	6	$28 \times 28$	$5 \times 5$	1	tanh
2	Усереднений пулінг	6	$14 \times 14$	$2 \times 2$	2	tanh
3	Згортка	16	$10 \times 10$	$5 \times 5$	1	tanh
4	Усереднений пулінг	16	$5 \times 5$	$2 \times 2$	2	tanh
5	Згортка	120	$1 \times 1$	$5 \times 5$	1	tanh
6	Повнозв'язний	-	84	-	-	tanh
Вихідний	Повнозв'язний	-	10	-	-	softmax

Рисунок В.3 – Архітектура згорткової нейромережі LeNet-5

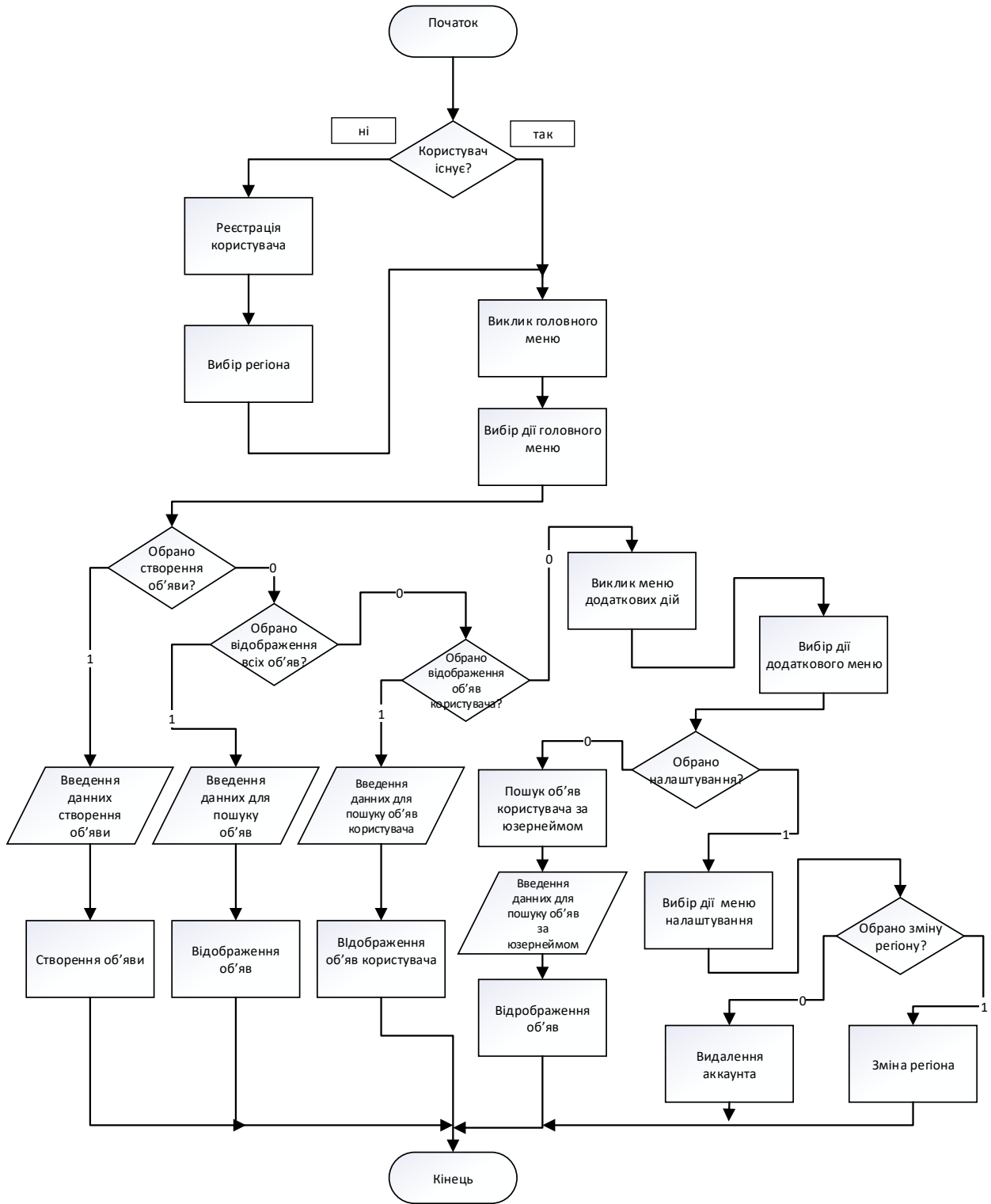


Рисунок В.4 – Граф-схема загального алгоритму роботи чат-бота

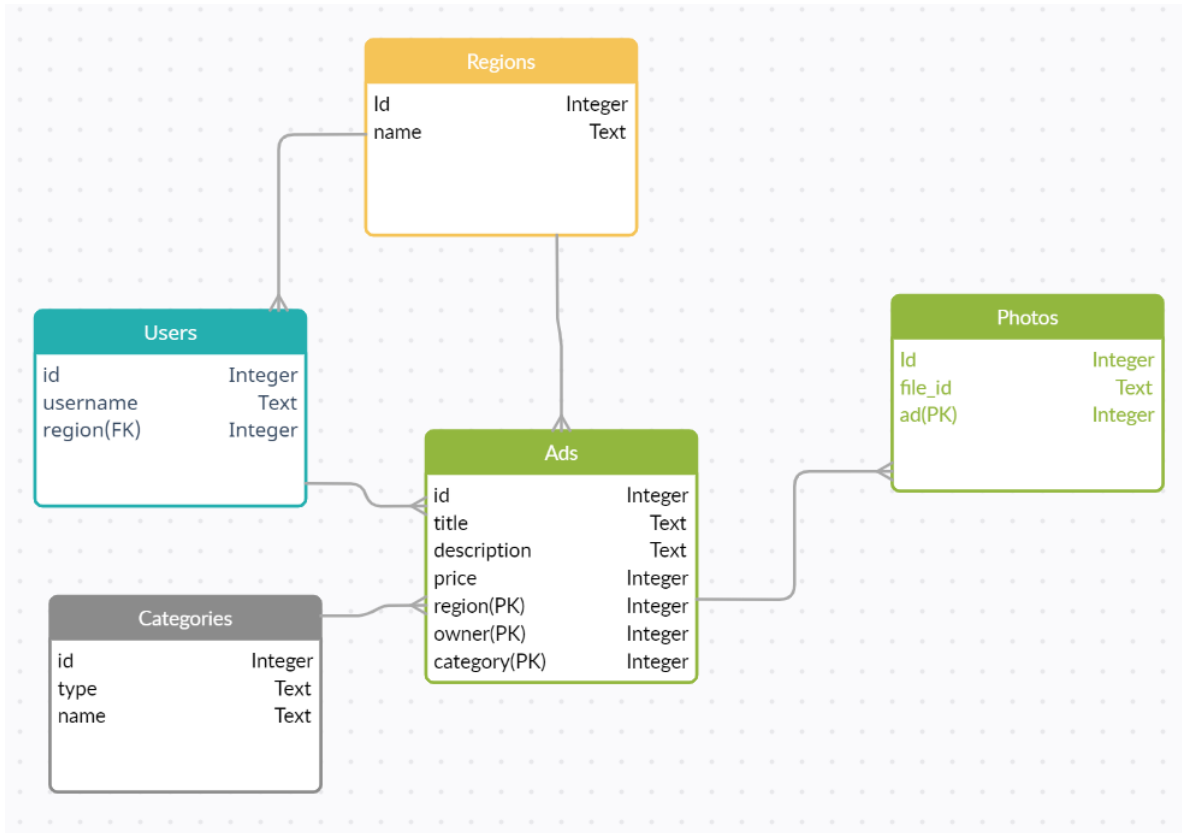


Рисунок В.5 – Схема бази даних телеграм бота для розміщення інформації про товари та послуги

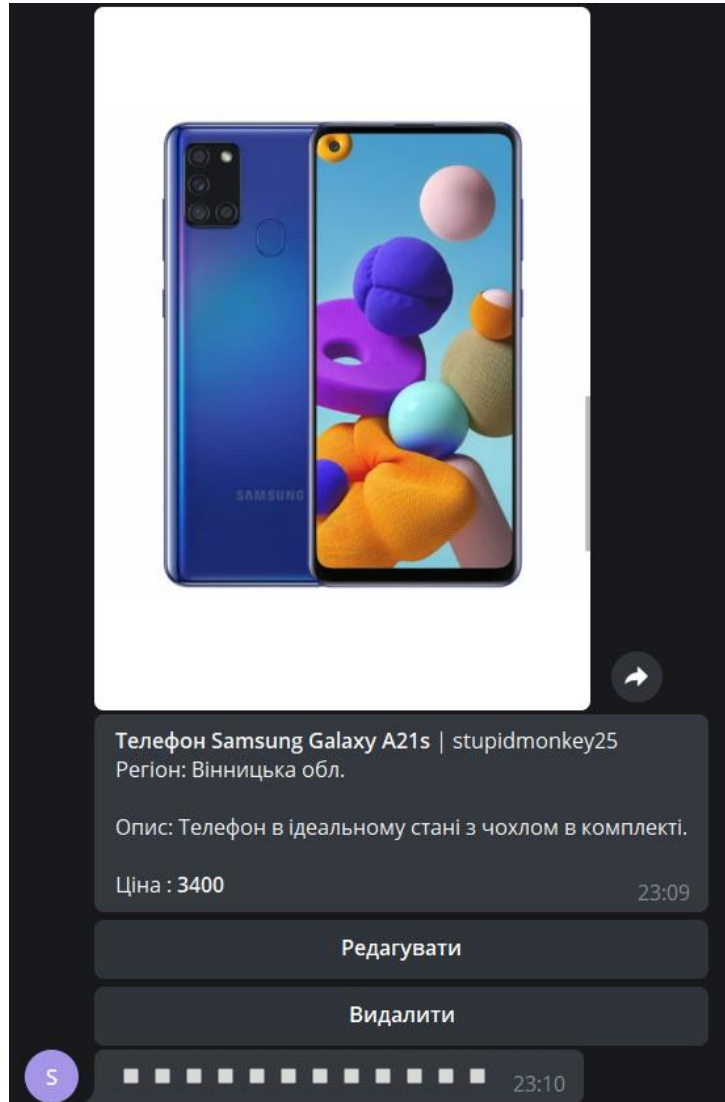
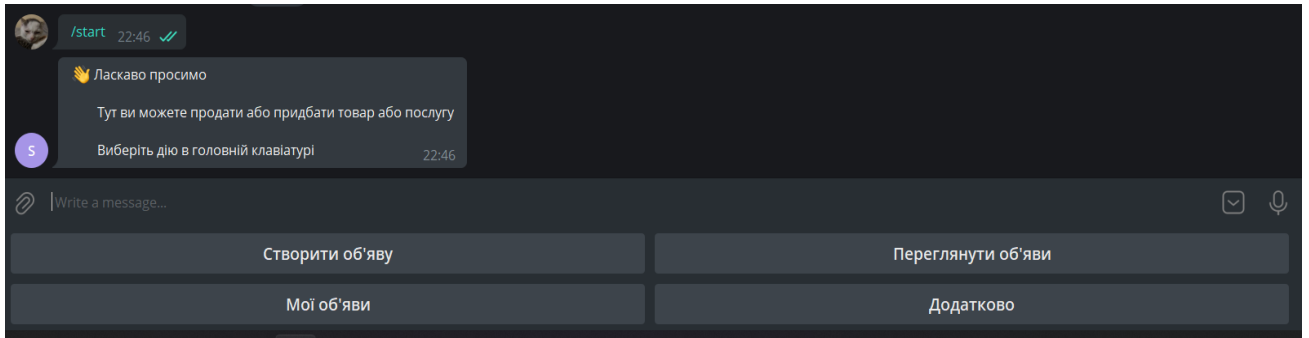


Рисунок В.6 – Робочі вікна чат-бота

## Додаток Г (довідниковий)

### Інструкція користувача

Після відкриття меню діалогу з ботом та натиснення кнопки “Start” почнеться процес реєстрації користувача з вибором регіону проживання, після чого в інтерфейсі з’явиться клавіатура для виклику меню (Рисунок Г.1).

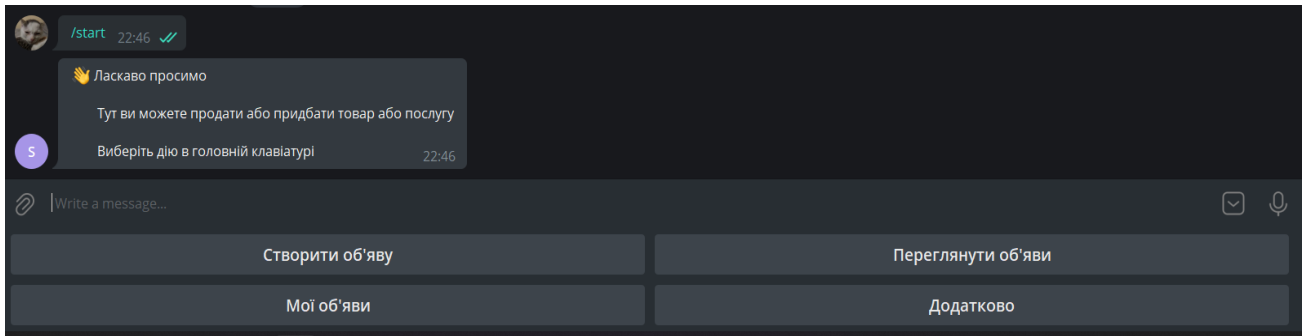


Рисунок Г.1 – Клавіатура головного меню та повідомлення з привітанням

Ми можемо створити об’яву в меню «Створити об’яву». Потрібно вибрати тип, категорію об’яви, та ввести заголовок, опис, ціну та відправити фото за бажанням.

Процес створення об’яви зображено на рисунку Г.2.

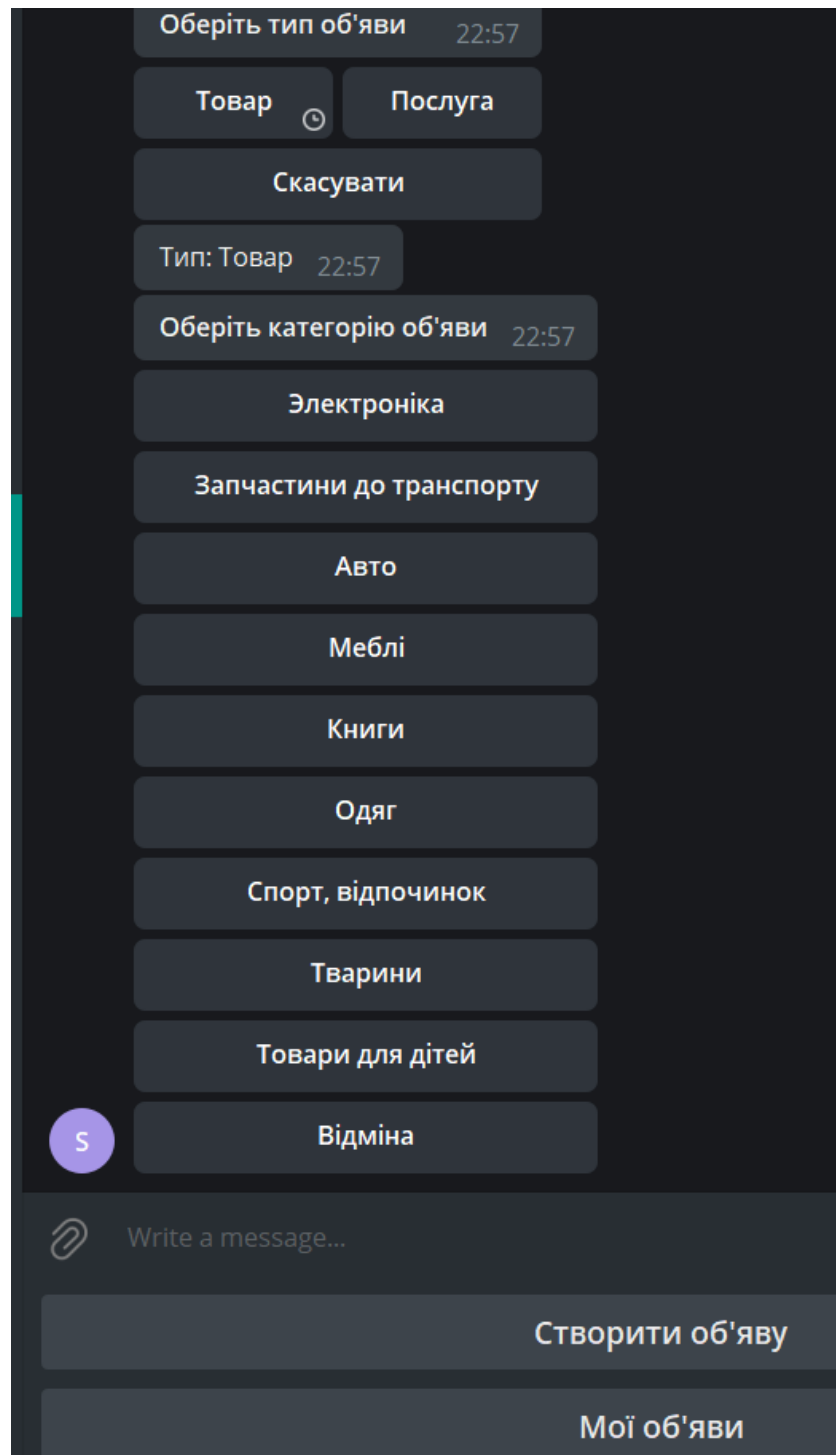


Рисунок Г.2 – Створення об'яви

В меню «Мої об'яви» можна переглянути свої об'яви за категорією та виконати над ними певні дії – редагувати або видалити ( рисунок Г.3 )

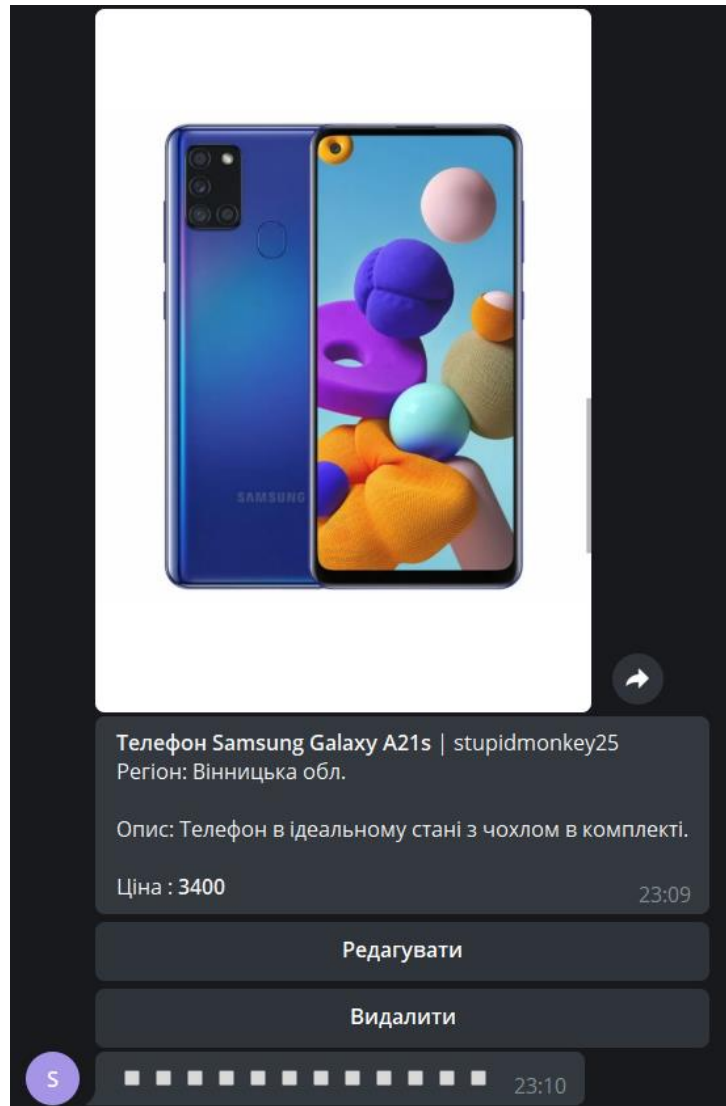


Рисунок Г.3 – Меню «Мої об'яви»

Переглянути всі об'яви можна в меню «Переглянути об'яви». Для цього потрібно обрати тип, категорію об'яви та регіон ( рис. Г.4).

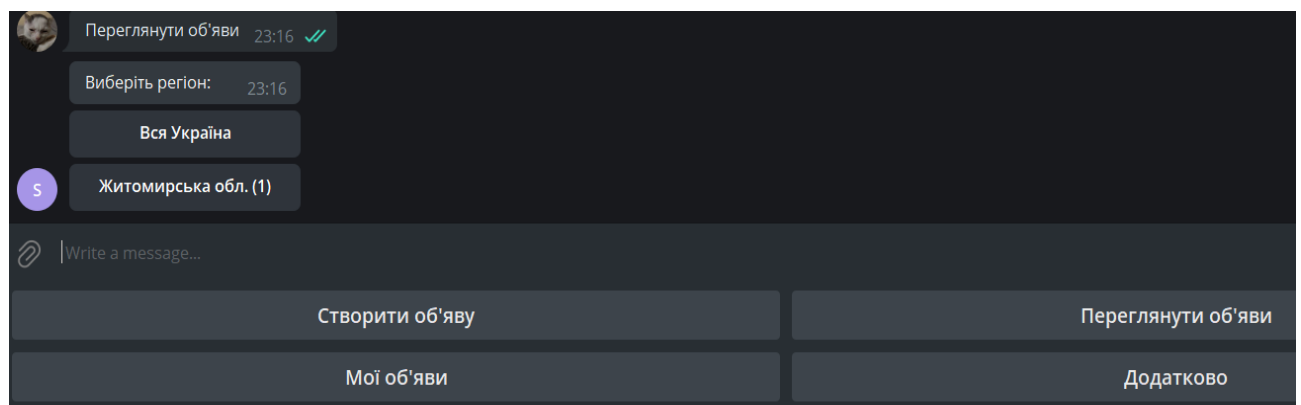


Рисунок Г.4 – Меню переглядання всіх об'яв

В меню «Додатково» знаходяться налаштування, в якому можна змінити регіон або видалити аккаунт, а також знайти об'яви за юзернеймом користувача.( рис. Г.5 )

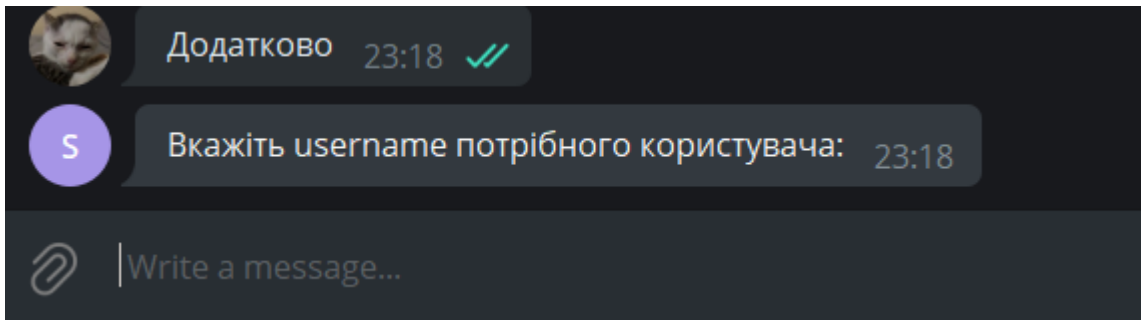


Рисунок Г.5 – Пошук об'яв за юзернеймом користувача.