

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**  
на тему:  
**«Технологія інтеграції чат-ботів у web-додатки»**  
**ПОЯСНЮВАЛЬНА ЗАПИСКА**

Виконав: студент 2 курсу, групи 1КІ-21м  
напряму підготовки (спеціальності)

123 – «Комп'ютерна інженерія»

(шифр і назва напряму підготовки, спеціальності)

Слободян А. О.

(прізвище та ініціали)

Керівник к.т.н., проф. Захарченко С. М.

(прізвище та ініціали)

Опонент к.т.н., доц. Карпінець В. В.

(прізвище та ініціали)

Допущено до захисту  
Завідувач кафедри ОТ  
д.т.н., проф. Азаров О.Д.

(прізвище та ініціали)

«22» 12 2022 р

Вінниця ВНТУ 2022

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Освітньо-кваліфікаційний рівень: магістр  
Спеціальність 123 — «Комп'ютерна інженерія»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

обчислювальної техніки

*В. Азаров* проф., д.т.н. О.Д. Азаров

«15» / 09 2021 р.

### **З А В Д А Н Н Я НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ**

Студенту Слободяну Андрію Олеговичу

1 Тема роботи. «Технологія інтеграції чат-ботів у web-додатки», керівник роботи Захарченко Сергій Михайлович, к.т.н., проф. кафедри ОТ затверджені наказом №205-А вищого навчального закладу від 15.09.2022 р.

2 Строк подання студентом роботи \_\_\_\_\_ 2022 р.

3 Вихідні дані до роботи — проаналізувати методи і засоби інтеграції чат-ботів у web-додатки. Розробити програмний продукт на основні проаналізованих методів.

4 Зміст пояснювальної записки: проаналізувати методи і засоби інтеграції чат-ботів у web-додатки розробка програмного продукту; тестування веб-застосунку та рекомендації користувачу;

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): технічне завдання, домашня сторінка програми. Лістинг основного коду програми.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1— Консультанти розділів роботи

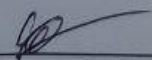
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Захрченко Сергій Михайлович		
5	Небава Миколи Іванович		

7. Дата видачі завдання 24.09.2022 року

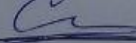
### КАЛЕНДАРНИЙ ПЛАН

Таблиця 2— Календарний план

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задач роботи	24.09.2022	вик.
2	Пошук матеріалів про методи покращення інтеграції чат-ботів	25.09.2022	вик.
3	Аналіз методів покращення інтеграції чат-ботів	7.10.2022	вик.
4	Аналіз сучасних інструментів для створення веб-застосунків	15.10.2022	вик.
5	Підготовка матеріалів та опис чат	21.10.2022	вик.
6	Підготовка економічної частини	14.11.2022	вик.
7	Оформлення пояснювальної записки	22.11.2022	вик.
8	Перевірка якості виконання магістерської кваліфікаційної роботи та усунення недоліків	19.12.2022	вик.

Студент  Слободян Андрій Олегович

( підпис )

Керівник роботи  к.т.н., проф. Захарченко Сергій Михайлович

( підпис )



## АНОТАЦІЯ

УДК 004.51

Слободян А.О. Технологія інтеграції чат-ботів у web-додатки. Магістерська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна Інженерія, Вінниця: ВНТУ, 2022.

На укр.мові. Бібліогр.: 34 назв; рис.: 24 ; табл. 16

У роботі розглянуто принципи Методи покращення доступності web-додатків для можливості освоєння web-ресурсів людьми з обмеженими можливостями. В роботі проведений аналіз сучасних підходів до вирішення цих задач, вибрано та проаналізовано аналоги. Розроблено web-додаток з дотриманням принципів покращення доступності сайтів.

Ключові слова: web-додаток, чат-бот, інтеграція.

## ANNOTATION

Slobodian A.O. Technology for integrating chatbots into web applications. Master's thesis on specialty 123 — Computer Engineering, Vinnytsia: VNTU, 2022.

In the Ukrainian language. Bibliography: 34 titles; Fig.: 14; table 16

The work examines the principles of methods of improving the accessibility of web applications for the possibility of mastering web resources by people with disabilities. The paper analyzes modern approaches to solving these problems, selects and analyzes analogues. The web application was developed in compliance with the principles of improving the accessibility of sites.

Keywords: web application, chatbot, integration.

## ЗМІСТ

ВСТУП .....	9
1 АНАЛІТИЧНИЙ ОГЛЯД ТЕХНОЛОГІЙ ВИКОРИСТАННЯ ЧАТ-БОТІВ.....	13
1.1 Аналіз характеристики чат-ботів .....	13
1.2 Платформи впровадження чат-ботів. ....	16
2 АНАЛІЗ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ ЧАТ-БОТУ ...	22
2.1 Класифікація чат-ботів.....	23
2.2 Нейронні мережі для навчання чат-ботів.....	27
2.3 Модель взаємодії чат ботів з платформою месенджера.....	29
2.4 Інструментальні засоби для створення та аналізу чат-ботів.....	30
2.5 Створення чат-ботів .....	32
3 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ЧАТ-БОТА .....	35
3.1 Огляд засобів створення чат-бота.....	35
3.2 Telegram Bot API.....	36
3.3 Мова програмування Python.....	37
3.4 Формат даних JSON .....	39
3.5 HTTP та HTTPS запити .....	42
3.6 База даних MongoDB.....	44
4 ОПИС РОЗРОБКИ ЧАТ-БОТА.....	47
4.1 Проектування функціоналу чат-бота.....	47
4.2 Створення чат-бота.....	48
4.3 Тестування чат-бота .....	55
5 ЕКОНОМІЧНА ЧАСТИНА.....	59
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	59

					08-23.МКР.0013.00.000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		Слободян А. О.			ТЕХНОЛОГІЯ ІНТЕГРАЦІЇ ЧАТ БОТУ У WEB-ЗАСТОСУНКИ	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>		Захарченко С.М						92
<i>Рецензент</i>		Карпінєць В. В				ВНТУ, гр. 1КІ-21м		
<i>Н.контр.</i>		Швець С. І.						
<i>Затвердж.</i>		Азаров О.Д						

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	62
5.3 Витрати на проведення науково-дослідної роботи.....	66
5.4 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором .....	67
5.5 Розрахунок ефективності вкладених інвестицій та періоду їх окупності .	70
ВИСНОВКИ.....	74
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	76
ДОДАТОК А.....	79
ДОДАТОК Б .....	83
ДОДАТОК В.....	89
ДОДАТОК Г .....	90
ДОДАТОК Д.....	91
ДОДАТОК Е .....	92
ДОДАТОК Ж.....	93

					Арк.
					8
Змн.	Арк.	№ докум.	Підпис	Дата	

## ВСТУП

В останні роки можна спостерігати тенденцію наскільки глибоко сучасні технології вкоренилися у житті суспільства. На цей час дуже складно знайти людину, яка б не користувалась гаджетами та інтернетом у повсякденному житті.

Наприклад, кожного дня майже всі володіють смартфоном, комп'ютером або безліччю інших пристроїв, які мають можливість підключення до інтернету. Найчастіше за допомогою цих пристроїв люди обмінюються інформацією, використовуючи для цього сервіси комунікації. Також неодмінною частиною нашого життя залишаються додатки та комп'ютерні програми, але від недавнього часу, стрімко набирають популярність чат-боти. Це пов'язано із простотою використання, швидкою розробкою та можливістю інтегрувати бота у різні сервіси, наприклад месенджери.

Чат-бот — це віртуальний помічник, який в залежності від свого призначення, створений для комунікації із користувачем за допомогою повідомлень.

Месенджери — це програми, які дозволяють передавати повідомлення в режимі реального часу через Інтернет. Еволюція месенджерів привела до того, що нині люди можуть відправляти не лише текстову інформацію, але аудіо і відео повідомлення та файли.

Як жоден інший, саме 2020 рік зміг показати швидкість переходу майже усіх сфер людської діяльності в онлайн режим. У період карантину, окрім студентів та школярів, досить велика кількість людей займались самоосвітою. Дехто навчався самостійно, використовуючи онлайн літературу та відеохостинг, а хтось за допомогою спеціальних платформ для дистанційного навчання. Тобто, ці люди отримали доступ до вже заздалегідь підготовлених матеріалів лекцій, мали можливість отримувати нагадування про події, завантажувати файли, з виконаними завданнями, проходити тестування тощо.

Нині люди вже звикли до дистанційної освіти, адже на це існує ряд переваг:

— гнучкість вибору та платформи та матеріалів;



- простота доступу;
- економія часу та коштів;
- нові знайомства з іншомовними студентами.

Гнучкість вибору. За рахунок різноманітних платформ для дистанційного навчання, у студентів з'явилась можливість самостійно обирати час, розклад, та навіть викладача.

Простота доступу. Нині існує досить велика кількість безкоштовних освітніх ресурсів, що являється чудовою можливістю підвищити рівень знань. Адже для цього необхідний лише гаджет та доступ до Інтернету.

Економія часу та коштів. В порівнянні з традиційною освітою, людям не потрібно витратити час та кошти на дорогу до навчального закладу, що є неодмінною перевагою. Також навчальні посібники та матеріали можна отримати у електронному вигляді.

Нові знайомства з іншомовними студентами. Завдяки дистанційній освіті з'явилась можливість навчатись з іншомовними студентами. Існують платформи для вивчення іноземної мови, що являється чудовою можливістю не лише отримати базові знання з певної мови, але дізнатись більше про традиції і культуру жителів цієї країни, та підвищити навички спілкування іноземною мовою.

Але використання таких платформ не завжди є зручним для людей, які більшу частину свого часу проводять у смартфонах. Набагато зручніше використовувати пристрій, який завжди під рукою, наприклад, у дорозі, або в подорожі. Але для цього необхідно створювати окремий програмний додаток. Використання чат-бота дозволяє зменшити вартість витрат для створення мобільного додатку, адже чат-бота можна інтегрувати у будь-який месенджер.

Створення інформаційного чат-бота це одне із найкращих рішень для власників освітніх ресурсів. Адже саме інформаційним чат-бот вважається, якщо він містить, опрацьовує та видає інформацію, а для того, щоб ним користуватись необхідний лише встановлений месенджер. За допомогою чат-боту можна швидко надіслати матеріали, посилання на лекцію тощо. Натомість

користувачі зможуть швидко отримати розклад, контактні дані викладача, літературу або всю іншу інформацію яку міститиме чат-бот у будь-якому месенджері. Для того, щоб запустити бота необхідно лише перейти за посиланням або знайти його по назві через пошук і розпочати діалог як зі звичайним користувачем.

Виходячи із розглянутого, завдання подальшого вдосконалення інтеграції чат-ботів є **актуальною задачею**.

**Метою дослідження** магістерської роботи є методи покращення інтеграції чат-ботів.

**Задачі дослідження** магістерської роботи:

- здійснити огляд стану роботи доступних чат-ботів на сайтах сайтів;
- розглянути та проаналізувати стандарти та методи інтеграції чат-ботів;
- створити чат-бот і досягти високого рівня доступності та функціональності шляхом оглянутих методів.

**Об'єкт дослідження** магістерської роботи — процес покращення функціональності чат-боту.

**Предмет дослідження** магістерської роботи — методи покращення функціональності чат-боту.

**Методи дослідження** магістерської роботи: використовувались методи статистичного аналізу для визначення потужності вибірки існуючих чат-ботів та їх функціональності, методи критеріального аналізу для формування узагальненого критерія функціональності чат-боту.

**Наукова новизна отриманих результатів** магістерської роботи полягає у тому, що:

- удосконалено методи розробки та інтеграції чат-боту за рахунок новітніх технологій, штучного інтелекту та API .

**Практичне значення одержаних результатів** магістерської роботи:

- розроблено чат-бот з функціоналом який надає можливість дистанційного навчання на базі месенджеру Telegram. Цей програмний додаток

демонструє всі можливі методи взаємодії, поміж яких власники освітніх ресурсів можуть обрати найзручніший та використовувати для організації дистанційного навчання.

**Апробація** Результати роботи опубліковані у матеріалах L регіональної науково-технічної конференції ВНТУ.

# 1 АНАЛІТИЧНИЙ ОГЛЯД ТЕХНОЛОГІЙ ВИКОРИСТАННЯ ЧАТ-БОТІВ

## 1.1 Аналіз характеристики чат-ботів

Нині чат-боти набирають великої популярності завдяки широкому використанню в багатьох сферах життєдіяльності людини. Ці віртуальні агенти можуть повідомити погоду, курс валют, записати клієнта на певну дату та навіть бути альтернативою онлайн бібліотеки.

Статистика ринку чат-ботів показує, що одна з причин, чому ця технологія стає все більш і більш популярною, полягає в тому, що чат-боти можуть відповісти на більшість запитань, які можуть поставити їм користувачі [2]. Для складніших питань все ще важливо мати кваліфікованих фахівців служби підтримки клієнтів. Але для щоденних питань служба чат-ботів зменшує витрати та пришвидшує час відгуку.

Якщо подивитись на статистику, то можна побачити швидкий ріст популярності чат-ботів (рис.1.1).

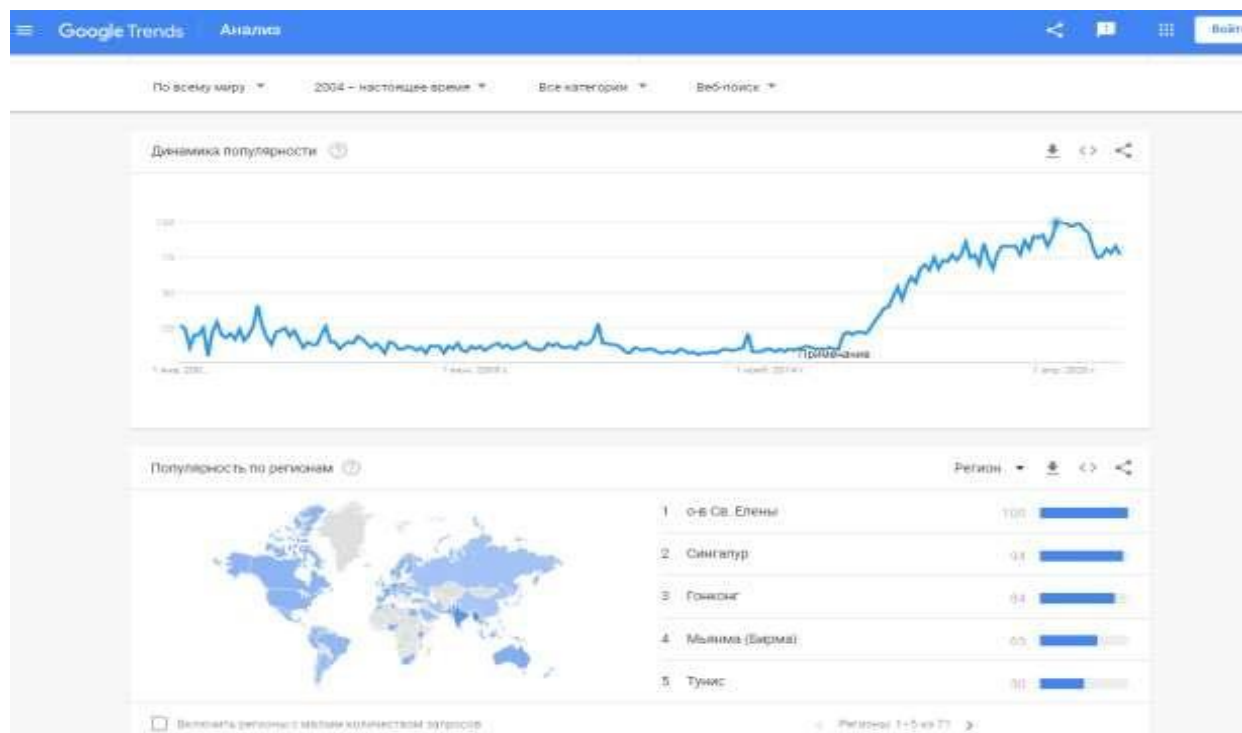


Рис.1.1. Статистика популярності чат-ботів у світі

На цей момент існує велика класифікація чат-ботів: по платформі впровадження, технології розробки, способу спілкування з користувачами і функціональності [3].

Але найпоширенішими залишається поділ з точки зору реалізації:

- бізнес класифікація яка полягає у сферах застосування чат-ботів;
- технічна класифікація полягає в основі створення чат-боту.

Бізнес класифікація:

- бот допомагає підібрати клієнту відповідний товар, відповідає на питання, розповідає про знижки та підводить клієнта до здійснення покупки.
- Інтеграція з системами-аналітики, CRM і бухгалтерією дає йому можливість складати звіти, аналізувати дані, заповнювати форми та ін.
- Замовити таксі, забронювати квитки, номер в готелі і столик в ресторані, оформити доставку продуктів, все це можна зробити за допомогою чат-бота. Медіа. З їх допомогою користувачі знаходяться в курсі останніх новин або вибирають і читають тільки цікаві для них розділи.
- Нагадуватиме про зустрічі, підкаже прогноз погоди та навіть знайде рецепт будь-якої страви.

За технічною класифікацією чат-боти поділяються на прості, розумні з підтримкою штучного інтелекту та гібридні.

Простий бот — це бот, який відповідає на запитання на основі заздалегідь встановленого вибору інтегрованих відповідей. Простих ботів також називаються ботами дерева прийняття рішень, як впливає з назви, вони використовують ряд визначених правил та нагадують текстову систему IVR (система попередньо записаних голосових повідомлень, що виконує функцію маршрутизації дзвінків всередині контакт-центру).

Такі боти не роблять жодних висновків з попередніх взаємодій та найкраще підходять для прямолінійних діалогів.

Простий бот зазвичай будується на базі кнопок, тому він ідеально підходить для опитування, підтримки продажів і практично для будь-якого

простого завдання автоматизації процесів, де сценарії спілкування чітко визначені.

Натомість розумні чат-боти з підтримкою штучного інтелекту створені для імітації майже людської взаємодії з клієнтами [4]. Для того, щоб вести вільні розмови і розуміти намір, мову та почуття розумні чат-боти використовують технології обробки природних мов (NLP, NLU тощо). Natural Language Processing (NLP) — це область штучного інтелекту, яка дозволяє комп'ютерам аналізувати та розуміти людську мову. Найбільша відмінність від простого чат-бота полягає у використанні моделей машинного навчання, що значно збільшує функціональність бота, оскільки він здатний ідентифікувати сотні різних запитань, написаних людиною.

Бот запрограмований на самостійне навчання, оскільки він вводиться в нові діалоги та слова. Фактично, коли чат-бот отримує нові голосові або текстові повідомлення, кількість запитів, на які він може відповісти, і точність кожної відповіді, яку він дає, збільшується. На відміну від простих чат-ботів, їм не вистачає суворих дерев рішень, які б керували їх роботою. Схема роботи таких ботів зображена на рис. 1.2.

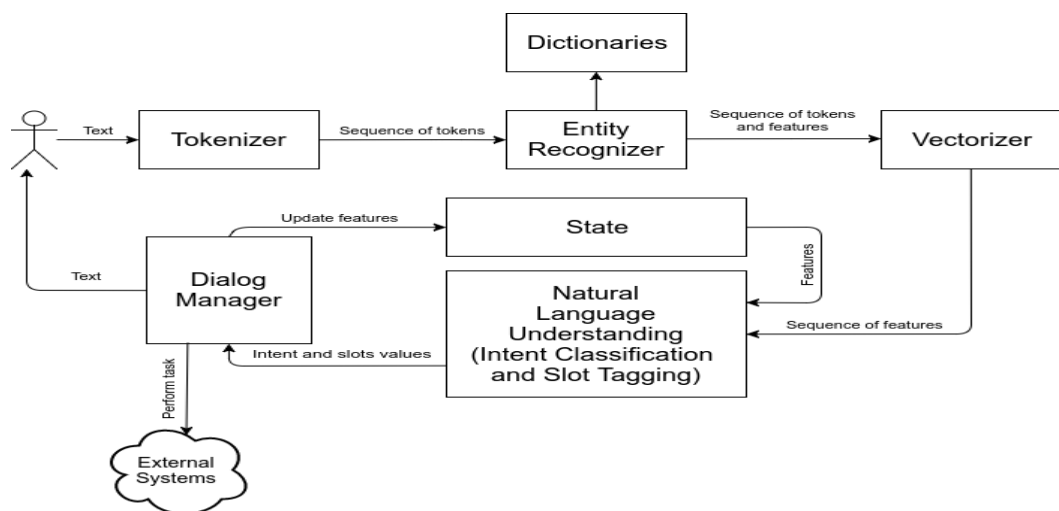


Рис.1.2. Схема роботи чат-бота з підтримкою штучного інтелекту

Взагалі типовий цикл роботи будь-якого чат-бота можна уявити ланцюжком наступних дій :



- отримання запиту від клієнта;
- розбір запиту, розуміння висловлювання і визначення намірів клієнта в контексті його бізнес-кейсу;
- виконання дій згідно заздалегідь визначеним сценарієм
- генерація відповіді на природній мові;
- надсилання відповіді клієнту.

Найскладнішим етапом роботи є розбір клієнтського запиту. Чат-боти на базі Machine Learning використовують для цього методи NLU і NLP [5]. Наприклад, для текстових чат-ботів процес розбору включає наступні етапи:

- Попередня обробка тексту, розбиття на слова, виправлення помилок, відкидання стоп-слів (артиклі, вигуки, сполучники і ін.), розширення запиту за допомогою словників синонімів
- Класифікація запиту на основі прикладів фраз і алгоритмів або формальних правил (шаблонів) Останній вид за технічною класифікацією є гібридний чат-бот.

Гібридний чат-бот — це поєднання боту побудованого на базі штучного інтелекту та простого. Зазвичай такі боти говорять: «Я бот, який може допомогти вам у розв'язанні таких питань; будь ласка, натисніть відповідну кнопку X, Y, Z або введіть своє запитання у поле нижче ». Особливість цих ботів полягає у тому, що вони використовують і дерева прийняття рішень і технології обробки природних мов.

## 1.2 Платформи впровадження чат-ботів.

Як вже було сказано раніше, чат-ботів можливо інтегрувати у будь-який веб сервіс. Але, як не дивно найбільшої популярності вони здобули завдяки месенджерам. Це все завдяки тому що, з кожним роком люди все частіше віддають перевагу смартфонам [7]. Наприклад частка мобільного трафіку в період з 2018 по 2019 роки зросла на 53,3%, приріст склав 8,6%. Таку статистику наводить App Annie у звіті State of Mobile 2020 (рис. 1.3).

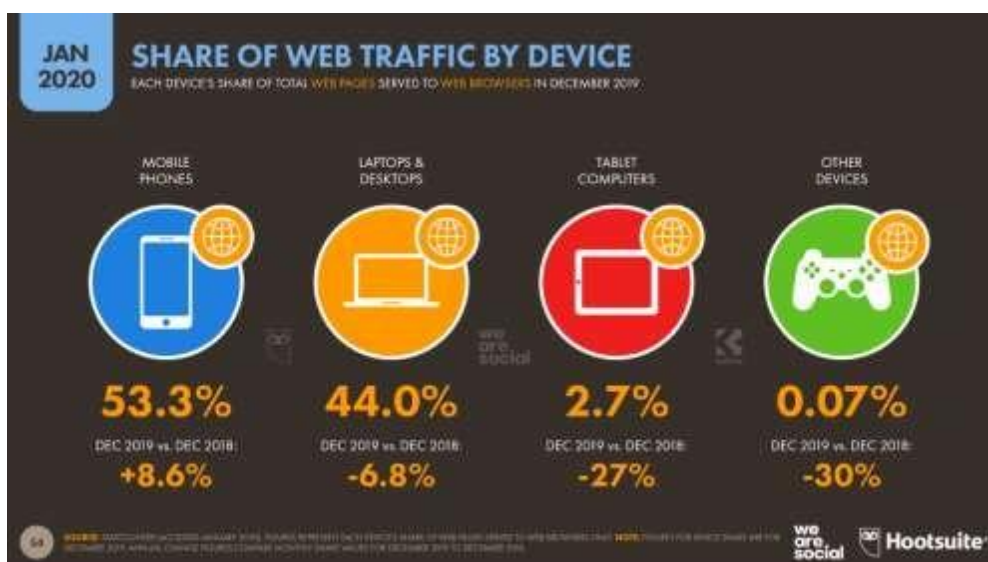


Рис.1.3. Збільшення частки мобільного трафіку

На рис. 1.4 зображено, що в середньому, сучасна людина витрачає 3,4 години часу у смартфоні. Більш ніж половина цього часу припадає на комунікацію через месенджери та соціальні мережі. Залишок часу люди витрачають на купівлю товарів і послуг в інтернеті, поглинають інформацію з розважальних сервісів та грають в ігри.

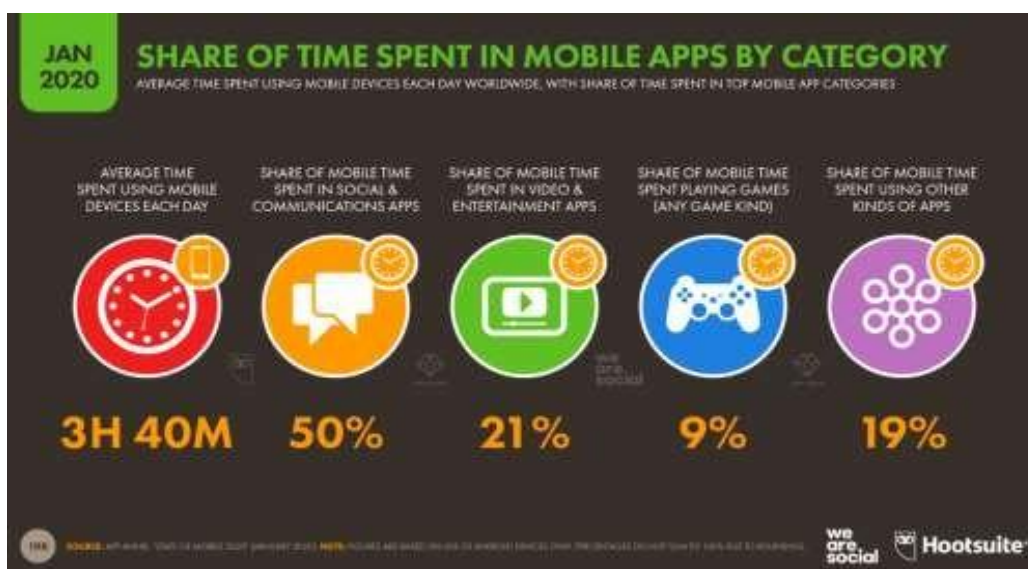


Рис. 1.4. Обсяг витраченого часу в інтернеті за добу

Тому, з'ясувавши, що люди більшу частину свого часу проводять за комунікацією у месенджерах, необхідно зробити аналіз найпопулярніших платформ для впровадження чат-бота.

Отримавши статистичні дані із сервісу Statista.com за період січня 2021 року, дізнались що першим лідером месенджерів у світі є WhatsApp [6]. Кількість користувачів, що встановили програму більше ніж двох мільярдів у світі (рис.1.5).

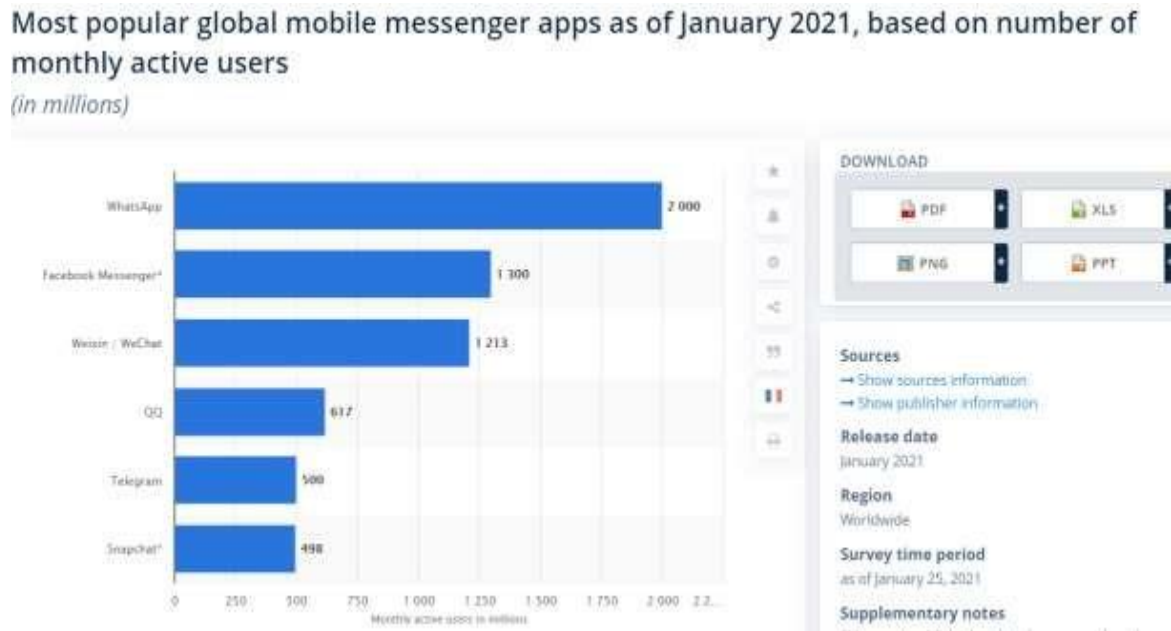


Рисунок 1.5. Статистика популярних месенджерів у світі

На період 2020 року популярними месенджерами стали:

- WhatsApp — 2 млрд. користувачів;
- Facebook Messenger — 1,3 млрд. користувачів;
- WeChat — 1,165 млрд. користувачів;
- QQ — 731 млн. користувачів;
- Telegram — 400 млн. користувачів;
- Snapchat — 398 млн. користувачів;
- Discord — 300 млн. користувачів;
- Viber — 260 млн. користувачів;
- Line — 250 млн. користувачів;
- Skype — 50 млн. користувачів.

Але також, зібравши статистичні дані щодо найбільш популярних месенджерів в Україні, із сервісу [Statista.com](https://www.statista.com) здалося зробити висновок, що

Viber був найпопулярнішим месенджером в Google Play Store в Україні, оскільки в червні 2020 року він охопив майже 98 відсотків користувачів смартфонів, що працюють на базі Android. Facebook Messenger і Telegram мали другий і третій показник охоплення, виміряні приблизно 75 і 74 відсотками відповідно. Але отримавши статистику із Google Trends можна зробити висновок, що нині Telegram в Україні займає перші позиції (рис.1.6).

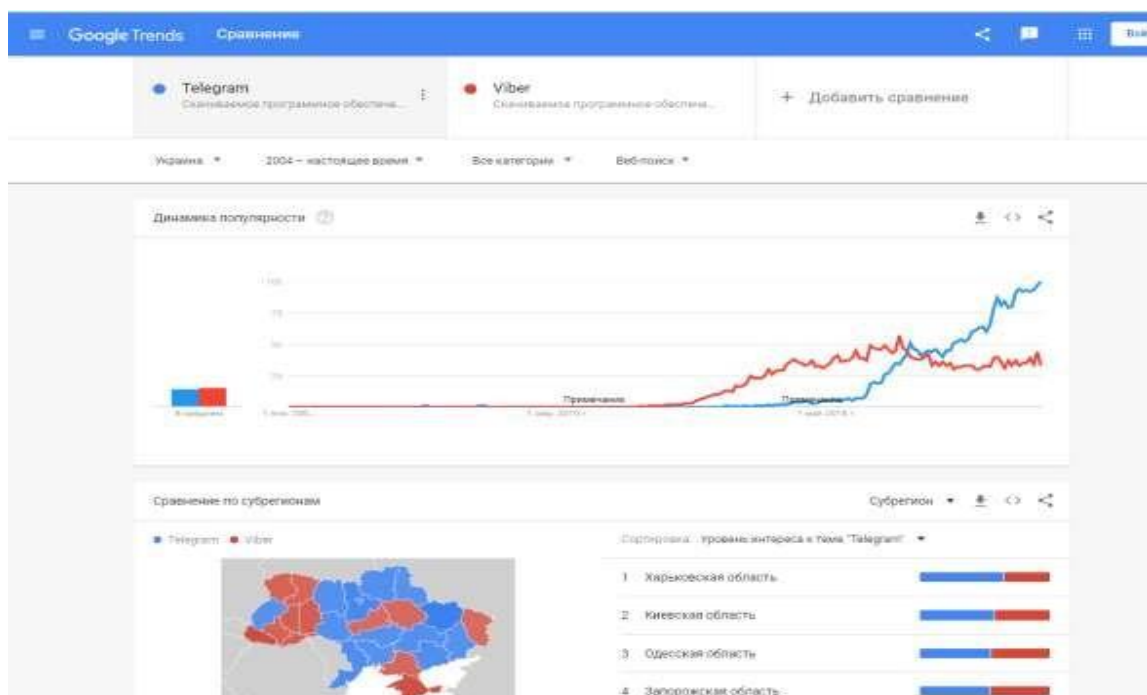


Рис.1.6. Статистика найпопулярніших месенджерів в Україні

Telegram — кросплатформний месенджер із відкритим кодом, функціями телефонного зв'язку по протоколу IP, що дозволяє обмінюватися текстовими, голосовими та відеоповідомленнями, фотографіями та файлами багатьох форматів. Станом на березень 2021 року офіційні клієнти для Telegram охоплюють:

- мобільні додатки для Android і iOS / iPadOS ;
- настільні додатки для Windows , Linux і macOS ;
- вебдодаток

Використовуючи месенджер Телеграм користувачі отримують не тільки платформу для спілкування, але і багато інших можливостей, завдяки, яким месенджер набув великої популярності.

З основних можна перерахувати:

- спілкуватись з користувачами по всьому світу, та здійснювати аудіо та відеодзвінки;
- відправляти пересилати і отримувати файли різних форматів;
- створювати групові та засекречені чати;
- знайти людину за псевдонімом, не знаючи номера телефону;
- відправляти знайдені в пошукових системах фото без збереження в пам'яті пристрою;
- користуватись хмарним сховищем даних. Будь-які файли, обсяг яких не обмежені, зберігаються на віддаленому сервері;
- можливість безплатно та без інтернету слухати музику. Для цього спеціально був розроблений медіа плеєр.

Месенджер має низку переваг. Серед основних:

- висока швидкість роботи, не дивлячись на розташування серверів в різних частинах планети, передача повідомлення займе частки секунди;
- автоматичне додавання списку користувачів сервісу з телефонної книги;
- простий початок роботи, для реєстрації потрібен тільки номер телефону;
- всі функції та можливості безплатні та без реклами;
- інноваційність, сервіс постійно оновлюється, можливості розширюються як з урахуванням пропозицій конкурентів, так і власними розробками, які не мають аналогів;
- високий рівень безпеки і захисту від злому;
- необмеженість, кількість відправлених в день повідомлень, обсяги завантажуваних в хмару файлів, часу використання;
- сумісність з усіма основними операційними системами, включаючи мобільні;
- Конфіденційність, усі чати, які існують у месенджері зашифровані.

Останній пункт можна віднести одночасно як до переваг, так і до недоліків.

Важливо розуміти, що всі повідомлення в Telegram завжди надійно зашифровані. Різниця між секретним і звичайним чатом тільки в методі шифрування: клієнт-клієнт для секретних чатів, клієнт-сервер / сервер-клієнт для звичайних [9]. Завдяки цьому, користувачі мають доступ до свого листування в звичайних чатах з будь-якого пристрою без втрати приватності.

У секретних чатах це не працює — він прив'язаний до конкретного пристрою. Вся історія повідомлень зберігається на одному пристрої і доступна тільки на ньому.

Тому така безпека передачі даних і привернула увагу «невидимого Інтернету» до месенджера. Це дозволило використовувати Telegram як кіберзлочинцям, такі звичайним користувачам, які, отримали одну з кращих систем захисту даних і простий інтерфейс. Ось чому у деяких країнах цей месенджер знаходиться під заборобою.

### 1.3. Чат-бот в месенджері Телеграм

Якщо порівняти статистику, коли почався ріст популярності чат-ботів тамесенджеру Телеграм, можна побачити що це сталося майже в один проміжок часу. А вже в червні 2015 року месенджер Телеграм відкрив платформу для створення ботів які відгукуються на команди користувачів і дають можливість взаємодії із зовнішніми сервісами. Боти, які створюються в Телеграм — являються одним із різновидів чат-ботів. Зазвичай, їх роблять публічними або приватними.

Публічні боти доступні будь-якому користувачеві. Відкрити їх можна використовуючи посилання, або через пошук.

Приватні чат-боти створюються спеціально для клієнтів однієї компанії (відвідувачів кафе, пацієнтів клініки і так далі). Особливість в тому, що приватних чат-ботів неможливо знайти у відкритому доступі — компанії підключають до них самі.



## 2 АНАЛІЗ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ ЧАТ-БОТУ

Інтернет, який спочатку був засобом передачі інформації, тепер все більше бере на себе функцію комунікатора. Глобальна мережа стає особливим комунікаційним середовищем, займаючи важливе місце в кожній сфері суспільного життя. Особливо це стосується сучасного покоління, в якому переважають мобільні пристрої.

Сьогодні проблема віртуального спілкування в мережах стала популярною завдяки комунікації, яка дозволяє швидко отримувати доступ до інформації, одночасно працювати в мережі багатьом користувачам, збирати й обмінюватися різноманітною інформацією, підтримувати освітній і професійний розвиток, спілкуватися для вирішення особистих завдань. А також ділові питання з клієнтами та партнерами, проведення різноманітних аналітичних досліджень та інші переваги.

Комунікаційна функція Інтернету набуває нових форм, пов'язаних із можливостями сучасних технологій. Комунікаційні технології на основі месенджерів і чат-ботів останнім часом набули особливого значення. Месенджери — це програми, які дозволяють надсилати повідомлення через Інтернет у реальному часі. Еволюція месенджерів призвела до того, що тепер вони можуть надсилати текстові повідомлення, а також аудіо- та відеоповідомлення. Сучасне покоління віддає перевагу мобільним пристроям настільним комп'ютерам. Його представники не звикли взаємодіяти в інтернет-середовищі за допомогою браузерів і спливаючих меню, але використовують смартфони для небагатьох завдань. Дослідження показують, що телефони частіше використовуються для обміну повідомленнями, ніж для будь-яких інших цілей [10]. Тому компанії наполегливо працюють, щоб привернути увагу онлайн-користувачів і створюють чат-ботів для інтеграції в месенджери.

Згідно з результатами дослідження, проведеного Flurry Analytics, попит на додатки для обміну повідомленнями в соціальних і мобільних мережах

продовжує зростати на відміну від інших сфер [15]. Так, у 2016 році, який пов'язаний з піком популярності чат-ботів, попит на додатки для обміну повідомленнями зріс на 44%, порівняно із середньорічним темпом зростання в 11% для всіх додатків, а час, який користувачі витрачають на обмін повідомленнями, збільшився на 394%. при цьому середній показник становить 69%. Крім того, за даними BI Intelligence [12], сукупна аудиторія користувачів чотирьох основних інструментів обміну миттєвими повідомленнями WhatsApp, WeChat, Messenger і Viber перевищує аудиторію чотирьох основних соціальних мереж Facebook, Instagram, Twitter і LinkedIn.

Вирішенню методичних проблем створення та використання чат-ботів у різних сферах діяльності присвячено велику кількість праць вітчизняних та зарубіжних науковців [1-3; 5; 7, 11–12, 16]. Автори дослідження [12] зазначають, що наразі користувачі в усьому світі використовують Messenger для різноманітних завдань, окрім простого обміну текстовими повідомленнями, а також для взаємодії з клієнтами компанії, пошуку відповідних продуктів, споживання контенту тощо. У роботах [3; 5; 11] подано огляд найпопулярніших засобів розробки чат-ботів. Роботи [1–2; 7; 13–14; 16] описують проблему створення чат-ботів та їх використання в різних сферах. Водночас цей напрямок динамічно розвивається, потребуючи більш детального аналізу та демонстрації методів, фреймворків, які використовуються для створення чат-ботів.

В інформаційних просторах спостерігаються дві тенденції: з одного боку, кількість доступної інформації зростає геометрично, а з іншого боку, самі дані фізично стискаються. Також друга тенденція стосується зберігання та передачі даних. Бібліотечні дані зведено до електронних сховищ, комунікації в соціальних мережах — лайків, твітів і публікацій, а також між користувачами мережі, компаніями та клієнтами — і взаємодії з чат-ботами.

## 2.1 Класифікація чат-ботів.

В сучасних месен- джерах чат-боти є універсальним засобом для вирішення різноманітних завдань. Бот (англ. Bot) — спеціальна програма, що

виконує автоматично або за заданим розкладом будь-які дії через інтерфейси, призначені для людей. Зазвичай боти призначаються для виконання одноманітної і повторюваної роботи, з максимально можливою швидкістю. Вони застосовуються за необхідності швидшої реакції, ніж у людини. Це ігрові боти, боти для Інтернет аукціонів тощо. Іноді їх застосовують для імітації дій людини в спілкуванні, тоді їх називають чат-ботами.

Чат-боти — це інтерактивні програмні платформи в Messenger, які діють подібно до людини. Такі програми вміють відповідати і задавати питання. Чат-боти використовуються в різних сферах для вирішення типових проблем. За даними дослідження Business Insider, 80% компаній використовуватимуть чат-боти до 2020 року; 42% учасників опитування вважають, що технологія чат-ботів найбільше покращить якість обслуговування клієнтів; 48% вже використовують технологію чат-ботів для таких функцій бізнесу, 40% планують запровадити певну форму цієї технології до 2020 р. [6].

Існує чимало роботів, які використовуються в бізнесі, охороні здоров'я та фінансах, і вони стають звичним явищем у процесі навчання. Освіта завжди відставала від інших галузей у використанні нових технологій, але вони все одно будуть використовуватися. Щоб мати чітке розуміння суті, сильних і слабких сторін кожного чат-бота, його необхідно класифікувати (табл. 2.1).

Чат-боти можна створювати для особистого та бізнес-користування. Персональний чат-бот діє як персональний помічник користувача, виконуючи такі завдання, як керування календарем, надсилання текстових повідомлень, відповідь на дзвінки, пошук і відтворення аудіо- та відеофайлів.

Бізнес-чат-боти розроблені для використання в бізнесі, щоб залучати клієнтів до розмов і виконувати різні процеси, пов'язані з маркетингом, продажами та іншими завданнями підтримки. Такі чат-боти використовуються в багатьох сферах бізнесу для автоматизації процесу спілкування з клієнтами, а також для виконання аналітичних функцій.

Таблиця 2.1 — Класифікація чат-ботів

Однака класифікації	Вид чат-бота
Користувач	Персональний Бізнесовий
Інтерфейс	Кнопковий Текстовий
Доступ	Доданий в групу За підпискою Вбудований в діалог
Призначення	Комунікаційний Функціональний
Принцип роботи	Шаблонний Який навчається

Взаємодія з користувачем у чат-ботах може бути кнопковою або текстовою. Кнопкові чат-боти мають інтерфейс у вигляді кнопок і команд. Бесіда організована таким чином, що користувач може вибрати категорії, питання або пропозиції, які можуть його зацікавити. Розмови відбуваються натисканням кнопок, і робот відповідає ними як командами. Такий чат-бот найбільше схожий на звичайний мобільний додаток, єдина відмінність — він не має власного інтерфейсу, а використовує для роботи месенджер. Команди для таких ботів визначаються під час створення.

Спілкування за допомогою текстових чат-ботів наближене до реального людського спілкування, але з деякими функціональними особливостями. Цей тип робота має ширший спектр можливостей, ніж робот-кнопка. Такі чат-боти також можуть відображати кнопки для швидшої навігації.

Доступ до чат-ботів можна отримати трьома основними способами: їх можна приєднати до груп, до яких належить користувач; ними можна «поділитися» з друзями в списку контактів, надіславши їм повідомлення або зателефонувавши безпосередньо в діалоговому вікні. Вбудовані боти (inline bots) зручні тим, що їх можна викликати в будь-якій розмові. Для цього просто напишіть символ @, а потім назву бота. Бот відкриється прямо в розмові і запропонує варіанти дій, результат можна буде відразу переслати співрозмовнику, який зараз перебуває в розмові.

Основне призначення чат-ботів — спілкування або заміна функціональності мобільних додатків. Комунікаційні боти використовуються для прискорення процесу взаємодії між клієнтами та компаніями. Найпоширенішими комунікаційними ботами є типи відповідей на популярні запитання.

Сьогодні користувачі мобільних пристроїв не хочуть встановлювати багато програм. 72% користувачів використовують до 7 додатків на день [9]. Тому найкращим підходом є інтеграція різних сервісів у програми, з якими люди вже знайомі. Такі програми є функціональними чат-ботами. Вони призначені для заміни мобільних додатків або веб-сайтів. Сучасні платформи чат-ботів здатні повністю реалізувати функціонал таких додатків: шукати, замовляти, бронювати, виконувати базові транзакції, працювати тощо. Перевага функціонального чат-бота полягає в зручному інтерфейсі, який спеціально розроблений під запити користувачів, на відміну від різноманітних веб-сайтів та додатків, до інтерфейсу яких потрібно звикати.

За принципом роботи чат-боти можна розділити на два типи: одні працюють за заздалегідь заданим шаблоном, а інші навчаються в процесі спілкування. Боти шаблонів (скриптів) не розуміють природної мови. Розмова в такому боті являє собою заздалегідь сформований шаблон, а сценарій — це дерево рішень, у якому відповідь на запитання відкриває новий, заздалегідь запрограмований сценарій. Їхні розмови зазвичай лінійні та структуровані. Боти-шаблони виділяють ключові слова та реагують на них під час спілкування з користувачами. У таких роботах необхідно писати команди для кожного слова або фрази. Якщо користувач не використовує ключові слова під час спілкування, робот не зрозуміє його і виконає дії, передбачені для ситуації, наприклад, завчасно зв'яжеться з оператором. Такі боти мають обмежений функціонал, але вони можуть бути корисні в певних ситуаціях.

Навчальні боти розроблені на основі рішень штучного інтелекту, які використовують обробку природної мови та машинне навчання під час побудови розмов. Вони працюють на основі аналізу діалогу для подальшого

вдосконалення своїх комунікативних навичок. Такі боти здатні обробляти природну мову і правильно відповідати на запитання.

## 2.2 Нейронні мережі для навчання чат-ботів

Нейронні мережі найчастіше використовуються для вирішення завдань машинного навчання. Нейронна мережа — це математична модель, програмна або апаратна реалізація якої нагадує організацію та функціонування біологічної нервової системи. Нейронні мережі здатні проводити глибокий аналіз оброблених даних, виявляти складні нелінійні залежності та неоднозначні зв'язки. Штучний нейрон — це структура, яка приймає сигнали, перетворює їх і передає іншим нейронам. Отже, штучна нейронна мережа складається з групи штучних нейронів, які взаємодіють один з одним шляхом передачі та перетворення сигналів [4]. Незважаючи на велику різноманітність варіантів нейронної мережі, всі вони мають спільні характеристики. на рис. На малюнку 1 зображено узагальнену модель штучного нейрона.

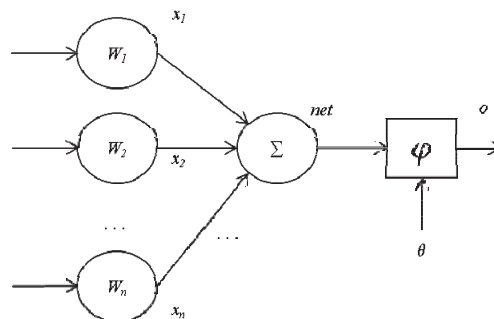


Рис. 2.1. Модель штучного нейрона

Штучний нейрон має вхід (  $x_t$  ), через який він отримує сигнал. Кожне з'єднання між нейронами відповідає вазі  $w_t$ . Коли сигнал проходить через з'єднання, його значення множиться на вагу з'єднання. Додайте всі отримані продукти, щоб отримати зважену суму  $net$ :

$$net = \sum_i^n x_i w_i .$$



Для обробки нелінійних залежностей в даних використовується функція активації (activation function)  $\varphi(\text{net})$ , яка перетворює значення зваженої суми  $\text{net}$  і формує вихідне значення нейрона. Функція активації нормалізує вхідні дані, тобто формує вихідні дані в потрібному діапазоні. В якості функцій активації найбільш часто використовуються лінійна, логістична і гіперболічна функції. Головні їх відмінності — це діапазон значень. Для прийняття рішень отримане значення функції активації (вихідний сигнал)  $o$  порівнюється з граничним значенням  $\theta$  і в залежності від результату порівняння приймається те чи інше рішення  $o(\text{net})$ :

Для обробки нелінійних залежностей у даних використовується функція активації  $\varphi(\text{net})$ , яка перетворює значення звуженої мережі для формування вихідних значень нейронів. Функція активації нормалізує вхідні дані, тобто формує вихідні дані в необхідному діапазоні. Як функції активації найчастіше використовуються лінійні, логістичні та гіперболічні функції. Основна їх відмінність — діапазон значень. Для прийняття рішення отримане значення  $o$  функції активації (вихідний сигнал) порівнюється з граничним значенням  $\theta$  і в залежності від результату порівняння приймається те чи інше рішення  $o(\text{net})$ :

$$o(\text{net}) = \begin{cases} 0, & \text{net} < \theta, \\ 1, & \text{net} >> \theta \end{cases}.$$

Нейронні мережі мають властивості навчання, які дають їм перевагу перед системами, заснованими на заздалегідь визначених алгоритмах. Мережа може навчатися на необроблених даних, тобто все, що їй потрібно, це база даних із сотнями тисяч відповідей на запитання. Цього достатньо для того, щоб мережа певною мірою зрозуміла мову та те, як відповідь пов'язана із запитанням з точки зору змісту, і могла адекватно відповісти на запитання. Дуже важливо навчити чат-бота розглядати контекст розмови з точки зору ключових слів. З огляду на контекст, ви можете використовувати нейронні мережі LSTM (Long Short-Term Memory) як з короткочасною, так і з довгостроковою пам'яттю. Її суть полягає в тому, що нейрони цієї мережі мають певний параметр,

який запам'ятовує те, про що йшлося в тому чи іншому контексті. Наступний приклад розмови показує, як нейронна мережа запам'ятовує контекст.

Користувач: Я навчаюсь в ВНТУ. (Нейронна мережа запам'ятовує, що користувач навчається в ВНТУ.)

Користувач: Де я навчаюсь?

Чат-бот: Ти навчаєшся в ВНТУ.

Користувач: На якому факультеті я навчаюсь?

Чат-бот: Я не знаю.

Користувач: У мене є друг і він навчається в ДОННУ. (Нейронна мережа запам'ятовує, що зараз мова йде про ДОННУ.)

Користувач: Де навчається мій друг?

Чат-бот: Твій друг навчається в ДОННУ.

Користувач: Я навчаюсь в ВТК. (Нейронна мережа забуває про попередній контекст і запам'ятовує новий.)

Зрозуміло, що нейронна мережа реагує на ключові слова при запам'ятовуванні контексту. Для мережі дуже важливо розуміти, на які слова вона має реагувати, а на які — ні. З цим завданням впорається технологія обробки природної мови NLP (Natural Language Processing) — загальний напрямок штучного інтелекту і математичної лінгвістики. НЛП вивчає проблему комп'ютерного аналізу і синтезу природної мови, що дозволяє судити про суть речення, розуміти його зміст і виділяти ключові слова.

### 2.3 Модель взаємодії чат ботів з платформою месенджера

Зростаюча популярність чат-ботів призвела до появи багатьох інструментів [2; 5; 11]. Чат-боти можна створювати з нуля, використовуючи різні мови програмування. В основному для цих цілей використовуються серверні мови програмування: Python, Ruby, Node.JS, PHP. Також є багато готових рішень, які допоможуть створити чат-ботів і навчити їх навичкам спілкування без програмування.

У загальному вигляді модель функціонування чат-бота включає три елементи (рис. 2.2):

Backend — бот, який отримує інформацію, обробляє її та відправляє назад;

Інтерфейсом є клієнтська платформа бота, якою може бути будь-який популярний месенджер, такий як Facebook Messenger, Telegram, Skype, Viber,

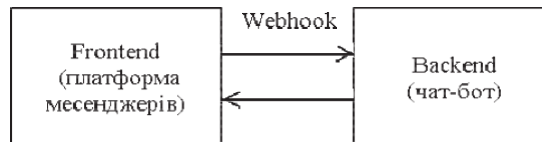


Рис. 2.2. Модель взаємодії чат-бота з платформою

канали мобільного зв'язку, такі як SMS, USSD, соціальні мережі або вбудований ресурс чату, який дозволяє отримувати та надсилати повідомлення в режимі реального часу. Від правильного вибору платформи залежить залучення клієнтів, можливість підвищення лояльності та винагорода від впровадження розробок;

Вебхук — це посилання на основі URL-адреси, яке з'єднує бота (сервер) і платформу чату (інтерфейс). Вебхуки дозволяють безпечно надсилати й отримувати повідомлення за допомогою стандартних запитів HTTP. У всіх сучасних месенджерах є докладна інструкція, як підключити бота і сам сервіс.

Ця модель позбавляє вас від прив'язки до однієї платформи — один і той же бот може підключатися до різних месенджерів за допомогою API.

#### 2.4 Інструментальні засоби для створення та аналізу чат-ботів.

Ви можете створити чат-бота одним із двох способів. Перший метод полягає у створенні робота за допомогою будь-якої мови програмування в певному фреймворку, а другий метод полягає у використанні не коду, а використання платформи розробки. Також після безпосереднього створення бота необхідно проаналізувати його роботу для подальшої оптимізації. З огляду на це, інструменти для створення та аналізу чат-ботів включають:

фреймворки, які вимагають певних навичок програмування, платформи для розробки чат-ботів і обробки природної мови (для створення ботів без програмування), а також служби аналізу ботів.

До найбільш поширених фреймворків створення чат-ботів відносяться:

**BotKit** — це набір інструментів з відкритим кодом, розроблених для Node.js. Підходить як перша платформа для навчання та експериментування з чат-ботами. Botkit має сервіс Botkit Studio, який містить стандартний набір програм, базові бібліотеки та плагіни для розширення функціональності ботів, а також дозволяє створювати боти для Facebook Messenger, Slack і Cisco Spark;

**Claudia Bot Builder** — це програма для створення ботів, яку можна використовувати в AWS Lambda. Він містить бібліотеку Claudia.js, яка допомагає створювати ботів для Facebook Messenger, Telegram, Skype, Slack slash, Twilio, Kik і GroupMe;

**Bottr.me** — це найпростіший фреймворк Node.js для створення та використання персональних чат-ботів. Ця послуга дозволяє тестувати створені вами продукти.

Платформа IBM Watson, безсумнівно, є лідером у розробці рішень для систем штучного інтелекту. Однак безкоштовне використання цієї платформи обмежене. Далі ми надаємо порівняльну ілюстрацію найпоширеніших платформ ШІ для розробки чат-ботів, які є у вільному доступі (табл. 2.2).

Таблиця 2.2 — Характеристика платформ для розроблення чат-ботів зі штучним інтелектом

Назва	Короткий опис	Клієнтська платформа	Інтеграція	Безкоштовна версія
SnatchBot	Потужна, гнучка і проста у використанні платформа, що забезпечує швидку інтелектуальну розробку чат-ботів для всіх сфер застосування. Діалог на основі алгоритмів розпізнавання людської мови NLP або тексту. Підтримка 6 мов. Машинне навчання	Багатоформний	Twilio	Є

## Продовження таблиці 2.2

Converse.ai	Графічний користувацький інтерфейс. Інтелектуальний аналіз діалогів за допомогою зворотного зв'язку через опитування. Перевірка і обробка контексту. Технічна підтримка клієнтів	Багато- платформний	SalesForce, Stripe, Pay- Pal, Clear Bit, Twilio, Airtable, HubSpot, FlightStats	Є (для нача- льного рівня)
Smooch	Однаковий інтерфейс для всіх платформ. Інтелектуальна розмова, на основі інтеграції з перекладачами, NLP і AI-двигунами. Управління учасниками розмови. Використання контенту для кращого розуміння намірів користувача і його автоматичного інформування	Багато- платформний	Stripe Twilio	Є (до 10 тис. користу- вачів в місяць). Пробний період
ChattyPeople	Створення ботів для спілкування з клієнтами та електронної комерції. Можливість приймати замовлення від месенджера і з коментарів, регулювати пропозиції і здійснювати операції	Facebook	E- commerce, які підтри- мують open cart	Є для на- чального рівня
Wit.ai	Спілкування на основі алгоритмів розпізнавання людської мови NLP або тексту. Підтримка 11 мов. Можливість побудови для певного тексту безлічі варіантів питань. Є вбудовані шаблони і плагіни для інтеграції текстових повідомлень, фото тощо. Можливість навчання бота певний період	Facebook	Python, Node.js, Ruby	Є

Аналіз поведінки чат-ботів під час взаємодії з користувачами є важливою частиною їх використання. Моніторинг чат-бота передбачає постійну оптимізацію його поведінки та сценаріїв розмов на основі аналізу взаємодії користувача з ботом. Серед безкоштовних сервісів для аналізу ефективності чат-ботів Botmetric, Chatbase, Botanalytics, Dashbot, Botlytics та інші (табл. 2.3). Також варто зазначити, що такі платформи, як Facebook, Telegram тощо, мають власні інструменти аналітики.

## 2.5 Створення чат-ботів

Створення чат-бота вимагає значних внесків на всіх етапах розробки. Основні завдання створення бота — навчити бота розуміти контекст запитів користувачів (26%) і будувати розмовні моделі (16%). Якщо розглядати навички, необхідні при створенні чат-бота, то їх можна визначити так: Business

Skills — 11%, Product Owner — 11%, Creative Technical Knowledge — 14%, Design — 16%, Coding — 27% [7].

Щоб створити чат-бота, потрібно визначити завдання, яке він буде вирішувати, вибрати платформу, на якій він працюватиме (Facebook, Telegram тощо), сервіс розробки та сервіс для його запуску. При цьому можна виділити кілька етапів.

Таблиця 2.3 — Характеристика сервісів для аналізу взаємодії користувачів з ботом

Назва	Короткий опис	Можливості	Безкоштовна версія
Botmetric	Гнучка аналітична система з відкритим кодом	Відстежувати показники роботи ботів – кількість користувачів, повідомлень відправлених боту, повідомлень від бота, завантаження зображень. Отримувати висновки з рекомендаціями щодо зміни діалогу	Є
Chatbase	Хмарний сервіс для аналізу і оптимізації	Аналіз ключових показники ефективності бота. Пошук помилок в роботі бота на основі технології машинного навчання	Є
Botanalytics	Відстеження життєвого циклу користувача	Сегментація діалогів. Визначення вузьких місць. Вимірювання ступеня залучення користувачів	Є з обмеженнями використання
Dashbot	Аналіз змісту розмов і аналізу настроїв користувачів	Відстежувати показники роботи ботів – кількість користувачів, утримання користувача тощо	Є з обмеженнями використання
Botlytics	Хмарна платформа для аналізу і комунікації ботів	Відстежувати повідомлення, які надсилає бот, їх кількість, а також діалоги, в яких він бере участь	Є

Щоб створити чат-бота, потрібно визначити завдання, яке він буде вирішувати, вибрати платформу, на якій він працюватиме (Facebook, Telegram тощо), сервіс розробки та сервіс для його запуску. При цьому можна виділити кілька етапів.

### 2.5.1 Визначення вимог:

— ознайомитися з тематикою чат-бота та визначити завдання, які він буде вирішувати;

— сформулюйте вимоги до розробки чат-бота: виберіть платформу для спілкування, виберіть сервіс для розробки та сервіс для запуску бота.



### 2.5.2 Розроблення чат-бота:

- створити комунікаційну архітектуру, яка дозволяє будувати ієрархії вмісту для спілкування;
- зберіть інформацію, необхідну для завершення розмови, щоб навчити робота розуміти різні варіації однієї концепції та давати правильну відповідь, це можна зробити шляхом проведення опитувань серед цільової аудиторії;
- встановлюйте сценарії діалогу відповідно до власних сценаріїв або відповідно до умов, за яких бот відповідає певним чином;
- створюйте контент і моделюйте сценарії діалогу, детально прораховуючи відповіді ботів, щоб спілкування виглядало природним;
- навчайте роботів, розробляючи чітко структуровані сценарії відповіді на основі великих баз даних, або використовуйте готові алгоритми навчання для інтерпретації відповідей за відсутності таких баз даних.

### 2.5.3 Підключення:

- інтегрувати платформу з іншими сервісами, а якщо платформа вже інтегрована з іншими системами, подібну схему інтеграції можна застосувати до чат-ботів;
- виконати тестування роботи чат-бота; активувати бота.

### 2.5.4 Аналіз та вдосконалення роботи:

- збирати та аналізувати інформацію про роботу бота;
- доопрацювати та вдосконалити поведінку чат-бота: включити нові запитання, покращити логіку та класифікацію понять, додати більше відповідей тощо.

Тому розробка чат-ботів має певні особливості та вимагає великої робочої сили та великих професійних навичок.

### 3 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ЧАТ-БОТА

#### 3.1 Огляд засобів створення чат-бота

Оскільки чат-боти набрали вже досить великої популярності, це призвело до того, що з'явилося безліч рішень для їх створення. Створити чат-бота можна двома способами:

- за допомогою мови програмування;
- за допомогою конструктора на сервісі

Загалом чат-бот, написаний певною мовою програмування, являється серверним додатком, в якому функції чату працюють через власний API. Для того, щоб створити такого бота необхідна певна інфраструктура: хостинг, сервер (фізичний або хмарний) та база даних. Можливості таких чат-ботів обмежуються лише можливостями платформи, на яку вони інтегруються.

Натомість чат-бот, який створюється власноруч, за допомогою конструктора, обмежується особливостям сервісу на якому він створюється.

Перед тим як створити чат-бота необхідно зрозуміти як саме відбувається взаємодія між користувачем та сервером [15].

Повідомлення, команди і запити, надіслані користувачами, передаються на програмне забезпечення, запущене на серверах розробників.

Сервер Telegram, який являється посередницьким та анонімним, обробляє шифрування і здійснює зворотний зв'язок між користувачем і утилітою.

Взаємодія між користувачем і ботом виглядає наступним чином:

- Користувач надсилає боту команду
- Бот передає команду на сервер
- Програма на сервері оброблює отриманий від бота запит
- Сервер віддає відповідь боту
- Бот виводить відповідь користувачеві на вікні керування.

І цей цикл повторюється раз за разом під час взаємодії з будь-яким телеграм- ботом. Спілкування відбувається за допомогою простого HTTPS-інтерфейсу, який є спрощеною версією API Telegram. Інакше цей інтерфейс

можна назвати програмним каталогом або бот-алгоритмом.

Схема обміну даними між користувачем і Telegram ботом наведена на рис. 2.1.

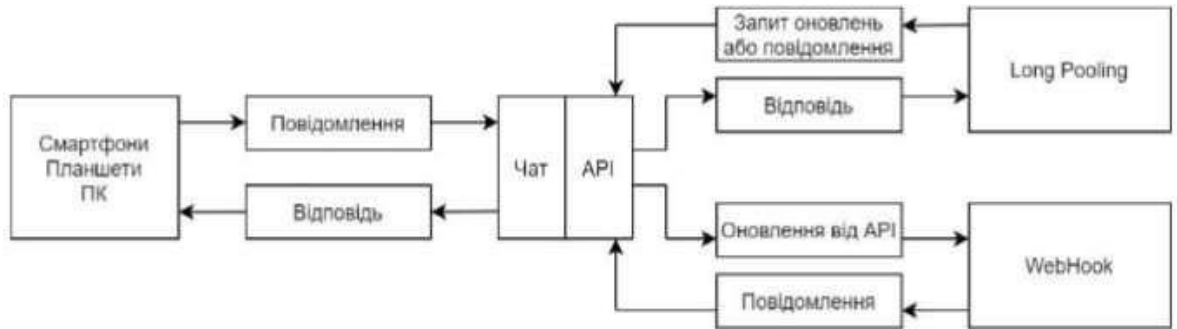


Рис. 2.1. Схема обміну даними між користувачем і Telegram ботом

### 3.2 Telegram Bot API

Для месенджера Telegram був створений протокол MTProto, що передбачає використання декількох протоколів шифрування [8]. Протокол підрозділяється на три практично незалежні компоненти:

- мова API запитів, це компонент високого рівня. З його використанням визначається метод для перетворення запитів та відповідей у двійкові повідомлення;

- криптографічний рівень, з його використанням визначається метод для шифрування повідомлень перед передачею через транспортний протокол;

- транспортний рівень, з його використанням визначається спосіб передачі між клієнтом та сервером повідомлень, за допомогою наявних мережевих протоколів (HTTP, HTTPS, WS, WSS, TCP, UDP).

Bot API — це інтерфейс на основі HTTP для створення ботів в Telegram [10].

Згідно з документацією Telegram Bot API існує два способи отримання оновлень, що відрізняються один від одного:

- за допомогою запитів;
- за допомогою вебхуків.

Під оновленням мається на увазі дія, що була виконана над ботом — наприклад, отримання запиту від користувача. Вхідні оновлення зберігаються до

того моменту, поки сервер не обробить їх, але не більш ніж 24 години.

Спочатку веб-додатки розроблялись навколо моделі клієнт / сервер, де веб- клієнт завжди є ініціатором транзакцій, вимагаючи дані від сервера. Таким чином, не було механізму, щоб сервер самостійно надсилав дані клієнту, не отримуючи до цього запиту.

Щоб подолати цей недолік, розробники веб-програм можуть застосувати техніку, яка називається HTTP Long Pooling, коли клієнт опитує сервер, що вимагає нової інформації. Сервер тримає запит відкритим, поки не з'являться нові дані. Отримавши доступ, сервер відповідає та надсилає нову інформацію. Коли клієнт отримує нову інформацію, він негайно надсилає інший запит, і операція повторюється.

Вебхуки працюють зовсім інакше. Якщо в чат приходить повідомлення, то месенджер Telegram сам говорить про це. Саме в цьому і полягає робота вебхука. У зв'язку з цим, відпадає необхідність періодично опитувати, тим самим, зникає причина помилок пошукових ботів. Однак для використання такої можливості, необхідно платити для установки повноцінного вебсервера.

Telegram Bot API надає декілька доступних методів для роботи:

- `getUpdates`, метод використовується для отримання оновлення за допомогою Long Pooling;
- `setWebhook`, метод, що необхідний для задання URL вебхука, на який бот буде відправляти оновлення;
- `getWebhookInfo`, метод збирає інформацію щодо того чи іншого вебхука;
- `sendMessage`, метод використовується для відправлення текстових повідомлень.

### 3.3 Мова програмування Python

В залежності від необхідної функціональності, створення чат-бота в Telegram досить просто виконується за допомогою використання однієї з серверних мов програмувань: Ruby, Python, PHP, Node.JS.

Серед цих мов, що дозволяють швидко і безпроблемно виконати створення бота, Python є одним з найбільш популярних рішень [11]. В основному це пов'язано з широкими можливостями, доступними як при використанні стандартних бібліотек, так і з застосуванням вже готових варіантів, таких як TelegramBotAPI, розрахованих на роботу безпосередньо з Telegram.

Python — це мова програмування яка допомагає підвищити продуктивність розробника, а не коду, який він пише. Шляхом простоти коду, подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з Java або C++. Ця мова програмування являється універсальною завдяки багатій стандартній бібліотеці, тому широко використовується у різних областях: веброзробці, машинному навчанні, розробці ігор, комп'ютерній безпеці тощо.

Інтерпретатор Python практично був реалізований на всіх платформах і операційних системах, що є безперечною перевагою. Першою такою мовою була C, однак типи даних цієї мови на різних машинах займають різну кількість пам'яті, а це представляє деяку перешкоду при написанні великих програм. Також, велике значення надається розширюваності мови. Як казав сам розробник Python, кожен програміст може вдосконалювати цю мову й ділитися своїми напрацюваннями з іншими. Інтерпретатор був розроблений мовою програмування C і вихідний код доступний для будь-яких змін. Його з легкістю можна додати у свою програму і використовувати як вбудовану оболонку.

Наступна перевага — наявність великого числа модулів, що забезпечують додаткові можливості мови. Ці модулі пишуться на Python, або мовою C і зазвичай створюються більш досвідченими програмістами.

Широковживані вважаються такі модулі:

— Numerical Python, цей модуль надає розширені математичні можливості, такі як маніпуляції з цілими векторами і матрицями;

— Tkinter, модуль, за допомогою якого будуються додатки

використанням графічного інтерфейсу користувача (GUI);

— OpenGL, модуль, завдяки якому отримується доступ до використання великої бібліотеки графічного моделювання двовимірних і тривимірних об'єктів Open Graphics Library;

— Pygame, набір крос-платформових модулів для Python, призначених для створення відеоігор. Містить у собі бібліотеки комп'ютерної графіки і звуку.

Одним з недоліків мови є невелика швидкість виконання програм. Але це сповна компенсується плюсами мови, які були описані раніше.

PyCharm — найпопулярніше середовище розробки, яке використовується спеціально для мови Python та являється кросплатформним. Підтримується на різних операційних системах таких як: MacOS, Linux, Windows. Використання PyCharm допомагає провести аналіз написаного коду, виділити синтаксис та отримати повідомлення про помилки інтерпретації. В цьому середовищі розробки спрощена система навігації по проектах та файлах, що забезпечує швидкий перегляд та перехід між методами, класами та місцями, де вони використовуються або викликаються.

### 3.4 Формат даних JSON

JSON — текстовий формат обміну даними, заснований на JavaScript. JSON вимагає менше коду і має менший розмір, що прискорює обробку і передачу даних. Не дивлячись на те, що JSON написаний на JavaScript, він не залежить від мови. Він не має яких-небудь потужних функцій, пов'язаних з перевіркою і схемою, які є у XML.

Завдяки популярності технології API REST, JSON отримав імпульс в програмуванні коду API і вебсервісів. Це текстовий, легкий і простий для розуміння формат даних, який не потребує додаткового коду для аналізу [13]. Також, JSON немає підтримки простору імен, додавання коментарів або написання метаданих. Цей текстовий формат підтримує кодування UTF і ASCII.

JSON з самого початку був заснований на двох структурах даних:

— Колекція пар ключ / значення. Ця концепція, у різних мовах програмування реалізується як об'єкт, запис, структура, словник, хеш, асоціативний масив, або іменованій список;

— Упорядкований список значень. Більшість мов програмування реалізують це як масив, вектор, список або послідовність.

Такі структури даних являються універсальними. В нотації JSON це виглядає так:

Значення — може приймати властивості об'єкта, масиву, номеру, рядка, true, false, null. Ці структури можуть бути вкладеними. (рис.3.2).

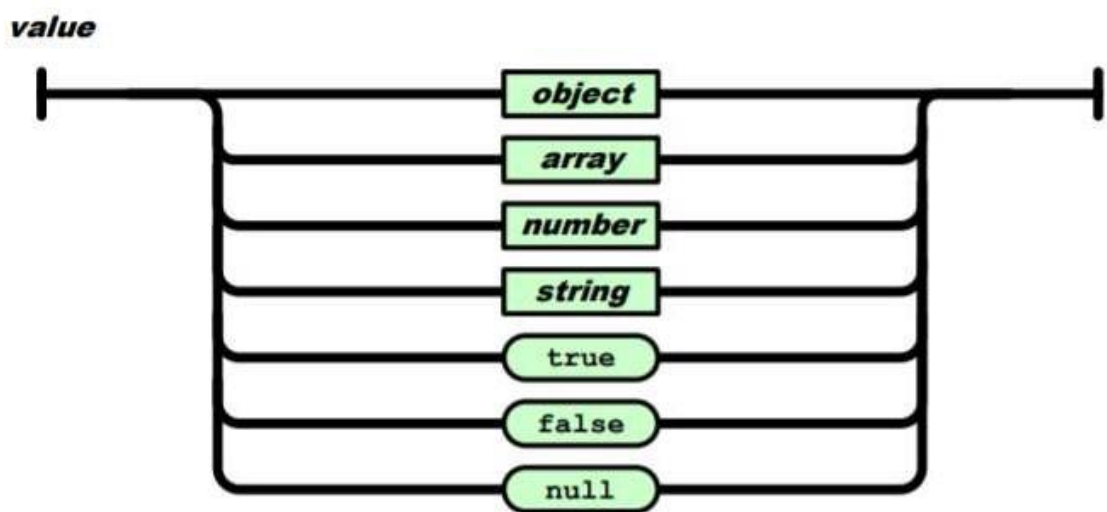


Рис.3.2. Схематичне представлення значення JSON

Структура об'єкта складається з неупорядкованого набору пар ключ / значення. Об'єкт починається з відкритої фігурної дужки і закінчується фігурною дужкою. Кожне ім'я супроводжується двокрапкою, а пари ключ / значення відокремлені один від одного комою. (рис.3.3).

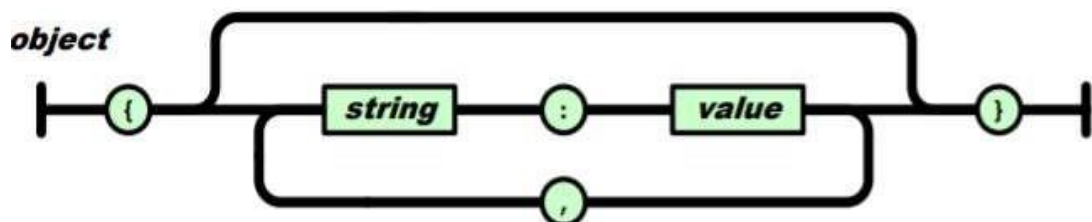


Рис.3.3. Схематичне представлення об'єкта JSON

Структура масиву — це пара квадратних дужок, що оточують нуль або більше значень. Значення відокремлюються один від одного комою. (рис.3.4).

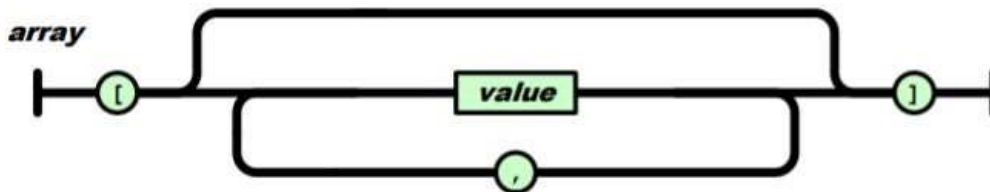


Рис.3.4. Схематичне представлення масиву JSON

Структура рядка складається з послідовності нуля або набору символів Unicode, занесених у подвійні лапки, використовуючи зворотну косу риску в якості символу екранування. Символ представляється як одно символний рядок. (рис.3.5).

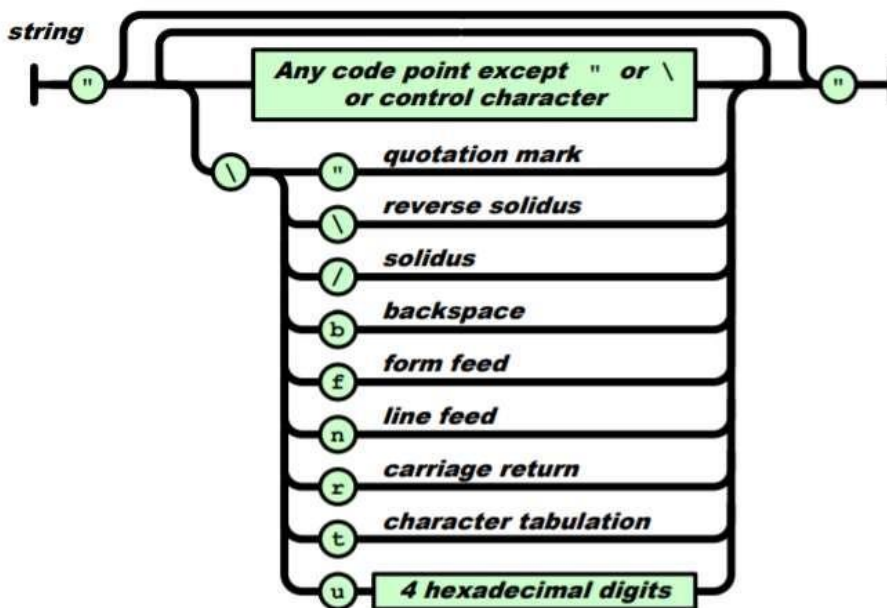


Рис.3.5. Схематичне представлення рядка JSON

У структурі числа в JSON не використовуються восьмирічні або шістнадцятирічні формати, але число передається так, як в мовах програмування С або Java (рис3.6).



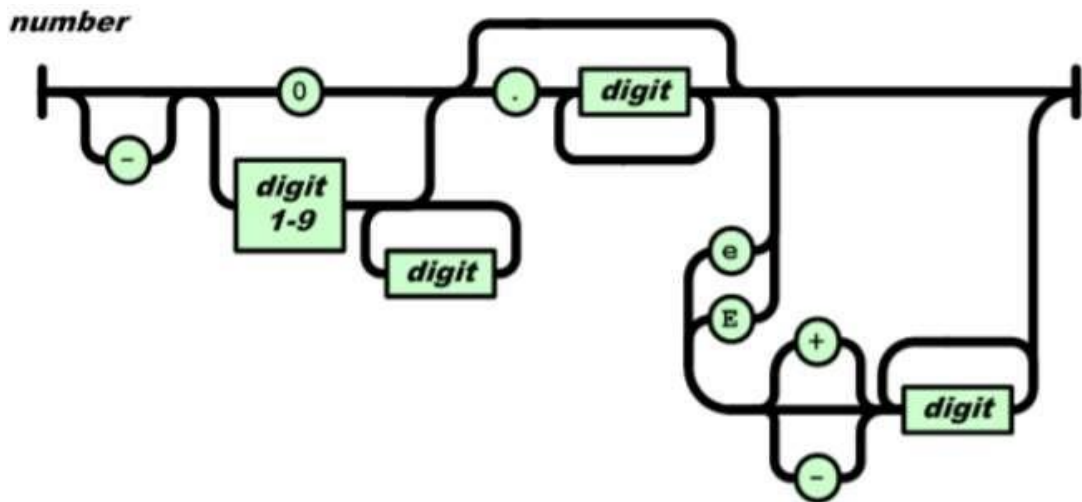


Рис.3.6. Схематичне представлення числа JSON

### 3.5 HTTP та HTTPS запити

Hyper Text Transfer Protocol — протокол передачі даних, для розподілених спільних інформаційних систем, що дозволяє користувачам передавати дані у всесвітній павутині. HTTP — це протокол прикладного рівня, який був винайдений поряд з HTML, щоб створити перший інтерактивний текстовий веббраузер: оригінальний World Wide Web. Нині протокол залишається одним з основних засобів використання Інтернету, адже являється основою взаємодії клієнт-серверної архітектури [12]. Клієнтами часто є браузери (Chrome, Edge, Safari), але вони можуть бути будь-якими типами програм або пристроїв. Сервери — це найчастіше комп'ютери в хмарі.

Клієнти та сервери спілкуються шляхом обміну окремими повідомленнями. Повідомлення, надіслані клієнтом, називаються запитами, а повідомлення, надіслані сервером на запит, називаються відповідями. HTTP — це протокол без збереження стану, тобто сервер не зберігає ніяких даних під час обміну повідомленнями.

Кожен запит (англ. Request) відправляється на сервер, який аналізує його і повертає відповідь (англ. Response).

Для створення HTTP запиту, існує дев'ять методів, але найбільш популярними являються: GET, POST і PUT.

Метод GET використовується для отримання інформації від сервера по

заданому URI ( URI в HTTP ). Запити клієнтів, що використовують метод GET повинні отримувати тільки дані і не повинні ніяк впливати на ці дані.

Для того, щоб відправити повідомлення, які завантажують дані на вебсервер використовуються POST і PUT.

Метод POST застосовується для передачі призначених для користувача даних заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються на серверметодом POST.

Різниця між PUT і POST полягає в тому, що при повторному застосуванні PUT дає той же результат, що і при першому (тобто у методу немає побічних ефектів), тоді як повторний виклик одного і того ж методу POST може мати такі ефекти, як наприклад, оформлення одного і того ж замовлення кілька разів.

Запит HTTP протоколу (рис.2.7) складається із укааних параметрів:

- метод (GET, PUT, POST), що описує дію, яку потрібно виконати
- ціль запиту, зазвичай URL-адреса, або абсолютний шлях протоколу, порту та домену;
- версія протоколу;
- заголовки.

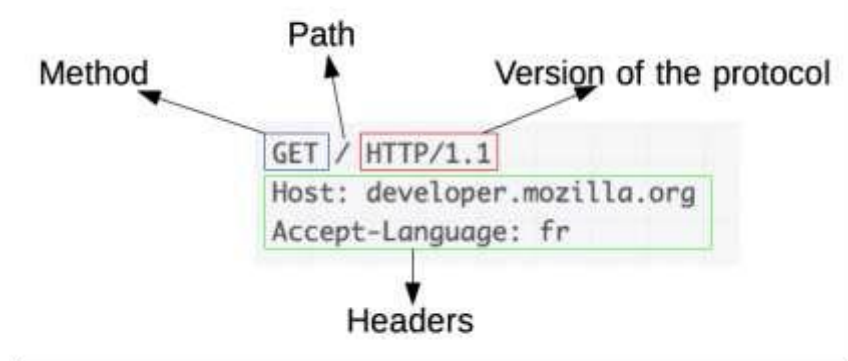


Рис. 3.7. Структура запиту HTTP протокол

Натомість відповідь HTTP (рис.3.8) протоколу містить у собі:

- версію протоколу;
- код стану, який вказує на успіх чи невдачу статусу;

- текст статусу, короткий, інформаційний, текстовий опис коду стану, щоб допомогти людині зрозуміти повідомлення HTTP;
- заголовки;
- тіло (опціональне поле).



Рис. 2.8. Структура відповіді HTTP протокол

В протоколі HTTP є певний недолік, незважаючи на його популярність та простоту — він не являється безпечним протоколом. Повідомлення POST відправляють дані на сервер у вигляді звичайного тексту, який можна легко перехопити. Зазвичай, відповіді, які надходять від сервера, також не зашифровані. Для безпечної роботи через інтернет використовують HTTPS.

HTTPS (англ. HyperText Transfer Protocol Secure) — розширення HTTP, яке підтримує захист даних при транспортуванні за допомогою шифрування інформації відповідно до стандартів SSL і TLS. Такий захист необхідний в комерційних ресурсах, де використовується конфіденційна інформація про персональні або платежів користувача.

### 3.6 База даних MongoDB

Оскільки, Telegram бота можна створити на будь-якій серверній мові програмування, так само до нього можна і під'єднати будь-яку базу даних. В залежності від завдання, і необхідної кількості зберігання інформації, на цю

мить момент існує широкий вибір баз даних. Для створення чат-бота для організації дистанційного навчання, вибір був зроблений на користь MongoDB.

MongoDB — це документно-орієнтована база даних [17]. Завдяки цьому вона працює швидше, має кращу масштабованість, не містить схем, таблиць, зовнішніх ключів та і в цілому досить сильно відрізняється від об'єктно-реляційної бази даних.

Спосіб зберігання даних в MongoDB в цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання в MongoDB застосовується формат, який називається BSON або скорочення від binary JSON. BSON дозволяє працювати з даними швидше: швидше виконувати пошук і обробку інформації.

MongoDB написана на C++, тому її легко перенести на найрізноманітніші платформи. MongoDB може бути розгорнута на платформах Windows, Linux, MacOS, Solaris.

Якщо реляційні бази даних зберігають рядки, то MongoDB зберігає документи. На відміну від рядків документи можуть зберігати складну за структурою інформацію. Документ можна уявити як сховище ключів і значень. Ключ — це мітка, з яким асоційована певна частина даних.

Також MongoDB та у реляційних баз даних є спільна ознака. У реляційних базах використовується таке поняття як первинний ключ, який являється унікальним ідентифікатором певних полів. Схоже поняття існує і в MongoDB, та називається `_id`. Різниця полягає в тому, що в реляційних базах є можливість привласнити пустому полю значення NULL, натомість у MongoDB, якщо ключ не отримує значення, то він не використовується у документі.

У MongoDB замість таблиць створюються колекції. Відрізняються вони від таблиць тим, що можуть містити об'єкти, які мають різну структуру і різний набір властивостей. Натомість таблиці зберігають лише однотипні жорстко структуровані об'єкти.

Для зберігання даних, MongoDB представляє набір реплік. У цьому

наборі є основний вузол, а також може бути набір вторинних вузлів. Всі вторинні вузли зберігають цілісність і автоматично оновлюються разом з оновленням головного вузла. Якщо основний вузол виходить із ладу, то один з вторинних вузлів стає головним.

На відміну від реляційних баз даних, MongoDB дозволяє зберігати різні документи з різним набором даних, однак при цьому розмір документа обмежується 16 Мб. Але MongoDB пропонує рішення — спеціальну технологію GridFS, яка дозволяє зберігати дані за розміром більше, ніж 16 Мб.

Система GridFS складається з двох колекцій. У першій колекції, яка називається files, зберігаються імена файлів, а також їх метадані, наприклад, розмір. А в іншій колекції, яка називається chunks, у вигляді невеликих сегментів зберігаються дані файлів, зазвичай сегментами по 256 Кб. Вся модель бази даних в MongoDB представлена на рис. 2.9.



Рис.2.9. Схематичне зображення моделі бази даних в MongoDB

## 4 ОПИС РОЗРОБКИ ЧАТ-БОТА

### 4.1 Проєктування функціоналу чат-бота

Так як, уже відомо як саме відбувається взаємодія між ботом та сервером, необхідно розробити функціонал. Оскільки цей бот для організації дистанційного навчання, то він продемонструє всі шляхи реалізації обміну даними та файлами між користувачем та викладачем. Тому було вирішено розробити два шляхи реалізації обміну файлами. Перший, який матиме викладач, це обмін за допомогою посилань на вебсервіси. Через те, що лекції відбуваються дистанційно за допомогою відеотелефонного зв'язку, це виявилось гарною можливістю створювати та зберігати запис. Викладач має змогу надіслати посилання на лекцію, або відправити посилання із заздалегідь готовим записом за допомогою чат-бота. Також за бажанням, додати посилання на документи або певні матеріали.

Використання саме такого типу обміну даними має низку переваг:

- інформація в безпеці, при роботі в документі сервіс фіксує будь-яку зміну;
- доступ з будь-якого місця, почати працювати в документі можна з одного пристрою, а закінчити на іншому;
- історія змін, дає змогу повернутись до попереднього варіанту;
- спільна робота в документі, можливість певній категорії людей дозволити редагувати документ, а для інших залишити доступним лише перегляд;
- можливість залишати в роботі коментарі;
- економія місця, всі документи знаходяться у хмарному сховищі.

Але хоч всі можливі сервіси намагаються швидше перейти в режим дистанційного доступу: проведення лекцій, трансляцій, створення документів тощо, але також, ще досі люди користуються створенням файлів на персональних пристроях. Тому в чат-боті продемонстрована можливість, зі сторони студента, надіслати файли викладачу. За правилами Telegram доступна

передача файлів обсягом до п'яти мегабайтів.

Також цей бот містить підключену базу даних, для зберігання документів. Як приклад, якщо необхідно розширити можливості цього бота і зробити його для декількох викладачів, і великого потоку студентів.

Як тільки відбувається взаємодія з ботом, користувач отримує вступне повідомлення-привітання, що коротко описує функції бота. Оскільки це чат-бот, який відповідає за організацію дистанційного навчання, то для нього було створено два інтерфейси взаємодії:

- для студентів;
- для викладача.

У цьому боті є спільні функції, і студенти і викладач мають змогу:

- отримати розклад;
- отримати контактні дані;
- отримати загальні відомості про курс.

Різниця полягає в тому, що студенти завдяки боту можуть:

- отримати завдання;
- надіслати певні матеріали викладачу.

Викладач у свою чергу може:

- надіслати завдання або допоміжні матеріали;
- у разі необхідності видалити завдання;
- замінити певні завдання;
- отримати файли, надіслані студентами у спеціальному каналі Telegram.

## 4.2 Створення чат-бота

Для створення Telegram боту необхідно знайти та запустити BotFather. BotFather — це бот, який допомагає створити нового бота, керує уже існуючими та допомагає їх налаштувати.

Після запуску, BotFather пропонує використати певні команди, але необхідно обрати саме /newbot. Надалі все відбувається дуже просто:

— необхідно написати унікальне ім'я нового бота, адже існування чат-ботів з однаковими іменами — неможливе, якщо бот з такою назвою вже існує, то BotFather запропонує вигадати інше ім'я;

— написати назву бота із закінченням bot у кінці;

— отримати маркер доступу (token) до HTTP API.

Маркер доступу — це унікальний ключ, що складається із набору унікальних цифр та букв, і дає можливість керувати ботом та його функціями. Під час створення програмного забезпечення, розробник використовує цей ключ для ідентифікації бота. Для безпеки бота, маркер доступу, бажано не розповсюджувати.

Після створення бота, розробник отримує певний список команд, для кращого налаштування бота:

— setname — встановити нове ім'я;

— setdescription — змінити опис боту;

— setabouttext — змінити інформацію щодо бота;

— setuserpic — змінити інформацію у профілі бота;

— setcommands — змінити список команд;

— deletebot — видалити бота;

— token — згенерувати новий токен авторизації;

— revoke — анулювати токен.

Ці дії допоможуть створити та налаштувати бота. При виконанні усіх перелічених дій BotFather відправить повідомлення про успіх створення Telegram бота. Але для його навчання на виконання будь-яких функцій потрібно написати відповідний програмний код.

Для реалізації всіх можливостей чат-бота, необхідно також отримати унікальний код власника бота, та код створеного каналу, для отримання файлів від студентів. Це потрібно для того, щоб надісланні дані отримувала лише певна людина, яка являється власником бота та каналу. Щоб отримати ці коди, потрібно скористатись додатковим Telegram ботом.

Всі ці дані записуються кодом в окремому файлі, в якому також



додається з'єднання із базою даних.

Для створення всіх можливих функцій чат-бота необхідно підключити декілька модулів та створити клас, у якому зберігаються всі з'єднання (рис.4.1).

```
import telebot
import pymongo
import ast
from telebot import types
from telebot.types import KeyboardButton

from utils import get_config

class Global:
    # bot - яка створює зв'язок між телеграмом і ботом через Токен
    bot = telebot.TeleBot(get_config()['Telegram']['TOKEN'])

    # Створення зв'язку бази даних з сервером
    client = pymongo.MongoClient(get_config()['DATABASES']['URL'])
    # Підключення до потрібної нам бази даних
    db = client[get_config()['DATABASES']['DB']]
    # Підключення до колекції
    collection = db[get_config()['DATABASES']['COLLECTION']]

    # Ініціалізація змінних {} - словник, [] - Список
    stringList = {}
    content_list = []
    load_list_to_db = {}
    key_word = {}
```

Рис. 4.1. Підключення до модулів та створення класу

Модуль TeleBot інкапсулює всі виклики API в одному класі [14]. Він забезпечує такі функції, як (send\_message, send\_document і т.д.) і кілька способів для прослуховування вхідних повідомлень.

У класі Global створюються всі з'єднання, та змінні, як словник, список і особлива змінна bot, завдяки якій створюється зв'язок між Телеграмом та ботом.

Список — це вид змінної у якій відразу кілька значень. Значення можуть бути різних типів, наприклад в одному списку може бути як і int (ціле число) так і float (число з плаваючою точкою) та навіть string (рядок, літери). Всі дані

в списку впорядковані індексацією. Більш детально всі можливі методи роботи зі списком наведено в додатку

Словник — це неупорядкований список. Замість індексації у всіх даних є свій буквенний ключ. Більш детально всі можливі методи роботи зі словником наведено в табл.4.2.

Таблиця 4.2

Метод	Функція, яку виконує
<code>my_dict.clear ()</code>	Очищує словник
<code>my_dict.clear ()</code>	Очищує словник
<code>my_dict.copy ()</code>	Повертає копію словника
<code>my_dict.get</code> (ключ [, за замовчуванням])	Повертає значення ключа, але якщо його немає, повертає default (за замовчуванням None)
<code>my_dict.items ()</code>	Повертає пари (ключ, значення)
<code>my_dict.keys ()</code>	Повертає ключі
<code>my_dict.pop</code> (ключ [, за замовчуванням])	Видаляє ключ і повертає значення. Якщо ключа немає, повертає default
<code>my_dict.popitem ()</code>	Видаляє пару (ключ, значення). Якщо словник порожній, надсилає виняток <code>KeyError</code>
<code>my_dict.values ()</code>	Повертає значення зі словника

Для створення функціоналу бота використовувались методи `stringList.items` та `stringList.clear` (рис.4.2). Функція дістає всі дані з бази даних і записує їх в змінні, метод `append` доповнює масив з даними, `clear` очищує всі дані з масиву перед записом.

`PyMongo` — це дистрибутив, що містить інструменти для роботи з `MongoDB`, і є рекомендованим способом роботи з цією базою даних на мові `Python`. Один примірник `MongoDB` може підтримувати кілька незалежних баз даних. Під час роботи з `PyMongo` надається доступ до баз даних, використовуючи стиль атрибута `MongoClient`. Як вже відомо колекція

представляє собою групу документів , що зберігаються в MongoDB, і можна розглядати як приблизно еквівалент таблиці реляційної бази даних.

Для обробки повідомлень використовується функція message handler. Після прийому повідомлення від Telegram його необхідно опрацювати в залежності від того, що це за повідомлення: текст, файл, посилання, тощо. Це можливо реалізувати якщо використовувати безліч конструкцій if-then-else,

```
def download_file():
    """
    """
    # Очищення змінних
    Global.content_list.clear()
    Global.stringList.clear()
    # Вибірка об'єктів з БД
    for files in Global.collection.find():
        # Запис ключових слів контенту
        Global.stringList[files['file_name']] = files['file_name']
        # Перевірка на наявність ключа в словнику
        if 'video_url' in files.keys():
            if files['video_url'] == 'Back':
                Global.content_list.append({'file_name': files['file_name'],
                                           'url': files['url']})
            else:
                Global.content_list.append({'file_name': files['file_name'],
                                           'url': files['url'],
                                           'video_url': files['video_url']})
        else:
            Global.content_list.append({'file_name': files['file_name'],
                                       'url': files['url']})
```

Рис. 4.2. Запис файлів у базу даних

проблеми автор бібліотеки pyTelegramBotAPI реалізував механізм handler, який використовує декоратори. Тобто у handler описується, в якому випадку необхідно виконувати туди іншу функцію.

Для реалізації функціоналу чат-бота використовувалось два види кнопок. Кожна кнопка використовується як самостійний об'єкт KeyboardButton або InlineKeyboardButton. Об'єкт KeyboardButton у даній реалізації використовувався для створення основних функціональних кнопок.(рис.4.3).

Кнопки InlineKeyboardButton діляться на три типи: URL-кнопки, Callback- кнопки і Switch-кнопки. Вбудована клавіатура являє собою об'єкт InlineKeyboardMarkup, а кожна така кнопка — це об'єкт InlineKeyboardButton.

Для того щоб створити URL-кнопку, потрібно вказати значення параметрів text (текст на кнопці) URL (веб-адреса). В результаті бот відправить повідомлення із посиланням (рис.4.4).

```
# отримати права вчителя цифри на teacher_id
if _id == 1234:
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=2)
    item_teaching = types.KeyboardButton('Інформація про курс')
    item_teacher = types.KeyboardButton('Викладачі')
    item_schedule = types.KeyboardButton('Розклад')
    item_lessons = types.KeyboardButton('Заняття')
    item_work = types.KeyboardButton('Завантажити роботу')
    item_change = types.KeyboardButton('Замінити роботу')
    item_delete = types.KeyboardButton('Видалити роботу')
    markup.add(item_teaching, item_teacher, item_schedule, item_lessons, item_work,
               item_change, item_delete)
    Global.bot.send_message(message.chat.id, "Вас вітає інформаційний чат-бот д...
                               "Будь ласка, використовуючи кнопки"
```

Рис.4.3. Частина коду опису створення кнопок

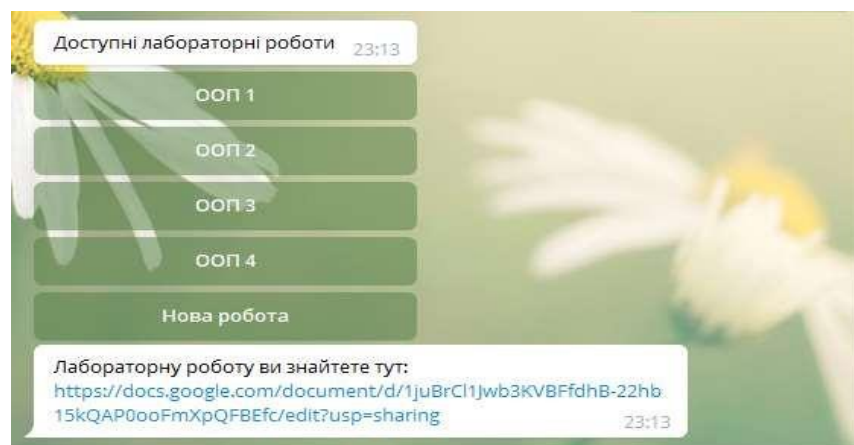


Рис.4.4. Приклад роботи URL кнопок

Кнопки Callback дозволяють виконувати довільні дії після їх натискання. Все залежить від того, які параметри кожна кнопка в собі несе. Відповідно, всі натискання будуть приводити до відправки боту об'єкта CallbackQuery, який містить поле data, в якому написаний певний рядок, закладений у кнопку, а також об'єкт Message, якщо повідомлення надіслано ботом в звичайному режимі. В даному випадку вона завантажує посилання на матеріали (рис.4.5).

```

@Global.bot.callback_query_handler(func=lambda call: True)
def handle_query(call):
    """
    """
    # Перевірка чи є значення на яке ми натиснули в словнику
    if call.data.startswith("value"):
        value_from_call_back = ast.literal_eval(call.data)[1]
        for x in Global.content_list:
            if x['file_name'] == value_from_call_back:
                Global.bot.answer_callback_query(callback_query_id=call.id,
                                                show_alert=True,
                                                text="You Clicked " + value_from_call_back)

            if 'video_url' in x.keys():
                Global.bot.send_message(chat_id=call.from_user.id, text=f"Лабораторну роботу ви знайт  

                f"Допомога при виконнанні: \n"  

                f"{x['video_url']}")

        else:
            Global.bot.send_message(chat_id=call.from_user.id, text="Лабораторну роботу ви знайт

```

Рис.4.5. Частина коду опису створення посилання для кнопки

Для взаємодії з базою даних було створено функції запису, заміни та видалення файлів. База даних створена для інтерфейсу викладача, тому вона лише зберігає назву, посилання та код об'єкта. Для того, щоб завантажити дані використовувалась функція `load_file`, яка приймає словник і через `insert_one` записує дані до бази даних (рис.4.6). Аналогічно для заміни використовувались `update_one`, а для видалення `delete_one`.

```

def load_file(send_dict):
    """
    Функція приймає словник і через insert_one записує дані до БД
    :param send_dict:
    :return:
    """
    if send_dict['video_url'] == 'Back':
        Global.collection.insert_one(
            {'file_name': str(send_dict['Name']),
             'url': str(send_dict['url'])})
    else:
        Global.collection.insert_one(
            {'file_name': str(send_dict['Name']),
             'url': str(send_dict['url']),
             'video_url': str(send_dict['video_url'])})

```

Рис.4.6. Частина коду для роботи з базою даних

Функція `rolling` запускає `Long Polling`, це означає, що бот повинен намагатися не припиняти роботу при виникненні будь-яких помилок. При цьому, звісно ж, за роботою бота потрібно стежити, бо сервера Telegram періодично перестають відповідати на запити або роблять це з великою затримкою приводячи до помилок.

### 4.3 Тестування чат-бота

Оскільки чат-бот створений для месенджера Telegram, то для тестування можна використовувати пристрої, що підтримують з ним взаємодію: комп'ютер, ноутбук, смартфон, тощо. Також користувач може використовувати веб версію Telegram.

Тестування чат-бота здійснювалося вручну, по заздалегідь підготовленим тестовим сценаріям:

- початок роботи: швидкість відгуку;
- швидкість відгуку на запит;
- коректне завантаження тексту;
- коректне завантаження посилань;
- перевірка коректного запису, читання та видалення з бази даних;
- перевірка надсилання документів різних форматів;
- перевірка відображення отриманих робів в каналі Telegram.

Тестування боту відбувалось:

- на комп'ютері в якості інтерфейсу викладача: операційна система Windows, з використанням додатку Telegram Desktop версія 2.7.4
- на смартфоні Samsung в якості інтерфейсу студента, операційна система Android, з використанням додатку Telegram версія 7.7.2

Перш за все проводилось тестування бота проводилось паралельно для обох інтерфейсів. Після приєднання до чату користувачу доступна тільки одна команда /start. Відправивши цю команду, у відповідь отримуємо основну інформацію про бота (рис.4.7).

Після цього отримуємо можливість взаємодії з ботом за допомогою кнопок.

Кнопки «Інформація про курс», «Викладачі», та «Розклад» являються інформативними, тобто з їх використанням можна отримати необхідну інформацію.

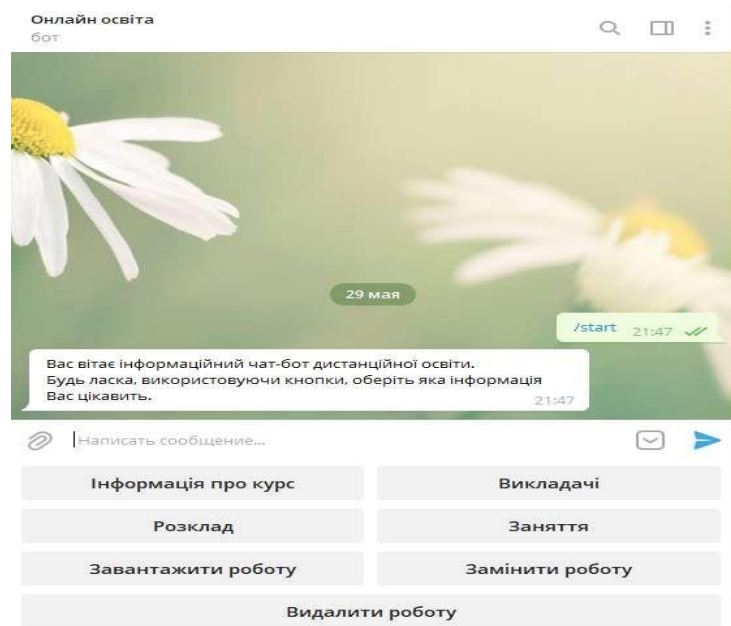


Рис.4.7. Початок роботи із ботом

Кнопка «Завантажити роботу» з боку взаємодії викладача, використовується для завантаження посилання із матеріалами. Бот пропонує ввести назву роботи та додати посилання. Також в разі необхідності є можливість прикріпити додаткові матеріали, або натиснути кнопку «Back», якщо в цьому немає необхідності (рис.4.8).



Рис.4.8. Завантаження роботи до чат-боту за допомогою посилання



В результаті робота з'явилась у переліку занять у вигляді кнопки. Якщо натиснути на цю кнопку, отримаємо необхідне посилання Також посилання на цю роботу з'явилось у базі даних (рис.4.9).

Якщо необхідно замінити певне завдання, для цього використовується кнопка «Замінити роботу». Чат-бот пропонує ввести назву роботи та ввести нове посилання. В результаті посилання замінюється як в боті так і в базі даних.

Якщо необхідно замінити певне завдання, для цього використовується кнопка «Замінити роботу». Чат-бот пропонує ввести назву роботи та ввести нове посилання. В результаті посилання замінюється як в боті так і в базі даних.

За допомогою кнопки «Видалити роботу», користувач має можливість видалити роботу. Для цього, аналогічно, потрібно ввести назву роботи, яку необхідно видалити. Якщо користувач вводить некоректно назву роботи, яку хочезмінити або видалити, то отримує повідомлення про помилку. Якщо дані введені вірно — робота видаляється повністю із бота та бази даних.

Якщо розглядати бота із інтерфейсу студента, то він відрізняється лише наявністю однієї кнопки «Завантажити роботу». За правилами Telegram користувач може надіслати файли формату: .PDF, .DOC, .MP4, .JPEG і обсяг файлу не повинен перевищувати 5 Мб. Якщо обсяг файлу буде більше користувач отримає повідомлення про помилку. Для того щоб відправити роботу, бот запропонує ввести назву роботи Коли студент надсилає роботу, викладач отримує її у спеціально створеному Telegram каналі (рис. 4.9).

У результаті створення третього розділу, розроблений інформаційний чат-бот для організації дистанційної освіти на базі месенджера Telegram. Він містить два інтерфейси взаємодії для організації обміну даними між студентом та викладачем. Різниця полягає в тому, що інтерфейс викладача більш розширений та пов'язаний з базою даних MongoDB, для можливості розширення функціоналу, на випадок, якщо буде велика кількість студентів та файлів.



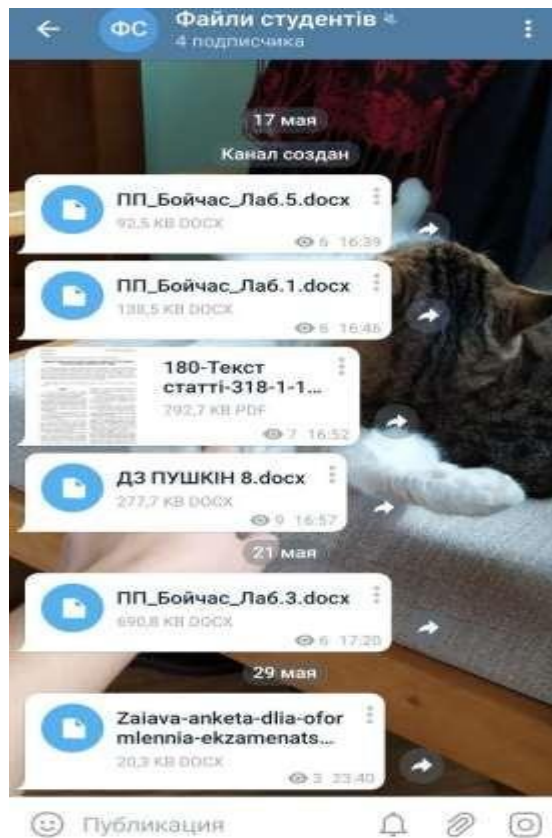


Рис. 4.9. Канал який зберігає файли студентів

Також створений другий інтерфейс взаємодії зі сторони студента. Студент у свою чергу може переглядати різні необхідні дані, такі як розклад, заняття, інформацію про курс та контактні дані викладачів. Також студент має можливість надсилати чат-боту файли у чотирьох різних форматах: .PDF, .DOC, .MP4, .JPEG і обсяг файлу не повинен перевищувати 5 Мб. Це пов'язано з правилами Телеграму. Надіслані студентом файли викладач має можливість бачити у спеціально створеному каналі. Так, як Telegram являється хмарним месенджером, то канал може містити необмежену кількість файлів. Створення такого чат-боту дає можливість організувати дистанційне навчання, в залежності від формату проведення занять, та необхідної взаємодії між студентом та викладачем.

В цьому розділі описаний процес розробки та тестування даного чат-бота. Для реалізації використовувалась мова програмування Python, з модулями: Telebot, pymongo та база даних MongoDB.

## 5 ЕКОНОМІЧНА ЧАСТИНА

### 5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку інтеграція чат боту у веб-застосунок для полегшення комунікації з користувачем — надання інформації користувачу. Чат-бот надає можливість вести спілкування з багатьма користувачами одночасно, не дивлячись на велику кількість задіяних сайтів та користувачів на них.

Особливістю програми є те, що дана технологія не потребує прямого втручання адміністратора.

Аналогів розробки є відносно небагато, а приблизна ціна інтеграції чат-боту такого типу від 1500 до 1800 USD / 60000-72000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертн	Концепція підтверджена розрахункам	Концепція перевірена на практиці	Перевірено роботоздат-ність продукту в реальних умовах
Ринкові переваги					
	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни	Ціна продукту значно нижче за

Продовження таблиці 5.1

	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в ана-	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в
Ринкові перспективи					
	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практик на здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
0	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виро-
1	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

## Завершення таблиці 5.1

2	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
---	---	--	---	--	---

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	4	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	3	3	4
Сума	43	41	44
Середньоарифметична сума балів	$(43+41+44) / 3 = 42,67$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький

Продовження таблиці 5.3

11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що інтегрований чат-бот надає можливість вести спілкування з багатьма користувачами одночасно, не дивлячись на велику кількість задіяних сайтів та користувачів на них. Особливістю програми є те, що дана технологія не потребує прямого втручання адміністратора.

## 5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де  $M$  — місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_p$  — число робочих днів в місяці, 20 днів;

$t$  — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 — Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	35500	1775,00	35	62125,000
Програміст	33000	1650,00	35	57750,000
Всього				119875,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 14,8 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 14,8 \% / 100 \% \quad (5.2)$$

$$Z_d = (119875,00 \cdot 14,8 \% / 100 \% ) = 17741,50 \text{ (грн.)}$$

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (119875,00 + 17741,50) \cdot 22 \% / 100 \% = 30275,63 \text{ (грн.)}$$

Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T_v} \cdot \frac{t_{вик}}{12} \text{ [Грн.]} \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$  — термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер

балансова вартість якого становить 32000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його фактичного використання — 1,75 міс.

$$A_{обл} = \frac{32000}{2} \times \frac{1,75}{12} = 2333,33 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.2.

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив (IDE PyCharm, підписка 9960 грн/рік, отже за 1,75 місяця = 1452,5 грн.) не амортизується, а його вартість включається у вартість розробки повністю,  $V_{нем.ак.} = 1452,5$  грн.

Таблиця 5.6 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (Acer Aspire 7)	32000	2	1,75	2333,33
Офісне обладнання (меблі)	23500	4	1,75	856,771
Приміщення	1050000	20	1,75	7656,250
Всього				10846,35

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл

електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (4.5)$$

де  $V$  — вартість 1 кВт-години електроенергії для 1 класу підприємства,  $V = 6,2$  грн./кВт;

$\Pi$  — встановлена потужність обладнання, кВт.  $\Pi = 0,45$  кВт;

$\Phi$  — фактична кількість годин роботи обладнання, годин.

$K_{\Pi}$  — коефіцієнт використання потужності,  $K_{\Pi} = 0,9$ .

$$V_e = 0,9 \cdot 0,45 \cdot 8 \cdot 35 \cdot 6,2 = 703,08 \text{ (грн.)}$$

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:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{iB}}{100\%}, \quad (4.6)$$

де  $H_{iB}$  — норма нарахування за статтею «Інші витрати».

$$I_B = 119875,00 \cdot 88\% / 100\% = 105490 \text{ (грн.)}$$

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:

$$H_{\text{нзв}} = (3_o + 3_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.7)$$

де  $H_{\text{нзв}}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{\text{нзв}} = 119875,00 * 135 \% / 100 \% = 161831 \text{ (грн.)}$$

### 5.3 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{\text{заг}} = 119875,00 + 17741,50 + 30275,63 + 10846,35 + 1452,5 + 703,08 + 105490 + 161831 = 448215,31 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

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

$$ЗВ = \frac{B_{\text{заг}}}{\eta} \text{ (грн)}, \quad (4.8)$$

де  $\eta$  – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то  $\eta=0,1$ ; технічного проектування, то  $\eta=0,2$ ; розробки конструкторської документації, то  $\eta=0,3$ ; розробки технологій, то  $\eta=0,4$ ; розробки дослідного зразка, то  $\eta=0,5$ ; розробки промислового зразка, то  $\eta=0,7$ ; впровадження, то  $\eta=0,9$ . Оберемо  $\eta = 0,5$ , так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 448215,31 / 0,5 = 896431 \text{ грн.}$$

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

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

— вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

— зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

— кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

— визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.10)$$

де  $\pm\Delta\Pi_0$  – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$\Pi_0$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,  $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$ ;

$\Pi_6$  – вартість програмного продукту у році до впровадження результатів розробки;

$\Delta N$  – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ .

$r$  – коефіцієнт, який враховує рентабельність продукту;

$\vartheta$  – ставка податку на прибуток, у 2022 році  $\vartheta = 18\%$ .

Припустимо, що при прогнозованій ціні 20000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 1500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 2500 шт., протягом другого року – на 2000 шт., протягом третього року на 1500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*1500 + (20000 + 1500)*2500)* 0,8333* 0,24) * (1 - 0,18) = 8199999,672 \text{ грн.}$$

$$\Delta\Pi_2 = (0*1500 + (20000 + 1500)*(2500+2000)* 0,8333* 0,24) * (1 - 0,18) = 15866999,365 \text{ грн.}$$

$$\Delta\Pi_3 = (0*1500 + (20000 + 1500)*(2500+2000+1500)* 0,8333* 0,24) * (1 - 0,18) = 21155999,154 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 45222998,19 грн.

5.5 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

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

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.11)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

$T$  – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;

$t$  – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$ПП = (8199999,672 / (1+0,1)^1) + (15866999,365 / (1+0,1)^2) + (21155999,154 / (1+0,1)^3) = 7454545,16 + 13113222,62 + 15894815,29 = 36462583,06 \text{ грн.}$$

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

$$PV = k_{\text{інв}} * ЗВ, \quad (5.12)$$

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  $k_{\text{інв}}=2\dots 5$ , але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 896431 = 1792861,26 \text{ грн.}$$

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

$$E_{\text{абс}} = \text{ПП} - PV, \quad (5.13)$$

$$E_{\text{абс}} = 36462583,06 - 1792861,26 = 34669721,81 \text{ грн.}$$

Оскільки  $E_{\text{абс}} > 0$  то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) 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.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_e$ . Для цього використаємо формулу:

$$E_g = T_{жс} \sqrt[3]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.14)$$

$T_{жс}$  – життєвий цикл наукової розробки, роки.

$$E_B = 3 \sqrt[3]{1 + 34669721,81/1792861,26} - 1 = 1,730$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d = (0,09...0,14)$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,5)$ .

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як  $E_B > \tau_{\min}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (5.16)$$

$$T_{ок} = 1 / 1,730 = 0,58 \text{ р.}$$

Оскільки  $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,58 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких

складає 896431 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,58 роки.



## ВИСНОВКИ

В ході виконання магістерської кваліфікаційної роботи, було проаналізовано, що дистанційне навчання являється актуальною темою, і тому створення чат-бота для організації дистанційного навчання, являється чудовою альтернативою різним платформам та додаткам, зі схожою функціональністю.

У роботі проаналізовані особливості існуючих підходів до створення чат-ботів, а саме:

— обґрунтовано, що чат-боти є одним з найбільш перспективних напрямків в найближчому майбутньому і є актуальною темою для багатьох наукових досліджень — обумовлено двома факторами, активне використання месенджерів і розвиток технологій штучного інтелекту;

— розроблена класифікація чат-ботів (за типами користувачів, видами інтерфейсів, способам доступу до боту, призначенням, принципом роботи);

— обґрунтована перевага нейронних мереж для створення чат-ботів, які працюють на основі навчання; описана модель взаємодії чат-ботів з платформами месенджерів через Webhook;

проведено порівняльний аналіз безкоштовних платформ і сервісів для аналізу взаємодії користувачів з ботом, який дає рекомендації для практичного використання інструментальних засобів при створенні ботів;

— визначено зміст основних етапів створення чат-ботів.

Відбулося дослідження технологій для побудови чат-бота. Виявлено, що обмін повідомлень у чат-боті відбувається шляхом клієнт-серверної архітектури. Всі необхідні дані передаються по HTTP протоколу за допомогою формату даних JSON. Для чат-бота продуманий функціонал для обміну інформації між викладачем та студентом. Завдяки тому, що дистанційне навчання нині відбувається з використанням вебсервісів, то викладач має змогу за допомогою бота надсилати посилання на запис лекції, на документи

або на певні матеріали, адже це має ряд переваг. Але також розглянутий другий варіант функціоналу, обміну інформації за допомогою звичайних файлів. Таку можливість продемонстровано з боку студента. Також підключена база даних MongoDB, як можливість для викладача зберігати велику кількість посилань у разі збільшення кількості студентів, або предметів.

Для реалізації обрано наступну мову програмування та технології: Python, модуль Telebot, модуль Pymongo, Telegram API та база даних MongoDB.

Результатом проведеного дослідження стало створення чат-бота для організації дистанційного навчання. Система демонструє всі можливі шляхи реалізації організації дистанційного навчання на базі месенджера Telegram.

Створений чат-бот розглядається як сучасна система, з можливістю розширення функціоналу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1.Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів / О.І. Провотар, Х.А. Ключко // Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка. – 2013. – № 3. – 6 с.
- 2.Сухас У. Oracle Intelligent Bots: Чат-боты с искусственным интеллектом [Електронний ресурс] / У. Сухас. –Oracle, 2017. – Режим доступу: <https://www.oracle.com/a/ocom/docs/chatbots.pdf>.
- 3.Украинец И. Системы аналитики для чат-ботов [Електронний ресурс] / И. Украинец. – Режим доступу: <http://ilyaukrainets.ru/chatbot/sistemy-analitiki-dlya-chat-botov/>.
4. Ясницкий Л.В. Интеллектуальные системы/ Л.В. Ясницкий. – М.: Лаборатория знаний, 2016. – 221 с.
5. A curated directory of chat bot resources & tools [Електронний ресурс]. – Режим доступу: <http://www.botsfloor.com/botstash/products/?category=Development%20Platfors>.
6. Beaver L. The Chatbots Explainer [Електронний ресурс] / L. Beaver. – BI Intelligence Copyright, 2016. – 23 p. – Режим доступу: <https://www.businessinsider.com/intelligence/chatbots-explainer>.
- 7.Chatbot Report 2018: Current landscape of how people create chatbots and how users expect to interact with them [Електронний ресурс]. – Режим доступу: <https://elearningindustry.com/chatbots-for-learning-support-10-reasonshttps://naiz.chat/NAIZ-report-18072018.pdf>.
8. Gartner Identifies Five Emerging Technology Trends That Will Blur the Lines Between Human and Machine [Електронний ресурс]. – Режим доступу: <https://www.gartner.com/en/newsroom/press-releases/2018-08-20-gartner-identifies-five-emerging-technology-trends-that-will-blur-the-lines-between-human-and-machine>.
- 9.Goncalves P. 10 graphs that show why your business should be available

through messaging apps [Електронний ресурс] / P. Goncalves. – Режим доступу: <https://medium.com/hijiffy/10-graphs-that-show-the-immense-power-of-messaging-apps-4a41385b24d6>

10. Kuang C. Why Chat May Be King Of The New Mobile Landscape [Електронний ресурс] / С. Kuang. – Режим доступу: <https://www.fastcompany.com/3064055/why-chat-may-be-king-of-the-new-mobile-landscape>.

11. Kumar C. 9 AI Platform to Help you in Creating Facebook Chatbot [Електронний ресурс] / С. Kumar. – Режим доступу: <https://geekflare.com/create-facebook-chatbot/>.

12. Messaging apps are now bigger than social networks [Електронний ресурс]. – Режим доступу: <https://www.businessinsider.com/the-messaging-app-report-2015-11>.

13. Michiels. E. Modelling Chatbots with a Cognitive System Allows for a Differentiating User Experience [Електронний ресурс] / E. Michiels. – Режим доступу: <http://ceur-ws.org/Vol-2027/paper24.pdf>.

14. Muldowney O. Chatbots. An Introduction And Easy Guide To Making Your Own / O. Muldowney. – Dublin, Curses & Magic, 2017. – 69 p.

15. On Their Tenth Anniversary, Mobile Apps Start Eating Their Own [Електронний ресурс]. – Режим доступу: <https://flurrymobile.tumblr.com/post/155761509355/on-their-tenth-anniversary-mobile-apps-start>.

16. Sheth B. The Bot Lifecycle. What to know before you make your chatbot [Електронний ресурс] / B. Sheth. – Режим доступу: <https://chatbotsmagazine.com/the-bot-lifecycle-1ff357430db7>.

17. ТОП 10 Найбільш популярних месенджерів України та світу. [Електронний ресурс] // [eo-marketing.com.ua](http://eo-marketing.com.ua) – 2020. – Режим доступу: <https://seo-marketing.com.ua/top-10-naibilsh-populiarnykh-mesendzheriv-ukrainy-ta-svitu/>

18. Reach of mobile messenger apps among Android users in Ukraine in

June 2020, by platform. [Электронный ресурс] / statista.com – 2020. – Режим доступа: <https://www.statista.com/statistics/1188579/most-popular-messengers-in-ukraine/>

19. MTProto Mobile Protocol. [Электронный ресурс] / core.telegram – 2017. — Режим доступа: <https://core.telegram.org/mtproto>

20. Five Different Types of Chatbot. [Электронный ресурс] / medium.com – 2019. – Режим доступа: <https://medium.com/voiceui/five-different-types-of-chatbot-17bb255b23b4>

21. Bots: An introduction for developers. [Электронный ресурс] / core.telegram 2017. – Режим доступа: <https://core.telegram.org/bots>

**ДОДАТОК А**

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ  
Завідувач кафедри ОТ  
проф., д.т.н..  
\_\_\_\_\_Азаров О.Д.

" \_\_\_\_ " \_\_\_\_\_ 2022 р.

**ТЕХНІЧНЕ ЗАВДАННЯ**  
на виконання магістерської кваліфікаційної роботи  
“Комп'ютерна система моніторингу регіонального радіомовлення”  
08-23. МКР.013.00.003.ТЗ

Науковий керівник: проф.  
\_\_\_\_\_ Захарченко С.М.

Студент групи 1КІ-21м  
\_\_\_\_\_ Слободян А.О.

## 1 Підстава для виконання

Підставою для виконання МКР є розкриття теми інтеграції чат-ботів до web-застосунків для усіх людей та аналіз сучасних методів створення. Актуальність роботи полягає в розробці чат-боту у якому буде високий рівень комфорту дистанційного навчання. Що дозволить збільшити продуктивність студентів і викладачів.

## 2 Мета МКР і призначення розробки

2.1 Мета проекту — огляд сучасних методів розробки чат-ботів, аналіз популярних методів інтеграції чатботів, аналіз способів інтеграції чат-ботів та їх розробки.

2.2 Призначення розробки — створення чат-боту з можливістю покращення методів інтеграції;

## 3 Вихідні дані для виконання МКР

3.1 Розкриття актуальності роботи над доступністю та важливість дистанційного навчання;

3.2 Аналіз усіх можливих проблем чат-ботів за параметром доступності та огляд методів вирішення цих проблем;

3.3 Розробити чат-бот з використанням розглянутих методів;

3.4 Виконання розрахунків для доведення доцільності нової розробки з економічної точки зору;

3.5 Протестувати чат-бот за параметром за параметром швидкодії;

## 4 Вимоги до виконання МКР;

Головна вимога — використати, методи інтеграції чат боту для досягнення максимального результату за параметром доступності

## 5 Етапи МКР та очікувані результати;

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз характеристики чат-ботів			Розділ 1
2	Огляд сучасних технологій розробки чат-ботів.			Розділ 1
3	Аналітичний огляд методів для розробки чат-ботів.			Розділ 2
4	Опис засобів для розробки чат-боту із використанням методів інтеграції.			Розділ 3
5	Розробка та тестування чат-боту за параметром доступності.			Розділ 4
6	Підготовка економічної частини			Розділ 5
7	Апробація та впровадження результатів дослідження			Тези доповідей
8	Опублікування результатів досліджень			Стаття
9	Оформлення пояснювальної записки, графічного матеріалу і презентації			ПЗ, графіч. матеріал і презентація
10	Підготовка супроводжуючих документів, їх підписування, проходження нормоконтролю та тесту на плагіат			Оформлені документи

## 6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

## 7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами.



Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

## 8 Вимоги до оформлювання та порядок виконання МКР

### 8.1 При оформлювання МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— міждержавний ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022;

— документами на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

**ДОДАТОК Б**

## Лістинг блоку реєстрації

```
import csv

import pymorphy2

import re

import psycopg2

conn = psycopg2.connect(dbname='energy', user='mao', password='daring',
host='localhost')

cursor = conn.cursor()

morph = pymorphy2.MorphAnalyzer(lang='ua')

answer_id=[]

answer = dict()

cursor.execute('SELECT id, answer FROM app.chats_answer;')

records = cursor.fetchall()

for row in records:

    answer[row[0]]=row[1]

questions=[]
```

```
cursor.execute('SELECT question, answer_id FROM app.chats_question;')
records = cursor.fetchall()
transform=0

for row in records:
    if row[0]>"":
        if row[1]>0:
            phrases=row[0]
            words=phrases.split(' ')
            phrase=""
            for word in words:
                word = morph.parse(word)[0].normal_form
                phrase = phrase + word + " "
            if (len(phrase)>0):
                questions.append(phrase.strip())
                answer_id.append(row[1])
                transform=transform+1

#print (questions)
#print (answer)
#print (answer_id)

cursor.close()
conn.close()
```

```
import sklearn

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.decomposition import TruncatedSVD

vectorizer_q = TfidfVectorizer()
vectorizer_q.fit(questions)
matrix_big_q = vectorizer_q.transform(questions)
print ("Размер матрицы: ")
print (matrix_big_q.shape)

if transform>200:
    transform=200
print(transform)
svd_q = TruncatedSVD(n_components=transform)
svd_q.fit(matrix_big_q)
matrix_small_q = svd_q.transform(matrix_big_q)
print ( svd_q.explained_variance_ratio_.sum())

# тело программы k=5, temperature=10.0 можно подбирать

import numpy as np

from sklearn.neighbors import BallTree

from sklearn.base import BaseEstimator
```

```
def softmax(x):  
    proba = np.exp(-x)  
    return proba / sum(proba)  
  
class NeighborSampler(BaseEstimator):  
    def __init__(self, k=5, temperature=10.0):  
        self.k=k  
        self.temperature = temperature  
    def fit(self, X, y):  
        self.tree_ = BallTree(X)  
        self.y_ = np.array(y)  
    def predict(self, X, random_state=None):  
        distances, indices = self.tree_.query(X, return_distance=True, k=self.k)  
        result = []  
        for distance, index in zip(distances, indices):  
            result.append(np.random.choice(index, p=softmax(distance * self.temperature)))  
        return self.y_[result]  
  
from sklearn.pipeline import make_pipeline  
  
ns_q = NeighborSampler()  
ns_q.fit(matrix_small_q, answer_id)  
pipe_q = make_pipeline(vectorizer_q, svd_q, ns_q)
```

```

import re

import telebot

telebot.apihelper.ENABLE_MIDDLEWARE = True

bot = telebot.TeleBot("299999999:sdfgnreognrtgortgmrtgm")

@bot.message_handler(commands=['start'])

def start_message(message):

    bot.send_message(message.from_user.id, "?")

@bot.message_handler(func=lambda message: True)

def get_text_messages(message):

    request=message.text

    words= re.split('\W',request)

    phrase=""

    for word in words:

        word = morph.parse(word)[0].normal_form

        phrase = phrase + word + " "

    reply_id  = int(pipe_q.predict([phrase.strip()]))

    bot.send_message(message.from_user.id, answer[reply_id])

    print("Запрос:", request, " \n\: ", phrase, " \n\t\tОТВЕТ :", answer[reply_id])

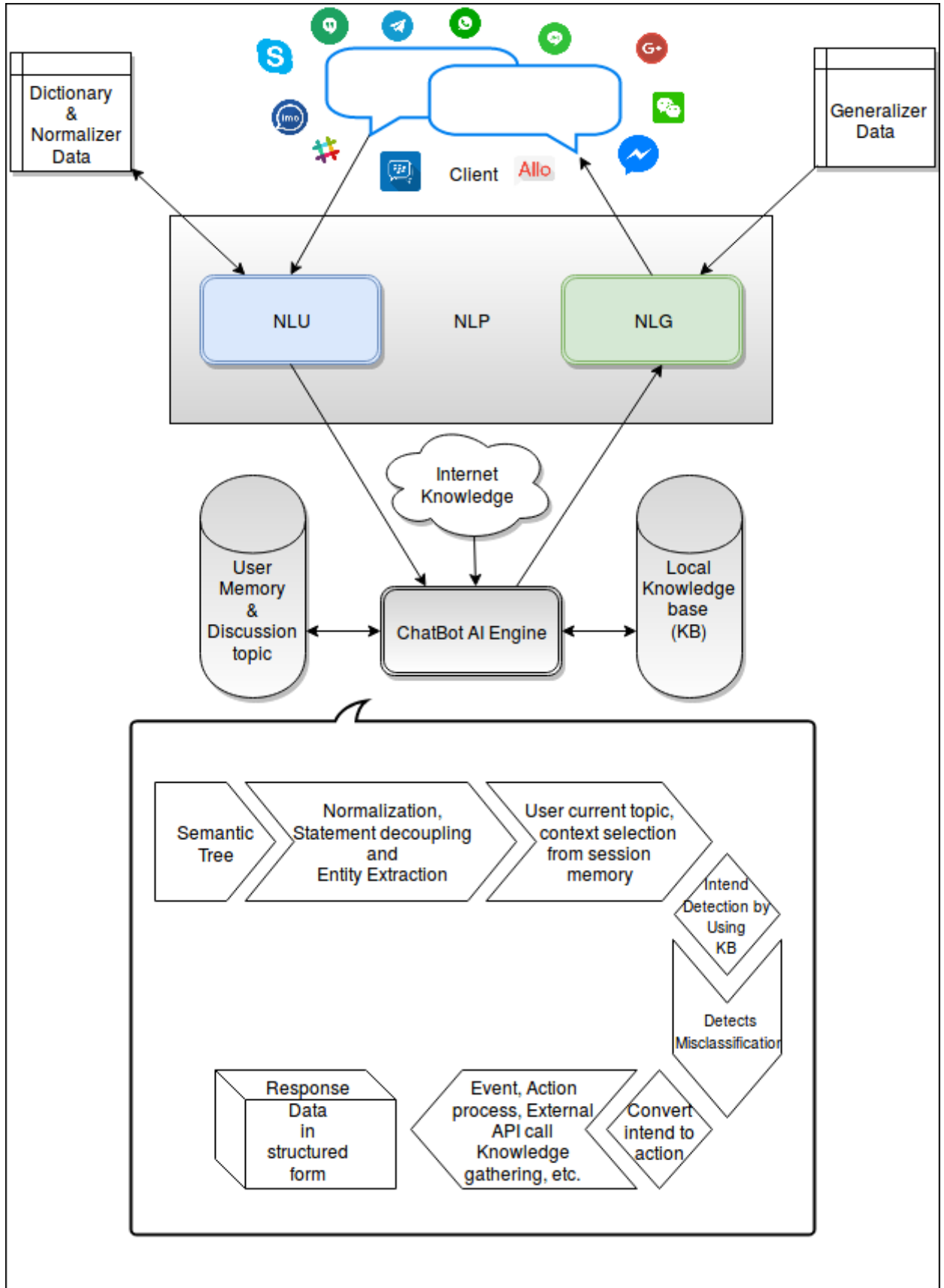
bot.infinity_polling(none_stop=True, interval=1)

```

```
print("Пишите ваш вопрос, слова exit или выход для выхода")
request=""
while request not in ['exit', 'выход']:
    request=input()
    words= re.split('\W',request)
    phrase=""
    for word in words:
        word = morph.parse(word)[0].normal_form
        phrase = phrase + word + " "
    reply_id = int(pipe_q.predict([phrase.strip()]))
    print (answer[reply_id])
```

# ДОДАТОК В

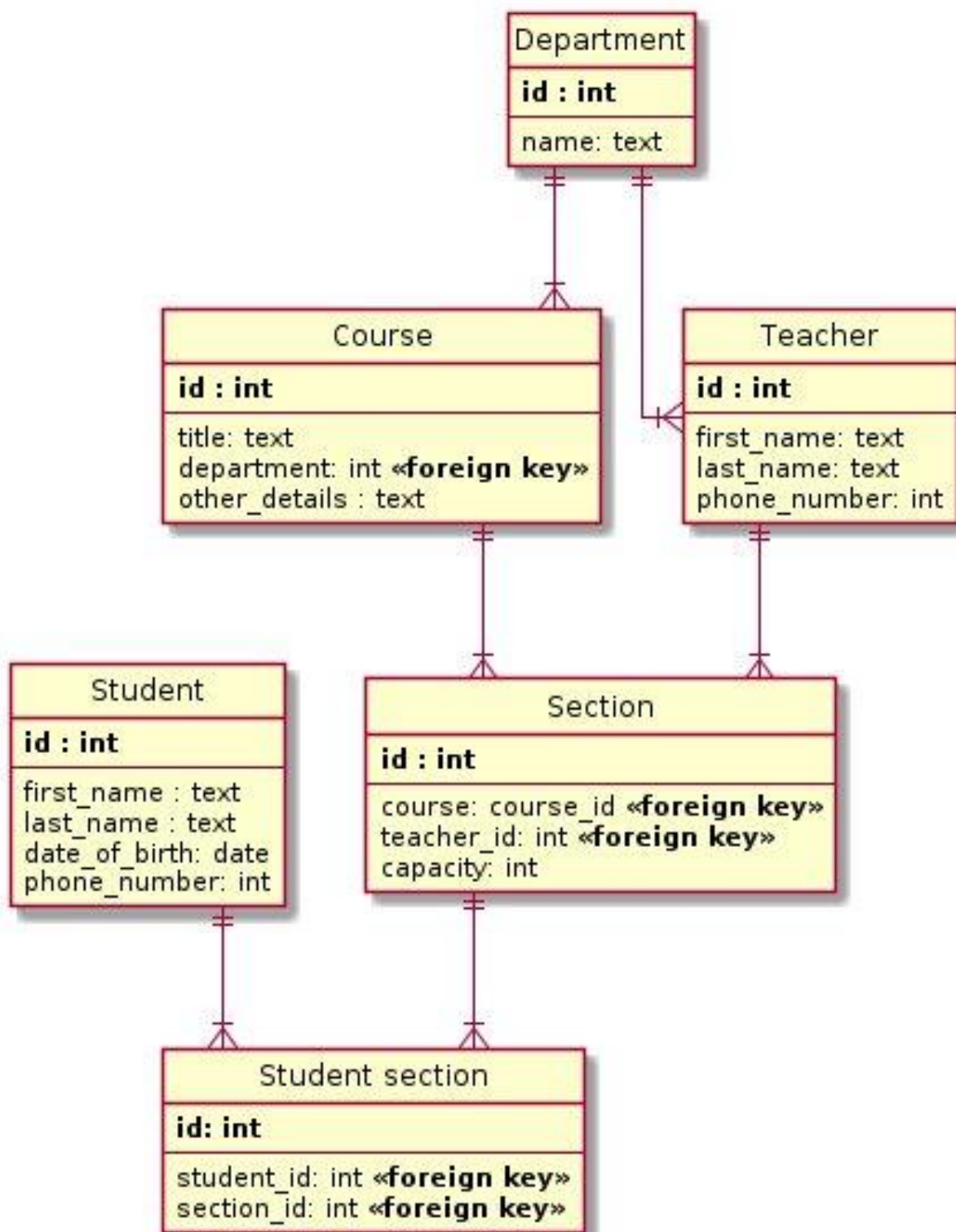
## Алгоритм роботи чат-боту





## ДОДАТОК Г

## Зв'язки в базі даних



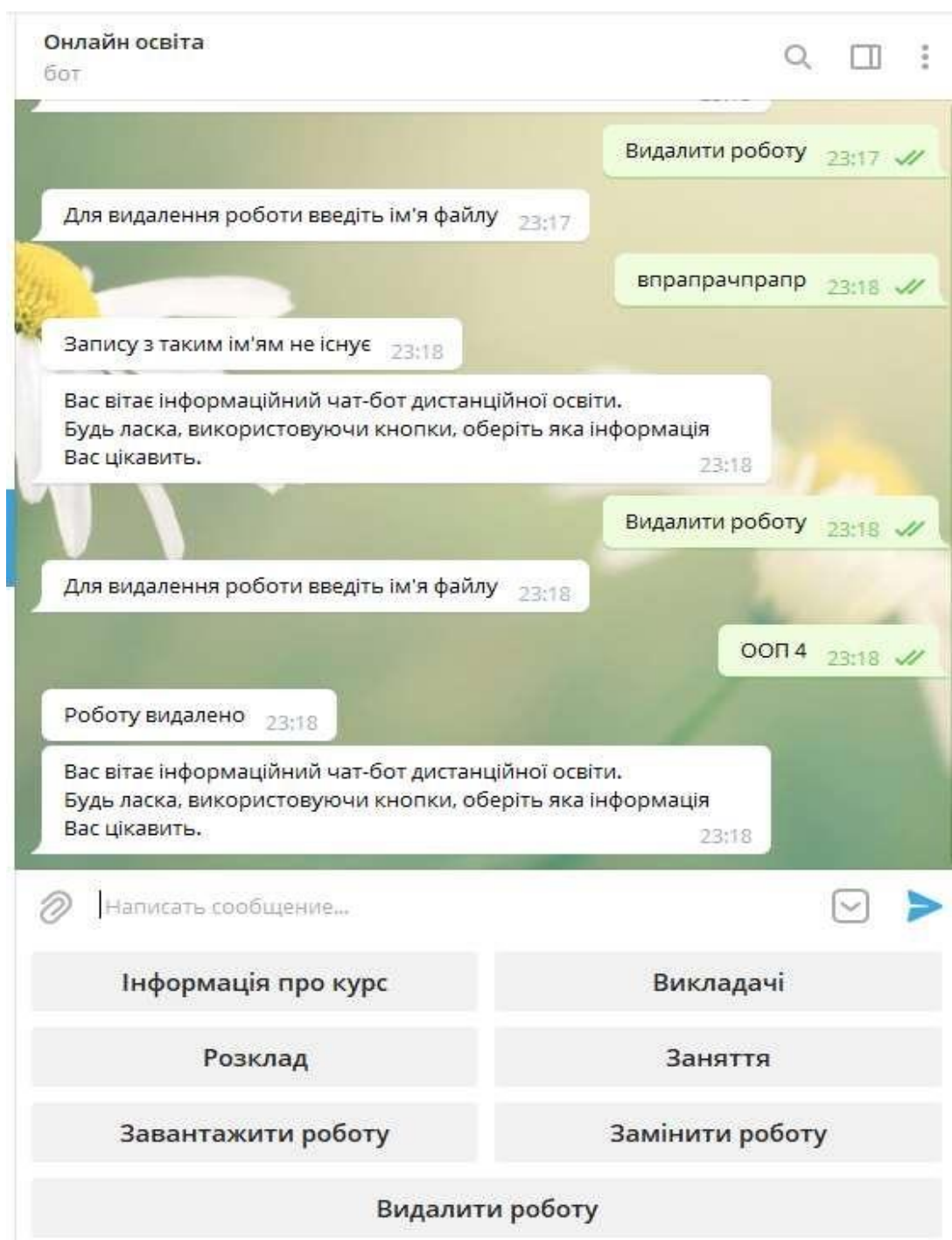
## ДОДАТОК Д

## Приклад роботи з чат ботом



## ДОДАТОК Е

## Видалення роботи з чат бота



## ДОДАТОК Ж

### ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Технологія інтеграції чат-ботів у веб-застосунки

Тип роботи: \_\_\_\_\_ магістерська кваліфікаційна робота \_\_\_\_\_  
(БДР, МКР)

Підрозділ \_\_\_\_\_ кафедра обчислювальної техніки \_\_\_\_\_  
(кафедра, факультет)

Показники звіту подібності Unіcheck

Оригінальність \_\_\_\_\_ 86.4% \_\_\_\_\_ Схожість \_\_\_\_\_ 13.6% \_\_\_\_\_

Аналіз звіту подібності (відмітити потрібне):

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

Особа, відповідальна за перевірку \_\_\_\_\_ Захарченко С.М.  
(підпис) (прізвище, ініціали)

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

Автор роботи \_\_\_\_\_ Слободян А.О.  
(підпис) (прізвище, ініціали)

Керівник роботи \_\_\_\_\_ Захарченко С.М.  
(підпис) (прізвище, ініціали)