

Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

До захисту допущено
Кафедрою комп'ютерних систем та робототехніки
протокол № __ від __ грудня 2025р.

завідувач кафедри _____ Максим ХРУСЛОВ
(підпис)

«__» _____ 2025 р.

Кваліфікаційна робота
здобувача другого (магістерського) рівня вищої освіти

**«МОДЕЛЬ КОМП'ЮТЕРНОЇ СИСТЕМИ ПІДТРИМКИ
КЛІЄНТІВ НА ОСНОВІ ЧАТ-БОТУ»**

Спеціальність 123 – *Комп'ютерна інженерія.*
Освітня програма *Комп'ютерна інженерія*

Виконавець _____ Олексій НОВІКОВ
(підпис)

Науковий керівник _____ Вікторія СТРИЛЕЦЬ
(підпис)

Харків – 2025

АНОТАЦІЯ

Пояснювальна записка до магістерської кваліфікаційної роботи складається зі вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. Загальний обсяг роботи складає 89 сторінок, із яких 70 сторінок основної частини з 12 рисунками, 7 таблицями, 26 найменувань списку використаних джерел та трьома додатками.

Метою кваліфікаційної роботи є підвищення зручності та ефективності процесу прийняття рішень при виборі компонентів персонального комп'ютера шляхом використання Telegram чат-бота з методами NLP для врахування запитів користувача, наданих природною мовою.

Об'єкт дослідження – процес підтримки прийняття клієнтських рішень при виборі та конфігурації комп'ютерних комплектуючих.

Предмет дослідження – моделі, методи та програмні засоби застосування технологій NLP у телеграм чат-ботах для автоматизації вибору комп'ютерних комплектуючих.

Проблема, яка вирішується в кваліфікаційній роботі, полягає в тому, щоб, використовуючи сучасні технології обробки природної мови NLP та методи інтелектуального аналізу даних, спростити процес підбору комп'ютерних комплектуючих, підвищити точність рекомендацій і зменшити час, необхідний для вибору сумісних компонентів.

Область застосування – сфера підбору та конфігурації комп'ютерних комплектуючих.

Ключові слова: *Telegram Bot, NLP, Python, Fuzzy Matching, NLTK, Stanza, python-telegram-bot, конфігуратор ПК, обробка природної мови, рекомендаційна система, користувацькі запити, інтеграція, база даних, автоматизація, підбір комплектуючих.*

ABSTRACT

The explanatory note to the master's thesis consists of an introduction, three sections, conclusions, a list of sources used and three appendices. The total volume of the work is 89 pages, of which 70 pages are the main part with 12 figures, 7 tables, 26 items of the list of sources used and three appendices.

The purpose of the qualification research is to enhance the convenience and efficiency of the decision-making process in selecting personal computer components by using a Telegram chatbot that employs NLP methods to interpret user requests expressed in natural language.

The object of research is the process of supporting customer decision-making in the selection and configuration of computer components.

The subject of research is the models, methods, and software tools for applying NLP technologies in Telegram chatbots to automate the selection of computer components.

The problem solved in the qualification work is to simplify the process of selecting computer components, improve the accuracy of recommendations, and reduce the time required to choose compatible parts by using modern natural language processing NLP technologies and intelligent data analysis methods.

Scope is the field of selection and configuration of computer components.

Keywords: *Telegram Bot, NLP, Python, Fuzzy Matching, NLTK, Stanza, python-telegram-bot, PC configurator, natural language processing, recommendation system, user queries, integration, database, automation, component selection.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ОСНОВ РОЗРОБКИ TELEGRAM ЧАТ-БОТІВ .	9
1.1 Поняття чат-ботів та їх роль у сучасних інформаційних системах	9
1.2 Telegram як платформа для розробки ботів.....	13
1.3 Технології та інструменти розробки Telegram чат-ботів	18
1.4 API та зовнішні сервіси	24
1.5 Ключові параметри та класифікація апаратного забезпечення персонального комп'ютера	26
1.6 Використання NLP у чат-ботах для обробки запитів користувачів	31
Висновки до розділу 1	35
РОЗДІЛ 2. АНАЛІЗ ТА ПРОЄКТУВАННЯ СИСТЕМИ КОНФІГУРАТОРА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА	36
2.1 Концепція конфігуратора ПК та його призначення	36
2.2 Аналіз існуючих аналогів ботів-конфігураторів	43
2.3 Опис функцій та можливостей моделі Telegram-бота	49
2.4 Модель системи Telegram чат-бота.....	53
Висновки до розділу 2	55
РОЗДІЛ 3. РЕАЛІЗАЦІЯ TELEGRAM ЧАТ-БОТА КОНФІГУРАТОРА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА	56
3.1 Інструменти, використані при реалізації чат-бота	56
3.2 Архітектура та структура бота.....	60
3.3 Демонстрація функціоналу роботи бота.....	63
3.4 Оцінка ефективності роботи бота	68
3.5 Актуальність, новизна та перспективи розвитку	69
Висновки до розділу 3	72

ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	77
Додаток А.....	77
Додаток Б.....	79
Додаток В.....	83

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

NLP – Natural Language Processing

API – Application Programming Interface

JSON – JavaScript Object Notation

CSV – Comma-Separated Values

NLTK – Natural Language Toolkit

RAM – Random Access Memory

GPU – Graphics Processing Unit

CPU – Central Processing Unit

SSD – Solid State Drive

HDD – Hard Disk Drive

ПК – персональний комп'ютер

БД – база даних

ВСТУП

У зв'язку з активним розвитком цифрових технологій та зростанням різноманіття комп'ютерних комплектуючих постає потреба у створенні інтелектуальних систем, що спрощують процес вибору та конфігурації персональних комп'ютерів. Застосування чат-ботів із технологіями обробки природної мови дозволяє автоматизувати цей процес, підвищуючи зручність і точність підбору компонентів.

Актуальність роботи. На сьогодні Telegram чат-боти стрімко набирають популярності як ефективний інструмент автоматизації сервісів і взаємодії з користувачами. Їх використання дозволяє значно скоротити час обробки запитів, знизити навантаження на персонал та забезпечити цілодобову доступність послуг.

Завдяки впровадженню технологій обробки природної мови NLP чат-боти здатні розуміти запити користувачів, сформульованих природною мовою, аналізувати контекст і надавати максимально релевантні відповіді.

У сфері підбору комп'ютерних комплектуючих такі рішення допомагають спростити комунікацію між користувачем і системою, автоматизувати процес пошуку та рекомендації компонентів, враховуючи індивідуальні потреби й технічні параметри. Це сприяє підвищенню зручності, швидкості та точності вибору необхідних деталей для персонального комп'ютера.

Метою дослідження є підвищення зручності та ефективності процесу прийняття рішень при виборі компонентів персонального комп'ютера шляхом використання Telegram чат-бота з методами NLP для врахування запитів користувача, наданих природною мовою.

Об'єкт дослідження – це процес підтримки прийняття клієнтських рішень при виборі та конфігурації комп'ютерних комплектуючих.

Методи дослідження: методи обробки природної мови NLP, методи побудови діалогових систем, підходи до організації компонентів програмного забезпечення, клієнт-серверна архітектура, fuzzy matching.

Предмет дослідження – моделі, методи та програмні засоби застосування технологій NLP у телеграм чат-ботах для автоматизації вибору комп'ютерних комплектуючих.

Завдання дослідження

1. Провести аналіз принципів побудови Telegram чат-ботів, NLP-технологій та можливостей їх використання у системах підтримки клієнтів.
2. Розробити архітектуру чат-бота для конфігурації персонального комп'ютера.
3. Реалізувати модуль обробки природної мови для інтерпретації користувацьких запитів.
4. Створити алгоритми підбору комп'ютерних комплектуючих на основі введених даних.
5. Провести тестування, оцінку точності роботи та ефективності запропонованої моделі підтримки клієнтів при конфігурації персонального комп'ютера.

РОЗДІЛ 1.

ДОСЛІДЖЕННЯ ОСНОВ РОЗРОБКИ TELEGRAM ЧАТ-БОТІВ

1.1 Поняття чат-ботів та їх роль у сучасних інформаційних системах

Чат-бот[1] (від англ. *chatbot*) – це програмний застосунок або інформаційна система, яка імітує діалог з користувачем за допомогою текстових чи голосових повідомлень. Основне призначення чат-бота полягає в автоматизації комунікації, наданні інформаційних послуг або виконанні типових завдань без безпосередньої участі людини-оператора.

Чат-бот може розглядатися як інструмент взаємодії з інформаційними системами, що замінює традиційні графічні інтерфейси, або як інтелектуальний агент, який використовує методи штучного інтелекту та машинного навчання для адаптації до поведінки користувача. Попри різні підходи, спільними характеристиками чат-ботів є:

- автоматизованість процесу спілкування;
- діалогова форма взаємодії;
- орієнтація на вирішення прикладних завдань користувача.

Важливо зазначити, що чат-боти[1] належать до класу «розмовних технологій» (*conversational technologies*), які забезпечують людино-машинний інтерфейс нового типу. На відміну від класичних веб-додатків або мобільних застосунків, чат-боти надають доступ до сервісів у більш природній для користувача формі – через звичні канали комунікації, такі як месенджери чи соціальні мережі. Це робить їх одним із найбільш перспективних інструментів у сучасних інформаційних системах.

Чат-боти сьогодні використовуються у різних сферах діяльності, від бізнесу та освіти до державних сервісів і технічної підтримки, тому розуміння їхніх типів дозволяє обирати оптимальні рішення для конкретних завдань та сценаріїв взаємодії з користувачем. Вибір відповідного типу бота визначає ефективність його роботи, здатність до адаптації та ступінь автоматизації процесів.

Класифікація чат-ботів[1] здійснюється на основі принципів їх роботи та рівня інтелектуальної складової. Розрізняють кілька основних типів, кожен із яких має свої особливості та сфери застосування.

- *Боти на основі правил (rule-based)*. Ці системи функціонують на основі заздалегідь визначених сценаріїв та алгоритмів обробки запитів користувача. Вони виконують обмежений набір дій та надають відповіді відповідно до встановлених правил. Такий підхід забезпечує стабільність роботи системи, проте не дозволяє адаптуватися до нових або нестандартних формулювань користувача.

- *Інтелектуальні боти (AI-driven)*. Використовують методи обробки природної мови NLP[23] та алгоритми машинного навчання для аналізу і інтерпретації запитів. Ці системи здатні самостійно навчатися на основі попередніх взаємодій, формувати динамічні відповіді та підвищувати якість обслуговування користувачів.

- *Гібридні боти*. Поєднують переваги обох підходів: стандартні сценарії обробляються за правилами, тоді як складні або нестандартні запити обробляються за допомогою AI. Це дозволяє забезпечити баланс між стабільністю та гнучкістю системи.

Розуміння класифікації чат-ботів є не лише теоретичною задачею, але й практичним інструментом для проектування ефективних систем взаємодії. Знання типів ботів допомагає розробникам передбачити можливості автоматизації, оцінити рівень складності реалізації та підібрати оптимальні технологічні засоби для конкретного завдання. Це, у свою чергу, підвищує якість взаємодії з користувачами, скорочує час обробки запитів і дозволяє ефективно інтегрувати чат-боти в існуючі інформаційні системи.

Чат-боти знайшли широке застосування у різних сферах діяльності завдяки своїй здатності автоматизувати взаємодію з користувачами, швидко обробляти запити та надавати необхідну інформацію. Основні **сфери застосування**:

- *Бізнес.* У комерційній діяльності чат-боти використовуються для обслуговування клієнтів, консультацій щодо товарів і послуг, обробки замовлень, проведення маркетингових кампаній та збору зворотного зв'язку. Вони дозволяють скоротити витрати на кол-центри та підвищити швидкість реакції на запити клієнтів.

- *Освіта.* У навчальному процесі боти застосовуються для автоматизованого консультування студентів, нагадування про дедлайни та навчальні матеріали, організації тестування та практичних завдань. Інтелектуальні боти здатні адаптувати навчальні рекомендації під рівень знань та інтереси користувача.

- *Державні послуги.* Чат-боти використовуються у державних сервісах для надання консультацій громадянам, інформування про документи, запису на прийом до установ, подачі заяв та звернень. Це підвищує ефективність обслуговування населення та скорочує час взаємодії з адміністративними структурами.

- *E-commerce.* У сфері електронної комерції чат-боти допомагають користувачам обирати товари, порівнювати характеристики та ціни, оформлювати замовлення та відстежувати доставку. Вони підвищують зручність покупок та сприяють збільшенню конверсії.

- *Технічна підтримка.* Боти здатні відповідати на стандартні запити користувачів щодо налаштування програмного забезпечення, усунення неполадок та вирішення типових проблем. Це зменшує навантаження на службу підтримки та скорочує час обробки звернень.

Завдяки широкому спектру можливостей чат-боти стають універсальним інструментом, що підвищує ефективність взаємодії користувача з інформаційними системами. Їхнє використання дозволяє забезпечити оперативний доступ до необхідної інформації, автоматизувати рутинні процеси та зменшити витрати на обслуговування. Розуміння сфер застосування є важливим етапом при проектуванні та впровадженні чат-ботів,

оскільки дозволяє визначити основні функції системи та оптимальні підходи до інтеграції в конкретну сферу діяльності.

Переваги чат-ботів у порівнянні з класичними веб-версіями та мобільними застосунками. Чат-боти дедалі частіше розглядаються як ефективна альтернатива традиційним веб-сайтам та мобільним застосункам. Це зумовлено їхньою здатністю забезпечувати більш швидку, персоналізовану та зручну взаємодію з користувачем.

У таблиці 1.1 описані переваги чат-ботів у порівнянні з веб-версіями та мобільними застосунками.

Таблиця 1.1

Порівняння чат-ботів з веб-версіями та мобільними додатками

Перевага	Чат-боти	Веб-/мобільні застосунки
Доступність	Доступні у будь-якому месенджері, не потребують встановлення	Потрібно завантажувати застосунок або відкривати сайт
Простота використання	Інтуїтивна взаємодія через діалог	Необхідність вивчати інтерфейс і меню
Персоналізація	Можуть враховувати історію діалогу та налаштовувати відповіді	Обмежена персоналізація, здебільшого через акаунт
Швидкість отримання інформації	Миттєві відповіді в діалоговій формі	Пошук потрібної функції чи розділу займає більше часу
Витрати на розробку та підтримку	Нижчі, оскільки часто використовуються готові платформи	Вищі, потребують регулярних оновлень і підтримки
Інтеграція з іншими сервісами	Легке підключення до CRM, баз даних, API	Складніша інтеграція, особливо у мобільних застосунках
Цілодобова доступність	Працюють безперервно, без участі людини	Залежить від серверів та служби підтримки

Загалом, чат-боти поєднують у собі функціональність, характерну для веб та мобільних застосунків, проте надають її у більш зручному та

економічно ефективному форматі. Їхня популярність зумовлена саме здатністю інтегруватися у повсякденну комунікацію користувачів, роблячи процес взаємодії з інформаційними системами більш природним.

1.2 Telegram як платформа для розробки ботів

Telegram сьогодні виступає однією з провідних платформ для створення та розгортання чат-ботів, що пояснюється поєднанням відкритої архітектури, розвиненої інфраструктури та широкої користувацької бази. Завдяки доступному та детально документованому Telegram API, розробники отримують можливість швидко інтегрувати різноманітні сценарії взаємодії з користувачами: від надання інформаційних послуг і автоматизації бізнес-процесів до створення повноцінних сервісних систем. Платформа підтримує обмін як текстовими повідомленнями, так і мультимедійним контентом, інтерактивними кнопками, формами та іншими інструментами, що значно розширює функціональні можливості ботів. У поєднанні з високим рівнем безпеки та наскрізним шифруванням, Telegram стає привабливим середовищем для побудови сучасних діалогових систем різного рівня складності.

1.2.1 Telegram Bot API: можливості, структура, принципи роботи

Telegram Bot API[3] – це набір інструментів і протоколів, які забезпечують програмну взаємодію між стороннім застосунком (ботом) та інфраструктурою Telegram. API реалізує стандартний механізм обміну даними у форматі JSON через HTTPS, що дозволяє створювати автоматизовані сервіси, здатні приймати та обробляти повідомлення користувачів, а також надсилати відповіді у текстовій чи мультимедійній формі.

Можливості Telegram Bot API охоплюють широкий спектр функцій, що дозволяють будувати багатофункціональні системи взаємодії з користувачами. Серед основних можливостей варто виділити:

- надсилання та отримання текстових повідомлень у режимі реального часу;

- підтримку мультимедійних даних (зображення, відео, аудіо, документи, геолокація);
- створення інтерактивних елементів інтерфейсу (inline-кнопки, меню, опитування);
- інтеграцію з зовнішніми сервісами через вебхуки та API сторонніх систем;
- підтримку багатомовності та персоналізації сценаріїв взаємодії.

Структурно Telegram Bot API[3] ґрунтується на обміні даними у форматі JSON через протокол HTTPS. Кожен бот ідентифікується за унікальним токеном доступу, що видається через сервіс BotFather. Обмін інформацією може відбуватися двома способами:

- Polling – періодичне опитування серверів Telegram для отримання оновлень;
- Webhook – асинхронне отримання даних шляхом надсилання їх на визначену URL-адресу сервера розробника.

Принцип роботи Telegram Bot API[3] базується на чіткому механізмі обміну даними між користувачем, сервером Telegram та серверною частиною бота. Взаємодія розпочинається з того, що користувач надсилає повідомлення у чаті з ботом. Це повідомлення потрапляє на сервери Telegram, де воно перетворюється у стандартизований структурований об'єкт (у форматі JSON), який містить всю необхідну інформацію: текст запиту, дані про користувача, час відправлення, тип контенту (текст, фото, відео, файл тощо).

Далі цей об'єкт передається на серверну частину програми, яку розробник реалізує для обробки повідомлень. На цьому етапі бот аналізує отримані дані та виконує запрограмовану логіку: це може бути пошук інформації, здійснення обчислень, звернення до бази даних, інтеграція із зовнішніми сервісами або генерація індивідуальних рекомендацій. Результат обробки повертається назад у вигляді відповіді, яка форматується у зручний

для сприйняття користувачем вигляд – текстове повідомлення, медіафайл, кнопки для вибору або інтерактивне меню.

Telegram Bot API виступає універсальним інструментом, що забезпечує стандартизовану комунікацію між користувачем і програмним забезпеченням, дозволяючи створювати гнучкі та масштабовані діалогові системи.

1.2.2 Основні методи API

Основні методи API[3, 4] забезпечують розробникам доступ до широкого спектра функцій, необхідних для створення повноцінного та інтерактивного чат-бота. Telegram Bot API побудований за принципами REST-архітектури та використовує прості HTTPS-запити, у відповідь на які сервер повертає дані у форматі JSON. Умовно всі методи можна поділити на кілька груп, що відображають їхнє призначення.

До найбільш уживаних належать:

- getUpdates – отримання нових повідомлень і подій, які надійшли до бота. Використовується у випадках, коли застосовується механізм long polling для взаємодії із сервером Telegram;

- sendMessage – надсилання текстових повідомлень користувачам або групам. Є одним із базових методів, який забезпечує зворотний зв'язок між ботом і користувачем;

- editMessageText та deleteMessage – методи, що дозволяють редагувати вже надіслані повідомлення або видаляти їх у разі потреби. Це забезпечує динамічність та актуальність інформації, що подається користувачеві;

- sendPhoto, sendVideo, sendDocument, sendAudio – набір методів для роботи з мультимедійними файлами. Завдяки цьому боти можуть надсилати зображення, відео, аудіо та документи, розширюючи спектр можливостей взаємодії;

- answerCallbackQuery – метод, який відповідає за обробку дій користувача з інтерактивними елементами, наприклад, кнопками inline-keyboard. Це дозволяє створювати багатокрокові сценарії та інтерактивні меню;

- getUserProfilePhotos – отримання фотографій профілю користувача, що може застосовуватися для персоналізації роботи бота;

- restrictChatMember, promoteChatMember, banChatMember – методи адміністрування, які дозволяють управляти доступом користувачів у групових чатах.

Окрім базових можливостей надсилання та обробки повідомлень, API підтримує розширені функції, зокрема створення опитувань (sendPoll), роботу з геолокацією (sendLocation, sendVenue), а також інтеграцію з платіжними системами (sendInvoice, answerShippingQuery, answerPreCheckoutQuery).

Отже, Telegram Bot API надає універсальний набір інструментів, що охоплює як прості завдання комунікації, так і складні сценарії інтеграції з зовнішніми системами, що робить платформу однією з найбільш гнучких для розробки чат-ботів.

1.2.3 Безпека та автентифікація

Безпека та автентифікація у Telegram Bot API[3] є ключовими аспектами, що забезпечують надійність і стабільність роботи чат-ботів. Кожен бот отримує унікальний токен доступу, який використовується для автентифікації запитів. Взаємодія з серверами Telegram відбувається через захищений протокол HTTPS, що гарантує захист від перехоплення даних. Крім того, Telegram надає інструменти адміністрування, що дозволяють контролювати дії користувачів у групах, обмежувати доступ або блокувати небажаних учасників. Важливим завданням розробника є також дотримання правил конфіденційності та захисту персональних даних користувачів.

Для створення бота використовується сервіс BotFather, який видає унікальний токен автентифікації. Цей токен є своєрідним ключем, що відкриває доступ до управління ботом, тому його компрометація може призвести до втрати контролю над системою. З метою захисту розробникам рекомендується зберігати токени у захищених середовищах, наприклад, у змінних середовища або у спеціалізованих менеджерах секретів.

Ще одним важливим аспектом є використання webhook-механізму. Він дозволяє передавати дані від серверів Telegram безпосередньо на сервер розробника у режимі реального часу. Для цього необхідно використовувати захищене з'єднання з валідним SSL-сертифікатом. Це унеможливорює перехоплення чи підміну даних під час передачі.

Telegram також дотримується принципу мінімізації доступу: API не надає розробникам повних персональних даних користувачів, обмежуючись лише основною інформацією (ID, ім'я, username, фото профілю). Це дозволяє знизити ризики витоку конфіденційних відомостей.

Для запобігання зловживанням Telegram встановлює обмеження на частоту викликів методів API (rate limits). Це захищає як сервери месенджера, так і користувачів від масових атак чи надмірного спаму. Додатково розробники можуть реалізувати власні механізми захисту, наприклад, обмеження кількості запитів від одного користувача або інтеграцію капчі.

Узагальнюючи, питання безпеки у Telegram Bot API реалізуються на кількох ключових рівнях: автентифікації через токени, захищеної передачі даних, обмеження доступу, захисту персональної інформації та контролю над активністю користувачів. Дотримання цих вимог забезпечує створення надійних і безпечних чат-ботів, здатних ефективно функціонувати в різних практичних сценаріях.

1.2.4 Переваги використання Telegram над іншими месенджерами

Telegram посідає особливе місце серед сучасних месенджерів завдяки поєднанню зручності, швидкодії та відкритості до інтеграції з зовнішніми системами. На відміну від багатьох конкурентів, цей сервіс орієнтований не лише на комунікацію між користувачами, але й на створення інфраструктури для автоматизованих інструментів, зокрема чат-ботів.

У таблиці 1.2 наведено порівняння основних характеристик, що визначають доцільність використання цієї платформи для розробки інтелектуальних систем взаємодії з користувачами.

Таблиця 1.2

Порівняння Telegram чат-ботів з іншими месенджерами

Критерій	Telegram	Інші месенджери
Доступ до API для ботів	Відкрите, безкоштовне, добре задокументоване API	Обмежений доступ, часто лише для бізнес-акаунтів або з комерційними умовами
Мультимедійні можливості	Текст, фото, відео, документи, голосові, відеоповідомлення, стікери	Аналогічні можливості, але часто з обмеженнями за розміром чи форматом файлів
Інтерактивність	Inline-кнопки, callback-запити, створювані клавіатури, багатокрокові сценарії	Мінімальні інтерактивні елементи або їхня реалізація значно складніша
Мультиплатформність	Доступний на Windows, Linux, macOS, Android, iOS та Web	Зазвичай обмежені мобільними платформами (Android, iOS), десктопні версії менш зручні
Безпека та конфіденційність	Двофакторна автентифікація, секретні чати, захищені протоколи	Є end-to-end шифрування, але часто відсутній прозорий контроль над персональними даними
Масштабованість і швидкодія	Хмарна архітектура, висока стабільність навіть при великих навантаженнях	Стабільність висока, але обмеження на інтеграцію з ботами або залежність від бізнес-моделі
Аудиторія	Швидко зростаюча міжнародна спільнота	Деякі мають більшу аудиторію, але з більш жорсткими обмеженнями

1.3 Технології та інструменти розробки Telegram чат-ботів

Створення сучасних чат-ботів потребує використання спеціалізованих інструментів і бібліотек, які забезпечують ефективну інтеграцію з месенджерами, обробку повідомлень користувачів та автоматизацію виконання завдань. Чат-боти виконують різноманітні функції, починаючи від надання інформації та консультацій, закінчуючи обробкою замовлень, бронюванням послуг або адмініструванням групових чатів.

Вибір інструментів для розробки визначається низкою факторів, серед яких – обрана мова програмування, рівень підтримки та документації, вимоги

до продуктивності та масштабованості, можливості підключення до баз даних і зовнішніх API, а також специфічні потреби проєкту. Крім того, важливим аспектом є простота інтеграції з різними платформами, на яких буде працювати бот, а також наявність готових бібліотек і фреймворків, що дозволяють значно прискорити процес розробки.

1.3.1 Мови програмування для розробки ботів

Однією з найпопулярніших мов програмування для створення чат-ботів є *Python*[5], що пояснюється простотою та зрозумілістю синтаксису, широкими можливостями для інтеграції з базами даних, веб-сервісами та API, а також багатою екосистемою бібліотек і фреймворків, які дозволяють швидко реалізовувати функціональні рішення. Для взаємодії з Telegram Bot API існують популярні бібліотеки, такі як *python-telegram-bot*, *aiogram* та *telebot*, що забезпечують легкий доступ до основних функцій API, обробку повідомлень, керування чатами та мультимедійними елементами. *Python*[5] підтримує асинхронне програмування, що дозволяє створювати високопродуктивні боти, здатні одночасно обслуговувати велику кількість користувачів без втрати швидкодії. Крім того, велика спільнота розробників, детальна документація та регулярне оновлення бібліотек роблять *Python* надійним і зручним інструментом для комерційних, освітніх та дослідницьких проєктів, забезпечуючи гнучкість, масштабованість та швидку інтеграцію з зовнішніми сервісами.

Хоча *Python*[5] є найпопулярнішою мовою для розробки Telegram чат-ботів, він не єдиний варіант для створення подібних застосунків. Розробники також активно використовують інші мови програмування, такі як *JavaScript*, *Java*, *PHP* та *Go*, кожна з яких має власні переваги та специфічні інструменти для інтеграції з Telegram Bot API, що дозволяє вибирати мову залежно від вимог проєкту, досвіду команди та цілей застосунку.

JavaScript[6] (*Node.js*)[7] застосовується для створення ботів, орієнтованих на високонавантажені системи та інтерактивні веб-сервіси. Завдяки асинхронній моделі виконання *Node.js*[7] забезпечує ефективну

обробку одночасних запитів користувачів. Для роботи з Telegram існують популярні бібліотеки, такі як `node-telegram-bot-api` та `telegraf`, що дозволяють реалізовувати функціональні та гнучкі чат-боти.

Java[8] часто використовується у корпоративних проєктах, де важлива стабільність, масштабованість та наявність інтеграції з корпоративними системами. Для розробки ботів у *Java*[8] застосовується бібліотека `TelegramBots`, яка дозволяє обробляти повідомлення, створювати команди та інтегрувати сторонні сервіси.

Go[9] (*Golang*) – мова програмування, що характеризується високою продуктивністю, низькою затримкою та ефективністю обробки запитів. Бібліотека `tgbotapi` забезпечує розробку Telegram-ботів із мінімальними накладними витратами, що робить *Go* придатною для систем із високим навантаженням та вимогами до швидкості реакції.

PHP[10] широко використовується у веб-розробці та підходить для створення ботів, інтегрованих у веб-сайти та сервіси. Бібліотека `php-telegram-bot` дозволяє розробляти функціональні боти, використовуючи знайомі веб-технології та наявну інфраструктуру.

Вибір мови програмування визначається вимогами проєкту, необхідністю інтеграції з існуючими сервісами та доступністю спеціалізованих бібліотек, що спрощують розробку, підтримку та масштабування чат-ботів. У контексті створення Telegram-ботів для конфігурації персональних комп'ютерів перевагу зазвичай надають *Python* через його універсальність, простоту та широкий набір інструментів для обробки запитів користувачів.

1.3.2 Основні бібліотеки для створення чат-ботів

Для розробки Telegram чат-ботів на *Python* використовуються спеціалізовані бібліотеки, що забезпечують зручну інтеграцію з `Telegram Bot API` та спрощують процес обробки повідомлень користувачів. Однією з найпопулярніших бібліотек є `python-telegram-bot`[11], яка пропонує повний набір функцій для створення ботів, включаючи роботу з повідомленнями,

клавіатурами, обробку команд та подій. Бібліотека підтримує синхронний та асинхронний режим роботи, має добре документовані API і активну спільноту розробників, що дозволяє швидко вирішувати проблеми та інтегрувати нові функції.

Ще однією сучасною бібліотекою є *aiogram*[12], яка базується на асинхронному програмуванні з використанням *asyncio*. Вона забезпечує високу продуктивність і масштабованість ботів, дозволяє обробляти одночасно велику кількість повідомлень та подій, що робить її особливо актуальною для створення складних та високонавантажених чат-ботів. *Aiogram* також підтримує сучасні можливості Telegram Bot API та інтегрується з популярними веб-фреймворками.

Для початківців або проєктів з меншою складністю використовується бібліотека *telebot*[13]. Вона характеризується простим і зрозумілим синтаксисом, що дозволяє швидко створювати базові боти та навчальні проєкти. *Telebot*[13] забезпечує легку обробку текстових повідомлень, клавіатур і команд, проте не має таких широких можливостей для асинхронності та масштабування, як *aiogram*.

1.3.3 Інструменти та методи для реалізації NLP

Ефективність чат-бота значною мірою залежить від його здатності правильно інтерпретувати користувацькі запити, які здебільшого формуються у вигляді природної мови. Для цього застосовуються інструменти та бібліотеки, що надають готові алгоритми для токенизації, морфологічного аналізу, визначення сутностей, побудови векторних представлень слів і роботи з контекстом. Використання таких засобів дозволяє підвищити точність відповідей бота та зробити його більш «розумним» і гнучким у взаємодії з користувачами.

NLTK[14] (Natural Language Toolkit) – одна з найстаріших та найпопулярніших бібліотек для обробки природної мови у Python. Вона містить велику кількість інструментів і мовних корпусів, які дають змогу виконувати токенизацію, стемінг, лематизацію, визначення частин мови,

побудову частотних словників, роботу з граматиками тощо. NLTK[14] часто використовується у навчальних цілях та для створення прототипів, оскільки має просту структуру та багату документацію. Проте у великих комерційних проєктах її швидкодія поступається більш сучасним бібліотекам.

spaCy[15] – високопродуктивна бібліотека, розроблена спеціально для реальних застосунків у сфері NLP. Вона підтримує сучасні алгоритми обробки текстів, включаючи виділення іменованих сутностей (NER), визначення частин мови, роботу з синтаксичними залежностями. *spaCy*[15] відзначається оптимізацією швидкості роботи, що робить її придатною для масштабних проєктів, де потрібно швидко обробляти великі обсяги текстових даних. Також вона має готові моделі для багатьох мов, включно з українською та російською, і легко інтегрується з бібліотекою Transformers для використання передових нейронних моделей.

Transformers[16] (Hugging Face) – потужна бібліотека, яка надає доступ до передових моделей на основі глибинного навчання, таких як GPT, BERT, RoBERTa, T5 та ін. Вона дозволяє розв’язувати широкий спектр завдань: від класифікації тексту та аналізу тональності до генерації відповідей у діалогах та автоматичного перекладу. Однією з головних переваг є наявність великої кількості попередньо навчених моделей, які можна одразу застосовувати або донавчати під конкретні задачі. Завдяки цьому *Transformers*[16] широко використовуються в інтелектуальних чат-ботах, які мають працювати з контекстом і надавати природні відповіді.

FuzzyWuzzy[17] та *rapidfuzz*[18] – бібліотеки для реалізації нечіткого пошуку та обробки рядків. Вони дозволяють порівнювати тексти за ступенем схожості, що особливо важливо для чат-ботів, які мають працювати з некоректними або з помилками запитами користувачів. Наприклад, якщо користувач введе "телеграм бот апи" замість "Telegram Bot API", бот зможе правильно зрозуміти запит і надати релевантну відповідь. *FuzzyWuzzy*[17] заснована на бібліотеці Levenshtein, проте *rapidfuzz*[18] є її сучасною та

оптимізованою альтернативою, яка працює значно швидше і підходить для високонавантажених застосунків.

1.3.4 СУБД для зберігання даних

Ефективне функціонування чат-ботів у більшості випадків потребує використання систем управління базами даних (СУБД), оскільки саме вони забезпечують збереження та організацію інформації про користувачів, їхні запити, історію взаємодій та довідкові дані. Використання СУБД дозволяє створювати масштабовані рішення, у яких дані можуть швидко оновлюватися, оброблятися та бути доступними для подальшої аналітики.

Серед найбільш поширених рішень у розробці чат-ботів можна виділити PostgreSQL, SQLite та MongoDB.

PostgreSQL[19] є потужною реляційною системою управління базами даних з відкритим кодом. Вона підтримує складні запити, транзакції, роботу з JSON, розширення та індексацію. PostgreSQL[19] часто обирають для чат-ботів, які потребують високої надійності, масштабованості та роботи з великим обсягом структурованих даних.

SQLite[20] – це легка вбудована реляційна СУБД, яка зберігає дані у вигляді одного файлу. Вона не вимагає окремого сервера і тому підходить для невеликих проєктів, прототипів або ботів, де немає потреби в обробці великих обсягів інформації. Простота використання та мінімальні вимоги до налаштування роблять SQLite[20] зручним інструментом для швидкого старту.

MongoDB[21] належить до нереляційних СУБД (NoSQL) і зберігає дані у форматі документів JSON. Це забезпечує гнучкість у роботі зі складними або неоднорідними структурами даних. MongoDB[21] добре підходить для чат-ботів, які працюють з динамічними користувацькими даними, що можуть мати різні поля та формат. Крім того, MongoDB[21] легко масштабувати у випадках зростання кількості користувачів і обсягів інформації.

У таблиці 1.3 продемонстровано порівняння основних різновидів СУБД.

Таблиця 1.3

Порівняння різних СУБД

Критерій	SQLite	PostgreSQL	MongoDB
Тип СУБД	Реляційна, файлова	Реляційна, серверна	Документо-орієнтована(NoSQL)
Архітектура	Легка, без сервера, дані в одному файлі	Потужні, серверна, масштабована	Масштабована, документи у форматі BSON
Швидкість	Дуже швидка для невеликих обсягів даних	Швидка, оптимізована для великих навантажень	Швидка при роботі з неструктурованими даними
Масштабованість	Обмежена(для невеликих проєктів)	Відмінна підтримка масштабування та реплікації	Дуже висока, зручно для великих систем
Використання	Локальні застосунки, мобільні чати та чат-боти	Великі системи, веб-застосунки, корпоративні БД	Великі дані, аналітика, гнучкі структури
Простота	Дуже проста у використанні	Складніша в налаштуванні	Гнучка, але потребує адаптації під модель

1.4 API та зовнішні сервіси

У процесі створення сучасних чат-ботів важливу роль відіграє інтеграція із зовнішніми сервісами та прикладними інтерфейсами програмування (API). Завдяки таким механізмам боти отримують доступ до актуальної інформації, розширюють функціональні можливості та забезпечують динамічну взаємодію з користувачем. Використання API дозволяє інтегрувати чат-бота з базами даних, прайс-агрегаторами, e-commerce платформами, системами пошуку чи навіть сервісами машинного навчання. Це перетворює чат-бота з простого інструмента для комунікації у багатофункціональний сервіс, здатний виконувати завдання різної складності.

1.4.1 Використання API для отримання даних з інших сервісів

Використання зовнішніх API є одним із ключових механізмів у розробці чат-ботів, оскільки саме через них забезпечується доступ до актуальної та

динамічної інформації. У випадку чат-ботів для підбору товарів інтеграція з API дозволяє отримувати відомості про їх наявність, вартість, характеристики та відгуки користувачів. Це робить роботу бота значно ефективнішою, адже він не обмежується статичною базою даних, а може звертатися до зовнішніх джерел у режимі реального часу.

Найчастіше для цього використовуються API прайс-агрегаторів та e-commerce платформ, які надають структуровані дані про товари у форматах JSON або XML. Наприклад, боти можуть взаємодіяти з відкритими API великих онлайн-магазинів, що дозволяє формувати актуальні списки товарів, порівнювати ціни від різних постачальників і надавати користувачу рекомендації з урахуванням бюджету. Крім того, існує можливість інтеграції з власними API організацій чи локальними базами даних, де зберігається інформація про доступність продукції, гарантійні умови та технічні параметри.

Таким чином, підключення до зовнішніх API перетворює чат-бота на універсальний інструмент, що поєднує функції консультанта, аналітика й інтерактивного каталогу, забезпечуючи користувачам швидкий доступ до актуальних відомостей.

1.4.2 Інтеграція з системами пошуку та фільтрації

Для реалізації пошукових систем можуть використовуватися як готові інструменти, так і власні рішення. Elasticsearch є одним із найпопулярніших фреймворків для високопродуктивного пошуку. Він підтримує індексування великих обсягів даних, багатофільтрові запити, сортування та агрегації, що дозволяє обробляти одночасні запити великої кількості користувачів і забезпечує масштабованість системи.

Власні реалізації пошуку та фільтрації зазвичай базуються на реляційних (PostgreSQL, SQLite) або документно-орієнтованих (MongoDB) базах даних. У цьому випадку розробник визначає структуру даних, логіку пошуку та правила застосування фільтрів, що дозволяє адаптувати систему під специфіку конкретного проєкту, контролювати оптимізацію запитів та

додавати додаткові функції, такі як персоналізація результатів чи історія запитів.

Для підвищення точності обробки запитів користувачів, особливо якщо вони можуть містити помилки або неточності, застосовується нечіткий пошук (fuzzy search). Бібліотеки RapidFuzz[18] та FuzzyWuzzy[17] забезпечують ефективні алгоритми порівняння рядків і можуть інтегруватися як із базами даних, так і з системами на кшталт Elasticsearch. Це дозволяє системі знаходити найбільш релевантні результати навіть при некоректно сформульованих запитах.

Системи пошуку та фільтрації можуть включати багаторівневі фільтри, що дозволяють обмежувати результати за різними критеріями, такими як категорії інформації, типи об'єктів, пріоритети, дата створення чи інші атрибути, які мають значення для конкретного застосування.

Інтеграція пошукової системи з чат-ботом зазвичай реалізується через API або пряме підключення до бази даних. Бот отримує запит від користувача, обробляє його, формує структурований запит до системи пошуку, отримує результати та передає їх у зручному форматі – текстові повідомлення, картки або інтерактивні меню.

Такий підхід дозволяє створювати інтуїтивні та динамічні системи, де користувач може уточнювати свій запит або додавати нові параметри фільтрації, а бот швидко адаптує результати, забезпечуючи зручну взаємодію та ефективне надання інформації.

1.5 Ключові параметри та класифікація апаратного забезпечення персонального комп'ютера

Сучасний персональний комп'ютер являє собою складну систему, яка складається з низки взаємопов'язаних апаратних компонентів. Кожен із них виконує специфічну функцію, але лише в комплексі вони забезпечують стабільну роботу, продуктивність та відповідність потребам користувача. Для створення ефективних інструментів автоматизованого підбору обладнання,

зокрема чат-ботів-конфігураторів, необхідно мати чітке розуміння класифікації, характеристик та принципів сумісності основних комплектуючих. У цьому підрозділі розглядаються ключові компоненти комп'ютера, їхні технічні параметри та фактори, які впливають на вибір користувачів.

1.5.1 Класифікація основних компонентів ПК

Класифікація основних компонентів персонального комп'ютера[22] ґрунтується на їх функціональному призначенні та ролі у забезпеченні роботи системи. Кожен з елементів виконує специфічні завдання, а разом вони формують єдине апаратне середовище, що дозволяє виконувати обчислення, обробляти інформацію та взаємодіяти з користувачем. До ключових компонентів належать:

- центральний процесор[22] (CPU) – головний елемент, що виконує арифметичні та логічні операції, координує роботу інших частин комп'ютера та визначає загальну продуктивність системи;
- материнська плата[22] – базова плата, що об'єднує всі компоненти в єдину систему, забезпечуючи їх сумісність та обмін даними;
- оперативна пам'ять[22] (RAM) – модулі, що зберігають дані та інструкції під час роботи системи, впливаючи на швидкість обчислень;
- відеокарта[22] (GPU) – спеціалізований пристрій для обробки графіки та візуалізації, який може бути інтегрованим у процесор або дискретним;
- накопичувачі[22] (HDD, SSD, NVMe) – пристрої для збереження даних, програм та операційної системи;
- блок живлення[22] (PSU) – забезпечує стабільне електроживлення всіх компонентів системи;
- корпуси та системи охолодження[22] – елементи, що відповідають за розміщення, захист і тепловий режим комплектуючих.

Така класифікація дозволяє систематизувати знання про апаратне забезпечення ПК та є основою для аналізу сумісності компонентів, їх продуктивності та доцільності використання у конкретних завданнях.

1.5.2 Базові технічні параметри комп'ютерних комплектуючих

Для оцінки та вибору комп'ютерних комплектуючих важливим є розуміння їхніх технічних характеристик, оскільки саме вони визначають рівень продуктивності, сумісність та енергоефективність системи. Кожен тип компонента має свій набір параметрів, які дозволяють порівнювати моделі між собою та підбирати оптимальне рішення під конкретні задачі. У таблиці наведені основні технічні характеристики, що використовуються для опису сучасних процесорів, материнських плат, оперативної пам'яті, відеокарт, накопичувачів та інших елементів персонального комп'ютера.

У таблиці 1.4 наведені основні технічні характеристики для базових комплектуючих комп'ютера.

Таблиця 1.4

Основні технічні характеристики комп'ютерних комплектуючих

Компонент	Ключові технічні характеристики
Процесор (CPU)	Тактова частота (GHz), кількість ядер і потоків, обсяг кеш-пам'яті (L1/L2/L3), TDP (Вт), підтримка інструкцій, сокет
Материнська плата	Чіпсет, форм-фактор (ATX, mATX, ITX), сокет процесора, кількість і тип слотів RAM, кількість слотів PCIe, підтримка інтерфейсів (SATA, M.2, USB, LAN), BIOS/UEFI
Оперативна пам'ять (RAM)	Тип (DDR3, DDR4, DDR5), обсяг (ГБ), частота (MHz), затримки (CL), кількість каналів (Single/Dual/Quad), напруга
Відеокарта (GPU)	Відеопам'ять (тип GDDR, обсяг ГБ), частота ядра (MHz/GHz), кількість CUDA/Stream-процесорів, шина пам'яті (bit), пропускна здатність (GB/s), TDP, підтримка API (DirectX, OpenGL, Vulkan)
Накопичувачі (HDD/SSD/NVMe)	Тип носія, обсяг пам'яті (ГБ/ТБ), інтерфейс

	(SATA, NVMe, PCIe), швидкість читання/запису (MB/s), ресурс запису (TBW), час доступу (ms)
Блок живлення (PSU)	Потужність (Вт), сертифікація ефективності (80 PLUS Bronze/Gold/Platinum/Titanium), коефіцієнт ККД (%), модульність кабелів, кількість ліній 12V
Корпус	Форм-фактор сумісних плат, підтримка довжини GPU та висоти кулера CPU, кількість відсіків під накопичувачі, кількість і тип вентиляторів, наявність RGB
Система охолодження	Тип (повітряне/рідинне), розмір радіатора (мм), кількість теплових трубок, швидкість обертання вентиляторів (RPM), рівень шуму (дБ), тепловідведення (Вт)

Таким чином, технічні характеристики є ключовим критерієм при виборі комп'ютерних комплектуючих. Розуміння основних параметрів дозволяє користувачам більш усвідомлено підходити до формування конфігурацій ПК, орієнтуючись на власні потреби – від офісних завдань до ресурсомістких обчислень чи геймінгу.

1.5.3 Принципи сумісності комплектуючих

Сумісність комп'ютерних комплектуючих є одним із ключових факторів при складанні оптимальної конфігурації ПК. Вона визначає, наскільки ефективно окремі компоненти можуть взаємодіяти між собою, забезпечуючи стабільну роботу системи та максимальну продуктивність. Основні принципи сумісності включають відповідність роз'ємів та сокетів, підтримку певних стандартів пам'яті та інтерфейсів, а також врахування енергоспоживання та теплових характеристик. Недотримання цих вимог може призвести до некоректної роботи системи, зниження продуктивності або навіть виходу компонентів з ладу.

При складанні комп'ютерної системи важливо враховувати сумісність окремих компонентів, оскільки навіть найпотужніші деталі можуть працювати некоректно або не працювати зовсім при невідповідності між собою.

Сумісність охоплює кілька аспектів, серед яких ключові: фізична сумісність, електричні параметри, протоколи обміну даними та теплові характеристики.

Для процесора (CPU) та материнської плати[22] основним критерієм є відповідність сокета та підтримуваного чіпсету. Різні покоління процесорів можуть мати різні сокети, що робить неможливим їхнє встановлення на плату з іншим роз'ємом. Крім того, чіпсет визначає, які функції та технології будуть доступні системі, включно з підтримкою кількості оперативної пам'яті, швидкістю шини та наявністю інтерфейсів для накопичувачів та периферії.

Сумісність оперативної пам'яті[22] (RAM) визначається типом модулів (DDR3, DDR4, DDR5), частотою та об'ємом, які підтримує материнська плата. Використання пам'яті, що не відповідає специфікації плати, може призвести до нестабільної роботи системи або неможливості її завантаження.

Відеокарти[22] (GPU) потребують відповідного роз'єму на материнській платі (зазвичай PCIe) та врахування фізичних габаритів для установки в корпусі. Крім того, важливим є забезпечення достатньої потужності блоком живлення та наявність необхідних конекторів для живлення.

Накопичувачі[22] (HDD, SSD, NVMe) сумісні з конкретними інтерфейсами материнської плати (SATA, PCIe) та корпусом. Використання накопичувачів із невідповідним інтерфейсом або форм-фактором обмежить їхню продуктивність або зробить неможливою установку.

Блок живлення[22] (PSU) повинен забезпечувати достатню потужність для всіх компонентів та відповідати стандартам енергоспоживання. Сертифікації 80 PLUS дозволяють оцінити ефективність і надійність енергопостачання.

Корпус та система охолодження[22] повинні враховувати розміри материнської плати, габарити відеокарти та процесора, а також забезпечувати оптимальний повітряний потік для підтримки температурного режиму компонентів.

Дотримання цих принципів сумісності дозволяє створити надійну, стабільну та продуктивну комп'ютерну систему, уникнути проблем при

встановленні та забезпечити ефективну експлуатацію протягом тривалого часу.

1.6 Використання NLP у чат-ботах для обробки запитів користувачів

Сучасні чат-боти використовують методи обробки природної мови NLP для ефективної взаємодії з користувачем. Завдяки NLP боти здатні аналізувати текстові повідомлення, розпізнавати наміри користувача, визначати ключові сутності та формувати відповідні відповіді. Такий підхід дозволяє значно підвищити якість комунікації, забезпечити персоналізацію та скоротити час, необхідний користувачеві для отримання потрібної інформації.

Використання NLP включає інтеграцію різних методів та алгоритмів, таких як класифікація тексту, аналіз намірів, обробка контексту діалогу, а також застосування моделей машинного навчання та глибинного навчання. Завдяки цьому чат-боти можуть виконувати складні завдання: від простої відповіді на стандартні запити до генерації рекомендацій і адаптації до поведінки користувача.

1.6.1 Основи Natural Language Processing

Natural Language Processing[23] (NLP) – це міждисциплінарна область штучного інтелекту та комп'ютерних наук, яка займається автоматичною обробкою, аналізом та генерацією людської мови. Основною метою NLP є створення систем, здатних розуміти, інтерпретувати та реагувати на текстові або голосові дані користувачів у спосіб, близький до людського спілкування.

У контексті чат-ботів NLP[23] дозволяє перетворювати вхідні повідомлення користувачів на структуровану інформацію, яку система може обробити, і формувати відповідь у зручній для користувача формі. Цей процес включає низку базових завдань: розпізнавання намірів, виділення ключових сутностей, роботу з контекстом діалогу та класифікацію тексту. Завдяки застосуванню NLP сучасні чат-боти можуть працювати не лише за задалегідь

визначеними сценаріями, а й адаптуватися до різних формулювань запитів та стилю спілкування користувача.

Крім того, NLP[23] є основою для використання більш складних алгоритмів та моделей штучного інтелекту, таких як трансформери і embeddings, які дозволяють системам враховувати семантичний контекст та здійснювати точнішу інтерпретацію інформації.

1.6.2 Базові методи NLP

Для ефективної обробки природної мови чат-боти використовують низку базових методів, які дозволяють перетворювати текст у форму, придатну для комп'ютерного аналізу. Нижче буде детально розглянуто та описано основні методи, що застосовуються розробниками для обробки запитів користувачів.

Токенізація – базовий етап обробки тексту, що полягає у розбитті текстового потоку на окремі елементи, або токени. Ці токени можуть бути словами, фразами, реченнями або символами, залежно від завдання. Токенізація є необхідною для подальшого аналізу тексту, оскільки дозволяє алгоритмам працювати з чітко визначеними одиницями, а не з безперервним рядком символів. Крім того, правильне визначення токенів впливає на точність наступних етапів обробки, таких як стемінг, лематизація та розпізнавання сутностей.

Стемінг – метод зведення слів до їхньої основної форми (стему), який спрощує текст, видаляючи закінчення та граматичні модифікації. Стемінг дозволяє системі враховувати всі форми слова як один об'єкт при пошуку або класифікації, що значно підвищує ефективність обробки запитів. Існують різні алгоритми стемінгу, наприклад, алгоритм Портера або Ловінгстона, кожен з яких має свої переваги та обмеження щодо точності та швидкодії.

Лематизація – більш точний метод нормалізації слів, який враховує морфологічні властивості та граматичний контекст. На відміну від стемінгу, лематизація зберігає зміст слова і дозволяє працювати з його канонічною формою, що важливо для аналізу семантики та побудови складніших моделей

обробки мови. Лематизація часто використовує словники та правила морфології мови, що забезпечує високу точність перетворення.

Виділення частин мови (POS tagging) – процес визначення граматичної ролі слова в реченні, наприклад, чи є слово іменником, дієсловом, прикметником тощо. Цей метод допомагає чат-боту розуміти структуру речення, що важливо для побудови синтаксичного аналізу, інтерпретації намірів користувача та подальшого визначення ключових елементів запиту.

Розпізнавання сутностей (NER) – метод, який дозволяє ідентифікувати у тексті конкретні об'єкти, такі як імена людей, назви компаній, дати, локації чи інші сутності. Завдяки NER, чат-бот може виділяти критичну інформацію з запиту користувача, що дозволяє точніше обробляти запити та будувати більш релевантні відповіді.

Класифікація та аналіз намірів – метод, спрямований на визначення типу запиту користувача, наприклад, інформаційний запит, прохання виконати дію або уточнення. Цей етап дозволяє чат-боту адаптувати поведінку та формувати відповідь, відповідно до наміру користувача. Для цього використовують різні алгоритми машинного навчання та правила на основі ключових слів, які дозволяють системі коректно інтерпретувати запити.

Морфологічний аналіз – процес вивчення внутрішньої структури слова та його граматичних властивостей (рід, число, відмінок, час, ступінь порівняння тощо). Мета морфологічного аналізу – визначити канонічну форму слова та його граматичні характеристики, що дозволяє системі коректно співставляти різні форми одного слова під час пошуку, класифікації чи інтерпретації запиту. Морфологічний аналіз є базовим етапом для лематизації та побудови точних моделей розпізнавання намірів користувача.

Синонімічний аналіз – метод визначення слів або фраз, які мають близьке або еквівалентне значення. Він дозволяє чат-боту розпізнавати запити користувача, навіть якщо вони сформульовані різними словами. Використання синонімічного аналізу підвищує гнучкість системи, покращує пошук

інформації у базах знань і дозволяє правильно інтерпретувати різні варіанти запитів, що стосуються однієї концепції чи об'єкта.

Використання цих методів у поєднанні дозволяє чат-боту структурувати вхідний текст, ідентифікувати ключові елементи запиту та забезпечити точне формування відповіді. Ці базові техніки створюють основу для більш складних підходів, таких як *embeddings* і трансформери, що дозволяють враховувати контекст та семантичну значущість слів у діалозі.

1.6.3 Використання *embeddings* та трансформерів

Сьогодні все частіше застосовують методи глибинного навчання для підвищення точності обробки запитів користувачів та розуміння контексту діалогу. У цьому напрямі широко використовуються такі технології, як *embeddings* та трансформери (наприклад, моделі BERT та GPT), які дозволяють значно підвищити якість генерації відповідей і адаптивність системи. Інтеграція цих підходів сприяє розвитку більш інтелектуальних чат-ботів, здатних ефективно взаємодіяти з користувачем навіть у складних або багатокрокових сценаріях.

Векторні представлення слів, або *embeddings*, є фундаментальною технологією для сучасних систем обробки природної мови. Вони перетворюють текстові одиниці – слова, фрази або цілі документи – у числові вектори в багатовимірному просторі. Завдяки цьому алгоритми можуть оцінювати семантичну близькість слів і виразів, що дозволяє чат-ботам розпізнавати синоніми, пов'язані поняття та контекстні відтінки значень. *Embeddings* використовуються для пошуку найбільш релевантних результатів, класифікації запитів та рекомендацій, а також для підготовки вхідних даних для моделей глибинного навчання.

Трансформери є сучасною архітектурою нейронних мереж, яка дозволяє ефективно моделювати залежності між словами у тексті, враховуючи контекст на різних рівнях. Моделі на основі трансформерів, такі як BERT і GPT, здатні генерувати змістовні відповіді, аналізувати складні запити користувача та підтримувати багатокрокові діалоги. Наприклад, BERT використовується для

класифікації намірів користувача та розпізнавання ключових сутностей у тексті, а GPT може формулювати природні, узгоджені та контекстуально релевантні відповіді.

Використання embeddings і трансформерів у чат-ботах дозволяє значно підвищити точність розуміння запитів, зменшити кількість помилкових інтерпретацій та покращити загальну якість взаємодії з користувачем. Комбінація цих методів забезпечує здатність системи не лише відповідати на прості прямі запити, а й обробляти складні, багатозначні фрази, враховуючи контекст попередніх повідомлень. Крім того, embeddings і трансформери дозволяють інтегрувати зовнішні бази знань та оновлювані моделі мови, що підвищує адаптивність чат-бота до різноманітних сценаріїв використання.

Висновки до розділу 1

У першому розділі проведено теоретичний аналіз основних понять, технологій та інструментів, що використовуються під час розробки чат-ботів. Розглянуто принципи їх функціонування, особливості взаємодії з користувачем, а також роль платформи Telegram як ефективного середовища для створення автоматизованих рішень.

Окреслено програмні засоби, бібліотеки та системи управління базами даних, які забезпечують реалізацію різноманітних функцій чат-бота – від обробки запитів до зберігання та пошуку інформації. Також визначено важливість застосування методів обробки природної мови NLP, що підвищують рівень інтелектуальності системи та якість взаємодії з користувачем.

Таким чином, сформовано теоретичне підґрунтя для подальшої практичної розробки чат-бота, що включає вибір технологій, побудову архітектури та реалізацію функціональних можливостей системи.

РОЗДІЛ 2.

АНАЛІЗ ТА ПРОЄКТУВАННЯ СИСТЕМИ КОНФІГУРАТОРА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА

Сучасний ринок комп'ютерних комплектуючих характеризується великою різноманітністю моделей, частими оновленнями та різними рівнями продуктивності та цінової категорії. У таких умовах користувачу складно зробити оптимальний вибір компонентів для персонального комп'ютера, особливо якщо враховувати сумісність, індивідуальні потреби та обмежений бюджет.

Для спрощення цього процесу широко застосовуються конфігуратори ПК – інтерактивні системи, які допомагають автоматизувати підбір компонентів, оцінювати їх сумісність і надавати рекомендації відповідно до вимог користувача. Аналіз існуючих рішень та розробка власної концепції конфігуратора є важливими етапами створення ефективного інструменту, що забезпечує зручність користування та точність підбору комплектуючих.

2.1 Концепція конфігуратора ПК та його призначення

2.1.1 Поняття конфігуратора як інструменту підбору комп'ютерних комплектуючих

Конфігуратор комп'ютера – це програмний інструмент, призначений для автоматизованого підбору компонентів персонального комп'ютера на основі заданих користувачем параметрів і вимог. Його основна мета полягає у спрощенні процесу складання комп'ютера та забезпеченні сумісності всіх обраних елементів.

Традиційно користувач, який планує зібрати ПК, стикається з необхідністю самотійно перевіряти технічні характеристики кожної деталі – сокет процесора, тип оперативної пам'яті, потужність блока живлення, інтерфейси накопичувачів тощо. Конфігуратор усуває цю потребу, виконуючи

аналіз сумісності автоматично, що значно знижує ризик помилок і скорочує час підбору.

Основні функції конфігуратора можна поділити на такі категорії:

– *Аналітична функція.* Система порівнює технічні параметри компонентів і перевіряє відповідність між ними.

– *Рекомендаційна функція.* На основі вхідних критеріїв (наприклад, бюджет або призначення ПК) конфігуратор пропонує оптимальні варіанти компонентів.

– *Інформаційна функція.* Користувачу надаються відомості про характеристики, рейтинги, вартість та наявність компонентів.

– *Контрольна функція.* Система повідомляє про несумісність або відсутність певного елемента, пропонуючи альтернативи.

Сучасні конфігуратори працюють на базі баз даних з великим обсягом технічних характеристик обладнання. Вони можуть використовувати алгоритми фільтрації та рекомендацій, засновані на пошуку за параметрами, ключовими словами або навіть на штучному інтелекті.

Завдяки автоматизації процесу вибору, конфігуратори ПК широко застосовуються як у сфері електронної комерції (на сайтах магазинів), так і в освітніх чи дослідницьких проєктах, де важливо забезпечити раціональність технічних рішень. Крім того, конфігуратори можуть інтегруватися з системами онлайн-оплати, базами даних постачальників або сервісами підтримки користувачів, створюючи повноцінний екосистемний інструмент.

Таким чином, конфігуратор виступає не лише як допоміжний сервіс, а як інтелектуальний асистент, що поєднує аналіз технічних характеристик, логіку сумісності та рекомендаційні механізми для формування оптимальної конфігурації персонального комп'ютера.

2.1.2 Основні цілі використання конфігураторів

Основною метою використання конфігураторів комп'ютерних систем є спрощення процесу підбору комплектуючих та забезпечення їхньої технічної сумісності. Такі системи дозволяють користувачам без спеціальної технічної

підготовки створювати оптимальні конфігурації ПК, орієнтовані на конкретні потреби – ігри, роботу з графікою, офісні завдання чи серверні рішення.

Конфігуратори виконують низку завдань, що охоплюють як *технічний*, так й *інформаційний* аспекти взаємодії з користувачем.

1. *Забезпечення сумісності компонентів*. Однією з ключових проблем під час складання ПК є перевірка відповідності технічних параметрів – типу сокета процесора, форм-фактора материнської плати, стандарту оперативної пам'яті, потужності блока живлення тощо. Конфігуратор автоматично перевіряє ці параметри, запобігаючи вибору несумісних компонентів.

2. *Автоматизація процесу вибору*. Замість ручного перегляду великої кількості товарів користувач отримує готові рекомендації на основі заданих критеріїв – бюджету, рівня продуктивності, бренду або типу використання. Це значно скорочує час прийняття рішення.

3. *Підвищення точності рекомендацій*. Використання алгоритмів аналізу даних або методів обробки природної мови NLP[23] дозволяє конфігураторам розуміти запити користувача у довільній формі та формувати відповіді, максимально наближені до його потреб.

4. *Освітня та інформаційна функція*. Багато користувачів не володіють достатніми знаннями щодо характеристик комплектуючих. Конфігуратори не лише пропонують сумісні варіанти, а й пояснюють причини вибору, надають додаткові відомості про продуктивність, енергоспоживання чи ринкову вартість компонентів.

5. *Оптимізація бюджету*. Система може враховувати цінові межі та формувати збалансовані конфігурації, що забезпечують найкраще співвідношення ціни й продуктивності. Це особливо важливо при складанні ПК для конкретних задач, де перевитрата ресурсів є недоцільною.

6. *Забезпечення зручності користувача*. Інтуїтивний інтерфейс і можливість отримання результатів у зручній формі (наприклад, через Telegram-бот або вебінтерфейс) роблять процес вибору комплектуючих швидким і доступним навіть для недосвідчених користувачів.

7. *Підтримка прийняття рішень.* Конфігуратор допомагає користувачу обрати не лише технічно можливу, а й найбільш доцільну конфігурацію, враховуючи цілі, сценарії використання та прогнозовану тривалість актуальності обраних компонентів.

Отже, головна цінність конфігуратора полягає в тому, що він поєднує в собі функції інформаційного довідника, експертної системи та інструменту рекомендацій, що дозволяє підвищити ефективність і точність процесу підбору комп'ютерних комплектуючих.

2.1.3 Користувацькі сценарії

Конфігуратор комп'ютерних систем орієнтований на різні категорії користувачів, кожна з яких має власні цілі, рівень підготовки та очікування від системи. Ефективність роботи конфігуратора значною мірою визначається тим, наскільки точно він може адаптуватися до конкретного сценарію використання.

1. *Новачки* (початківці користувачі). Ця категорія користувачів не має глибоких технічних знань і часто не орієнтується у специфікаціях компонентів.

Для них важливо, щоб система забезпечувала:

- простий і зрозумілий інтерфейс взаємодії;
- наочні рекомендації з поясненням вибору (“чому саме цей процесор підходить”);
- можливість вибору за простими критеріями, наприклад, “для навчання”, “для офісу”, “для дому”;
- попередження про несумісність або перевищення бюджету.

У таких випадках конфігуратор виступає як інтелектуальний консультант, що супроводжує користувача на кожному етапі підбору компонентів, мінімізуючи кількість технічних термінів.

2. *Геймери.* Для користувачів, які збирають ПК для ігор, пріоритетом є продуктивність системи та баланс між потужністю графічного процесора, процесора і системи охолодження.

Конфігуратор для геймерів зазвичай має такі функції:

- фільтрація за рівнем FPS або продуктивності у популярних іграх;
- рекомендації оптимальних пар “CPU + GPU”;
- можливість вибору конфігурацій за типом ігор (кіберспорт, AAA-проекти, VR тощо);
- орієнтація на співвідношення ціна/продуктивність.

Крім того, геймери часто прагнуть до персоналізації – тож конфігуратор може пропонувати компоненти з урахуванням дизайну корпусу, RGB-підсвічування чи сумісності з периферією.

3. *Професійні користувачі.* До цієї групи належать фахівці, які працюють із ресурсоемними програмами – інженери, дизайнери, програмісти, відеомонтажери тощо. Їхні потреби відрізняються більш складними технічними вимогами.

Для них конфігуратор повинен:

- дозволяти задавати розширені параметри пошуку (кількість ядер, обсяг оперативної пам’яті, підтримувані інтерфейси тощо);
- враховувати специфіку програмного забезпечення (наприклад, сумісність із CAD, Blender, Adobe Premiere);
- надавати аналітичну інформацію щодо продуктивності в конкретних задачах (рендеринг, кодування, обчислення);
- пропонувати можливість збереження та порівняння кількох конфігурацій.

Для цієї категорії важливою є не лише автоматизація, а й гнучкість налаштувань, яка дозволяє адаптувати систему під професійні потреби.

2.1.4 Типи конфігураторів

Залежно від способу взаємодії з користувачем та середовища, у якому реалізовано функціонал, конфігуратори комп'ютерних систем можна поділити на три основні типи: вебплатформи, мобільні додатки та інтегровані боти.

1. *Вебплатформи.* Це найпоширеніший тип конфігураторів, реалізований у вигляді вебсайту з інтерактивним інтерфейсом. Користувач може послідовно обирати компоненти, переглядати їх характеристики, сумісність і загальну вартість системи. Прикладами таких рішень є PCPartPicker, CyberPowerPC, Rozetka Build Configurator.

Перевагами вебплатформ є:

- зручність доступу з будь-якого пристрою;
- можливість швидкого оновлення бази даних;
- інтеграція з інтернет-магазинами;
- наочне відображення конфігурації та ціни.

Недоліком є необхідність постійного підключення до Інтернету та складність у реалізації персоналізованої взаємодії з користувачем.

2. *Мобільні додатки.* Мобільні конфігуратори розробляються для операційних систем Android або iOS та надають користувачу можливість створювати конфігурації у зручному мобільному форматі.

Переваги:

- доступність у будь-який час і місці;
- можливість збереження збірок локально;
- інтеграція з системами сповіщень, що дозволяє оперативно отримувати оновлення про ціни чи наявність компонентів.

Однак такі додатки потребують окремої підтримки під різні платформи та частого оновлення, що підвищує складність розробки.

3. *Інтегровані боти.* Це сучасний тип конфігураторів, реалізований у вигляді чат-ботів у месенджерах, зокрема Telegram. Вони поєднують переваги інтерактивної взаємодії та простоти доступу. Користувач може формулювати

запит у звичайній текстовій формі, а бот – інтерпретувати його за допомогою методів обробки природної мови NLP і надати персоналізовану відповідь.

Переваги цього підходу:

- відсутність необхідності встановлення окремих додатків;
- природна комунікація у форматі “питання–відповідь”;
- можливість інтеграції з базами даних, API магазинів та системами рекомендацій.

Такий тип конфігуратора є зручним для користувачів із базовим рівнем технічних знань і є основою для розробки Telegram-бота-конфігуратора, реалізованого в межах цієї роботи.

У таблиці 2.1 наведено порівняння основних типів конфігураторів ПК за ключовими характеристиками.

Таблиця 2.1

Порівняння основних типів конфігураторів ПК

Критерій	Вебплатформа	Мобільний додаток	Інтегрований бот
Доступність	Через будь-який браузер	Через мобільні пристрої (Android, iOS)	Через месенджер, доступний у користувача
Зручність використання	Висока, завдяки візуальному інтерфейсу	Висока, оптимізована під сенсорне керування	Висока, завдяки спілкуванню в природній формі
Необхідність встановлення	Не потребує	Потребує інсталяції	Не потребує
Персоналізація	Обмежена	Висока (збереження профілю, історії вибору)	Залежить від реалізації NLP
Швидкість роботи	Залежить від браузера та інтернету	Зазвичай висока	Висока, але залежить від сервера
Інтеграція з магазинами	Легка, напряму через API	Потребує окремої реалізації	Може здійснюватися через зовнішні сервіси
Основні користувачі	Геймери, ентузіасти	Користувачі смартфонів	Новачки, які шукають швидке рішення
Приклади	PCPartPicker, Logical Increments	BuildMyPC App, Custom PC Builder	Telegram PC Configurator Bot

Таким чином, кожен тип конфігуратора має свої переваги й обмеження, однак найперспективнішим напрямом розвитку на сьогодні є *інтегровані боти*, які поєднують гнучкість вебплатформ із персоналізованим підходом до користувача.

2.2 Аналіз існуючих аналогів ботів-конфігураторів

Для обґрунтування підходів, застосованих у розробці власного Telegram-бота-конфігуратора, важливо оцінити існуючі рішення на ринку. Аналіз охоплює популярні вебплатформи, мобільні додатки та інтегровані боти, які забезпечують підбір комп'ютерних комплектуючих. Вивчення функціональних можливостей цих систем дозволяє визначити їхні переваги та недоліки, а також виявити особливості взаємодії з користувачем, рівень автоматизації процесу підбору і методи забезпечення сумісності компонентів.

Особлива увага приділяється способам представлення інформації про комплектуючі, наявності рекомендаційних механізмів, можливості фільтрації та сортування компонентів за різними критеріями. Крім того, оцінюється користувацький досвід у взаємодії з платформами, включаючи швидкість отримання результату, інтуїтивність інтерфейсу та доступність персоналізації під конкретні потреби користувача.

2.2.1 Огляд популярних платформ

PCPartPicker[24] – одна з найпопулярніших вебплатформ для підбору комплектуючих ПК, яка орієнтована як на новачків, так і на досвідчених користувачів (рис. 2.1). Система дозволяє обирати процесори, материнські плати, відеокарти, оперативну пам'ять, накопичувачі та інші компоненти, одночасно перевіряючи їхню сумісність.

Ключові функції *PCPartPicker*[24]:

– автоматична перевірка сумісності. Платформа контролює відповідність сокетів, форм-факторів, типів пам'яті та потужності блока живлення;

- рекомендації та фільтри. Користувач може сортувати компоненти за ціною, брендом, продуктивністю та популярністю;
- порівняння конфігурацій. Система дозволяє створювати декілька збірок і порівнювати їх за технічними характеристиками та вартістю;
- оцінки та відгуки користувачів. Надає інформацію про досвід інших користувачів і рейтинг компонентів;
- інтеграція з магазинами. Платформа показує ціни та наявність у різних інтернет-магазинах, що дозволяє швидко здійснити покупку.

Component	Selection	Base	Promo	Shipping	Tax	Availability	Price	Where
CPU	+ Choose A CPU							
CPU Cooler	+ Choose A CPU Cooler							
Motherboard	+ Choose A Motherboard							
Memory	+ Choose Memory							
Storage	+ Choose Storage							
Video Card	+ Choose A Video Card							
Case	+ Choose A Case							
Power Supply	+ Choose A Power Supply							
Operating System	+ Choose An Operating System							
Monitor	+ Choose A Monitor							
Expansion Cards / Networking	Sound Cards, Wired Network Adapters, Wireless Network Adapters							
Peripherals	Headphones, Keyboards, Mice, Speakers, Webcams							
Accessories / Other	Case Accessories, Case Fans, Fan Controllers, Thermal Compound, External Storage, Optical Drives, UPS Systems							

Рисунок 2.1 – Інтерфейс вебплатформи PCPartPicker[24]

Платформа розроблена таким чином, щоб користувач міг швидко і легко орієнтуватися у великій кількості компонентів. Зручність проявляється у:

- інтуїтивно зрозумілому інтерфейсі із наочним відображенням компонентів та їх характеристик;
- можливості збереження конфігурацій і швидкого переходу між різними збірками;
- детальних описах та технічних характеристиках кожного компонента;
- рейтингах та відгуках користувачів, які допомагають зробити обґрунтований вибір навіть без глибоких технічних знань.

PCPartPicker забезпечує надійний контроль сумісності компонентів, що дозволяє уникнути помилок при складанні ПК а саме:

- автоматична перевірка відповідності сокетів процесора та материнської плати;
- контроль форм-факторів корпусу та материнської плати;
- перевірка сумісності типів пам'яті та обсягів;
- оцінка необхідної потужності блока живлення та сумісності інших компонентів;
- попередження про будь-які невідповідності для запобігання помилок при складанні ПК.

Logical Increments[25] – це онлайн-сервіс, що допомагає користувачам підібрати оптимальні конфігурації ПК залежно від бюджету та призначення. Платформа орієнтована переважно на користувачів, які не мають достатнього технічного досвіду для самостійного підбору комплектуючих, тому пропонує готові варіанти збірок, розділені за рівнями продуктивності – від базових до високопродуктивних систем (рис. 2.2).

	Graphics	CPU	CPU Cooler	Motherboard	RAM	HDD	SSD	Power Supply	Case	Total
Destitute	UHD 610 \$0	G5900 \$44	Stock \$0	ASUS B460M-A \$68	4GB DDR4 \$21	None \$0	256GB \$38	Tt 500W (W) \$40	Sama Jazovo \$43	\$254
		G5905 \$50	212 Black \$25	ASUS H510M-E \$126	8GB DDR4 \$33	2TB \$60	\$12GB \$48	MSI 550W (B) \$55	Thermaltake V100 \$50	
Poor	Vega 8 \$0	R3 3200G \$74	Stock \$0	ASRock B450M-HDV \$80	4GB DDR4 \$21	None \$0	256GB \$38	Tt 500W (W) \$40	Sama Jazovo \$43	\$296
			212 Black \$25	ASUS B450M-A \$130	8GB DDR4 \$33	2TB \$60	\$12GB \$48	MSI 550W (B) \$55	Thermaltake V100 \$50	
Minimum	Vega 7 \$0	R5 5600G \$135	Stock \$0	ASRock B450M-HDV \$80	4GB DDR4 \$21	2TB \$60	\$12GB \$48	Tt 500W (W) \$40	Sama Jazovo \$43	\$427
			212 Black \$25	ASUS B450M-A \$130	8GB DDR4 \$33		1TB \$68	MSI 550W (B) \$55	Thermaltake V100 \$50	
Entry	▲ RX 550 \$71	i3 12100F \$80	Stock \$0	ASRock Z790 PRO RS \$139	8GB DDR5 \$30	2TB \$60	\$12GB \$48	Tt 550W (B) \$60	Focus G \$65	\$553
			A620 PRO SE \$30	Gigabyte B760M DS3H \$160	16GB DDR5 \$71		1TB \$68	Super Flower 550W (B) \$65	Nova Mesh \$70	
Modest	RX 6500XT \$180	i3 12100F \$80	Stock \$0	ASRock Z790 PRO RS \$139	8GB DDR5 \$30	2TB \$60	\$12GB \$48	Tt 550W (B) \$60	Focus G \$65	\$662
			A620 PRO SE \$30	Gigabyte B760M DS3H \$160	16GB DDR5 \$71		1TB \$68	Super Flower 550W (B) \$65	Nova Mesh \$70	
Fair	Arc A750 \$200	i3 12100F \$80	Spirit 120 \$50	ASRock Z790 PRO RS \$139	8GB DDR5 \$30	2TB \$60	\$12GB \$48	Tt 550W (B) \$60	Nova Mesh \$70	\$737
			NH-U12S Rdx \$55	Gigabyte B760M DS3H \$160	16GB DDR5 \$71		1TB \$68	Super Flower 550W (B) \$65	Focus 2 \$80	
Good	Arc A750 \$200	▲ i5 12400F \$109	Spirit 120 \$50	ASRock Z790 PRO RS \$139	8GB DDR5 \$30	2TB \$60	\$12GB \$48	Tt 550W (B) \$60	Pure Base 500 \$80	\$776
			NH-U12S Rdx \$55	Gigabyte B760M DS3H \$160	16GB DDR5 \$71	4TB \$85	1TB \$68	Super Flower 550W (B) \$65	H510 \$95	

Рисунок 2.2 – Інтерфейс вебплатформи Logical Increments[25]

Основний функціонал сервісу полягає у наданні рекомендацій щодо оптимального поєднання компонентів, виходячи з виділеного бюджету. Кожен рівень супроводжується списком рекомендованих процесорів, відеокарт, пам'яті, накопичувачів та інших комплектуючих, що забезпечують збалансовану продуктивність системи.

Особливістю Logical Increments[25] є те, що користувач не може самостійно змінювати або комбінувати компоненти всередині конфігурації безпосередньо на сайті. На відміну від платформ типу PCPartPicker, сервіс не має інтерактивного «конструктора ПК» і не виконує перевірку сумісності у реальному часі. Натомість він пропонує узгоджені варіанти збірок, підібрані експертами, що спрощує процес вибору, але обмежує гнучкість користувача.

Функціонал додатку:

- Готові збірки. Платформа пропонує попередньо підібрані конфігурації для різних бюджетів і потреб: офісні ПК, геймерські системи, робочі станції тощо.
- Рекомендації експертів. Усі комбінації комплектуючих перевірені фахівцями, що гарантує їхню взаємну сумісність.
- Посилання на магазини. Для кожної збірки надано перелік рекомендованих товарів із посиланнями на інтернет-магазини.
- Орієнтація на бюджет. Користувач може швидко знайти збірку, що відповідає його фінансовим можливостям. категоріях.

Logical Increments[25] є досить зручним для користувача: має інтуїтивно зрозумілу таблицю з рівнями («Entry», «Good», «Great», «Superb» тощо); кожен рівень містить опис очікуваної продуктивності та призначення (робота, ігри, професійні завдання); користувач легко може знайти збірку, яка відповідає його потребам і бюджету.

Зі сторони перевірки сумісності Logical Increments[25] не надає інтерактивної перевірки сумісності у режимі реального часу. Усі запропоновані збірки є заздалегідь перевіреними експертами. Користувач не

має змоги змінювати окремі компоненти безпосередньо на сайті, що мінімізує ризик помилки, але обмежує гнучкість конфігурації.

Telemart[26] – це онлайн-сервіс від українського магазину Telemart, який дозволяє користувачам самостійно збирати персональний комп'ютер, підбираючи сумісні комплектуючі та перевіряючи конфігурацію в режимі реального часу (рис. 2.3). Платформа орієнтована як на новачків, так і на досвідчених користувачів, які хочуть скласти збалансовану систему під свій бюджет і цілі.

Комплектуючі

Процесор	+ Додати
Материнська плата	+ Додати
Відеокарта	+ Додати
Модулі пам'яті	+ Додати
SSD диск	+ Додати
HDD диск	+ Додати
Система охолодження	+ Додати

Рисунок 2.3 – Інтерфейс конфігуратора Telemart[26]

Ключові функції Telemart[26]:

– Підбір комплектуючих. Користувач може обирати процесори, материнські плати, відеокарти, оперативну пам'ять, накопичувачі та інші компоненти для складання ПК з нуля або для оновлення існуючого.

- Реальна перевірка сумісності. Платформа автоматично перевіряє сумісність вибраних компонентів за основними параметрами (сокет, форм-фактор, потужність блоку живлення, обмеження слотів).
- Підтримка фільтрів і критеріїв. Користувач може обмежувати варіанти компонентів за брендом, ціною, характеристиками або рейтингом.
- Інтеграція з магазинами. Після формування конфігурації користувач може одразу додати вибрані комплектуючі до кошика Telemart для покупки.

За зручністю використання:

- інтерфейс дозволяє розпочати збірку з будь-якого компоненту, без фіксованого порядку вибору;
- система повідомляє про помилки або невідповідності в конфігурації в режимі реального часу, підказуючи, які компоненти слід замінити;
- є можливість звернутися за консультацією до фахівців магазину у разі складних або нестандартних збірок.

Особливості перевірки сумісності. Усі обрані компоненти перевіряються автоматично в процесі конфігурації. Система повідомляє про потенційні конфлікти та обмеження для кожного елемента, що дозволяє уникнути помилок і гарантує працездатність зібраної системи, одночасно залишаючи користувачу повну свободу у виборі компонентів.

2.2.2 Порівняльна характеристика аналогів

Для аналізу сучасних інструментів підбору комплектуючих ПК було розглянуто три популярні сервіси: PCPartPicker, Logical Increments та Telemart. Кожен із них має власні особливості, які впливають на зручність використання, функціональні можливості та ступінь контролю користувача над вибором компонентів.

PCPartPicker[24] пропонує гнучкий інтерактивний конструктор, що дозволяє користувачу самостійно комбінувати процесори, відеокарти, материнські плати та інші комплектуючі. Платформа автоматично перевіряє

сумісність компонентів та надає можливість порівняти різні конфігурації, а також інтегрує інформацію про ціни і наявність у різних інтернет-магазинах.

Logical Increments[25] орієнтований на користувачів, які шукають готові збалансовані конфігурації для різних бюджетів і сценаріїв використання. Платформа пропонує таблицю рівнів продуктивності, де для кожного рівня зазначено рекомендовані комплектуючі. Сервіс не дозволяє самостійно комбінувати компоненти, однак усі запропоновані варіанти перевірені експертами і гарантовано сумісні.

Telemart[26] – онлайн-конфігуратор українського магазину, що дозволяє користувачеві збирати ПК, обираючи будь-які компоненти та перевіряючи сумісність у реальному часі. Платформа поєднує свободу вибору, автоматичну перевірку сумісності і можливість одразу додати вибрані компоненти до кошика для покупки.

У таблиці 2.2 наведено порівняльну таблицю ключових переваг та недоліків кожного сервісу.

Таблиця 2.2

Ключові переваги та недоліки кожного з сервісів

Платформа	Переваги	Недоліки
PCPartPicker[24]	Інтерактивний конструктор; перевірка сумісності; порівняння конфігурацій; інтеграція з магазинами	Може бути складним для новачків
Logical Increments[25]	Простота використання; готові перевірені збірки; зрозуміла таблиця рівнів продуктивності	Відсутність можливості самостійного комбінування компонентів; обмежена гнучкість
Telemart[26]	Свобода вибору компонентів; перевірка сумісності у реальному часі; інтеграція з кошиком магазину	Обмежений вибір магазинів (тільки Telemart);

2.3 Опис функцій та можливостей моделі Telegram-бота

Розроблений Telegram-бот-конфігуратор ПК є інтерактивним інструментом, що дозволяє користувачам здійснювати підбір комп'ютерних компонентів у зручному середовищі месенджера. Завдяки використанню

методів обробки природної мови NLP[23], бот здатний інтерпретувати текстові запити користувача та надавати релевантні рекомендації. Він поєднує автоматизацію процесу вибору, персоналізацію рекомендацій та підтримку різних категорій користувачів, забезпечуючи швидку та ефективну взаємодію з системою.

2.3.1 Основна ідея створення бота-конфігуратора

Основна ідея створення Telegram-бота-конфігуратора полягає у спрощенні процесу підбору комп'ютерних компонентів для користувачів різного рівня підготовки – від новачків до досвідчених геймерів і професійних користувачів. Традиційний підбір ПК може бути складним через велику кількість параметрів компонентів і необхідність орієнтуватися у їхніх характеристиках. Бот забезпечує зручний інструмент, який дозволяє швидко формувати рекомендації щодо компонентів, враховуючи бюджет, призначення та технічні вимоги користувача.

Використання методів обробки природної мови NLP[23] дозволяє боту аналізувати текстові запити користувачів і формувати персоналізовані рекомендації. Це забезпечує адаптацію під індивідуальні потреби користувача, оскільки бот здатний визначати ключові вимоги у довільних текстових запитах і пропонувати оптимальні компоненти відповідно до цих параметрів.

Реалізація цього бота спрямована на зменшення навантаження на користувача при виборі комплектуючих, спрощення процесу прийняття рішень та скорочення часу на підбір оптимальної конфігурації ПК. В результаті користувач отримує ефективний та інтерактивний інструмент, який надає рекомендації та допомагає зорієнтуватися серед великої кількості доступних компонентів.

2.3.2 Основні функції чат-бота

Розроблений Telegram-бот-конфігуратор ПК надає користувачам інструмент для ефективного підбору комп'ютерних компонентів. Основні функції бота спрямовані на спрощення взаємодії, персоналізацію

рекомендацій та забезпечення швидкого доступу до інформації про доступні комплектуючі. Система дозволяє користувачеві орієнтуватися серед великої кількості компонентів, враховувати власні потреби та бюджет, а також отримувати рекомендації у зручному текстовому форматі в Telegram.

Далі наведені основні функції які підтримує даний чат-бот.

1. *Пошук за категоріями (CPU, GPU, RAM тощо).* Користувач може обирати певну категорію компонентів і отримувати список доступних варіантів із коротким описом характеристик, що дозволяє швидко орієнтуватися в асортименті. Такий пошук значно спрощує навігацію та економить час при виборі конкретного типу компоненту.

2. *Персоналізовані рекомендації на основі NLP.* Бот аналізує текстові повідомлення користувача, виявляє ключові вимоги, такі як призначення ПК, бюджет або пріоритетні характеристики компонентів, і на цій основі пропонує оптимальні варіанти. NLP дозволяє системі розуміти природну мову, враховувати контекст і формулювати рекомендації, максимально наближені до очікувань користувача.

3. *Використання нечіткого пошуку (fuzzy matching).* Ця функція дозволяє коректно обробляти запити навіть у випадку орфографічних помилок, скорочень або неточних назв компонентів. Наприклад, користувач може написати «Ryzen 5900X» або «Rizen 5900» – бот розпізнає, що йдеться про один і той же процесор, і пропонує відповідні варіанти.

4. *Фільтрація та сортування компонентів.* Користувач може відбирати компоненти за різними параметрами: ціною, продуктивністю, брендом або популярністю. Це дозволяє звужити коло вибору і швидше знаходити відповідні компоненти, зосереджуючись на тих характеристиках, які є ключовими для конкретного користувача.

5. *Інтерактивна взаємодія.* Бот відповідає на довільні текстові запити, надає детальні характеристики компонентів і приклади їх застосування. Завдяки інтерактивності користувач може уточнювати запит або отримувати додаткову інформацію про будь-який компонент у реальному часі.

2.3.3 Ключові відмінності від аналогів

Розроблений Telegram-бот-конфігуратор ПК має ряд особливостей, які відрізняють його від традиційних платформ і вебсервісів для підбору комплектуючих. На відміну від популярних аналогів, таких як PCPartPicker чи Logical Increments, бот не обмежується статичними фільтрами або готовими збірками. Він забезпечує більш інтерактивний і персоналізований підхід, адаптуючись до індивідуальних потреб користувача і реагуючи на довільні текстові запити.

Ключові відмінності бота-конфігуратора можна виділити такі:

- *Інтеграція в Telegram.* Бот працює безпосередньо в популярному месенджері, що забезпечує зручний доступ без необхідності переходу на вебсайти або встановлення додаткових програм. Користувач може отримувати рекомендації та взаємодіяти з ботом у знайомому середовищі.

- *Персоналізація рекомендацій за допомогою NLP.* Система аналізує текстові запити користувача, визначає його потреби та формує рекомендації на основі довільних повідомлень. Це дозволяє враховувати бюджет, призначення ПК та пріоритетні параметри компонентів без необхідності вручну задавати фільтри.

- *Використання нечіткого пошуку (fuzzy matching).* Бот здатний коректно обробляти запити навіть із орфографічними помилками або неточними назвами компонентів, забезпечуючи швидке отримання релевантних результатів.

- *Адаптивність до різних типів запитів.* Користувач може формулювати запити у вільній формі, а бот все одно надає релевантні рекомендації, що робить взаємодію більш інтуїтивною та ефективною.

- *Зручність і швидкість підбору.* Завдяки поєднанню персоналізації та інтерактивності, користувач швидко отримує рекомендації без необхідності самостійно переглядати великий каталог компонентів.

Такий підхід дозволяє виділити Telegram-бота серед інших конфігураторів ПК, оскільки він поєднує простоту використання з високою точністю та персоналізацією рекомендацій.

2.4. Модель системи Telegram чат-бота

Для кращого розуміння принципів функціонування Telegram чат-бота-конфігуратора ПК було побудовано кілька моделей системи, які відображають її структуру, логіку роботи та взаємодію між основними компонентами. Такі моделі дозволяють візуалізувати процес обробки користувацьких запитів, внутрішні зв'язки між модулями програми та послідовність виконання дій.

На рис. 2.4 схема архітектури чат-бота, яка відображає основні компоненти системи та взаємозв'язки між ними. Користувач надсилає повідомлення через Telegram API, яке передається на сервер, де послідовно працюють модулі попередньої обробки тексту, NLP-аналізу та пошуку. Після аналізу запиту система звертається до бази даних, отримує релевантні результати та повертає сформовану відповідь користувачу в Telegram.

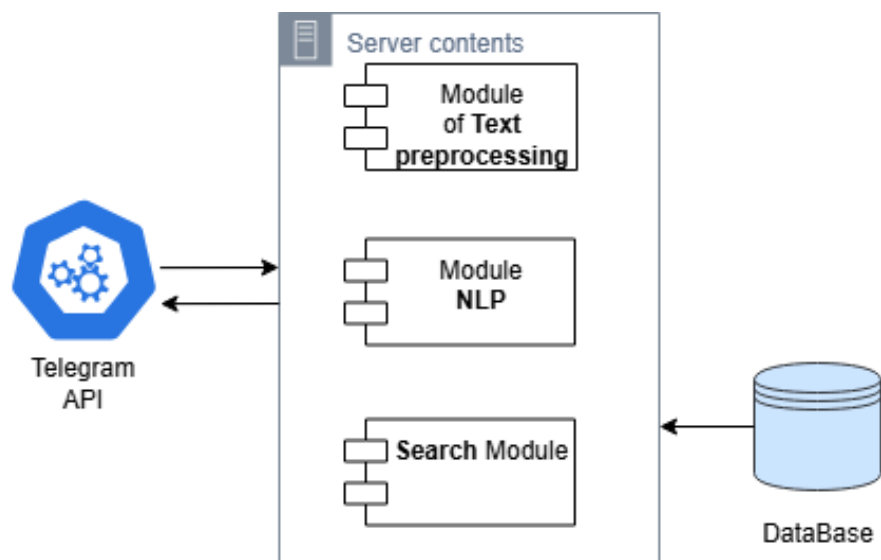


Рисунок 2.4 – Схема архітектури чат-бота

На рис. 2.5 наведена спрощена діаграма обробки повідомлень, яка демонструє, як бот приймає текстові повідомлення від користувача, аналізує їх зміст за допомогою NLP, визначає категорію запиту і формує відповідь із відповідними рекомендаціями.

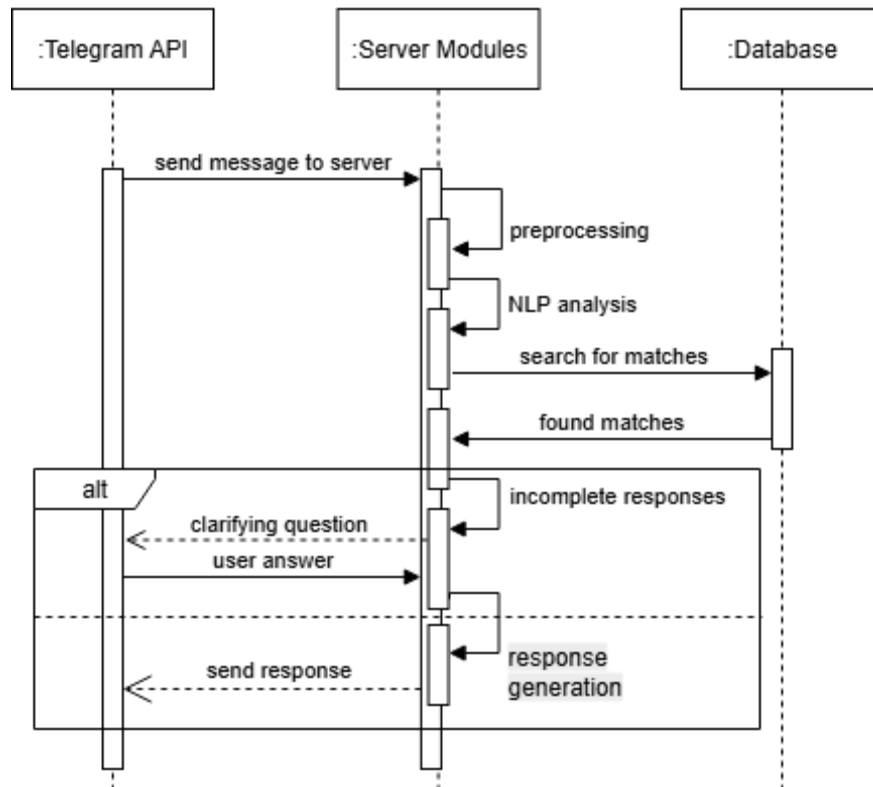


Рисунок 2.5 – Спрощена діаграма обробки повідомлень

На рис. 2.6 зображена діаграма активностей функції пошуку відеокарт, яка детальніше описує процес виконання однієї з основних функцій чат-бота – підбору відеокарти. На ній відображено послідовність дій: від введення користувацького запиту до відправлення результатів підбору, включно з етапами аналізу тексту та формування відповіді.

Розроблені моделі допомагають систематизувати архітектуру програмного продукту, спростити подальшу розробку і тестування, а також забезпечити узгодженість між функціональними компонентами Telegram-бота.

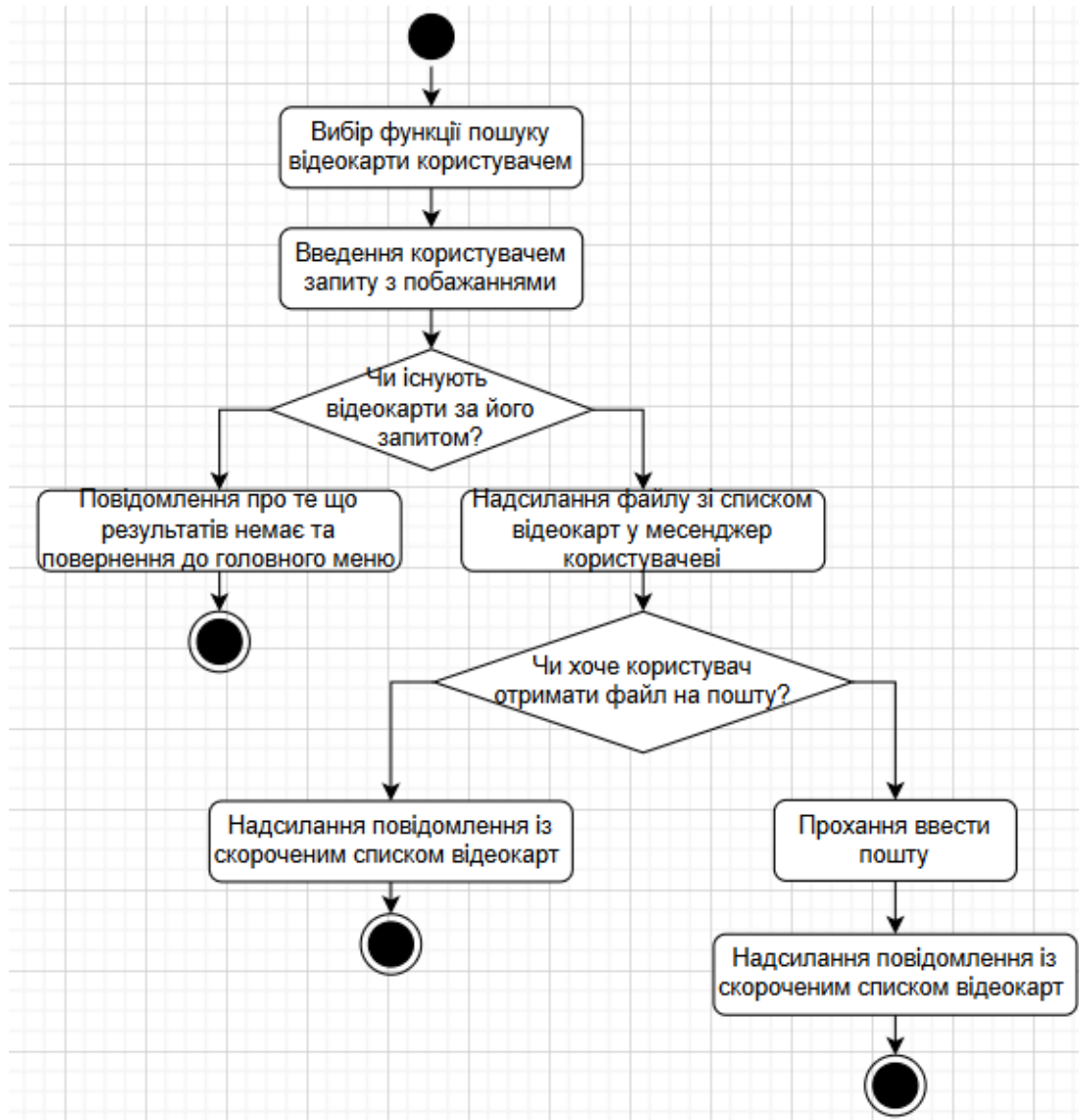


Рисунок 2.6 – Діаграма активностей функції пошуку відеокарт

Висновки до розділу 2

У другому розділі було розглянуто архітектуру Telegram чат-бота для підбору комп'ютерних комплектуючих, що поєднує методи обробки природної мови NLP та механізми пошуку за схожістю. Описано основні компоненти системи, принципи їх взаємодії та логіку обробки користувацьких запитів. Представлені схеми архітектури та діаграма обробки повідомлень ілюструють послідовність роботи бота – від отримання запиту через Telegram API до формування персоналізованої відповіді. Реалізована модель забезпечує гнучкість, масштабованість і можливість подальшого розширення функціональності системи.

РОЗДІЛ 3.

РЕАЛІЗАЦІЯ TELEGRAM ЧАТ-БОТА КОНФІГУРАТОРА ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА

3.1 Інструменти, використані при реалізації чат-бота

Для створення Telegram чат-бота конфігуратора персонального комп'ютера було використано набір сучасних технологій, які забезпечують стабільну роботу системи, зручність розробки та можливість подальшого розширення функціоналу. Вибір інструментів обумовлений необхідністю обробки природної мови, реалізації нечіткого пошуку та інтеграції з месенджером Telegram.

3.1.1 Мова програмування

Обрана мова програмування для реалізації даного чат-бота, стала Python[5] – високорівнева мова з відкритим вихідним кодом, що поєднує простоту синтаксису та потужність функціональних можливостей. Вона чудово підходить для розробки веб-застосунків, аналітичних систем, а також інтелектуальних рішень, зокрема таких, що використовують методи машинного навчання та обробки природної мови.

Python[5] має велику кількість готових бібліотек і фреймворків, що значно скорочують час розробки. У межах реалізації даного проєкту ця мова дозволила швидко організувати структуру чат-бота, реалізувати обробку користувацьких запитів, інтеграцію з Telegram API та модулі інтелектуального аналізу тексту. Простота синтаксису Python[5] забезпечує зручність підтримки коду та можливість розширення функціональності системи у майбутньому.

Особливу роль Python[2, 5] відіграє у сфері обробки природної мови NLP, що є ключовим елементом у роботі чат-бота. За допомогою бібліотек, таких як NLTK, Stanza та re, здійснюється попередня обробка введених користувачем запитів: токенизація, лематизація, фільтрація та пошук ключових слів. Це дозволяє системі розпізнавати смислове значення тексту та

надавати відповідні рекомендації щодо вибору комп'ютерних комплектуючих.

Крім того, Python[5] має зручні інструменти для роботи з Telegram API, серед яких найбільш поширеною є бібліотека `python-telegram-bot`. Вона забезпечує обробку повідомлень у реальному часі, підтримує різні типи даних (текст, фото, кнопки) та дозволяє легко налаштовувати логіку діалогу між користувачем і ботом. Такий підхід робить Python оптимальним вибором для створення інтелектуального чат-бота-конфігуратора.

3.1.2 Засіб інтеграції чат-бота з платформою Telegram

Для реалізації функціоналу чат-бота використано Telegram Bot API – офіційний інтерфейс програмування, який надає розробникам можливість створювати ботів, здатних взаємодіяти з користувачами через платформу Telegram. Цей інструмент забезпечує обмін повідомленнями, обробку команд, надсилання медіафайлів, створення кнопок, меню та інтерактивних елементів, що дозволяє будувати зручну і динамічну логіку спілкування з користувачем.

Telegram Bot API працює за принципом HTTP-запитів до серверів Telegram, через які бот може отримувати нові повідомлення (метод `getUpdates`) або надсилати відповіді користувачам (метод `sendMessage`). Такий підхід гарантує стабільну роботу системи, незалежно від кількості користувачів чи обсягу оброблюваних даних.

Завдяки Telegram Bot API реалізовано основну логіку спілкування між користувачем і системою: отримання текстових запитів, їх передача до модулів NLP для аналізу, формування відповіді та її повернення у вигляді зручного повідомлення. Простота інтеграції, стабільність та хороша документація роблять Telegram Bot API оптимальним вибором для створення надійних і функціональних ботів.

3.1.3 Бібліотека `python-telegram-bot` як основний інструмент розробки бота

Для реалізації Telegram чат-бота у рамках даного проєкту була обрана бібліотека `python-telegram-bot`[11], яка є популярним інструментом для

створення ботів на Python і забезпечує зручний інтерфейс для роботи з Telegram Bot API. Ця бібліотека дозволяє реалізувати як базові функції обробки повідомлень користувачів, так і складні сценарії взаємодії з ботом, включно з обробкою команд, клавіатур, inline-кнопок та callback-запитів.

Однією з ключових переваг є простота інтеграції з Telegram API, що скорочує час розробки та дозволяє зосередитися на логіці взаємодії з користувачем. Бібліотека підтримує асинхронне програмування, що забезпечує високу продуктивність бота при одночасній обробці великої кількості запитів, а наявність готових об'єктів і хендлерів спрощує обробку текстових повідомлень, команд, медіа-файлів та інших типів даних.

Завдяки цій бібліотеці реалізація складної логіки, включно з інтеграцією алгоритмів обробки природної мови для аналізу та інтерпретації запитів користувачів, стала більш гнучкою та структурованою. У рамках проєкту бібліотека використовується для прийому та обробки повідомлень від користувачів, організації діалогових сценаріїв із застосуванням меню та inline-кнопок, передачі отриманих даних у модулі обробки запитів та рекомендацій компонентів ПК, а також для повернення користувачу персоналізованих відповідей у вигляді тексту, списків та кнопок.

3.1.4 Бібліотеки NLP для обробки тексту

Для забезпечення аналізу та обробки текстових запитів користувачів у рамках даного проєкту застосовуються сучасні бібліотеки обробки природної мови NLP[23]. Однією з основних бібліотек є *NLTK*[14], яка дозволяє проводити токенізацію тексту, лематизацію та частиномовний аналіз. Використання *NLTK*[14] забезпечує можливість:

- розбивати текст на окремі слова та речення;
- визначати граматичні форми та базові лексичні значення;
- проводити початковий частиномовний аналіз, що необхідно для точного розуміння запитів користувачів.

Для більш глибокого морфологічного аналізу української мови застосовується бібліотека *Stanza*, яка дозволяє визначати граматичні категорії

слів, їхні морфологічні ознаки та синтаксичні зв'язки у реченні. Це підвищує точність обробки природної мови для українських текстів.

Додатково для розширення функціоналу та спрощення обробки текстових даних використовуються spaCy та TextBlob, що забезпечують:

- розпізнавання іменованих сутностей;
- визначення тональності тексту;
- спрощені методи лінгвістичного аналізу.

Окрему роль у системі відіграє бібліотека *RapidFuzz*[18], яка реалізує алгоритми нечіткого порівняння рядків (fuzzy matching). Вона використовується для виявлення подібності між словами та фразами, що дозволяє чат-боту коректно інтерпретувати запити користувача навіть за наявності орфографічних помилок або неточностей у формулюваннях. Завдяки цьому підвищується гнучкість і “людяність” взаємодії з користувачем, оскільки бот здатний розуміти запити, сформульовані у різних формах.

Крім спеціалізованих NLP-бібліотек, у роботі з текстом застосовується стандартна бібліотека *re*, яка дозволяє працювати з регулярними виразами для пошуку шаблонів у тексті та його попередньої обробки. Це дозволяє виділяти ключові слова, очищати повідомлення від небажаних символів та структурувати дані перед їх передачею до алгоритмів NLP, що підвищує точність аналізу запитів користувачів.

Поєднання всіх згаданих бібліотек забезпечує комплексний підхід до обробки запитів користувачів та формування коректних і релевантних відповідей у чат-боті.

3.1.5 Інші інструменти

У процесі розроблення Telegram чат-бота для автоматизації роботи з користувацькими запитами було використано низку допоміжних бібліотек та модулів, які не належать безпосередньо до NLP чи Telegram API, але відіграють важливу роль у забезпеченні функціональності, безпеки та зручності роботи системи.

Одним із базових інструментів є бібліотека `pandas`, яка використовується для роботи з табличними даними. Вона дозволяє зберігати, обробляти та фільтрувати інформацію у структурованому форматі, що особливо важливо під час формування бази даних для обробки користувацьких запитів і зберігання результатів взаємодії з ботом. `Pandas` надає гнучкі можливості для роботи з `DataFrame`, включно з очищенням, сортуванням, об'єднанням та агрегацією даних.

Також у роботі застосовується модуль `drive`, який забезпечує взаємодію з зовнішніми джерелами даних або хмарними сервісами, наприклад, `Google Drive`. Це дозволяє зберігати проміжні результати чи журнали роботи бота у віддаленому сховищі для подальшого аналізу або резервного копіювання.

Для надсилання електронних повідомлень у рамках функціоналу нотифікацій чи зворотного зв'язку використано модуль `email.message` у поєднанні з бібліотеками `smtplib` та `ssl`.

- `smtplib` реалізує протокол `SMTP`, який дає змогу програмно надсилати листи через поштові сервери.
- `ssl` використовується для шифрування з'єднання, забезпечуючи безпеку під час передачі конфіденційної інформації.
- `email.message` допомагає формувати структуру електронних листів, додаючи заголовки, тіло повідомлення чи вкладення.

Загалом, використання цих інструментів дозволяє інтегрувати `Telegram`-бот із зовнішніми сервісами, організувати надійний обмін даними та забезпечувати розширену функціональність системи. Їхнє поєднання створює основу для комплексного підходу до автоматизації роботи чат-бота – від збирання даних до безпечного сповіщення користувача.

3.2 Архітектура та структура бота

3.2.1 Загальна архітектура системи

Архітектура `Telegram` чат-бота побудована за модульним принципом, що забезпечує гнучкість, масштабованість та зручність супроводу

програмного коду. В основі системи лежить чіткий розподіл функціональних компонентів, кожен з яких відповідає за окремий етап обробки користувацьких запитів. Загальна структура може бути представлена як послідовність взаємодії таких основних блоків: Telegram API, Bot logic, NLP module та Database.

Компонент *Telegram API* забезпечує взаємодію між користувачем і системою. За його допомогою бот отримує вхідні повідомлення, обробляє команди, надсилає відповіді та мультимедійний контент. Цей рівень відповідає за обмін даними через сервери Telegram і виступає комунікаційним посередником між користувачем та внутрішньою логікою програми.

Другий рівень *Bot logic* реалізує основні алгоритми поведінки бота. Він відповідає за прийняття рішень, виклик необхідних функцій, перевірку умов і передачу даних до NLP-модуля. Саме тут визначається, яку відповідь має надати бот залежно від контексту запиту користувача. Логіка програми організована у вигляді окремих функцій і класів, що підвищує читабельність і спрощує модифікацію коду.

Модуль *NLP*[23] (Natural Language Processing) виконує попередню обробку природної мови, аналізує зміст запиту користувача, виявляє ключові слова та наміри. Він забезпечує розуміння тексту за допомогою інструментів токенізації, лематизації, морфологічного та частиномовного аналізу. Саме цей рівень надає боту можливість інтерпретувати запити, сформульовані природною мовою.

Останнім елементом архітектури є *Database*, що використовується для зберігання необхідних даних – наприклад, інформації про користувачів, історію діалогів, довідники чи набори комп'ютерних комплектуючих. Взаємодія з базою даних дає змогу забезпечити персоналізацію, швидкий пошук і доступ до попередніх результатів запитів.

3.2.2 Алгоритм роботи програми

Алгоритм роботи Telegram чат-бота базується на послідовній взаємодії між користувачем, Telegram API, логічним модулем програми та

компонентами обробки природної мови. Робота системи починається з надходження запиту від користувача і завершується формуванням текстової відповіді, яка надсилається назад у чат.

Після запуску програми бот ініціалізує з'єднання з Telegram API та переходить у режим очікування повідомлень. Коли користувач надсилає запит, система отримує його у вигляді текстового повідомлення, яке передається до основного логічного модуля програми. Цей модуль визначає тип запиту та вирішує, яким чином його необхідно обробити.

На наступному етапі повідомлення передається до модуля обробки природної мови NLP. Тут виконується токенізація тексту, лематизація, стемінг, частиномовний аналіз та виявлення ключових слів. Застосовуються бібліотеки NLTK, Stanza, spaCy, TextBlob, а також RapidFuzz[18] – для пошуку найбільш схожих варіантів у випадку неточних або неповних запитів. Отримані результати аналізу дають змогу визначити зміст запиту й намір користувача.

Після аналізу тексту логічний модуль звертається до відповідних функцій, щоб знайти потрібну інформацію. Для цього використовуються методи фільтрації та пошуку у таблицях, сформованих за допомогою pandas.

Після отримання необхідних даних формується структурована відповідь. Вона містить результати пошуку по запитові користувача. Перед відправленням повідомлення проходить фінальне форматування, після чого передається назад до Telegram API для надсилання користувачу.

Таким чином, робота чат-бота відбувається циклічно:

- 1) отримання запиту;
- 2) аналіз тексту;
- 3) пошук потрібної інформації;
- 4) формування відповіді;
- 5) відправлення результату користувачу.

Така послідовність забезпечує логічність, стабільність і гнучкість у роботі системи, дозволяючи легко адаптувати її під нові типи запитів та сценарії використання.

На рисунку 3.1 зображено процес обробки повідомлення користувача Telegram ботом для підбору комп'ютерних комплектуючих.

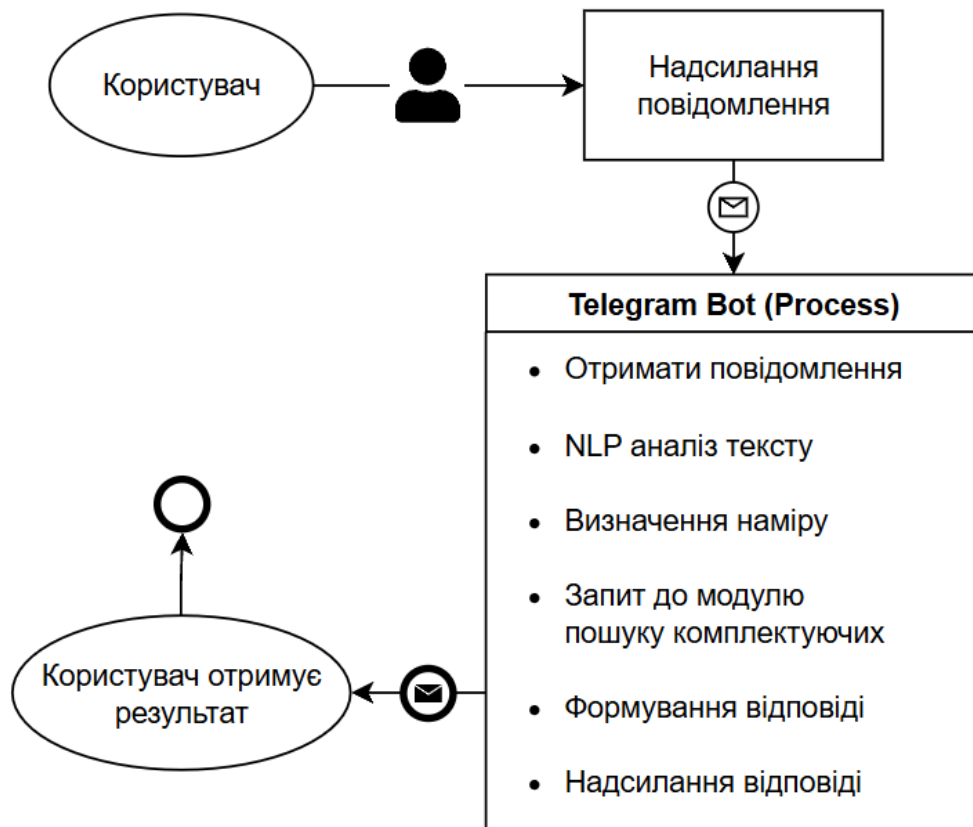


Рисунок 3.1 – BPMN-діаграма роботи чат-бота

3.3 Демонстрація функціоналу роботи бота

Демонстрація функціоналу чат-бота покликана показати практичну реалізацію його основних можливостей, логіку роботи та принцип взаємодії з користувачем у середовищі Telegram. Створений бот виконує роль інтелектуального асистента, який допомагає користувачу швидко знайти потрібні комп'ютерні комплектуючі, орієнтуючись на задані параметри та індивідуальні запити.

Після запуску програми користувач взаємодіє з ботом через Telegram-інтерфейс. Першим етапом є активація команди /start, після якої бот надсилає вітальне повідомлення та пропонує меню основних функцій. Це меню реалізовано за допомогою інлайн-клавіатури, що забезпечує зручну навігацію та швидкий вибір дій. На рис. 3.2 продемонстровано вигляд головного меню.

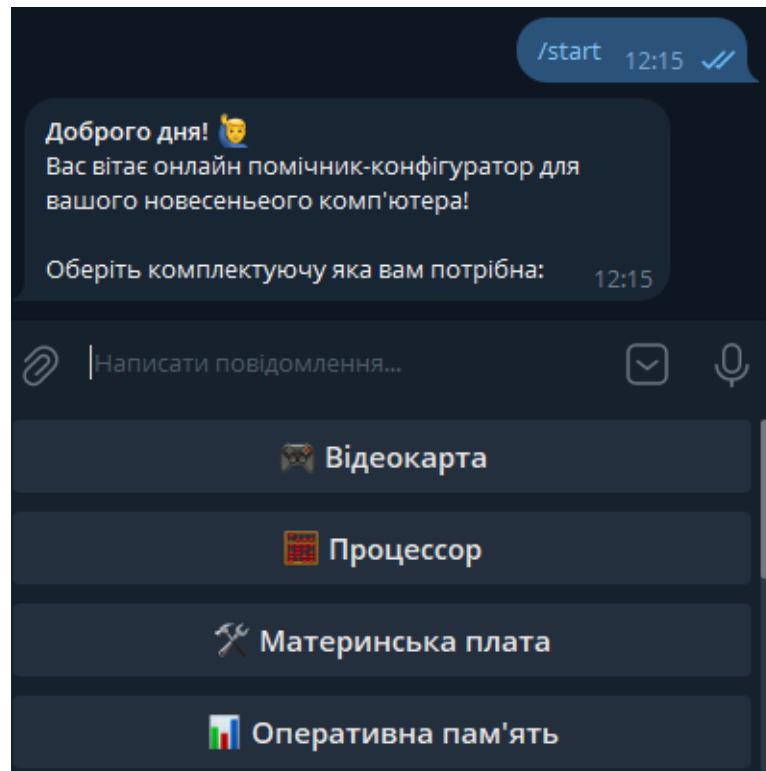


Рисунок 3.2 – Головне меню чат-бота

У головному меню представлено шість основних напрямів пошуку комплектуючих. Користувач може обрати категорію, що його цікавить, наприклад процесори, материнські плати, оперативна пам'ять, накопичувачі SSD чи HDD, відеокарти або блоки живлення.

Після вибору категорії бот переходить до етапу прийняття запиту від користувача. Користувач може задати такі параметри як: ціновий діапазон, бренд, тип роз'єму, обсяг пам'яті чи інші технічні характеристики. На основі введених даних бот обробляє запит за допомогою алгоритмів обробки природної мови NLP та механізму fuzzy matching, що дозволяє коректно

інтерпретувати навіть неточні або неповні формулювання. На р. 3.3 зображено приклад роботи функції пошуку відеокарти.

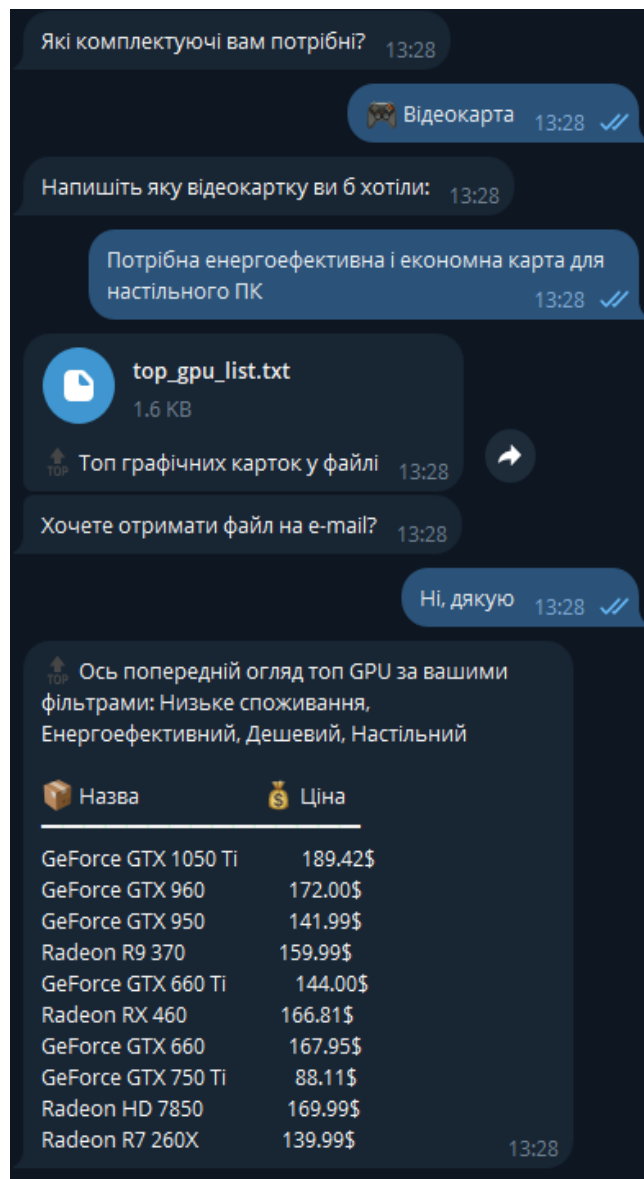


Рисунок 3.3 – Робота функції пошуку відеокарт

У результаті користувач отримує скорочену відповідь із переліком знайдених комплектуючих, який містить назву та орієнтовну ціну. Також користувач отримує файл який містить детальну інформацію про кожен з варіантів. Додатково чат-бот пропонує надіслати листа на електронну пошту якщо це є зручним для користувача. Таким чином, чат-бот поєднує елементи автоматизації, інтелектуальної обробки тексту та зручного користувацького

інтерфейсу, забезпечуючи ефективний процес підбору компонентів для персонального комп'ютера.

Далі буде продемонстровано роботу ще декількох функцій при різних запитах. На рис. 3.4 зображено роботу функції пошуку блоку живлення.

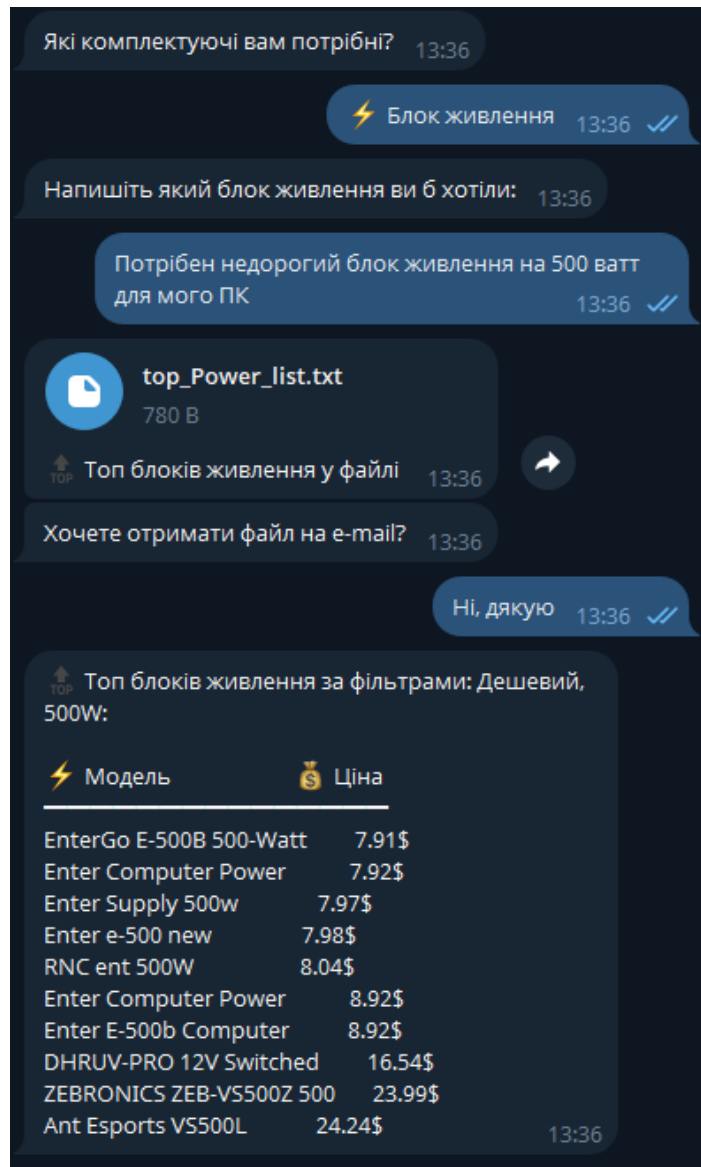


Рисунок 3.4 – Робота функції пошуку блоку живлення

На рис. 3.5 показано як бот буде реагувати на запити, які містять помилки.

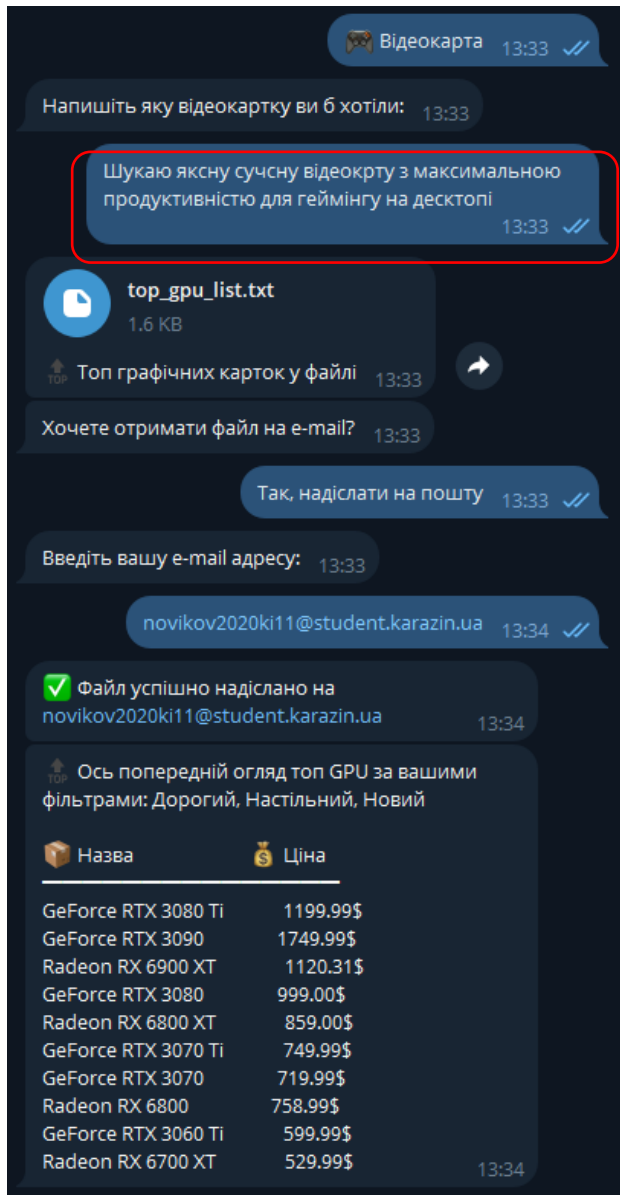


Рисунок 3.5 – Робота функції пошуку відеокарт при наявності помилок у запиті

На рис. 3.6 показано приклад файлу, який буде надіслано користувачеві у месенджер та, за потреби, на пошту.

☐ Топ GPU за фільтрами: Дорогий, Настільний, Новий

gpuName	G3Dmark	G2Dmark	total_performance	performance_per_dollar	gpuValue	TDP	powerPerformance	testYear	price	category
GeForce RTX 3080 Ti	26887	1031	27918	23.265194	22.41	350.0	76.82	2021	1199.99	Desktop
GeForce RTX 3090	26395	999	27394	15.653804	15.08	350.0	75.41	2020	1749.99	Desktop
Radeon RX 6900 XT	25458	1102	26560	23.707724	22.72	300.0	84.86	2020	1120.31	Desktop
GeForce RTX 3080	24853	1003	25856	25.881882	24.88	320.0	77.66	2020	999.00	Desktop
Radeon RX 6800 XT	23364	1078	24442	28.454016	27.20	300.0	77.88	2020	859.00	Desktop
GeForce RTX 3070 Ti	23367	1003	24370	32.493767	31.16	290.0	80.58	2021	749.99	Desktop
GeForce RTX 3070	22093	969	23062	32.031000	30.69	220.0	100.42	2020	719.99	Desktop
Radeon RX 6800	20667	1030	21697	28.586674	27.23	250.0	82.67	2020	758.99	Desktop
GeForce RTX 3060 Ti	20206	961	21167	35.278921	33.68	200.0	101.03	2020	599.99	Desktop
Radeon RX 6700 XT	18993	1014	20007	37.749769	35.84	230.0	82.58	2021	529.99	Desktop

Рисунок 3.6 – Приклад сформованого файлу

3.4 Оцінка ефективності роботи бота

Першим параметром оцінки є *точність* відповіді бота. Для цього були проведені тестові сесії з різними запитами користувачів, включно з формулюваннями, що містять орфографічні помилки або варіації формулювань. Використання модулів NLP та алгоритмів нечіткого порівняння рядків *fuzzy matching* дозволяє боту визначати правильний намір користувача навіть у разі неточностей у тексті. Результати тестів показали задовільний рівень відповідності між запитом і сформованою відповіддю.

Другим показником є *швидкість обробки повідомлень*. Вона вимірювалася як час від отримання повідомлення до відправлення відповіді користувачу. В середньому, бот обробляє запит за кілька секунд, що забезпечує комфортну взаємодію та відсутність затримок у діалозі. Така продуктивність є достатньою для чат-ботів інформаційного типу.

Ще одним аспектом оцінки є здатність системи працювати з *різними типами запитів*. Бот був протестований на запитах, що містять ключові слова, описові фрази та частково некоректні формулювання. Завдяки інтеграції бібліотек NLP і RapidFuzz[18] система змогла визначати релевантні результати у більшості випадків, що підтверджує її гнучкість і адаптивність.

Крім того, ефективність оцінюється за *показником користувацького досвіду*, що включає зручність взаємодії, зрозумілість відповідей та логічність побудови діалогу. В ході експериментальних перевірок було помічено, що бот надає чіткі, структуровані і корисні відповіді, що свідчить про задовільний рівень зручності.

Таким чином, проведений аналіз демонструє, що Telegram чат-бот ефективно виконує поставлені завдання: забезпечує швидку та точну обробку запитів, адаптується до різних формулювань користувачів і підтримує стабільну взаємодію. Це підтверджує доцільність використання запропонованої архітектури та обраних інструментів для автоматизації процесу підбору комп'ютерних комплектуючих.

Для більш наочного представлення функціональних відмінностей між розробленим Telegram чат-ботом та аналогом конфігуратора ПК від Telemart проведено порівняльний аналіз, результати якого наведено у таблиці 3.1.

Таблиця 3.1

Порівняння з аналогом

Характеристика	Розроблений Telegram чат-бот	Онлайн-конфігуратор Telemart
Тип взаємодії	Діалогова форма спілкування у Telegram	Веб-інтерфейс із покроковим вибором компонентів
Спосіб введення даних	Текстові запити користувача у довільній формі	Вибір зі списку доступних комплектуючих
Інтелектуальна обробка тексту	Використання NLP і fuzzy matching для розпізнавання наміру	Відсутня — лише статичні поля вибору
Гнучкість підбору	Може обробляти неточні, неповні або описові запити	Вимагає точного вибору кожного елемента
Перевірка сумісності компонентів	Може бути реалізована додатково	Автоматично вбудована у систему
Форма результату	Текстова відповідь із поясненнями та рекомендаціями	Готова конфігурація без опису
Зручність взаємодії	Не потребує відкриття сайту, працює безпосередньо в Telegram	Потребує роботи у браузері на сайті Telemart

3.5 Актуальність, новизна та перспективи розвитку

3.5.1 Особливості реалізованого рішення

Розроблене рішення ґрунтується на використанні технологій обробки природної мови NLP, що забезпечує його інтелектуальність та адаптивність. Завдяки цим технологіям бот здатний розуміти зміст користувацьких запитів, навіть якщо вони сформульовані у довільній формі, з помилками чи розмовними конструкціями. Це дозволяє вийти за межі стандартних сценаріїв діалогу, характерних для більшості чат-ботів, і реалізувати більш природну модель взаємодії між користувачем і системою.

Основна новизна полягає саме у застосуванні методів NLP для аналізу, лематизації та семантичного розпізнавання тексту у сфері підбору комп'ютерних комплектуючих. Це дає змогу системі визначати контекст запиту, знаходити ключові поняття й формувати релевантну відповідь без необхідності жорстко прописаних шаблонів. Такий підхід значно підвищує точність обробки інформації та якість взаємодії з користувачем.

Ще однією важливою особливістю є механізм автоматичного формування розширених результатів. Після обробки запиту система може створювати структурований файл із детальною інформацією, який користувач отримує безпосередньо у чаті або на електронну пошту. Це підвищує зручність використання й робить систему більш функціональною у порівнянні зі звичайними текстовими ботами.

Архітектура рішення має гнучку структуру, що дозволяє масштабувати програму, розширювати її базу даних і вдосконалювати алгоритми обробки запитів без суттєвих змін у логіці. Завдяки цьому проєкт має потенціал подальшого розвитку – зокрема, у напрямі підвищення точності мовного аналізу, підтримки багатомовності або інтеграції з іншими інформаційними системами.

Таким чином, запропоноване рішення поєднує у собі інтелектуальні методи NLP, адаптивну логіку відповіді та гнучку архітектуру, що робить його сучасним і перспективним прикладом використання штучного інтелекту в чат-ботах.

3.5.2 Потенційні напрями вдосконалення

Розроблений Telegram чат-бот має значний потенціал для подальшого розвитку та вдосконалення, що дозволяє підвищити його ефективність та практичну цінність. Одним із напрямків є збільшення словника системи, що забезпечить більш точне розпізнавання запитів користувачів, врахування різних формулювань, синонімів та можливих помилок у тексті. Це дозволить покращити точність обробки інформації та підвищити релевантність рекомендацій.

Ще одним важливим напрямком є додавання функції списку вподобань користувача, яка дозволяє зберігати окремі комплектуючі, що зацікавили користувача. Така можливість підвищує зручність роботи з ботом, оскільки користувач може швидко повернутися до раніше вибраних компонентів, порівняти їх або використати при подальшому формуванні конфігурації.

Систематичне оновлення датасетів новими моделями компонентів є ще одним перспективним напрямком. Це дозволяє підтримувати актуальність і повноту рекомендацій, відображати сучасні тенденції та нові технічні характеристики, що важливо для користувачів, які прагнуть отримати оптимальні сучасні конфігурації.

Крім того, доцільним є додавання посилань на купівлю окремих комплектуючих, що зробить взаємодію з системою більш практичною та допоможе користувачу швидко перейти від рекомендацій до реального придбання товарів. Це дозволить зменшити кількість додаткових дій і підвищити зручність процесу.

Серед інших напрямків розвитку – поглиблення використання технологій NLP, що дозволить краще аналізувати контекст запитів, розпізнавати складні мовні конструкції та покращувати адаптивність відповідей. Можливе також розширення каналів комунікації, інтеграція з іншими платформами та вдосконалення механізмів формування детальних результатів у різних форматах.

Додатково перспективним напрямком є використання машинного навчання для підвищення якості рекомендацій. Це дозволить системі навчатися на попередніх запитах і вподобаннях користувачів, передбачати найбільш оптимальні конфігурації та адаптуватися під індивідуальні потреби кожного користувача, що значно підвищить точність і персоналізацію результатів.

Завдяки реалізації цих напрямків розроблена система зможе забезпечувати більш інтелектуальну, персоналізовану та зручну взаємодію, що

підвищить її практичну цінність, перспективність для подальшого використання та потенціал масштабування в майбутньому.

Висновки до розділу 3

У розділі було детально розглянуто процес реалізації Telegram-бота-конфігуратора ПК. Було описано використані інструменти, серед яких Python, бібліотеки для роботи з Telegram API та обробки природної мови NLP, а також інструменти для роботи з даними та реалізації нечіткого пошуку.

Окремо висвітлено архітектуру системи та алгоритм роботи бота від надходження запиту до формування персоналізованої відповіді. Розглянуто демонстрацію функціоналу, наведено приклади взаємодії користувача з ботом, формування конфігурацій та візуалізацію результатів.

Також у розділі проведено оцінку ефективності роботи бота за критеріями точності обробки запитів, швидкості відповіді та зручності взаємодії. Порівняно результати з аналогічними сервісами, а також проаналізовано типові сценарії використання та можливі помилки користувачів.

Особлива увага приділена новизні реалізованого рішення: інтеграції NLP та методів fuzzy matching, простоті використання та перевагам над існуючими рішеннями. Визначено перспективні напрями вдосконалення, що включають розширення словника, додавання списку вподобань користувача, оновлення бази даних новими моделями, посилення на придбання комплектуючих та застосування машинного навчання для підвищення якості рекомендацій.

Загалом даний розділ підтвердив практичну цінність розробленого бота, його актуальність, новизну та потенціал для подальшого розвитку і вдосконалення.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було здійснено комплексне дослідження можливостей застосування телеграм-ботів для автоматизації процесу підбору комп'ютерних комплектуючих. Проведено аналіз сучасних технологій обробки природної мови NLP, що дозволяють інтерпретувати текстові запити користувачів у довільній формі та надавати релевантні рекомендації. Розглянуті методи пошуку й класифікації даних дали змогу визначити оптимальні підходи для побудови інтерактивної системи підбору конфігурацій ПК.

Дослідження наявних аналогів чат-ботів у сфері IT-консалтингу виявило їхні обмеження: відсутність персоналізації, вузький функціонал та недостатня гнучкість у роботі з даними. Це дозволило сформулювати вимоги до майбутньої системи та розробити архітектуру, орієнтовану на масштабованість, високу точність рекомендацій і зручність користувацької взаємодії.

У рамках реалізації було розроблено телеграм чат-бот, який забезпечує підбір комп'ютерних комплектуючих відповідно до бюджету, технічних потреб і користувацьких уподобань. Система використовує NLP для обробки запитів та алгоритми пошуку компонентів із динамічними правилами сумісності. Це значно спрощує процес конфігурації ПК, зменшує ймовірність помилок під час підбору компонентів та підвищує рівень доступності комп'ютерних технологій для широкого кола користувачів.

Реалізований інструмент демонструє потенціал подальшого розвитку: інтеграцію онлайн-магазинів, розширення бази комплектуючих, впровадження рекомендаційних моделей на основі машинного навчання та підтримку мультимодальних запитів. Отримані результати підтверджують, що застосування телеграм-ботів з технологіями NLP є перспективним напрямом для автоматизації процесів вибору та оптимізації комп'ютерних систем.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is a chatbot? [Електронний ресурс] – режим доступу URL: <https://www.ibm.com/think/topics/chatbots> (дата звернення 01.10.2025).
2. Lydia Evelyn, Bruce Hopkins. Beginning ChatGPT for Python: Build Intelligent Applications with OpenAI APIs. Apress 2024. 208 с.
3. Telegram API. [Електронний ресурс] – режим доступу URL: <https://core.telegram.org/> (дата звернення 01.10.2025).
4. Telegram API methods. [Електронний ресурс] – режим доступу URL: <https://core.telegram.org/methods> (дата звернення 02.10.2025).
5. Python documentation. [Електронний ресурс] – режим доступу URL: <https://docs.python.org/3/> (дата звернення 05.10.2025).
6. JavaScript documentation. [Електронний ресурс] – режим доступу URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення 05.10.2025).
7. Node.js documentation. [Електронний ресурс] – режим доступу URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/Introduction (дата звернення 05.10.2025).
8. Java documentation. [Електронний ресурс] – режим доступу URL: <https://docs.oracle.com/en/java/> (дата звернення 05.10.2025).
9. Go documentation. [Електронний ресурс] – режим доступу URL: <https://go.dev/doc/> (дата звернення 06.10.2025).
10. PHP documentation. [Електронний ресурс] – режим доступу URL: <https://www.php.net/docs.php> (дата звернення 06.10.2025).
11. Python-telegram-bot library documentation. [Електронний ресурс] – режим доступу URL: <https://docs.python-telegram-bot.org/en/stable/> (дата звернення 06.10.2025).
12. Aiogram library documentation. [Електронний ресурс] – режим доступу URL: <https://docs.aiogram.dev/en/v3.22.0/> (дата звернення 06.10.2025).

13. Telebot library documentation. [Електронний ресурс] – режим доступу URL: <https://pytba.readthedocs.io/en/latest/> (дата звернення 07.10.2025).
14. NLTK library documentation. [Електронний ресурс] – режим доступу URL: <https://www.nltk.org/> (дата звернення 07.10.2025).
15. SpaCy library documentation. [Електронний ресурс] – режим доступу URL: <https://spacy.io/api/doc> (дата звернення 07.10.2025).
16. Transformers library documentation. [Електронний ресурс] – режим доступу URL: <https://huggingface.co/docs/transformers/index> (дата звернення 07.10.2025).
17. FuzzyWuzzy library documentation. [Електронний ресурс] – режим доступу URL: <https://www.geeksforgeeks.org/python/fuzzywuzzy-python-library/> (дата звернення 07.10.2025).
18. RapidFuzz library documentation. [Електронний ресурс] – режим доступу URL: <https://pypi.org/project/RapidFuzz/> (дата звернення 07.10.2025).
19. PostgreSQL що це таке і чому воно важливе? [Електронний ресурс] – режим доступу URL: <https://wox.in.ua/threads/postgresql-scho-ce-take-i-chomu-vono-vazhlive.69/> (дата звернення 05.10.2025).
20. SQLite documentation. [Електронний ресурс] – режим доступу URL: <https://sqlite.org/docs.html> (дата звернення 08.10.2025).
21. MongoDB: що це за СУБД, переваги та недоліки. [Електронний ресурс] – режим доступу URL: <https://goit.global/ua/articles/mongodb-shcho-tse-take-ta-yak-prazui/> (дата звернення 08.10.2025).
22. Introduction to Personal Computer Hardware. [Електронний ресурс] – режим доступу URL: <https://www.ciscopress.com/articles/article.asp?p=3188196&seqNum=5> (дата звернення 03.10.2025).
23. Hobson Lane, Hannes Napke, Cole Howard. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python. Manning 2019. 544 с.

- 24.PCPartPicker. [Электронный ресурс] – режим доступа URL:
<https://pcpartpicker.com/> (дата звернення 03.10.2025).
- 25.Logical Increments. [Электронный ресурс] – режим доступа URL:
<https://www.logicalincrements.com/> (дата звернення 03.10.2025).
- 26.Telemart. [Электронный ресурс] – режим доступа URL:
<https://telemart.ua/ua/assembly-start.html> (дата звернення 03.10.2025).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Магістр
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних
систем та робототехніки
к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року



ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

НОВІКОВА Олексія Едуардовича
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«МОДЕЛЬ КОМП'ЮТЕРНОЇ СИСТЕМИ ПІДТРИМКИ КЛІЄНТІВ НА ОСНОВІ ЧАТ-БОТУ»**

керівник роботи **СТРІЛЕЦЬ Вікторія Євгенівна, кандидат технічних наук, доцент.**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету № 4101-5/3554 від 30 вересня 2025 року

2. Строк подання студентом роботи 30 листопада 2025 року

3. Перелік питань, які потрібно розробити

1. Провести аналіз принципів побудови Telegram чат-ботів, NLP-технологій та можливостей їх використання у системах підтримки клієнтів.
2. Розробити архітектуру чат-бота для конфігурації персонального комп'ютера.
3. Реалізувати модуль обробки природної мови для інтерпретації користувацьких запитів.
4. Створити алгоритми підбору комп'ютерних комплектуючих на основі введених даних.
5. Провести тестування, оцінку точності роботи та ефективності запропонованої моделі підтримки клієнтів при конфігурації персонального комп'ютера.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми роботи та керівника	30.09.2025 - 01.10.2025
2	Аналіз наукових джерел і методичних матеріалів щодо створення Telegram чат-ботів та застосування NLP-технологій	01.10.2025 - 03.10.2025
3	Вивчення існуючих підходів до автоматизації вибору комп'ютерних комплектуючих	03.10.2025 - 05.10.2025
4	Вибір інструментів і бібліотек для реалізації NLP-модулів	05.10.2025 - 08.10.2025
5	Розробка загальної архітектури Telegram чат-бота для конфігурації ПК	08.10.2025 - 17.10.2025
6	Реалізація основного функціоналу Telegram-бота з використанням Telegram API	17.10.2025- 25.10.2025
7	Інтеграція NLP-модуля для аналізу й обробки користувацьких запитів у природній мові	25.10.2025- 05.11.2025
8	Проведення функціонального тестування чат-бота та усунення виявлених помилок	05.11.2025- 07.11.2025
9	Оцінка точності обробки користувацьких запитів і ефективності роботи системи	07.11.2025- 08.11.2025
11	Оформлення звіту про науково-дослідну практику Написання статті за матеріалами кваліфікаційної роботи.	11.11.2025- 15.11.2025-
12	Підготовка і оформлення звітних матеріалів кваліфікаційної роботи. Оформлення списку літератури	15.11.2025- 18.11.2025-
13	Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР.	18.11.2025- 25.11.2025
14	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи.	25.11.2025- 27.11.2025
15	Оформлення звіту про переддипломну практику	27.11.2025 - 29.11.2025
16	Представлення кваліфікаційної роботи керівнику та рецензенту	29.11.2025 - 30.11.2025

5. Дата видачі завдання *02 жовтня 2025 року.*

Студент

О. Е. Новіков

ініціали, прізвище

підпис

Керівник роботи

В. Є. Стрілець

ініціали, прізвище

підпис

Затверджую

« _____ » _____ 2025 р.

**Технічне завдання
на розробку програмного виробу «Модель комп'ютерної системи
підтримки клієнтів на основі чат-боту»**

1.	Введення	1.1. Назва: Модель комп'ютерної системи підтримки клієнтів на основі чат-боту. 1.2. Галузь застосування: інформаційні технології, штучний інтелект, комп'ютерні системи
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу магістра № 4101-5/3554 від «30» вересня 2025 р. (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: автоматизувати процес підбору комп'ютерних комплектуючих за допомогою Telegram бота із використанням технологій обробки природної мови. 3.2. Призначення розробки: забезпечити користувачам зручний інструмент для формування оптимальної конфігурації персонального комп'ютера відповідно до заданих параметрів, бюджету та індивідуальних потреб, використовуючи інтелектуальний аналіз текстових запитів. 3.3. Вихідні дані розробки: текстові запити користувачів, база даних комп'ютерних комплектуючих, параметри користувацьких вимог
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: забезпечення зручного та швидкого підбору комп'ютерних комплектуючих на основі запитів користувача, формулювання рекомендацій із використанням технологій NLP, можливість надсилання результатів у чаті Telegram або на електронну пошту користувача. 4.2. Вимоги до надійності: забезпечення стабільної роботи Telegram-бота під час різних типів запитів користувача, коректна обробка помилкових або нечітких запитів, а також збереження безпеки доступу до бази даних комплектуючих.

		<p>4.3. Вимоги до умов експлуатації: доступ до глобальної мережі Internet.</p> <p>4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми підходить ПК або мобільний пристрій із будь-якою операційною системою (Windows, Linux, Mac OS, Android).</p> <p>4.5. Вимоги до інформаційної та програмної сумісності: підтримка сучасних ОС (Windows, Linux, Android), встановлений Python із необхідними бібліотеками (python-telegram-bot, NLTK, Stanza, RapidFuzz), наявність доступу до Telegram API.</p> <p>4.6. Вимоги до маркування та упаковки: вимоги до маркування та упаковки не представляються.</p> <p>4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються, можливе зберігання на хмарних носіях або локальних дисках.</p> <p>4.8. Спеціальні вимоги: захист персональних даних користувача та конфіденційності його запитів.</p>	
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «Модель комп'ютерної системи підтримки клієнтів на основі чат-боту» вважати:</p> <ol style="list-style-type: none"> 1) Ознайомчу сторінку з правилами користування ботом та прикладами запитів користувача. 2) Програму і методику випробувань розробленої системи (представити як додаток В до пояснювальної записки до кваліфікаційної роботи). 3) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи). 	
6.	Вимоги до техніко-економічних показників	<p>Програмною документацією до виробу «Модель комп'ютерної системи підтримки клієнтів на основі чат-боту» вважати:</p> <ol style="list-style-type: none"> 1) Дане Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи). 2) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи). 3) Джерела базової інформації. 	
7.	Стадії і етапи розробки	Дата	Назва етапу
		від 30 вересня 2025 до 1 жовтня 2025	Затвердження теми роботи та керівника.

		<p>від 1 жовтня 2025 до 3 жовтня 2025</p> <p>від 3 жовтня 2025 до 5 жовтня 2025</p> <p>від 5 жовтня 2025 до 8 жовтня 2025</p> <p>від 8 жовтня 2025 до 17 жовтня 2025</p> <p>від 17 жовтня 2025 до 25 жовтня 2025</p> <p>від 25 жовтня 2025 до 5 листопада 2025</p> <p>від 5 листопада 2025 до 7 листопада 2025</p> <p>від 7 листопада 2025 до 8 листопада 2025</p> <p>від 11 листопада 2025 до 15 листопада 2025</p> <p>від 15 листопада 2025</p>	<p>Аналіз наукових джерел і методичних матеріалів щодо створення Telegram чат-ботів та застосування NLP-технологій.</p> <p>Вивчення існуючих підходів до автоматизації вибору комп'ютерних комплектуючих.</p> <p>Вибір інструментів і бібліотек для реалізації NLP-модулів.</p> <p>Розробка загальної архітектури Telegram чат-бота для конфігурації ПК.</p> <p>Реалізація основного функціоналу Telegram-бота з використанням Telegram API.</p> <p>Інтеграція NLP-модуля для аналізу й обробки користувацьких запитів у природній мові</p> <p>Проведення функціонального тестування чат-бота та усунення виявлених помилок</p> <p>Оцінка точності обробки користувацьких запитів і ефективності роботи системи</p> <p>Оформлення звіту про науково-дослідну практику</p> <p>Написання статті за матеріалами кваліфікаційної роботи.</p> <p>Підготовка і оформлення звітних матеріалів кваліфікаційної роботи.</p>
--	--	---	--

		до 18 листопада 2025	Оформлення списку літератури.
		від 18 листопада 2025 до 25 листопада 2025	Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР.
		від 25 листопада 2025 до 27 листопада 2025	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи.
		від 27 листопада 2025 до 29 листопада 2025	Оформлення звіту про переддипломну практику
		від 29 листопада 2025 до 30 листопада 2025	Представлення кваліфікаційної роботи керівнику та рецензенту
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> 1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику. 	

Виконавець
студент групи КІ- 61
Новіков О. Е.



Замовник
канд. техн. наук
Стрілець В. Є.



Програма і методика випробувань
програмного виробу
«Модель комп'ютерної системи
підтримки клієнтів на основі чат-боту»

1 Об'єкт випробувань

1. Назва програмного виробу: «Модель комп'ютерної системи підтримки клієнтів на основі чат-боту».
2. Галузь застосування: Інформаційні технології.
3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

1. Вимоги до функціональних характеристик: забезпечення зручного та швидкого підбору комп'ютерних комплектуючих на основі запитів користувача, формулювання рекомендацій із використанням технологій NLP, можливість надсилання результатів у чаті Telegram або на електронну пошту користувача.

2. Вимоги до надійності: забезпечення стабільної роботи Telegram-бота під час різних типів запитів користувача, коректна обробка помилкових або нечітких запитів, а також збереження безпеки доступу до бази даних комплектуючих.

3. Вимоги до умов експлуатації: доступ до глобальної мережі Internet.

4. Вимоги до складу і параметрів технічних засобів: для виконання програми підходить ПК або мобільний пристрій із будь-якою операційною системою (Windows, Linux, Mac OS, Android).

5. Вимоги до інформаційної та програмної сумісності: підтримка сучасних ОС (Windows, Linux, Android), встановлений Python із необхідними бібліотеками (python-telegram-bot, NLTK, Stanza, RapidFuzz), наявність доступу до Telegram API.

6. Вимоги до маркування та упаковки: вимоги до маркування та упаковки не представляються.

7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються, можливе зберігання на хмарних носіях або локальних дисках.

8. Спеціальні вимоги: захист персональних даних користувача та конфіденційності його запитів.

5. Вимоги до програмної документації

Програмою документацією до виробу «Модель комп'ютерної системи підтримки клієнтів на основі чат-боту» вважати:

1. Ознайомчу сторінку з правилами користування ботом та прикладами запитів користувача.

2. Програму і методику випробувань розробленої системи (представити як додаток В до пояснювальної записки до кваліфікаційної роботи).

3. Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Для проведення випробувань необхідний ПК на якому встановлено Telegram.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає:

1. Перевірку відповідності технічних характеристик програми вимогам технічного завдання;
2. Перевірку ступеня виконання функціональних вимог до програми;
3. Методику проведення перевірок, що входять до переліку по 2 етапу випробувань.
4. Запуск додатку Telegram;

5. Знайти у додатку необхідного чат-бота
6. Почати спілкування з ботом;
7. Перевірити чи коректно працює бот, та виконує усі доступні функції.

Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

Тест 1

1. Перевірка виконання програми;
2. Перевірка роботи головного меню;
3. Поява усіх функцій головного меню(вибір комплектуючої для пошуку).

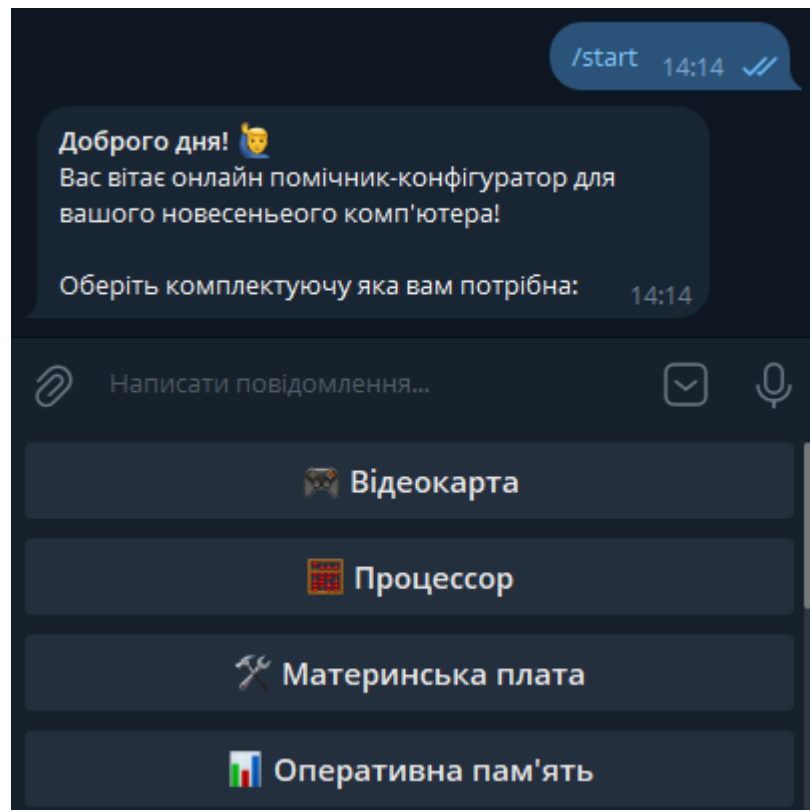


Рис. В.1 – Тест 1.

Тест вважається успішним якщо після вводу команди «/start» або натискання кнопки «Почати» (якщо це перша взаємодія з ботом) користувач отримає привітально повідомлення та буде виведено усі функції(вибір комплектуючої для пошуку).

Тест 2

1. Перевірка виконання програми;
2. Перевірка роботи функції пошуку відеокарти;
3. Отримання списку відеокарт найкращих за заданим критерієм у вигляді файлу, та повідомлення.

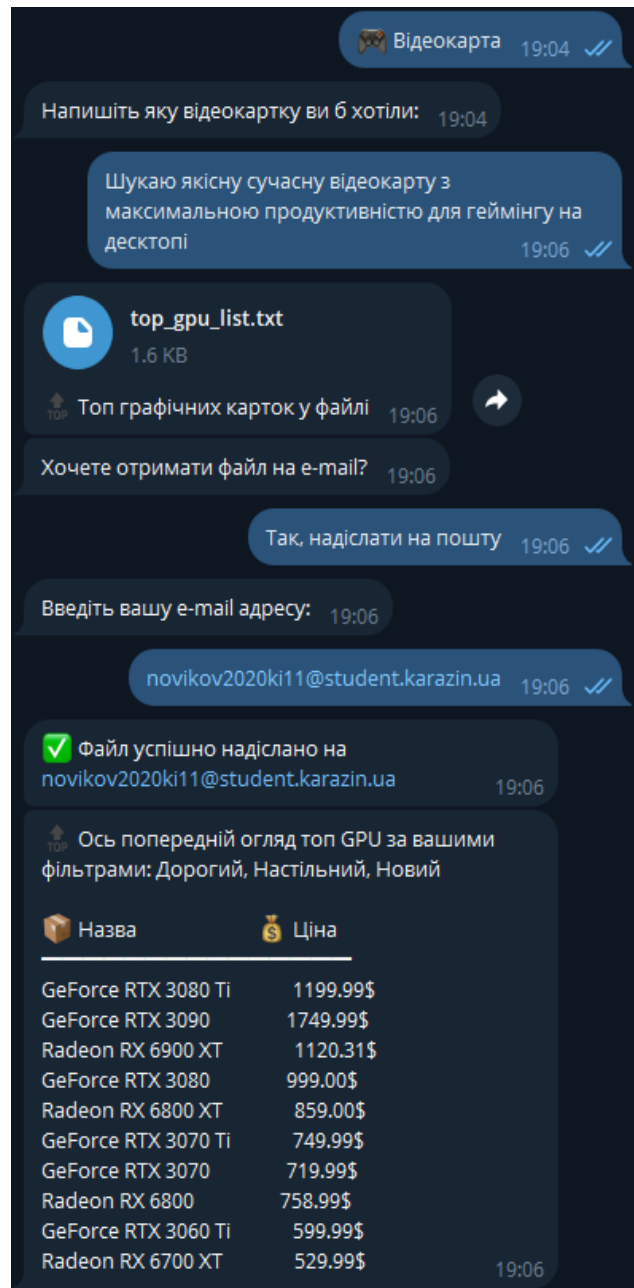


Рис. В.2 – Тест 2.

Тест вважається успішним якщо користувач отримує файл зі списком комплектуючих які підходять його запиту, отримує питання щодо надсилання цього файлу на пошту та скорочений список у вигляді повідомлення у месенджері.

Тест 3

1. Перевірка виконання програми;
2. Перевірка роботи функції перегляду списку порталів де можна замовити комплектуючі;
3. Отримання списку сайтів на яких можна замовити комплектуючі.

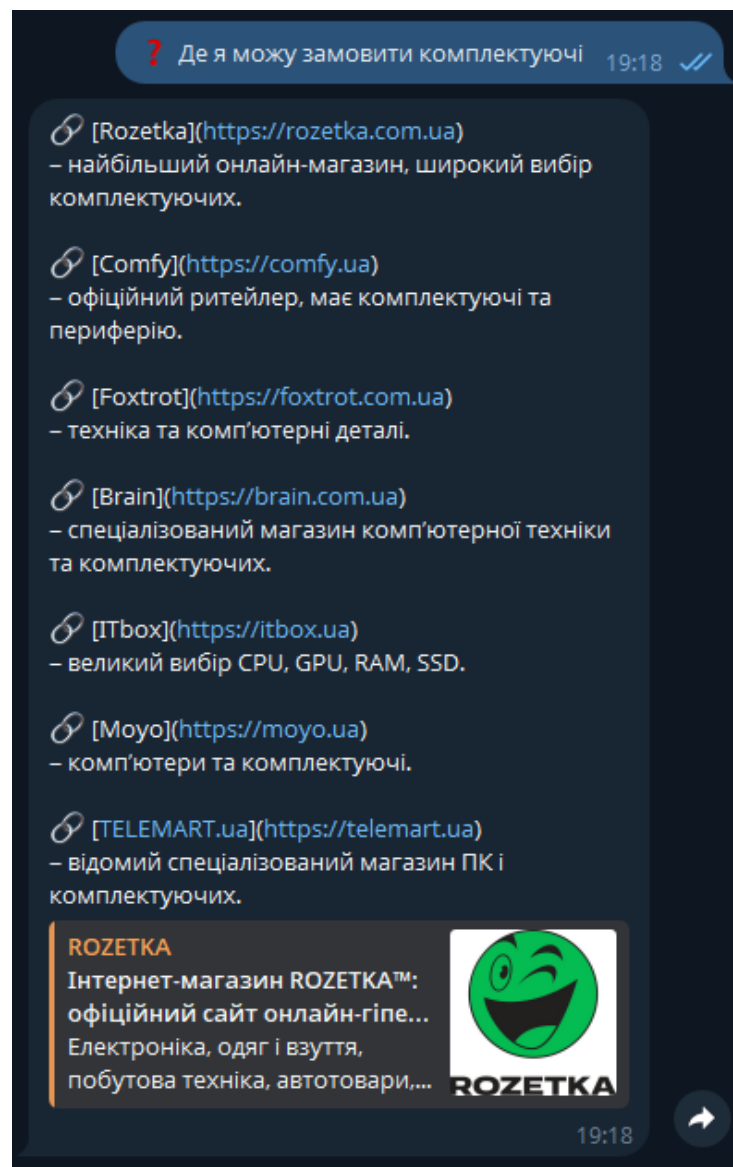


Рис. В.3 – Тест 3.

Тест вважається успішним якщо користувач отримує повідомлення із списком джерел на яких можна замовити комплектуючі.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КІ-61, Новіков О. Е.

