МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА
Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту
"_____" _____ 2023 р.
Зав. кафедри, к.т.н., доцент
_____ Ліщук Р.І.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Інформаційна технологія інтеграції чат-ботів у web-додатки

Виконав – студент 2 курсу, групи 21м-кн
освітньої програми "Комп'ютерні науки"
спеціальності 122 Комп'ютерні науки
Короташ Олександр Вікторович
Науковий керівник – д.т.н., професор
Кучерук Володимир Юрійович

Умань – 2023

Реферат

на кваліфікаційну роботу «Інформаційна технологія підвищення стійкості текстових паролів»
студента факультету економіки та підприємництва
Уманського національного університету садівництва
Короташа Олександра Вікторовича

Структура та обсяг роботи - пояснювальна записка до кваліфікаційної роботи магістра складається з 3 розділів, містить 66 сторінок, 25 рисунків, 5 додатків, 41 літературних джерел.

Кваліфікаційна робота магістра присвячена створенню чат-бота для організації дистанційного навчання. У роботі проаналізовані особливості існуючих підходів до створення чат-ботів, а саме: обґрунтовано, що чат-боти є одним з найбільш перспективних напрямків в найближчому майбутньому і є актуальною темою для багатьох наукових досліджень - обумовлено двома факторами, активне використання месенджерів і розвиток технологій штучного інтелекту; розроблена класифікація чат-ботів (за типами користувачів, видами інтерфейсів, способам доступу до боту, призначенням, принципом роботи); обґрунтована перевага нейронних мереж для створення чат-ботів, які працюють на основі навчання; описана модель взаємодії чат-ботів з платформами месенджерів через Webhook.

Проведено дослідження технологій для побудови чат-бота. Виявлено, що обмін повідомлень у чат-боті відбувається шляхом клієнт-серверної архітектури. Всі необхідні дані передаються по HTTP протоколу за допомогою формату даних JSON. Для чат-бота продуманий функціонал для обміну інформації між викладачем та студентом. Завдяки тому, що дистанційне навчання нині відбувається з використанням веб-сервісів, то викладач має змогу за допомогою бота надсилати посилання на запис лекції, на документи або на певні матеріали, адже це має ряд переваг. Але також розглянутий другий варіант функціоналу, обміну інформації за допомогою звичайних файлів. Таку можливість продемонстровано з боку студента. Також підключена база даних

MongoDB, як можливість для викладача зберігати велику кількість посилань у разі збільшення кількості студентів, або предметів. Результатом проведеного дослідження стало створення чат-бота для організації дистанційного навчання. Система демонструє всі можливі шляхи реалізації організації дистанційного навчання на базі месенджера Telegram. Створений чат-бот розглядається як сучасна система, з можливістю розширення функціоналу.

Анотація

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

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

Annotation

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

Keywords: web application, chatbot, integration.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ САДІВНИЦТВА

Факультет економіки і підприємництва
Кафедра _____Інформаційних технологій_____
Освітня програма _____Комп'ютерні науки_____
Спеціальність_____122 Комп'ютерні науки_____
(код, назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ ___Ліщук Р.І._____
(підпис) (ініціали, прізвище)
"_____" _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студенту_____*Короташу Олександру Вікторовичу*_____
(прізвище, ім'я, по батькові)
1. Тема проекту *Інформаційна технологія підвищення стійкості текстових паролів*
керівник роботи__*Кучерук Володимир Юрійович, д.т.н., професор кафедри інформаційних технологій*_____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від "12"_червня_2023 року №_01-06/130
2. Строк подання студентом роботи_____8 пункт, 28.05.2023_____
3. Вихідні дані до роботи __*проаналізувати методи і засоби інтеграції чат-ботів у web-додатки. Розробити програмний продукт на основі проаналізованих методів*
4. Перелік питань, які потрібно розробити
_*Аналітичний огляд предметної області, алгоритмічне та програмне забезпечення засобу, тестування веб застосунку*
5. Перелік графічного матеріалу__4 таблиці, 25 рисунків, 5 додатків, 41 джерел
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ___07.09.2022_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми кваліфікаційної роботи	05.09.2022	
2.	Одержання індивідуального завдання	07.09.2022	
3.	Складання календарного плану написання кваліфікаційної роботи	08.09.2022	
4.	Підготовка до виконання кваліфікаційної роботи - підбір та вивчення літератури - участь у виконанні науково-дослідних робіт - виконання курсових робіт - інші заходи	Впродовж I семестру	
5.	Уточнення теми кваліфікаційної роботи та календарного плану-графіка, виходячи зі специфіки базового підприємства, установи	До початку II семестру	
6.	Підготовка: - першого розділу - другого розділу - третього розділу - висновків	 09.02.2023 23.03.2023 26.04.2023 10.05.2023	
7.	Доопрацювання кваліфікаційної роботи з урахуванням зауважень наукового керівника	18.05.2023	
8.	Написання та оформлення роботи в остаточному варіанті	25.05.2023	
9.	Перевірка роботи на академічний плагіат	12.06.2023	
10.	Одержання відгуку наукового керівника	13.06.2023	
11.	Попередній захист роботи на кафедрі	14.06.2023	
12.	Затвердження роботи завідувачем кафедри	15.06.2023	
13.	Одержання рецензії	16.06.2023	
14.	Захист кваліфікаційної роботи	19-21.06.2023	

Студент_____ ___**Короташ О.В.**_____
(підпис) (прізвище та ініціали)

Керівник роботи _____ ____**Кучерук В.Ю.**_____
(підпис) (прізвище та ініціали)

ЗМІСТ

Перелік умовних позначень, скорочень і термінів …… 7
ВСТУП 9
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТЕХНОЛОГІЙ ВИКОРИСТАННЯ ЧАТ-БОТІВ...……...…...……12
1.1. Аналіз характеристик чат-ботів 12
1.2. Платформи впровадження чат-ботів. 15
1.3. Постановка задачі ……..20
РОЗДІЛ 2. ІНТЕЛЕКТУАЛЬНІ ТЕХНОЛОГІІ В РОЗРОБЦІ ЧАТ-БОТУ 21
2.1. Нейронні мережі для навчання чат-ботів 22
2.2. Модель взаємодії чат ботів з платформою месенджера 28
2.3. Інструментальні засоби для створення та аналізу чат-ботів. 29
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЧАТ-БОТА 33
3.1. Засоби створення чат-бота 33
3.2. Проєктування функціоналу чат-бота 44
3.3. Створення та тестування чат-бота 45
ВИСНОВКИ 55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 57
ДОДАТОК А. Лістинг блоку реєстрації 61
ДОДАТОК Б. Алгоритм роботи чат-блоку 63
ДОДАТОК В. Зв’язки в базі даних 64
ДОДАТОК Г. Приклад роботи з чат-ботом 65
ДОДАТОК Д. Видалення роботи з чат-бота.....66

Перелік умовних позначень, скорочень і термінів

API	application programming interface
NLP	Natural Language Processing
NLU	Natural-language understanding
BI	Business Intelligence
LSTM	Long Short-Term Memory
SMS	Short Message Service
USSD	Unstructured Supplementary Service Data
HTTP	Hyper Text Transfer Protocol
ІІІ	ІШтучний Інтеллект
HTTPS	HyperText Transfer Protocol Secure
WS	Web Socket
WSS	WebSockets Secure
UDP	User Datagram Protocol
GUI	Graphical user interface
JSON	JavaScript Object Notation
XML	EXtensible Markup Language
UTF	Unicode Transformation Format
ASCII	American Standard Code for Information Interchange
SSL	Secure Sockets Layer
TLS	Transport Layer Security
BSON	Binary JavaScript Object Notation
URL	Uniform Resource Locator

ВСТУП

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

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

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

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

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

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

- гнучкість вибору та платформи та матеріалів;
- простота доступу;
- економія часу та коштів;

- нові знайомства з іншомовними студентами.

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

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

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

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

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

Створення інформаційного чат-бота це одне із найкращих рішень для власників освітніх ресурсів. Адже саме інформаційним чат-бот вважається, якщо він містить, опрацьовує та видає інформацію, а для того, щоб ним користуватись необхідний лише встановлений месенджер. За допомогою чат-боту можна швидко надіслати матеріали, посилання на лекцію тощо. Натомість користувачі зможуть швидко отримати розклад, контактні дані викладача, літературу або всю іншу інформацію яку міститиме чат-бот у будь-якому месенджері. Для того, щоб запустити бота необхідно лише перейти за

посиланням або знайти його по назві через пошук і розпочати діалог як зі звичайним користувачем.

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

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

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

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

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

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

Наукова новизна отриманих результатів магістерської роботи полягає у тому, що - удосконалено методи розробки та інтеграції чат-боту за рахунок новітніх технологій, штучного інтелекту та API.

Практичне значення одержаних результатів магістерської роботи: розроблено чат-бот з функціоналом який надає можливість дистанційного навчання на базі месенджеру Telegram. Цей програмний додаток 12 демонструє всі можливі методи взаємодії, поміж яких власники освітніх ресурсів можуть обрати найзручніший та використовувати для організації дистанційного навчання.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТЕХНОЛОГІЙ ВИКОРИСТАННЯ ЧАТ-БОТІВ

1.1. Аналіз характеристик чат-ботів

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

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

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

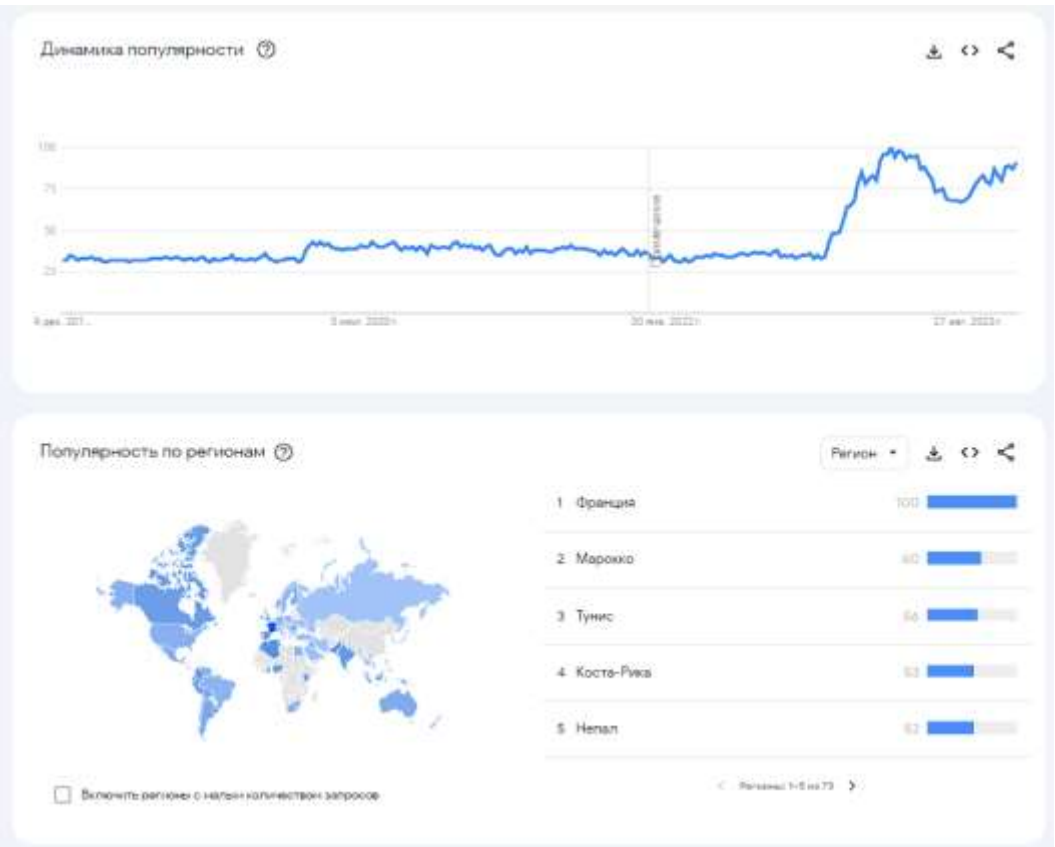


Рисунок 1.1 - Статистика популярності чат-ботів у світі

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

Але найпоширенішими залишається поділ з точки зору реалізації:

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

Бізнес класифікація:

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

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

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

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

Простий бот зазвичай будується на базі кнопок, тому він ідеально підходить для опитування, підтримки продажів і практично для будь-якого простого завдання автоматизації процесів, де сценарії спілкування чітко визначені.

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

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

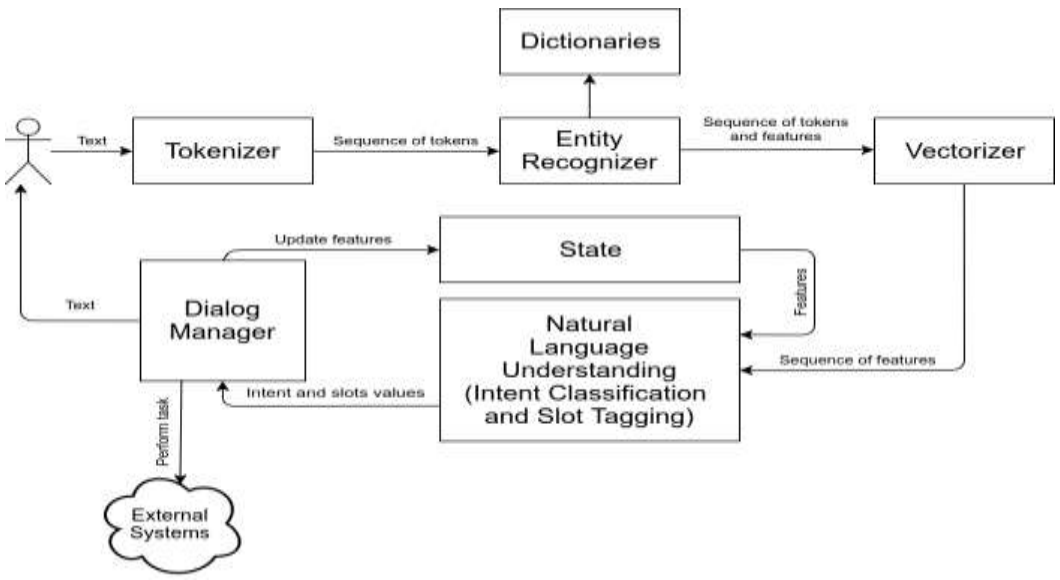


Рисунок 1.2 - Схема роботи чат-бота з підтримкою штучного інтелекту

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

- отримання запиту від клієнта;

- розбір запиту, розуміння висловлювання і визначення намірів клієнта в контексті його бізнес-кейсу;

- виконання дій згідно заздалегідь визначеним сценарієм;
- генерація відповіді на природній мові;
- надсилання відповіді клієнту.

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

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

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

1.2. Платформи впровадження чат-ботів

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

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

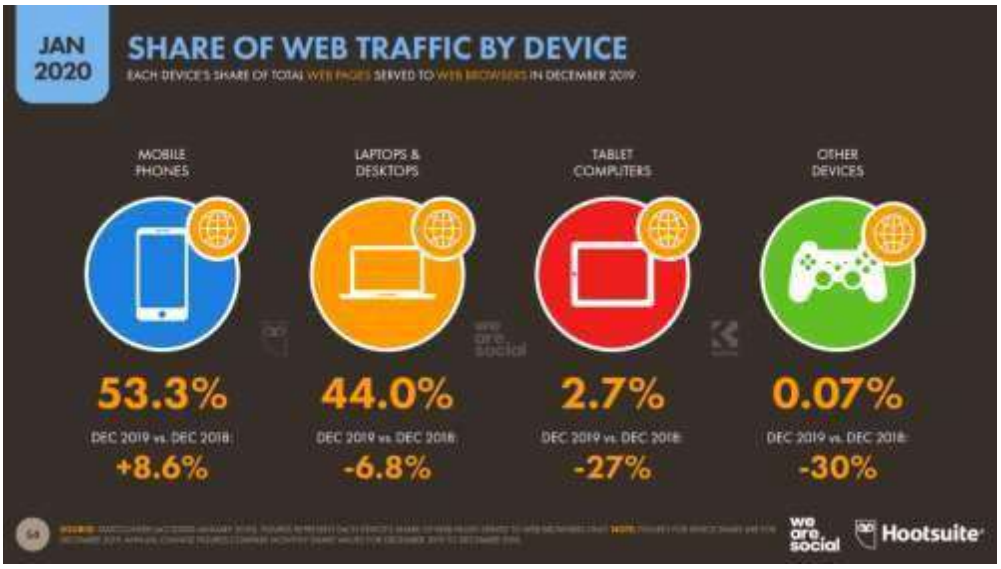


Рисунок 1.3 - Збільшення частки мобільного трафіку

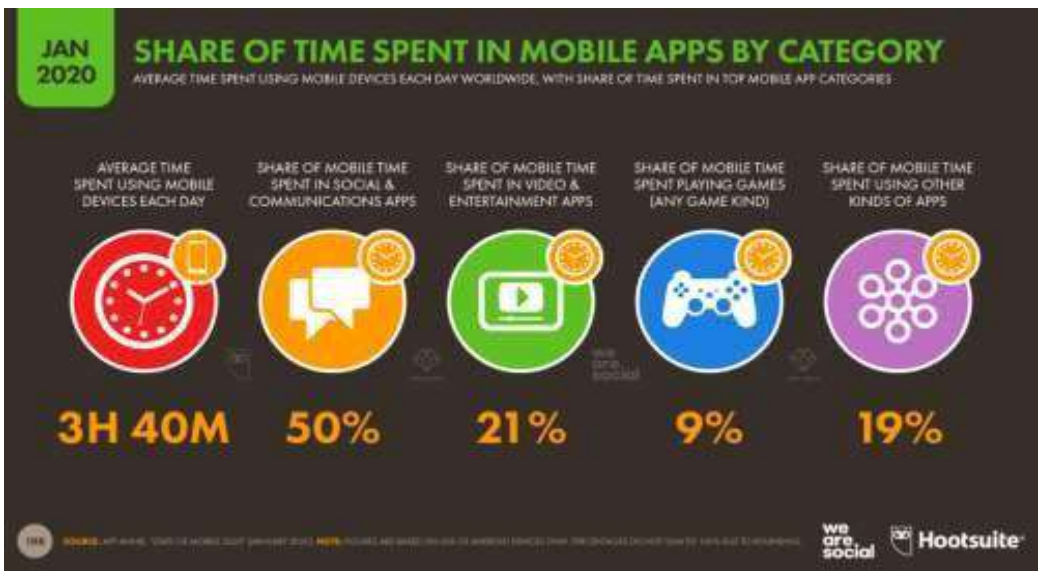


Рисунок 1.4 - Обсяг витраченого часу в інтернеті за добу

Тому, з’ясувавши, що люди більшу частину свого часу проводять за комунікацією у месенджерах, необхідно зробити аналіз найпопулярніших платформ для впровадження чат-бота. Отримавши статистичні дані із сервісу Statista.com за період січня 2021 року, дізнались що першим лідером месенджерів у світі є WhatsApp [6]. Кількість користувачів, що встановили програму більше ніж двох мільярдів у світі (рис. 1.5).

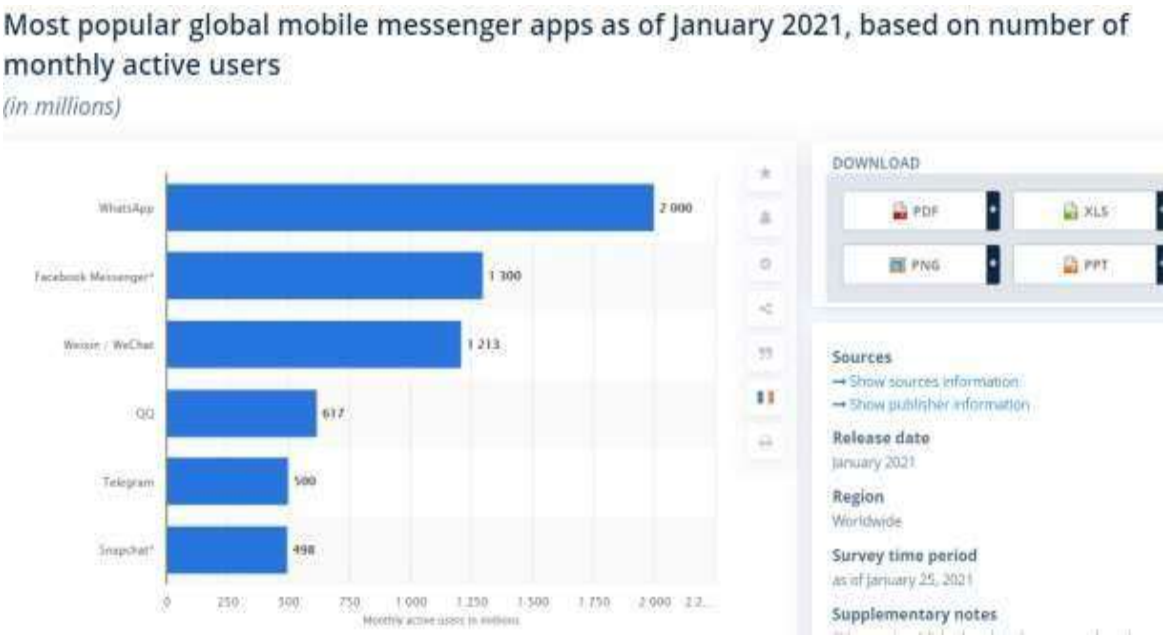


Рисунок 1.5 - Статистика популярних месенджерів у світі

На період 2023 року популярними месенджерами стали**:**

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

Але також, зібравши статистичні дані щодо найбільш популярних месенджерів в Україні, із сервісу Statista.com здалося зробити висновок, що Viber був найпопулярнішим месенджером в Google Play Store в Україні, оскільки в червні 2023 року він охопив майже 98 відсотків користувачів смартфонів, що працюють на базі Android. Facebook Messenger і Telegram мали другий і третій показник охоплення, виміряні приблизно 75 і 74 відсотками відповідно. Але отримавши статистику із Google Trends можна зробити висновок, що нині Telegram в Україні займає перші позиції (рис. 1.6).

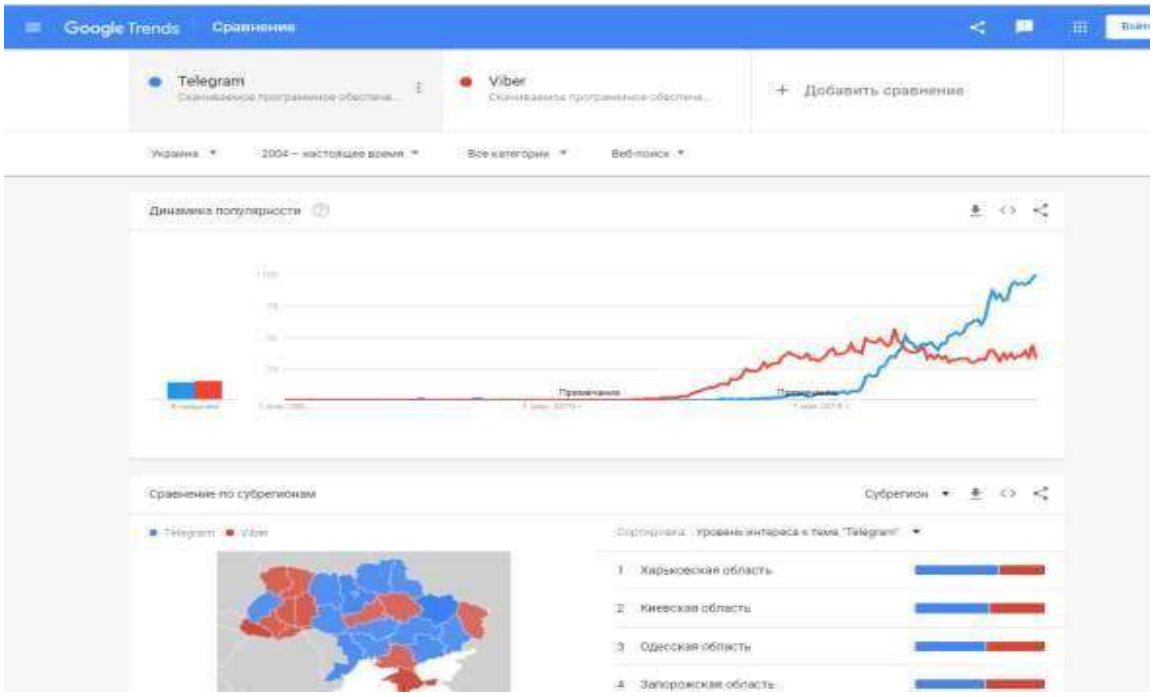


Рисунок 1.6 - Статистика найпопулярніших месенджерів в Україні

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

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

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

З основних можна перерахувати:

- спілкуватись з користувачами по всьому світу, та здійснювати аудіо та відео-дзвінки;
- відправляти пересилати і отримувати файли різних форматів;
- створювати групові та засекречені чати;
- знайти людину за псевдонімом, не знаючи номера телефону;
- відправляти знайдені в пошукових системах фото без збереження в

пам’яті пристрою;

- користуватись хмарним сховищем даних. Будь-які файли, обсяг яких не обмежені, зберігаються на віддаленому сервері;

- можливість безплатно та без інтернету слухати музику. Для цього спеціально був розроблений медіа плеєр.

Месенджер має низку переваг. Серед основних:

- висока швидкість роботи, не дивлячись на розташування серверів в різних частинах планети, передача повідомлення займе частки секунди;

- автоматичне додавання списку користувачів сервісу з телефонної книги;

- простий початок роботи, для реєстрації потрібен тільки номер телефону;

- всі функції та можливості безплатні та без реклами;

- інноваційність, сервіс постійно оновлюється, можливості розширюються як з урахуванням пропозицій конкурентів, так і власними розробками, які не мають аналогів;

- високий рівень безпеки і захисту від злому;

- необмеженість, кількість відправлених в день повідомлень, обсяги завантажуваних в хмару файлів, часу використання;

- сумісність з усіма основними операційними системами, включаючи мобільні;

- Конфіденційність, усі чати, які існують у месенджері зашифровані.

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

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

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

Тому така безпека передачі даних і привернула увагу «невидимого Інтернету» до месенджера. Це дозволило використовувати Telegram як кібер-

злочинцям, такі звичайним користувачам, які, отримали одну з кращих систем захисту даних і простий інтерфейс. Ось чому у деяких країнах цей месенджер знаходиться під забороною.

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

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

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

1.3 Постановка задачі

Ціллю магістерської кваліфікаційної роботи є розробка методів покращення інтеграції чат-ботів. Для цього потрібно здійснити: огляд стану роботи доступних чат-ботів на сайтах сайтів; розглянути та проаналізувати стандарти та методи інтеграції чат-ботів; створити чат-бот і досягти високого рівня доступності та функціональності шляхом оглянутих методів.

Практичне значення полягатиме в розробці чат-бот з функціоналом який надає можливість дистанційного навчання на базі месенджеру Telegram. Цей програмний додаток демонструє всі можливі методи взаємодії, поміж яких власники освітніх ресурсів можуть обрати найзручніший та використовувати для організації дистанційного навчання.

РОЗДІЛ 2. ІНТЕЛЕКТУАЛЬНІ ТЕХНОЛОГІЇ В РОЗРОБЦІ ЧАТ-БОТУ

2.1. Нейронні мережі для навчання чат-ботів

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

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

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

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

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

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

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

В сучасних месенджерах чат-боти є універсальним засобом для вирішення різноманітних завдань. Бот (англ. Bot) - спеціальна програма, що виконує автоматично або за заданим розкладом будь-які дії через інтерфейси, призначені для людей. Зазвичай боти призначаються для виконання одноманітної і повторюваної роботи, з максимально можливою швидкістю.

Вони застосовуються за необхідності швидшої реакції, ніж у людини. Це ігрові боти, боти для Інтернет аукціонів тощо. Іноді їх застосовують для імітації дій людини в спілкуванні, тоді їх називають чат-ботами.

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

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

Таблиця 2.1 - Класифікація чат-ботів

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

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

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

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

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

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

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

Сьогодні користувачі мобільних пристроїв не хочуть встановлювати багато програм. 72% користувачів використовують до 7 додатків на день [9].

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

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

Навчальні боти розроблені на основі рішень штучного інтелекту, які використовують обробку природної мови та машинне навчання під час побудови розмов. Вони працюють на основі аналізу діалогу для подальшого вдосконалення своїх комунікативних навичок. Такі боти здатні обробляти природну мову і правильно відповідати на запитання.

Нейронні мережі найчастіше використовуються для вирішення завдань машинного навчання. Нейронна мережа - це математична модель, програмна або апаратна реалізація якої нагадує організацію та функціонування біологічної нервової системи. Нейронні мережі здатні проводити глибокий аналіз оброблених даних, виявляти складні нелінійні залежності та неоднозначні зв’язки. Штучний нейрон - це структура, яка приймає сигнали, перетворює їх і

передає іншим нейронам. Отже, штучна нейронна мережа складається з групи штучних нейронів, які взаємодіють один з одним шляхом передачі та перетворення сигналів [4]. Незважаючи на велику різноманітність варіантів нейронної мережі, всі вони мають спільні характеристики. на рис. 2.1 зображено узагальнену модель штучного нейрона.

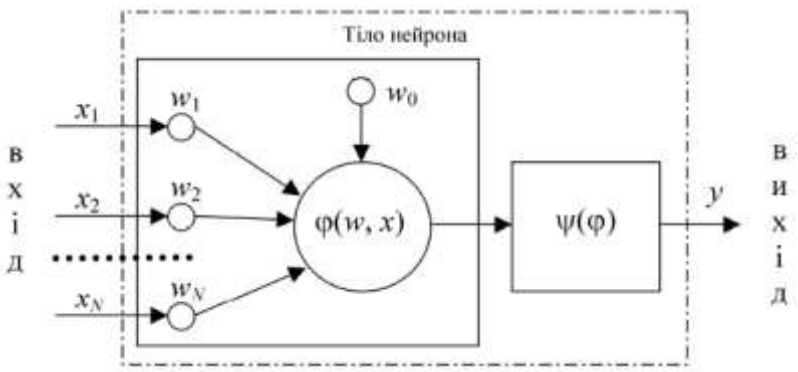


Рисунок 2.1 - Модель штучного нейрона

Штучний нейрон (*формальний нейрон*, *нейроподібний елемент*) – це примітивний обчислювальний пристрій (або його модель), що має кілька входів і один вихід, і є основним обчислювальним елементом нейронної мережі.

На вхід одношарового нейрона надходить вхідний вектор – набір вхідних сигналів $x = \{x_j\}, j = 1,2, \dots N$ де N - кількість входів.

Кожний вхідний сигнал x_j зважується (масштабується певним чином) відносно зіставленої йому ваги зв'язку (вагового коефіцієнта) ω_j, яка моделює перетворення сигналу у синапсі (міжнейронному контакті). Зважена сума *net* отримується як:

$$net = \sum_j^n x_j \omega_j.$$

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

рішення.

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

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

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

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

Користувач: Де я навчаюсь?

Чат-бот: Ти навчаєшся в УНУС.

Користувач: На якому факультеті я навчаюсь?

Чат-бот: Я не знаю.

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

Користувач: Де навчається мій друг?

Чат-бот: Твій друг навчається в КПІ.

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

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

2.2. Модель взаємодії чат ботів з платформою месенджера

Зростаюча популярність чат-ботів призвела до появи багатьох інструментів [2; 5; 11]. Чат-боти можна створювати з нуля, використовуючи різні мови програмування. В основному для цих цілей використовуються серверні мови програмування: Python, Ruby, Node.JS, PHP. Також є багато готових рішень, які допоможуть створити чат-ботів і навчити їх навичкам спілкування без програмування. У загальному вигляді модель функціонування чат-бота включає три елементи (рис. 2.2).

Backend - бот, який отримує інформацію, обробляє її та відправляє назад.

Інтерфейсом є клієнтська платформа бота, якою може бути будь-який популярний месенджер, такий як Facebook Messenger, Telegram, Skype, Viber, канали мобільного зв’язку, такі як SMS, USSD, соціальні мережі або вбудований ресурс чату, який дозволяє отримувати та надсилати повідомлення в режимі реального часу. Від правильного вибору платформи залежить залучення клієнтів, можливість підвищення лояльності та винагорода від впровадження розробок; Вебхук - це посилання на основі URL-адреси, яке з’єднує бота (сервер) і платформу чату (інтерфейс). Вебхуки дозволяють безпечно надсилати й отримувати повідомлення за допомогою стандартних запитів HTTP. У всіх

сучасних месенджерах є докладна інструкція, як підключити бота і сам сервіс.

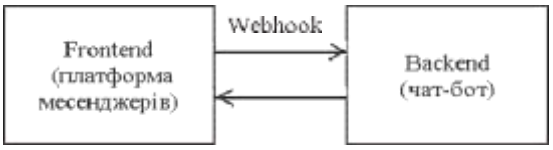


Рисунок 2.2 - Модель взаємодії чат-бота з платформою

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

2.3. Інструментальні засоби для створення та аналізу чат-ботів

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

До найбільш поширених фреймворків створення чат-ботів відносяться: BotKit - це набір інструментів з відкритим кодом, розроблених для Node.js. Підходить як перша платформа для навчання та експериментування з чат-ботами. Botkit має сервіс Botkit Studio, який містить стандартний набір програм, базові бібліотеки та плагіни для розширення функціональності ботів, а також дозволяє створювати боти для Facebook Messenger, Slack і Cisco Spark; Claudia Bot Builder - це програма для створення ботів, яку можна використовувати в AWS Lambda. Він містить бібліотеку Claudia.js, яка допомагає створювати ботів для Facebook Messenger, Telegram, Skype, Slack slash, Twilio, Kik і GroupMe; Bottr.me - це найпростіший фреймворк Node.js для створення та використання персональних чат-ботів. Ця послуга дозволяє тестувати створені вами продукти.

Платформа IBM Watson, безсумнівно, є лідером у розробці рішень для

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

Таблиця 2.2 - Характеристика платформ для розроблення чат-ботів

Назва	Короткий опис	Клієнтська платформа	Інтеграція	Безкоштовна версія
SnatchBot	Потужна, гнучка і проста у використанні платформа, що забезпечує швидку інтелектуальну розробку чат-ботів для всіх сфер застосування. Діалог на основі алгоритмів розпізнавання людської мови NLP або тексту. Підтримка 6 мов. Машинне навчання	Багатоплатформний	Twilio	Є
Converse.ai	Графічний користувальницький інтерфейс. Інтелектуальний аналіз діалогів за допомогою зворотного зв'язку через опитування. Перевірка і обробка контексту. Технічна підтримка клієнтів	Багатоплатформний	SalesForce, Stripe, PayPal, Clear Bit, Twilio, Airtable, HubSpot, FlightStats	Є (для початкового рівня)
ChattyPeople	Створення ботів для спілкування з клієнтами та електронної комерції. Можливість приймати замовлення від месенджера і з коментарів, регулювати пропозиції і здійснювати операції	Facebook	E-commerce, які підтримують open cart	Є для початкового рівня
Smooch	Однаковий інтерфейс для всіх платформ. Інтелектуальна розмова, на основі інтеграції з перекладачами, NLP і AI-двигунами. Управління учасниками розмови. Використання контенту для кращого розуміння намірів користувача і його автоматичного інформування	Багатоплатформний	Stripe Twilio	Є (до 10 тис. користувачів в місяць). Пробний період
Wit.ai	Спілкування на основі алгоритмів розпізнавання людської мови NLP або тексту. Підтримка 11 мов. Можливість побудови для певного тексту безлічі варіантів питань. Є вбудовані шаблони і плагіни для інтеграції текстових повідомлень, фото тощо. Можливість навчання бота певний період	Facebook	Python, Node.js, Ruby	Є

Аналіз поведінки чат-ботів під час взаємодії з користувачами є важливою частиною їх використання. Моніторинг чат-бота передбачає постійну оптимізацію його поведінки та сценаріїв розмов на основі аналізу взаємодії користувача з ботом. Серед безкоштовних сервісів для аналізу ефективності чат-ботів Botmetric, Chatbase, Botanalytics, Dashbot, Botlytics та інші (табл. 2.3).

Також варто зазначити, що такі платформи, як Facebook, Telegram тощо, мають власні інструменти аналітики.

Створення чат-бота вимагає значних внесків на всіх етапах розробки. Основні завдання створення бота - навчити бота розуміти контекст запитів користувачів (26%) і будувати розмовні моделі (16%). Якщо розглядати навички, необхідні при створенні чат-бота, то їх можна визначити так: Business Skills - 11%, Product Owner - 11%, Creative Technical Knowledge - 14%, Design - 16%, Coding - 27% [7].

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

Таблиця 2.3 - Характеристика сервісів для аналізу взаємодії користувачів з ботом

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

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

Етап 1. Визначення вимог:

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

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

Етап 2. Розроблення чат-бота:

- створити комунікаційну архітектуру, яка дозволяє будувати ієрархії вмісту для спілкування;

- зібрати інформацію, необхідну для завершення розмови, щоб навчити робота розуміти різні варіації однієї концепції та давати правильну відповідь, це можна зробити шляхом проведення опитувань серед цільової аудиторії;

- встановлюйте сценарії діалогу відповідно до власних сценаріїв або відповідно до умов, за яких бот відповідає певним чином;

- створюйте контент і моделюйте сценарії діалогу, детально прораховуючи відповіді ботів, щоб спілкування виглядало природним;

- навчайте роботів, розробляючи чітко структуровані сценарії відповіді на основі великих баз даних, або використовуйте готові алгоритми навчання для інтерпретації відповідей за відсутності таких баз даних.

Етап 3. Підключення:

- інтегрувати платформу з іншими сервісами, а якщо платформа вже інтегрована з іншими системами, подібну схему інтеграції можна застосувати до чат-ботів;

- виконати тестування роботи чат-бота; активувати бота.

Етап 4. Аналіз та вдосконалення роботи:

- збирати та аналізувати інформацію про роботу бота;

- доопрацювати та вдосконалити поведінку чат-бота: включити нові запитання, покращити логіку та класифікацію понять, додати більше відповідей тощо.

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

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЧАТ-БОТА

3.1. Засоби створення чат-бота

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

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

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

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

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

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

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

Взаємодія між користувачем і ботом виглядає наступним чином:

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

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

Схема обміну даними між користувачем і Telegram ботом наведена на рис. 3.1.

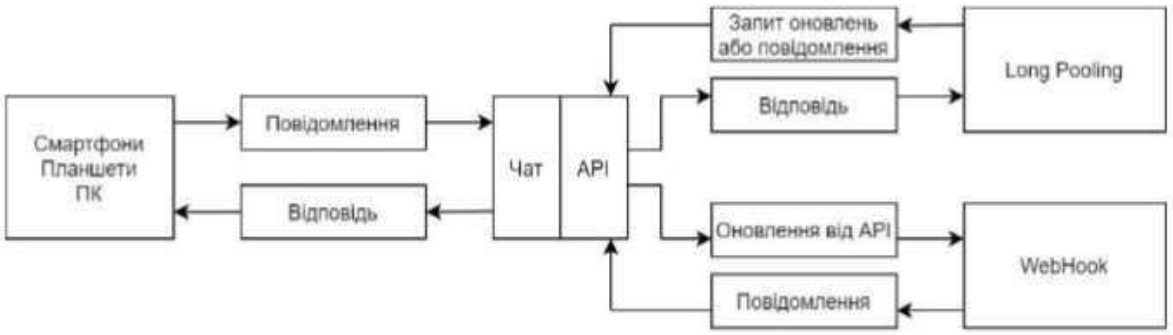


Рисунок 3.1 - Схема обміну даними між користувачем і Telegram ботом

Telegram Bot API

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

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

- криптографічний рівень, з його використанням визначається метод для шифрування повідомлень перед передачею через транспортний протокол; - транспортний рівень, з його використанням визначається спосіб передачі між клієнтом та сервером повідомлень, за допомогою наявних мережевих протоколів (HTTP, HTTPS, WS, WSS, TCP, UDP).

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

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

- за допомогою запитів;

- за допомогою вебхуків.

Під оновленням мається на увазі дія, що була виконана над ботом - наприклад, отримання запиту від користувача. Вхідні оновлення зберігаються до того моменту, поки сервер не обробить їх, але не більш ніж 24 години.

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

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

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

Telegram Bot API надає декілька доступних методів для роботи:

- getUpdates, метод використовується для отримування оновлення за допомогою Long Pooling;

- setWebhook, метод, що необхідний для задання URL вебхука, на який ботбуде відправляти оновлення;

- getWebhookInfo, метод збирає інформацію щодо того чи іншого вебхука;

- sendMessage, метод використовується для відправлення текстових повідомлень.

Мова програмування Python

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

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

TelegramBotAPI, розрахованих на роботу безпосередньо з Telegram.

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

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

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

Широковживані вважаються такі модулі:

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

- Tkinter, модуль, за допомогою якого будуються додатки використанням графічного інтерфейсу користувача (GUI);

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

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

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

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

Формат даних JSON

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

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

JSON з самого початку був заснований на двох структурах даних:

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

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

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

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

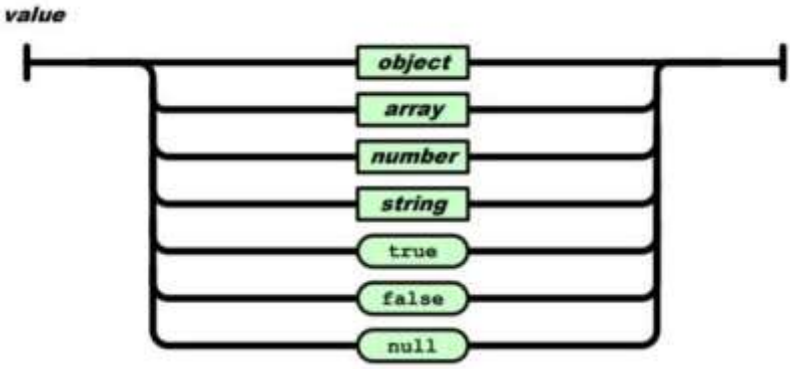


Рисунок 3.2 - Схематичне представлення значення JSON

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

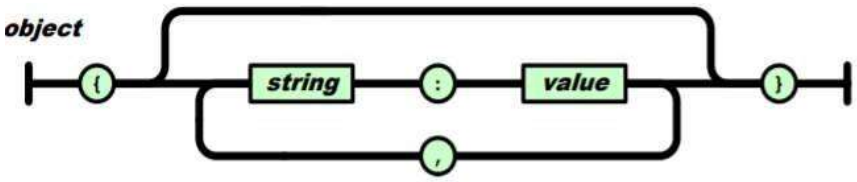


Рисунок 3.3 - Схематичне представлення об’єкта JSON

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

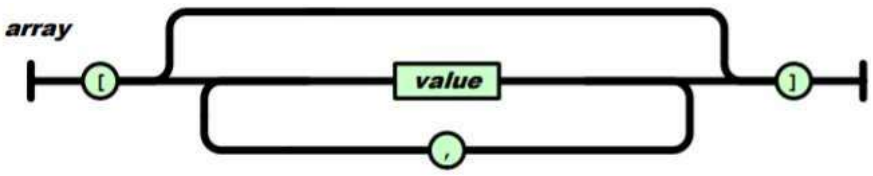


Рисунок 3.4 - Схематичне представлення масиву JSON

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

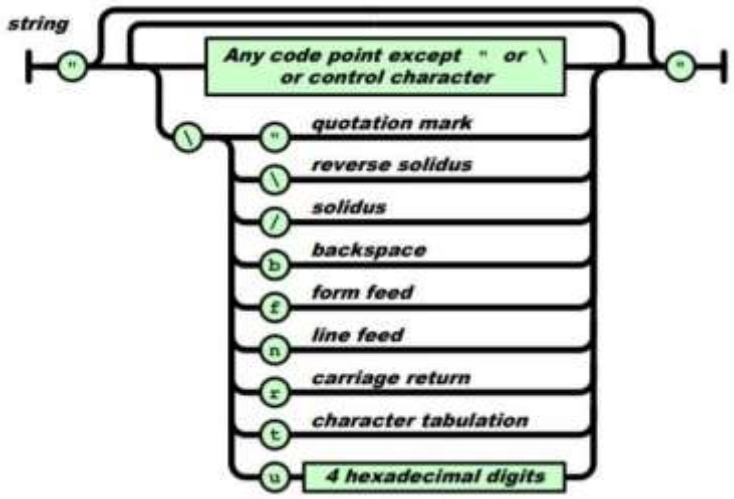


Рисунок 3.5 - Схематичне представлення рядка JSON

У структурі числа в JSON не використовуються восьмирічні або шістнадцятирічні формати, але число передається так, як в мовах програмування C або Java (рис. 3.6).

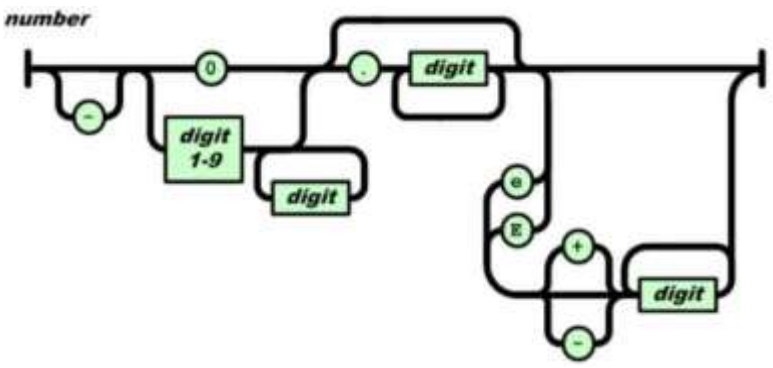


Рисунок 3.6 - Схематичне представлення числа JSON

HTTP та HTTPS запити

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

Сервери - це найчастіше комп’ютери в хмарі.

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

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

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

Метод GET використовується для отримання інформації від сервера по заданому URI (URI в HTTP). Запити клієнтів, що використовують метод GET повинні отримувати тільки дані і не повинні ніяк впливати на ці дані.

Для того, щоб відправити повідомлення, які завантажують дані на веб-сервер використовуються POST і PUT.

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

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

Запит HTTP протоколу (рис. 3.7) складається із указаних параметрів:

- метод (GET, PUT, POST), що описує дію, яку потрібно виконати

- ціль запиту, зазвичай URL-адреса, або абсолютний шлях протоколу, порту та домену;

- версія протоколу;

- заголовки.

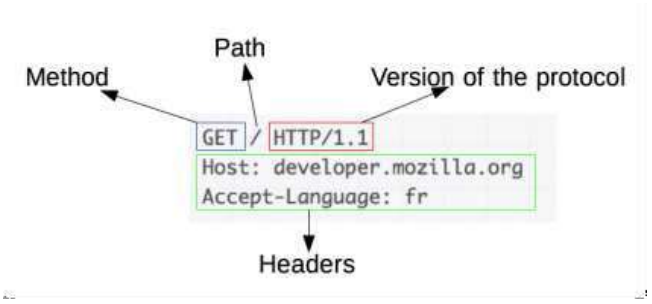


Рисунок 3.7 - Структура запиту HTTP протокол

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

- версію протоколу;

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

- текст статусу, короткий, інформаційний, текстовий опис коду стану, щоб допомогти людині зрозуміти повідомлення HTTP;

- заголовки;

- тіло (опціональне поле).

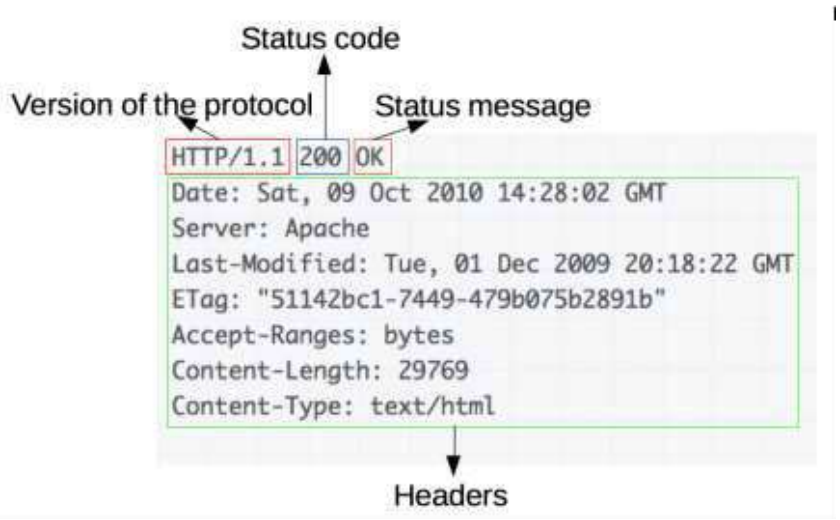


Рисунок 3.8 - Структура відповіді HTTP протокол

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

HTTPS (англ. HyperText Transfer Protocol Secure) - розширення HTTP, яке підтримує захист даних при транспортуванні за допомогою шифрування

інформації відповідно до стандартів SSL і TLS. Такий захист необхідний в комерційних ресурсах, де використовується конфіденційна інформація про персональні або платежів користувача.

База даних MongoDB

Оскільки, Telegram бота можна створити на будь-якій серверній мові програмування, так само до нього можна і під'єднати будь-яку базу даних. В залежності від завдання, і необхідної кількості зберігання інформації, на цю мить момент існує широкий вибір баз даних. Для створення чат-бота для організації дистанційного навчання, вибір був зроблений на користь MongoDB.

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

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

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

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

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

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

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

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

Система GridFS складається з двох колекцій. У першій колекції, яка називається files, зберігаються імена файлів, а також їх метадані, наприклад, розмір. А в іншій колекції, яка називається chunks, у вигляді невеликих сегментів зберігаються дані файлів, зазвичай сегментами по 256 Кб. Вся модель бази даних в MongoDB представлена на рис. 3.9.

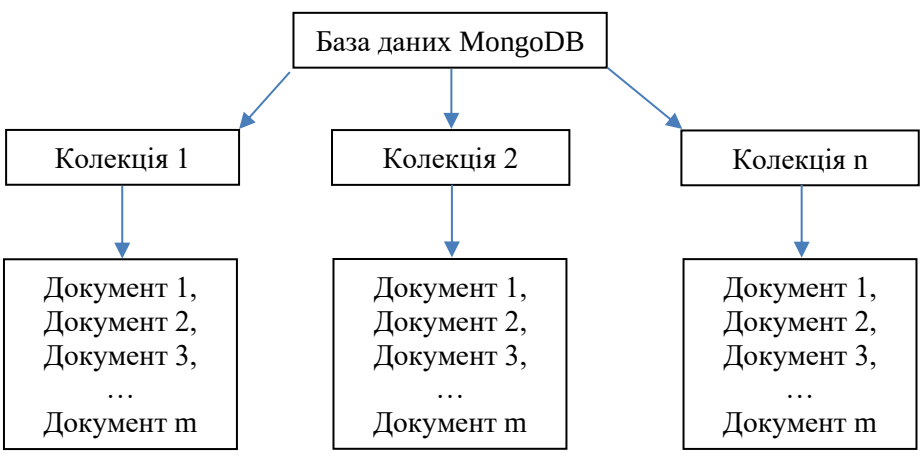


Рисунок 3.9 - Схематичне зображення моделі бази даних в MongoDB

3.2. Проєктування функціоналу чат-бота

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

Використання саме такого типу обміну даними має низку переваг:

- інформація в безпеці, при роботі в документі сервіс фіксує будь-яку зміну;

- доступ з будь-якого місця, почати працювати в документі можна з одного пристрою, а закінчити на іншому;

- історія змін, дає змогу повернутись до попереднього варіанту;

- спільна робота в документі, можливість певній категорії людей дозволити редагувати документ, а для інших залишити доступним лише перегляд;

- можливість залишати в роботі коментарі;

- економія місця, всі документи знаходяться у хмарному сховищі.

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

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

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

- для студентів;
- для викладача.

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

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

Різниця полягає в тому, що студенти завдяки боту можуть:

- отримати завдання;
- надіслати певні матеріали викладачу.

Викладач у свою чергу може:

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

3.3. Створення та тестування чат-бота

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

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

- необхідно написати унікальне ім’я нового бота, адже існування чат- ботів з однаковими іменами - неможливе, якщо бот з такою назвою вже існує, то BotFather запропонує вигадати інше ім’я;
- написати назву бота із закінченням bot у кінці;
- отримати маркер доступу (token) до HTTP API.

Маркер доступу - це унікальний ключ, що складається із набору

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

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

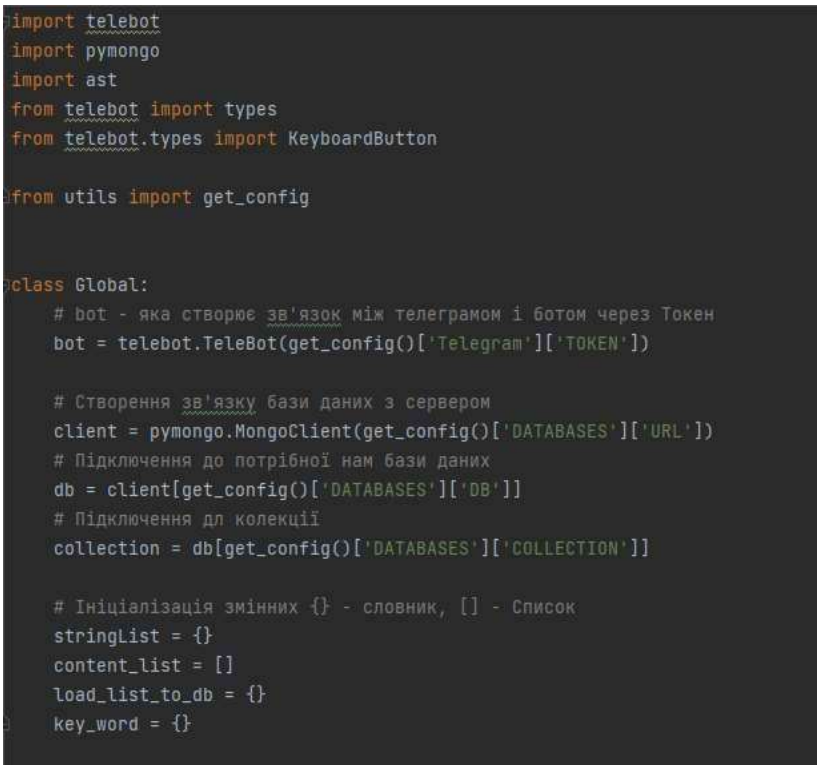
- setname - встановити нове ім’я;
- setdesctiption -змінити опис боту;
- setabouttext - змінити інформацію щодо бота;
- setuserpic - змінити інформацію у профілі бота;
- setcommands - змінити список команд;
- deletebot - видалити бота;
- token - згенерувати новий токен авторизації;
- revoke - анулювати токен.

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

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

Всі ці дані записуються кодом в окремому файлі, в якому також додається з’єднання із базою даних.

Для створення всіх можливих функцій чат-бота необхідно підключити декілька модулів та створити клас, у якому зберігаються всі з’єднання (рис. 3.10).



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

from utils import get_config


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

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

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

Рисунок 3.10 - Підключення до модулів та створення класу

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

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

Список - це вид змінної у якій відразу кілька значень. Значення можуть бути різних типів, наприклад в одному списку може бути як і int (ціле число) так і float (число з плаваючою точкою) та навіть string (рядок, літери). Всі дані в списку впорядковані індексацією. Більш детально всі можливі методи роботи зі списком наведено в додатку.

Словник - це невпорядкований список. Замість індексації у всіх даних є свій буквений ключ. Більш детально всі можливі методи роботи зі словником наведено в табл. 3.1.

Таблиця 3.1 - Методи роботи із словником

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

Для створення функціоналу бота використовувались методи stringList.items та stringList.clear (рис. 3.11). Функція дістає всі дані з бази даних і записує їх в змінні, метод append доповнює масив з даними, clear очищує всі дані з масиву перед записом.

PyMongo - це дистрибутив, що містить інструменти для роботи з MongoDB, і єрекомендованим способом роботи з цією базою даних на мові Python. Один примірник MongoDB може підтримувати кілька незалежних баз даних. Під час роботи з PyMongo надається доступ до баз даних, використовуючи стиль атрибута MongoClient. Як вже відомо колекція представляє собою групу документів , що зберігаються в MongoDB, і можна розглядати як приблизно еквівалент таблиці реляційної бази даних.

Для обробки повідомлень використовується функція message handler. Після прийому повідомлення від Telegram його необхідно опрацювати в залежності від того, що це за повідомлення: текст, файл, посилання, тощо. Це можливо реалізувати якщо використовувати безліч конструкцій if-then-else, проблеми автор бібліотеки pyTelegramBotAPI реалізував механізм handler, який використовує декоратори. Тобто у handler описується, в якому випадку необхідно виконувати тучи іншу функцію.



Рисунок 3.11 - Запис файлів у базу даних

Для реалізації функціоналу чат-бота використовувалось два види кнопок. Кожна кнопка використовується як самостійний об'єкт KeyboardButton або InlineKeyboardButton. Об’єкт KeyboardButton у даній реалізації використовувався для створення основних функціональних кнопок (рис. 3.12).

Кнопки InlineKeyboardButton діляться на три типи: URL-кнопки, Callback-кнопки і Switch-кнопки. Вбудована клавіатура являє собою об'єкт InlineKeyboardMarkup, а кожна така кнопка - це об'єкт InlineKeyboardButton. Для того щоб створити URL-кнопку, потрібно вказати значення параметрів text (текст на кнопці) URL (веб-адреса). В результаті бот відправить повідомлення із посиланням (рис. 3.13).

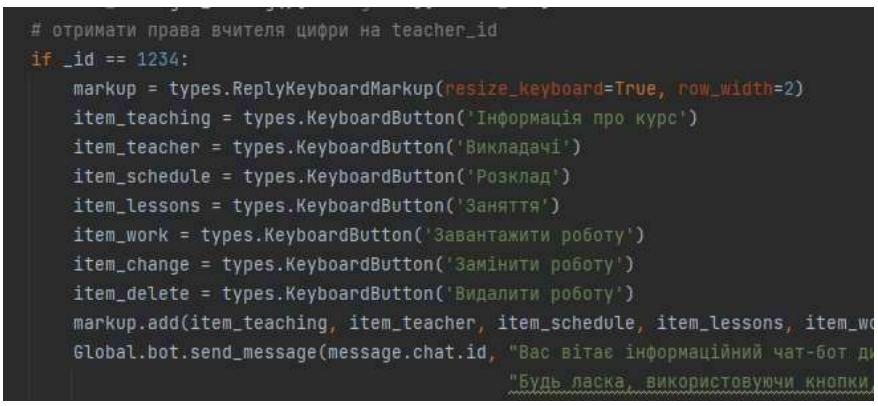


Рисунок 3.12 - Частина коду опису створення кнопок

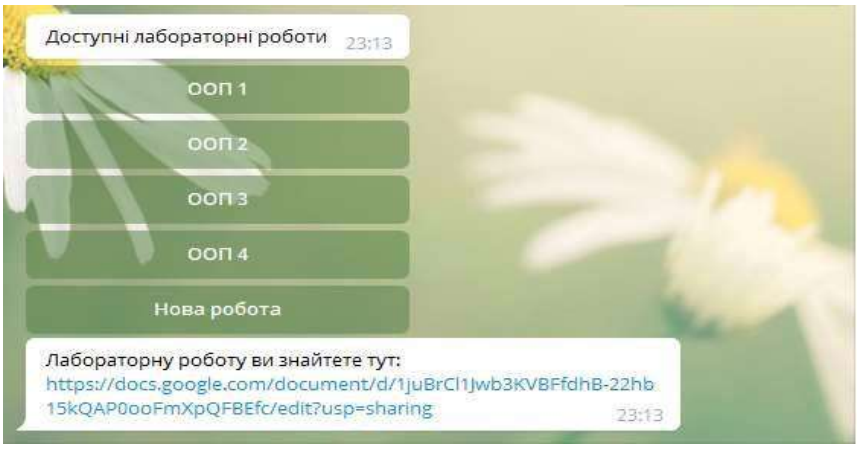


Рисунок 3.13 - Приклад роботи URL кнопок

Кнопки Callback дозволяють виконувати довільні дії після їх натискання. Все залежить від того, які параметри кожна кнопка в собі несе. Відповідно, всі натискання будуть приводити до відправки боту об'єкта CallbackQuery, який містить поле data, в якому написаний певний рядок, закладений у кнопку, а також об'єкт Message, якщо повідомлення надіслано ботом в звичайному режимі. В даному випадку вона завантажує посилання на матеріали (рис. 3.14).

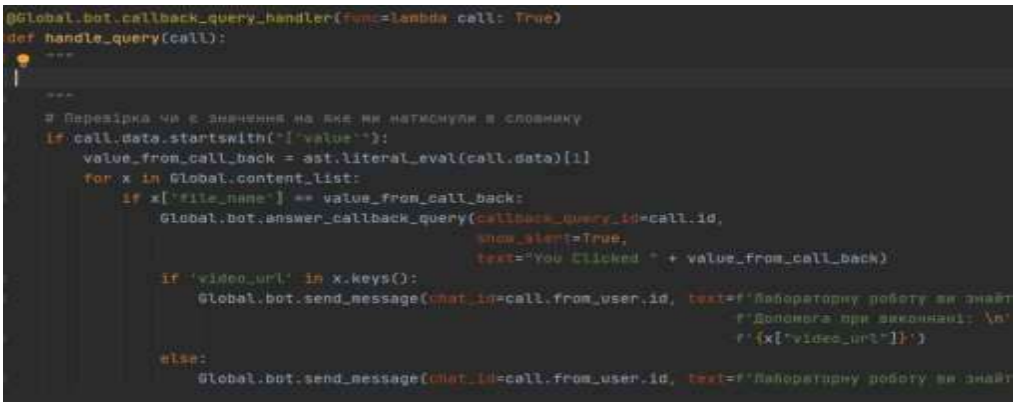
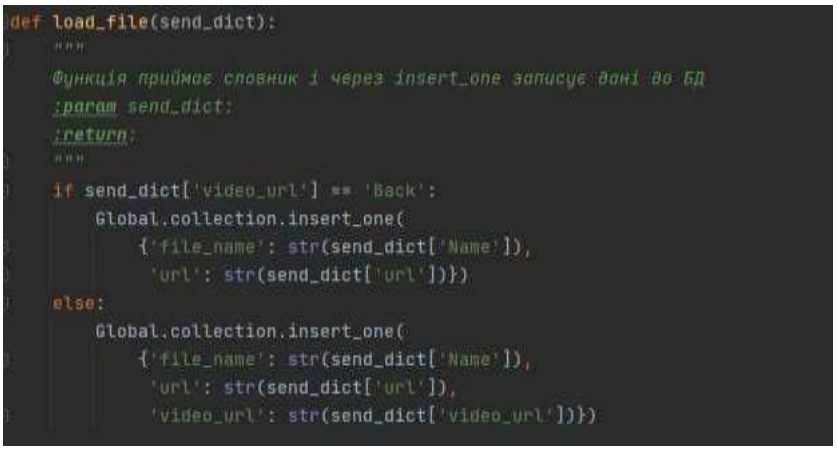


Рисунок 3.14 - Частина коду опису створення посилання для кнопки

Для взаємодії з базою даних було створено функції запису, заміни та видалення файлів. База даних створена для інтерфейсу викладача, тому вона лише зберігає назву, посилання та код об’єкта. Для того, щоб завантажити дані використовувалась функція load_file, яка приймає словник і через insert_one записує дані до бази даних (рис. 3.15). Аналогічно для заміни використовувались update_one, а для видалення delete_one.



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

Рисунок 3.15 - Частина коду для роботи з базою даних

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

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

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

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

Тестування боту відбувалось:

- на комп'ютері в якості інтерфейсу викладача: операційна система Windows, з використанням додатку Telegram Desktop.

- на смартфоні Samsung в якості інтерфейсу студента, операційна система

Android, з використанням додатку Telegram.

Перш за все проводилось тестування бота проводилось паралельно для обох інтерфейсів. Після приєднання до чату користувачу доступна тільки одна команда /start. Відправивши цю команду, у відповідь отримуємо основну інформацію про бота (рис. 3.16).

Після цього отримуємо можливість взаємодії з ботом за допомогою кнопок.

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



Рисунок 3.16 - Початок роботи із ботом

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

В результаті робота з’явилась у переліку занять у вигляді кнопки. Якщо натиснути на цю кнопку, отримаємо необхідне посилання Також посилання на цю роботу з’явилось у базі даних (рис. 3.17).

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



Рисунок 3.17 - Завантаження роботи до чат-боту за допомогою посилання

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

Якщо розглядати бота із інтерфейсу студента, то він відрізняється лише наявністю однієї кнопки «Завантажити роботу». За правилами Telegram користувач може надіслати файли формату: .PDF, .DOC, .MP4, .JPEG і обсяг файлу не повинен перевищувати 5 Мб. Якщо обсяг файлу буде більше користувач отримає повідомлення про помилку. Для того щоб відправити роботу, бот запропонує ввести назву роботи Коли студент надсилає роботу, викладач отримує її у спеціально створеному Telegram каналі (рис. 3.18).

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

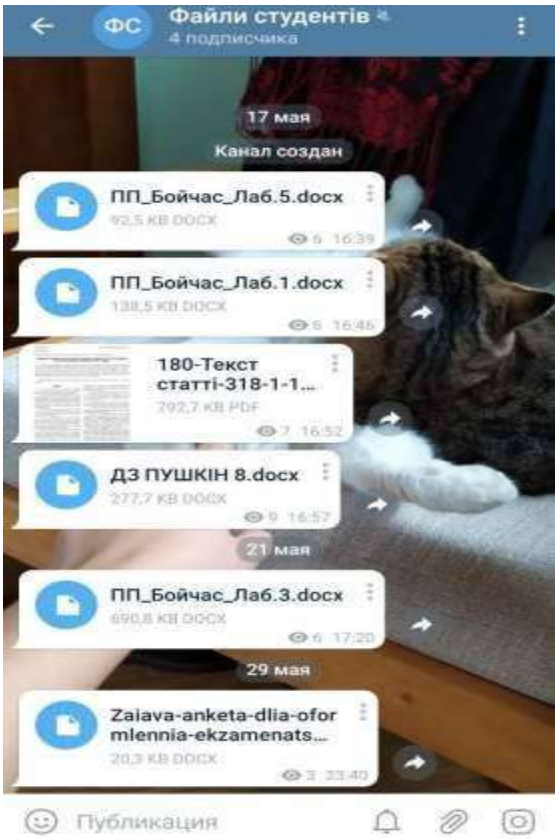


Рисунок 3.18 - Канал який зберігає файли студентів

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

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

ВИСНОВКИ

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

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

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

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

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

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

- визначено зміст основних етапів створення чат-ботів.

Відбулося дослідження технологій для побудови чат-бота. Виявлено, що обмін повідомлень у чат-боті відбувається шляхом клієнт-серверної архітектури. Всі необхідні дані передаються по HTTP протоколу за допомогою формату даних JSON. Для чат-бота продуманий функціонал для обміну інформації між викладачем та студентом. Завдяки тому, що дистанційне навчання нині відбувається з використанням веб-сервісів, то викладач має змогу за допомогою бота надсилати посилання на запис лекції, на документи або на певні матеріали, адже це має ряд переваг. Але також розглянутий другий варіант функціоналу, обміну інформації за допомогою звичайних файлів. Таку можливість продемонстровано з боку студента. Також підключена база даних MongoDB, як можливість для викладача зберігати велику кількість посилань у

разі збільшення кількості студентів, або предметів.

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

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

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

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів / О.І. Провотар, Х.А. Клочко // Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка. – 2013. – № 3. – 6 с.

2. Сухас У. Oracle Intelligent Bots: Чат-боти із штучним інтелектом [Електронний ресурс] / У. Сухас. – Oracle, 2017. – Режим доступу: https://www.oracle.com/a/ocom/docs/chatbots.pdf.

3. Українець І. Системи аналітики для чат-ботів [Електронний ресурс] / І. Українець. – Режим доступу: http://ilyaukrainets.ru/chatbot/sistemy-analitiki- dlya-chat-botov/.

4. Ясницький Л.В. Інтелектуальні системи/ Л.В. Ясницький. – М.: Лабораторія знань, 2016. – 221 с.

5. A curated directory of chat bot resources & tools [Електронний ресурс]. – Режим доступу: http://www.botsfloor.com/botstash/products/?category=Development%20Platfors.

6. Beaver L. The Chatbots Explainer [Електронний ресурс] / L. Beaver. – BI Intelligence Copyright, 2016. – 23 p. – Режим доступу: https://www.businessinsider.com/intelligence/chatbots-explainer.

7. Chatbot Report 2018: Current landscape of how people create chatbots and how users expect to interact with them [Електронний ресурс]. – Режим доступу: https://elearningindustry.com/chatbots-for-learning-support-10-reasonshttps://naiz.chat/NAIZ-report-18072018.pdf.

8. Gartner Identifies Five Emerging Technology Trends That Will Blur the Lines Between Human and Machine [Елек- тронний ресурс]. – Режим доступу: https://www.gartner.com/en/newsroom/press-releases/2018-08-20-gartner-identifies-five-emerging-technology-trends-that-will-blur-the-lines-between-human-and-machine.

9. Goncalves P. 10 graphs that show why your business should be available through messaging apps [Електронний ресурс] / P. Goncalves. – Режим доступу: https://medium.com/hijiffy/10-graphs-that-show-the-immense-power-of-messaging-apps-4a41385b24d6

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

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

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

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

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

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

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

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

18. Reach of mobile messenger apps among Android users in Ukraine in June 2020, by platform. [Електронний ресурс] / statista.com – 2020. – Режим доступу: https://www.statista.com/statistics/1188579/most-popular- messengers-in-ukraine/

19. MTProto Mobile Protocol. [Електронний ресурс] / core.telegram – 2017. - Режим доступу: https://core.telegram.org/mtproto

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

21. Bots: An introduction for developers. [Електронний ресурс] /core.telegram 2017. – Режим доступу: https://core.telegram.org/bots

22. For the next generation of intelligent applications. [Електронний ресурс] /mongodb.com 2023. – Режим доступу: https://www.mongodb.com/

23. Working with MongoDB. [Електронний ресурс] /code.visualstudio.com 2023. – Режим доступу: https://code.visualstudio.com/docs/azure/mongodb

24. What is MongoDB. [Електронний ресурс] /intellipaat.com 2023. – Режим доступу: https://intellipaat.com/blog/what-is-mongodb/

25. Why Use MongoDB^ What It Is and What Are the Benefits [Електронний ресурс] /simplilearn.com 2023. – Режим доступу: https://www.simplilearn.com/tutorials/mongodb-tutorial/what-is-mongodb

26. Redirecting. Google. URL: http://www.google.com/url?q=http://www.securelist.com/ru/threats/spam?chapter=154&sa=D&sntz=1&usg=AFQjCNFASTYolbPBH4r4pOfQo5V2TOIuvg.

27. Спам, види спаму і боротьба зі спамом - Спам і боротьба із спамом - - Статті про віруси, антивіруси і антивірусні програми - Безкоштовні програми для комп'ютера. Безкоштовні програми для комп'ютера - Безкоштовні антивіруси і антивірусні програми. URL: https://best-free-soft.at.ua/publ/spam_vidi_spamu_i_borotba_zi_spamom/1-1-0-33.

28. Економічна правда. Топ-10 корисних Telegram-ботів для українських користувачів. Економічна правда. URL: https://www.epravda.com.ua/publications/2017/08/7/627822/.

29. Telegram FAQ. Telegram. URL: https://telegram.org/faq.

30. Telegram bot API. Telegram APIs. URL: https://core.telegram.org/bots/api.

31. Простий telegram-бот на python за 30 хвилин. Хабр. URL: https://habr.com/ua/post/442800/

32. Telegram чат-бот с ШІ в 30 рядків коду на Python. Хабр. URL:

https://habr.com/ua/post/346606/

33. Json. JSON. URL: https://www.json.org/json-ru.html

34. Джехі. Автоматизоване тестування ботів для Telegram. Хабр. URL: https://habr.com/ua/post/322816/.

35. Огляд протоколу HTTP - HTTP | MDN. MDN Web Docs. URL: https://developer.mozilla.org/ru/docs/Web/HTTP/Overview.

36. Самовчитель Python | Python 3 для початківців - уроки програмування. URL: https://pythonworld.com/manual-python.

37. Корисні Telegram-боти – список. AIN.UA. URL: https://ain.ua/2017/08/14/30-telegram-botov/.

38. Barker S. Deep Learning Chatbots: Everything You Need to Know | HackerNoon. HackerNoon - read, write and learn about any technology. URL: https://hackernoon.com/deep-learning-chatbot-everything-you-need-to- know-r11jm30bc.

39. Кафедра інтелектуальних інформаційних систем. Технології створення чат-бота для надання користувачам технічної допомоги 2022р. Соловйова А.Б. 122 – БКР – 401з.210901101

40. Welcome to python.org. Python.org. URL: https://www.python.org.

41. Free and open search: the creators of elasticsearch, ELK & kibana | elastic. Free and Open Search: The Creators of Elasticsearch, ELK & Kibana | Elastic. URL: https://www.elastic.co.

ДОДАТОК А

Лістинг блоку реєстрації

```
import csv
import pymorphy2 import re
import psycopg2
conn = psycopg2.connect(dbname='energy', user='mao', password='daring', host='localhost')
cursor = conn.cursor()

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

answer_id=[] answer = dict()

cursor.execute('SELECT id, answer FROM app.chats_answer;') records = cursor.fetchall()
for row in records:
 answer[row[0]]=row[1]

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

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

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

cursor.close() conn.close()
import sklearn
from sklearn.feature_extraction.text import TfidfVectorizer


from sklearn.decomposition import TruncatedSVD

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

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


# тіло програми k=5, temperature=10.0 можна підібрати import numpy as np

from sklearn.neighbors import BallTree from sklearn.base import BaseEstimator
```

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


from sklearn.pipeline import make_pipeline

ns_q = NeighborSampler()
ns_q.fit(matrix_small_q, answer_id)
pipe_q = make_pipeline(vectorizer_q, svd_q, ns_q)

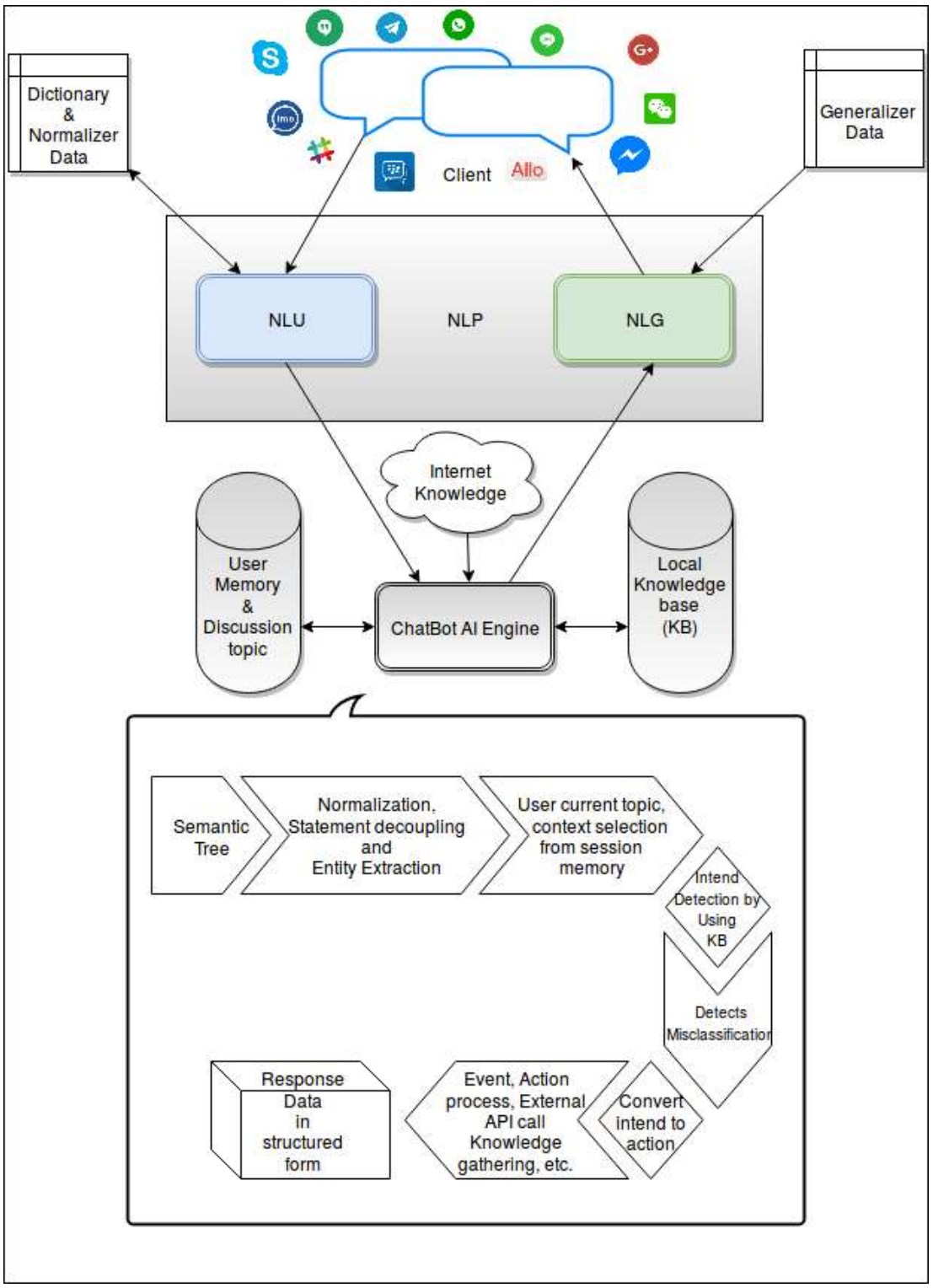
import re
import telebot
telebot.apihelper.ENABLE_MIDDLEWARE = True
bot = telebot.TeleBot("299999999:sdfgnreognrtgortgmrtgmrtgm")


@bot.message_handler(commands=['start'])
def start_message(message):
bot.send_message(message.from_user.id, "?")


@bot.message_handler(func=lambda message: True)
def get_text_messages(message):
                        request=message.text
                        words= re.split('\W',request)
                        phrase=""
                        for word in words:
                                word = morph.parse(word)[0].normal_form
                                phrase = phrase + word + " "
                        reply_id       = int(pipe_q.predict([phrase.strip()]))
                        bot.send_message(message.from_user.id, answer[reply_id])
                        print("Запрос:", request, " \n\: ", phrase, " \n\t\tОтвет :", answer[reply_id])
bot.infinity_polling(none_stop=True, interval=1)
print("Пишіть ваше питання, слова exit або вихід для виходу") request=""
while request not in ['exit', 'выход']:
 request=input()
 words= re.split('\W',request)
 phrase=""
 for word in words:
 word = morph.parse(word)[0].normal_form
 phrase = phrase + word + " "
 reply_id = int(pipe_q.predict([phrase.strip()]))
 print (answer[reply_id])
```

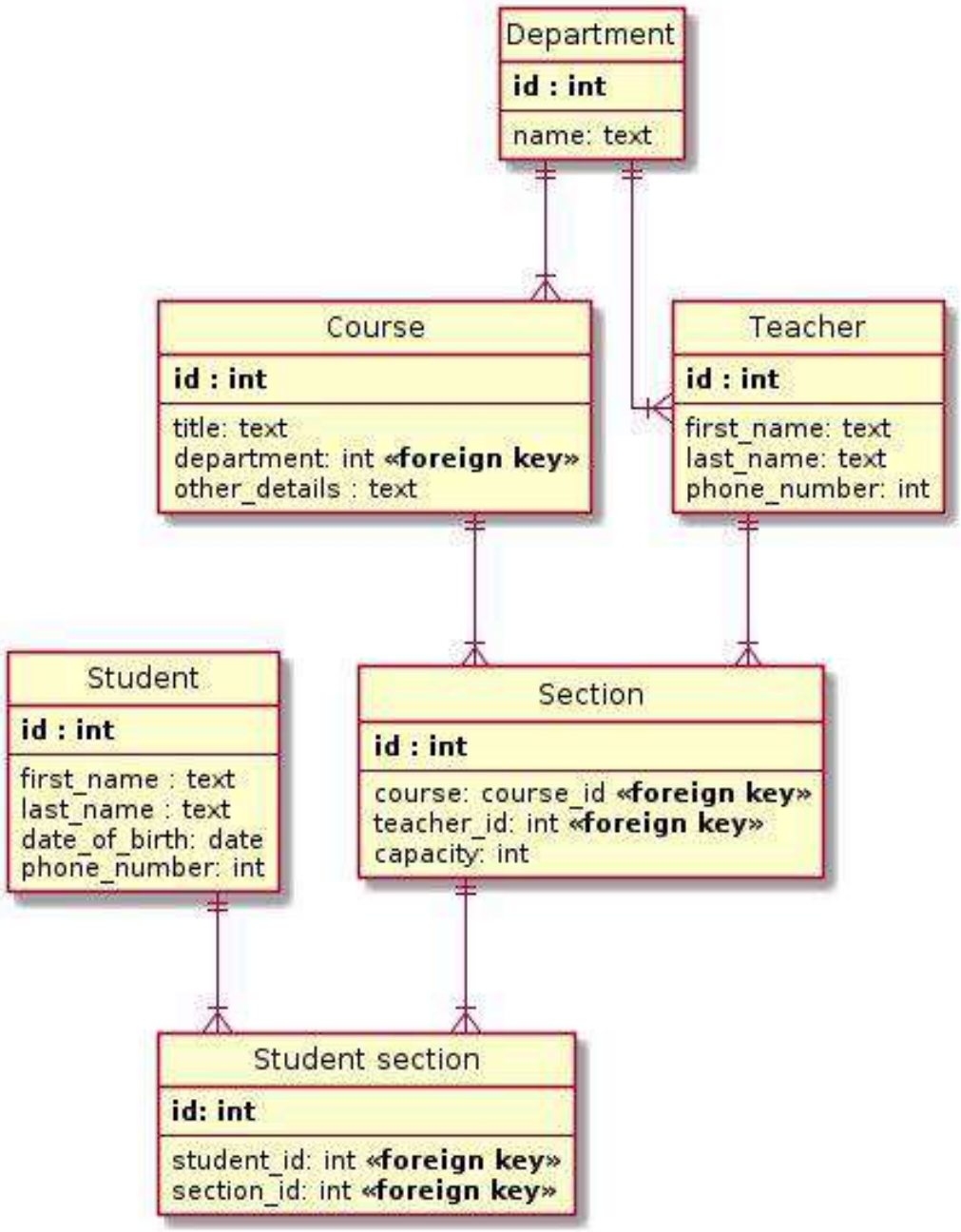
ДОДАТОК Б

Алгоритм роботи чат-боту



ДОДАТОК В

Зв’язки в базі даних



ДОДАТОК Г

Приклад роботи з чат-ботом



ДОДАТОК Д

Видалення роботи з чат бота

