

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет
Кафедра

Комп'ютерної інженерії та управління
Комп'ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель інтелектуальної системи чат ботів

(тема)

Виконав:

студент IV курсу, групи КІТМ-21-2
Безсонов С. О.
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерні
інтелектуальні технології
(повна назва освітньої програми)

Керівник: проф. Аксак Н. Г.
(посада, прізвище, ініціали)

Допускається до захисту

(підпис)

Зав. кафедри

(підпис)

О.Г. Руденко

2022 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	другий (магістерський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:
Зав. кафедри

_____ (підпис)
« _____ » _____ 202_ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Безсонову Сергію Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Модель інтелектуальної системи чат-ботів

затверджена наказом по університету від " 07 " листопада 2022 р. № 1455 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____

3. Вхідні дані до роботи _____

Мова програмування: JavaScript.

Фреймворк: React.js.

Система оновлення станів: Redux.

4. Перелік питань, що потрібно опрацювати у роботі _____

1.

2.

3.

4.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та актуальності проблеми інтелектуалізації чат ботів	08.11 – 13.11	
2	Огляд сучасної літератури за напрямком магістерської роботи	13.11 – 20.11	
3	Аналіз підходів інтелектуалізації	21.11 – 23.11	
4	Огляд та вибір комплектуючих майбутньої системи	24.11 – 26.11	
5	Розробка програмного забезпечення та мобільного додатку для інтелектуалізації Чат ботів	26.11 – 30.12	
6	Оформлення пояснювальної записки	01.12 – 07.12	
7	Оформлення графічного матеріалу	08.12 – 10.12	

Дата видачі завдання 07.11.2022

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Проф. Аксак Н.Г..
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 85 с., 24 рис., 4 табл., 1 дод., 32 джерел.

JAVASCRIPT, REACT JS, REDUX, ЧАТ-БОТ, LSTM, РОЗМОВНИЙ АГЕНТ

Метою кваліфікаційної роботи є розробка моделі інтелектуальної системи чат бота, здатної завдяки мовленню підвищити ефективність сприймання інформації та зменшити когнітивні зусилля, а також отримати більше задоволення від обслуговування, ніж текстова взаємодія.

Проведений лабораторний експеримент із 116 учасниками, які виконували завдання з пошуку інформації за допомогою розмовного агента або чат-бота.

Об'єктом дослідження є процес побудови інтелектуальної системи чат ботів.

Предметом дослідження є розробка моделі інтелектуальної системи чат-ботів.

Отримані результати можна використовувати для підвищення рівня якості обслуговування, гарантуючи, що персоналізоване обслуговування доступне для задоволення потреб клієнтів будь-коли та будь-де.

ABSTRACT

Master's thesis: 85 pages, 24 figures, 4 tables, 1 appendices, 32 sources.

JAVASCRIPT, REACT JS, REDUX, CHATBOT, LSTM,
CONVERSATIONAL AGENT

The goal of the qualification work is to develop a model of an intelligent chat bot system, capable, thanks to speech, on the one hand, to increase the efficiency of information perception and reduce cognitive efforts, and on the other hand, to get more satisfaction from the service than text interaction. A laboratory experiment was conducted with 116 participants who performed information search tasks using a conversational agent or chatbot.

The object of research is the process of building an intelligent system of chat bots.

The subject of the research is the development of a model of an intelligent system of chatbots.

The results obtained can be used to improve the level of service quality, ensuring that personalized service is available to meet the needs of customers anytime, anywhere.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ Комп'ютерних інтелектуальних технологій та систем _____

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти другий (магістерський)

Модель інтелектуальної системи чат ботів

Виконав:

студент 2 курсу, групи КІТм-21-2

Безсонов С.О.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма

Комп'ютерні інтелектуальні технології

Керівник проф. Н.Г.Аксак

2022 р.

АНОТАЦІЯ

Актуальність теми дослідження.

За останні роки чат-боти почали швидко розвиватися в багатьох галузях, включаючи маркетинг, допоміжні системи, освіту, охорону здоров'я, культурну спадщину та розваги.

Чат-боти замінюють деякі роботи, які традиційно виконують люди, наприклад онлайн-агенти з обслуговування клієнтів і викладачі. Від початкової стадії чат-ботів на основі правил до епохи швидкого розвитку штучного інтелекту (ШІ) продуктивність чат-ботів постійно покращується. Зараз чат-боти можуть «спілкуватися» як люди, і вони можуть вчитися на досвіді.

Чат-боти використовують машинне навчання виявлення моделей спілкування. Завдяки постійному взаємодії з людьми вони вчаться наслідувати реальні розмови і реагують на усні чи письмові запити, допомагаючи знайти відповіді. Оскільки чат-боти використовують штучний інтелект (ШІ), то розуміють мову, а чи не просто команди. Таким чином, після кожного діалогу вони стають розумнішими.

Традиційна взаємодія агентів із обслуговування передбачає пряму особисту взаємодію клієнта/працівника, але соціальні мережі тепер задовольняють потреби клієнтів щодо негайної відповіді за межами фактичного об'єкта. Клієнти, які користувалися як онлайн-, так і офлайн-сервісами, тепер вважають, що онлайн-сервіси ефективні, доступні та економлять час і кошти. Поширення цифрових послуг і каналів цифрового маркетингу надає брендам нові можливості для задоволення клієнтів. Сектор розкоші слідує цій тенденції, впроваджуючи цифрові послуги, які пропонують цілодобове обслуговування клієнтів через Chatbot, онлайн-систему чату.

Концепція Chatbot добре поєднується з цінностями бренду роздрібної

торгівлі класу «люкс» щодо надання чудових послуг для споживачів, які готові платити більше. Крім того, Chatbot розроблено для того, щоб розвивати майбутні стосунки між брендом і споживачем.

У цій роботі досліджується вплив особистості чат-бота на досвід користувача та досліджується, як користувачі сприймають особистість агента, коли вона передається через текст.

Дослідження показує, що електронний сервіс Chatbot забезпечує інтерактивні та захоплюючі зустрічі з брендом/обслуговуванням клієнтів.

Метою кваліфікаційної роботи є розробка моделі інтелектуальної системи чат бота, здатної завдяки мовленню підвищити ефективність сприймання інформації та зменшити когнітивні зусилля, а також отримати більше задоволення від обслуговування, ніж текстова взаємодія. Проведений лабораторний експеримент із 116 учасниками, які виконували завдання з пошуку інформації за допомогою розмовного агента або чат-бота.

Об'єктом дослідження є процес побудови інтелектуальної системи чат ботів.

Предметом дослідження є розробка моделі інтелектуальної системи чат-ботів.

Отримані результати можна використовувати для підвищення рівня якості обслуговування, гарантуючи, що персоналізоване обслуговування доступне для задоволення потреб клієнтів будь-коли та будь-де.

У першому розділі розглянуто можливості чат-ботів.

Чат-боти можуть бути корисними інструментами для людей із психічними розладами, особливо для тих, хто не бажає звертатися за консультацією щодо психічного здоров'я через стигматизацію. Існує безліч чат-ботів, які використовуються для різних психічних розладів і цілей.

За останнє десятиліття спостерігалось збільшення використання інформаційних технологій для навчання та підтримки людей з деменцією та осіб, які доглядають за ними. У той же час технології чат-ботів стають все

більш популярними для використання населенням і, як було визначено, мають переваги для надання медичної допомоги. Однак мало відомо про те, як технології чат-ботів можуть принести користь людям із деменцією та особам, які за ними доглядають.

Цифровізація освіти та переосмислення досвіду навчання є одним із великих викликів у цю епоху інформації. У сфері електронного навчання застосування чат-бота як частини навчання продемонструвало цікавий потенціал, як як навчального, так і адміністративного інструменту. Чат-боти були “в тренді” протягом кількох років, і було опубліковано чимало робіт, які досліджують їх в освітньому секторі, хоча, схоже, дуже мало інтересу до узагальнення цих знань. Оскільки в цей час було розроблено досить багато технологій чат-ботів, які демонструють різноманітні функції, дослідження було обмежено дослідженням лише ботів, заснованих на похідній мові XML AIML.

Результати показують, що чат-боти в освіті мають чимало застосувань і навіть більше можливостей. Чат-бот на основі AIML може бути як простим, так і складним для реалізації, все залежить від зусиль, докладених для впровадження.

Також були розглянуті переваги та недоліки чат-ботів.

До переваг можна віднести:

Цілодобова доступність.

Миттєві відповіді та багатозадачність.

Чат-бот легко налаштувати.

Використання чат-ботів дозволяє економити гроші.

До недоліків можна віднести:

Клієнти частіше обирають живе спілкування.

Чат-боти не можуть вникнути в суть праці.

Немає здатності приймання рішень.

Було зроблено огляд сучасною науковою літератури по створенню та розробці чат-ботів.

У другому розділі були досліджені складнощі створення чат-ботів. Основна складність у створенні програми для месенджерів полягає у розробці зручного та функціонального інтерфейсу, виходячи з того, які можливості ми маємо у своєму розпорядженні всередині месенджера.

Якщо нам потрібно створити складну послідовність дій і реалізувати якийсь більш-менш виразний спосіб отримати складний сценарій поведінки користувача в багатьох месенджерах, ми зіткнемося з великими проблемами щодо зручності побудови відповідного інтерфейсу. Наприклад, це може бути складна форма, в якій послідовність заповнення/вибору впливає на наступний крок та/або дії у додатку.

Великий сегмент ботів закриває завдання взаємодії всередині месенджера. Це як офіційні офіси, так і чати для спілкування з відомими людьми або компаніями, а також додатковий функціонал для модерації чатів і спільнот.

Наведена класифікація чат-ботів.

У чат-ботів є кілька класифікацій: за алгоритмом, виглядом та функціональністю. За алгоритмом бувають примітивні та саморозвинені. За виглядом чат-боти діляться: кнопковий та текстовий. За функціоналом виділяють: комунікаційні та функціональні.

Також розглянуті моделі сценаріїв чат-ботів, такі як структура вибору шляху “Печера часу“, “Рукавичка“, “Гілки та пляшкова шийка“, “Квест“, “Відкрита картка“.

Було розглянуто 6 інструментів для створення чат-ботів, що найбільш широко представлені на світовому ринку:

- Flow XO — онлайн-сервіс для створення чат-ботів та їх інтеграції в популярні месенджери та чати на сайтах.
- Botsify — це платформа, де люди та компанії можуть створювати власні інтелектуальні чат-боти, не знаючи мови програмування.
- Dexter — роботи створюються не за допомогою додавання і перетягування блоків, а в текстовому редакторі.

– Gerabot – український онлайн-конструктор для створення та управління чат-ботами, завдяки якому можна замовити створення чат-бота під завдання бізнесу.

– Chatbot – платформа для створення чат-ботів під завдання основних напрямів бізнесу (маркетинг, продаж, сапорт клієнтів).

По результатах порівняльного аналізу зроблено висновки, щодо найбільшої популярності.

Переважає більшість популярних чат-бот платформ має інструменти моніторингу різних метрик, таких як: динаміка кількості звернень динаміка кількості користувачів, часті теми та ключові слова звернень, інструменти перегляду невдалих діалогів.

Третій розділ присвячений проектуванню та розробці інтелектуального чат-бота. Вимоги до проектування чат-бота включають точне представлення знань, стратегію генерації відповідей і набір попередньо визначених нейтральних відповідей, щоб відповісти, коли висловлювання користувача не зрозуміло. Першим кроком у проектуванні є її розділення на складові частини відповідно до стандарту, щоб можна було дотримуватися модульного підходу до розробки. Розроблено загальну архітектуру чат-бота.

Сучасні сайти відрізняються від тих, якими ми користувалися раніше: вони динамічніші. З їхньою допомогою можна купувати товари, спілкуватися з друзями чи писати коментарі.

Такий реактивний підхід передбачає роботу з даними, при зміні яких автоматично змінюються інші частини програми, зокрема інтерфейс. При використанні класичної зв'язки JavaScript та HTML перемальовка інтерфейсу дуже ускладнюється.

Такий стан справ підштовхнув розробників React до створення хуків, що дозволяють розширити можливості функціональних компонентів або нівелювати деякі проблеми, які можуть виникати через їхню специфіку. Хуки виявилися настільки зручними, що стали основою React-розробки.

В розділі детально розглянуті найважливіші хуки: `useState`, `useContext`,

DidUpdate, WillUnmount, useRef.

Redux - це інструмент для керування станом даних і інтерфейсом користувача в додатках JavaScript з великою кількістю сутностей. Являє собою бібліотеку JavaScript. Redux - це універсальний інструмент розробки, який можна використовувати у поєднанні з різними бібліотеками та фреймворками.

Четвертий розділ присвячений моделюванню інтелектуальної системи чат-бота. Для побудови моделі дослідження були зроблені припущення щодо обсягу інформації, яку користувачі повинні прослухати, залежить від завдань, які вони виконують.

Гіпотеза Г1: Клієнти досягають вищого рівня ефективності за допомогою мовної взаємодії, ніж за допомогою текстової взаємодії.

Гіпотеза Г2: Цілеспрямованість позитивно пом'якшує вплив модальності взаємодії на ефективність.

Створення тексту передбачає процес свідомого застосування навмисно засвоєних правил, тоді як люди інтуїтивно будують і обробляють мову.

Таким чином, мовлення вимагає більше когнітивних зусиль у емпіричних завданнях:

Гіпотеза Г3: Клієнти сприймають менше когнітивних зусиль, коли вони взаємодіють за допомогою мови, ніж за допомогою тексту.

Гіпотеза Г4: Цілеспрямованість позитивно пом'якшує вплив модальності взаємодії на когнітивні зусилля.

Для завдань, які сприймаються як менш цілеспрямовані, взаємодія служить для вивчення різних варіантів. Таким чином, переваги розмови в двосторонній розмові повинні ставати більш вираженими в міру збільшення тривалості взаємодії.

Гіпотеза Г5: Клієнти отримують більше задоволення, коли спілкуються за допомогою мови, ніж за допомогою тексту.

Гіпотеза Г6: Цілеспрямованість негативно зменшує вплив модальності

взаємодії на задоволення.

Високі когнітивні зусилля негативно впливають на задоволеність клієнтів результатом пошуку. Крім того, задоволеність клієнтів залежить не лише від зовнішньої мотивації, а й від внутрішніх очікувань.

Таким чином, можна очікувати, що задоволення також сприяє задоволенню клієнтів від зустрічі з послугами:

Гіпотеза Г7: Сприйняття ефективності позитивно впливає на задоволеність клієнтів зустріччю з послугами.

Гіпотеза Г8: Когнітивні зусилля негативно впливають на задоволеність клієнтів від зустрічі з послугами.

Гіпотеза Г9: Задоволення позитивно впливає на задоволеність клієнтів від зустрічі з послугами.

Побудований взаїмозв'язок взаємодії та цілеспрямованості завдання. Наведена методика проведення експериментальної процедури.

Для аналізів використовувалась ефективна платформа для аналітики даних і їх візуалізації Tableau, яка дозволяє передавати дані всім користувачам.

У результаті виконання роботи було досліджено моделі інтелектуальних систем чат ботів. Виявлені плюси та мінуси чат ботів в сучасному житті. Запропоновано рекомендації щодо використання чат ботів в таких сферах як: медицина такі складні захворювання, як дименція, освіта.

Розроблена модель інтелектуального чат боту. Проведений лабораторний експеримент із 116 учасниками, які мали виконати завдання з пошуку інформації за допомогою розмовного агента або чат-бота. Результати показують, що мовлення демонструє вищу сприйману ефективність, менші когнітивні зусилля, більше задоволення та більше задоволення від обслуговування, ніж текстова взаємодія. Також виявлено, що ці ефекти залежать від цілеспрямованості завдання

JAVASCRIPT, REACT JS, REDUX, ЧАТ-БОТ, LSTM, РОЗМОВНИЙ АГЕНТ

Використані публікації керівника та співробітників кафедри, що становлять теоретичну базу роботи

1. Ахак N., Korablyov M., Ushakov M. The Development of a Multi-Agent System for Controlling an Autonomous Robot // Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I, (ICTERI-2021). – Kherson, Ukraine, 2021. – Vol-3013, pp. 96-105. <http://ceur-ws.org>. ISSN 1613-0073.

2. Ахак N., Korablyov M., Rosinskiy D. MapReduce Hadoop Models for Distributed Neural Network Processing of Big Data Using Cloud Services / Advances in Intelligent Systems and Computing IV. / Editors: Shakhovska, Natalya; Medykovskyy, Mykola O., Springer, 2019. – pp. 387 – 400. ISSN: 2194-5357 //doi.org/10.1007/978-3-030-33695-0

3. Ахак N.G. Multi agent knowledge management architecture /N.G. Ахак // «Information-Management Systems and Technologies»: Materials of the IV International Scientific Conference, Odessa. – 2015.- p.237-239.

4. Аксак Н.Г. Мультиагентная архитектура управления знаниями. *«Інформаційні управляючі системи та технології»* (ІУСТ-ОДЕСА-2017). Матеріали VI Міжнародної науково-практичної конференції. - Одеса. 2017. - С. 268-270

5. Аксак Н.Г. Концепція побудови мультиагентних систем розподіленої нейромережевої обробки великих даних . ВІСНИК ХНТУ №3(66), - ТОМ 1, - 2018 р. – С. 205-212.

ЗМІСТ

Перелік умовних позначень, символів , одиниць, скорочень, термінів	16
Вступ	17
1 Актуальність проблеми та постановка задачі	19
1.1 Огляд можливостей чат-ботів	19
1.1.1 Чат-боти у сфері психічного здоров'я	19
1.1.2 Чат-боти для підтримки людей з деменцією та осіб, які за ними доглядають	20
1.1.3 Чат-боти в освіті	21
1.2 Переваги та недоліки чат-ботів	22
1.3 Огляд літератури	23
2 Електронний сервіс Chatbot	28
2.1 Складнощі створення чат-ботів	28
2.2 Класифікація чат-ботів	30
2.3 Моделі сценаріїв чат-ботів	32
2.4 Сучасні онлайн сервіси для створення чат-ботів	36
2.4.1 Flow XO	36
2.4.2 Botsify	37
2.4.3 Dexter	38
2.4.4 Gerabot	39
2.4.5 Chatbot	40
2.5 Порівняння платформ для створювання чат-ботів	41
3 Проектування інтелектуального чат-бота	44
3.1 Платформи для розробки чат-ботів	46
3.1.1 Фреймворк react.js	46
3.1.2 JSX	47
3.1.3 React-hooks	49
3.1.4 Redux	55

4	Моделювання інтелектуальної системи чат-бота	62
4.1	Методика проведення експериментів	66
4.2	Результати експериментів	67
	Висновки	73
	Перелік посилань	74
	Додаток	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ , ОДИНИЦЬ,
СКОРОЧЕНЬ, ТЕРМІНІВ

IoT – Internet of Things

MAS – Multi-agent System

ГА – голосовий агент

РА – Розмовний агент

URL – Uniform Resource Locator

API – Application Programming Interface

ВСТУП

За останні роки чат-боти почали швидко розвиватися в багатьох галузях, включаючи маркетинг, допоміжні системи, освіту, охорону здоров'я, культурну спадщину та розваги.

Чат-боти замінюють деякі роботи, які традиційно виконують люди, наприклад онлайн-агенти з обслуговування клієнтів і викладачі. Від початкової стадії чат-ботів на основі правил до епохи швидкого розвитку штучного інтелекту (ШІ) продуктивність чат-ботів постійно покращується. Зараз чат-боти можуть «спілкуватися» як люди, і вони можуть вчитися на досвіді.

Чат-бот (англ. chatbot) - це програма, яка імітує реальну розмову з користувачем. Чат-боти дозволяють спілкуватися за допомогою текстових або аудіо повідомлень на сайтах, месенджерах, мобільних додатках або по телефону.

Чат-боти використовують машинне навчання виявлення моделей спілкування. Завдяки постійному взаємодії з людьми вони вчаться наслідувати реальні розмови і реагують на усні чи письмові запити, допомагаючи знайти відповіді. Оскільки чат-боти використовують штучний інтелект (ШІ), то розуміють мову, а чи не просто команди. Таким чином, після кожного діалогу вони стають розумнішими. Варто відзначити, що окрім чат-ботів з ШІ, є й ті, які працюють на основі запрограмованих сценаріїв з множинним вибором, наприклад, опція А веде до опції В і таке інше.

Невмираючий тренд останніх років залишається актуальним попри все. За 2021 рік чат-ботів почали використовувати 85% сервісів, які надають послуги онлайн. На даний момент вони справляються з більшістю нескладних питань. Працюють цілодобово та оперативне.

У майбутньому чат-боти зможуть брати на себе більш складні функції. З простих порадників щодо вирішення нескладних питань вони поступово перетворюються на окремий мультифункціональний сервіс. Незважаючи на

багаторічну еволюцію, є відчуття, що бренди лише починають розкривати можливості технології. Потенціал для розвитку та інтеграції креативних механік видається дуже значним.

Вже зараз бренди використовують чат-ботів навіть у такій консервативній галузі, як продаж автомобілів. Можна згадати кейс Kia, яка запустила чат-бота Kian у Facebook. До чат-боту зверталися утричі частіше, ніж до форми запитань на сайті компанії. Конверсія при використанні чат-бота становила 21% проти 7% на сайті.

Показовим є і кейс просунутого чат-бота Emirates Airlines. Він об'єднав у собі ІІ для діалогів та медійну рекламу на сайті. Отриману під час спілкування з користувачем інформацію бот використовував для персоналізації контексту сторінки, що продає. Це спричинило зростання залученості на 87%.

Традиційна взаємодія агентів із обслуговування передбачає пряму особисту взаємодію клієнта/працівника, але соціальні мережі тепер задовольняють потреби клієнтів щодо негайної відповіді за межами фактичного об'єкта. Клієнти, які користувалися як онлайн-, так і офлайн-сервісами, тепер вважають, що онлайн-сервіси ефективні, доступні та економлять час і кошти. Поширення цифрових послуг і каналів цифрового маркетингу надає брендам нові можливості для задоволення клієнтів. Сектор розкоші слідує цій тенденції, впроваджуючи цифрові послуги, які пропонують цілодобове обслуговування клієнтів через Chatbot, онлайн-систему чату.

Концепція Chatbot добре поєднується з цінностями бренду роздрібною торгівлі класу «люкс» щодо надання чудових послуг для споживачів, які готові платити більше. Chatbot пропонує новий рівень підтримки якості обслуговування, гарантуючи, що персоналізоване обслуговування доступне для задоволення потреб клієнтів будь-коли та будь-де. Крім того, Chatbot розроблено для того, щоб розвивати майбутні стосунки між брендом і споживачем.

1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Огляд можливостей чат-ботів

1.1.1 Чат-боти у сфері психічного здоров'я

Чат-боти можуть бути корисними інструментами для людей із психічними розладами, особливо для тих, хто не бажає звертатися за консультацією щодо психічного здоров'я через стигматизацію.

Хоча було проведено численні дослідження щодо використання чат-ботів для психічного здоров'я, існує потреба в систематичному об'єднанні цих доказів, щоб інформувати постачальників послуг у сфері психічного здоров'я та потенційних користувачів про основні функції чат-ботів та їх потенційне використання, а також інформувати майбутні дослідження про основні прогалини попередньої літератури.

У пошуку дослідження було використано сім бібліографічних баз даних (Medline, Embase, PsycINFO, Cochrane Central Register of Controlled Trials, IEEE Xplore, ACM Digital Library та Google Scholar). Крім того, було проведено зворотну та пряму перевірку списку посилань включених досліджень та відповідних оглядів. Вибір дослідження та вилучення даних здійснювали два рецензенти незалежно. Витягнуті дані були синтезовані з використанням нарративного підходу. Чат-боти були класифіковані відповідно до їх цілей, платформ, генерації відповідей, ініціативи діалогу, модальностей введення та виведення, втілення та цільових розладів.

З 1039 отриманих цитат 53 унікальні дослідження були включені в цей огляд. Включені дослідження оцінювали 41 різного чат-бота. Зазвичай чат-боти використовувалися для лікування (n=17), навчання (n=12) і скринінгу (n=10). У більшості досліджень чат-боти базувалися на правилах (n=49) і

впроваджувалися в автономне програмне забезпечення (n=37). У 46 дослідженнях чат-боти контролювали та вели розмови. Хоча найчастіше використовуваним способом введення була лише письмова мова (n=26), найбільш часто використовуваним способом виводу була комбінація письмової, усної та візуальної мов (n=28). У більшості досліджень чат-боти включали віртуальні представлення (n=44). Найбільш поширеним фокусом чат-ботів була депресія (n=16) або аутизм (n=10).

Дослідження щодо чат-ботів у сфері психічного здоров'я тільки починаються. Існує безліч чат-ботів, які використовуються для різних психічних розладів і цілей. Постачальники медичних послуг повинні порівняти чат-ботів, знайдених у цьому огляді, щоб допомогти потенційним користувачам вибрати найбільш відповідний чат-бот для підтримки їхніх потреб у психічному здоров'ї. Потрібні додаткові огляди, щоб узагальнити докази щодо ефективності та прийнятності чат-ботів для психічного здоров'я.

1.1.2 Чат-боти для підтримки людей з деменцією та осіб, які за ними доглядають

За останнє десятиліття спостерігалось збільшення використання інформаційних технологій для навчання та підтримки людей з деменцією та осіб, які доглядають за ними. У той же час технології чат-ботів стають все більш популярними для використання населенням і, як було визначено, мають переваги для надання медичної допомоги. Однак мало відомо про те, як технології чат-ботів можуть принести користь людям із деменцією та особам, які за ними доглядають.

Чат-боти були ідентифіковані шляхом систематичного пошуку в Google Play Store, Apple App Store, Alexa Skills та в Інтернеті. Для оцінки функцій і вмісту виявлених додатків використовувався інструмент оцінки на основі фактичних даних. Оцінка була проведена за угодою між 4 окремими

рецензентами.

Хоча технології чат-ботів добре запроваджені та широко використовуються суспільством, їх розробка для людей із деменцією та осіб, які за ними доглядають, перебуває на зародковому етапі. Враховуючи успішне використання чат-ботів в інших закладах охорони здоров'я та для інших програм, є можливості інтегрувати цю технологію в лікування деменції. Однак для того, щоб оцінити їхній потенціал для адекватного навчання та підтримки цих груп населення, потрібно більше науково-обґрунтованих чат-ботів, які пройшли оцінку кінцевих користувачів.

1.1.3 Чат-боти в освіті

Цифровізація освіти та переосмислення досвіду навчання є одним із великих викликів у цю епоху інформації. У сфері електронного навчання застосування чат-бота як частини навчання продемонструвало цікавий потенціал, як як навчального, так і адміністративного інструменту. Чат-боти були “в тренді” протягом кількох років, і було опубліковано чимало робіт, які досліджують їх в освітньому секторі, хоча, схоже, дуже мало інтересу до узагальнення цих знань. Оскільки в цей час було розроблено досить багато технологій чат-ботів, які демонструють різноманітні функції, дослідження було обмежено дослідженням лише ботів, заснованих на похідній мові XML AIML.

Результати показують, що чат-боти в освіті мають чимало застосувань і навіть більше можливостей. Чат-бот на основі AIML може бути як простим, так і складним для реалізації, все залежить від зусиль, докладених для впровадження. Інструмент різноманітний і може використовуватися для багатьох різних цілей, єдиним обмеженням є креативність і уява творців. Чат-боти у навчанні можуть виконувати такі функції, як: збирання завдання, видача домашнього завдання, бути в ролі таймера, надавати літературу, відповідати на запитання та інше.

1.2 Переваги та недоліки чат-ботів

Переваги:

- Цілодобова доступність. Оскільки для роботи чат-ботів не потрібні співробітники служби підтримки, вони можуть працювати 24/7. А це означає, що клієнти можуть отримати відповіді на свої запитання будь-коли вдень чи вночі.

- Миттєві відповіді та багатозадачність. Клієнти хочуть отримувати швидкі відповіді і з чат-ботами вони їх отримують. Чат-боти мають вже запрограмовані відповіді, що дає перевагу над живими співробітниками. Ще один плюс у скарбничку ботів - багатозадачність. Вони можуть одночасно обробляти велику кількість запитів клієнтів, що людині буде не під силу.

- Їх легко налаштувати. Чат-бот легко налаштувати. Навіть люди без особливих технічних навичок зможуть встановити та запустити їх.

- Економія грошей. Одна з причин, через яку компанії використовують чат-ботів у тому, що вони економічніші, ніж найм співробітників. При цьому ботам не потрібно виходити у відпустку, брати лікарняну, вони працюють постійно і готові обслужити відразу кілька клієнтів.

Недоліки:

- Клієнти частіше обирають живе спілкування. Живі співробітники служби підтримки підвищують рейтинг задоволеності клієнтів. Користувачі вважають за краще вирішувати питання з людьми, тому що бот не завжди може дати зрозумілу відповідь, а значить - марна. Люди вважають за краще мати справу з реальною людиною, тому що з нею легше будувати стосунки.

- Чат-боти не можуть вникнути в суть праці. Якими б добрими не були ваші чат-боти, вони все одно не зможуть розібратися в складніших питаннях і запропонувати клієнту найкраще рішення. Бот не зможе швидко зреагувати у непередбаченій ситуації.

- Немає здатності приймання рішень. Чат-боти можуть допомогти вам тільки з відповідями на прості запитання. Вони не запрограмовані на

ситуації, у яких потрібно ухвалювати рішення. Якщо клієнт запитає у робота думка, чому товар 1 краще, ніж товар 2, то автоматизована система не дасть відповіді, як і на безліч інших конкретних питань.

В процесі виконання роботи потрібно: зробити аналіз існуючих інтелектуальних систем чат-ботів, виявити їх плюси та мінуси, дослідити сучасні моделі роботи чат-ботів, та їх систем.

1.4 Огляд літератури

Ботам судилося домінувати в тому, як люди взаємодіють з Інтернетом речей, який продовжує розвиватися навколо них [1]. Незважаючи на свій інтелектуальний потенціал, який все ще розвивається, великі компанії (наприклад, Apple, Google і Amazon) уже розмістили (чат)ботів у центрі своїх флагманських пристроїв. Зараз боти наповнюють Інтернет, діючи як гіді, купців і помічників. Однак чат-боти, розроблені як комунікатори, ще не зробили вагомого внеску у своє, можливо, найприродніше покликання: партнерів з вивчення іноземних мов. Цей огляд стосується трьох питань, пов'язаних із цією проблемою: Чому чат-боти ще не є центром вивчення іноземних мов? Над чим працюють два ключові розробники чат-ботів, які могли б підштовхнути чат-ботів до уваги вивчення мови? 3. Що можуть спільно зробити дослідники, педагоги та розробники, щоб підтримати чат-ботів як партнерів із вивчення іноземних мов прямо зараз? Автор робить висновки - протягом перших трьох десятиліть чат-боти вирости від дослідницького програмного забезпечення до трансляції.

Протягом двох десятиліть що послідувало за цим початковим зростанням, прогресом у перетворенні тексту в мову, та дедалі більшим використанням, смартфони та домашні помічники зробили чат-боти частиною повсякденного життя багатьох користувачів. Користувачі випробовують такі відомі чат-боти, як Cleverbot, для вивчення мови цілей протягом десятиліть. Незважаючи на дослідження, які вказують на сильні

сторони [9] та слабкі сторони [10] випадкових співрозмовників із чат-ботами, незначний прогрес щодо чат-ботів як істотних партнерів по вивченню мови. За останні три роки такі компанії, як Mondly і Duolingo працював над тим, щоб заповнити цю прогалину, розмістивши чат-ботів у центрі своєї платформи для вивчення мови.

Золотий вік для вивчення мов, який обіцяють чат-боти, все ще на горизонті. Поряд із зусиллями поточного загального чат-бота та онлайн-розробників для вивчення мови, існують можливості для більш ефективного використання. Розробники можуть, також плідно співпрацювати з дослідниками в ширшій галузі цифрового мультимедійного навчання, яка встановила ряд невеликих, але ефективних заходів, за допомогою яких ЗМІ, як правило, і агенти розмови, зокрема, можуть бути вдосконалені. Чат-боти – це новий, революційний етап вивчення іноземних мов. Наразі вони а корисний інструмент, який продовжує рости та розвиватися. У цьому огляді пропонується використовувати чат-ботів ефективніше прямо зараз із відносно невеликими коригуваннями та застосуванням досліджень з інших сфер освітніх технологій.

Огляд особливостей чат-ботів у сфері психічного здоров'я [2]. Автор цієї статті вважає, що чат-боти можуть бути корисними інструментами для людей із психічними розладами, особливо для тих, хто не бажає звертатися за консультацією щодо психічного здоров'я через стигматизацію. Вони використовували сім бібліографічних баз даних (Medline, Embase, PsycINFO, Cochrane Central Register of Controlled Trials, IEEE Xplore, ACM Digital Library та Google Scholar). Крім того, провели зворотню та пряму перевірку списку посилань включених досліджень та відповідних оглядів. Вибір дослідження та вилучення даних здійснювали два рецензенти незалежно. Витягнуті дані були синтезовані з використанням нарративного підходу. Чат-боти були класифіковані відповідно до їх цілей, платформ, генерації відповідей, ініціативи діалогу, модальностей введення та виведення, втілення та цільових розладів.

З 1039 отриманих цитат 53 унікальні дослідження були включені в цей огляд. Включені дослідження оцінювали 41 різного чат-бота. Зазвичай чат-боти використовувалися для лікування 17, навчання 12 і скринінгу 10. У більшості досліджень чат-боти базувалися на правилах 49 і впроваджувалися в автономне програмне забезпечення 37. У 46 дослідженнях чат-боти контролювали та вели розмови. Хоча найчастіше використовуваним способом введення була лише письмова мова 26, найбільш часто використовуваним способом виводу була комбінація письмової, усної та візуальної мов 28. У більшості досліджень чат-боти включали віртуальні представлення 44. Найбільш поширеним фокусом чат-ботів була депресія 16 або аутизм 10. Дослідження використання чат-ботів у сфері психічного здоров'я тільки починаються. За результатами дослідження було рекомендовано звертатись до лікарів аби ті радили якого бота використовувати.

Чат-боти в освіті: минуша тенденція чи цінний педагогічний інструмент [4]? В цьому дослідженні автор акцентує увагу що використовував лише ботів на XML AIML. За результатами виявилось, що чат боти мають достатньо застосувань и багато можливостей. Автор робить висновки, що чат-боти в освіті - інструмент різноманітний і може використовуватися для багатьох різних цілей, єдиним обмеженням є креативність і уява творців.

Використання чат-ботів для організації квестового модульного навчання студентів інженерних спеціальностей [5]. У статті розглядаються основні проблеми під час навчання студентів інженерних та інформаційних технологій іноземної мови. Одним із експериментальних шляхів вирішення проблеми є конспект, який базується на спеціально розробленій системі невимушених завдань - квестів. Маючи справу зі студентами інженерних спеціальностей, у яких часто відсутня мотивація до оволодіння іноземною мовою та дещо обмежений час на навчання, виправданий навчальним планом, вкрай важливо базувати курс англійської мови на професійних та комунікативних викликах, з якими стикається майбутній фахівець. .

Відповідно до цього основного принципу було розроблено серію додаткових вправ - так званих квестів - для подальшого залучення студентів до вивчення мови, а також для розвитку їхніх міжкультурних та комунікативних компетенцій, академічних та професійних навичок. Навантаження, яке створює така диверсифікована програма, зменшується за допомогою засобів соціальних мереж і чат-ботів для автоматизації процесів підрахунку балів і спілкування.

Симуляція дії людини в циклі в реальному часі за допомогою мобільних агентів, чат-ботів і системи визначення натовпу для розумних міст [6]. Ця стаття дає детальний огляд і представлення проблем і методологій, які використовуються для вивчення та контролю великомасштабних і складних соціально-технічних систем за допомогою агентних методів та чат-ботів. Автор використовував чат-ботів для розпізнавання натовпу, аби доповнена віртуальна реальність була схожа на реальну.

Нафто-газові чат-боти [7]. Сенді, Неш та Ральф – 3 програми чат-ботів, які були розроблені спеціально для пошуку відповідей на складні запитання професіоналів з нафти та газу. Їх інноваційне програмне забезпечення дозволило мільйонам людей вести діалог за допомогою ноутбуків, смартфонів і колонок. Споживачі використовують НЛП переважно вдома для простих запитів; музика та звіти про погоду завершують два найпопулярніші використання в одному нещодавньому опитуванні.

Але зараз існує поштовх до впровадження цієї технології в офіси по всьому світу, де вона має потенціал для підвищення ефективності працівників. Це ринкове випробування лише триває в нафтогазовому бізнесі, де впровадження буде залежати від здатності віртуального помічника швидко генерувати надійні оцінки складних проблем, пов'язаних із резервуарами, сейсмічними даними та каротажними даними.

Інтегрований, повсюдний і спільний майнінг процесів за допомогою чат-ботів [8]. У статті автор розповідає як використувув чат бота для

комунікації. У цій статті розглянута інноваційна техніка завершення процесів інтелектуального аналізу у спільній, повсюдній, незалежній та інтегрованій платформі спосіб. Техніка використовує ідею чат-ботів і, включаючи їх в існуючі обговорення обміну миттєвими повідомленнями, до яких можна мати прямий доступ функції видобутку процесів.

Таким чином, метою кваліфікаційної роботи є розробка моделі інтелектуальної системи чат бота, здатної завдяки мовленню підвищити ефективність сприймання інформації та зменшити когнітивні зусилля, а також отримати більше задоволення від обслуговування, ніж текстова взаємодія.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити існуючі системи та платформи для створення інтелектуальних чат-ботів;
- розробити модель інтелектуальної системи чат-бота для забезпечення взаємодії користувачів за допомогою розмови та обміну текстовими повідомленнями;
- створити методику проведення експериментів;
- проаналізувати результати, отримані під час моделювання.

2 ЕЛЕКТРОННИЙ СЕРВІС СНАТВОТ

2.1 Складнощі створення чат-ботів

Для початку варто зрозуміти одну принципову річ: чат-бот – це, перш за все, додаток, який ні в чому не поступається нативному додатку чи сайту. Основою всіх згаданих вище рішень зазвичай є інтерфейс користувача (UI, Front-End) і певний набір дій на стороні сервера, який обробляє запити користувача (Server, Back-End), а також взаємодіє з базою даних і інші послуги. вимоги до продукту повинні бути виконані. Це стандартна класифікація, іншого сучасний інтернет запропонувати нам не може.

По суті, чат-бот — це такий же продукт, як і веб-сайт, але він має одну важливу особливість: робота над інтерфейсом користувача передана аутсорсингу. Тобто ми реалізуємо логіку програми, і відображаємо і взаємодіємо через інтерфейс чату (звичайно, в сучасному розумінні це якийсь месенджер). На даному етапі варто відкинути питання побудови цього самого інтерфейсу для зручності використання через третю програму (знаменитий UX), а також сам процес адаптації подання вашого додатку до можливостей інтерфейсу та чату.

Враховуючи відповідь питанням, що таке чат-бот, можна дійти невітнішого висновку, що чат-бот може вирішувати те ж саме коло завдань, як і будь-який інший продукт. Звичайно, враховуючи деякі особливості:

- найчастіше відбувається мінімальна взаємодія з файловою системою користувача та операційною системою (на відміну від рідної програми);
- обмеження інтерфейсу вибраною програмою чату;
- залежність від продуктивності зовнішнього продукту (це дуже актуально останнім часом);
- найскладніше автоматизоване тестування.

Основна складність у створенні програми для месенджерів полягає у розробці зручного та функціонального інтерфейсу, виходячи з того, які можливості ми маємо у своєму розпорядженні всередині месенджера. Відповідно, при створенні тих чи інших користувачів-флюу у нас виникають складності, з якими доводиться миритися, а іноді, як у випадку зі Slack, у нас з'являються величезні можливості, засновані на блоках, які нам дозволено використовувати.

Якщо нам потрібно створити складну послідовність дій і реалізувати якийсь більш-менш виразний спосіб отримати складний сценарій поведінки користувача в багатьох месенджерах, ми зіткнемося з великими проблемами щодо зручності побудови відповідного інтерфейсу. Наприклад, це може бути складна форма, в якій послідовність заповнення/вибору впливає на наступний крок та/або дії у додатку.

Яке місце вже займають боти в нашому житті? Звичайно, в першу чергу слід згадати ботів для криптовалюти та роботи з нею, купівлі авіа- та залізничних квитків. Так, ви можете знайти багато прикладів із подібною функціональністю. Наприклад, державна залізнична компанія «Укрзалізниця» має ботів для продажу квитків у Telegram і Viber.

Великий сегмент ботів закриває завдання взаємодії всередині месенджера. Це як офіційні офіси, так і чати для спілкування з відомими людьми або компаніями, а також додатковий функціонал для модерації чатів і спільнот. Наступну групу в списку можна назвати групою, в якій ботів можна назвати помічниками в повсякденній діяльності користувачів. Це і фінансовий облік, і різні допоміжні інструменти для моніторингу та управління різними ресурсами, не виходячи з улюбленого месенджера.

Крім того, в цю нішу поступово входить великий бізнес. Багато компаній організовують роботу своєї технічної підтримки через ботів і при цьому допускають кросплатформену взаємодію. Це означає, що люди можуть писати такому боту з різних месенджерів, але при цьому мати схоже або

ідентичне виконання будь-якого запиту.

Слід також згадати, що багато робіт, які зараз можна знайти в тому ж Telegram і мають відносно велику кількість користувачів, мають додатковий функціонал, доступний за рахунок пожертв: як у вигляді підписки, так і у вигляді одноразової оплати. Такі роботи часто мають комерційний успіх, оскільки обслуговування сервера для роботи коштує в рази менше, ніж сайту з аналогічним навантаженням за рахунок розподілу необхідних користувачеві ресурсів, в тому числі і на сервері месенджера.

2.2 Класифікація чат-ботів

У чат-ботів є кілька класифікацій: за алгоритмом, видом та функціональністю.

За алгоритмом бувають:

- Примітивні. До примітивних (обмежених) відносяться найпростіші чат-боти. Вони мають невеликий функціонал і заздалегідь підготовлений скрипт, яким вони спілкуються з користувачами.

- Саморозвинені. В основі чат-бота, що саморозвивається, лежить нейронна мережа. Завдяки цьому бот розуміє, що пише користувач і, аналізуючи інформацію, видає цілком реалістичні відповіді. Іноді може здатися, що ви спілкуєтеся зі справжньою людиною!

На вигляд чат-боти діляться:

- Кнопковий. Спілкування із чат-ботом відбувається за допомогою спеціальних кнопок, на яких запропоновано заздалегідь підготовлені відповіді.

- Текстовий. Діалог між чат-ботом та користувачем відбувається за допомогою тексту. Бот розпізнає ключові слова у запиті користувача та на основі отриманої інформації видає відповідь.

За функціоналом виділяють:

- Комунікаційні. Комунікаційні чат-боти беруть він обов'язки

консультантів. Їхнє основне завдання - це спілкування з користувачем. У їх функціонал може входити відповіді на запитання користувачів, допомога у підборі товару чи послуги, а також інформування про різні акції та знижки.

- Функціональні. Завдяки функціональним чат-ботам стає можливим здійснювати певні дії, такі як купівля товарів чи послуг, проведення оплати тощо.

Популярність чат-ботів - це не просто віяння якихось сучасних трендів, а дійсно дуже зручна та ефективна річ для вирішення типових завдань, наприклад

- допомогти користувачеві зробити замовлення;
- прийняти заявку;
- зібрати зворотний зв'язок від користувачів;
- дати консультацію;
- зібрати контактні дані;
- реалізувати різні товари та послуги;
- допомогти користувачеві записатися в салон краси/ прийом до лікаря та ін;
- завантажити музику, фільм чи книгу.

Це далеко не весь список із можливостями чат-ботів. Є навіть боти для створення ботів.

Очевидний плюс для бізнесменів у тому, що один раз вклавшись у розробку чат-бота для своєї компанії, можна заощадити круглу суму на зарплатах співробітникам. А ще боти не беруть лікарняні, не спізнюються на роботу і завжди мають настрій.

Чат-бот зручний не лише для підприємців, а й для клієнтів. Однією з головних причин є здатність чат-бота відповідати 24/7, що рідко зустрічається при роботі з людським фактором. Причому при спілкуванні з чат-ботом відповідь надходить миттєво, не потрібно чекати до наступного дня, поки консультант прочитає повідомлення і вирішить, що вам відповісти.

2.3 Моделі сценаріїв чат-ботів

Найстаріша та найбільш очевидна структура вибору шляху “Печера часу”. За цією схемою побудовано більшість найпростіших чат-ботів (рис. 2.1).

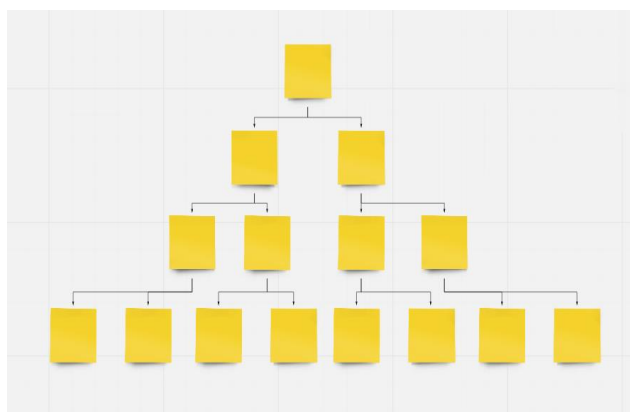


Рисунок 2.1 - Структура вибору шляху “Печера часу”

Вона досить універсальна: чат-боти – консультанти, чат-боти – ігри, техпідтримка. За такою схемою можна побудувати навіть інтернет-магазин у месенджері: чат-бот з’ясовує потреби клієнта і на виході видає два-три варіанти відповідного товару (рис.2.2).

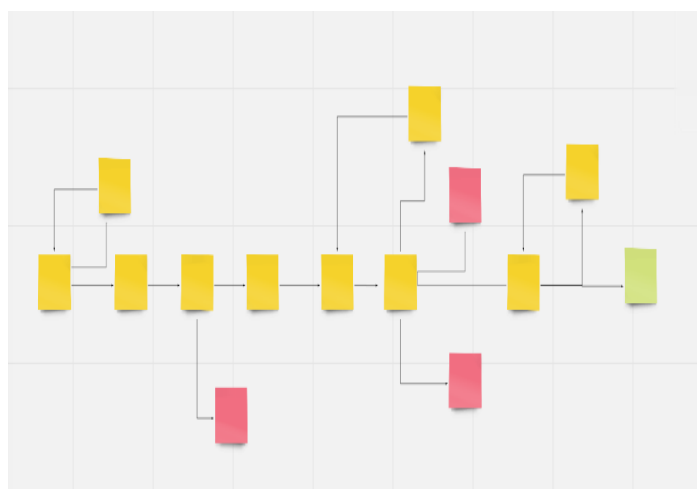


Рисунок 2.2 - Структура вибору шляху “Рукавичка”

Схема більш довша ніж широка. Зазвичай вона має одну центральну вісь, від якої виходять гілки, що закінчується або зупинкою чат-бота, або поверненням на один або кілька кроків назад.

Такі схеми добре підходять для створення опалювальних воронок, навчальних проектів. Ми відправляємо якийсь контент клієнту, який потім задає питання, ми розуміємо, чи готовий клієнт рухатися далі (рис. 2.3).

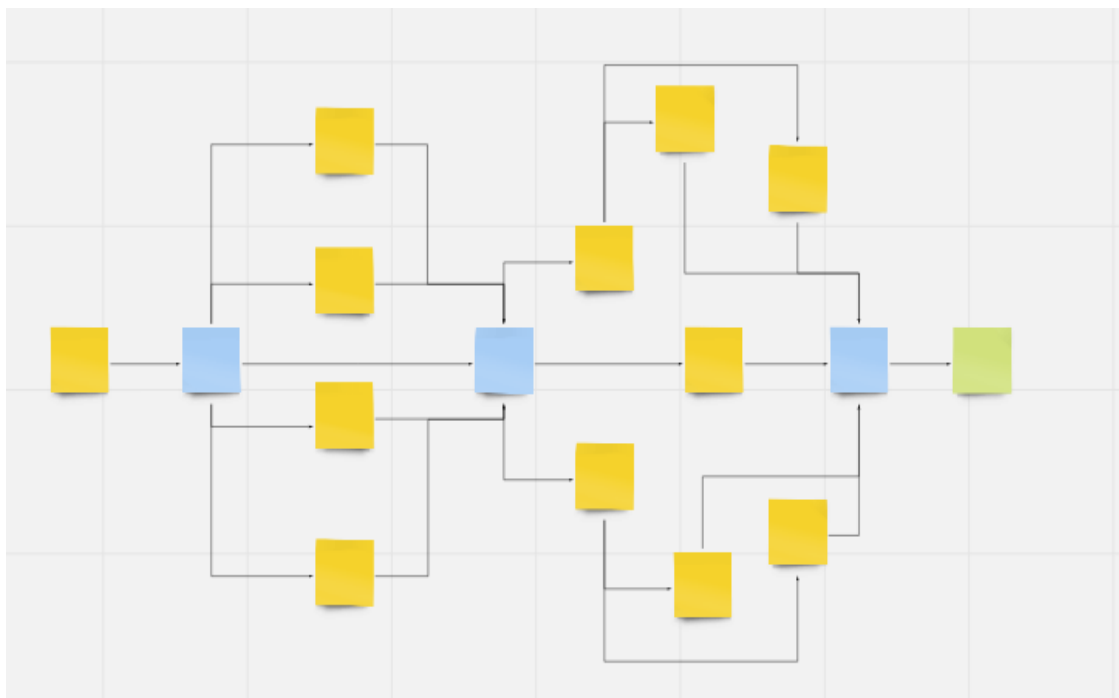


Рисунок 2.3 - Структура вибору шляху “Гілки та пляшкова шийка”

Фреймворк чат-бота розвивається, але завжди наближається до ключових віх. Таку схему можна назвати більш вдосконаленим варіантом рукавички. Зараз він використовується не так часто, тому що чат-боти ще в зародковому стані і користувачі ще не дуже вибагливі, але зовсім скоро все зміниться.

В принципі, використання також має бути зрозумілим. Припустимо, у нас стандартна схема: ковпачок-магніт → натяжний трос - основний продукт → максимізатор доходу → зворотний шлях. Це основна лінія, яка повинна йти до нашого клієнта, але ми створюємо ілюзію вибору, пропонуючи йому

різні варіанти з урахуванням його переваг (рис.2.4).

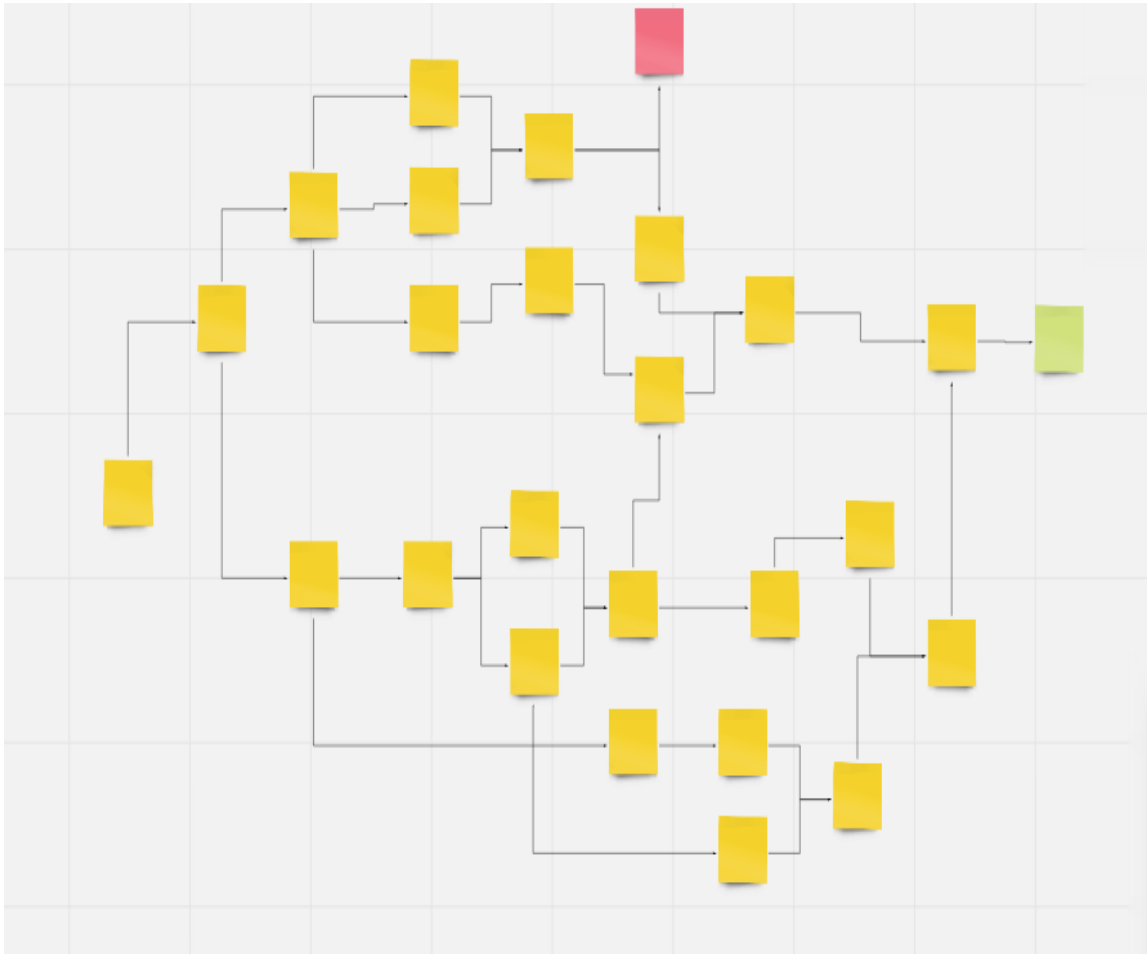


Рисунок 2.4 - Структура вибору шляху “Квест”

Структура складна, хоч і дуже цікава для користувача. Через його складність його не так вже й непрактично використовувати в маркетингових цілях, але ви повинні дуже чітко розуміти, навіщо ви збираєтеся виконувати цю роботу.

Відмінно підходить для гейміфікації. Якщо вам вдасться створити дійсно цікавий квест, великі шанси, що він поширюватиметься вірусно та безкоштовно (рис. 2.5).

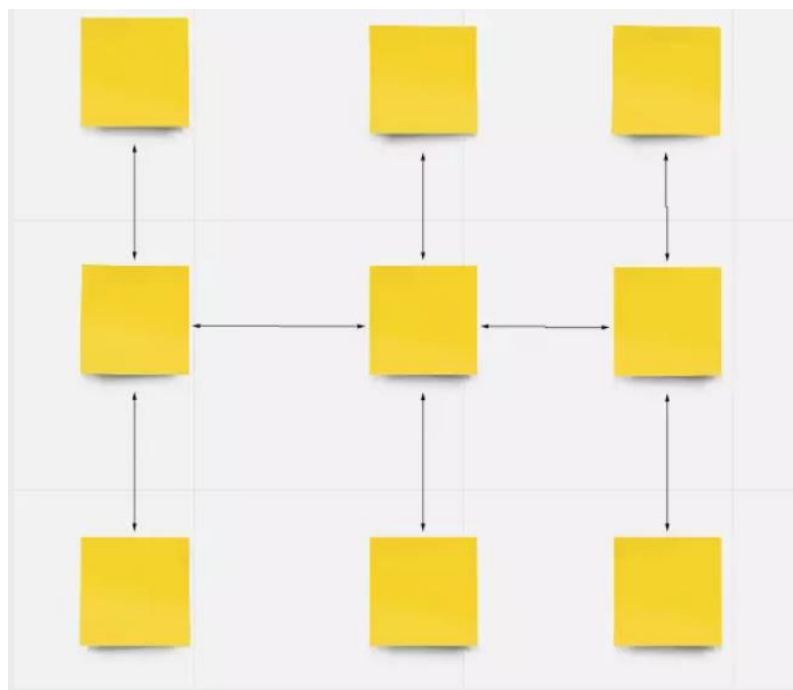


Рисунок 2.5 - Структура вибору шляху “Відкрита картка”

Контур, який можна змішувати в обох напрямках. Особисті кабінети в месенджері мають таку структуру. Або можна зробити, наприклад, путівник торгового центру, де загальне меню - поверх торгового центру, є поділ за типами відділів, з яких зроблені посилання на конкретні магазини (знижки з конкретних (платних) магазинів).

2.4 Сучасні онлайн сервіси для створення чат-ботів

2.4.1 Flow XO

Flow XO — онлайн-сервіс для створення чат-ботів та їх інтеграції в популярні месенджери та чати на сайтах. Сервіс має багатофункціональні засоби автоматизації ботів, які допоможуть покращити комунікацію між компанією та користувачами, а також підвищити залученість споживачів.

В інтерфейсі Flow XO компанії можуть створювати робочі області, де будуть керувати роботом, додавати необхідні функції, переглядати

листування з користувачами та багато іншого. На етапах автоматизації бота ви можете додавати дії та елементи, які спрощують розробку.

Чат з користувачами можна переглянути в будь-який час. Це дозволяє аналізувати логіку роботи та комунікаційну ефективність. Також компанія може отримувати аналітичні дані у звітах: кількість користувачів, характеристики продуктивності, ступінь залучення користувачів.

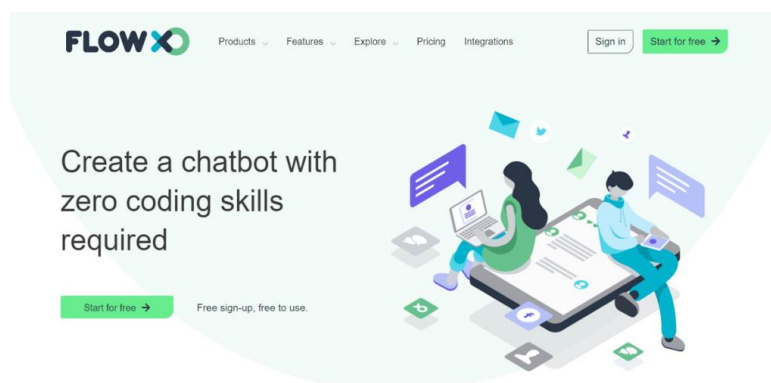


Рисунок 2.6 – Головний екран Flow XO платформи

Особливості Flow XO:

- Редактор логіки робота.
- Веб-віджет для просування робіт.
- Вбудований веб-месенджер.
- Шаблони.
- Розмови з користувачами через email/Службу підтримки.
- Збереження атрибутів користувача.
- Отримання та надсилання даних через HTTP / JSON.
- Розподіл контенту RSS для користувачів.
- Push-повідомлення користувачам на вимогу.
- Отримання зображень, розташування та інших файлів.
- Початок взаємодії прямими повідомленнями, згадками.
- Тригери.
- Запитання та перевірка формату відповідей.

- Типи питань.
- Фільтри.
- Виконує JavaScript, текст, номери та дати.
- Підтримка кількох мов. Це дані з товариства Стартпак.
- Інтеграція з популярними месенджерами: Slack, Telegram, WhatsApp, Messenger та іншими сервісами.
- Доступний безкоштовний тариф з обмеженнями, та платні, від 19\$ на місяць.

2.4.2 Botsify

Botsify — це платформа, де люди та компанії можуть створювати власні інтелектуальні чат-боти, не знаючи мови програмування. Англomовний сервіс для створення чат-ботів із використанням штучного інтелекту.

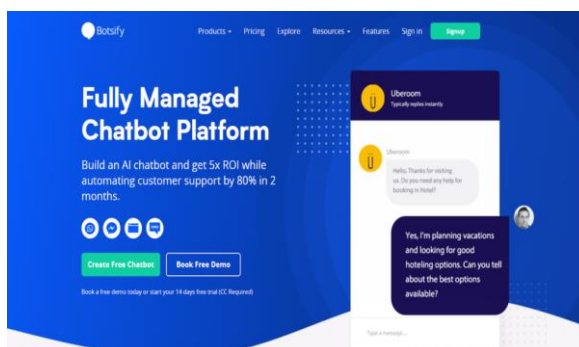


Рисунок 2.7 – Головний екран Botsify платформи

Основні можливості:

- візуальний онлайн-конструктор чат-ботів;
- навчання роботів за допомогою алгоритмів машинного навчання;
- підбір схожих фраз із запитів користувачів;
- підтримка швидких відповідей;
- збір та кваліфікація лідів;
- підтримка 190 мов;

- підключення оператора до діалогу.
- WhatsApp, Facebook Messenger, Telegram, SMS можна встановити як чат на сайті.

Вартість: мінімальний тарифний план – 49 \$/міс. На цьому плані можна створити 2 чат-боти, доступні три платформи (Facebook, SMS та сайт), кількість користувачів – до 5000. Вартість тарифного плану з доступом до всіх платформ та безліміту за кількістю користувачів – 499 \$/міс.

2.4.3 Dexter

Англомовний сервіс пропонує платформу для створення чат-ботів. Головна фішка платформи - роботи створюються тут не звичним конструктором за допомогою додавання і перетягування блоків, а в текстовому редакторі. Як вважають самі творці сервісу, це дозволяє максимально просто створювати ботів (не складніше, ніж написати коротку історію), а уникнення блоків і схем дає можливість не перевантажувати користувачів.

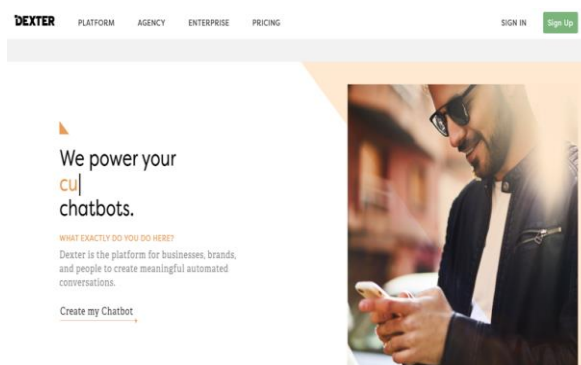


Рисунок 2.8 – Головний екран Dexter платформи

Основні можливості:

- надсилання повідомлень за заданим сценарієм;
- ведення та зберігання історії повідомлень з користувачами;
- збір та обробка даних користувачів;

- відправлення файлів (відео, аудіо, картинки);
- сегментація розсилок;
- розсилання повідомлень та повідомлень за розкладом;
- Slack, Facebook Messenger, Twilio, Twitter, Alexa, Viber, сайт.

Вартість: мінімальний тарифний план – 20\$/міс. Варто відзначити, незважаючи на заяви творців платформи про простоту текстового створення роботів, для більшості користувачів візуальні конструктори все-таки будуть звичнішими і зрозумілішими.

2.4.4 Gerabot

Український онлайн-конструктор для створення та управління чат-ботами. Також у Gerabot можна замовити створення чат-бота під завдання бізнесу.

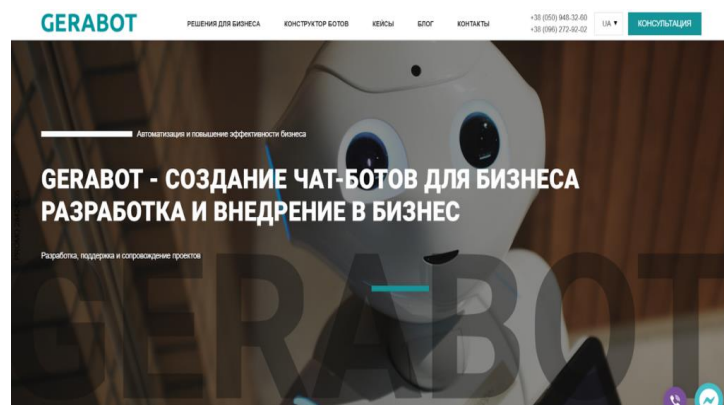


Рисунок 2.9 – Головний екран Gerabot платформи

Основні можливості:

- налаштування сценаріїв взаємодії з користувачами;
- підключення оператора до чату з користувачем (за запитом користувача або за заданим сценарієм);
- прийом заявок від користувачів;
- консультування клієнтів та надання потрібної інформації.

- Viber, Telegram, Facebook, сайт.
- вартість - від 480 грн. на місяць.

2.4.5 Chatbot

Англомовна платформа для створення чат-ботів під завдання основних напрямів бізнесу (маркетинг, продаж, сапорт клієнтів).

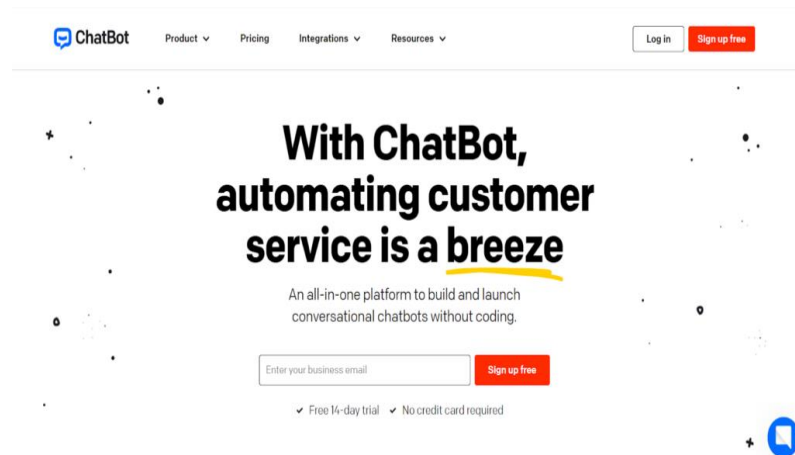


Рисунок 2.10 – Головний екран Chatbot платформи

Основні можливості:

- візуальний конструктор типу drag-and-drop;
- можливість тестування готового робота прямо в інтерфейсі платформи;
- кастомізація віджету чату для сайту;
- збір даних про користувачів, зберігання та сегментація бази;
- приймання замовлень;
- надсилання зображень та інших типів файлів;
- кнопки для швидких відповідей;
- аналіз повідомлень та навчання за допомогою алгоритмів штучного інтелекту;
- база готових шаблонів для типових сценаріїв, шаблони можна

редагувати та кастомізувати під себе.

- Facebook Messenger, Slack, LiveChat, сайт.

Вартість: мінімальний тариф - 50 \$ / міс. У ньому можна створити один чат-бот, доступно 1000 чатів з користувачами на місяць. На мінімальному тарифі функціонал доступний не в повному обсязі (наприклад, немає можливості додавати кнопки для швидких дій).

2.4 Порівняння платформ для створення чат-ботів

У роботі було розглянуто 5 інструментів для створення чат-ботів, що найбільш широко представлені на світовому ринку. Ступінь проникнення кожного продукту на ринок оцінювалася за такими ознаками, як кількість публікацій із згадкою платформи, кількість публічних кейсів щодо її використання, частота згадок у професійній спільноті, а також за результатами аналізу думки експертів ринку.

В результаті більш детального ознайомлення з представленими сервісами, та вивчення низки інших менш популярних сервісів, але не менш хороших у плані функціоналу та реалізації. Було зроблено докладний аналіз, розглянуті платформи можна розділити на два типи:

- Рішення, які не вимагають знань програмування та містять візуальний конструктор потоків

- Інструменти для розробників без візуальних конструкторів діалогів.

У відсотковому співвідношенні рішень, які не вимагають знань програмування у 2 рази більше, ніж других. Більшість рішень, випущених у 2017-2022 роках — з візуальним інтерфейсом, створеним спеціально для нефахівців.

Це пов'язано з тим, що чат-боти набирають все більшої популярності: цей інструмент взаємодії з клієнтами подобається споживачам і допомагає компаніям економити на контактних центрах. Їх освоюють навіть невеликі компанії, у штаті яких немає розробників для створення власних рішень, як

бюджету на аутсорсинг розробки.

Таблиця 2.1 – Порівняння платформ для створення чат-ботів

SERVICE	Link	Чи можна інтегрувати з Google Dialog Flow або Microsoft QNA Maker*	Process visualization	Ready templates	Free period	Integrations or your own CRM	Communication platforms	Additional functionality
Flow XO	https://flowxo.com/	https://flowxo.com/integrations/	блок-схеми	шаблони сценаріїв	безстроково з обмеженнями	100+ інтеграцій, у тому числі кілька CRM	Facebook Messenger, Telegram, Twilio SMS, Clack, сайт	Аналітика взаємодій із користувачами
Botsify	https://botsify.com/	https://botsify.com/integrations	блок-схеми	шаблони для різних ніш	14 днів	через Zapier	Сайт, WhatsApp, Facebook Messenger, SMS	Статистика та аналітика, послуги розробників
Dexter	https://rundexter.com/	-	текстовий редактор	готові шаблони	14 днів	через API	Slack, Facebook Messenger, Twilio, Twitter, Alexa, Viber, сайт	Статистика та аналітика, розсилки по базі користувачів, які взаємодіяли з ботом
Gerabot	https://gerabot.com/costructor	-	блок-схеми	готові рішення за окрему платню	1 місяць	своя CRM	Viber, Telegram, Facebook, сайт	Послуги розробників
Chatbot	https://www.chatbot.com/	https://www.chatbot.com/chatbot-templates/livechat-essentials-template/#template-features	блок-схеми	готові шаблони	14 днів	через API або через Zapier	Facebook Messenger, Slack, LiveChat, сайт	Аналіз продуктивності, А/Б-тестування

Висновки дослідження:

- Серед найбільш популярних і функціональних інструментів обробки природної мови можна виділити IBM Watson, Amazon Lex, Microsoft LUIS, Google Dialogflow, Wit.ai, Rasa.

- За 2-3 роки з'явилося безліч локальних продуктів для розробки чат-ботів з мовною специфікою: Recast.AI у Франції, Xenioo в Італії, DeepPavlov.ai, AgentBot в Аргентині, Botsify у Пакистані, Engati та Morph.ai в Індії та інші. За функціональним наповненням і якістю вони можуть конкурувати з великими міжнародними вендорами, переважаючи їх із погляду глибшого опрацювання мовних і культурних особливостей.

- Є ознаки консолідації цього ринку, тобто вступу його до початкового етапу процесу дозрівання. Так, кілька стартапів були придбані більшими компаніями. Ari.ai був поглинений компанією Google, Semantic Machines - Microsoft, Motion.ai куплений Hubspot, KITT.AI - Baidu, ChattyPeople -

компанією MobileMonkey.

- Рішення різняться за рівнем свободи, яку має розробник. Так, платформа Imperson дозволяє створити чат-бота з унікальною зовнішністю, який розмовляє бажаним голосом. По суті такий віртуальний персонаж стає обличчям бренду, що відображає його цінності. Створення більш персоналізованих та емоційних чат-ботів може стати однією з тенденцій майбутнього.

- Переважна більшість популярних чат-бот платформ має інструменти моніторингу різних метрик, таких як: динаміка кількості звернень динаміка кількості користувачів, часті теми та ключові слова звернень, інструменти перегляду невдалих діалогів.

Є кілька тенденцій, які матимуть значний вплив на розвиток індустрії чат-ботів у найближчому майбутньому. Перш за все, це збільшення частки комбінованих рішень, у яких робот не повністю замінює роботу людини, а доповнює її повторними рутинними діями. Асистенти оператора, інтегровані з системами RPA (Robotic Process Automation), виглядають найбільш перспективними в найближчі кілька років. Другий напрямок — розробка інструментів для швидкого отримання знань і побудови онтологій на неструктурованих даних. Іншими словами, це системи, в які можна завантажити набір різних текстів, а вони будуть самостійно витягувати з них семантичні зв'язки і будувати мовні моделі, характерні для даної предметної області. Наприклад, за допомогою таких інструментів можна буде швидко навчити роздрібного чат-бота розрізняти продовольчі та непродовольчі товари. Третя тенденція — швидкий обмін знаннями між роботами. І, нарешті, за кілька років ми побачимо збільшення частки більш персоналізованих віртуальних помічників з унікальною індивідуальністю та адаптованих до конкретного клієнта.

3 ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА

Вимоги до проектування чат-бота включають точне представлення знань, стратегію генерації відповідей і набір попередньо визначених нейтральних відповідей, щоб відповісти, коли висловлювання користувача не зрозуміло [18, 19] . Першим кроком у проектуванні будь-якої системи є її розділення на складові частини відповідно до стандарту, щоб можна було дотримуватися модульного підходу до розробки [20]. На рисунку 3.1 представлено загальну модель чат-бота.

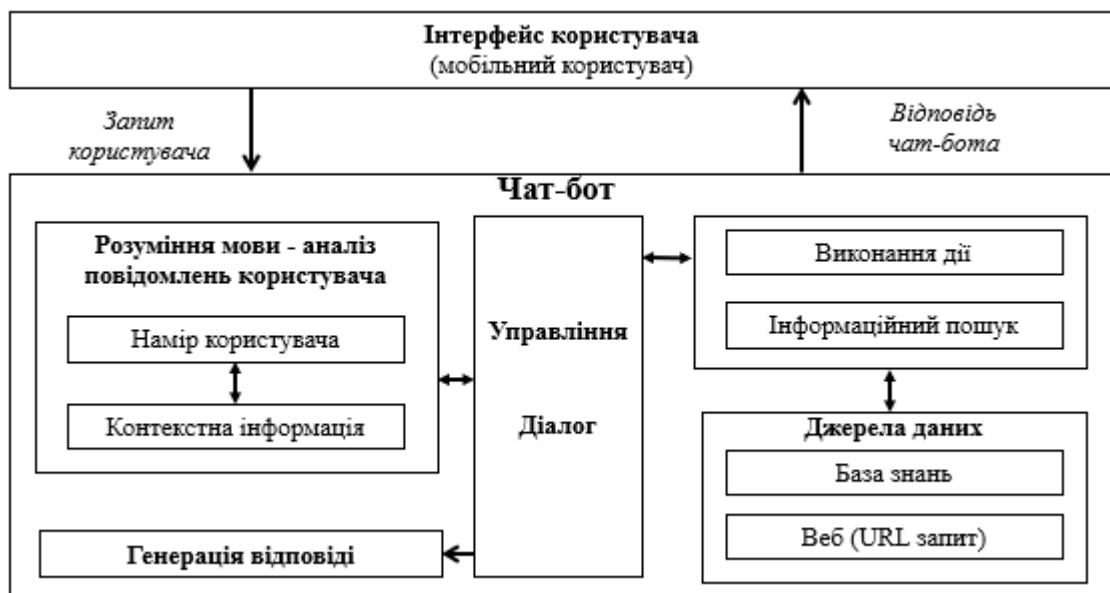


Рисунок 3.1 – Загальна модель чат-бота

Процес починається із запиту користувача, наприклад «Що означає середовище?», до чат-бота за допомогою програми обміну повідомленнями, як-от Facebook, Slack, WhatsApp, WeChat або Skype, або програми, що використовує текстове чи мовленнєве введення, як-от Amazon Echo [21].

Після того, як чат-бот отримує запит користувача, компонент розуміння мови аналізує його, щоб визначити намір користувача та пов'язану інформацію (намір: «перекласти», сутності: [слово: «середовище»]) [22] .

Коли чат-бот досягає найкращої інтерпретації, він повинен визначити, як діяти далі [23]. Він може безпосередньо діяти на основі нової інформації, запам'ятовувати все, що він зрозумів, і чекати, щоб побачити, що станеться далі, вимагати більше контекстної інформації або запитувати роз'яснення.

Коли запит зрозумілий, відбувається виконання дії та пошук інформації. Чат-бот виконує запитані дії або отримує цікаві дані зі своїх джерел даних, якими може бути база даних, відома як База знань чат-бота, або зовнішні ресурси, доступ до яких здійснюється через виклик API [24].

Після отримання компонент генерації відповідей використовує генерацію природної мови (NLG) [25], щоб підготувати людиноподібну відповідь природною мовою для користувача на основі інформації про наміри та контекст, яку повертає компонент аналізу повідомлень користувача.

Компонент керування діалогом зберігає та оновлює контекст розмови, який є поточним наміром, ідентифікованими об'єктами або відсутніми об'єктами, необхідними для виконання запитів користувача. Крім того, він запитує відсутню інформацію, обробляє пояснення користувачів і задає додаткові запитання. Наприклад, чат-бот може відповісти: «Чи не хочете ви також сказати мені приклад речення зі словом середовище?» [26].

Для розробки чат-бота доступно багато комерційних і відкритих варіантів. Кількість технологій, пов'язаних із чат-ботами, вже є надзвичайною та зростає з кожним днем [27]. Чат-боти розробляються двома способами: за допомогою будь-якої мови програмування, як-от Java, Clojure, Python, C++, PHP, Ruby та Lisp, або за допомогою найсучасніших платформ. Найбільш поширені шість провідних хмарних платформ NLU, які розробники можуть використовувати для створення додатків, здатних розуміти природні мови: DialogFlow від Google, Microsoft LUIS, IBM Watson Conversation, Amazon Lex і SAP Conversation AI. Усі ці платформи підтримуються машинним навчанням. Вони поділяють деякі стандартні функції (вони базуються на хмарі, підтримують різні мови програмування та

природні мови), але суттєво відрізняються в інших аспектах [28 – 30]. Іншими відомими платформами для розробки чат-ботів є React.js, JSX, REACT-HOOKS, REDUX.

3.1 Платформи для розробки чат-ботів

Сучасні сайти відрізняються від тих, якими ми користувалися раніше: вони динамічніші. З їхньою допомогою можна купувати товари, спілкуватися з друзями чи писати коментарі.

Змінилася структура сайтів. Спочатку вони склалися із набору окремих HTML-сторінок. Зараз вони є одним HTML-документом, зовнішній вигляд якого змінюється динамічно через JavaScript.

Такий підхід отримав назву SPA - single-page application, або односторінковий додаток. SPA робить сайти повноцінними програмами, які не перезавантажуються між переходами по сторінках. Щоб виконати перехід, потрібно повністю перемалювати контент документа за допомогою JavaScript та отриманих даних.

Подібні складності призвели до створення реактивних фреймворків та бібліотек. Ідея реактивності така: замість того, щоб щоразу вручну в коді в імперативному вигляді змінювати інтерфейс програми, набагато зручніше спочатку в декларативному вигляді задати зв'язок між даними та їх відображенням і надалі код змінювати тільки дані. Інтерфейс автоматично перемальовуватиметься під нові дані.

3.1.1 Фреймворк react.js

Тобто реактивний підхід передбачає роботу з даними, при зміні яких автоматично змінюються інші частини програми, зокрема інтерфейс. React став однією з перших та найпопулярнішою на початок 2022 року реактивною бібліотекою для роботи з інтерфейсами.

React (React.js) — JavaScript-бібліотека з відкритим вихідним кодом для розробки інтерфейсів користувача.

React розробляється та підтримується Facebook, Instagram та спільнотою окремих розробників та корпорацій.

React може використовуватися для розробки односторінкових та мобільних додатків. Його мета — надати високу швидкість розробки, простоту та масштабованість. Як бібліотека для розробки інтерфейсів користувача React часто використовується з іншими бібліотеками, такими як MobX, Redux і GraphQL.

React був створений Джорданом Валке, одним із важливих розробників з Facebook. На нього вплинув XHP - HTML-фреймворк для PHP. Вперше React використовувався в стрічці новин Facebook в 2011 році і пізніше в стрічці Instagram в 2012 році. Вихідний код React відкрито у травні 2013 року на конференції «JSConf US».

React Native анонсовано на конференції Facebook React.js Conf у лютому 2015 року, а вихідний код відкритий у березні 2015 року. Він дозволяє розробляти нативні Android-, iOS- та UWP-додатки з використанням React.

18 квітня 2017 року Facebook анонсував React Fiber, переписану та оптимізовану версію React. React Fiber стане основою розробки всіх майбутніх функцій та покращень.

3.1.2 JSX

React враховує той факт, що логіка рендерингу нерозривно пов'язана з іншою логікою інтерфейсу користувача: як обробляються події, як стан змінюється з часом і як дані готуються до відображення.

Замість того, щоб штучно розділяти технології, поміщаючи розмітку та логіку в окремі файли, React поділяє завдання між так званими одиницями або просто компонентами, які містять і те, й інше.

Використовувати JSX не є обов'язковою вимогою, але як і більшість людей я вважають його корисним як наочний посібник при роботі з інтерфейсом користувача всередині коду JavaScript. Це також дозволяє React відображати більш корисні повідомлення про помилки та попередження.

Функціональний компонент у React - це JavaScript-функція, яка на вході отримує об'єкт з даними для відтворення компонента - їх називають props, а на виході повертає опис інтерфейсу, який потрібно відобразити (рис.3.2).

```
import React, { FC } from 'react' 6.9k (gzipped: 2.7k)

interface IHelloWorld {
  name: string;
}

const HelloComponent: FC<IHelloWorld> = ({ name }) => {
  return (
    <div>Hello my name is { name } </div>
  );
};

export default HelloComponent;
```

Рисунок 3.2 – Приклад вітального компонента у React

Цей код легко розібратися, навіть якщо ви не знаєте синтаксису React. Тут як функції описаний React-компонент HelloComponent. На вході він отримує об'єкт із даними, що містить властивість name з ім'ям користувача. Функція повертає спеціальний код, схожий на розмітку HTML. React дійсно відобразить на сторінці DOM-дерево – воно відповідатиме розмітці, яку повернуло HelloComponent. Але на відміну від звичайного HTML ми змогли підставити в розмітку дані з об'єкта {name}.

Код на React виглядає легшим і лаконічнішим рішенням на чистому JS. Ми не описуємо як перемальовувати інтерфейс. Натомість ми вказуємо, що

потрібно відобразити на місці компонента, і використовуємо для цього синтаксис, схожий на HTML. Такий підхід дозволяє відносно легко вирішувати і складніші завдання.

Для перемальовок React використовує Virtual DOM. Це дозволяє оновлювати тільки вузли DOM-дерева, що змінилися, а не перемальовувати всю сторінку. Завдяки цьому програми не витрачають зайві ресурси.

React вирішує проблеми фронтендерів при розробці інтерфейсів динамічних сайтів та SPA-додатків. При використанні класичної зв'язки JavaScript та HTML перемальовка інтерфейсу дуже ускладнюється. React не обмежує можливості розробника, на відміну від будь-яких фреймворків.

3.1.3 React-hooks

Коли спільнота вперше познайомилася з функціональними компонентами, вони служили лише для того, щоб виводити інформацію. Вони не мали ні стану, ні методів життєвого циклу. Вони були дуже простими – у цьому й полягала їхня проблема. Часто виникала ситуація, що компонент, написаний у функціональному стилі, мав надалі мати стан чи методи життєвого циклу, а такої можливості не було. Доводилося переписувати їх у класові компоненти, а це далеко не найцікавіше заняття для розробника.

Такий стан справ підштовхнув розробників React до створення хуків, що дозволяють розширити можливості функціональних компонентів або нівелювати деякі проблеми, які можуть виникати через їхню специфіку. Хуки виявилися настільки зручними, що стали основою React-розробки. Розглянемо докладніше.

Почнемо з найпростішого і найважливішого хука - useState. З назви стає зрозуміло, що він пов'язаний зі станом компонента. Саме завдяки йому у функціональних компонентів виник стан (рис.3.3).

```

2   import React, { useState } from 'react'; 6.9k (gzipped: 2.7k)
3
4   function Example() {
5     const [count, setCount] = useState(0);
6     return (
7       <div>
8         <p>You clicked {count} times</p>
9         <button onClick={() => setCount(count + 1)}>
10          Click me
11        </button>
12      </div>
13    );
14  }

```

Рисунок 3.3 – Приклад хука useState

Якщо ми подивимось на п'яту строчку код, то побачимо як у ній створюється стан та метод, який змінюватиме це значення. Хук useState по суті приймає в якості параметра початкове значення, тобто на початковому етапі нашої величини буде мати значення 0. І повертає useState масив з двох елементів: перший - стан, другий - метод, який буде його змінювати. Розробники хуків використовували досить витончений підхід. При використанні деструктуризації він дозволяє задати будь-яке значення стану та методу мінімальною кількістю коду.

Щоб передати якісь дані компонент, ми можемо використовувати props. Але є альтернативний спосіб – хук useContext.

Якщо ви раніше його не використовували, контекст дозволяє передавати дані від батьківського компонента до дочірнього, минаючи проміжні.

Щоб було зрозуміліше, створил невеликий приклад, який дозволить зрозуміти його роботу. У нас є три компоненти. Перший з них Top – зовнішній, другий –Middle, тобто проміжний, а третій назовемо Last, і він буде внутрішнім. По суті всі вони будуть вкладені один в одного. Наше завдання – передача даних із компонента Top до компонентуLast, минаючи Middle, оскільки до нього ці дані стосунку не мають (рис.3.4).

```

1
2 import {createContext, useContext} from "react"; 4.4k (gzipped: 1.9k)
3
4 const MyContext = createContext("without provider");
5
6 const Top = () => {
7   return (
8     <MyContext.Provider value="Hello, i am External">
9       <Intermediate />
10    </MyContext.Provider>
11  );
12 };
13
14 const Middle = () => {
15   return <Internal />;
16 };
17
18 const last = () => {
19   const context = useContext(MyContext);
20
21   return `I am Internal component. I have got the message from External: "${context}"`;
22 };
23

```

Рисунок 3.4 – Приклад хука useContext

Щоб використовувати контекст, ми створюємо об'єкт MyContext, викликаючи метод createContext. У компоненті Top обертаємо компонент Middle компонент MyContext.Provider. Тим самим говоримо, що всі вкладені в нього компоненти зможуть отримати доступ до даних, які ми передаємо, розміщуючи їх у параметрі value.

Причому вони будуть доступні лише в тих компонентах, які це потрібно. Для цього ми повинні використовувати хук useContext, а як аргумент у нього буде об'єкт MyContext. Хук useContext поверне нам дані, передані в параметр value MyContext.Provider, які ми помістимо в змінну context. Його значення потрапить у змінну context у тому випадку, якщо ви раптом забудете створити обгортку MyContext.Provider, тобто він допоможе не допустити помилку через неухважність.

Завдяки хуку useContext можна використовувати context у функціональних компонентах, і дані будуть попадати лише в ті компоненти, у яких вони потрібні. Також він позбавить проблеми з drops drilling.

Якщо ви працювали із класовими компонентами, то знайомі з методами

життєвого циклу. Вони служать у тому, щоб здійснювати якісь операції різних стадіях життя компонента. Для цього у нас є два хуки - `useEffect` і `useLayoutEffect`. Вони схожі між собою, за винятком невеликої різниці у рендерингу. Що стосується `useLayoutEffect` React не запускає рендеринг побудованого DOM дерева до того часу, поки не відпрацює `useLayoutEffect`. Якщо ж ми беремо `useEffect`, то React відразу запускає рендеринг побудованого DOM, не чекаючи запуску `useEffect`.

За допомогою цих двох хуків у функціональних компонентах можна змоделювати роботу трьох методів життєвого циклу – `componentDidMount`, `componentDidUpdate`, `componentWillUnmount`. Більш точно їх робота імітує використання `LayoutEffect`, тому що в класових компонентах малювання DOM-дерева не запускається до тих пір, поки не відпрацює метод `componentDidMount`.

Оскільки ці два хуки мають один і той же інтерфейс, продемонструємо його на популярнішому хуку - `useEffect`, а для іншого все буде аналогічно.

`useEffect` приймає два аргументи:

- `callback`. У середині нього все корисне навантаження, яке ми хочемо описати. Наприклад, можна робити запити на сервер, завдання обробників подій на документ або ще щось;
- масив, що з аргументів. При зміні значення всередині будь-якого з них запускатиметься наш `callback`. Саме завдяки цьому аргументу ми можемо імітувати методи життєвого циклу.

Розглянемо імітацію `componentDidMount`. Хук `useEffect` запускається не тільки при зміні елементів масиву з другого аргументу, але й після монтування компонента. Фактично `componentDidMount` запускається на тій самій стадії. Якщо ми вкажемо як другий аргумент порожній масив, `callback` запускатиметься на стадії монтування компонента. А оскільки жодних залежностей для хука всередині масиву ми не поставили, то аргумент `callback` не запускатиметься більше (рис. 3.5).

```

1
2  const App = ({data}) => {
3    useEffect(() => {
4      console.log("componentDidMount");
5    }, []);
6
7    return null;
8  };
9
10

```

Рисунок 3.5 – Імітація componentDidMount

Також можлива імітація компонента DidUpdate. Але у випадку з гачками є кращий спосіб. Оскільки в масиві ми можемо вказати тільки ті залежності, які нам потрібні, у нас є більш гнучкий аналог методу componentDidMount, який запускається при зміні необхідних параметрів. Якщо ви хочете зробити аналогію з componentDidMount, ви можете вказати всі параметри та стани як залежності. Важливо мати на увазі, що useEffect також запускається на етапі монтування (рис.3.6).

```

1
2  const App = ({data}) => {
3    useEffect(() => {
4      console.log("componentDidUpdate");
5    }, [data]);
6
7    return null;
8  };
9

```

Рисунок 3.6 - Імітація componentDidUpdate

Тепер розглянемо імітацію компонента WillUnmount. Для цього просто повертаємо з зворотного виклику useEffect. Практично повернутий зворотний виклик – це і є аналог компонента WillUnmount, який часто використовується для відключення обробників файлів документа. У випадку з функціональним

компонентом ми будемо відв'язувати їх всередині зворотного виклику (рис.3.7).

```
1
2  const App = ({data}) => {
3    useEffect(() => {
4      return () => {
5        console.log("componentWillUnmount");
6      };
7    }, []);
8
9    return null;
10 };
11
```

Рисунок 3.7 – Імітація WillUnmount

Бувають ситуації, коли потрібно звернутися до якогось DOM-об'єкта безпосередньо. Для цього існує хук useRef (рис. 3.8).

```
2  const App = () => {
3    const ref = useRef();
4
5    useEffect(() => {
6      console.log(ref.current);
7    }, []);
8
9    return <div ref={ref} />;
10 };

```

Рисунок 3.8 – Приклад хука useRef

Ми створюємо об'єкт ref і вказуємо його як елемент, що означає DOM-об'єкт, до якого ми хочемо звернутися, а також прописуємо цей об'єкт як параметр. Далі ми можемо взаємодіяти з Dom-об'єктом безпосередньо, якби ми знайшли його за допомогою селектора. І тому використовуємо властивість current у об'єкта ref.

Крім цього, можна застосувати useRef, якщо нам потрібно буде запам'ятати дані в компоненті. Але не рекомендуємо викликати рендерингу у разі зміни стану компонента.

3.1.4 Redux

Redux - це інструмент для керування станом даних і інтерфейсом користувача в додатках JavaScript з великою кількістю сутностей. Являє собою бібліотеку JavaScript.

Назва складається з двох слів: reduce та flux. Reduce - це функція, яка наводить велику структуру даних до одного значення. Flux – передача даних на одну сторону. Редакс працює ґрунтуючись на цих двох поняттях – тому вони винесені у назву.

Зазвичай Redux використовується у зв'язці з фреймворками для JavaScript: React, TypeScript, Vue, Angular та іншими. Рідше він буває необхідний написання коду на чистому JS. Має відкритий вихідний код та доступний безкоштовно. З усіма залежностями важить лише близько 2 Кб.

Механізм локального зберігання компонентів, що йде в комплекті з базовою бібліотекою React, незручний у використанні, оскільки таке сховище ізольоване. Наприклад, якщо ви хочете, щоб різні незалежні компоненти реагували на певну подію, вам доведеться поширити локальний стан на дочірні компоненти або підняти його до найближчого батьківського компонента. В обох випадках це незручно, та залишає після себе поганий код з, так званим “props-drilling”. Код стає більш заплутаним і важким для читання, а компоненти стають залежними від їхньої вкладеності. Redux усуває цю проблему, тому що всі стани легко доступні для всього проекту.

Redux - це універсальний інструмент розробки, який можна використовувати у поєднанні з різними бібліотеками та фреймворками.

Архітектура Redux обертається навколо односпрямованого потоку даних.

Це означає, що всі дані у вашій програмі дотримуються однакового шаблону життєвого циклу, що робить логіку вашої програми більш передбачуваною та зрозумілою. Це також сприяє більшій нормалізації даних, щоб у вас не було кількох втрачених копій тих самих даних, які нічого не

знають одна про одну.

Життєвий цикл даних у будь-якому Redux-додатку включає 4 кроки:

- Ви викликаєте `store.dispatch(action)`. Екшен - це простий javascript-об'єкт, який описує, що трапилося (рис 3.9).

```
2
3 { type: 'LIKE_ARTICLE', articleId: 42 }
4 { type: 'FETCH_USER_SUCCESS', response: { id: 3, name: 'Mary' } }
5 { type: 'ADD_TODO', text: 'Read the Redux docs.' }
```

Рисунок 3.9 – Приклад action в Redux

Екшен можна викликати `store.dispatch(action)` з будь-якого місця в додаток, включаючи компоненти та колбеки або навіть із запланованими інтервалами.

- Redux-store викликає функцію-редюсер, який ви йому передали. Store передасть два аргументи при виклику редюсера: поточне дерево стану (current state tree) та екшен (action) (рис.3.10).

```
1
2 let previousState = {
3   visibleTodoFilter: 'SHOW_ALL',
4   todos: [
5     {
6       text: 'Read the docs.',
7       complete: false
8     }
9   ]
10 }
11
12 let action = {
13   type: 'ADD_TODO',
14   text: 'Understand the flow.'
15 }
16
17 let nextState = todoApp(previousState, action)
```

Рисунок 3.10 – Приклад роботи store в Redux

- Головний редюсер може комбінувати результат роботи кількох редюсерів у єдине дерево стану програми.

- Як структурувати головний редюсер - вибір кожного. Redux поставляється з хелпером `combineReducers()`, корисним для поділу головного редюсера на окремі функції, які керують окремими гілками стану дерева, `combineReducers()` працює наступним чином (рис. 3.11).

```
2 function todos(state = [], action) {
3   return nextState
4 }
5
6 function visibleTodoFilter(state = 'SHOW_ALL', action) {
7   return nextState
8 }
9
10 let todoApp = combineReducers({
11   todos,
12   visibleTodoFilter
13 })
14
15 let nextTodos = todos(state.todos, action)
16 let nextVisibleTodoFilter = visibleTodoFilter(state.visibleTodoFilter, action)
```

Рисунок 3.11 – Приклад `combineReducers()`

Коли ми викликаємо екшен, `todoApp`, який повернув `combineReducers`, викличе обидва редюсери. Потім обидва набори станів знову зібрані в єдиний стан. Так як `combineReducers()` - це просто зручна утиліта, ми не зобов'язані її використовувати і можемо написати головний редюсер самостійно!

- `Redux-stor` зберігає повне дерево стану, яке повертає головний редюсер. Це нове дерево є наступним станом нашої програми. Кожного слухача, зареєстрованого за допомогою `store.subscribe(listener)`, буде викликано. Слухачі можуть викликати `store.getState()` для отримання поточного стану програми. Тепер UI може бути оновлено для відображення нового стану програми. Якщо ми використовуємо такі біндинги, як `React Redux`, то це та точка, в якій варто викликати `component.setState(newState)` (рис. 3.12).

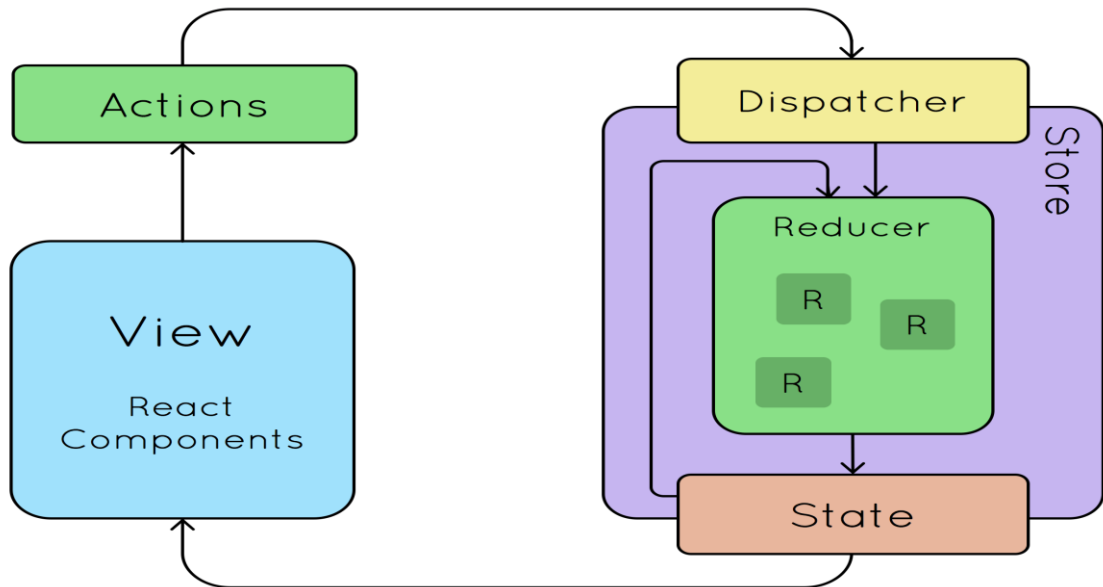


Рисунок 3.12 – Життєвий цикл компонента в Redux

Redux Toolkit це набір практичних рішень та методів, покликаних спростити розробку додатків за допомогою Redux. Мета, яку мають розробники цієї бібліотеки, - спростити типові варіанти використання Redux. Цей інструмент не є універсальним рішенням для всіх можливих варіантів використання Redux, але спрощує код, який необхідно писати розробникам.

Redux Toolkit виконує такі функції:

- допомагає швидко розпочати використання Redux;
- спрощує роботу з типовими завданнями та кодом Redux;
- дозволяє використовувати найкращі практики Redux за замовчуванням;
- пропонує рішення, які зменшують недовіру до бойлерплейтів.

Redux Toolkit надає набір як спеціально розроблених, так і додає ряд інструментів, що добре себе зарекомендували, які зазвичай використовуються спільно з Redux. Такий підхід дозволяє розробнику вирішити, як і які інструменти використовувати у своєму додатку.

Найбільш значущими функціями, які надає бібліотека Redux Toolkit є:

- `configureStore` - функція, призначена спростити процес створення та налаштування сховища;
- `createReducer` - функція, що допомагає лаконічно та зрозуміло описати та створити ред'юсер;
- `createAction` - повертає функцію творця дії для заданого рядка типу дії;
- `createSlice` - поєднує в собі функціонал `createAction` і `createReducer`;
- `createSelector` — функція з бібліотеки `Reselect`, яка переекспортована для простоти використання.

Функція `configureStore` дозволяє автоматично комбінувати ред'юсери, додати міدلвари `Redux` (за замовчуванням включає `redux-thunk`), а також використовувати розширення `Redux DevTools`. Як вхідні параметри функція `configureStore` приймає об'єкт з такими властивостями:

- `reducer` — набір ред'юсерів користувача,
- `middleware` - опціональний параметр, що задає масив міدلварів, призначених для підключення до сховища,
- `devTools` — параметр логічного типу, що дозволяє увімкнути встановлене у браузер розширення `Redux DevTools` (значення за промовчанням — `true`),
- `preloadedState` - опціональний параметр, що задає початковий стан сховища,
- `enhancers` - опціональний параметр, що задає набір підсилювачів.

Функція `createAction`. Спочатку окремо оголошується константа, що визначає тип дії, після чого функція творця дії цього типу. Функція `createAction` об'єднує ці об'яви в одне. На вхід вона приймає тип дії та повертає творця дії для цього типу. Автор дії може бути викликаний або без аргументів, або з деяким аргументом (корисне навантаження), значення якого буде поміщено в полі `payload`, створеної дії.

Функція `createReducer`. У деяких випадках може знадобитися написати додаткову логіку для налаштування значення корисного навантаження,

наприклад, прийняти кілька параметрів для творця дії, створити випадковий ідентифікатор або отримати поточну мітку. Для цього `createAction` приймає необов'язковий другий аргумент - функцію, яка використовуватиметься для оновлення значення корисного навантаження.

Редюсери часто реалізуються за допомогою оператора `switch`, з одним регістром для кожного обробленого типу дії. Цей підхід працює добре, але не позбавлений бойлерплейта і схильний до помилок. Наприклад, легко забути описати випадок `default` або встановити початковий стан. Функція `createReducer` спрощує створення функцій ред'юсера, визначаючи їх як таблиці пошуку функцій обробки кожного типу дії. Вона також дозволяє суттєво спростити логіку імутабельного поновлення, написавши код у мутабельному стилі всередині ред'юсерів.

Мутабельний стиль обробки подій доступний завдяки використанню бібліотеки `Immer`. Функція обробник може або "мутувати" переданий `state` для зміни властивостей, або повертати новий `state`, як при роботі в імутабельному стилі, але завдяки `Immer` реальна мутація об'єкта не здійснюється. Перший варіант набагато простіше для роботи та сприйняття, особливо при зміні об'єкта з глибокою вкладеністю.

Функція `createSlice`. Як вхідні параметри функція `createSlice` приймає об'єкт із наступними полями:

- `name` - простір імен створюваних дій (`${name}/${action.type}`);
- `initialState` - початковий стан ред'юсера;
- `reducers` – об'єкт з обробниками. Кожен обробник приймає функцію з аргументами `state` і `action`, `action` містить у собі дані у властивості `payload` та ім'я події у властивості `name`. Крім того, є можливість попередньої зміни даних, отриманих з події, перед їх потраплянням до ред'юсера (наприклад, додати `id` до елементів колекції). Для цього замість функції необхідно передати об'єкт з полями `reducer` та `prepare`, де `reducer` – це функція-обробник дії, а `prepare` – функція-обробник корисного навантаження, що повертає оновлений `payload`;

- `extraReducers` - об'єкт, що містить редьюсери іншого зрізу. Цей параметр може знадобитися в разі потреби оновлення об'єкта, що стосується іншого зрізу. Докладніше про цю можливість можна дізнатися з відповідного розділу офіційної документації.

Бібліотека `Redux Toolkit` не вносить нічого нового в управління сховищем, вона надає ряд набагато зручніших засобів для написання коду, ніж були до цього. Дані засоби дозволяють не тільки зробити процес розробки більш зручним, зрозумілим і швидким, але і більш ефективним, за рахунок наявності в бібліотеці ряду інструментів, що добре зарекомендували себе раніше.

4 МОДЕЛЮВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ЧАТ-БОТА

Завдяки швидкому розвитку штучного інтелекту, зокрема обробки природної мови, мовлення є перспективним способом взаємодії між людьми та комп'ютерами. Голосові помічники або голосові агенти (ГА) забезпечують мовну взаємодію з існуючими пристроями (наприклад, смартфонами) і служать основним інтерфейсом нових пристроїв (наприклад, розумних колонок). Подібно до чат-ботів, які взаємодіють у письмовій формі, ГА залучають користувача до розмов, налаштованих для допомоги в різних завданнях, таких як відповіді на запитання та керування розумним будинком. І ГА, і чат-боти можна назвати розмовними агентами (РА).

4.1 Модель дослідження

Обсяг інформації, яку користувачі повинні прослухати, залежить від завдань, які вони виконують. Оскільки клієнти слухають менше інформації під час завдань, які вони сприймають як більш цілеспрямовані, переваги мовленнєвої взаємодії мають бути більш помітними, якщо рівень цілеспрямованості високий. Однак, якщо клієнти хочуть порівняти й оцінити різні альтернативи в експериментальних завданнях, їм потрібно слухати більше інформації послідовно під час мовленнєвих взаємодій, що знижує їхню ефективність порівняно з текстовими взаємодіями.

Тому зробимо такі припущення:

Гіпотеза Г1: Клієнти досягають вищого рівня ефективності за допомогою мовної взаємодії, ніж за допомогою текстової взаємодії.

Гіпотеза Г2: Цілеспрямованість позитивно пом'якшує вплив модальності взаємодії на ефективність.

Створення тексту передбачає процес свідомого застосування навмисно засвоєних правил, тоді як люди інтуїтивно будують і

обробляють мову.

Таким чином, мовлення вимагає більше когнітивних зусиль у емпіричних завданнях:

Гіпотеза Г3: Клієнти сприймають менше когнітивних зусиль, коли вони взаємодіють за допомогою мови, ніж за допомогою тексту.

Гіпотеза Г4: Цілеспрямованість позитивно пом'якшує вплив модальності взаємодії на когнітивні зусилля.

Для завдань, які сприймаються як менш цілеспрямовані, взаємодія служить для вивчення різних варіантів. Таким чином, переваги розмови в двосторонній розмові повинні ставати більш вираженими в міру збільшення тривалості взаємодії.

Гіпотеза Г5: Клієнти отримують більше задоволення, коли спілкуються за допомогою мови, ніж за допомогою тексту.

Гіпотеза Г6: Цілеспрямованість негативно зменшує вплив модальності взаємодії на задоволення.

Високі когнітивні зусилля негативно впливають на задоволеність клієнтів результатом пошуку. Крім того, задоволеність клієнтів залежить не лише від зовнішньої мотивації, а й від внутрішніх очікувань.

Таким чином, можна очікувати, що задоволення також сприяє задоволенню клієнтів від зустрічі з послугами:

Гіпотеза Г7: Сприйняття ефективності позитивно впливає на задоволеність клієнтів зустріччю з послугами.

Гіпотеза Г8: Когнітивні зусилля негативно впливають на задоволеність клієнтів від зустрічі з послугами.

Гіпотеза Г9: Задоволення позитивно впливає на задоволеність клієнтів від зустрічі з послугами.

На рисунку 4.1 наведена модель взаємозв'язку гіпотези про відповідність між взаємодією та цілеспрямованістю завдання щодо їхнього впливу на задоволеність клієнтів. Мовлення відрізняється від текстової взаємодії своєю сприйманою ефективністю, когнітивними зусиллями та насолодою, що, зрештою, впливає на задоволеність

клієнтів зустрічю з послугами. Однак це співвідношення може змінюватися залежно від передбачуваної цілеспрямованості пошукового завдання.

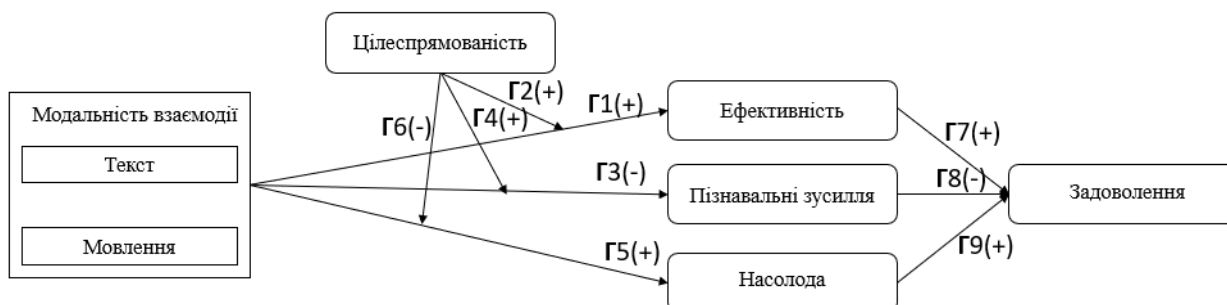


Рисунок 4.1 – Модель взаємозв’язку гіпотези про відповідність між взаємодією та цілеспрямованістю завдання щодо їхнього впливу на задоволеність клієнтів

Проведений експеримент між суб’єктами 2 (модальність взаємодії: мовлення проти тексту) \times 2 (тип пошукового завдання: цілеспрямоване чи експериментальне). Спочатку проведений експеримент з модальністю взаємодії як міжпредметним фактором і типом завдання як внутрішньо предметним фактором. У цьому експерименті випадковим чином розподілили учасників у групу, яка взаємодіяла б або через мову, або через текст зі спеціально створеним РА, щоб вирішити – для експериментальних цілей – єдине цілеспрямоване та одне експериментальне пошукове завдання у випадковому порядку. Був використовуваний алгоритм рандомізації, інтегрований у програмне забезпечення для опитування Qualtrics, яке також використовувалося для інструкцій та анкет експерименту.

Усі взаємодії відбувалися з однією людиною, щоб забезпечити контрольовані та приватні умови для кожного учасника. Учасники обирали зустріч через онлайн-форму реєстрації. Після прибуття дослідник надав учасникам інформацію про загальну процедуру дослідження та технічні пристрої (ноутбук і РА), які використовувалися в експерименті. Учасники

мали змогу коротко ознайомитись із налаштуванням та поставити відкриті запитання. Після цього дослідник залишив кімнату, але залишався доступним у будь-який час, якщо виникли проблеми. На цьому етапі учасники почали експеримент на ноутбучі та ознайомилися з процесом експерименту, зображеним на рисунку 4.2 .



Рисунок 4.2 – Експериментальна процедура

Давши загальні інструкції щодо експерименту, коротко визначено РА та попросено учасників відповісти на запитання про їхній попередній досвід роботи з СА. Потім ми випадковим чином розподілили учасників за умовами тексту або мовлення та надали їм конкретні інструкції щодо взаємодії з СА.

Учасники провели зразкове завдання для ознайомлення з системою. Коли учасники відчували себе впевнено, користуючись системою, вони переходили до фактичного завдання, яке, випадково, було або цілеспрямованим, або експериментальним. Після виконання завдання ми провели опитувальник після виконання завдання з перевіркою маніпуляцій і уваги, а також поставили запитання щодо суб'єктивних показників (передбачувана ефективність, когнітивні зусилля, насолода та задоволення від зустрічі з послугами). Експеримент завершився демографічними запитаннями.

4.1 Методика проведення експериментів

Щоб маніпулювати передбачуваною цілеспрямованістю завдання, ми адаптували завдання пошуку ресторану, яке вже успішно використовувалося для представлення цілеспрямованих і емпіричних пошукових завдань і для порівняння мовної та текстової взаємодії. Завдання цілеспрямованого пошуку пропонувало учасникам знайти ресторан, який, щоб вони могли поужити з другом, мав відповідати певним критеріям (кухня, розташування, ціна). У дослідницькому дослідницькому завданні ми попросили учасників вільно досліджувати ресторани відповідно до власних уподобань, не вказуючи конкретних цілей чи приписувальних критеріїв. Щоб запобігти будь-яким залежним від сценарію ефектам, ми дещо змінили сценарій, призначивши пошук у ресторані чи кафе. Учасники можуть припинити взаємодію в будь-який момент. У таблиці 4.1 вказані умови.

Розроблено РА під назвою Restaurant Finder на базі хмарної платформи Google DialogFlow, щоб уможливити взаємодію як на основі мови, так і на основі тексту, використовуючи однакову структуру діалогу та викликаючи ідентичні відповіді [31, 32]. DialogFlow — це розмовна платформа, яка використовує алгоритми машинного навчання для обробки природної мови. Реалізоване структура діалогового дерева. РА надав рекомендації щодо

ресторану чи кафе на основі вподобань учасників щодо кухні, місця розташування та ціни. Залежно від умов, поряд з ноутбуком ставили або розумну колонку, або смартфон. У режимі мовлення учасники взаємодіяли через розумний динамік з Google Nest Mini, кажучи: «Ok Google, запусти пошук ресторану».

Таблиця 4.1 – Умови експерименту

		Модальність взаємодії	
		Мовлення (Google Nest Mini)	Текст (Google Pixel)
Тип пошукового завдання	Цілеспрямований	Шукайте найдешевший японський ресторан біля університету	Шукайте найдешевший японський ресторан біля університету
	Досвід	Шукайте будь-який ресторан без попередньо визначених критеріїв	Шукайте будь-який ресторан без попередньо визначених критеріїв

Реалізований агент DialogFlow за допомогою програми Google Assistant на розумній колонці. У текстовому режимі учасники починали взаємодію через вікно чату месенджера на смартфоні Google Pixel, ввівши «ok google, запусти пошук ресторану». У цьому випадку реалізований агент DialogFlow як демонстрація у вікні браузера на смартфоні. Обидві реалізації базувалися на тому самому агенті DialogFlow і, отже, залучав користувача до того самого потоку розмови. В таблиці 4.2 подано зразок діалогу до кожного завдання.

4.2 Результати експериментів

В опитувальнику після виконання завдання були виміряні всі приховані змінні за допомогою шкал, адаптованих із діючої літератури, щоб

забезпечити достовірність вмісту, одночасно модифікуючи елементи відповідно до цільового контексту. Була використана 7-бальна шкала в діапазоні від «зовсім не згоден» (1) до «повністю згоден» (7) для вимірювання когнітивних зусиль, ефективності, і задоволення. Задоволеність оцінювалася за 7-бальною. Конструкція цілеспрямованості завдання була адаптована за 7-бальною шкалою.

Таблиця 4.2 – Зразок діалогу кожного типу пошукового завдання

Цілеспрямований діалог	Досвідний діалог
<u>РА</u> : Ласкаво просимо до служби пошуку ресторанів! Я радий порекомендувати вам ресторан. Що б ви хотіли з'їсти?	<u>РА</u> : Ласкаво просимо до служби пошуку ресторанів! Я радий порекомендувати вам ресторан. Що б ви хотіли з'їсти?
<u>Користувач</u> : Я шукаю японський ресторан біля університету.	<u>Користувач</u> : Я ще не знаю. Що б ви порадили?
<u>РА</u> : Добре, я шукаю японську кухню біля університету. Скільки грошей ви хотіли б витратити?	<u>РА</u> : Я радий надати вам рекомендацію! Виберіть кухню, наприклад, українську європейську, японську або грузинську.
<u>Користувач</u> : Це має бути найдешевший ресторан.	<u>Користувач</u> : Хм, а як щодо ресторану сімейних традицій?
<u>РА</u> : Гаразд, я знайшов ресторан, який відповідає вашим критеріям. Найдешевший японський ресторан – «Япошка». Тепер ви можете запитати додаткову інформацію або закрити програму.	<u>РА</u> : Добре, я шукаю ресторан сімейних традицій. Де б ви хотіли поїсти?
	<u>Користувач</u> : В центрі міста.
	<u>РА</u> : Добре, я шукатиму в центрі міста. Скільки б ви хотіли витратити?
	<u>Користувач</u> : Менше 800 грн.
	<u>РА</u> : Гаразд, я знайшов три ресторани, які відповідають вашим критеріям. Davidoff пропонує невимушену атмосферу, майже домашній затишок і спокійну обстановку. Тепер ви можете запитати додаткову інформацію або закрити програму.

Цілеспрямований діалог	Досвідний діалог
	<u>Користувач</u> : Яка ще рекомендація?
	<u>РА</u> : Друга рекомендація — Familia, де можна познайомитися з старовинними гастрономічними традиціями старого Харкова. Ви можете запитати додаткову інформацію або закрити програму.
	<u>Користувач</u> : Які відгуки? ...

Крім того, зібрана інформація про індивідуальне походження учасників (вік, стать, професія, сфера навчання та дохід), риси особистості, навички роботи з Інтернетом та екстраверсія. В якості контрольного питання учасникам пропонувалося вказати, чи вдалося їм успішно вирішити завдання. Ми використали це запитання для оцінки ефективності. Запитання про реалістичність сценарію оцінювало, наскільки добре учасники змогли ознайомитися з описаною ситуацією та завданням. Коментарі учасників не виявили серйозних проблем чи непорозумінь.

Для збору даних проведено два попередніх тести, щоб навчити та оцінити РА за допомогою об'єктивних і суб'єктивних показників. Під час першого попереднього тесту 16 учасників спілкувалися і записували все, що їм спало на думку. На основі цього відгуку було переглянуто РА та відкориговано структуру розмови для другого об'єктивного оцінювання. Щоб зменшити системні помилки в основному лабораторному експерименті, мета другого попереднього тесту полягала в тому, щоб навчити варіаціям запитів користувачів. Тому було розповсюджено посилання на демонстрацію DialogFlow дослідникам і студентам нашого університету, щоб зібрати з них реальні дані розмов. На основі цих даних бесід вручну призначили невідповідні запити користувачів із 64 бесід. Оскільки додаток спрямований на надання конкретної послуги, додатково оцінено, чи змогли учасники

вирішити завдання. Це дало задовільні результати, оскільки 96,6% учасників змогли знайти ресторан.

Щоб залучити студентів, ми оголосили про лабораторний експеримент через списки студентської розсилки. По черзі експеримент завершили 119 учасників. Ми виключили трьох учасників із подальшого аналізу, оскільки перевірка їхньої уваги та журнали взаємодії показали, що вони не дотримувались інструкцій. З решти 116 учасників 60,3% були дівчата, а середній вік становив 19 роки ($M=18,97$; $SD=4,10$). Більшість учасників були студентами у сфері ІТ (63,8%), потім йшли медицина (6,9%) та економічні науки (6,9%). 80,9% учасників, які розкрили свій дохід, заробляли менше 3000 грн на місяць (5,2% вирішили не розголошувати). Майже всі учасники раніше використовували мовні або текстові РА. Понад 80% учасників використовували текстові РА принаймні щотижня; лише 4,3% вказали, що ніколи не спілкувалися з чат-ботом. 74% наших учасників щотижня використовували мовні РА, тоді як 7,8% ніколи ними не користувалися.

Для аналізів використовувалась ефективна платформа для аналітики даних і їх візуалізації Tableau, яка дозволяє передавати дані всім користувачам.

Результати експериментів не показали суттєвої різниці між статтю учасників, сферою навчання, доходом і попереднім досвідом роботи. Таким чином, учасники були однорідно розподілені між групами. Крім того, проведений односторонній дисперсійний аналіз віку, особистої інноваційності, веб-навичок та екстраверсії, не виявивши суттєвих відмінностей ($p > .1$). Таким чином, можна припустити, що групові відмінності в демографічних або контрольних змінних не вплинули на результати експериментів. Проведений аналіз не виявив суттєвих відмінностей між пошуком кафе та ресторану щодо посередницьких і залежних змінних ($p > .1$). Крім того, у всіх групах учасники вказали, що вони сприймають експеримент як реалістичний ($M = 6,17$; $SD = ,97$). Далі проведений односторонній дисперсійний аналіз для перевірки маніпуляцій,

який виявив значні групові відмінності щодо сприйманої цілеспрямованості завдання ($F = 17,79, p < .001$).

Було виявлено, що загальний вплив мови на задоволеність учасників обслуговуванням є значним ($\beta = .189, p < .01$). Загалом, сприймана ефективність, сприйняті когнітивні зусилля та сприйняте задоволення пояснюють 56,7% дисперсії в задоволенні учасників зустріччю з послугами.

В таблиці 4.3 наведено узагальнення гіпотез та результатів.

Таблиця 4.3 – Узагальнення гіпотез та результатів

Гіпотези		Результати		
Г1	Модальність мовлення	Ефективність ↑	$\beta = 0,167, p = 0,071$	Підтримується
Г2	Модальність мовлення x Цілеспрямованість завдання	Ефективність ↑	$\beta = 0,063, p = 0,476$	Відхилено
Г3	Модальність мовлення	Пізнавальні зусилля ↓	$\beta = -0,199, p = 0,021$	Підтримується
Г4	Модальність мовлення x Цілеспрямованість завдання	Пізнавальні зусилля ↓	$\beta = -0,183, p = 0,056$	Відхилено (інший напрямок)
Г5	Модальність мовлення	Насолода ↑	$\beta = 0,265, p = 0,002$	Підтримується
Г6	Модальність мовлення x Цілеспрямованість завдання	Насолода ↓	$\beta = .263, p = .007$	Відхилено (інший напрямок)
Г7	Ефективність	Задоволення ↑	$\beta = .271, p = .007$	Підтримується
Г8	Пізнавальні зусилля	Задоволення ↓	$\beta = -,383, p = ,000$	Підтримується
Г9	Насолода	Задоволення ↑	$\beta = 0,254, p = 0,002$	Підтримується

З точки зору сприйнятої ефективності, когнітивних зусиль, насолоди та задоволення методи взаємодії, засновані на мовленні, сприймаються як кращі за взаємодії, які засновані на тексті. Показано, що на ці ефекти впливає цілеспрямованість завдання.

Таким чином, з постійним технічним прогресом все частіше прийдеться вирішувати, хто повинен відповідати на запити клієнтів: людина, розмовний агент чи чат-бот.

ВИСНОВКИ

У результаті виконання роботи було досліджено моделі інтелектуальних систем чат ботів. Виявлені плюси та мінуси чат ботів в сучасному житті. Запропоновано рекомендації щодо використання чат ботів в таких сферах як: медицина такі складні захворювання, як дименція, освіта.

Розроблена модель інтелектуального чат боту, яка завдяки мовленню здатна підвищити ефективність сприймання інформації та зменшити когнітивні зусилля, а також отримати більше задоволення від обслуговування, ніж текстова взаємодія.

Проведений лабораторний експеримент із 116 учасниками, які мали виконати завдання з пошуку інформації за допомогою розмовного агента або чат-бота. Результати показують, що мовлення демонструє вищу сприйману ефективність, менші когнітивні зусилля, більше задоволення та більше задоволення від обслуговування, ніж текстова взаємодія. Також виявлено, що ці ефекти залежать від цілеспрямованості завдання.

ПЕРЕЛІК ПОСИЛАНЬ

1. <https://scholarspace.manoa.hawaii.edu/items/baae564d-6187-437e-bd30-8d64f2a2527b> - Bots for language learning
2. <https://www.sciencedirect.com/science/article/abs/pii/S1386505619307166> - An overview of the features of chatbots in mental health: A scoping review.
3. <https://www.jmir.org/2021/6/e25006/> - Chatbots to Support People With Dementia and Their Caregivers: Systematic Review of Functions and Quality
4. <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1223692&dswid=-9912> - Chatbots in education: A passing trend or a valuable pedagogical tool?
5. <https://www.atlantis-press.com/proceedings/iscfec-18/55912419> - Using Chat-bots for Managing Quest-based Modular Training of Engineering Students
6. <https://www.mdpi.com/1424-8220/19/20/4356> - Real-Time Human-In-The-Loop Simulation with Mobile Agents, Chat Bots, and Crowd Sensing for Smart Cities
7. <https://onepetro.org/JPT/article-abstract/71/02/34/207958> - The Oil and Gas Chat Bots Are Coming
8. <http://ceur-ws.org/Vol-2420/paperDT6.pdf> - Integrated, Ubiquitous and Collaborative. Process Mining with Chat Bots
9. <https://reactjs.org> - React
10. <https://redux.js.org> - Redux
11. <https://redux-toolkit.js.org> - Redux-toolkit
12. Khot, T., Sabharwal, A., Clark, P.: SciTail: a textual entailment dataset from science question answering. In: AAAI (2018)
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
14. Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick,

C., Tafjord, O.: Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge. ArXiv abs/1803.05457 (2018)

15. Welbl, J., Liu, N.F., Gardner, M.: Crowdsourcing multiple choice science questions. ArXiv abs/1707.06209 (2017)

16. Bisk, Y., Zellers, R., Le Bras, R., Gao, J., Choi, Y.: PIQA: Reasoning about Physical Commonsense in Natural Language. ArXiv abs/1911.11641 (2020)

17. Hussain, S., Ameri Sianaki, O., & Ababneh, N. (2019, March). A survey on conversational agents/chatbots classification and design techniques. In Workshops of the International Conference on Advanced Information Networking and Applications (pp. 946-956). Springer, Cham.

18. Axak N.G. Multi agent knowledge management architecture «*Information-Management Systems and Technologies*»: Materials of the IV International Scientific Conference, Odessa. – 2015.- p.237-239.

19. Аксак Н.Г. Мультиагентная архитектура управления знаниями. «*Інформаційні управляючі системи та технології*» (ІУСТ-ОДЕСА-2017). Матеріали VI Міжнародної науково-практичної конференції. - Одеса. 2017. - С. 268-270

20. Аксак Н.Г. Концепція побудови мультиагентних систем розподіленої нейромережевої обробки великих даних . ВІСНИК ХНТУ №3(66), - ТОМ 1, - 2018 р. – С. 205-212.

21. Adam, M., Wessel, M., & Benlian, A. (2021). AI-Based Chatbots in Customer Service and Their Effects on User Compliance. *Electronic Markets*, 31, 427–445.

22. Balakrishnan, J., Dwivedi, Y. K., Hughes, L., & Boy, F. (2021). Enablers and Inhibitors of AI-Powered Voice Assistants: A Dual-Factor Approach by Integrating the Status Quo Bias and Technology Acceptance Model. *Information Systems Frontiers*.

23. Chen, J., Le, H., & Tran, S. (2021). Understanding Automated Conversational Agent as a Decision Aid: Matching Agent's Conversation with Customer's Shopping Task. *Internet Research, ahead-of-print*.

24. Behera, R. K., Bala, P. K., & Ray, A. (2021). Cognitive Chatbot for Personalised Contextual Customer Service: Behind the Scene and Beyond the Hype. *Information Systems Frontiers*.

25. Benlian, A., Klumpe, J., & Hinz, O. (2020). Mitigating the Intrusive Effects of Smart Home Assistants by Using Anthropomorphic Design Features: A Multi-Method Investigation. *Information Systems Journal*, 30(6), 1010–1042.

26. Cheng, X., Bao, Y., Zarifis, A., Gong, W., & Mou, J. (2021). Exploring Consumers' Response to Text-Based Chatbots in E-Commerce: The Moderating Role of Task Complexity and Chatbot Disclosure. *Internet Research, ahead-of-print*.

27. Moussawi, S. (2018). User Experiences with Personal Intelligent Agents: A Sensory, Physical, Functional and Cognitive Affordances View. Proceedings of the ACM SIGMIS Conference on Computers and People Research, New York, USA.

28. Mozafari, N., Weiger, W., & Hammerschmidt, M. (2021a). Resolving the Chatbot Disclosure Dilemma: Leveraging Selective Self-Presentation to Mitigate the Negative Effect of Chatbot Disclosure. Proceedings of the 54th Hawaii International Conference on System Sciences, Hawaii, USA.

29. Mozafari, N., Weiger, W., & Hammerschmidt, M. (2021b). Trust Me, I'm a Bot - Repercussions of Chatbot Disclosure in Different Service Frontline Settings. *Journal of Service Management*, 33, 1–33.

30. Dai, H., & Salam, A. F. (2020). An Empirical Assessment of Service Quality, Service Consumption Experience and Relational Exchange in Electronic Mediated Environment (EME). *Information Systems Frontiers*, 22(4), 843–862.

31. Axak N., Korablyov M., Ushakov M. The Development of a Multi-Agent System for Controlling an Autonomous Robot // Proceedings of the 17th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I,

(ICTERI-2021). – Kherson, Ukraine, 2021. – Vol-3013, pp. 96-105. <http://ceur-ws.org>. ISSN 1613-0073

32. Axak N., Korablyov M., Rosinskiy D. MapReduce Hadoop Models for Distributed Neural Network Processing of Big Data Using Cloud Services / Advances in Intelligent Systems and Computing IV. / Editors: Shakhovska, Natalya; Medykovskyy, Mykola O., Springer, 2019. – pp. 387 – 400. ISSN: 2194-5357 //doi.org/10.1007/978-3-030-33695-0