

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ**

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: Розробка інтерактивного цифрового помічника з елементами
штучного інтелекту на базі Dialogflow

Виконала студентка 2 курсу групи МІС-21
спеціальності 122 Комп'ютерні науки
Молчанова Анастасія Юріївна

Керівник д.т.н., професор
Казакова Надія Феліксівна

Рецензент регіональний координатор
програми EGAP Копиченко І.Ю.

Одеса 2022

АНОТАЦІЯ

на магістерську кваліфікаційну роботу

«Розробка інтерактивного цифрового помічника з елементами штучного інтелекту на базі Dialogflow»

студентки Молчанової Анастасії Юріївни

Актуальність теми. Стрімкий розвиток інформаційних технологій зумовлює зміни в підходах організацій, в тому числі вищих навчальних закладів, до комунікації з користувачами їхніх послуг. Інтерактивний помічник може служити цілодобовою довідковою системою. Його вартість набагато менша, ніж утримання додаткових співробітників, він доступний миттєво та цілодобово, їм просто користуватись, він відповідає сучасному темпу життя та поліпшує комунікацію з зацікавленими користувачами.

Метою кваліфікаційної роботи є розробка інтерактивного цифрового помічника факультету комп'ютерних наук, управління та адміністрування Одеського державного екологічного університету з елементами штучного інтелекту для обробки запитів природною мовою. Даний помічник буде корисним для абітурієнтів, студентів та викладачів факультету. Для досягнення даної мети були вирішені наступні задачі: виконаний аналіз предметної області використання цифрових помічників, порівняльний аналіз існуючих аналогів, вибір програмних засобів розробки, проектування системи, реалізація цифрового помічника та тестування його роботи.

Об'єкт дослідження: інтерактивний помічник для інформаційної підтримки учасників освітнього процесу. Предмет дослідження: методи та засоби розробки цифрового помічника з використанням технологій обробки природної мови. Методи дослідження: аналіз та синтез для визначення переваг та недоліків існуючих аналогів, UML-проекування, методи побудови діалогових систем, які використані для розробки інформаційного агента, інтегрованого з популярним месенджером, експериментальне дослідження для перевірки достовірності отриманих результатів.

В роботі відображена значущість інтерактивних помічників в інформаційному просторі, проведений аналіз засобів для створення агентів з використанням технології обробки природної мови. Результатом проведеної роботи є інтерактивний цифровий «Помічник ФКНУА ОДЕКУ». Практичне значення роботи полягає в швидкому та зручному наданні користувачам актуальної інформації щодо факультету КНУА за допомогою боту Telegram. Новизна полягає у застосуванні нового підходу до комунікації з учасниками освітнього процесу за допомогою технології обробки природної мови, що дає можливість пришвидшити процес інформаційної підтримки та зробити її ближчою до користувачів у більш зручному, звичному та природному поданні.

Розробленого помічника можна використовувати під час вступної кампанії для популяризації факультету серед абітурієнтів, інформаційна підтримка допоможе зменшити кількість питань, які виникають у студентів, та вирішувати їх швидше.

Кваліфікаційна робота магістра складається з вступу, 5 розділів, висновків, переліку посилань на 20 найменувань, 1 додаток. Повний обсяг проекту становить 82 сторінки, містить 37 рисунків і 2 таблиці.

Ключові слова: інтерактивний помічник, інформаційний агент факультету, ФКНУА ОДЕКУ, обробка природної мови, Dialogflow, Telegram.

SUMMARY

for a master's thesis

«Development of an interactive digital assistant with AI based on Dialogflow»

by student Molchanova Anastasiia

The relevance of the topic. The rapid development of information technologies causes changes in the approaches of organizations, including higher education institutions, to communication with users of their services. The interactive assistant can serve as a 24/7 help system. Its cost is much lower than maintaining additional employees, it's available instantly and around the clock, it's easy to use, it corresponds to the modern pace of life and improves communication with interested users.

The purpose of the qualification work is to develop an interactive digital assistant of the Faculty of Computer Sciences, Management and Administration of the Odessa State Environmental University with elements of artificial intelligence for processing requests in natural language. This assistant will be useful for applicants, students and teachers of the faculty. To achieve this goal, the following tasks were solved: analysis of the subject area, comparative analysis of existing analogues, selection of development software, system design, implementation of the assistant and testing.

Research object: an interactive assistant for informational support of participants in the educational process. Research subject: methods and development tools for developing a digital assistant using natural language processing technologies. Research methods: analysis and synthesis to determine the advantages and disadvantages of existing analogues, UML design, methods of building dialog systems that are used to develop an information agent integrated with a popular messenger, experimental research to verify the reliability of the obtained results.

The work reflects the importance of interactive assistants in the information space, also the analysis of development tools for creating agents using natural language processing technology was carried out. The result of the work is the

interactive digital «Assistant of FCSMA OSENU». The practical significance of the work consists in quickly and conveniently providing users with up-to-date information about the CSMA faculty using the Telegram bot. The novelty consists in applying a new approach to communication with the participants of the educational process using natural language processing technology, which makes it possible to speed up the process of information support and make it closer to users in a more convenient, familiar and natural format.

The developed assistant can be used during the admissions campaign to popularize the faculty among applicants, informational support will help reduce the number of questions students have and solve them faster.

The master's thesis consists of an introduction, 5 chapters, conclusions, a list of references for 20 titles, 1 appendix. The full volume of the project is 82 pages, it contains 37 figures and 2 tables.

Keywords: interactive assistant, faculty information agent, FKNUA ODEKU, natural language processing, NLP, Dialogflow, Telegram.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	9
ВСТУП	11
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ АНАЛОГІВ	13
1.1 Аналіз предметної області	13
1.2 Аналіз існуючих програмних систем.....	14
2 ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ	19
2.1 Деталізація задач проекту	19
2.2 Обґрунтування вибору типу агента	20
2.3 Вибір платформи для інтеграції з помічником.....	24
3. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ ОБРОБКИ ПРИРОДНОЇ МОВИ.....	29
3.1 Поняття обробки природної мови	29
3.2 Глибоке навчання у NLP	32
3.3 Застосування технології обробки природної мови.....	34
3.4 Інструменти для розробки помічника з використанням технології NLP.....	40
4. ПРОЕКТУВАННЯ ІНТЕРАКТИВНОГО ПОМІЧНИКА.....	46
4.1 Проектування варіантів використання системи	46
4.2 Створення діаграми діяльності.....	47
4.3 Проектування логіки сценаріїв використання	49
5. ПРОГРАМНА РЕАЛІЗАЦІЯ «ПОМІЧНИКА ФКНУА ОДЕКУ».....	51
5.1 Створення агента в Google Dialogflow	51
5.2 Розгортання агента на платформі Heroku.....	58
5.3 Підключення платформи Firebase	66
5.4 Інтеграція помічника з Telegram	68
5.5 Опис та тестування інтерактивного помічника	70
ВИСНОВКИ.....	78
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	80
ДОДАТОК А Управління проектом.....	83

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД	– база даних
ОДЕКУ	– Одеський державний екологічний університет
ОС	– операційна система
ПЗ	– програмне забезпечення
ФКНУА	– Факультет комп'ютерних наук, управління та адміністрування
AI, ШІ	– Artificial intelligence – штучний інтелект
API	– Application Programming Interface – програмний інтерфейс програми
APM	– Application Performance Monitoring – моніторинг роботи застосунків
ASIC	– Application-Specific Integrated Circuit – інтегральна схема для специфічного застосування
CASE	– Computer-Aided Software Engineering – системи автоматизованого проектування моделей
CPU	– Central Processing Unit – центральний процесор
CRM	– Customer Relationship Management – управління взаємовідносинами з клієнтами
DNM	– Dynamic Memory Network – мережа динамічної пам'яті
GPS	– Global Positioning System – система глобального позиціонування
GPU	– Graphics Processing Unit – графічний процесор
GRU	– Gated recurrent units – керований рекурентний блок
HTTP(S)	– HyperText Transfer Protocol (Secure) – протокол передачі гіпертекста (безпечний)
IDE	– Integrated Development Environment – інтегроване середовище розробки
IM	– Instant Message – Миттєві повідомлення
IoT	– Internet of Things – Інтернет речей
LSTM	– Long Short-Term Memory – довга короткочасна пам'ять

ML	– Machine Learning – машинне навчання
NLP	– Natural Language Processing – обробка природної мови
NPM	– Node Package Manager – менеджер пакетів Node.js
PaaS	– Platform as a service – платформа як послуга
QA	– Question-answering – система «Питання-відповідь»
REST	– Representational State Transfer – передача репрезентативного стану
RNN	– Recurrent Neural Networks – рекурентні нейронні мережі
SDK	– Software Development Kit – набір засобів розробки
TPU	– Tensor Processing Unit – тензорний процесор
UML	– Unified Modeling Language – уніфікована мова моделювання
URL	– Uniform Resource Locator – єдиний вказівник на ресурс
VoIP	– Voice over IP – голосовий зв'язок через протокол IP
WBS	– Work Breakdown Structure – ієрархічна структура робіт

ВСТУП

Швидкий розвиток інтернет-технологій зумовлює значні зміни в способах взаємодії організацій з користувачами. Онлайн-присутність та можливість онлайн-зв'язку є необхідними властивостями сучасної організації будь-якого типу. З популяризацією соціальних мереж та месенджерів, особливо для мобільних пристроїв, з'явилося поняття інтерактивних помічників або агентів. Сучасний розвиток алгоритмів штучного інтелекту дозволяє створювати багатофункціональних агентів, які симулюють та замінюють спілкування користувачів з технічною підтримкою, автоматизують рутинну роботу та спрощують отримання послуг та інформації.

При необхідності отримати інформацію абітурієнти та студенти закладів вищої освіти також в першу чергу звертаються до мережі Інтернет: офіційних сайтів та соціальних мереж закладів. Згідно з даними аналітики сайту Одеського державного екологічного університету сума візитів за місяць в середньому складає 23 тисячі відвідувань [1]. Статистика відвідуваності вебсторінки ОДЕКУ показує наявність зацікавленості у вступі, необхідності пошуку інформації про університет, освітні послуги, розкладу, контактів факультетів та кафедр тощо. У середньому відвідувачі сайту переглядають 6 сторінок для того, щоб знайти потрібну інформацію.

Метою даної кваліфікаційної роботи є розробка інтерактивного цифрового помічника з елементами штучного інтелекту для більш зручного та швидкого пошуку інформації про факультет комп'ютерних наук, управління та адміністрування Одеського державного екологічного університету. Даний помічник буде корисним для студентів та викладачів факультету, які зможуть отримати контактну інформацію, розклад занять, останні новини тощо. До того, даний помічник буде цікавим та пізнавальним для абітурієнтів, адже він може надати інформацію щодо вступу, вартості навчання, спеціальностей та переваг університету і факультету КНУА.

Матеріали за темою даної кваліфікаційної роботи були представлені на XXI Науковій конференції молодих вчених ОДЕКУ у секції інформаційних технологій та опубліковані у збірнику тез матеріалів [2].

Структура кваліфікаційної роботи магістра складається з вступу, 5 розділів, висновків, переліку посилань на 20 найменувань, 1 додаток. Повний обсяг проекту становить 82 сторінки, містить 37 рисунків і 2 таблиці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ АНАЛОГІВ

1.1 Аналіз предметної області

Інтерактивний інтелектуальний помічник - це форма розмовного штучного інтелекту, призначена для спрощення взаємодії людини з комп'ютерами. Інтелектуальні помічники не є новою технологією. Перша комп'ютерна програма обробки природної мови ELIZA була створена Джозефом Вайзенбаумом у 1964 році. З того часу відбулося значне збільшення використання обробки природної мови для витягування ключових даних із взаємодії людей. У 2016 році, коли Facebook відкрив свій Messenger API для розробників, інтерактивні помічники отримали значне поширення. Одним із ключових застосувань обробки природної мови було створення інтерактивних помічників у розмовному чаті та голосових помічників, які використовуються в мобільних і веб-застосунках, щоб діяти як агенти обслуговування клієнтів, які обслуговують віртуальні потреби клієнтів.

У 2019 році дослідницький інститут Cargemini опублікував результати опитування про вплив інтерактивних помічників на користувачів після того, як організації включили їх до своїх сервісів [3]. Ключові результати цього опитування показали, що багато клієнтів були дуже задоволені рівнем взаємодії, який вони отримали від цих помічників у чаті, і що кількість користувачів, які охоче користувалися цими помічниками, швидко зростала.

Бренди та компанії головним чином зосереджуються на інтерактивних помічниках, щоб надавати інформацію та допомагати людям швидше виконувати завдання, оскільки машинне навчання дозволяє інтерактивним помічникам ставати все більш досконалими. Перевагами таких помічників є те, що інформація за запитом доступна цілодобово та без вихідних, а послуги компаній доступні в один клік у природній та розмовній формі.

З огляду на ці причини створення інформаційного інтерактивного цифрового помічника з елементами штучного інтелекту для абітурієнтів, студентів та викладачів факультету комп'ютерних наук, управління та адміністрування ОДЕКУ буде актуальною задачею. Цей помічник дозволить отримати

необхідну інформацію швидко, природною мовою та у знайомому інтерфейсі.

1.2 Аналіз існуючих програмних систем

На даний час створюються різноманітні інтерактивні помічники для окремих закладів вищої освіти або для абітурієнтів з усієї України. Розглянемо деякі з них.

Бот «КРОК - Помічник» створений з метою допомогти студентам та викладачам приватного університету економіки та права «КРОК», м. Київ (рис. 1.1)



Рисунок 1.1 – Помічник «КРОК - Помічник»

Основні його функції – це отримання розкладу студентів та викладачів та своєї фінансової інформації для студентів. Після першого вибору групи для перегляду розкладу можна зберегти налаштування, і при натисканні «Розклад» буде відкриватись розклад збереженої групи. Є детальна інструкція користувача, в якій прописаний весь функціонал помічника [4]. Недоліки – навігація лише кнопками меню, необхідно шукати потрібну кнопку в різних меню, щоб отримати необхідну інформацію, текстові команди/запити не розпізнаються. Крім того, всі результати переходів крізь меню відображаються в одному повідомленні і замінюють один одного, тож неможливо переглянути історію запитів (рис. 1.2).

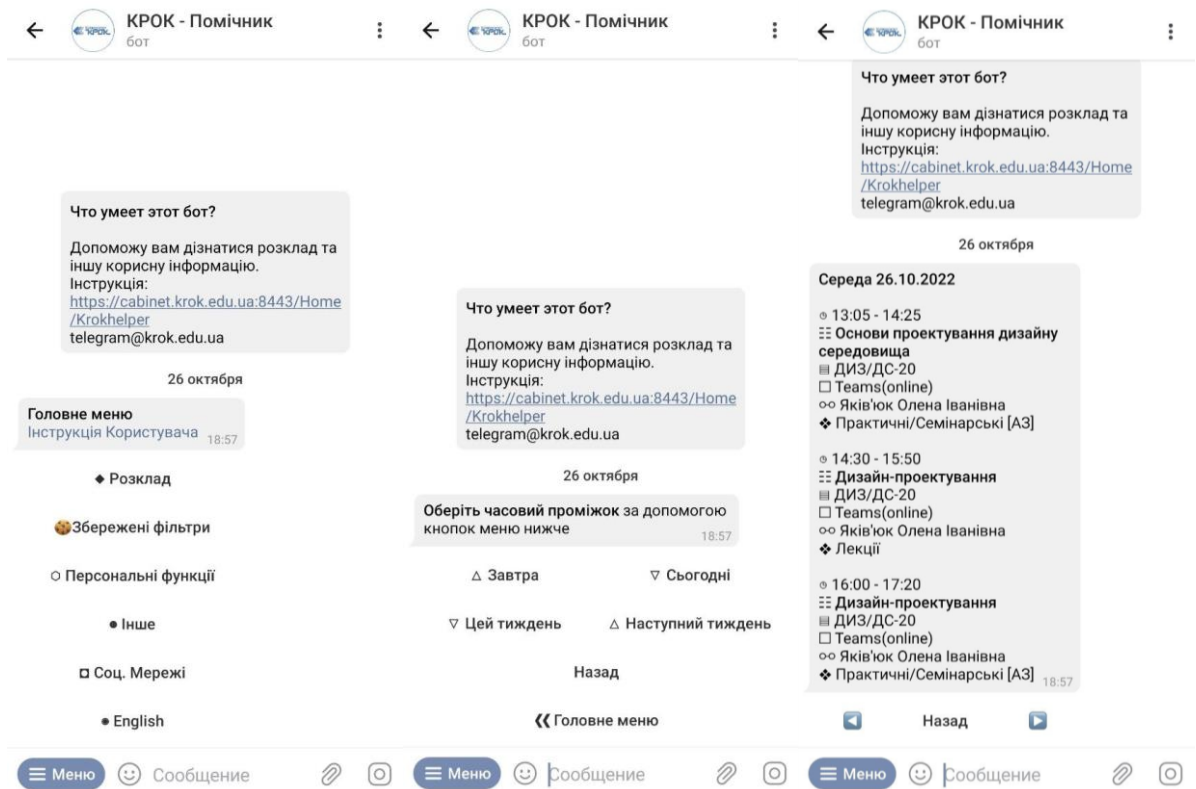


Рисунок 1.2 – Скріншоти помічника «КРОК - Помічник»

Помічник «Розклад VNU» надає доступ до розкладу студентів та викладачів Волинського національного університету імені Лесі Українки (рис. 1.3).



Розклад VNU

@vnu_rozklad_bot

Рисунок 1.3 – Помічник «Розклад VNU»

З першого користування можна обрати свою навчальну групу або привіще викладача для отримання необхідного розкладу. Недоліки – обмеже-

ний функціонал, тільки отримання розкладу, відсутні контакти факультету або кафедр, навігація лише кнопками (рис. 1.4).

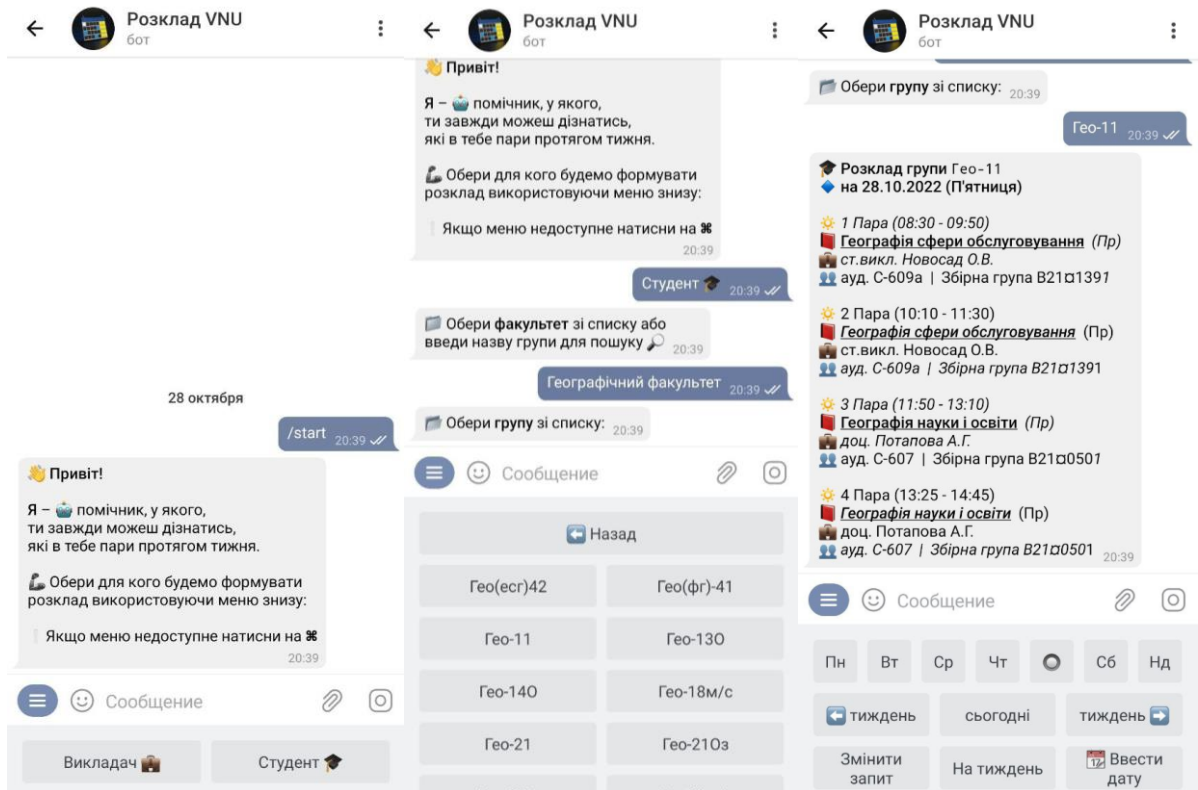


Рисунок 1.4 – Скріншоти помічника «Розклад VNU»

Бот «DekanatFitNewBot» створений для факультету інформаційних технологій КНУ імені Тараса Шевченка (рис. 1.5).



DekanatFitNewBot

@DekanatFitNewBot

В цьому боті Ви можете напряму спілкуватися з працівниками деканату факультету інформаційних технологій

Рисунок 1.5 – Помічник «DekanatFitNewBot»

Помічник дозволяє спілкуватися з працівниками деканату факультету інформаційних технологій, отримати консультацію стосовно навчання від фахівців деканату, надіслати заяви, чек про сплату за навчання, отримати електронну адресу викладача. Головним недоліком є те, що фактично бот не є цифровим помічником, адже на всі запити відповідають фахівці університету, причому лише в будні дні, робочі години.

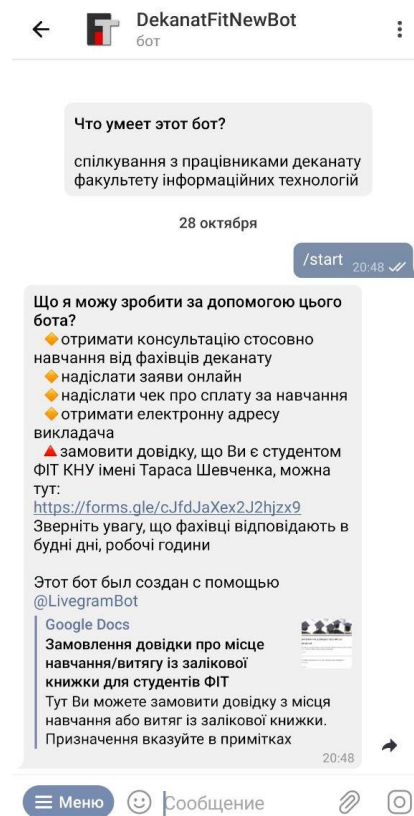


Рисунок 1.6 – Скріншот помічника «DekanatFitNewBot»

Порівняння функцій перерахованих вище помічників різних закладів вищої освіти України наведено у табл. 1. В якості критеріїв порівняння були обрані функціонал, тобто наявність функцій окрім надання розкладу занять, спосіб навігації або спілкування з агентом та використання технології обробки природної мови.

Таблиця 1 – Порівняння інтерактивних помічників університетів України

Критерій порівняння	КРОК-Помічник	Розклад VNU	DekanatFitNew Bot
Функціонал (наявність функцій окрім надання розкладу занять)	Так	Ні	Так
Навігація	Кнопки меню	Кнопки меню	Запити природною мовою
Обробка природної мови	Ні	Ні	Ні

Таким чином, під час виконання першого розділу магістерської роботи була обрана предметна область, в рамках якої розглянуті програмні аналоги і їх особливості. Переваги та недоліки розглянутих помічників-аналогів були взяті до уваги при розробці власного агента.

2 ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ

2.1 Деталізація задач проекту

Метою кваліфікаційної роботи є розробка інтерактивного цифрового помічника (агента) факультету комп'ютерних наук, управління та адміністрування Одеського державного екологічного університету з елементами штучного інтелекту для обробки запитів природною мовою. Даний помічник буде корисним для абітурієнтів, студентів та викладачів факультету.

Для досягнення поставленої мети в роботі були сформульовані наступні завдання:

- провести аналіз предметної області використання цифрових помічників;
- провести порівняльний аналіз існуючих аналогів;
- обґрунтувати вибір програмних засобів розробки та технологій;
- провести проектування системи;
- виконати програмну реалізацію цифрового помічника;
- підготувати інструкцію користувача;
- виконати тестування розгорнутого проекту.

Під час деталізації задач проекту було проведено планування робіт, що представлено у Додатку А.

Після проведення детального аналізу предметної області наступними етапами є проектування та програмна реалізація помічника. Після того, як буде виконана розробка інтерактивного помічника, наступним необхідним етапом є тестування. Недоліки, виявлені на етапі тестування мають бути проаналізовані та виправлені.

Перед початком розробки інтерактивного цифрового помічника були визначені функціональні та нефункціональні вимоги до нього.

До функціональних вимог належать:

- відображення розкладу занять для обраної групи або викладача;
- надання інформації про факультет та кафедри;

- можливість зберігати обрану групу або викладача для швидкого отримання розкладу;
- надання інформації щодо вступу, контактів приймальної комісії, спеціальностей, вартості навчання тощо;
- відображення останніх новин факультету;
- розпізнавання текстових запитів користувачів, обробка природної мови.

Нефункціональними вимогами до агента є надійність та доступність для використання в будь-який момент часу; адекватна реакція в разі збоїв в роботі, збереження всієї інформації в базі даних, така побудова системи, при якій буде легко створювати нові модулі та функції або виправляти код з метою покращення.

2.2 Обґрунтування вибору типу агента

Інтерактивні помічники можуть бути реалізовані як окремий програмний застосунок, web-сторінка або інтегровані у застосунки для обміну повідомленнями [5].

Мобільні або десктопні застосунки призначені для певної платформи, наприклад, iOS, Android або Windows. Користувач додатково завантажує та встановлює програму на свій пристрій з магазину програм. Перевагами застосунків є:

1. Інтерактивність. Застосунок – новий канал взаємодії з поточними користувачами. Програми можуть надавати новий функціонал та інтегрувати недоступні на сайті функції.

2. Підстроювання під користувача. Програми дозволяють користувачам налаштовувати свої уподобання відразу після завантаження та налаштовувати їх відповідно до своїх потреб. Програми також можуть відслідковувати взаємодію з користувачем та використовувати його для надання індивідуальних рекомендацій та оновлень, що робить програму більш корисною для ко-

ристувача. Застосунки також дозволяють компаніям надсилати повідомлення користувачам, які адаптовані до їх інтересів, розташування, поведінки тощо. Таке налаштування дозволяє користувачеві максимально ефективно використовувати програму.

3. Можливість працювати в автономному режимі. Програми можуть працювати без підключення до Інтернету. Хоча багатьом програмам для виконання більшості завдань потрібне підключення до мережі, вони можуть пропонувати користувачам деякий контент та функції в автономному режимі. Завдяки цій перевазі користувачі можуть отримати доступ до певної інформації в будь-який час і в будь-якому місці.

4. Використання можливостей пристрою. Мобільні або десктопні програми можуть отримувати доступ і використовувати вбудовані функції пристрою, такі як камера, GPS, місцезнаходження, спілкування із зовнішніми IoT-пристроями або розумною побутовою технікою. Використання можливостей пристрою призводить до більшої зручної взаємодії з користувачем.

Звичайно, у застосунків є недоліки, серед яких: необхідність створення окремих застосунків для кожної операційної системи, складність розробки та вартість розгортання, зокрема публікації в магазинах, необхідність завантаження додаткових програм на пристрій та вивчення інтерфейсу кожного нового застосунку.

Адаптивні веб-сайти – це веб-сайти, які можуть підлаштовуватись під роботу з екранами різних розмірів. Переваги адаптивних веб-сайтів:

1. Доступні для всіх користувачів. На відміну від мобільних програм, які працюють лише на певних платформах (iOS, Android тощо), до сайту можна отримати доступ з будь-якого пристрою незалежно від операційної системи, якщо доступне підключення до Інтернету. Однак важливо пам'ятати, що доступ до мережі, якість та швидкість – все це фактори, які впливатимуть на роботу вашого сайту. Адаптивні веб-сайти також не потрібно завантажувати або інсталювати.

2. Користувачам не потрібно оновлюватись. Користувачам не потрібно витрачати час на встановлення нових версій та оновлень, щоб покращити роботу сайту. Оскільки веб-сайти легко оновлювати, виправляти помилки та підтримувати, користувачі, швидше за все, не помітять процесу оновлення та зможуть одразу перейти до нових можливостей.

3. Економічно ефективні. Створити сайт набагато дешевше, ніж застосунок.

Серед мінусів адаптивних веб-сайтів – швидкість роботи, перевага мобільних користувачів проводити час в застосунках, а не в браузері.

Інтерактивні помічники, які впроваджені у застосунки для обміну повідомленнями (месенджери, соціальні мережі), наразі дуже популярні. Це обумовлено багатьма причинами [6]:

1. Немає необхідності вивчати новий інтерфейс користувача

Веб-сайти та мобільні застосунки мають різні візуальні інтерфейси, тому користувачам постійно доводиться вивчати нові. Оскільки месенджери використовуються все частіше, спілкування в них з організаціями за допомогою інтерактивних помічників здається чимось природним і простим. Інтерфейс користувача при користуванні інтерактивним помічником в застосунках для обміну повідомленнями є стандартним, тому не потрібно вивчати новий інтерфейс користувача. Оскільки в наш час більшість людей не бажає і не має часу на вивчення нових користувацьких інтерфейсів, текстові повідомлення та голосові команди – основні і зручні способи взаємодії. До того ж зараз використання програм для обміну повідомленнями стає настільки природним, що навіть найменш технічні люди добре справляються із спілкуванням у чатах.

2. По-справжньому персоналізований досвід користувача

Здатність цифрових помічників аналізувати мову, розуміти наміри користувачів, завжди знати про контекст означає, що можна створювати набагато більш значущі та персоналізовані враження. Для інтерактивних поміч-

ників дуже легко сегментувати свою аудиторію та надавати різним користувачам різний досвід.

Щоразу, коли хтось відкриває месенджер і починає спілкуватися в чаті, організація дізнається особу користувача та має доступ до всіх попередніх взаємодій. Таким чином, цифровий помічник стає розумнішим із кожною новою взаємодією та зможе постійно вчитися на поведінці користувача, пропонуючи набагато більш персоналізовані відповіді на основі попередніх повідомлень, скарг і запитів.

3. Повторне залучення користувачів доречним способом

Існує багато інструментів, які дозволяють організаціям та компаніям спілкуватися з користувачами, від електронного маркетингу до соціальних мереж і веб-сайтів, що полегшують охоплення великої аудиторії. На жаль, ці інструменти не дозволяють спілкуватися з людьми на індивідуальному рівні. Принадність цифрових помічників полягає в тому, що можна робити індивідуальне націлювання та охоплювати велику кількість людей на індивідуальному рівні.

Зібрана інформація про користувачів може бути використана для повторної взаємодії з користувачами та побудови міцніших стосунків, які призведуть до бажаної мети, наприклад, замовлення товару. Організації можуть надсилати релевантні персоналізовані повідомлення в потрібний час і робити це через одну з найпопулярніших програм на смартфонах кожного.

4. Кращий і швидший досвід

Розмовна мова – це найприродніший спосіб спілкування між людьми, а програми обміну повідомленнями — це найновіший спосіб використовувати мову як основний спосіб зв'язку між користувачами та організаціями. Однією з великих переваг інтерактивних помічників є те, що навіть базові можливості обробки природної мови (NLP) дають користувачам можливість майже миттєво отримувати інформацію, яку вони шукають, замість того, щоб шукати по всьому веб-сайту.

5. Безпроблемна передача до агента-людини

Інтерактивні помічники самі по собі не вирішують усі проблеми, які можуть виникнути у користувачів, адже веб-сайти та застосунки також цього не роблять. У цифрових помічників це відбувається через обмеження технології NLP, а також через те, що іноді вони (або веб-сайт) просто не можуть надати очікувану відповідь. Однак у інтерактивних помічниках завжди є можливість мати агента-людину, готову безперервно вступити в розмову. На веб-сайті важче зрозуміти, коли у користувача з'являються сумніви, і коли він закриває вкладку, він втрачається назавжди. А з цифровим помічником навпаки – відповідь від агента-людини призведе до сповіщення на телефоні, що знову залучить користувача [7].

Таким чином, за підсумком аналізу існуючих варіантів розробки помічника, була обрана його інтеграція із застосунками для обміну повідомленнями, адже таке рішення не залежить від операційної системи, а також є зручним і зрозумілим для користувачів.

2.3 Вибір платформи для інтеграції з помічником

Понад 3 мільярди людей щоденно використовують програми обміну повідомленнями, що робить їх одними з найпопулярніших типів програм. Найпопулярніші месенджери у світі – WhatsApp, WeChat, Facebook Messenger, Telegram. Для порівняння, по даним на жовтень 2022 року WhatsApp має 2 мільярда активних користувачів щомісяця, Facebook Messenger – 976 мільйонів, Telegram – 700 мільйонів, Viber – 260 мільйонів. В Україні найбільш популярними застосунками обміну повідомленнями є Viber, WhatsApp та Telegram [8].

На рис. 2.1 наведена статистика найбільш використовуваних у світі медіа-платформ згідно даним платформи DataReportal.

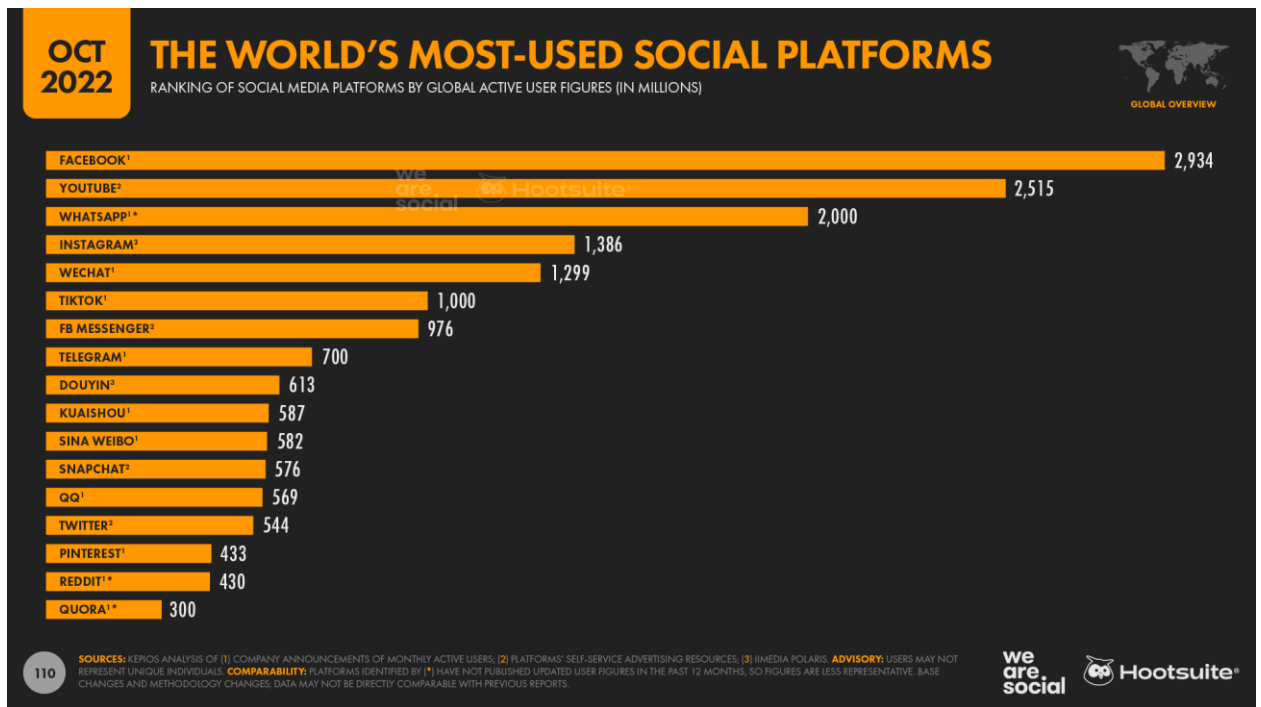


Рисунок 2.1 – Найбільш використовувані у світі соціальні медіа-платформи

Viber – програма-месенджер, яка дозволяє надсилати повідомлення, здійснювати відео- та голосові VoIP-дзвінки через інтернет. Голосові дзвінки між користувачами зі встановленим Viber безкоштовні (оплачується лише інтернет-трафік за тарифом оператора зв'язку). Viber має можливість надсилати текстові, голосові та відеоповідомлення, документи, зображення, відеозаписи та файли, а також працювати в автономному режимі.

Для авторизації користувачів та пошуку контактів програма використовує номер телефону та передає зміст телефонної адресної книги (імена та телефони всіх контактів) на сервери корпорації Viber Media S.à r.l., Люксембург. Вони ж збирають інформацію про дзвінки та передані повідомлення, тривалість дзвінків, учасників дзвінків і чатів – з метою покращення якості обслуговування та в інших цілях.

Viber спочатку був запущений як програма VoIP для голосових викликів. 31 березня 2011 року було випущено Viber 2.0, який додав можливості обміну миттєвими повідомленнями (IM). У липні 2012 року груповий обмін

повідомленнями та механізм HD Voice були додані до програм для Android та iOS. Сьогодні користувачі можуть надсилати кілька типів медіафайлів, включаючи фотографії, відео, GIF-файли, файли, аудіо-повідомлення та наклейки. У грудні 2012 року Viber додав до програми «стікери». У жовтні 2013 року було оголошено про вихід Viber 4.0 із «ринком» стікерів, де Viber продаватиме стікери як джерело доходу. Крім того, у версії 4.0 була представлена функція Viber Out, яка надає користувачам можливість дзвонити на мобільні та стаціонарні номери через VoIP без необхідності використання програми. Голосова підтримка була офіційно додана для всіх пристроїв у 2013.

У листопаді 2016 року Viber версії 6.5 запусив Public Accounts, щоб дозволити брендам брати участь у просуванні та обслуговуванні клієнтів на платформі. Додаток інтегрується з програмним забезпеченням CRM і пропонує API чат-ботів для обслуговування клієнтів. Сьогодні в програмі також доступно багато різних типів чат-ботів, зокрема ті, що надають новини та важливу інформацію, вікторини тощо. Документація Viber REST API доступна на офіційному сайті месенджера.

WhatsApp – американський безкоштовний сервіс обміну миттєвими повідомленнями та голосового зв'язку по IP, що належить компанії Meta. Він дозволяє користувачам надсилати текстові та голосові повідомлення, здійснювати голосові та відеодзвінки, обмінюватися зображеннями, документами, розташуванням користувача та іншим контентом.

Клієнт працює на платформах Android, iOS, S40, KaiOS, а також Windows, macOS та у вигляді веб-програми.

Компанія WhatsApp Inc., яка створила месенджер, була заснована Яном Кумом та Браяном Ектоном 24 лютого 2009 року і розташована в Маунтін-В'ю, США; з жовтня 2014 року належить Meta. З 2016 року додаток офіційно став безкоштовним і до цього дня є таким, користувач оплачує лише використаний додатком інтернет-трафік. Додатком користується понад мільярда осіб. У 2015 року WhatsApp став найпопулярнішим додатком для обміну по-

відомленнями у світі, і станом на квітень 2022 року налічувалося понад 5 мільярдів установок програми через Google Play. WhatsApp став основним засобом електронного зв'язку в багатьох країнах та регіонах, включаючи Латинську Америку, Індійський субконтинент та більшу частину Європи та Африки.

Створити чат-бот для WhatsApp можна лише у спеціальних конструкторах, які надають сторонні компанії. У WhatsApp закритий API, тому легальних способів встановити чат-бот у нього немає – ця принципова політика компанії, яка спрямована на захист користувачів від спаму та нав'язливої реклами.

Telegram – це глобально доступна зашифрована безкоштовна міжплатформна хмарна служба обміну миттєвими повідомленнями (IM). Послуга також надає додаткові наскрізні зашифровані чати та відеодзвінки, VoIP, обмін файлами та інші функції. Він був запущений для iOS 14 серпня 2013 року та Android 20 жовтня 2013 року. Сервери Telegram розподілені по всьому світу з п'ятьма центрами обробки даних у різних частинах світу, а операційний центр знаходиться в Дубаї, Об'єднані Арабські Емірати. Доступні різні клієнтські програми для комп'ютерних і мобільних платформ, включаючи офіційні програми для Android, iOS, Windows, macOS і Linux. Для реєстрації потрібен пристрій iOS або Android і робочий номер телефону. Є також дві офіційні програми-двійники Telegram, WebK і WebZ, і численні неофіційні клієнти, які використовують протокол Telegram. Усі офіційні компоненти Telegram мають відкритий вихідний код, за винятком сервера, який є закритим і пропрієтарним.

Telegram надає додаткові чати з наскрізним шифруванням. Хмарні чати та групи шифруються між програмою та сервером, тому провайдери та інші сторонні особи в мережі не можуть отримати доступ до даних. Користувачі можуть надсилати текстові та голосові повідомлення, здійснювати голосові та відеодзвінки та ділитися необмеженою кількістю зображень, документів (2

ГБ на файл), місцеположення користувачів, анімованих наклейок, контактів та аудіофайлів. Користувачі також можуть стежити за каналами

У січні 2021 року Telegram перевищив 500 мільйонів активних користувачів на місяць. У січні 2021 року це був додаток із найбільшою кількістю завантажень у світі, станом на кінець серпня 2021 року його було завантажено 1 мільярд. У червні 2022 року Telegram перевищив 700 мільйонів активних користувачів на місяць і увійшов в топ-5 найбільш завантажуваних застосунків у світі. Того ж місяця було представлено Telegram Premium, не-обов'язкову платну підписку з додатковими функціями.

Telegram випередив WhatsApp і Facebook Messenger і став найпопулярнішим додатком для обміну миттєвими повідомленнями в Білорусі, Молдові, Йорданії, Вірменії, Азербайджані, Казахстані, Киргизстані, Камбоджі, Ефіопії, Росії та Україні.

Telegram використовує власний протокол шифрування MTProto. MTProto API (він же Telegram API) - це API, через який застосунок Телеграм зв'язується з сервером. Telegram API повністю відкритий, тому будь-який розробник може написати свій клієнт месенджера.

Для написання ботів було створено Telegram Bot API – надбудову над Telegram API. Щоб використовувати Bot API, не потрібно нічого знати про те, як працює протокол MTProto шифрування — допоміжний сервер буде сам обробляти все шифрування і зв'язок з Telegram API. Розробник з'єднується з сервером через простий інтерфейс HTTPS, який надає просту версію Telegram API. Серед спрощень Bot API: робота через вебхуки, спрощена розмітка повідомлень та інше. Telegram має дуже обширну та детальну документацію щодо Telegram API [9].

За підсумком аналізу популярних месенджерів, для інтеграції з інтелектуальним помічником був обраний Telegram, тому що він є найбільш популярним, має багато функцій та відкрите API.

3. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ ОБРОБКИ ПРИРОДНОЇ МОВИ

3.1 Поняття обробки природної мови

Обробка природної мови (англ. Natural Language Processing, NLP) – підрозділ штучного інтелекту, спрямований на вивчення методів аналізу та синтезу природної мови. NLP – це область інформаційних технологій, що знаходиться на перетині машинного навчання та математичної лінгвістики.

Алгоритми обробки природної мови працюють за принципом, що схожий на обробку мови людиною, та складаються з 3 процесів: сприйняття, розуміння сенсу та реагування.

Сприйняття — процес перекладу сенсорного сигналу в символний вид. Для системи NLP це процес перекладу вхідної інформації на зрозумілий для машини набір символів. Сучасні нейронні мережі успішно справляються з завданням сприйняття інформації.

Реакція – це результат прийняття рішення. Це досить просте завдання, що вимагає формування набору можливих відповідей на підставі сенсу сприйнятої фрази та контексту. Завдання реагування успішно вирішується шляхом зважування альтернатив і порівняння результатів один з одним.

Розуміння сенсу - це найскладніше завдання, з яким не завжди справляються навіть люди зі своїм природним інтелектом. Через незнання контексту та неправильну інтерпретацію фрази можуть виникати непорозуміння або навіть серйозні конфлікти. На сьогодні поширені такі види аналізу під час вирішення завдань розуміння мови [10]:

- Статистичний;
- Формально-граматичний;
- Нейромережевий.

Статистичний метод широко застосовується в сервісах машинного перекладу, автоматичних рецензентах та деяких чат-ботах. Суть методу полягає у «годуванні» моделі величезної кількості масивів текстів, у яких встановлено статистичні закономірності. Потім такі моделі використовуються для перекладів текстів або генерування нових, іноді з розумінням контексту.

Формально-граматичний підхід є математичним апаратом, що дозволяє точно і однозначно визначити сенс фрази природною мовою настільки, наскільки це можливо для машини. Однак це не завжди вдається зробити, оскільки сенс деяких фраз незрозумілий навіть людям. Для розвинених мов на зразок української, російської чи англійської точний і детальний опис мови в математичних термінах є дуже складною проблемою. Тому формально-граматичний підхід майже не використовується для аналізу природних розвинених мов, а найчастіше застосовується для синтаксичного аналізу штучних мов.

У нейромережевому підході для розпізнавання сенсу вхідної фрази та генерації реакції системи штучного інтелекту використовуються нейронні мережі глибокого навчання. Вони навчаються на парах стимул-реакція, де стимулом є фраза природною мовою, а реакцією – відповідь системи на неї або будь-які дії системи. Це найбільш перспективний підхід, але він має всі негативні якості нейронних мереж. Одним із головних недоліків нейронних мереж є кількість даних, які їм потрібні для навчання. Через це та кількість циклів навчання часто потрібний доступ до потужного та дорогого комп'ютерного обладнання, оснащеного високопродуктивними графічними процесорами. Іншим викликом для створення високоточних моделей є неточності у наборах даних, які можуть припускати люди при їх створенні.

Більшість проблем використання нейронних мереж вирішують великі мовні моделі, однак і з ними є низка складнощів. Наприклад, до останнього часу такі моделі не були доступними, проте зараз великі компанії все частіше стали викладати моделі у відкритий доступ. Однак багато моделей працюють лише з популярними мовами. Крім технологічних недоліків, NLP також можуть використовувати у зловмисних цілях, наприклад, для підробки голосу.

Для виконання всіх завдань у процесі обробки природної мови використовуються дві основні техніки: синтаксичний та семантичний аналіз.

Синтаксичний аналіз пов'язаний з розстановкою слів із забезпеченням граматичної точності. Цей аналітичний алгоритм розставляє слова до ство-

рення зв'язного речення без помилок. Техніка оцінює відповідність природної мови граматичним правилам для бездоганного розуміння. Алгоритми виділяють групу слів та застосовують граматичні правила для визначення їхнього значення. Синтаксичними техніками є:

- Лематизація – це лінгвістичний процес, який групує модульовані слова, які можна аналізувати за допомогою одного терміна, що характеризується у вигляді леми (словникової форми).
- Морфологічна сегментація – розбиває групу слів на осмислені фрази чи морфеми.
- Сегментація слів – пов'язана з поділом структурованих речень на складові слова.
- Позначення (тегування) частин мови – визначення частин мови у кожному реченні для застосування граматичних правил.
- Синтаксичний розбір (парсинг) – виконання граматичного аналізу кожного речення.
- Розрив речення – відокремлення одного речення від іншого, встановлення таким чином межі набору слів.
- Стеммінг – зв'язок відміненого слова з його кореневою формою.

При семантичному аналізі увага зосереджується на контекстуальному значенні слів. Цей метод – найскладніша частина процесу природного навчання, який досі перебуває у стадії розробки. Методи семантичного аналізу:

- Розпізнавання сутностей за іменами – ідентифікація та категоризацією слів за певними групами, такими як імена людей або місця.
- Визначення значень слів – додавання контекстного значення до слова на основі структури речення.
- Генерація природної мови – використання бази даних для розшифровки логічного значення слова та перетворення зібраної інформації на людську мову.

3.2 Глибоке навчання у NLP

Істотна частина технологій NLP працює завдяки глибокому навчанню (deep learning) — галузі машинного навчання, яка почала набирати популярність лише на початку цього 2010-х рр. з таких причин:

- Нагромаджено великі обсяги тренувальних даних;
- Розроблено обчислювальні потужності: багатоядерні CPU та GPU;
- Створено нові моделі та алгоритми з розширеними можливостями та покращеною продуктивністю, з гнучким навчанням на проміжних уявленнях;
- З'явилися навчальні методи з використанням контексту, нові методи регуляризації та оптимізації.

У глибокому навчанні алгоритм намагається автоматично отримати найкращі ознаки чи подання із сирих вхідних даних. Створені вручну ознаки часто надто спеціалізовані, неповні та потребують часу на створення та валідацію. На противагу цьому, виявлені глибоким навчанням ознаки легко пристосовуються [11].

Глибоке навчання пропонує гнучкий, універсальний та навчальний фреймворк для представлення світу як візуальної, так і лінгвістичної інформації. Спочатку це призвело до проривів у галузях розпізнавання мови та комп'ютерного зору.

У традиційному NLP слова сприймаються як дискретні символи, які далі представляються як вектори. Проблема зі словами – дискретними символами – відсутність визначення схожості для векторів. Тому альтернатива — навчитися кодувати схожість у вектори.

Векторне уявлення - метод уявлення рядків, як векторів зі значеннями. Будується щільний вектор для кожного слова так, щоб слова, що зустрічаються в схожих контекстах, мали схожі вектори. Векторне уявлення вважається стартовою точкою для більшості NLP завдань і робить глибоке навчання ефективним на маленьких наборах даних. Техніки векторних уявлень

Word2vec і GloVe, створених Google та Stanford користуються популярністю і часто використовуються для завдань NLP.

Word2vec приймає великий корпус тексту, у якому кожне слово у фіксованому словнику представлено у вигляді вектора. Далі алгоритм пробігає по кожній позиції t в тексті, яка є центральним словом s і контекстним словом o . Далі використовується схожість векторів слів для s і o , щоб розрахувати ймовірність o при заданому s (або навпаки), і продовжується регулювання вектору слів для максимізації цієї ймовірності (рис. 3.1).

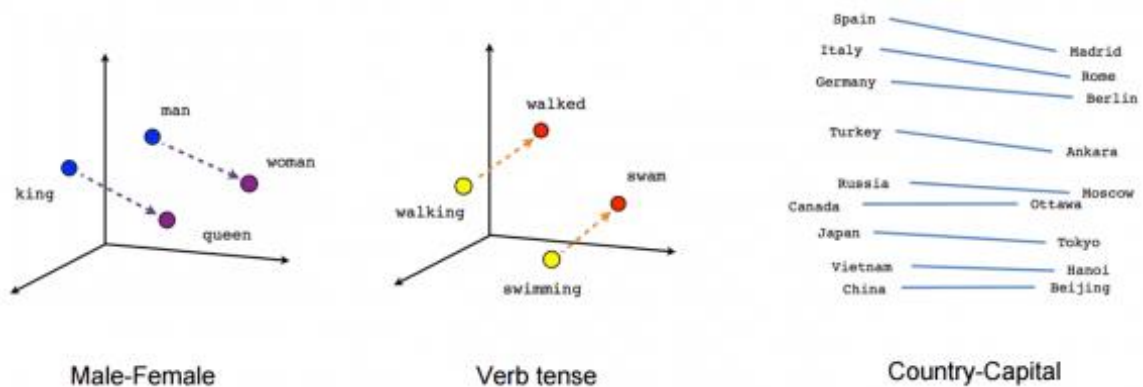


Рисунок 3.1 – Регулювання векторів слів

Для досягнення найкращого результату Word2vec з набору даних видаляються марні слова (або слова з великою частотою появи, такі як сполучники). Це допомагає покращити точність моделі та скоротити час на тренування. Крім того, використовується негативна вибірка для кожного входу, оновлюючи ваги для всіх правильних міток, але тільки на невеликій кількості некоректних міток.

GloVe прагне вирішити цю проблему захопленням значення одного векторного уявлення зі структурою всього доступного для огляду корпусу. Щоб це зробити, модель шукає глобальні збіги числа слів та мінімізує середньоквадратичне відхилення. Така схема дозволяє ототожнювати схожість слова з векторною відстанню.

3.3 Застосування технології обробки природної мови

Сьогодні NLP застосовується для вирішення безлічі завдань в різних галузях:

- письмовий чи усний пошук;
- показ релевантної онлайн реклами;
- автоматичний переклад;
- фільтрація спаму;
- виправлення граматичних та контекстних помилок у тексті;
- аналіз настроїв для завдань маркетингу;
- передбачення наступних слів в контексті;
- розпізнавання мови;
- голосові та текстові помічники.

Машинний переклад - перетворення тексту природною мовою в еквівалентний за змістом текст іншою мовою. Робить це програма чи машина без участі людини. У машинному перекладі використовується статистика використання слів по сусідству. Системи машинного перекладу знаходять широке комерційне застосування, оскільки переклади мов світу — індустрія обсягом \$40 мільярдів на рік.

У традиційних системах машинного перекладу доводиться використовувати паралельний корпус — набір текстів, кожен із яких перекладено однією чи кількома іншими мовами. Наприклад, маючи вихідну мову u (Українська) і цільову e (Англійська), потрібно побудувати статистичну модель, що включає ймовірне формулювання для правила Байєса, модель перекладу $p(u|e)$, навчену на паралельному корпусі, і модель мови $p(e)$, навчену лише на корпусі з англійською мовою.

Звичайно, цей підхід пропускає безліч важливих деталей, що потребує великої кількості спроектованих вручну ознак та складається з різних і незалежних завдань машинного навчання.

Нейромережевий машинний переклад - підхід до моделювання перекладу за допомогою рекурентної нейронної мережі (Recurrent Neural Network, RNN). RNN — нейронна мережа з залежністю від попередніх станів, яка має зв'язки між проходами. Нейрони одержують інформацію з попередніх шарів, а також із самих себе на попередньому кроці. Це означає, що порядок, в якому подаються на вхід дані та тренується мережа, важливий.

Стандартна модель нейромережевого машинного перекладу є наскрізною нейромережею, де вихідне речення кодується RNN-кодувальником, а цільове слово передбачається за допомогою іншої RNN, яка називається декодер. Кодувальник "читає" вихідне речення зі швидкістю один символ в одиницю часу та об'єднує вихідне речення в останньому прихованому шарі. Декодер використовує зворотне поширення помилки вивчення цього об'єднання і повертає переведений варіант. Такий нейромережевий машинний переклад всього через 2 роки з початку досліджень став стандартом машинного перекладу [12].

Головна проблема RNN — проблема зникнення градієнта, коли інформація втрачається з часом. Інтуїтивно здається, що це не є серйозною проблемою, оскільки це лише ваги, а не стан нейронів. Але з часом ваги стають місцями, де зберігається інформація з минулого. Якщо вага набуде значення 0 або 100000, попередній стан не буде надто інформативним. Як наслідок, RNN будуть відчувати складності у запам'ятовуванні слів, що стоять далі в послідовності, а передбачення будуть робитися тільки на основі крайніх слів.

Мережі короткостроково-довгострокової пам'яті (Long/short term memory, далі LSTM) намагаються боротися з проблемою градієнта зникнення, вводячи гейти і комірку пам'яті. Кожен нейрон являє собою клітинку пам'яті з трьома гейтами: на вхід, на вихід та забування. Ці затвори виконують функцію охоронців для інформації, дозволяючи або забороняючи її потік (рис. 3.2).

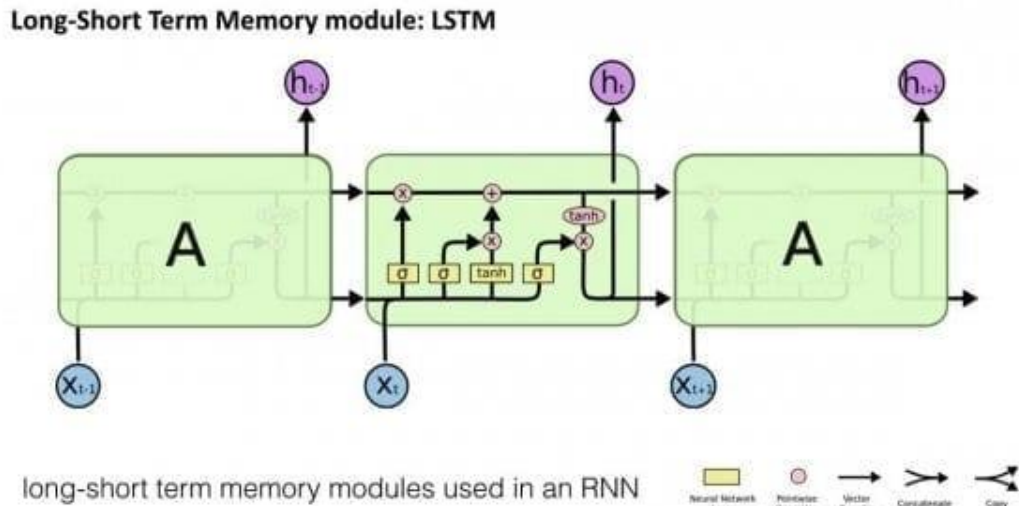


Рисунок 3.2 – Модуль мережі короткостроково-довгострокової пам'яті

Вхідний гейт визначає, яка кількість інформації з попереднього шару зберігатиметься в цьому осередку; вихідний гейт виконує роботу на іншому кінці та визначає, яка частина наступного шару дізнається про стан поточної комірки. Гейт забування контролює міру збереження значення у пам'яті: якщо при вивченні книги починається новий розділ, іноді для нейромережі необхідно забути деякі слова з попереднього розділу.

Було досліджено, що LSTM здатні навчатися на складних послідовностях і, наприклад, писати в стилі якогось письменника або складати примітивну музику. LSTM поширені та використовуються в машинному перекладі. Крім цього, це стандартна модель більшості завдань маркування послідовності, які складаються з великої кількості даних.

Ще одне застосування технології NLP – системи «Питання-відповідь» (Question-answering, QA). Їхня ідея полягає у витягуванні інформації безпосередньо з документа, розмови, онлайн пошуку або будь-якого іншого місця, яке задовольняє потреби користувача. Замість того, щоб змушувати користувача читати повний текст, QA системи вважають за краще давати короткі та лаконічні відповіді. Сьогодні QA системи легко комбінуються з інтерактив-

ними помічниками, виходять за межі пошуку текстових документів та отримують інформацію навіть з набору картинок.

Більшість завдань NLP можуть бути розглянуті як завдання «питання-відповідь». Парадигма проста: надсилається запит, на який машина надає відповідь. Завдяки читанню тексту чи набору інструкцій розумна система має знаходити відповідь на велику кількість питань. Звичайно, потрібно створити модель для відповідей на загальні питання.

Спеціально для завдань QA створено та оптимізовано потужну архітектуру глибокого навчання — Мережа Динамічної Пам'яті (Dynamic Memory Network, DMN). Навчена на тренувальному наборі з вхідних даних та питань, DMN формує епізодичні спогади та використовує їх для генерації відповідних відповідей. Ця архітектура складається з наступних компонентів: модуль семантичної пам'яті, вхідний модуль, модуль питань, модуль епізодичної пам'яті та модуль відповідей (рис. 3.3).

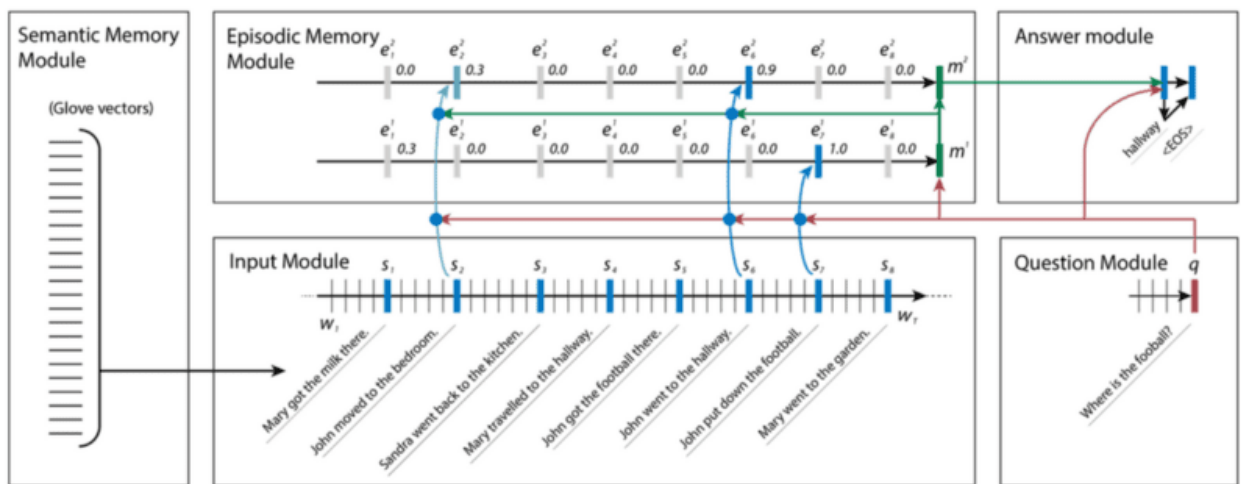


Illustration of DMN performing transitive inference.

Рисунок 3.3 – Архітектура мережі динамічної пам'яті

Модуль семантичної пам'яті, аналогічний базі знань, складається з попередньо підготовлених векторів, які використовуються для створення послідовностей векторних уявлень слів з вхідних речень. Ці вектори використовуються як вхідні дані моделі.

Вхідний модуль переробляє пов'язані з питанням вхідні вектори на набір векторів, званих фактами. Цей модуль реалізований за допомогою керуваного рекурентного блоку (Gated Recurrent Unit, GRU), який дозволяє мережі дізнатися релевантність пропозиції, що розглядається.

Модуль питань опрацьовує питання слово за словом і видає вектор, використовуючи той же GRU, що і у вхідному модулі, з такими ж вагами.

Модуль епізодичної пам'яті зберігає витягнуті на вході вектори фактів та питань, закодовані як вкладення.

Модуль відповідей генерує відповідну відповідь. На останньому етапі епізодична пам'ять містить необхідну для відповіді інформацію. Цей модуль використовує інший GRU, навчений із класифікацією крос-ентропійної помилки правильної послідовності, яка конвертується назад на природну мову.

Людині складно вручну виділити короткий зміст великого обсягу тексту. Тому ще одна проблема, яка вирішується за допомогою технології NLP – створення точного та лаконічного резюме вхідного документа. Вилучення короткого змісту - важливий інструмент для допомоги в інтерпретації текстової інформації. Push-сповіщення та дайджести статей привертають велику увагу, а кількість завдань зі створення розумних та точних резюме для великих фрагментів тексту зростає з кожним днем.

Автоматичне вилучення короткого змісту з тексту працює в такий спосіб. Спочатку обраховується частота появи слова у повному текстовому документі, потім 100 найчастіших слів зберігаються і сортуються. Після цього кожне речення оцінюється за кількістю часто вживаних слів, причому вага більше у слова, що найчастіше зустрічається. Нарешті, перші X пропозицій сортуються з урахуванням положення в оригінальному тексті.

Зі збереженням простоти та узагальнюючої здатності алгоритм автоматичного вилучення короткого змісту здатний працювати у складних ситуаціях. Наприклад, багато реалізацій зазнають невдачі на текстах іноземними мовами або з унікальними словниковими асоціаціями, які не містяться в стандартних масивах текстів.

Виділяють два фундаментальні підходи до скорочення тексту: витягувальний та абстрактний. Перший витягує слова та фрази з оригінального тексту для створення резюме. Останній вивчає внутрішнє мовне уявлення, щоб створити людиноподібний виклад, перефразовуючи оригінальний текст.

Методи у витягувальному скороченні працюють на основі вибору підмножини. Це досягається за рахунок отримання фраз або речень зі статті для формування резюме.

Моделі для абстрактного резюмування використовують глибоке навчання, що дозволило досягти значних проривів. Деякі результати великих компаній в області NLP:

- Facebook Neural Attention — нейромережева архітектура, яка використовує локальну модель з механізмом уваги, здатним генерувати кожне слово резюме в залежності від вхідного речення.
- Google Sequence-to-sequence використовує архітектуру «кодер-декодер». Кодер відповідає за читання вихідного документа та кодування у внутрішнє подання. Декодер відповідає за генерацію кожного слова у зведенні на виході та використовує кодоване подання вихідного документа.
- IBM Watson використовує схожу Sequence-to-sequence модель, але з властивостями уважної та двонаправленої рекурентної нейромережі.

Природна мова — це не просто слова та їх чітке значення. Навпаки, це нюанси та складність. За тим, як людина ставить запитання, можна визначити її почуття: чи вона нудьгує, чи вона зла або чи зацікавлена. На основі вибору слів і пунктуації можна визначити, чи розлютився співбесідник, навіть у повністю текстовій розмові. Ще один напрямок технології NLP – аналіз настроїв. Це процес інтерпретації значення великих текстових одиниць (сутностей, описових термінів, фактів, аргументів, історій) за допомогою семантичного складу менших елементів.

Традиційний підхід до аналізу настроїв полягає в тому, щоб розглядати речення як мішок слів і переглядати підібраний список «позитивних» і «нега-

тивних» слів, щоб визначити настрої цього конкретного речення. Для цього необхідні вручну розроблені функції для фіксації настрою, що займає надзвичайно багато часу та не піддається масштабуванню.

Сучасний підхід глибокого навчання для аналізу настроїв можна використовувати для морфології, синтаксису та логічної семантики, з яких найефективнішою є рекурсивна нейронна мережа. Основним припущенням для розробки рекурсивної нейронної мережі є те, що рекурсія є природним способом опису мови. Рекурсія корисна для усунення неоднозначності, корисна в деяких завданнях для посилення на конкретні фрази та надзвичайно добре працює для завдань, які використовують структуру граматичного дерева. Рекурсивні нейронні мережі ідеально підходять для налаштувань, які мають вкладену ієрархію та внутрішню рекурсивну структуру.

3.4 Інструменти для розробки помічника з використанням технології NLP

Існують спеціальні інструменти для створення інтелектуальних помічників з технологією NLP. Серед таких платформ – Botmother, Nartik, QuickReply.ai, Flow XO, Dialogflow [13].

Nartik – індійська корпоративна розмовна платформа розробки штучного інтелекту, заснована в серпні 2013 року та придбана компанією Reliance Industries Limited у 2019 році. Компанія розробляє технологію, яка дозволяє підприємствам створювати розмовні AI системи, які дозволяють користувачам спілкуватися з додатками та електронними пристроями у вільному, природному форматі, використовуючи голосові або текстові команди. Nartik пропонує сучасні алгоритми, потужний інструмент для проектування, створення та підтримки інтерактивних помічників, інформаційну панель для перегляду звітів про показники та отримання огляду розмов (рис. 3.4). Помічник можна інтегрувати з будь-якою попередньо існуючою або спеціальною системою CRM.

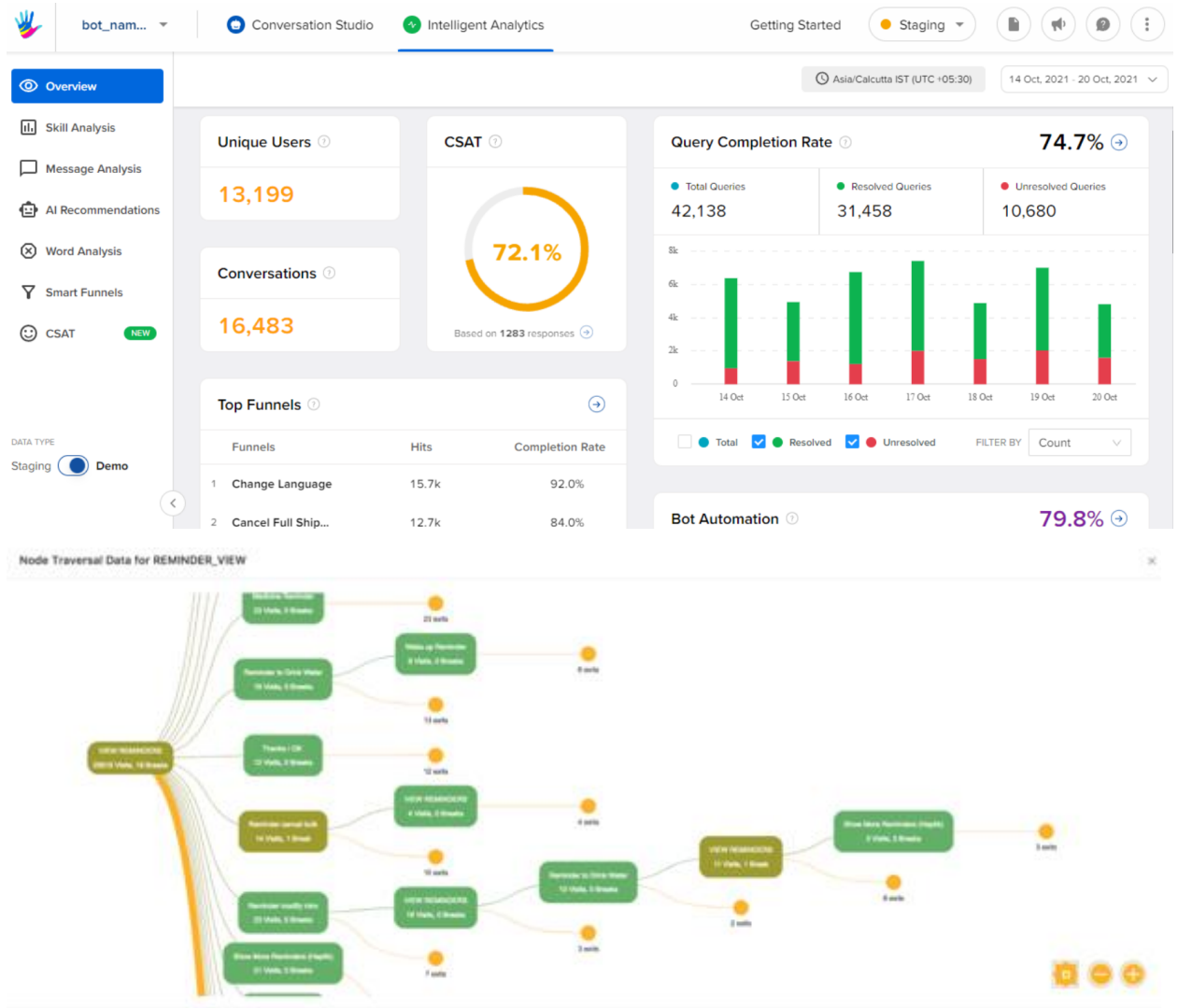


Рисунок 3.4 – Інтерфейс платформи Parik

QuickReply.ai – платформа для створення інтерактивних помічників з ШІ. Однак особливістю QuickReply.ai є те, що вона створена в першу чергу для власників магазинів Shopify. Платформа має попередньо створений і навчений помічник, який глибоко інтегрований з Shopify. Він може вирішувати найпоширеніші запити користувачів, пов'язані зі статусом замовлення, політикою відшкодування, скасуванням, оплатою доставки тощо (рис. 3.5).

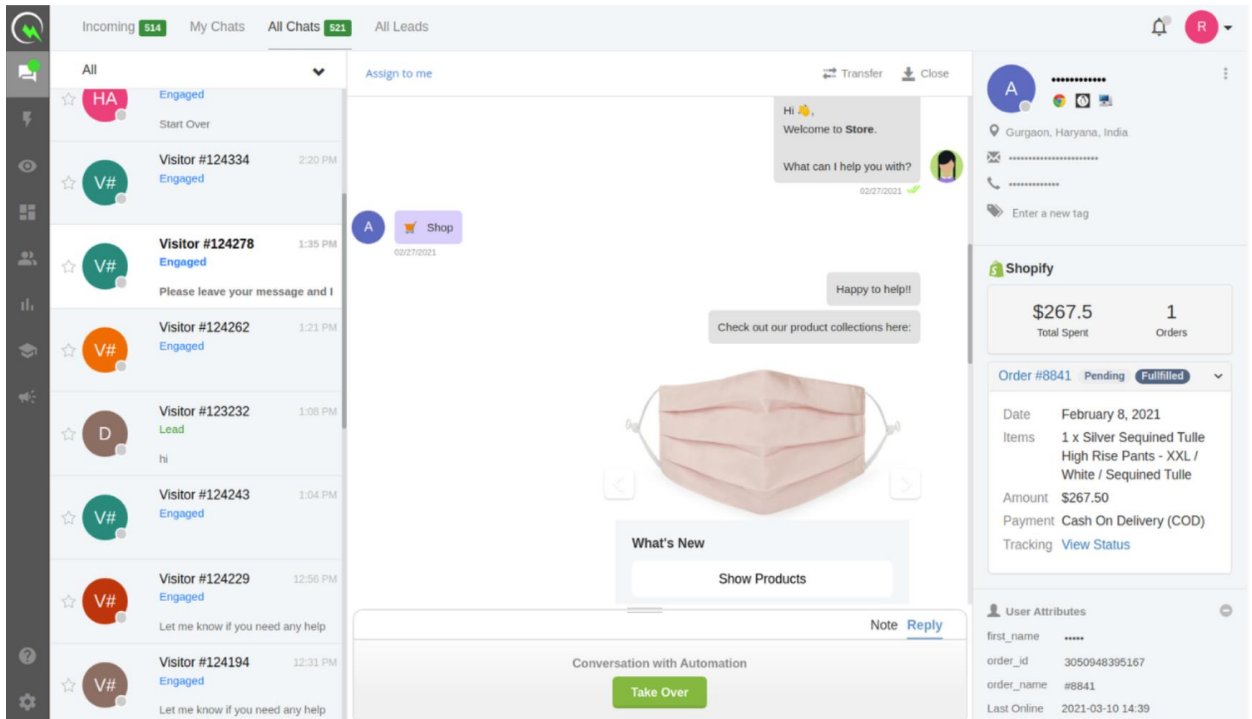


Рисунок 3.5 – Інтерфейс платформи QuickReply.ai

Flow XO — це рішення для створення інтелектуальних помічників без коду, а також для розміщення та розгортання їх на різних каналах/платформах (рис. 3.6). Flow XO має два основних продукти: боти та робочі потоки. Потоки – це алгоритм розмови між вашим помічником і користувачем, а боти – це підключення цих потоків до певних каналів. На жаль, система Flow XO обмежує функціональність і дизайн помічників, оскільки для створення чогось складного вимагає інтеграції сторонніх сервісів. Інші розробники надають багато вбудованих інструментів у своїх платформах, а Flow XO повністю покладається на сторонні інтеграції.

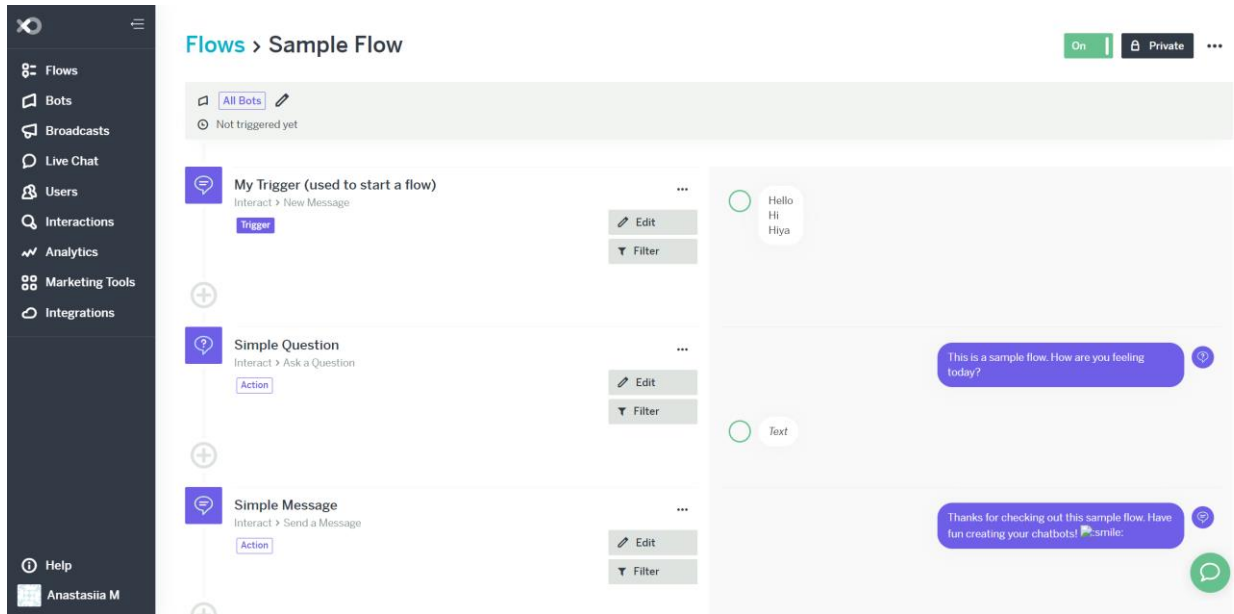


Рисунок 3.6 – Інтерфейс платформи Flow XO

Dialogflow — це платформа для розуміння природної мови, яка використовується для розробки та інтеграції розмовного інтерфейсу користувача в мобільні додатки, веб-додатки, пристрої, боти, інтерактивні системи голосової відповіді тощо (рис. 3.7). Її можна використовувати як для текстових, так і для голосових чат-ботів на основі штучного інтелекту.

У вересні 2014 року Speaktoit випустили `ari.ai` – голосову систему підтримки Асистента для сторонніх розробників, що дозволяла додавати голосові інтерфейси до програм на базі Android, iOS, HTML5 і Cordova. SDK містив розпізнавання голосу, розуміння природної мови та перетворення тексту в мовлення. `ari.ai` пропонував веб-інтерфейс для створення та тестування сценаріїв розмов. Платформа базувалася на механізмі обробки природної мови, створеному Speaktoit для програми Assistant. `ari.ai` дозволяв розробникам Інтернету речей включати голосові інтерфейси природної мови у свої продукти. Веб-сайти Assistant і Speaktoit тепер переспрямовують на веб-сайт `ari.ai`, який переспрямовує на веб-сайт Dialogflow.

Google купив компанію у вересні 2016 року. Організація припинила роботу з програмою Assistant 15 грудня 2016 року. У жовтні 2017 року її пе-

рейменували на Dialogflow. У листопаді 2017 року Dialogflow став частиною Google Cloud Platform.

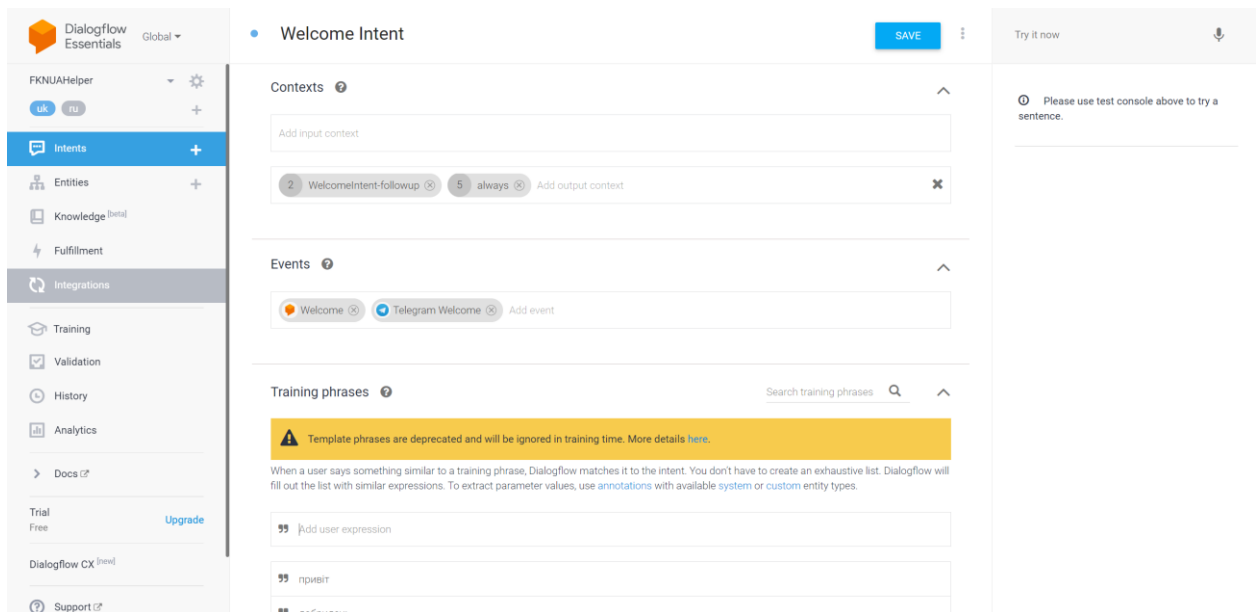


Рисунок 3.7 – Інтерфейс платформи Dialogflow

Однією з цілей Dialogflow є абстрагуватися від складнощів створення програми з використанням обробки природної мови та надати консоль, де користувачі можуть візуально створювати, проектувати та навчати інтерактивного помічника на основі штучного інтелекту.

Дослідники та інженери Google розробили власне спеціалізоване обчислювальне обладнання під назвою Tensor Processing Units (TPU). TPU – це тип інтегральних схем для певних програм (ASIC), які допомагають значно прискорити процеси машинного навчання (ML), наприклад в розмовних агентах. Використання TPU означає, що розмовні агенти з підтримкою ML скорочують час досягнення точності, оскільки вони навчаються складним алгоритмам і мережевим моделям. Те, на що раніше витрачалися місяці на звичайному обладнанні, займає години на TPU.

З огляду на широкі можливості Dialogflow для створення інтерактивних помічників з обробкою природної мови для використання в своїх служ-

бах, дуже розвинута документація, яка містить найкращі практики, стандартні вказівки щодо проектування та рішення для типових помилок та швидкодію алгоритмів машинного навчання, для реалізації помічника було обрано саме цю платформу.

4. ПРОЕКТУВАННЯ ІНТЕРАКТИВНОГО ПОМІЧНИКА

Після проведення аналізу предметної області, помічників-аналогів та вибору засобів розробки, необхідним етапом є проектування інформаційної системи. На цьому етапі було створено UML-діаграми варіантів використання (прецедентів), активності та послідовностей. Всі діаграми були створені в об'єктно-орієнтованому CASE-засобі проектування інформаційних систем IBM Rational Rose. IBM Rational Rose є інтегрованим засобом проектування архітектури, аналізу, моделювання та розробки ІС.

4.1 Проектування варіантів використання системи

Діаграма варіантів використання (прецедентів) відображає функціональне призначення проектованої програмної системи. Суть діаграми прецедентів полягає в тому, що систему представляють як групу акторів, які за допомогою варіантів використання взаємодіють із нею.

Актор – це сутність, що взаємодіє з системою для вирішення деяких завдань. Актором може бути людина, інша система, пристрій або програмний засіб. В даному випадку акторами визначено користувача інтерактивного помічника, адміністратора та систему Dialogflow, яка обробляє запити користувача. На діаграмі також зображена хмарна база даних.

Після визначення акторів системи, необхідно сформулювати перелік усіх варіантів використання, з якими будуть взаємодіяти визначені актори. Серед основних варіантів використання системи: з боку користувача – отримання розкладу занять, інформації про кафедри, посилань на сайти дистанційного навчання, інформації про вартість навчання, вступ, спеціальності та навчальні програми тощо, з боку адміністратора – додавання розкладу, зміна або видалення його з бази, додавання тренувальних фраз та відповідей до агента.

За результатами сформованих варіантів використання та акторів системи було розроблено діаграму варіантів використання, яку наведено на рис. 4.1.

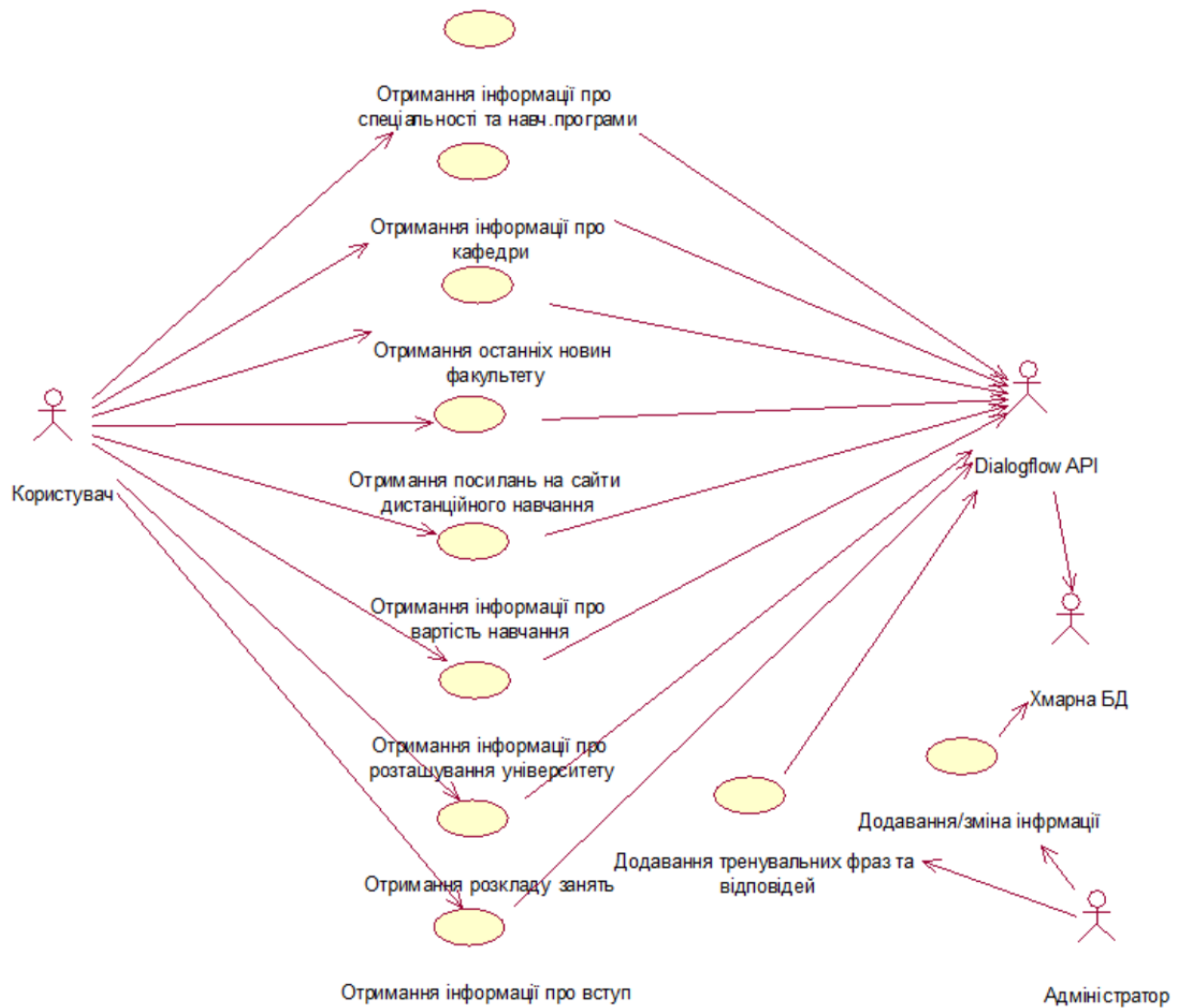


Рисунок 4.1 – Діаграма прецедентів

4.2 Створення діаграми діяльності

Для моделювання процесу виконання операцій в мові UML використовуються діаграми діяльності, або діаграма активностей. На діаграмі діяльності відображається логіка або послідовність переходу від однієї діяльності до іншої, при цьому увага фіксується на результаті діяльності. Сам результат може привести до зміни стану системи або повернення деякого значення. Діаграма діяльності візуально представляє ряд дій або потік керування в системі, подібний до блок-схеми або діаграми потоку даних. Діаграми діяльності можуть описувати кроки на діаграмі варіантів використання. Змодельовані

дії можуть бути послідовними та одночасними. В обох випадках діаграма діяльності має початок (початковий стан) і кінець (кінцевий стан).

Для системи була створена діаграма діяльності, яка відображає основний процес проектованої системи – введення користувачем запиту та його обробка помічником (рис. 4.2).



Рисунок 4.2 – Діаграма діяльності

4.3 Проектування логіки сценаріїв використання

Для більш детального опису логіки сценаріїв використання і уточнення діаграм прецедентів використовуються діаграми послідовності.

Документація Dialogflow наводить загальну діаграму послідовності спілкування з інтерактивним помічником, створеним за допомогою системи та інтегрованим з власним веб-хуком [14] (рис. 4.3).

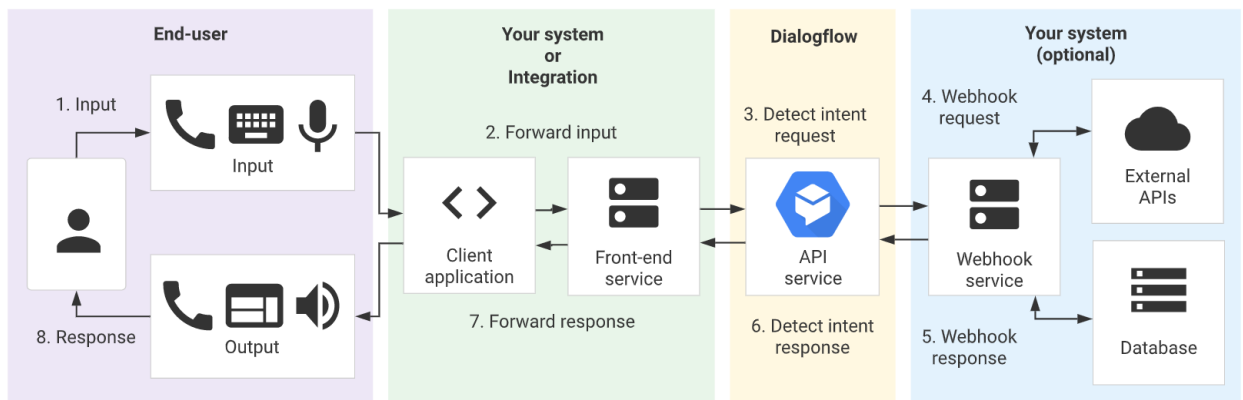


Рисунок 4.3 – Послідовність кроків при спілкуванні з агентом Dialogflow

Головна послідовність сценарію спілкування з інтерактивним помічником ФКНУА ОДЕКУ також складається з взаємодій, які відображені на цій схемі: кінцевий користувач робить запит природною мовою через Telegram. Telegram API передає цей запит в форматі JSON до Dialogflow API для визначення наміру. API Dialogflow отримує запит на виявлення наміру. Вбудована система NLP зіставляє вхідні дані з намірами та параметрами, встановлює необхідні параметри та оновлює стан сеансу. Якщо в намірі активований веб-хук, Dialogflow надсилає запит службі веб-хуку. Служба веб-хуку (Express), розташована на Heroku, отримує запит та виконує необхідні дії, як-от виклик зовнішніх API, надсилання запитів отримання даних або оновлення бази даних тощо. Служба веб-хуку створює відповідь і надсилає її назад у Dialogflow. Dialogflow створює відповідь на виявлення наміру. Якщо було

викликано веб-хук, він використовує відповідь, надану у відповіді веб-хуку. Якщо веб-хук не викликався, він використовує статичну відповідь, визначену в агенті. Dialogflow надсилає відповідь до Telegram API, який відображає її кінцевому користувачу.

UML-діаграма послідовності зображена на рис. 4.4:

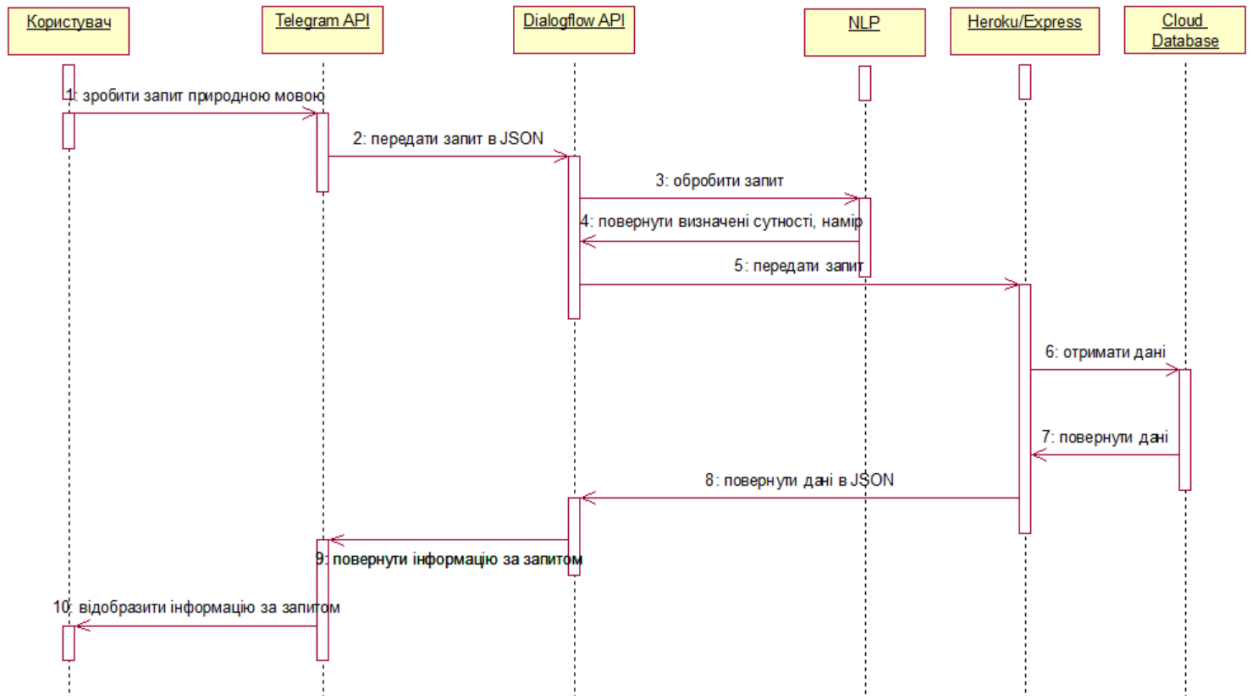


Рисунок 4.4 – Діаграма послідовностей

Таким чином, на етапі проектування системи були створені UML-діаграми прецедентів, активності і послідовності, які допомогли зрозуміти основні варіанти використання системи та логіку роботи агента.

5. ПРОГРАМНА РЕАЛІЗАЦІЯ «ПОМІЧНИКА ФКНУА ОДЕКУ»

5.1 Створення агента в Google Dialogflow

У Dialogflow існує певна термінологія. Основними термінами є агенти, наміри, сутності, навчальні фрази, контекст, база знань, Fulfillment.

Агент у Dialogflow представляє інтерактивного помічника, створеного користувачем для взаємодії з іншими кінцевими користувачами та виконання операцій обробки даних щодо інформації, яку він отримує. Інші компоненти об'єднуються, щоб утворити агента, і кожного разу, коли один із цих компонентів оновлюється, агент негайно проходить повторне навчання, щоб зміни вступили в силу.

Користувачі, які хочуть створити повноцінного інтерактивного помічника у найкоротші терміни, можуть вибрати агента з попередньо створених агентів. Це шаблони, які містять основні наміри та відповіді, необхідні для розмовного помічника.

Намір, подібно до буквального значення, є кінцевою метою користувача в кожному реченні під час взаємодії з агентом. Для одного агента можна створити кілька намірів для обробки кожного речення в розмові і з'єднати між собою за допомогою контекстів.

Виходячи з наміру, агент може зрозуміти кінцеву мету речення. Наприклад, агент, створений для обробки замовлень їжі, може зрозуміти, що кінцева мета клієнта – розмістити замовлення на страву або отримати рекомендації щодо доступних страв із меню за допомогою створених намірів.

Сутності — це засіб, за допомогою якого Dialogflow обробляє та витягує певні дані з введених кінцевим користувачем даних. Прикладом цього є сутність «Марка автомобіля», додана до наміру. Назви транспортних засобів будуть витягнуті з кожного вхідного речення як сутності «Марка автомобіля».

За замовчуванням агент має деякі системні сутності, які були попередньо визначені під час його створення. Dialogflow також має можливість ви-

значати користувацькі сутності та додавати значення, які розпізнаються в цій сутності.

Навчальні фрази є основним способом, за допомогою якого агент може розпізнати намір кінцевого користувача, який взаємодіє з агентом. Наявність великої кількості навчальних фраз у намірі підвищує точність розпізнавання наміру агентом. Документація Dialogflow рекомендує додати до створеного наміру принаймні 10-20 навчальних фраз.

Щоб зробити навчальні фрази зручнішими для повторного використання, Dialogflow дає можливість анотувати конкретні слова у навчальній фразі. Коли слово у фразі анотовано, Dialogflow розпізнає його як заповнювач для значень, які будуть надані під час введення кінцевим користувачем.

Контексти — це імена рядків, які використовуються для керування потоком розмови з агентом. Для кожного наміру можна додати кілька контекстів введення, а також кілька контекстів виводу. Коли кінцевий користувач створює речення, яке розпізнається наміром, вихідні контексти стають активними, і один із них використовується для відповідності наступному наміру.

Можна представити контекст як вхідні та вихідні двері, а намір як будівлю. Контекст введення використовується під час входу в будівлю, і приймаються лише відвідувачі, які були перераховані в намірі, тоді як вихідні двері є тим, що з'єднує відвідувачів з іншою будівлею, яка є іншим наміром.

База знань являє собою великий набір інформації, звідки агент може отримати дані під час відповіді на намір. Це може бути документ у будь-якому форматі, наприклад, txt, pdf, csv та інші підтримувані типи документів. У машинному навчанні базу знань можна назвати навчальним набором даних.

Прикладом сценарію, коли агент може посилатися на базу знань, є ситуація, коли агента використовують для отримання додаткової інформації про послугу чи бізнес. У цьому сценарії агент може звертатися до розділу «Поширені запитання» служби як до своєї бази знань.

Dialogflow Fulfillment дозволяє агенту давати більш динамічну відповідь на розпізнаний намір, а не статично створену відповідь. Це може бути виклик визначеної служби для виконання такої дії, як створення або отримання даних із бази даних. Виконання наміру досягається за допомогою вебхука. Після ввімкнення відповідний намір надсилає запит API до вебхука, налаштованого для агента Dialogflow.

Консоль Dialogflow – це місце, де агент створюється, проектується та навчається перед інтеграцією з іншими службами. Dialogflow також надає кінцеві точки REST API для користувачів, які не хочуть використовувати консоль під час створення агентів за допомогою Dialogflow.

У кожному новоствореному проекті Dialogflow пропонує створити агента з такими полями:

- Ім'я для ідентифікації агента.
- Мова, якою відповідатиме агент. За замовчуванням використовується англійська.
- Проект у Google Cloud, з яким можна пов'язати агента.

Одразу після натискання «Створити» буде збережений новий агент, і буде показана вкладка «Наміри» з намірами «Резервний намір за замовчуванням» і «Намір привітання за замовчуванням». Ці два наміри створюються за замовчуванням для кожного агента Dialogflow. Намір привітання за замовчуванням – коли кінцевий користувач починає розмову з агентом. Резервний намір за замовчуванням – коли агент не розпізнає вираз користувача.

У намірі «Резервний намір за замовчуванням» немає навчальних фраз. Навчальні фрази резервного наміру – це негативні приклади, які агент не співставлятиме з жодним іншим наміром. Однак є відповіді, такі як «Я тебе не розумію», «Це мені не дуже зрозуміло», «Перепрошую, повтори ще раз», «Не розумію, що ти маєш на увазі?» тощо. Ці відповіді вказують на те, що агент не зміг розпізнати речення кінцевого користувача. Під час будь-яких розмов із агентом ці відповіді використовуються лише тоді, коли агент не може розпізнати речення, яке вводить або вимовляє користувач. Хоча цих речень до-

статньо, щоб вказати, що агент не розуміє останнє введене речення, можна допомогти кінцевому користувачеві, надавши йому додаткову інформацію, щоб підказати користувачеві, що агент може розпізнати. Для цього можна замінити наведені речення та натиснути «Зберегти».

У намірі «Привітання за замовчуванням» перший розділ – це розділ «Контекст». Контексти введення та виведення спочатку порожні. Виходячи з потоку розмови агента можна створити потрібні контексти, які будуть ставати активними, коли наміри збігаються.

Після розділу «Контекст» йдуть «Події» наміру. До списку подій у намірі «Привітання за замовчуванням» є тип події Welcome, що вказує на те, що цей намір використовуватиметься першим під час завантаження агента.

Далі йдуть навчальні фрази для наміру. Оскільки він створений за замовчуванням, він уже містить декілька фраз, які кінцевий користувач, швидше за все, введе або вимовить, коли вперше взаємодіє з агентом, такі як «привіт», «добрий день», «вітаю» тощо. Коли кінцевий користувач введе речення, подібне до перелічених у навчальних фразах, агент відповість, використовуючи вибрану відповідь із розділу списку відповідей. Кожного разу після додавання та збереження нових фраз, Dialogflow негайно перенавчає агента.

Останній розділ на сторінці намірів – це розділ «Виконання» («Fulfillment»). Він використовується для надання даних агенту для використання як відповідей даних від зовнішнього API або джерела. Щоб використовувати його, необхідно увімкнути опцію виклику Webhook у розділі «Виконання» та налаштувати виконання для цього агента на вкладці «Виконання».

На вкладці «Виконання» на консолі розробник має можливість використовувати вебхук, який дає можливість використовувати будь-який розгорнутий API через його кінцеву точку або використовувати редактор вбудованого коду для створення безсерверної програми для розгортання як хмарної функції в Google Cloud.

Кожного разу, коли кінцевий користувач взаємодіє з агентом і намір збігається, до кінцевої точки надсилається запит POST. Серед різних полів об'єкта в тілі запиту є об'єкт `queryResult`. Його структура має вигляд:

```
{
  "queryResult": {
    "queryText": "Вираз кінцевого користувача",
    "parameters": {
      "param-name": "Значення параметру"
    }
  }
}
```

Хоча в `queryResult` є інші поля, часто об'єкт `parameters` є важливішим, оскільки він містить параметр, витягнутий із тексту користувача. Цей параметр буде інформацією, яку запитує користувач, і її можна використовувати для запиту до бази даних.

Cloud Functions діє як посередник у всій структурі. Агент Dialogflow надсилає параметр, отриманий із тексту кінцевого користувача, до хмарних функцій в даних запиту, а хмарна функція, у свою чергу, запитує базу даних для документа, використовуючи отримані дані, і надсилає запитувані дані у відповіді до агента.

Щоб реалізувати наведену систему проектування, необхідно створити хмарну функцію локально на машині розробки, а потім підключити її до агента Dialogflow за допомогою спеціальної опції веб-хуку. Щоб створити та розгорнути хмарну функцію, можна використовувати вбудований редактор на вкладці «Виконання».

Якщо з будь-якої причини запит на веб-хук буде невдалим, Dialogflow відреагує за допомогою однієї з відповідей у списку. Однак можна з'ясувати, чому запит не вдалося виконати, використовуючи інструмент діагностичної інформації, який оновлюється для кожної розмови. У ньому є вкладки «Відповідь Raw API», «Запит на виконання», «Відповідь на виконання» та «Статус виконання», які містять дані у форматі JSON. Вибравши вкладку «Відповідь на виконання», ми можемо побачити відповідь від вебхука, який є хмарною функцією, що працює на нашій локальній машині.

При додаванні навчальних фраз не вказуються всі можливі варіанти запитів користувача. Щоб зробити значення у фразі динамічним, в Dialogflow є сутності. Сутності представляють загальні типи даних. Існують системні сутності. Деякі приклади наведені в табл. 2.

Таблиця 2 – Системні сутності Dialogflow

Сутність	Назва	Приклад фрази користувача	Приклади вихідних даних
Дата і час	@sys.date-time	завтра о 4:30 вечора завтра по обіді	{ "date_time": "2018-04-06T16:30:00-06:00" } { "startDateTime": "2018-04-06T12:00:00-06:00", "endDateTime": "2018-04-06T16:00:00-06:00" }
Дата	@sys.date	завтра	2018-04-06T12:00:00-06:00
Період дат	@sys.date-period	квітень	{ "startDate": "2018-04-01T12:00:00-06:00", "endDate": "2018-04-30T12:00:00-06:00" }
Час	@sys.time	4:30 вечора	2018-04-05T16:30:00-06:00
Період часу	@sys.time-period	післяобідній час	{ "startTime": "2018-04-05T12:00:00-06:00", "endTime": "2018-04-05T16:00:00-06:00" }
Число	@sys.number	один	1
Номер рейсу	@sys.flight-number	LH4234	LH 4234
Адреса ел. пошти	@sys.email	user@example.com example собака gmail крапка com	"user@example.com" "example@gmail.com"
Номер телефону	@sys.phone-number	(123) 456 7890 +1 (123) 456-7890	"1234567890" "+11234567890"
Будь-які дані	@sys.any	квітка	квітка
URL	@sys.url	www.dialogflow.com	www.dialogflow.com

Dialogflow також дає розробникам можливість створювати користувацькі сутності для використання. Для цього необхідно натиснути «Створити сутність», дати назву новій сутності, а потім вказати можливі значення та їх синоніми. Приклад створеної сутності «факультети» наведений на рис. 5.1.

faculties SAVE

Define synonyms Regex entity Allow automated expansion Fuzzy matching

Факультет комп'ютерних наук, управління та адміністрування	Факультет комп'ютерних наук, управління та адміністрування, ФКНУА, ФКН, комп'ютерних наук, менеджменту, Факультет КНУА
Гідрометеорологічний інститут	Гідрометеорологічний інститут, гідромет, гідро, метеорологічний
Природоохоронний факультет	Природоохоронний факультет, природоохоронний, охорона природи

[Click here to edit entry](#)

Рисунок 5.1 – Створена сутність «faculties»

Якщо поставити галочку «Define synonyms», це дозволить вказати фрази, які може ввести користувач і які відносяться до цього ж значення сутності. «Regex entity» дозволяє створювати сутності за допомогою регулярних виразів [15].

При активному «Allow automated expansion» агент може розпізнавати значення, які не були надані явно, але контекстуально подібні до інших значень.

Працюючи з цим розширенням, необхідно враховувати наступне:

- Увімкнення автоматичного розширення не гарантує розпізнавання сутності. Для кінцевого списку слід надати повний список замість того, щоб надавати частковий список і вмикати автоматичне розширення.
- Якщо увімкнути автоматичне розширення в кількох об'єктах, це може спричинити конфлікти та неочікувані результати класифікації.

- Щоб забезпечити кращу якість вилучення параметрів, вкрай важливо надати різноманітні навчальні дані, які охоплюють усі випадки використання, у яких дану сутність можна знайти в очікуваному трафіку агента. Якщо прикладів недостатньо, автоматизоване розширення сутностей може не працювати належним чином.

Опція «Fuzzy matching» (Нечітка відповідність) – тип вилучення параметрів, які відповідають сутності приблизно, а не точно. При цьому Dialogflow буде намагатися знайти збіги, навіть якщо користувачі напишуть слова неправильно або введуть лише частину слів із значення сутності.

Після збереження значень сутності агент буде негайно перенавчений за допомогою нових значень, і після завершення навчання можна перевірити, ввівши текст у поле введення в правій частині. Відповіді в рамках наміру будуть отримані або з попередньо створених статичних варіантів відповідей, або функції за допомогою веб-хуку Fulfillment. Якщо намір не визначений або веб-хук не виконується успішно, отримується відповідь, яка слугує резервним варіантом при помилці.

5.2 Розгортання агента на платформі Heroku

Веб-хуки – це сервіси, на яких розміщується бізнес-логіка. Під час сеансу веб-хуки дозволяють використовувати дані, отримані за допомогою обробки природної мови Dialogflow, для створення динамічних відповідей, перевірки зібраних даних або ініціювання дій на сервері.

Щоб використовувати веб-хук, необхідно вказати його URL. Вказаний сервер отримає запит POST від Dialogflow у формі відповіді на запит користувача, який відповідає якомусь з намірів із ввімкненим веб-хуком. Веб-хук повинен відповідати вимогам версії API, що використовується в агенті:

- Він повинен обробляти запити HTTPS. HTTP не підтримується.
- URL-адреса для запитів має бути загальнодоступною.

- Він повинен обробляти запити POST за допомогою JSON WebhookRequest і відповідати за допомогою JSON WebhookResponse.
- Якщо агент не інтегрується з доступом до приватної мережі Service Directory, виклики веб-хука вважаються поза периметром служби та блокуються під час увімкнення засобів керування службами VPC.

Консоль Dialogflow має вбудований редактор коду, який можна використовувати для створення коду виконання та розгортання коду в Cloud Functions. Cloud Functions обов'язково прив'язується до Cloud Billing (платіжного аккаунта Google), однак ця послуга доступна безкоштовно до певної кількості щомісячних викликів. Незалежно від масштабів проекту, необхідно зареєструватися та надати дійсний платіжний рахунок.

Після вмикання вбудованого редактора, код виконання Node.js попередньо заповнюється обробниками за замовчуванням для намірів за замовчуванням, які включено для всіх агентів. Код також містить інструкції з коментарями щодо додавання обробників для намірів, визначених розробником. Вбудований редактор призначений для простого тестування виконання та створення прототипів. Для розгортання робочого агента, Google Dialogflow рекомендує використовувати інші служби. Одна з таких служб – Heroku.

Heroku — це хмарна платформа як послуга (PaaS), що підтримує різні мови програмування. Це одна з перших хмарних платформ, яка була розроблена у червні 2007 року і підтримувала лише мову програмування Ruby. Наразі вона підтримує Java, Node.js, Scala, Clojure, Python, PHP і Go. З цієї причини Heroku називають поліготною платформою, оскільки вона має функції для розробника, щоб створювати, запускати та масштабувати програми більшістю мов.

Мережа Heroku запускає програми клієнта у віртуальних контейнерах, які виконуються в надійному середовищі виконання [16]. Heroku називає ці контейнери «Dynos» (Діно). Ці Діно можуть запускати код, написаний на Node, Ruby, PHP, Go, Scala, Python, Java або Clojure. Кожен Діно розповсюджений по «сітці Діно», яка складається з кількох серверів. Програми, які за-

пускаються на Heroku, зазвичай мають унікальний домен, який використовується для маршрутизації HTTP-запитів до правильного контейнера програми. Heroku дозволяє розробнику миттєво масштабувати програму, просто збільшуючи кількість Діно або змінюючи тип Діно, у якому працює додаток.

Керування програмами на Heroku здійснюється за допомогою Git. Усі сервіси Heroku розміщені на платформі хмарних обчислень Amazon EC2. Для розгортання програми потрібно просто перенести кодову базу в Heroku.

В даній роботі було вирішено створити API за допомогою Express.js та розмістити на Heroku.

Node.js – середовище виконання коду JavaScript поза браузером, яке дозволяє писати серверний код для веб-сторінок та веб-застосунків, а також для програм командного рядка. З цим середовищем виконання JavaScript легко працювати, і воно забезпечує виконання коду на стороні сервера. Також воно забезпечує високу масштабованість, а також прискорює цикли розробки. Node.js – одна з найкращих технологій із розвиненою спільнотою open source, тому її використовують компанії всіх масштабів, серед яких Netflix, NASA, Trello, PayPal, LinkedIn тощо. На практиці розробка на традиційному Node.js практично не ведеться, а використовується Express.

Express (Express.js) - це фреймворк для Node.js, який реалізує шар функцій, необхідних створення ефективних додатків і API [17]. Node.js Express має готові функції обробки HTTP запитів, причому для кожного методу HTTP є своя функція, що особливо зручно при створенні REST API. Метод `use()` використовується для створення проміжних обробників – Middleware.

Важливим елементом при розробці на Node.js/Express є NPM. NPM (англ. – node package manager) - це менеджер пакетів JavaScript, який:

- допомагає публікувати проекти на Node.js, які розповсюджуються з відкритим вихідним кодом;
- допомагає взаємодіяти з сервером, браузером та онлайн-майданчиками за допомогою командного рядка;

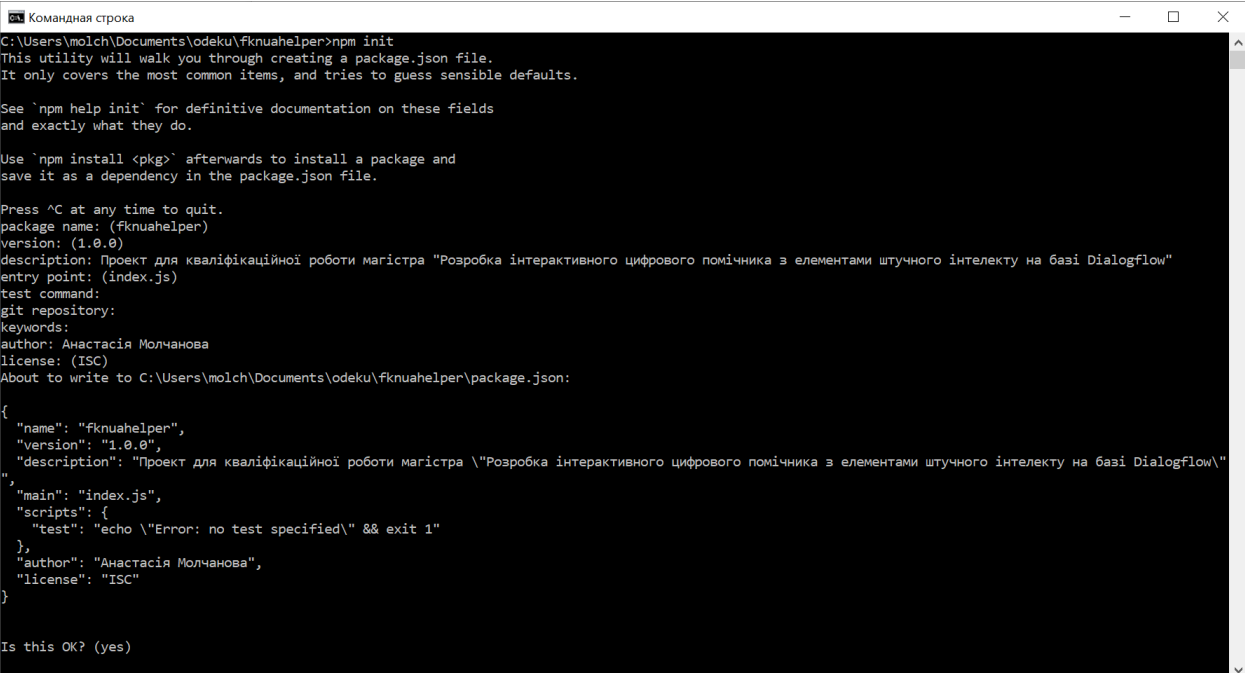
- допомагає встановлювати, видаляти та керувати пакетами, що використовуються для запуску проекту тощо.

NPM складається з двох важливих частин:

- інтерфейсу командного рядка - інструмент для взаємодії з пакетами;
- онлайн-репозиторія - місця, де зберігаються JavaScript-пакети.

Онлайн-репозиторій NPM розміщується за адресою www.npmjs.com. По суті, він виконує роль «інтернет-магазину» пакетів, де автори пакетів можуть розмістити їх задля загального користування, а користувачі можуть скористатися будь-яким представленим пакетом.

Щоб створити проект, треба створити головний файл `index.js` та ініціалізувати запуск проекту за допомогою команди `$ npm init` у терміналі. Ця команда створює з проекту файл із розширенням `.json`. Після введення команди спливатимуть підказки від «`npm init`», на які потрібно буде відповісти і прийняти їх. Після підтвердження та прийняття підказок `npm init` продемонструє сформований «`package.json`» (рис. 5.2). Потім завжди є можливість його відредагувати, якщо це знадобиться.



```

Командная строка
C:\Users\molch\Documents\odeku\fknuahelper>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (fknuahelper)
version: (1.0.0)
description: Проект для кваліфікаційної роботи магістра "Розробка інтерактивного цифрового помічника з елементами штучного інтелекту на базі Dialogflow"
entry point: (index.js)
test command:
git repository:
keywords:
author: Анастасія Молчанова
license: (ISC)
About to write to C:\Users\molch\Documents\odeku\fknuahelper\package.json:
{
  "name": "fknuahelper",
  "version": "1.0.0",
  "description": "Проект для кваліфікаційної роботи магістра \"Розробка інтерактивного цифрового помічника з елементами штучного інтелекту на базі Dialogflow\"",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Анастасія Молчанова",
  "license": "ISC"
}
Is this OK? (yes)

```

Рисунок 5.2 – Ініціалізація проекту та формування `package.json`

Також через термінал можна встановити потрібні npm-модулі. Наприклад, модуль `express` встановлюється командою:

```
npm install express --save
```

Всі завантажені npm-пакекти встановлюються в проєкті в папку з ім'ям `node_modules`.

Для використання `Express` на початку треба створити об'єкт, який представлятиме додаток:

```
const app = express();
```

Так як дані з `Dialogflow` відправляються на веб-хук за допомогою методу `POST`, для обробки необхідно визначити `Express` функцію `app.post()` [<https://www.geeksforgeeks.org/express-js-app-post-function/>]. Перший параметр функції – адреса, на яку йде відправка. Дані надсилаються за адресою веб-хука, на якому викликається «`dialogflow-fulfillment`». Другий параметр – проміжний обробник. Таким чином, функція може мати вигляд:

```
app.post('/dialogflow-fulfillment', (request, response) => {
  dialogflowFulfillment(request, response)
})
```

Приклад запиту `Fulfillment request`, який `Dialogflow` відправляє на бекенд-сервіс, коли увімкнено `Fulfillment`:

```
{
  "responseId": "52095926-2f5c-4ebd-94e8-aaca23fdb007-4e2f8dda",
  "queryResult": {
    "queryText": "Хто завкафедри ІТ?",
    "parameters": {
      "cafedra": "інформаційних технологій"
    },
    "allRequiredParamsPresent": true,
    "fulfillmentText": "Інформація про кафедру інформаційних технологій",
    "fulfillmentMessages": [
      {
        "text": {
          "text": [
            "Інформація про кафедру інформаційних технологій"
          ]
        }
      }
    ]
  },
}
```

```

"outputContexts": [
  {
    "name": "projects/odekuinfobotua-
mgoo/locations/global/agent/sessions/ede2eb91-4aef-4c0f-0d07-
f73b0603fedb/contexts/___system_counters___",
    "parameters": {
      "no-input": 0,
      "no-match": 0,
      "cafedra": "інформаційних технологій",
      "cafedra.original": "ІТ"
    }
  }
],
"intent": {
  "name": "projects/odekuinfobotua-
mgoo/locations/global/agent/intents/0c69d221-2f56-493f-91b1-
ab07ac66bcd1",
  "displayName": "Cafedra Info Intent"
},
"intentDetectionConfidence": 1,
"diagnosticInfo": {
  "query_sentiment_analysis_error": "INVALID_ARGUMENT: The
language ru is not supported for document_sentiment analysis"
},
"languageCode": "uk"
},
"originalDetectIntentRequest": {
  "source": "DIALOGFLOW_CONSOLE",
  "payload": {}
},
"session": "projects/odekuinfobotua-
mgoo/locations/global/agent/sessions/ede2eb91-4aef-4c0f-0d07-
f73b0603fedb"
}

```

Для розміщення коду `node.js` на платформі Heroku необхідно зареєструвати аккаунт на `heroku.com`. Це безкоштовно. Для нового проекту треба створити новий застосунок: вказати його назву та обрати регіон для розміщення коду на відповідних серверах. Після створення застосунку Heroku надає покрокову інструкцію для його розгортання за допомогою Heroku Git через командний рядок. Зокрема, спочатку необхідно встановити Heroku CLI. Інтерфейс командного рядка Heroku (CLI) дозволяє створювати програми Heroku і керувати ними безпосередньо з терміналу.

```
$ heroku login
```

Використовуючи Git, необхідно клонувати вихідний код `fknu-helper` на локальну машину.

```
$ heroku git:clone -a fknu-helper
$ cd fknu-helper
```

Далі можна розгорнути свій код у Героку за допомогою Git.

```
$ git add .
$ git commit -am "опис змін"
$ git push heroku master
```

Для більш зручного завантаження коду можна налаштувати відповідні установки в IDE. Наприклад, в роботі використовується JetBrains Webstorm 2022.2.3 – інтегрована середа для розробки на JavaScript та пов'язаних з нею технологіях (рис. 5.3).

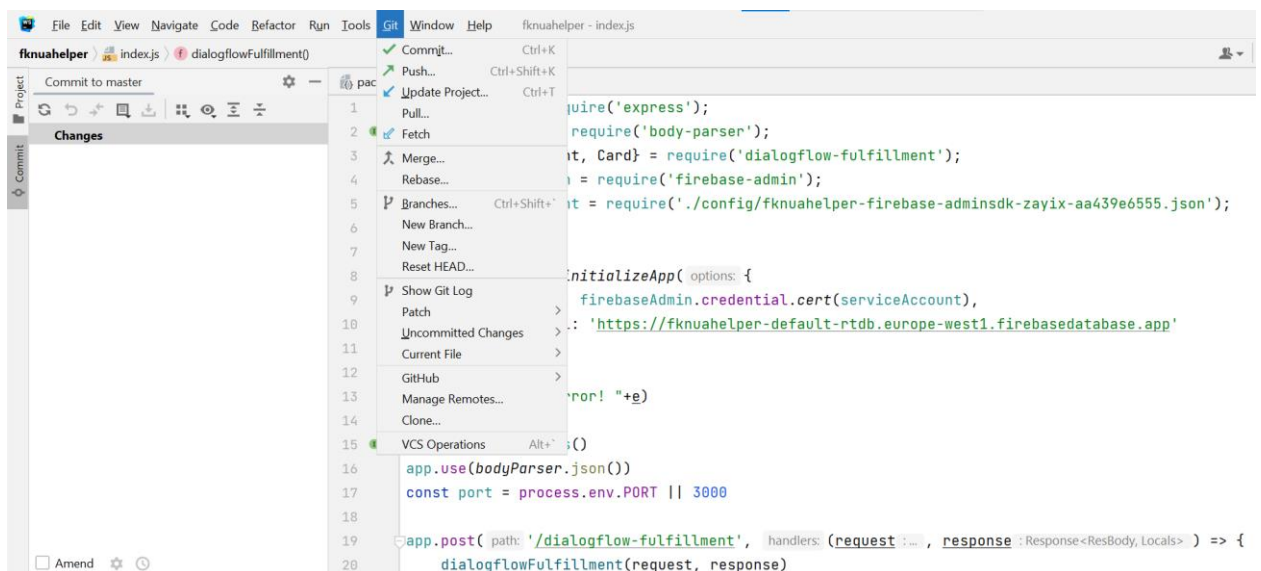


Рисунок 5.3 – Налаштування Git в IDE Webstorm

В налаштуваннях застосунку Героку можна знайти URL, за яким розміщений код застосунку. Цей URL необхідно вказати на сторінці Fulfillment в налаштуваннях агента Dialogflow з вказанням назви програми, яка викликається, - `dialogflow-fulfillment` (рис. 5.4).

⚡ Fulfillment

Webhook ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

BASIC AUTH

Рисунок 5.4 – URL веб-хуку в Dialogflow Fulfillment

Якщо в проєкті використовується безкоштовний Діно і він не отримує запити протягом 30 хвилин, то він автоматично переходить в режим сну. Це не підходить для інтерактивного помічника, який повинен завжди бути доступним. Тому в роботі було також використано платформу для моніторингу роботи застосунків One Relic APM як доповнення Героку. Її можна підключити до застосунку Героку та створити простий монітор, який буде раз у заданий час відправляти запити до веб-хуку і так тримати його активним (рис. 5.5).

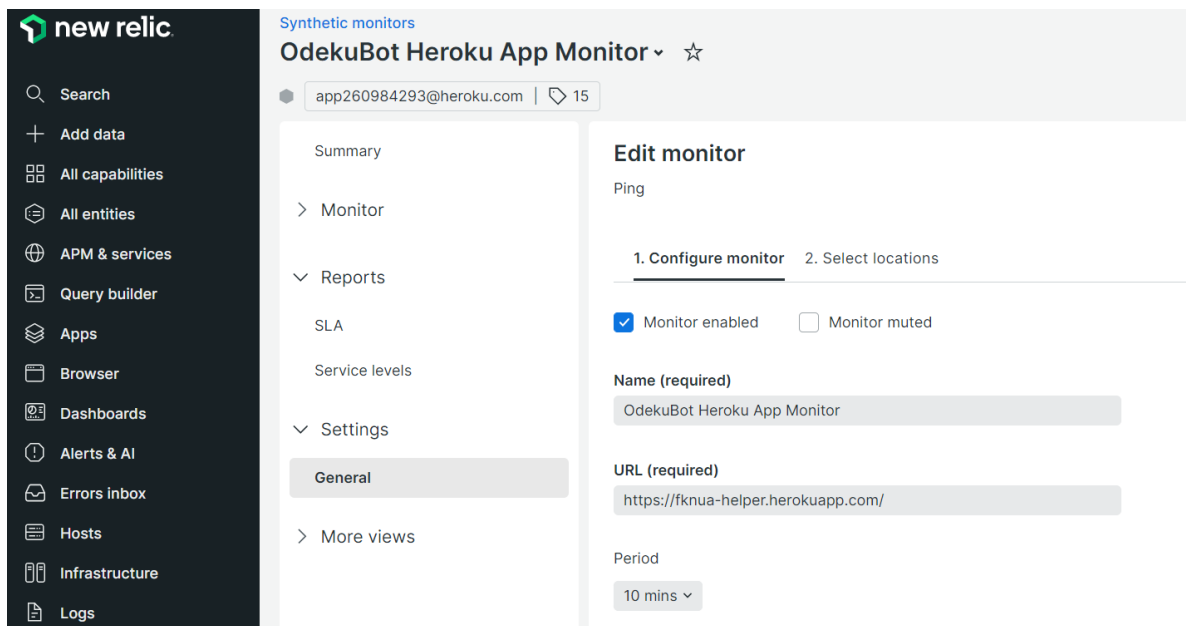


Рисунок 5.5 – Монітор One Relic

5.3 Підключення платформи Firebase

Для зберігання даних помічника було обрано хмарну базу даних Firebase Realtime Database. Firebase – це платформа від Google, розроблена для створення мобільних та веб-застосунків.

Дані в Firebase Realtime Database зберігаються в форматі JSON і синхронізуються в реальному часі з кожним підключеним клієнтом.

База даних Firebase Realtime дозволяє створювати багатofункціональні програми для спільної роботи, забезпечуючи безпечний доступ до бази даних безпосередньо з клієнтського коду. Дані зберігаються локально, і навіть в автономному режимі події в реальному часі продовжують активуватися, що дає кінцевому користувачу можливість швидко реагувати. Коли пристрій відновлює з'єднання, база даних реального часу синхронізує зміни локальних даних з віддаленими оновленнями, які відбулися, коли клієнт був в автономному режимі, автоматично об'єднуючи будь-які конфлікти.

Firebase Realtime Database – це база даних NoSQL, тому вона відрізняється оптимізацією і функціональністю в порівнянні з реляційною базою даних [18]. API бази даних реального часу призначені тільки для операцій, які можуть виконуватися швидко. Це дозволяє створювати відмінні умови для роботи в реальному часі, що є необхідним для інтерактивного помічника.

Безкоштовне використання Firebase Realtime Database має деякі обмеження:

- до 100 одночасних підключень до бази;
- до 1 ГБ даних, що зберігаються;
- до 360 МБ / на день даних, що завантажуються з бази.

Щоб додати базу Firebase Realtime Database до помічника, необхідно створити новий проект Google Firebase. Після створення бази треба отримати ключ для її адміністрування в розділі налаштувань «Firebase Admin SDK». В цьому розділі також наведений код для використання цього ключа в різних застосунках, включаючи написані на Node.js (рис. 5.6).

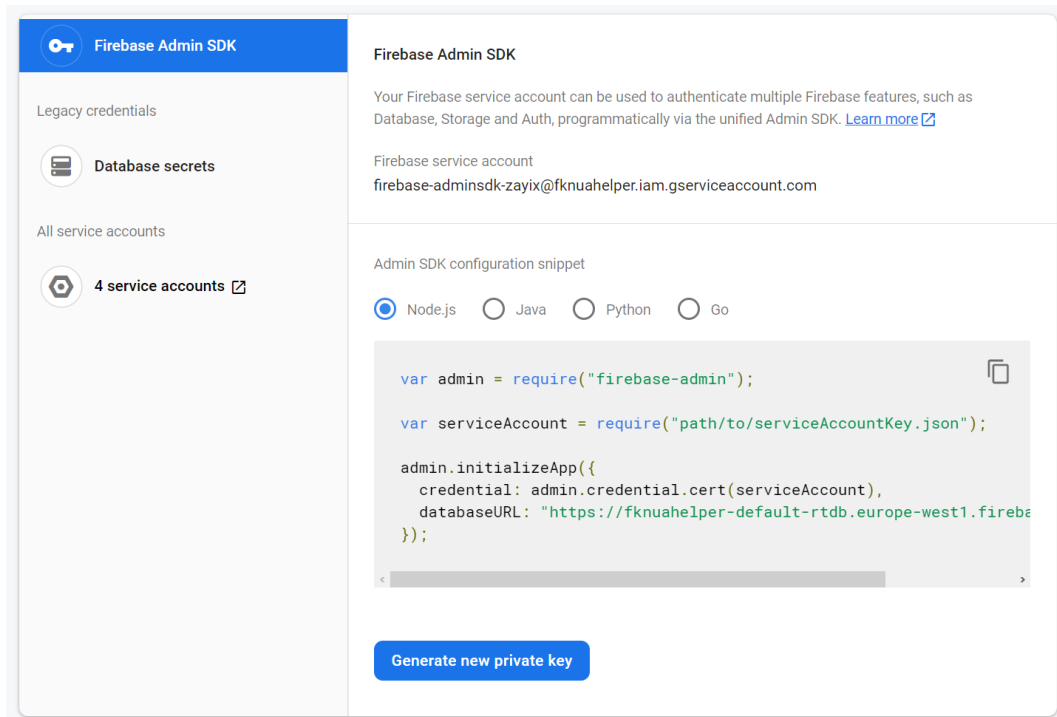


Рисунок 5.6 – Отримання ключа Firebase Admin SDK

Хоча база даних Realtime Database – нереляційна, основні її сутності можна представити на схемі даних (рис. 5.7).

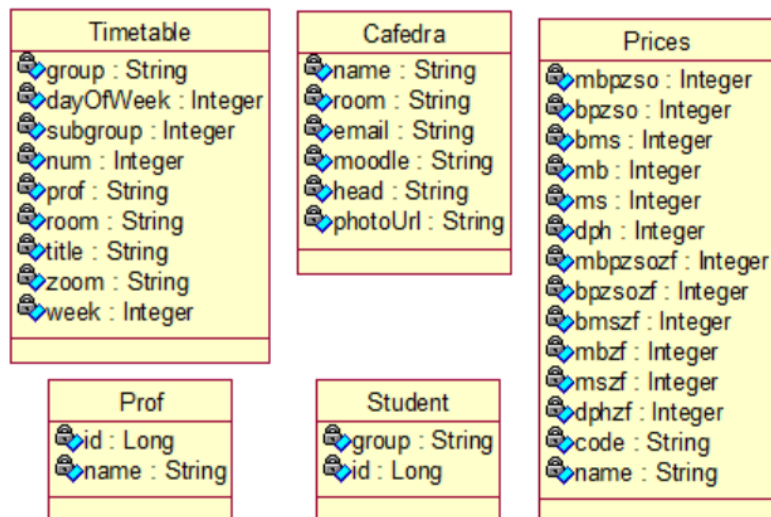


Рисунок 5.7 – Схема даних агента Dialogflow

Дані в Firebase Realtime Database зберігаються в форматі JSON. Приклад опису об'єкта у форматі JSON:

```
"safedra": {
  "інформаційних технологій": {
    "email": "kaf-infotech@odeku.edu.ua",
    "head": "д.техн.н., професор, Казакова Надія Феліксівна",
    "moodle": "dpt15s.odeku.edu.ua",
    "photoUrl": "odeku.edu.ua/wp-content/uploads/kozakova-2021.jpg",
    "room": "330 (1)"
  },
  "автоматизованих систем моніторингу навколишнього середовища та інформатики": {
    "email": "avto@odeku.edu.ua",
    "head": "к.т.н., доц. Перелигін Борис Вікторович",
    "moodle": "dpt01s.odeku.edu.ua",
    "photoUrl": "odeku.edu.ua/wp-content/uploads/perelyigin_301x322.jpg",
    "room": "124 (1)"
  }
}
```

5.4 Інтеграція помічника з Telegram

Для того, щоб інтегрувати помічника з месенджером Telegram, необхідно створити спеціальний акаунт, який називається «бот». Для цього потрібно скористатися @BotFather, який, по суті, є ботом для створення та адміністрування інших ботів.

Команда /newbot створює новий бот. Необхідно вказати назву нового профілю та псевдонім, який обов'язково повинне закінчуватись на «bot». Назва відобразатиметься як ім'я профіля, а псевдонім використовуватиметься для згадок і створення посилань. Після цього BotFather надасть токен API для інтеграцій з іншими платформами та управлінням помічником (рис. 5.8).

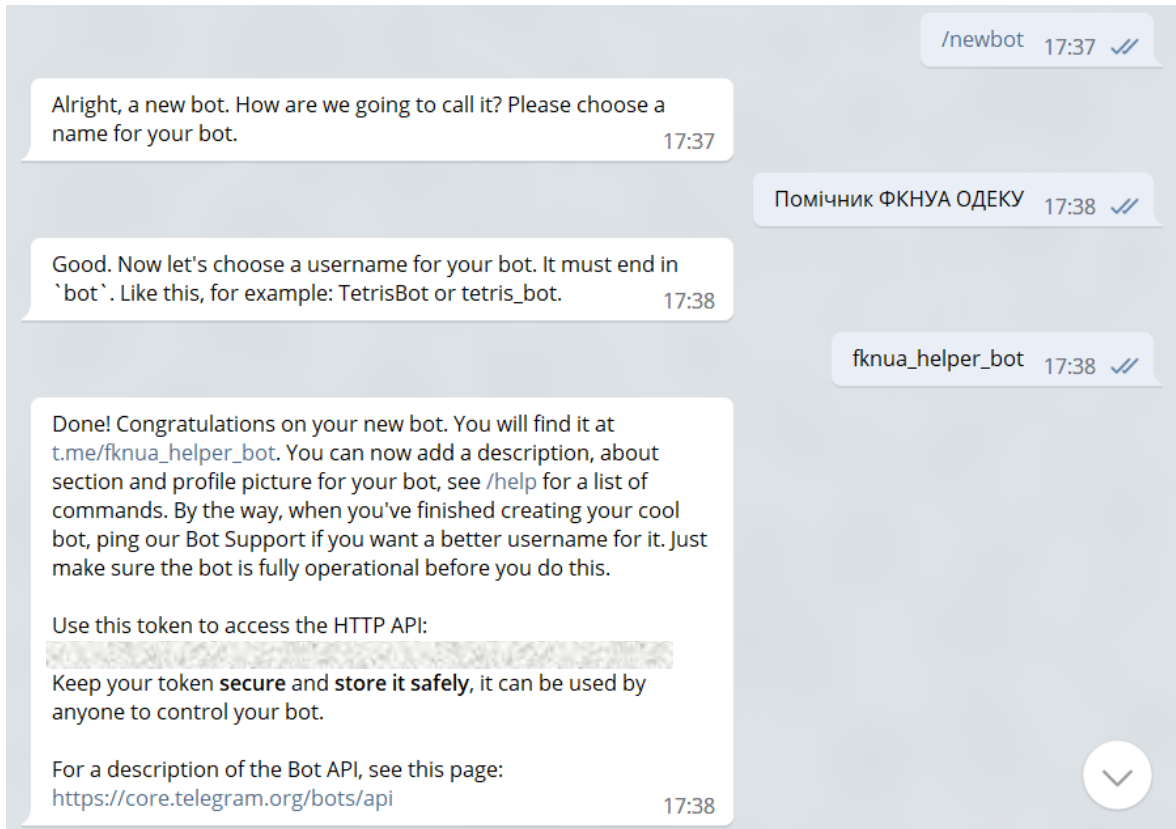


Рисунок 5.8 – Створення боту в Telegram

За допомогою відповідних команд можна додати текст опису та зображення опису, також головне зображення і основні команди, які будуть доступні як меню помічника (рис. 5.9).



Помічник ФКНУА ОДЕКУ

@fknu_helper_bot

Інформаційний помічник факультету КНУА
Одеського державного екологічного
університету 🏛️

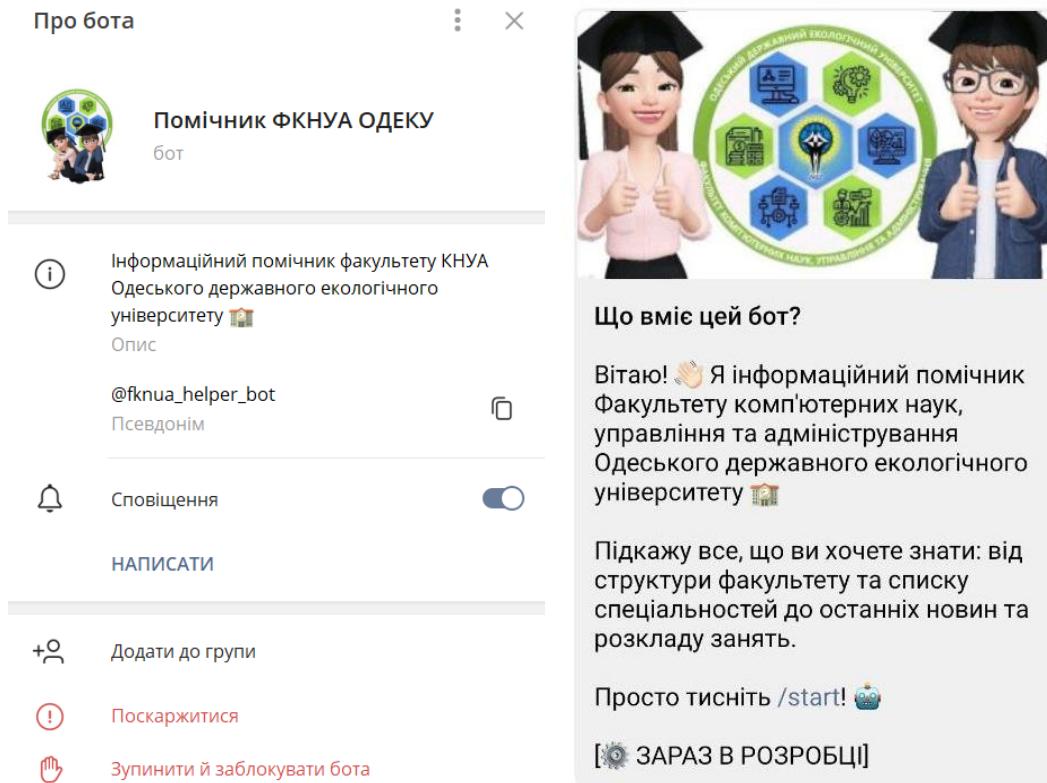


Рисунок 5.9 – Опис помічника ФКНУА ОДЕКУ в месенджері Telegram

Dialogflow надає розробникам можливість інтегрувати агента в різні платформи спілкування, включаючи платформи соціальних мереж, такі як Facebook Messenger, Telegram, Viber, Skype, Twitter, Line, Kik, Twillo, Slack тощо. У документації Dialogflow перераховано доступні типи інтеграцій і платформ у межах кожного типу інтеграції.

Для того, щоб інтегрувати агента помічника Dialogflow з Telegram, необхідно в розділі Integrations обрати Telegram зі списку платформ та ввести токен створеного бота [19].

5.5 Опис та тестування інтерактивного помічника

Для початку роботи з помічником, необхідно натиснути «Старт» в нижній частині екрану або ввести команду «/start». Помічник надішле вітальне повідомлення та коротку інструкцію, як ним користуватися (рис. 5.10)

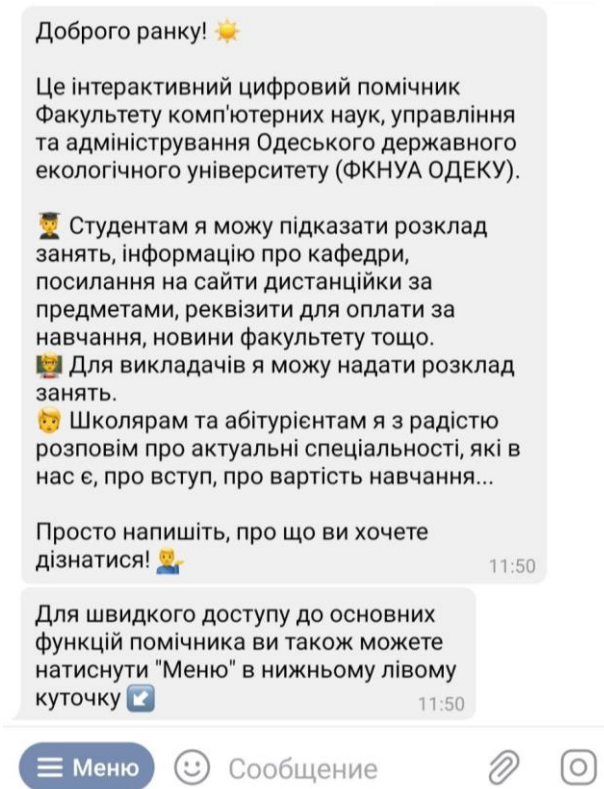


Рисунок 5.10 – Вітальне повідомлення помічника

Можна скористатися меню агента, в якому наведені його основні функції: отримання розкладу занять, посилань на сайти дистанційного навчання кафедр, інформації про кафедри, останніх новин факультету тощо (рис. 5.11).

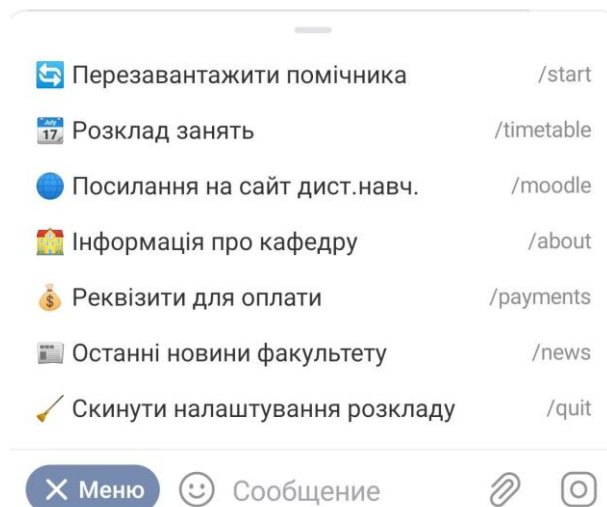


Рисунок 5.11 – Меню швидких команд помічника

Наприклад, якщо натиснути «Інформація про кафедру», агент уточнить, інформація саме про яку кафедру цікавить користувача, і після цього надасть основні відомості, включаючи аудиторію кафедри, електронну пошту, сайт дистанційного навчання, ім'я та фото завідуючого кафедри (рис. 5.12).

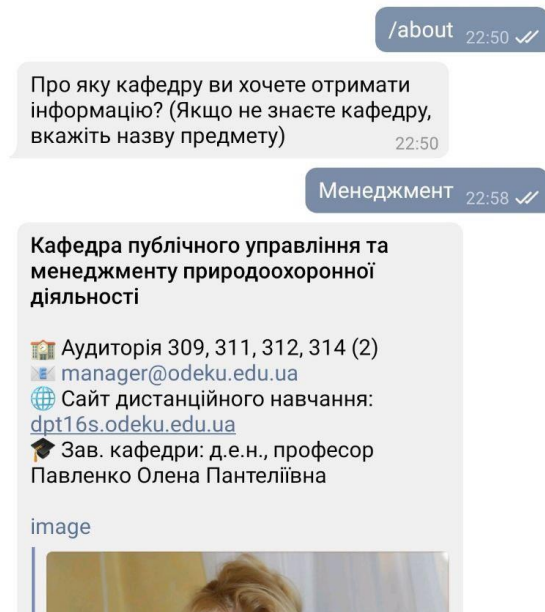


Рисунок 5.12 – Отримання інформації про задану кафедру

Той самий результат можна отримати, сформувавши запит природною мовою. Dialogflow розпізнає запит, навіть якщо користувач припустить помилки в словах і надасть неповну назву кафедри (рис. 5.13).

Подібна схема працює також з іншими темами, на які помічник може дати відповідь. Наприклад, щоб дізнатися останні новини факультету, можна ввести команду меню `«/news»` або спитати помічника «що нового», «новини» тощо (рис. 5.14). Це можливо завдяки тому, що Dialogflow не потребує вичерпного списку можливих фраз користувача, а за допомогою технологій NLP та мовної моделі самостійно розширює його подібними. Новини отримуються за допомогою парсингу з сайту факультету [20].

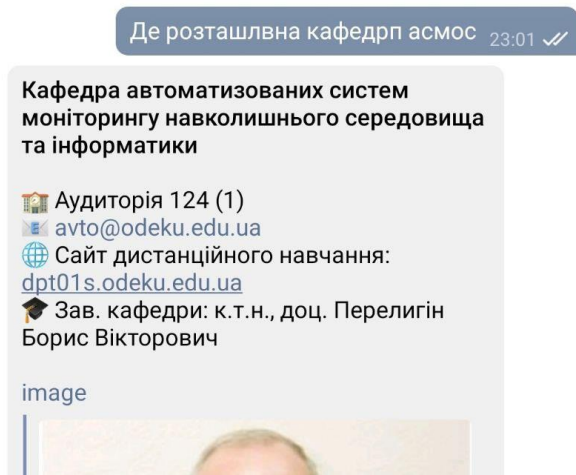


Рисунок 5.13 – Відповідь агента на запит природною мовою з помилками

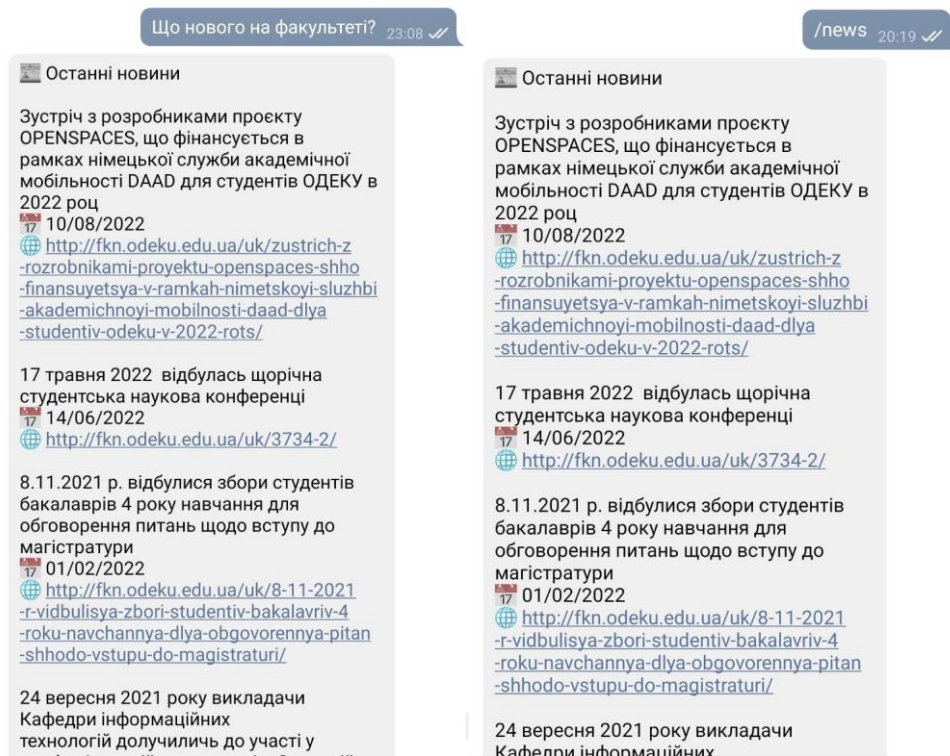


Рисунок 5.14 – Отримання останніх новин факультету

Для студентів та викладачів корисною функцією є розклад занять. Для того, щоб його переглянути, студентам необхідно ввести назву групи, а викладачам – своє прізвище (рис. 5.15). В базі даних помічника зберігається тільки ідентифікатор чату з користувачем та вказана група або прізвище.

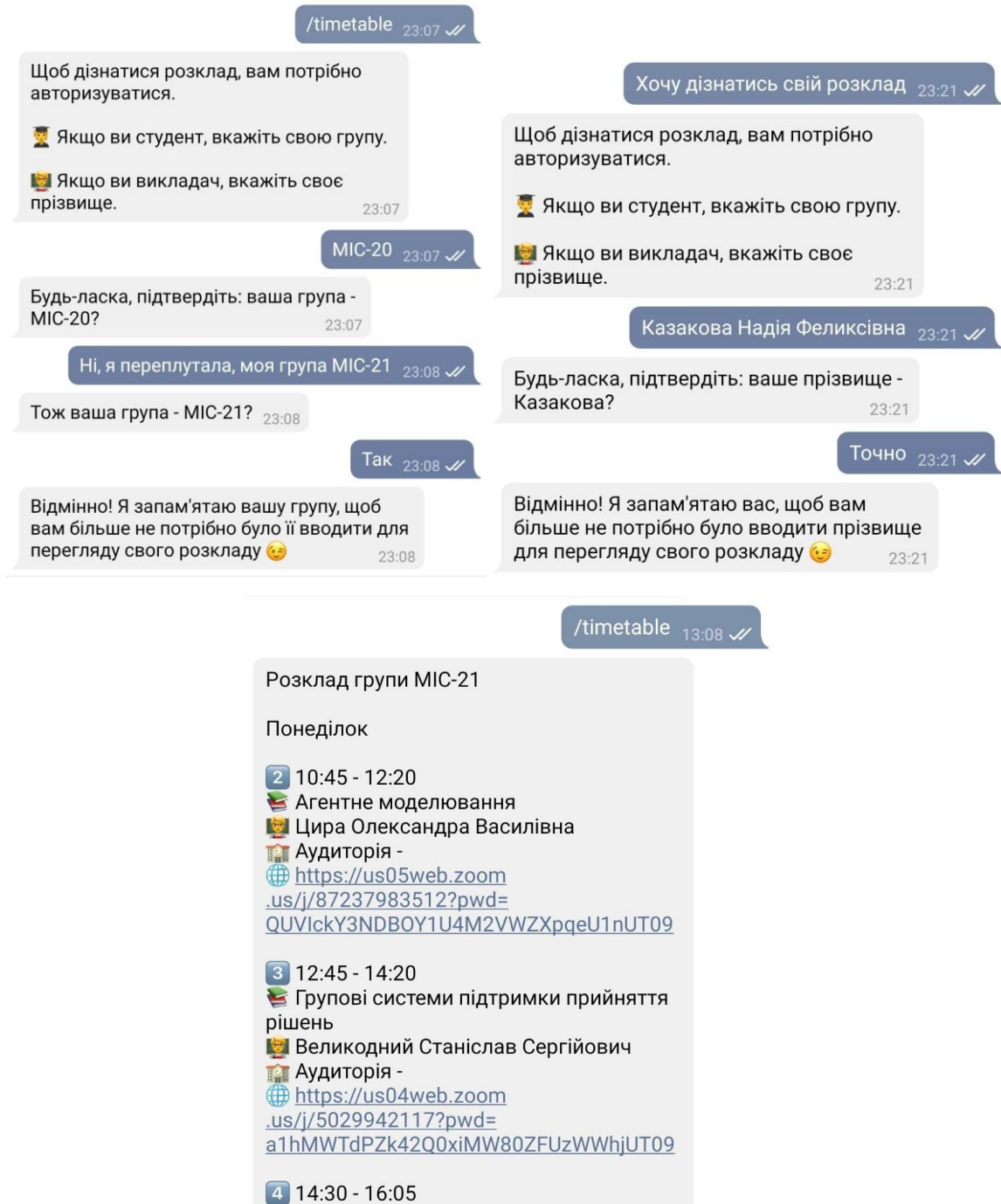


Рисунок 5.15 – Отримання розкладу занять студентів та викладачів

Студенти також можуть швидко отримати реквізити для оплати за надання освітніх послуг або проживання у гуртожитку (рис. 5.16).

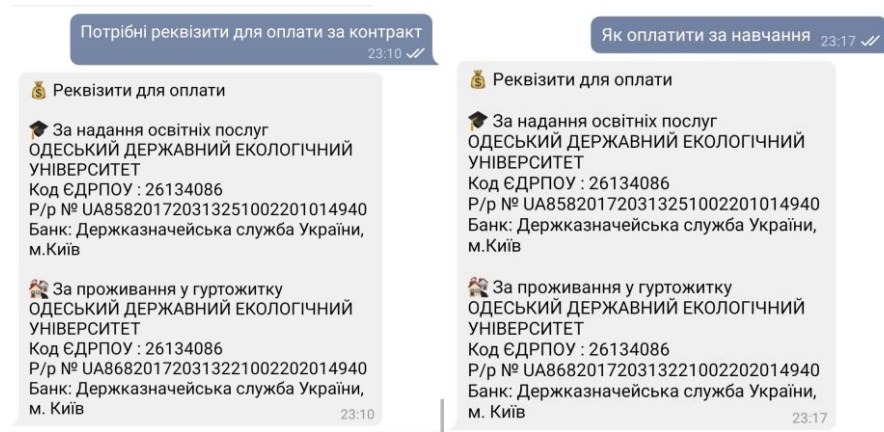


Рисунок 5.16 – Реквізити для оплати контракту

В сучасних умовах дистанційне навчання є невід’ємною частиною освітнього процесу. Студентам часто потрібно заходити на сторінки сайтів дистанційного навчання різних кафедр. Помічник ФКНУА ОДЕКУ може підказати, до якої саме кафедри відноситься предмет і надати посилання на moodle (рис. 5.17).

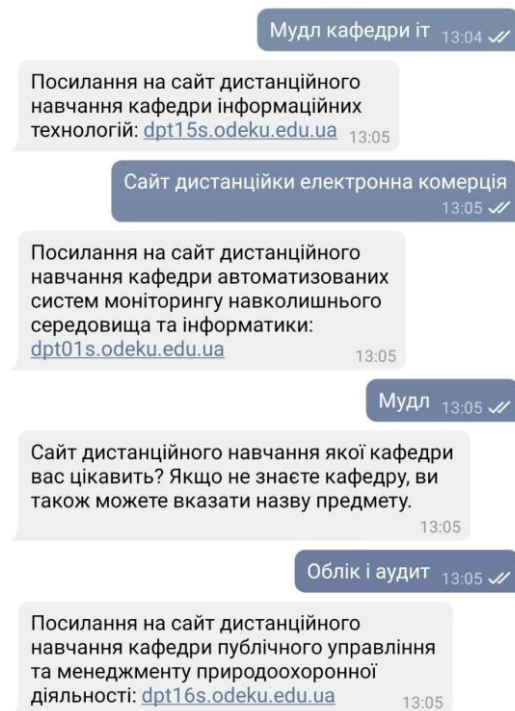


Рисунок 5.17 – Посилання на сайт дистанційного навчання кафедр

Помічник також буде корисним для абітурієнтів університету, які хочуть дізнатися про спеціальності, які є на факультеті, про вартість навчання, про місцезнаходження університету або про роботу приймальної комісії. Для цього користувач повинен задати помічнику природною мовою відповідне питання, яке його цікавить.

Наприклад, на запити «де ви знаходитесь» або «як дістатися до університету» помічника відповідь повідомленням з адресою ВНЗ, маршрутами, якими можна до нього дістатись, посиланнями на адресу університету на Google Maps та в 2Гіс, а також цікавим фактом про його місцезнаходження (рис. 5.18)

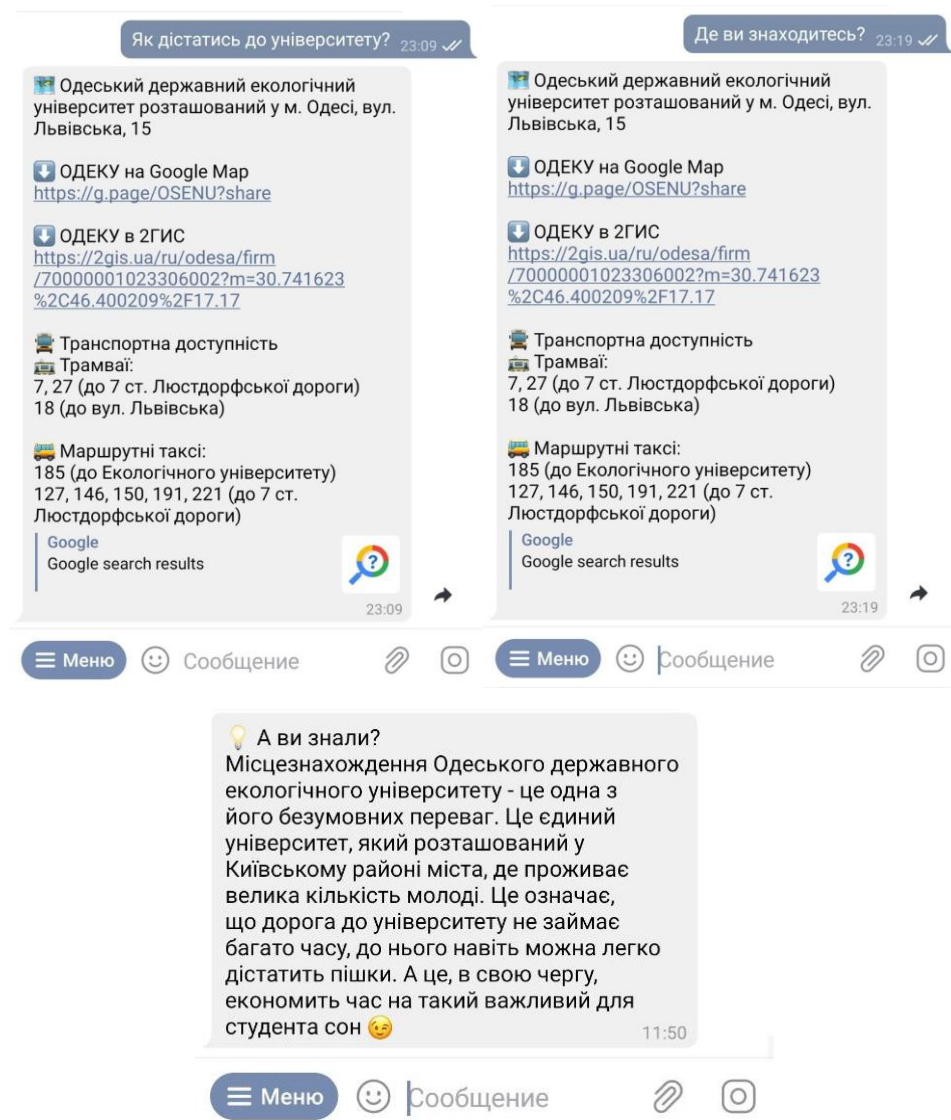


Рисунок 5.18 – Інформація про розташування ОДЕКУ

Також абітурієнти можуть дізнатися, фахівців яких спеціальностей готує факультет та про переваги кожної з них (рис. 5.19).

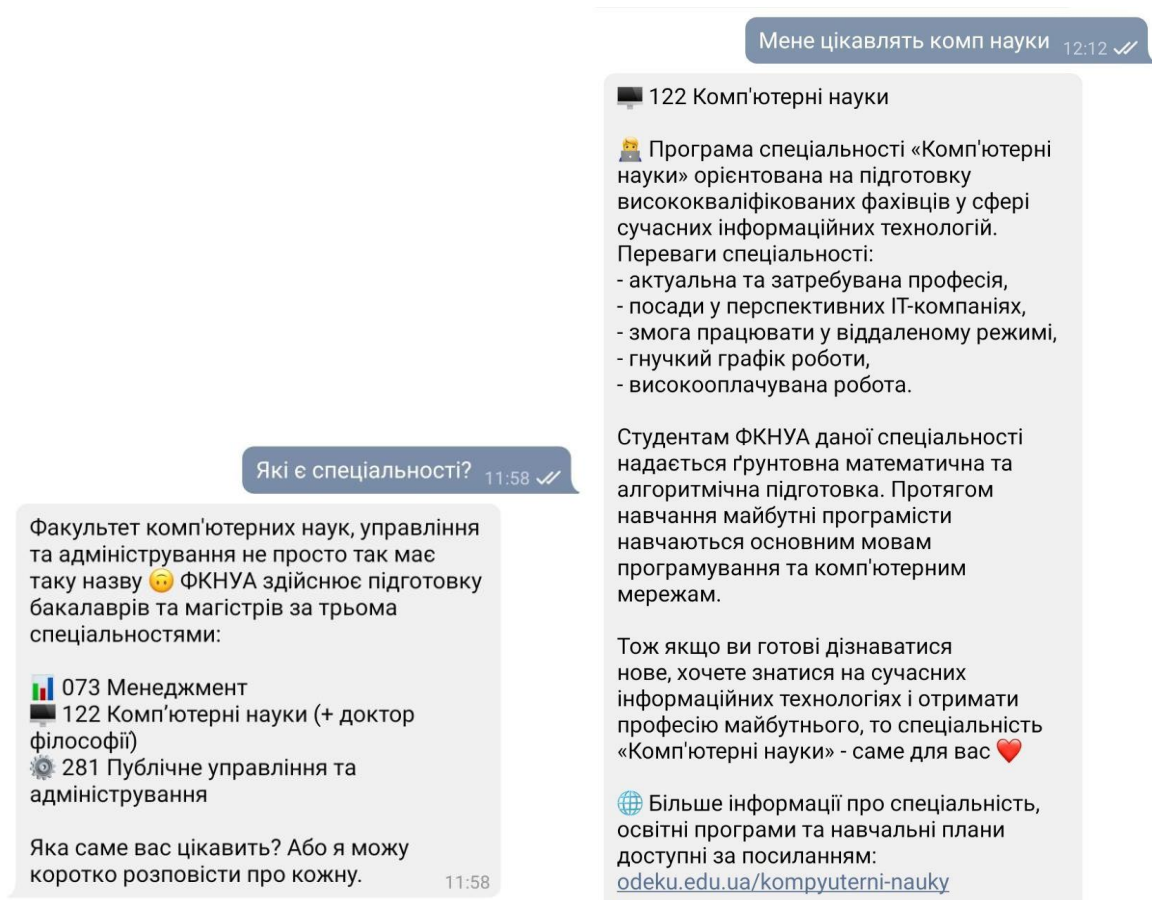


Рисунок 5.19 – Інформація про спеціальності

ВИСНОВКИ

Метою даної кваліфікаційної роботи є розробка інтерактивного цифрового помічника з елементами штучного інтелекту для більш зручного та швидкого пошуку інформації про факультет комп'ютерних наук, управління та адміністрування Одеського державного екологічного університету.

У результаті виконання кваліфікаційної роботи бакалавра був розроблений інтерактивний цифровий помічник факультету комп'ютерних наук, управління та адміністрування Одеського державного екологічного університету з використанням технології обробки природної мови (NLP). За допомогою помічника студенти та викладачі можуть отримати останні новини, посилання на сайти дистанційного навчання, розклад занять, тощо. До того, «Помічник ФКНУА ОДЕКУ» цікавий та пізнавальний для абітурієнтів, адже надає контактну інформацію факультету та кафедр, інформацію щодо вступу, вартості навчання, спеціальностей та переваг університету і факультету КНУА. В ході роботи було проаналізовано предметну область, що дало можливість визначити актуальність розробки. Проведений порівняльний аналіз існуючих помічників-аналогів показав, що університети України в більшості не використовують технологію NLP в своїх помічниках, тож не мають повноцінного аналога розробленого «Помічника ФКНУА ОДЕКУ».

Для вибору інструментів розробки були проаналізовані різні засоби та платформи і було обрано платформу Dialogflow з огляду на простоту використання, широкі можливості для створення інтерактивних помічників з обробкою природної мови в рамках безкоштовного плану, розвинуту документацію та швидкодію алгоритмів машинного навчання завдяки розробленому Google власному спеціалізованому обчислювальному обладнанню. Аналіз варіантів реалізації цифрового помічника та популярності систем миттєвого обміну повідомленнями в Україні показав, що оптимальним месенджером для інтеграції з агентом Dialogflow є Telegram.

На етапі проектування були створені UML-діаграми варіантів використання, активності і послідовності, які дозволили унаочнити логіку роботи агента.

В роботі представлені можливості помічника та приклади його використання. Було виконано тестування роботи помічника студентами різних груп факультету. Проблем при використанні виявлено не було.

Матеріали за темою даної кваліфікаційної роботи були представлені на XXI Науковій конференції молодих вчених ОДЕКУ у секції інформаційних технологій та опубліковані у збірнику тез матеріалів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Similarweb – odeku.edu.ua. [Електронний ресурс]. – Режим доступу: <https://www.similarweb.com/website/odeku.edu.ua>. – Загол. з екрана. – Дата звернення 13.11.2022
2. М34 Матеріали XXI наукової конференції молодих вчених Одеського державного екологічного університету, 23-31 травня. Одеса: ОДЕКУ. 2022. с. 191-193.
3. ANNUAL REPORT 2019. [Електронний ресурс]. – Режим доступу: <https://reports.cargemini.com/2019/en/annual-report.html>. – Загол. з екрана. – Дата звернення 05.10.2022
4. Крок Помічник. [Електронний ресурс]. – Режим доступу: <https://cabinet.krok.edu.ua:8443/Home/Krokhelper>. – Загол. з екрана. – Дата звернення 05.10.2022
5. Что лучше: приложение или мобильная версия сайта? [Електронний ресурс]. – Режим доступу: <https://appbooster.com/blog/chto-luchshie-mobilnoie-prilozhieniie-ili-mobilnaia-viersiia-saita>. – Загол. з екрана. – Дата звернення 14.10.2022
6. 10 Awesome Chatbot Benefits for Your Business [Електронний ресурс]. – Режим доступу: <https://www.revechat.com/blog/chatbot-business-benefits>. – Загол. з екрана. – Дата звернення 08.10.2022
7. Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів. О.І. Провотар, Х.А. Клочко. Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка. – 2013. – № 3. – 6 с.
8. Global Social Media Statistics. [Електронний ресурс]. – Режим доступу: <https://datareportal.com/social-media-users>. – Загол. з екрана. – Дата звернення 08.10.2022

9. Bots: An introduction for developers. [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots>. – Загол. з экрана. – Дата звернення 21.10.2022
10. Jurafsky Daniel. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. - 3rd edition. - Prentice Hall, 2019. - 621 p.
11. An In-Depth Guide to Natural Language Processing (NLP) [Электронный ресурс]. – Режим доступа: <https://datascience.eu/natural-language-processing/natural-language-processing-nlp>. – Загол. з экрана. – Дата звернення 21.10.2022
12. The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I) [Электронный ресурс]. – Режим доступа: <https://heartbeat.comet.ml/the-7-nlp-techniques-that-will-change-how-you-communicate-in-the-future-part-i-f0114b2f0497>. – Загол. з экрана. – Дата звернення 25.10.2022
13. Top 5 NLP Chatbot Platforms [Электронный ресурс]. – Режим доступа: <https://chatbotslife.com/best-nlp-chatbot-platform-d3d80f441078>. – Загол. з экрана. – Дата звернення 15.10.2022
14. Dialogflow Fulfillment [Электронный ресурс]. – Режим доступа: <https://cloud.google.com/dialogflow/es/docs/fulfillment-overview>. – Загол. з экрана. – Дата звернення 17.10.2022
15. Dialogflow Entities [Электронный ресурс]. – Режим доступа: <https://cloud.google.com/dialogflow/es/docs/entities-overview>. – Загол. з экрана. – Дата звернення 17.10.2022
16. Heroku Dev Center [Электронный ресурс]. – Режим доступа: <https://devcenter.heroku.com>. – Загол. з экрана. – Дата звернення 20.10.2022
17. Node.js Express [Электронный ресурс]. – Режим доступа: <https://nodejsdev.ru/doc/express>. – Загол. з экрана. – Дата звернення 18.10.2022

18. Firebase Realtime Database Integration [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/database>. – Загол. з екрана. – Дата звернення 06.11.2022
19. Dialogflow Telegram Integration [Електронний ресурс]. – Режим доступу: <https://cloud.google.com/dialogflow/es/docs/integrations/telegram>. – Загол. з екрана. – Дата звернення 03.11.2022
20. Одеський державний екологічний університет. Факультет комп'ютерних наук, управління та адміністрування [Електронний ресурс]. – Режим доступу: <http://fkn.odeku.edu.ua/uk>. – Загол. з екрана. – Дата звернення 10.10.2022

ДОДАТОК А

Управління проектом

Одним з етапів планування робіт є створення WBS діаграми. WBS діаграма проекту (англ. – Work Breakdown Structure – ієрархічна структура робіт) – це розбиття проекту на конкретні результати, які мають бути досягнуті для досягнення цілей проекту.

Така структурна декомпозиція досягається за допомогою послідовної декомпозиції. Інструмент спрямований на детальне планування, визначення та розподіл індивідуальної відповідальності виконавців, оцінку вартості продукту та проекту в цілому.

Основні терміни та поняття:

Work (Робота) - безперервне фізичне або розумове зусилля, спрямоване на подолання перешкод та досягнення цілей чи результатів; специфічне завдання, обов'язок, функція або завдання, які часто є частиною фази або іншою, більшою за обсягом роботи; щось, що виробляється чи виконується внаслідок зусилля чи застосування навичок (кваліфікації).

Breakdown (Декомпозиція) — поділ на частини чи категорії, виділення простих складових.

Structure (Структура) - фіксована впорядкована безліч об'єктів і відносин між ними, класифікація чогось за заданою основою.

Ці визначення означають, що ієрархічна структура робіт має такі характеристики:

- Описує з необхідною точністю зміст робіт проекту.
- Визначає весь обсяг робіт із проекту.
- Формується як ієрархічна структура
- Представляє обсяг робіт пакету як перелік робіт, які мають вимірний чи порівняний результат.

- Має об'єктивний чи вимірюваний результат, що розглядається як результат роботи чи сукупність результатів робіт.

WBS-структура процесу створення інтерактивного помічника ФКНУА ОДЕКУ наведена на рис. А.1.

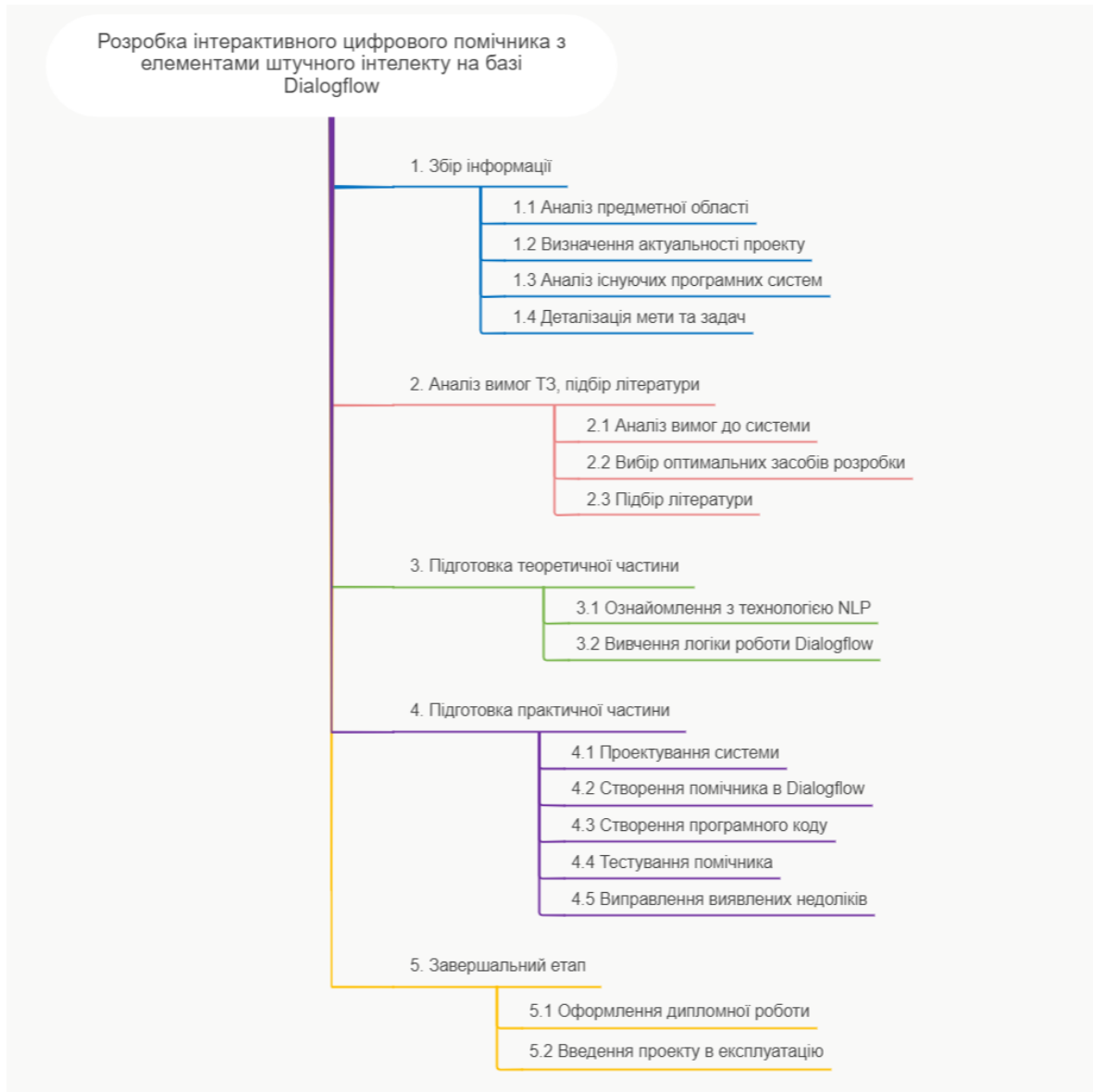


Рисунок А.1 – WBS діаграма

Діаграма Ганта – один із найпопулярніших і найкорисніших способів відображення завдань у часі, який зазвичай використовується в управлінні проектами, є. Ліворуч на діаграмі наведено список завдань, а вгорі – шкала

часу. Кожне завдання представлено смугою; положення та довжина смуги відображає дату початку, тривалість і дату завершення діяльності. Це дозволяє з першого погляду побачити:

- які є завдання;
- коли кожне завдання починається і закінчується;
- яка тривалість кожного завдання;
- де завдання перетинаються та наскільки довго;
- дата початку та закінчення всього проекту.

Діаграма Ганта виконання даної кваліфікаційної роботи створена за допомогою інструмента Microsoft Office Excel (рис. А.2).

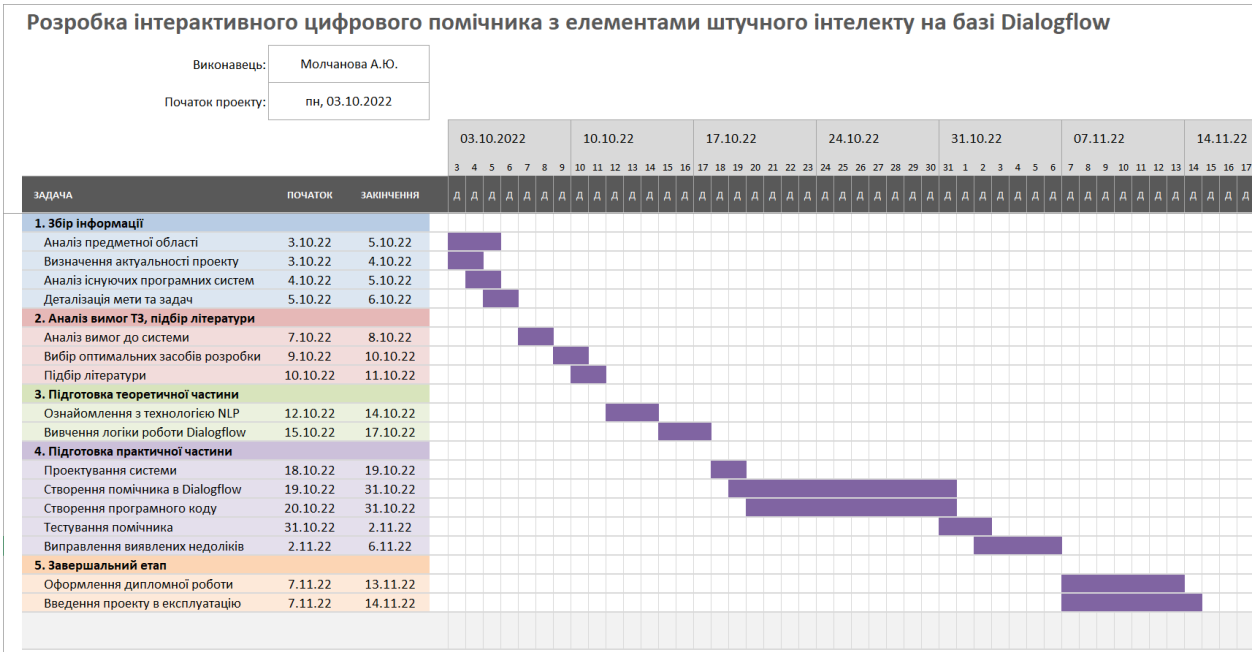


Рисунок А.2 – Діаграма Ганта