

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки
Факультет комп'ютерних наук та технологій
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему РОЗРОБКА ЧАТ БОТА З ВИКОРИСТАННЯМ СИСТЕМИ РОЗПІЗНАВАННЯ
ТЕКСТУ

Виконав: студент 4 курсу, групи КНТ-518сп
спеціальності _____

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

«Комп'ютерна інженерія»

Латаш Владислав Олександрович

(прізвище та ініціали)

Керівник Ільяшенко М.Б.

(прізвище та ініціали)

Рецензент Качан О.І.

(прізвище та ініціали)

м. Запоріжжя

2021 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти (освітній ступінь) бакалаврський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерна інженерія
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри Кудерметов Р.К.

“ ” 2021 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Латаш Владислав Олександрович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка чат бота з використанням системи розпізнавання тексту

керівник проекту (роботи) Ільяшенко Матвій Борисович, к. т. н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 17 ” березня 2021 року № 81

2. Строк подання студентом проекту (роботи) 06 травня 2021 року

3. Вихідні дані до проекту (роботи) аналіз стану досліджень з питань розробки чат-боту, постановка задачі та методи дослідження, проектування інформаційного чат-боту, розробка інформаційного чат-боту

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 аналіз стану досліджень з питань розробки чат-боту

2) постановка задачі та методи дослідження;

3) проектування інформаційного чат-боту.

4) розробка інформаційного чат-боту

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

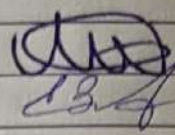
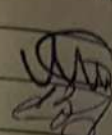
1. Структура чат-боту

2. Діаграма варіантів використання інформаційної системи

3. Алгоритм роботи чат – бота

4. Етапи розробки інформаційної системи

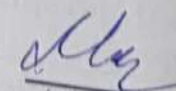
6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконав завдання
1-3 <i>Г/Колібраць</i>	Ільяшенко М.Б., к. т. н., доцент <i>Велик О.В., асист.</i>		

7. Дата видачі завдання 01.03.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Прим.
1	Дослідження ринку чат-ботів	01.03.2021 р.	
2	Мета та задачі		
3	Формування технічних вимог		
4	Аналіз документації Telegram		
5	Аналіз документації Google Search та Google Assistant	15.03.2021 р. 20.03.2021 р.	
6	Аналіз документації Dialogflow		
7	Аналіз документації text recognition	01.04.2021 р. 10.04.2021 р.	
7	Проектування роботи чат-боту	алгоритму 15.04.2021 р.	
8	Оформлення графічного матеріалу		
9	Оформлення допоміжного матеріалу	20.04.2021 р. 01.05.2021 р.	

Студент 
(підпис)
Керівник проекту (роботи) 
(підпис)

В.О. Латаш
(ініціали та прізвище)
М.Б. Ільяшенко
(ініціали та прізвище)

РЕФЕРАТ

ПЗ: 46 с., 38 рис., 6 табл., 31 джерел.

КОМУТАТОРИ, КОМП'ЮТЕРНА МЕРЕЖА, ЛОКАЛЬНА МЕРЕЖА, МАРШРУТИЗАТОРИ, СЕРВЕР, ТОПОЛОГІЯ, GIGABIT ETHERNET, INTERNET, WI-FI

Актуальність. Пошук інформації є невід'ємною частиною кожного користувача смартфона. Всі верстви населення так чи інакше звертаються до пошукових систем. Використання сучасних месенджерів набуває великого попиту. Одним із таких месенджерів є Telegram. Інформаційна система Telegram є основоположником такої програми, як чат-бот. Тому була поставлена актуальна задача по розробці пошукової системи в месенджері Telegram, інакше кажучи створення комп'ютерної програми чат-бот. Дана програма надаватиме широку можливість для тих, на чийх девайсах не існує голосових помічників, або оптимізувати пошук до однієї цілої системи за допомогою сервісів компанії Google.

Об'єкт дослідження. Чат-бот, як пошукова система.

Предмет дослідження. Алгоритм пошуку інформації та спілкування в Telegram.

Гіпотеза розробки. Якщо використати технологію сумісного використання сервісів Google, то можна розробити алгоритми для спілкування та пошуку без використання пошукової системи в смартфоні.

Мета. Розробка інформаційного чат-боту в месенджері Telegram.

ЗМІСТ

Вступ.....	6
1 Аналіз стану досліджень з питань розробки чат-боту.....	8
1.1 Дослідження ринку чат-ботів.....	8
1.2 Аналіз аналогів та визначення проблеми	10
2 Постановка задачі та методи дослідження.....	14
2.1 Мета та задачі	14
2.2 Методи реалізації.....	16
3 Проектування інформаційного чат-боту	20
3.1 Вимоги до інформаційної системи	20
3.2 Структура інформаційної системи.....	21
3.3 Моделювання інформаційної системи.....	22
3.4 Моделювання варіантів використання чат-боту	26
4 Розробка інформаційного чат-боту.....	28
4.1 Розробка логічної структури чат-боту	28
4.2 Опис функціонування чат-боту.....	33
Висновки.....	43
Перелік джерел посилання	44
Додаток А.....	46
Додаток Б	63
Додаток В.....	64

ПЕРЕЛІК ГРАФІЧНОГО МАТЕРІАЛУ:

Пл1 – Структура чат-боту

Пл2 – Діаграма варіантів використання інформаційної системи

Пл3 – Алгоритм роботи чат – бота

Пл4 – Етапи розробки інформаційної системи

ВСТУП

На сьогоднішній день чат-боти як сучасні інструменти комунікацій стали широко використовуватися в багатьох сферах життєдіяльності людини.

З метою встановлення контакту з користувачами мережі Інтернет. Найбільшу популярність чат-боти отримали, коли почалося їх використання в месенджерах і соціальних мережах (наприклад, в Telegram, Viber, Facebook,).

Завдяки новому інструменту віртуальної комунікації з'явилася можливість дізнаватися про спеціальні пропозиції онлайн, отримувати посилання свіжих новин та спеціальних пропозицій в сфері товарів і послуг, здійснювати більш складні операції. В даний час найбільше значення чат-боти мають у формуванні такої моделі поведінки, яка буде максимально наближена до людської.

Під чат-ботами в загальному сенсі зазвичай розуміють спеціальні програми, які здійснюють інтернет-спілкування найчастіше з одним або декількома користувачам, використовуючи штучний інтелект. Сучасні вигди інтернет-комунікацій здатні виступати в якості віртуального співрозмовника повторювати і відтворювати письмовий набір знаків людини, надаючи запрограмовану відповідь на задані питання. В силу того, що контакт з чат-ботами є не тільки цікавим заняттям, але і корисним для людини з точки зору отримання нової інформації і рішення споживчих завдань, варто відзначити, що інтернет-аудиторія прийняла їх появу доброзичливо.

Багато організацій в свою чергу побачили в інтересі аудиторії до чат-ботів можливості для власного просування і стали розробляти власні чат-боти, використовуючи їх для залучення уваги споживачів до компанії за рахунок використання сучасного інструменту комунікації. Чат-бот стає каналом просування або частиною рекламної кампанії як для великих брендів, так і для новачків. Потрібно позначити таку тенденцію: спочатку деякі компанії почали використовувати їх тільки як визначення трендів, тобто стала формуватися певна мода на використання чат-ботів без урахування їх практичної значущості.

Можливості інтерфейса програми були досить зрозумілі і прості. На основі заздалегідь завантажених шаблонів, запам'ятовуванні слів, введених самим користувачем, і відповіді на відповідний запит був заснований алгоритм роботи чат-бота.

Чат-боти можуть використовуватися для спілкування між користувачами, а також в розважальних, інформаційних цілях (чат-боти можуть повідомити погоду, курс валют, записати клієнта на прийом, здійснити реєстрацію дзвінків) і в службах підтримки.

Поняття «чат-бот» походить від двох англійських слів: tochat - невимушена розмова в мережі Інтернет, bot (robot) - скорочено робот, з чого випливає, що це роботи, призначені для здійснення комунікацій з користувачами в мережі Інтернет, які виконують дії відповідно до закладеному сценарію. Така програма інтернет-комунікації як чат-бот заснована на сучасних технологіях. В її основі лежать такі інформаційно-комунікаційні технології, як штучний інтелект, нейронні мережі, що дозволяє наближати комунікації чат-ботів до людського віртуального спілкування і вирішувати з їх допомогою комерційні та маркетингові задачі [1] .

1 АНАЛІЗ СТАНУ ДОСЛІДЖЕНЬ З ПИТАНЬ РОЗРОБКИ ЧАТ-БОТУ

1.1 Дослідження ринку чат-ботів

Виходячи з поточної оцінки стану ринку, чат-боти, особливо для месенджерів – дуже перспективний напрям, який в даний час переживає бурхливе зростання (рис. 1.1).

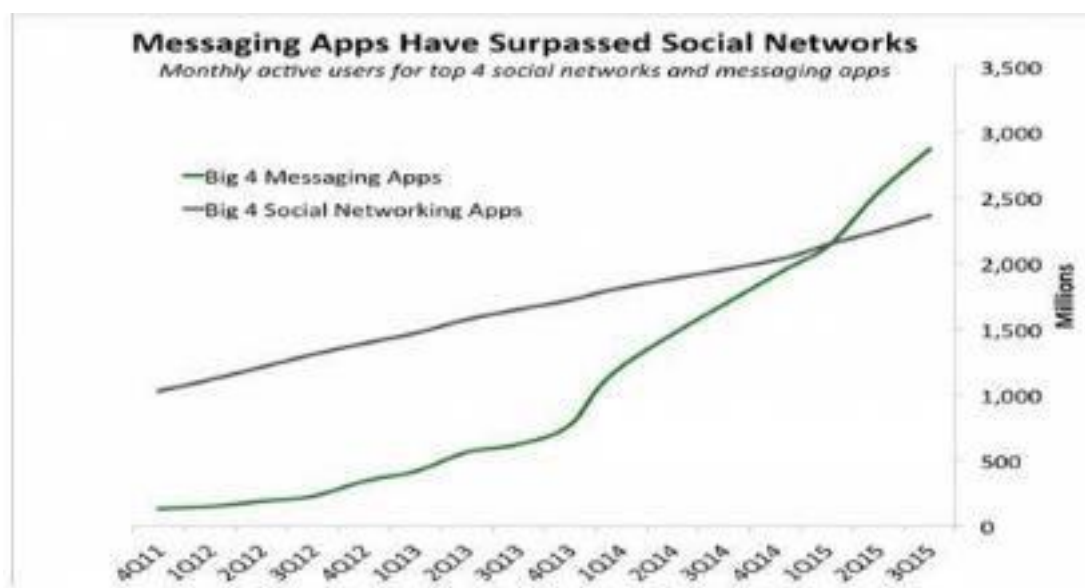


Рисунок 1.1 – Тенденції ринку чат-ботів

За оцінкою аналітиків у найближчому майбутньому чат - боти будуть мати все більшу значимість. Вони можуть повністю замінити класичні пошукові движки і соціальні мережі [2]. Переваги ботів забезпечують зручність людино-комп'ютерної взаємодії, швидкість реакції, можливість їх налаштування під користувача. Прикладом може бути Facebook чат-бот M, який в поточний момент знаходиться в режимі бета-тестування – це персональний цифровий асистент всередині месенджера, що виконує прості задачі (в т.ч. рекомендації з надання послуг). Технічно Facebook-M оснований на сучасних технологіях машинного навчання (глибинні нейромережі). Крім Facebook, в напрямку розвитку технологій

чат-ботів активно рухається Google і Apple [3].

Основні напрямки використання чат-ботів:

— у сфері бізнесу та електронної комерції. У даних сферах чат-боти мають найбільший функціонал і можуть вирішувати багато завдань, включаючи онлайн-консультування, рекламні комунікації, продажу та ін. [4];

— у сфері отримання інформації. Такі чат-боти використовуються в електронних засобах масової інформації для надання користувачам цікавої їм новинної інформації. Наприклад, доктор-бот може порадити лікування на основі озвучених симптомів або підказати адреси найближчих лікарень [5];

— у сфері навчання іноземним мовам та інших дисциплін, так звані бот-вчителі. Вони надають навчальний матеріал (найчастіше це невеликі за обсягом тексти) з дисципліни, оцінюють знання учнів. В даний час чат-боти допомагають самостійно вивчати іноземні мови [6];

— у сфері розваг. Чат-боти в сфері розваги можуть вести діалог з користувачем, надавати йому цікаву інформацію, грати з ним і виконувати інші розважальні функції [7].

Таким чином, чат-боти активно використовуються в бізнесі, освіті, ЗМІ та сфері розваг, що характеризує їх як досить універсальний інструмент комунікацій. Чат-боти в сфері бізнесу можуть виконувати різні функції, пов'язані із здійсненням внутрішніх та зовнішніх комунікацій в організації: автоматична розсилка документів, повідомлень, призначення зустрічей, обслуговування та консультування клієнтів, з'ясування потреб клієнтів, уявлення відповідей на прості питання, пошук про товар для економії часу і швидкого прийняття рішення про покупку. Чат-боти можуть аналізувати і збирати всю отриману під час комунікацій з користувачами інформацію для подальшого використання з метою поліпшення роботи чат-бота, так як у досліджуваного механізму є функція запам'ятовування отриманої інформації. Тобто чат-боти можуть автоматично збирати дані про клієнтів і при повторному зверненні можуть відтворювати власну базу, щоб не ставити клієнту питання раніше чим надати інформацію про запитувані товари або послугу [8].

Даний дипломний проект буде корисним та актуальним у побутовому використанні. Адже використання боту в інформаційній системі Telegram звужує ресурс використання інших додатків, котрі можуть бути зайвими. В країні у кожного користувача є девайс, але більша частина смартфонів не в змозі підтримувати голосових асистентів. Тому була поставлена задача автоматизувати роботу пошуку в мережі Інтернет. Він є ефективним та зручним у використанні. Також завдяки зручним надбудовам в месенджері бот має змогу працювати при низькому трафіку, а також підтримувати спілкування з користувачем.

1.2 Аналіз аналогів та визначення проблеми

Перед тим як почати розробляти свій продукт, необхідно було ознайомитись з ринком чат-ботів, адже бот, котрий має будь-який інформаційний напрям або, інакше кажучи, якщо є зворотня відповідь, то він відразу вважається інформаційним. Важливо зазначити, що аналогів серед багатьох телеграм ботів знайдено не було. Навіть якщо вони існують, то безпосередньо це комерційні проекти, котрі зберігаються лише конфіденційно. Оцінювати його серед інших ботів теж неможливо, адже кожний бот виконує певну задачу і працює безпосередньо в своїй предметній області. Оскільки майже всі боти по своїй структурі інтерфейсу є прості та зрозумілі, тому наявність інструкційне є необхідною. Зараз будуть наведені аналоги інформаційних чат-ботів.

Кіноман 2.0 (@kinoman) шукає фільми і серіали, радить рандомні картини для перегляду, стежить за ситуацією на торентах. (рис 1.2) .

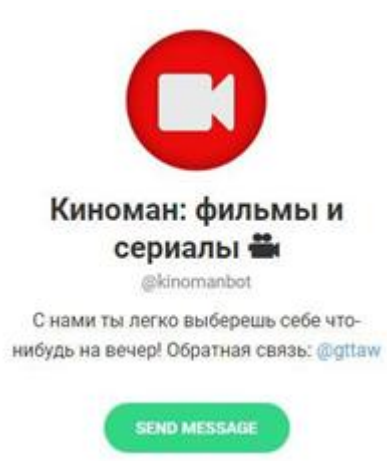


Рисунок 1.2 – Бот Кіноман 2.0

MyCookBot (@mycookbot) на жаль, цей бот не вміє доводити продукти до готового стану за допомогою нейромереж, але він підказує рецепти і вибирає страви з ваших інгредієнтів. Бот зображений на рисунку 1.3.



Рисунок 1.3 – Бот MyCookBot

SoundCloud (@scloud_bot) бот, що імітує клієнт, музичного сервісу SoundCloud прямо в діалоговому вікні. Все по-дорослому: скачування треків, чарти по жанрам, підключення акаунта (рис 1.4).

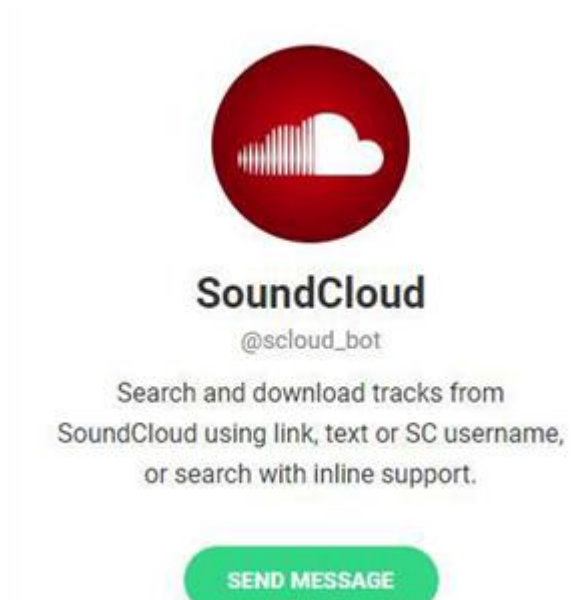


Рисунок 1.4 – Бот SoundCloud

Wolfram Alpha (@WolframBot) вирішує прості математичні задачки. Більше не треба завантажувати дурні математичні калькулятори. Він навіть графіки побудує і відправить. Розпізнає також і інші запити. Бот зображений на рис.1.5.



Рисунок 1.5 – Бот Wolfram Alpha

Дані боти є у відкритому доступі для завантаження на смартфон. Вони є

досить корисними у сфері свого використання. Звісно, їх можна назвати інформаційними, адже окрім того, що вони ведуть діалог, вони повертають результат: чи то рецепт, музичний файл або відповідь математичної задачі. У мережі є ще дуже багато різноманітних ботів, і всі вони за своїм змістом унікальні [9].

Тому було обране рішення розробити саме інформаційний бот, котрий позиціонує себе як пошукова система, яка може спілкуватися зі своїм користувачем. Дана пошукова система є дипломним проектом і називається assistant. Вона зображена на рис.1.6.

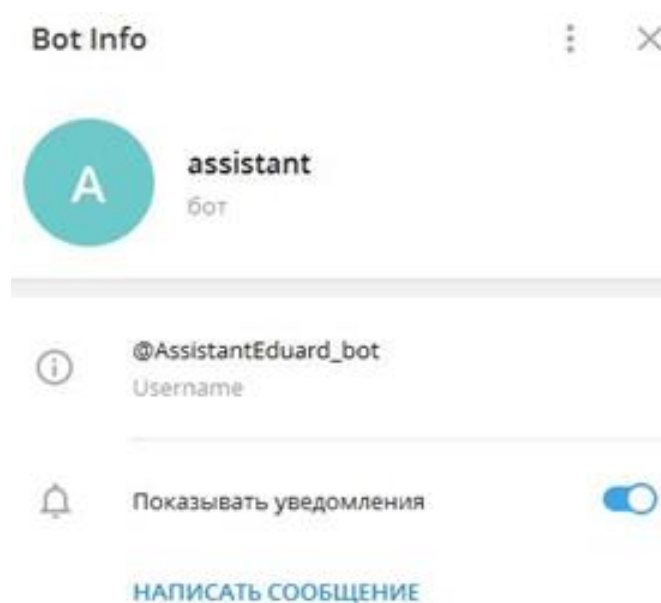


Рисунок 1.6 – Бот assistant

І для того, щоб зрозуміти для якої верстви населення бот буде актуальним та цікавим, необхідно його віднести до певної категорії. Подібні програми мають різні функціонали, тому сфера використання віртуальних помічників постійно підвищується. Це можуть бути різні сфери, такі як, доставка їжі, пошуковий сервіс, портал новин, науковий форум та інші.

За цільовим призначенням можна виділити чотири основні типи роботів. Вони показані в табл. 1.1.

Таблиця 1.1 – Відношення до типу чат – бота

Види чат - ботів	Категорії			
	Новини	Квитки	Їжа	Пошуковий
Розважальні	-	-	-	+
Консультанти	-	-	-	-
Помічники	-	-	-	+
CRM-система	-	-	-	-

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Головна мета – розробити інформаційний чат-бот в соціальній мережі Telegram. Дана соціальна мережа є першою, яка запровадила розроблення чат-ботів у месенджері.

Даний бот має можливість виконувати такі функції:

- симуляція живого спілкування з користувачем;
- пошук інформації за допомогою тексту;
- пошук інформації за допомогою голосового повідомлення.

Даний бот може зацікавити не тільки з точки зору реалізації, а й з точки зору подальших вдосконалень. Даний бот має необмежені можливості вдосконалення, наприклад:

- додати нові мови для розпізнавання;
- додати відображення гарячих новин;
- вдосконалити команди для розпізнавання
- додати персональні моделі поведінки боту (наприклад, відображення ринку криптовалют);
- додавати нові продукти компанії Google та інших компаній та сервісів.

Даний бот повинен мати зручний інструмент пошуку та відображення необхідної інформації.

Виходячи з аналізу предметної області, а також аналогів даної

інформаційної системи, можна представити даний бот як інформаційна система в інформаційній системі. Існуючі аналоги дали зрозуміти, що дана розробка буде актуальною та необхідною у ряді повсякденного використання. Вище зазначені пункти дали зрозуміти, що в Telegram можна знайти бота будь-якого характеру, але не тільки інформаційного з точки зору пошуку та збору інформації.

Telegram Messenger - це програма для обміну повідомленнями, яка працює через Інтернет так само, як WhatsApp та Facebook Messenger. Це означає, що людина може безкоштовно відправляти повідомлення за допомогою з'єднання Wi-Fi або мобільної передачі даних (якщо даних достатньо).

Додаток прив'язується до номера телефону, потім приходить перевіряюче повідомлення з кодом (на мобільній версії для Android код підтягується автоматично). Спілкуватися можна як з конкретними користувачами, які вже встановили додаток, так і організовувати групові чати до 200 чоловік і відправляти запрошення, повідомлення з закликом. Ще мить - двісті чоловік в чаті. Розробники вважають це однією з найголовніших фішок свого застосування.

Завдяки захищеності месенджера і захисту від перехоплення додаток став популярним в Гонконзі під час протестів. Спочатку про мітинг домовлялися через Facebook, але цю інформацію легко дістати.

Основним пунктом продажу телеграми є безпека. Це означає, що всі дії, включаючи чати, групи та медіа-зашифровані, не будуть видимі, не будучи розшифрованими першими, навіть якщо вони перехоплені.

Telegram може використовуватися на смартфонах, планшетах, ноутбуках і настільних комп'ютерах. Програма Telegram доступна для Android, iOS, Windows Phone, WindowsNT, MacOS і Linux.

2.2 Методи реалізації

Для реалізації необхідно визначитись з мовою програмування, адже не запрограмувати бота такого рівня неможливо. Були проаналізовані мови програмування і обрано мову Python.

Python - це мова програмування загального призначення, націлена в першу чергу на підвищення продуктивності самого програміста, ніж коду, який він пише. Говорячи простою людською мовою, на Python можна написати практично що завгодно (веб, настільні додатки, ігри, скрипти по автоматизації, комплексні системи розрахунку, системи управління життєзабезпеченням, чат – боти і багато іншого) без значних проблем [10].

Більш того, поріг входження низький, а код багато в чому лаконічний і зрозумілий навіть тому, хто ніколи його не використовував. За рахунок простоти коду подальший супровід програм, написаних на Python, стає легше і приємніше в порівнянні з Java або C ++. З точки зору бізнесу перевага полягає у скороченні витрат і збільшенні продуктивності праці співробітників [11].

Крім сотні тисяч індивідуальних розробників і невеликих компаній, Python підтримують такі гіганти ІТ як:

- Google;
- Dropbox;
- Mozilla;
- Facebook;
- RedHat;
- Telegram.

І саме головне на даній мові програмування компанія Google та месенджері Telegram надають зручні API на мові Python. Написання боту здійснювалось в середовищі програмування JetBrains PyCharm Community Edition 2018. Середовище зображено на рис. 2.3.

Для побудови будуть використовуватись продукти компанії Google, а саме:

Google Assistant, Dialogflow, Google Speech-to-Text, Google Search.

Асистент є голосовим помічником Google. Спочатку це було розширення Google Now, яке було особистим-у той час як розширення на існуючих Google "OK Google" вже мала можливість голосового управління. Спочатку Google Now вміло витягувала релевантну інформацію: вона знала, де людина працює, її зустрічі і плани подорожей, спортивні команди, які їй сподобалися, і те, що її цікавило. Він міг представити особисту інформацію, яка мала значення. Google вже давно може замінити Google Now, але помічник чудово працює в одному просторі, поєднуючи ці персоналізовані елементи з широким спектром голосового управління.



Рисунок 2.3 – JetBrains PyCharm Community Edition 2018

Google Assistant підтримує як текстовий, так і голосовий запис, і з задоволенням слідує за розмовою, незважаючи на метод запису [12]. Сторінка "OK Google" охоплює голосові команди, голосовий пошук і керування пристроєм, що активується голосом, дозволяючи нагадувати, надсилати повідомлення, перевіряти зустрічі і інше на пристрої Android, як і Siri на Apple iPhone або iPad.

Google Assistant надає такі можливості як:

- керувати своїми пристроями та розумним домом;
- доступ до інформації з календарів та іншої особистої інформації;
- знайти інформацію в Інтернеті: від бронювання ресторанів до напрямків, погоди та новин;
- керувати музикою;
- відтворювати вміст Chromecast або інших сумісних пристроїв;
- запускати таймери та нагадування;
- призначення та відправлення повідомлень;
- відкриття програми на телефоні;
- читати свої сповіщення;
- переклади в режимі реального часу.

Продовження розмови означає, що людині не доведеться говорити "Неу Google" для подальших запитів. Google також може розпізнавати голосові профілі для різних людей, тому він знає, хто з ним розмовляє, і може адаптувати відповіді, що інші системи також починають пропонувати [13].

Dialogflow – це комплекс розгортання для створення розмовних інтерфейсів для веб-сайтів, мобільних додатків, популярних платформ обміну повідомленнями та пристроїв IoT. Його можна використовувати для створення інтерфейсів (наприклад, чат-ботів і розмовних IVR), які дозволяють здійснювати природні та багаті взаємодії між користувачами та бізнесом. Користувачі Dialogflow Enterprise Edition мають доступ до Google Cloud Support і угоди про рівень обслуговування (SLA) для розгортання продукції [14].

Працює за допомогою машинного навчання Google. Звичайне розуміння мови розпізнає намір користувача і витягує попередньо побудовані об'єкти такі, як час, дата і цифри. Людина може навчити свого агента ідентифікувати типи спеціальних об'єктів, надавши невеликий набір прикладів.

Можна також використовувати готові агенти як шаблони. Розгортати на різних платформах і мовах. Dialogflow підтримує 20 мов і інтеграцію з одним натисканням на 14 різних платформ. Залучивши більше аудиторій, будуючи

багатомовним агентом з глобальним доступом.

Застосування найпередовіших алгоритмів нейронної мережі та глибокого навчання до звуку для розпізнавання мови з безпрецедентною точністю. Хмарна точність від слова до тексту з часом поліпшується, оскільки Google покращує внутрішню технологію розпізнавання мови, яка використовується продуктами Google. Автоматично ідентифікує розмовну мову.

Використовуючи CloudSpeech-to-Textrecognition, можна визначити, якою мовою говорять у висловленні (обмеження до чотирьох мов). Це можна використовувати для голосового пошуку (наприклад, «Яка температура в Парижі?»). Повертає текстову транскрипцію в режимі реального часу для короткого або довгого звуку.

Cloud Speech-to-Text recognition може надсилати текстові результати, відразу ж повертаючи текст, оскільки він розпізнається з потокового аудіо або як говорить користувач. Крім того, CloudSpeech-to-Textrecognition може повертати розпізнаний текст з аудіофайлу, що зберігається у файлі. Він здатний аналізувати короткі і довгі аудіо.

Автоматично транскрибує власні іменники і контекстне форматування. Cloud Speech-to-text recognition, призначений для ефективної роботи з реальною мовою, може точно транскрибувати власні іменники і належним чином формувати мову (наприклад, дати, номери телефонів). Google підтримує більш 10 власних іменників у порівнянні з кількістю слів у всьому Оксфордському словнику англійської мови [15].

Google Search, який також називається Google Web Search або просто Google – це веб-пошукова система, розроблена компанією Google LLC . Порядок результатів пошуку, повернутих Google, базується, зокрема, на системі рангу пріоритетів, яка називається " PageRank". Пошук Google також надає різні варіанти налаштування пошуку, використовуючи символи, які включають, виключають, визначають та вимагають певної поведінки пошуку, а також надають спеціалізований інтерактивний досвід, наприклад, статус рейсу та відстеження пакетів, прогноз погоди, конверсії валюти, одиниці та час визначення та багато

іншого.

Google індексує сотні терабайт інформації з веб-сторінок. Для недоступних веб-сайтів Google надає посилання на кешовані версії сайту, сформовані за допомогою останньої індексації цієї сторінки пошуковою системою. Користувачі також можуть активувати "Безпечний пошук" - технологія фільтрації, спрямована на запобігання появі явних і порнографічних даних у результатах пошуку [16].

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЧАТ-БОТУ

3.1 Вимоги до інформаційної системи

Інформаційна система, яка має вигляд чат-боту, повинна робити пошук різноманітної інформації у месенджері Telegram. Бот повинен мати зручний інтерфейс без зайвих надбудов. Користувач повинен одразу зрозуміти, яким чином виконувати пошук. Дана система оснащена новітніми сервісами компанії Google, що надає можливість розробнику зробити якісний, зручний, мобільний, актуальний чат-бот. З аналізу задач, огляду методів та програмного забезпечення можна сформулювати вимоги до інформаційної системи.

Система повинна:

- працювати на всіх версіях месенджера Telegram;
- робити голосові запити щодо пошуку;
- робити текстові запити щодо пошуку;
- вести діалог з системою;
- підтримувати 2 мови для запитів пошуку.

3.2 Структура інформаційної системи

Для розуміння роботи чат-бота та її складових побудована її структура роботи за допомогою абстрактної моделі (рис. 3.1). Система включає в себе сервіси, API Google та месенджер Телеграм, котрий завантажений на девайсі користувача.

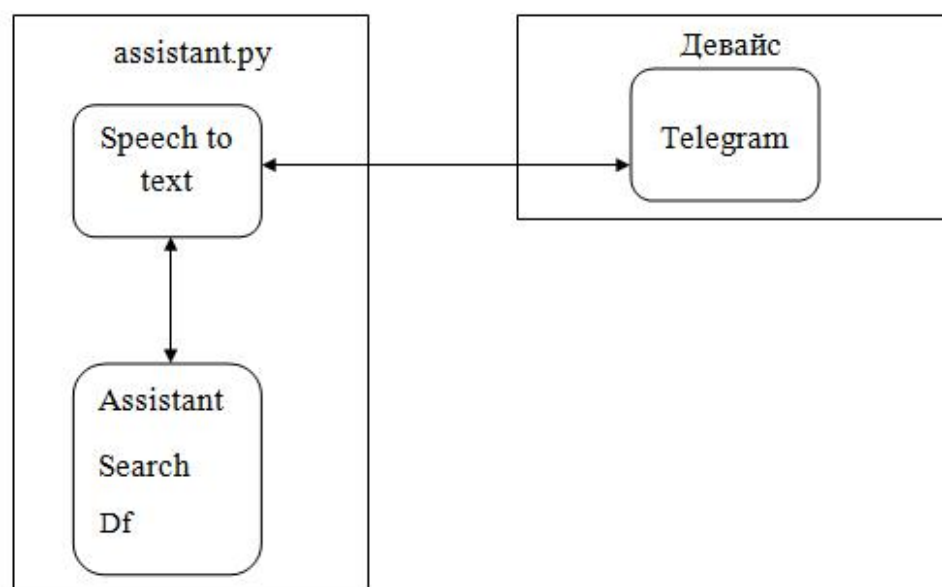


Рисунок 3.1 – Структура чат-боту

В даній структурі чітко можна побачити принцип роботи самого чат-боту. В розділах вище описані всі сервіси Google, які використовуються в даній інформаційній системі. В даній структурі зображені блоки з їх логічними структурами та додатками.

Додаток Telegram відіграє функцію платформи для розробки боту, а також його розміщення. assistant.py - файл з логікою роботи більшості сервісів, котрі є у самому боті. Speech-to-text recognition відіграє роль синтезу мови, котра після обробки аудіо чи тексту робить запит на сервіси Assistant, Search, Dialogflow. Після обробки запиту сервіси надсилають результат вже на девайс, а сам результат

буде знаходитись у самому боті, де і був проведений запит. Пошук проводиться у вигляді спілкування.[17]

3.3 Моделювання інформаційної системи

Перед тим, як розробити інформаційну систему, необхідно розробити модель її роботи, продемонструвати усі процеси, послідовність їх виконання та можливі варіанти розвитку подій всередині самої системи. Згідно з цим необхідно побудувати контекстну діаграму IDEF0 та її декомпозицію, а також варіанти використання (usecase).

В першу чергу необхідно побудувати контекстну діаграму IDEF0(рис. 3.2). Головним процесом є створення чат-боту з використанням сервісів Google. На вході маємо API key для доступу до сервісів Google, а також технічне завдання для розробки чат-боту. Відповідно на виході отримуємо вже готову інформаційну систему. Механізмами виступають Speech-to-text-recognition, з якого можна перетворити голос в текст, розробник, який розробляє саму систему, технічне забезпечення, мова програмування python, а також API Google. Усі процеси системи регулюються документацією сервісів Google, які будуть використані при розробці, а також документацією Telegram.

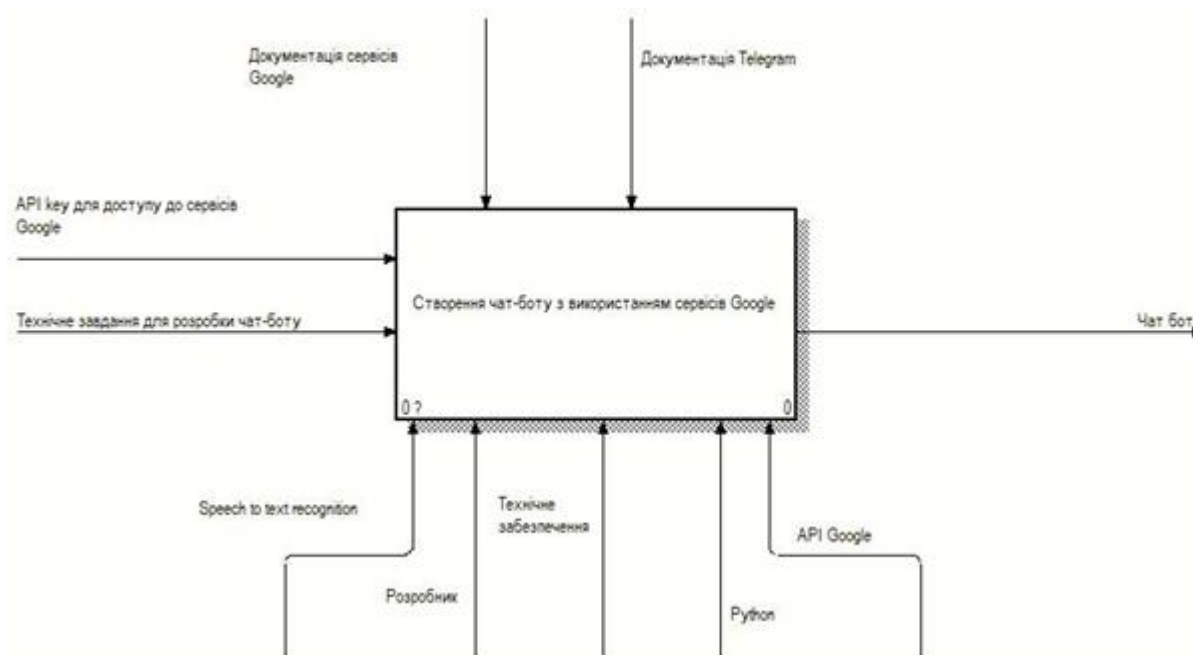


Рисунок 3.2 – Діаграма IDEF0

Контекстна діаграма дає загальний опис системи. Для того щоб побачити повний опис системи, необхідно виконати декомпозицію. Наступна дія дозволить детально ознайомитись з логікою послідовності виконання робіт для реалізації вихідного продукту.

Діаграма А-0 була розбита на три підрівні: отримання доступу до Telegram, підключення сервісів Google, створення алгоритму. Це обумовлено тим, що процес створення функціонального веб-ресурсу поділяється на два етапи:

- перший етап: надання доступу до месенджера для здійснення маніпуляцій;
- другий етап: з'єднання з кожним сервісом окремо;
- третій етап: створення логічної структури роботи чат-боту.

Опис першого етапу формування даних:

- вхідні дані: технічне завдання для розробки чат – боту;
- вихідні дані: отримання зв'язку з Telegram;
- управління: документація Telegram, документація сервісів Google;
- механізми: розробник, технічне забезпечення, python.

Опис другого етапу формування даних:

- вхідні дані: отримання зв'язку з Telegram;
- вихідні дані: отримання зв'язку з Google;
- управління: документація сервісів Google;
- механізми: розробник, python, speech to text recognition.

Опис третього етапу формування даних:

- вхідні дані: отримання зв'язку з Google, API key для доступу до сервісів Google;
- вихідні дані: чат –бот;
- управління: документація Telegram;
- механізми: розробник, python, API Google.

Діаграма другого рівня представлена на (рис.3.3).

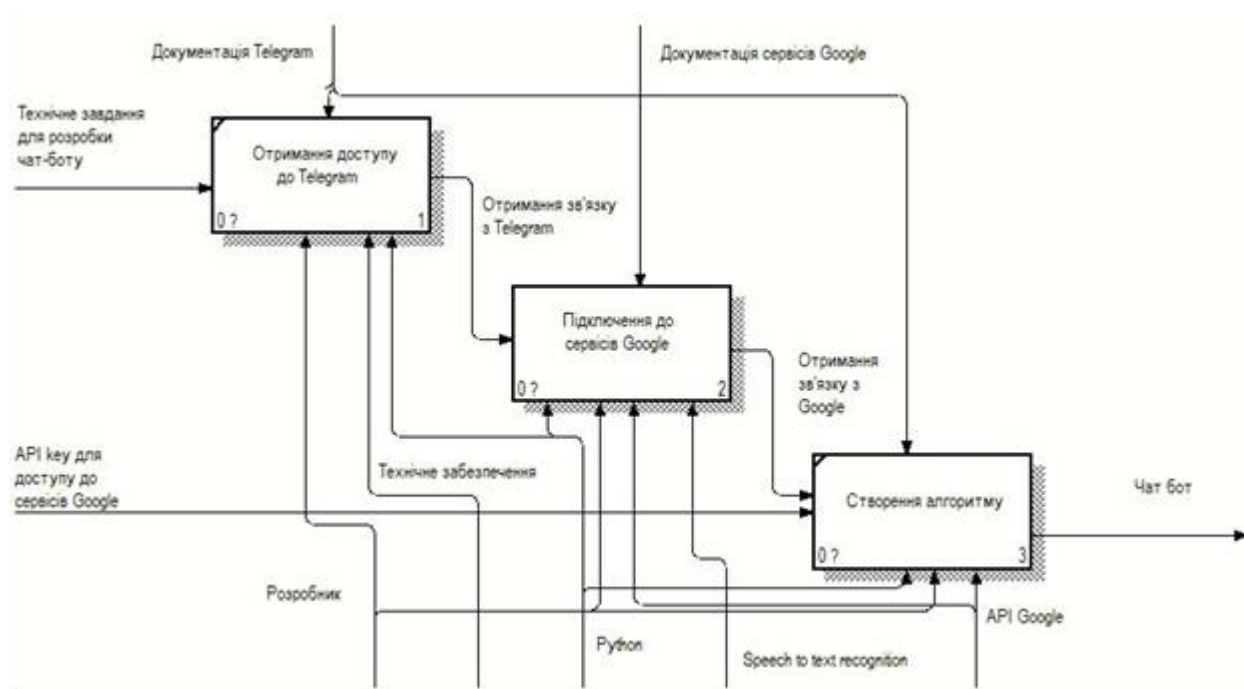


Рисунок 3.3 – Діаграма другого рівня

Після декомпозиції контекстної діаграми на три процеси було виконано розбиття кожного з них на певний перелік робіт, результатом кожної з яких є вхідними даними для наступної.[18]

Процес підключення до сервісів містить в собі такі етапи: підключення до Google Assistant, підключення до Google Search, отримання готової відповіді на запит. Це зумовлено тим, що перед тим, як підключитись до Google, необхідно зробити ряд заходів за допомогою мови програмування.

Для процесу підключення до сервісів Google Assistant були визначені такі дані:

- вхідні дані: отримання зв'язку з Telegram;
- вихідні дані: результат пошуку інформації;
- управління: розробник, python, API Google;
- механізми: документація сервісів Google.

Оскільки в даній системі використовується Google Assistant не масштабною версії, то необхідно додати пошукову систему Google Search для того, щоб бот працював безвідмовно. Були визначені такі дані:

- вхідні дані: результат пошуку інформації;
- вихідні дані: масштабований пошук;
- управління: розробник, python, speech-to-text-recognition;
- механізми: документація сервісів Google.

Останнім процесом є отримання відповіді на запит. Інакше кажучи, готова інформаційна система, яка включає в себе такі дані:

- вхідні дані: масштабований пошук;
- вихідні дані: отримання зв'язку Google;
- управління: API Google, speech-to-text-recognition;
- механізми: документація сервісів Google.

Діаграма декомпозиції процесу підключення до сервісів Google зображена на (рис.3.4).

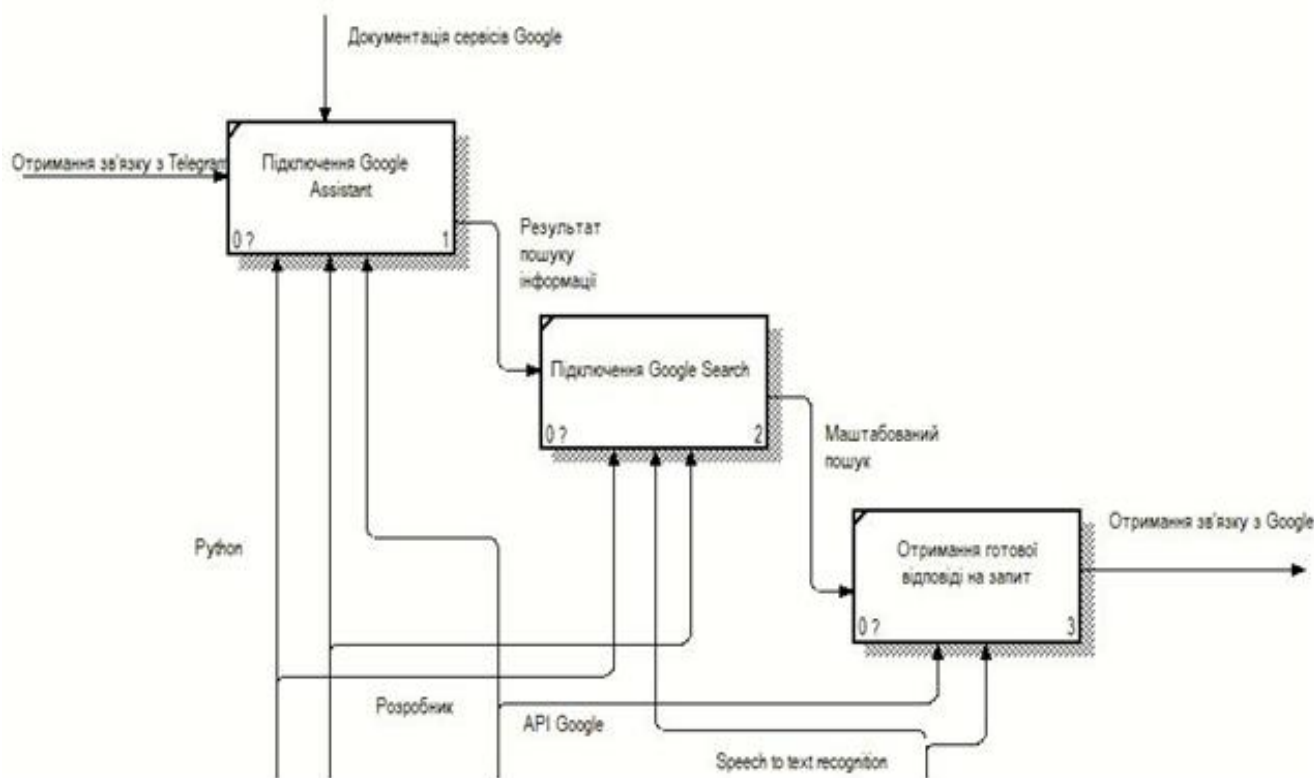


Рисунок 3.4 – Діаграма декомпозиції другого етапу

3.4 Моделювання варіантів використання чат-боту

Конструкції UML створюються з багатьох модельних елементів, які позначають різні частини системи програмного забезпечення. Елементи UML використовуються для побудови діаграм, які відповідають певній частині системи або точці зору на систему.

Для розробки діаграми Use Case були визначені такі актори:

- Admin-розробник, який повністю може маніпулювати чат-ботом, забезпечуючи технічну модернізацію системи;
- User-користувач, який може користуватись лише тими функціями бота, які в нього вбудовані.

Після того, як були визначені всі актори, які будуть взаємодіяти з системою, необхідно сформулювати перелік варіантів використання. Вони відповідають

вимогам, яким повинен відповідати чат-бот, що зазначені у Додатку А.

Варіанти використання для чат-боту:

- Telegram;
- текстовий пошук;
- голосовий пошук;
- додавання надбудов;
- зміна правил;
- зміна конфігурацій;
- Dialogflow.

Виходячи з сформованих даних про акторів та всіх можливих варіантів використання чат-боту, розроблена Use Case діаграма. Вона представлена на рисунку 3.5.

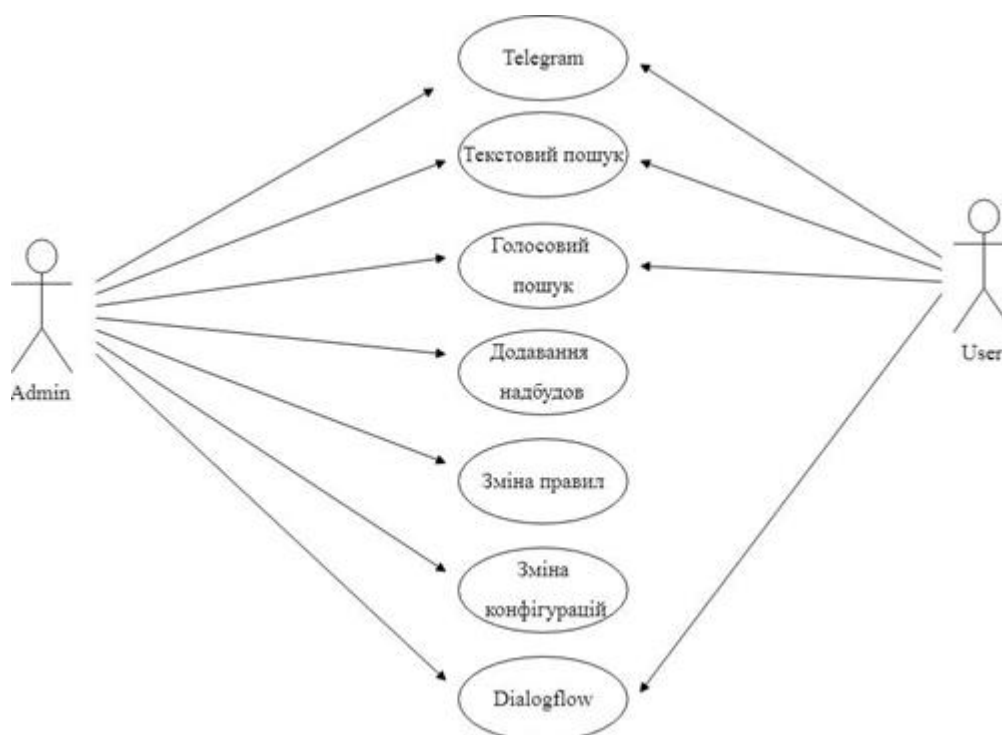


Рисунок 3.5 – Діаграма варіантів використання

4 РОЗРОБКА ІНФОРМАЦІЙНОГО ЧАТ-БОТУ

4.1 Розробка логічної структури чат-боту

Описаний у розділі 2 функціонал робить його унікальним та досить простим у використанні. Щоб зрозуміти, яким чином буде функціонувати даний чат-бот, була розроблена схема, яка зображена на рисунку 4.1.

Виходячи з цього, ми бачимо його унікальність. Дослідження полягає в тому, щоб дослідити, яким чином будуть працювати API як одне ціле. Адже на сьогоднішній час є актуальною робота з API, вони набагато полегшують проектування будь-якого проекту для будь-яких задач. [18]

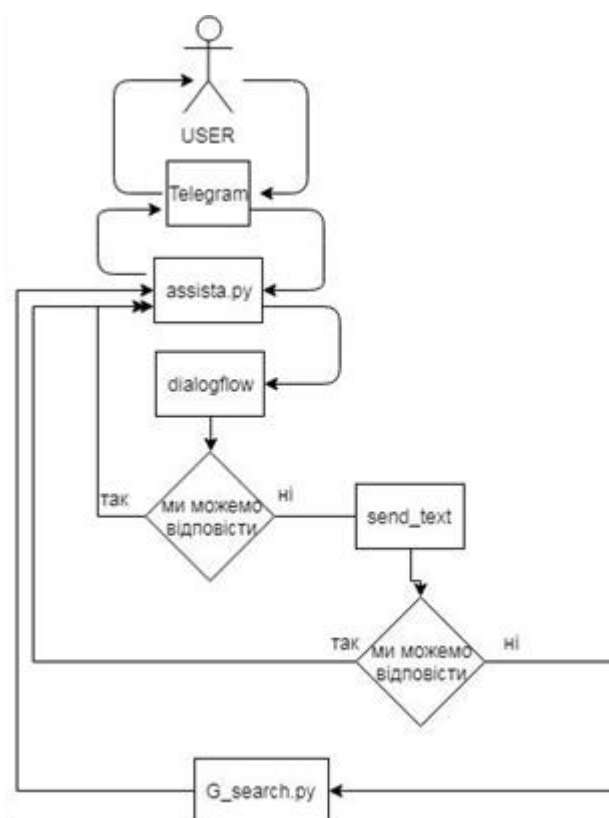


Рисунок 4.1 – Схема роботи чат – бота

Першим кроком необхідно було створити бота в системі телеграм та отримати доступ до месенджеру. Отримання доступу здійснювалось стандартним

методом. Отримання API токenu зображено на рисунку 4.2.



Рисунок 4.2 – Отримання доступу до Telegram боту

Наступний крок-створення скрипту, який об'єднає продукти компанії Googleз месенджером Telegram. На (рис.4.3) представлена схема папок проекту на мові програмування Python.

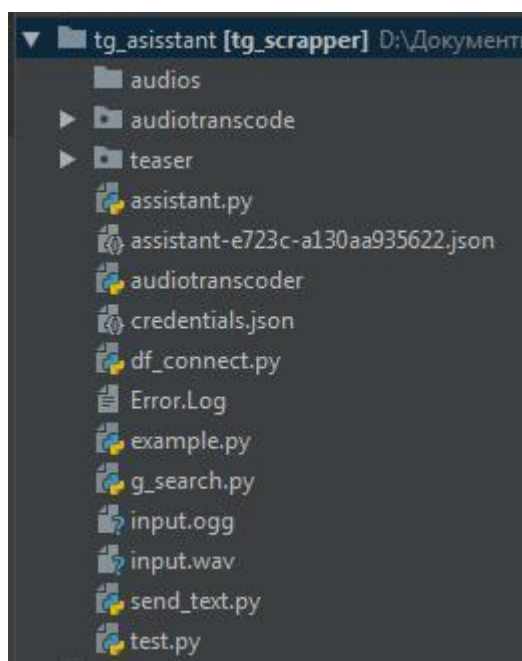


Рисунок 4.3 – Структура проекту

Папка `audios` використовується для збереження усіх аудіо повідомлень від користувача. Папка `audiotranscode` використовується для імпорту бібліотеки, яка змінює формат аудіофайла. Фрагмент програмного коду для збереження аудіо повідомлень:

```
at=audiotranscode.Audiotranscode()
at.transcode('{} .ogg'.format(name), '{} .wav'.format(name))
```

На даний момент папка `teaser` не використовується, але може бути підключена для створення короткого опису великого тексту.[19]

Файл `assistant.py` виконує роль моста між різними продуктами Google та месенджером і є його основною частиною, яка приймає та відправляє повідомлення користувачів та продуктів Google. Це було досягнуто за допомогою бібліотеки `telebot`. Фрагмент програмного коду для ініціалізації боту:

```
bot = telebot.TeleBot('632423476907230:AAEoEya4zob2ZSWgYSr7H
k6JG6nKVUvPLk4')
```

Бот має функціонал для привітання при старті розмови. Фрагмент коду для привітання користувача:

```
@bot.mes_handler(commands3= ['start'], cont_types=['text'])
def send_welcome (mes) :
    bot.send_mes(mes.chat.id, "Hello, I'm Scrapper, your personal
assistant.")
```

Основна частина боту, яка відповідає за обробку повідомлень-запитів користувача та відправлення відповіді бота. Фрагмент коду для обробки усіх повідомлень користувачів:

```
@bot.mes_handler(cont_types=['voice', 'text'])
def get_voice(mes):
    bot.send_mes(mes.chat.id, 'Ok, let me check what you wrote.')
```

`Bot.polling`- це команда, яка запускає сервер боту та дозволяє зберігати повідомлення про помилки у файл. Фрагмент коду для старту бота

Лістинг 4.1 - Фрагмент коду для старту бота

```
def telegram_polling():
    try:
        bot.polling(none_stop=True, timeout=60) #
```

```

except:
    traceback_error_string=traceback.format_exc()
    with open("Error.Log", "a") as myfile:
        myfile.write("\r\n\r\n" + time.strftime("%c")+"\r\n<<ERROR
pollng>>\r\n"+ traceback_error_string + "\r\n<<ERROR pollng>>")
    bot.stop_polling()
    time.sleep(10)
    telegram_polling()

```

Файли з розширенням json - це службові файли, які використовуються для авторизації та взаємодії з продуктами Google.

Файл df_connect.py використовується для взаємозв'язку між скриптом та Dialogflow. Фрагмент коду для взаємодії бота та Dialogflow.

Лістинг 4.2 - Фрагмент коду для взаємодії бота та Dialogflow

```

import dialogflow_v2 as dialogflow
session_client = dialogflow.SessionClient()
session = session_client.session_path('assistant-e723c',
'assistant-e723c')
text_input = dialogflow.types.TextInput(
    text=text, language_code=lang_code)
quer_input = dialogflow.types.QueryInput(text=text_input)
respon = session_client.detect_intent(
    session=session, quer_input=quer_input)
print(respon)
if respon.quer_result.intent.display_name == 'Default Fallback
Intent':
    return None

```

Error.log файл зберігає звіти про помилки, щоб мати можливість відстежити, де, в який час та за яких обставин виникла та чи інша помилка. Фрагмент коду для звіту про помилки.

Лістинг 4.3 - Фрагмент коду для звіту про помилки

```

Sat Jun 2 01:15:16 2018 «ERROR polling»

Traceback (most recent call last):
  File "/usr/local/lib/python3.5/dist-packages/urllib3/connectionpool.py", line 387, in _make_request
    six.raise_from(e, None)

  File "<string>", line 2, in raise_from

  File "/usr/local/lib/python3.5/dist-packages/urllib3/connectionpool.py", line 383, in _make_request

```

```

httplib_response = conn.getresponse()

    File    "/usr/lib/python3.5/http/client.py",   line    1197,   in
getresponse response.begin()

    File    "/usr/lib/python3.5/http/client.py",   line    297,   in begin
version, status, reason = self._read_status()

    File    "/usr/lib/python3.5/http/client.py",   line    258,   in
_read_status line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-
1")

    File    "/usr/lib/python3.5/socket.py",   line    575,   in readinto
return self._sock.recv_into(b)

    File                                          "/usr/local/lib/python3.5/dist-
packages/urllib3/contrib/pyopenssl.py", line 294, in recv_into raise
timeout('The read operation timed out') socket.timeout: The read
operation timed out

```

test.py та example.py файл використовується для проведення тестів та вирішення проблем пов'язаних з конфліктами між модулями.

Файл g_search.py дозволяє відправляти текстові повідомлення в Google Search API та повертає результат пошуку. Фрагмент коду файлу g_search.py.

Лістинг 4.4 - Фрагмент коду файлу g_search.py

```

def search_result(query):
    print('GOT QUERY: ', query)
    search_results = google.search(query, num_page)
    return_results = []
    for result in search_results:
        return_results.append('{}\n{}\nURL:      {}'.format(result.name,
result.description, result.link))
    print(result.name, result.link, result.description)
    result = {'response': return_results}

```

Файл send_text.py використовується для відправлення запитів в Google Assistant API та відображення результату. Спочатку вказуються налаштування асистенту. Фрагмент коду для налаштування Google Assistant.

Лістинг 4.5 - Фрагмент коду для налаштування Google Assistant

```

def iter_assist_zaproq():
    conf = embedded_assist_pb2.AssistConf(
    audio_out_conf=embedded_assist_pb2.AudioOutConf(
    encoding='LINEAR16',

```

```

sample_rate_hertz=16000,
volume_percentage=0,
),
dialog_state_in=embedded_assist_pb2.DialogStateIn(
leng_coding=ego .leng_coding,
conv_state=ego .convn_state,
is_new_conv=ego .is_new_conv,
),
dece_conf=embedded_assist_pb2.DeceConf(
dece_id=ego .dece_id,
dece_model_id=ego .dece_model_id,
),
txt_mas=txt_mas,
)

```

Наступний крок - надіслати запит до Google Assistant API та повернути результат. Фрагмент коду для запиту в Google Assistant API:

Лістинг 4.6 - Фрагмент коду для запиту в Google Assistant API

```

grpc_channel =
google.auth.transp.grpc.secure_authenticated_channel(credentials,
http_zaproq, api_endpoint)
logging.info('Connecting to %s', api_endpoint)
with SampleTxtAssist(lang, dece_model_id, dece_id, ekr,
grpc_channel, grpc_deadline) as assist:
while True:
mas = mas
vidp_txt, vidp_html = assist.assist(txt_mas=mas)
return vidp_txt

```

4.2 Опис функціонування чат-боту

Помічник створений для постійного двостороннього діалогу між людиною і Google, який розуміє наш світ і допомагає виконувати необхідні завдання. Він дозволяє полегшити покупку квитків в кіно, коли людина знаходиться в русі, або пошук ідеального ресторану для родини, щоб влаштувати швидкий перекус перед початком показу фільму, а потім допомогти дістатися до кінотеатру. Оскільки API не розширене і немає синхронізації з іншими сервісами, то Google Assistant видає результат, зображений на рис. 4.14 - 4.15.[20]

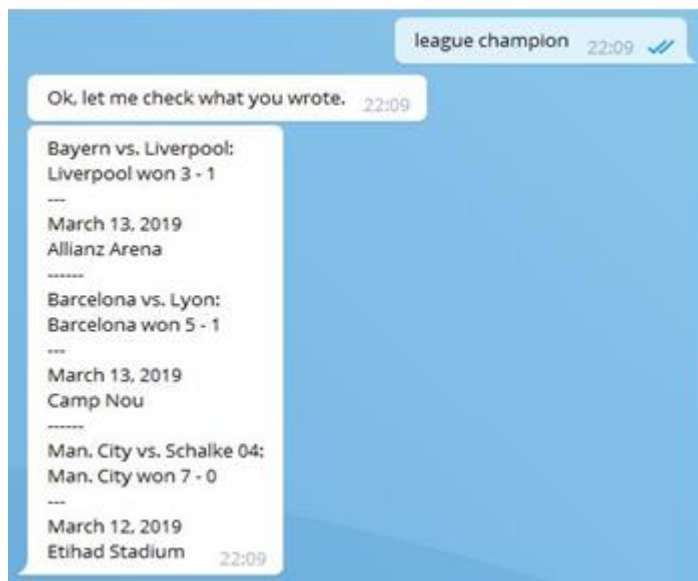


Рисунок 4.14 – Результат від Google Assistant

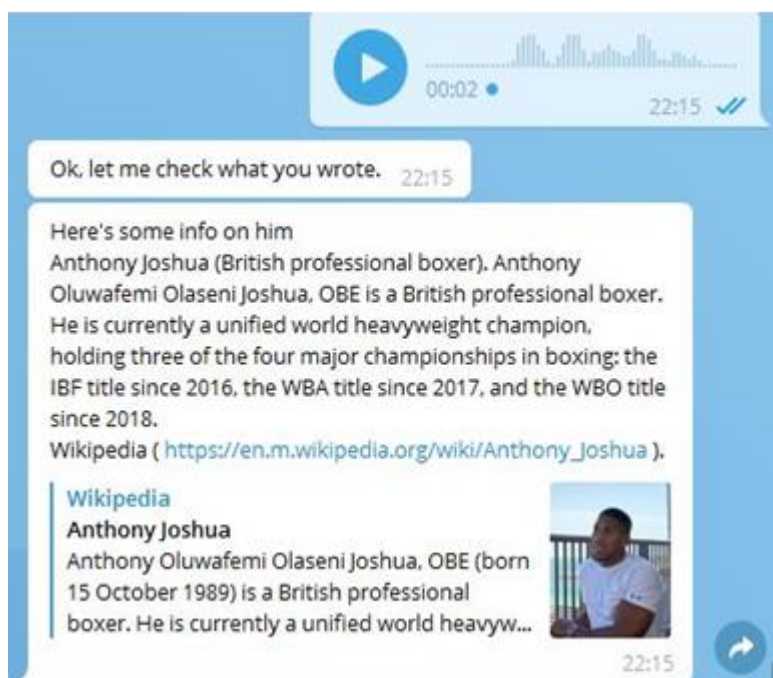


Рисунок 4.15 – Результат від Google Assistant

Google Assistant створена за допомогою нейронних мереж , завдяки котрим ми бачимо саме таку відповідь на наші запити. Асистент спростовує відповідь і робить її формалізованою та чіткою.

Google Search - це пошукова система, розроблена компанією Google. Найбільша пошукова система в світі, котра є на першому місці по результатам точності відповідей на запити. Google отримує інформацію з різних джерел. Ось приклади деяких з них:

- веб-сторінки;
- контент, доданий користувачами, наприклад, опублікований ними в сервісах "Google Мій бізнес" і "Google Карти";
- відскановані книги;
- відкриті бази даних в Інтернеті;
- безліч інших джерел.

Таким чином в даній роботі дана система відіграє другорядну роль. Тобто, якщо асистент неспроможний дати відповідь, одразу відповідь буде надаватись у формі 10 перших посилань сторінки. Це зроблено для того, щоб чат-бот був максимально приближений у наданні відповіді. Приклад результату Google Search зображений на (рис 4.16).[21]

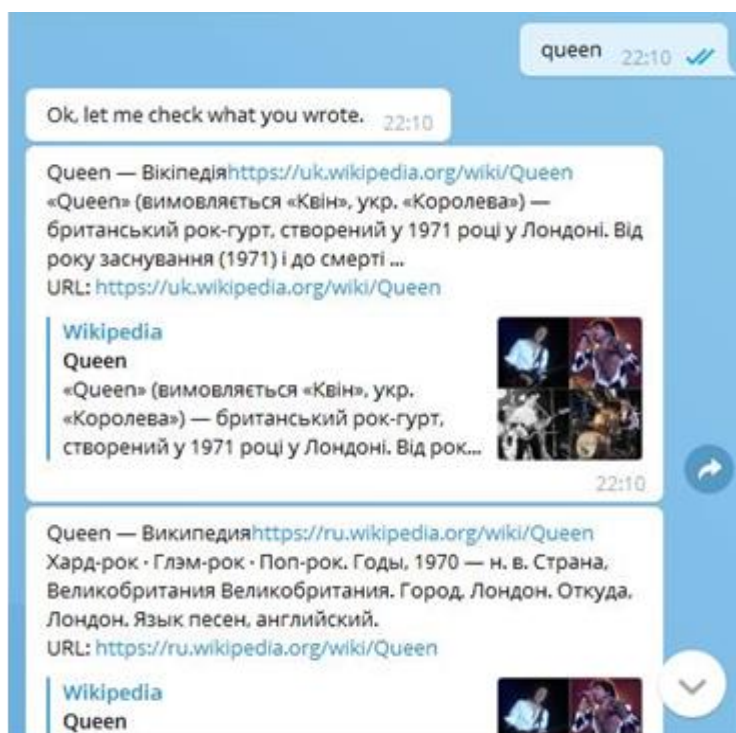


Рисунок 4.16 – Результат від Google Search

Для того щоб бот був більш живим, його необхідно навчити певному спілкуванню з користувачем. Цю задачу виконує Google Assistant, за таким принципом зроблена сама API. Але API працює тільки на англійській мові, спілкування відбувається на англійській мові, і запити виконуються на англійській мові. Приклад спілкування зображений на рисунку 4.17.

Для того щоб бот проводив спілкування зрозумілою мовою, був підключений сервіс компанії Google Dialogflow.

Dialogflow – це онлайн сервіс від Google, який дозволяє створювати своїх чат-ботів або функції для Googleасистента (це називається agent). По суті, Dialogflow вміє зіставити фразу користувача на природній мові (і російською теж) з деяким «немовним» значенням (званим intent) і що-небудь відповісти.



Рисунок 4.17 – Спілкування з чат - ботом

До того ж він ще й тримає контекст діалогу, щоб на наступну фразу можна було зреагувати в контексті попередніх реплік. Інструмент досить простий в управлінні і, крім того, зберігає всю статистику запитів, щоб можна було в один

клік збільшити навчання агента новими intent або поповнювати існуючі. Власне так і відбувається навчання: людина просто показує агенту приклади запитів, які відносяться до того чи іншого intent, а він автоматично будує модель і все краще, і краще розуміє користувача. Також він може виділити з фрази якісь сутності, наприклад, дати або міста, якщо це необхідно для виконання якоїсь функції. Із всіх можливих функцій була використана функція Small Talk. Вона використовується для надання відповідей на випадкові розмови.

Ця функція може значно покращити досвід користувачів, висвітлюючи поширені питання, які можуть не стосуватися намірів агента. В першу чергу, щоб під'єднатись до сервісу, необхідно перейти за посиланням (рис 4.18).[22]

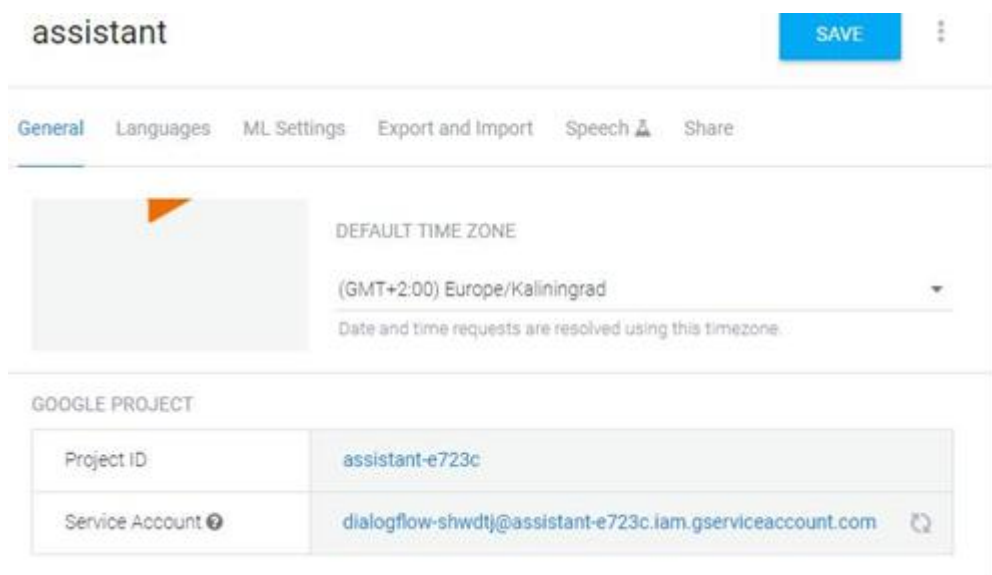


Рисунок 4.18 – Сервісний акаунт

Наступним кроком йде створення сервісного акаунта для того, щоб отримати ключ доступу в форматі json для взаємодії з Dialogflow ботом через python скрипт (рис 4.19). Також можна побачити сервісні акаунти, які взаємодіють з ботом (рис 4.20). Далі необхідно створити власний сервісний акаунт (рис 4.21).

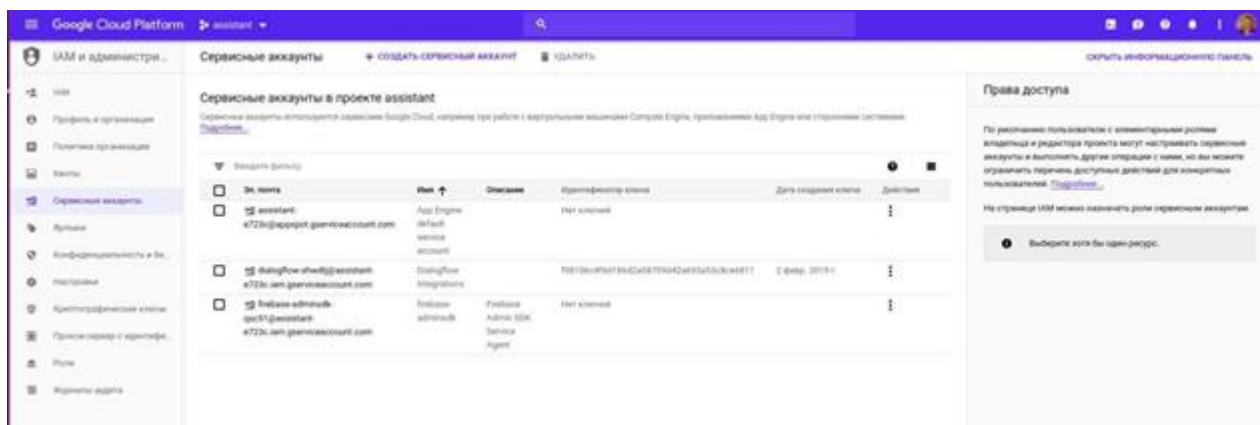


Рисунок 4.19 – Створення сервісного акаунта

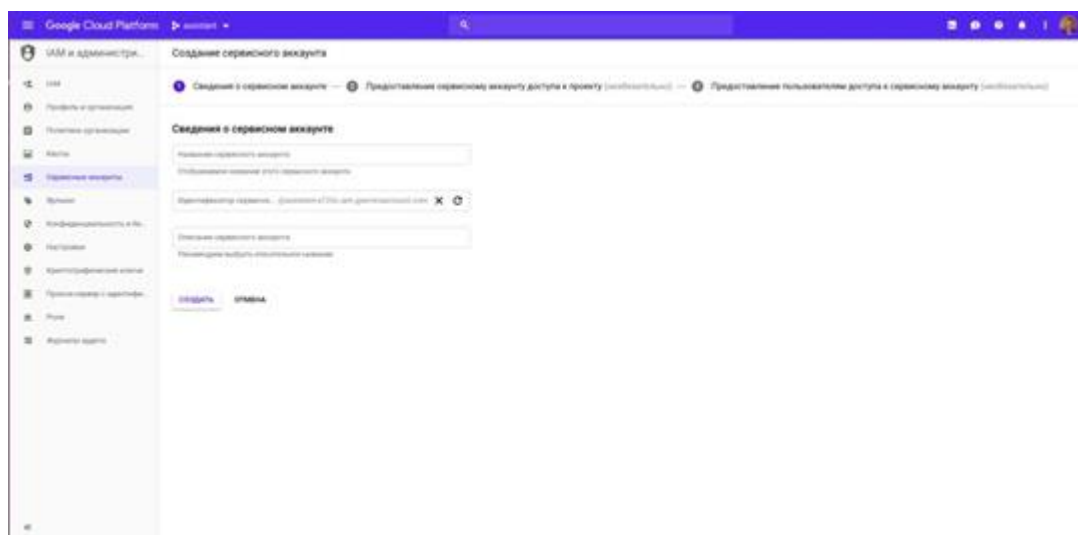


Рисунок 4.20 – Існуючі сервісні акаунти

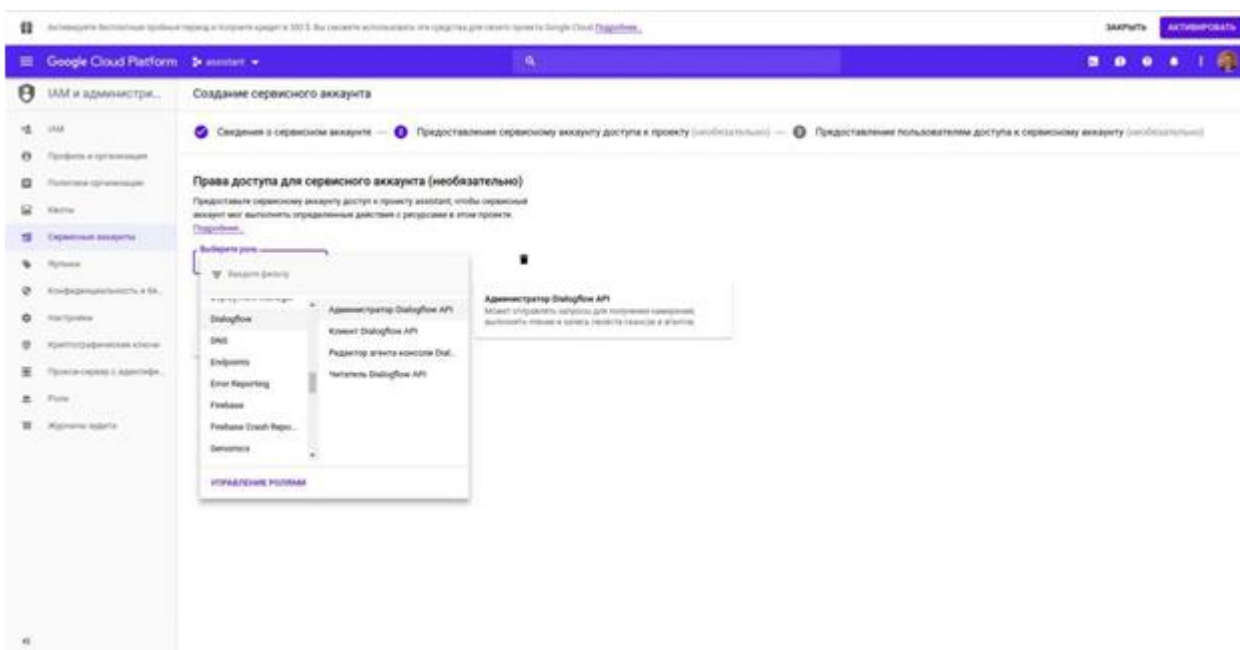


Рисунок 4.20 – Особистий сервісний акаунт

Далі налагоджуємо особистий рівень доступу сервісного акаунту (рівень адміністратора). Створюємо ключ у форматі json файла (рис. 4.21).

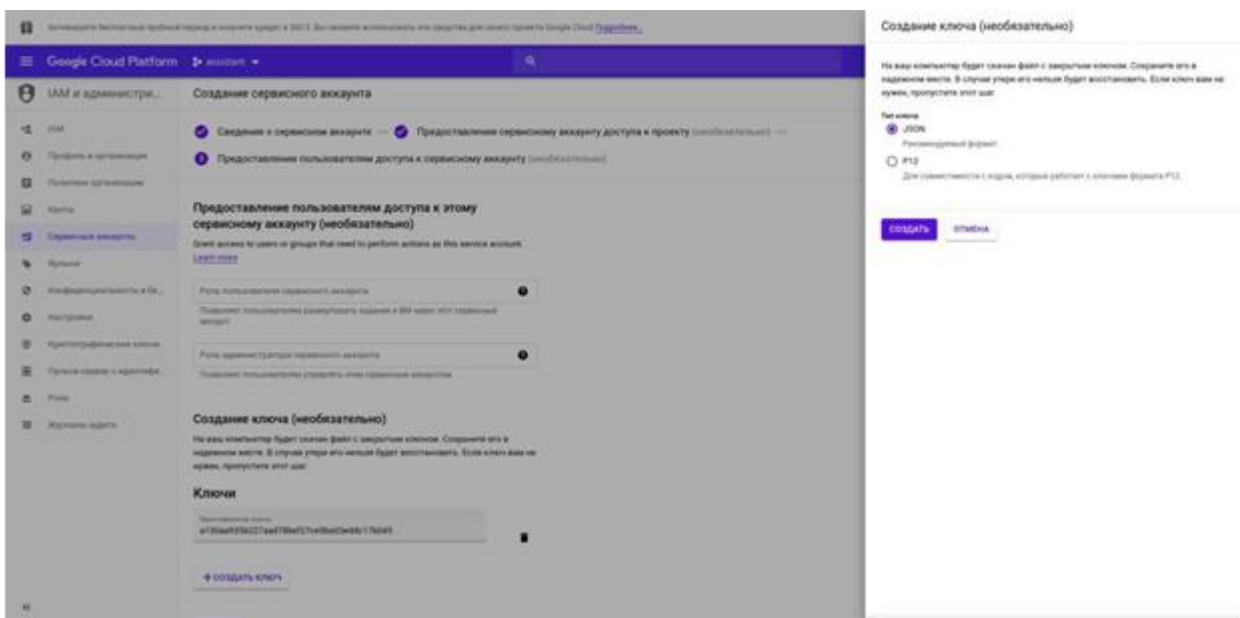


Рисунок 4.21 – Рівень доступу

З метою широкого спілкування можна обрати певні категорії, в яких система

може вільно вести бесіду.

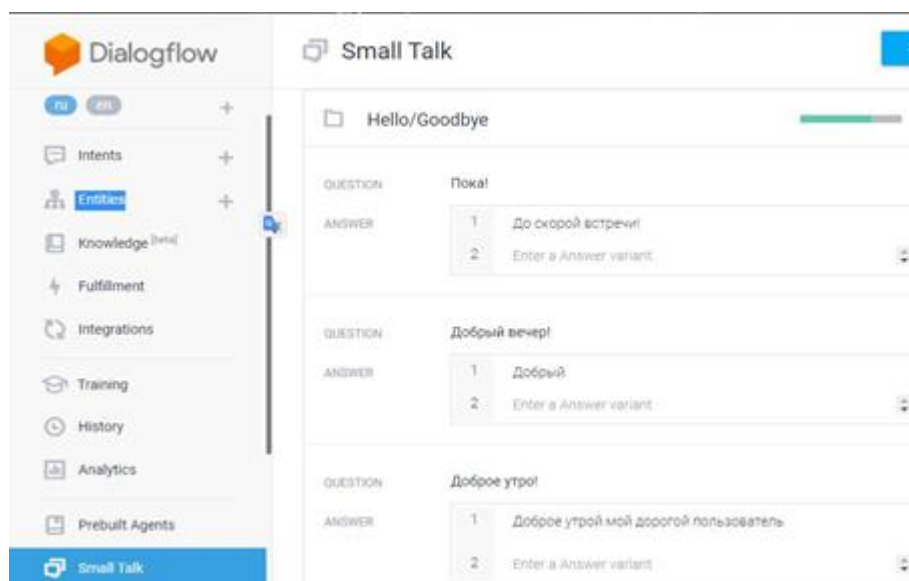


Рисунок 4.22 – Налаштування категорій Dialogflow

Результат роботи Dialogflow в месенджері Telegram зображений на рис.4.23-4.24.

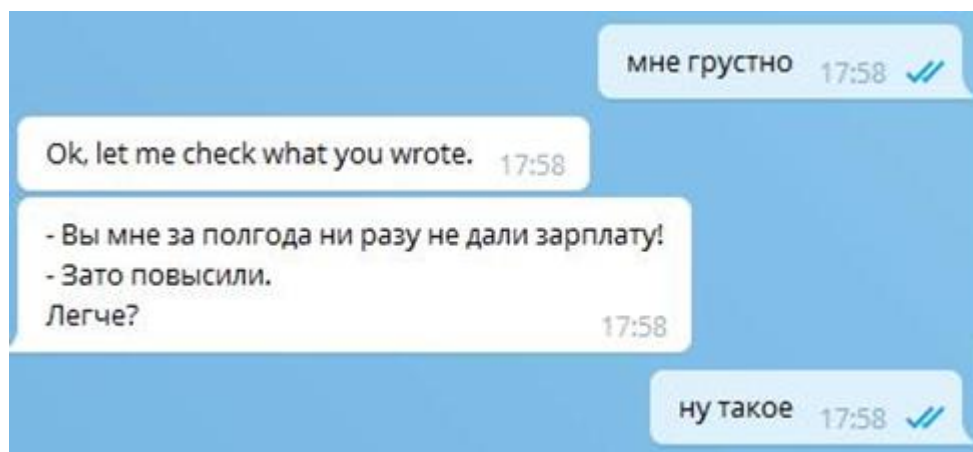


Рисунок 4.23 – Результат роботи Dialogflow

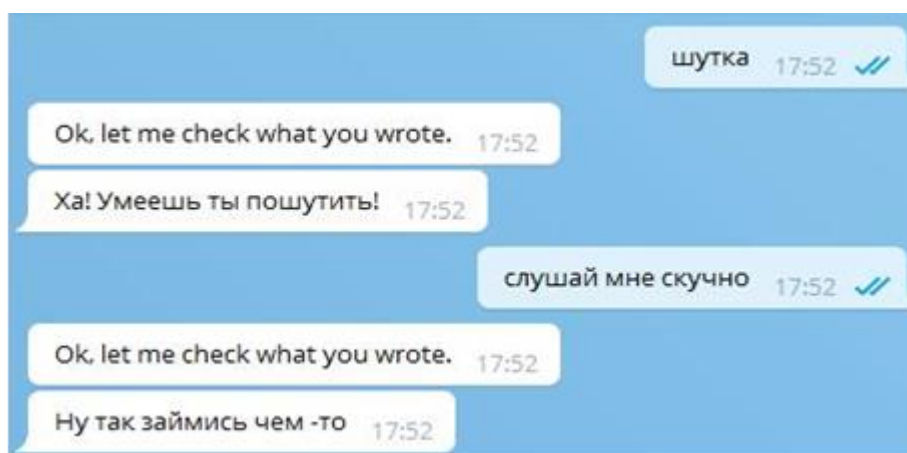


Рисунок 4.24 – Результат работы Dialogflow

Для того щоб пошук здійснювати можна було не лише текстом, а і голосом, була підключена API Speech-to-Text recognition, котра дозволяє синтезувати мову в текст, що робить чат-бота зручнішим та мобільним. Насправді подібних синтезів мови доволі багато, але був здійснений їх аналіз та було прийняте рішення обрати Speech-to-Text recognition.

Результат підключення одразу можна побачити у консольному вікні середовища програмування, яке зображене на рис.4.26. В месенджері даний синтез мови працює таким чином, що необхідно відправити аудіо повідомлення. Приклад зображений на рис. 4.27.

```
GOT QUERY: кто такая леди гага?
Lady Gaga — Википедия https://uk.wikipedia.org/wiki/Lady\_Gaga
Леди Гага — Википедия https://ru.wikipedia.org/wiki/Леди\_Гага
Леди Гага (Lady Gaga) фото, биография, последние новости ... https://uznayvse.ru/znamenitosti/bi
Леди Гага — биография, фото, личная жизнь, новости, песни 2019 ... https://24smi.org/celebrity/6
Леди Гага | Биография Lady Gaga | Фото | Личная жизнь | Журнал ... https://www.cosmo.ru/stars/bi
Леди Гага - все новости, личная жизнь и биография звезды https://viva.ua/stars-bio/ledi-gaga
30 фактов о Леди Гаге - Фактрум https://www.factroom.ru/facts/5160
Леди Гага (Lady Gaga) (Актриса, Музыкант): фото, биография ... https://www.vokrug.tv/person/sho
FROM GOOGLE SEARCH
meest ukrainian
```

Рисунок 4.26 – Результат работы Speech-to-Text recognition



Рисунок 4.27 – Результат работы Speech-to-Text recognition в Telegram

ВИСНОВКИ

Цифровізація економіки і нова тенденція орієнтації систем підтримки прийняття рішень на використання мобільних пристроїв ставлять задачу розробки нового покоління чат - ботів.

Виходячи з аналізу предметної області, а саме дослідження актуальності та необхідності чат-ботів у сучасному житті та їх використання сучасних месенджерів, було вирішено розробити бота, який зможе відповісти на будь-який інформаційний запит.

Аналіз аналогів дав можливість зрозуміти, які типи та які категорії сучасних чат – ботів можна знайти у мережі. Завдяки даному аналізу було вирішено питання актуальності та працездатності боту. Дана розробка буде безкоштовна і буде у відкритому доступі для всіх верств населення, у яких є смартфон, підключення до мережі Інтернет, а також месенджер Телеграм.

Функціональні можливості чат-бота:

- володіти спілкуванням з користувачем;
- пошук інформації за допомогою тексту;
- пошук інформації за допомогою голосового повідомлення.

При реалізації продукту створено WBS-графічне представлення усіх робіт проекту, OBS-схема виконавців цих робіт та на їх основі матрицю відповідальності особи до елементарної роботи, побудовано діаграму Ганта та PDM-мережу, які показують терміни виконання усіх робіт та їх порядок виконання.

В зв'язку з проблемами ефективного використання Telegram для ряду девайсів були виконані та описані принципи побудови ефективної пошукової системи в соціальній мережі Telegram. Реалізовані алгоритми взаємодії користувача з всесвітньою мережею. Чат-бот, реалізований за допомогою скриптової мови Python, а також сервісів Google. Дослідна експлуатація чат боту довела конструктивність підходу, намічено шляхи його вдосконалення.[23]

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Чат-бот - что это такое, как работает, виды, преимущества и недостатки, как создать. URL: <http://kak-bog.ru/chat-bot-cto-eto-takoe-kak-rabotaet-vidy-preimushchestva> (дата звернення 10.05.2021)
2. Чат-боты: обзор и состояние технологий в отрасли т. URL: https://webguru.pro/blog/chat_bots_development/ (дата звернення 10.05.2021)
3. Чат-боты: обзор и состояние технологий в отрасли. URL: <http://nlpx.net/archives/425> (дата звернення 10.05.2021)
4. Чат-бот для бизнеса. URL: <https://www.integrity.com.ua/activities/chatbots-for-business/> (дата звернення 10.05.2021)
5. Exploring chatbots for news. URL: <https://medium.com/nos-digital/exploring-chatbots-for-news-7f12d226c53d> (дата звернення 10.05.2021)
6. AI and Chatbots in Education. URL: <https://chatbotsmagazine.com/ai-and-chatbots-in-education-what-does-the-futurehold-9772f5c13960> (дата звернення 10.05.2021)
7. Advantages of Chatbot Integration into Entertainment Industry. URL: <https://chatbotslife.com/advantages-of-chatbot-integration-into-entertainment-industry-f62bfed3c003> (дата звернення 10.05.2021)
8. Чат-бот : зачем создавать, кому использовать. URL: <https://blog.click.ru/chat-bot-na-sajte-zachem-sozdavat-komu-ispolzovat-i-kak-nastroit/> (дата звернення 10.05.2021)
9. 50 лучших чат-ботов для Telegram. URL: <https://uip.me/2016/04/50-popular-telegram-bots/> (дата звернення 10.05.2021)
10. python-telegram-bot. URL: <https://github.com/python-telegram-bot/python-telegram-bot>. (дата звернення 10.05.2021)
11. Лутц М. Программирование на Python, том II, 4-е издание. Пер. сангл. СПб.: Символ-Плюс. 2011. 992 с.
12. How Assistant work & Introduction to Dialogflow. URL:

<https://codeburst.io/2-how-assistant-work-introduction-to-dialogflow-319a72ba2db>
(дата звернення 10.05.2021)

13. Introduction to the Google Assistant Library. URL: <https://developers.google.com/assistant/sdk/guides/library/python> (дата звернення 10.05.2021)

14. Messaging, simplified. URL: https://www.nexmo.com/products/sms/build?utm_source=google_search&utm_medium=paid&utm_campaign=EMEA_EURO_EN_SMS_General_B&utm_term= (дата звернення 10.05.2021)

15. SpeechRecognition 3.8.1. URL: <https://pypi.org/project/SpeechRecognition/>
(дата звернення 10.05.2021)

16. Enhance Your Presence on Search. URL: <https://developers.google.com/search/> (дата звернення 10.05.2021)

17. Telegram APIs. URL: <https://core.telegram.org/> (дата звернення 10.05.2021)

18. pyTelegramBot API. URL: <https://github.com/eternnoir/pyTelegramBotAPI>
(дата звернення 10.05.2021)

19. Пишем telegram-бота на python с помощью библиотеки telebot. URL: <https://habr.com/ru/post/448310/> (дата звернення 10.05.2021)

20. Cloud Speech-to-Text documentation. URL: <https://cloud.google.com/speech-to-text/docs/> (дата звернення 10.05.2021)

21. Как создать Telegram бота с помощью Python. URL: <https://medium.com/nuances-of-programming/> (дата звернення 10.05.2021)

22. Роботы: информация для разработчиков. URL: <https://tlgrm.ru/docs/bots>
(дата звернення 10.05.2021)

23. Инструкция: Как создавать ботов в Telegram. URL: <https://habr.com/ru/post/262247/> (дата звернення 10.05.2021)

ДОДАТОК А

ПРИКЛАДИ РОБОТИ ЧАТ-БОТУ

Приклад 1

Діалог з ботом через Google Assistant

Оскільки у відкритому доступі API Google Assistant є на англійській мові, то спілкування буде проводитись виключно англійською мовою. Починати діалог, можна з відомої ключової фрази «OK Google», результат зображений на рис.А.1.

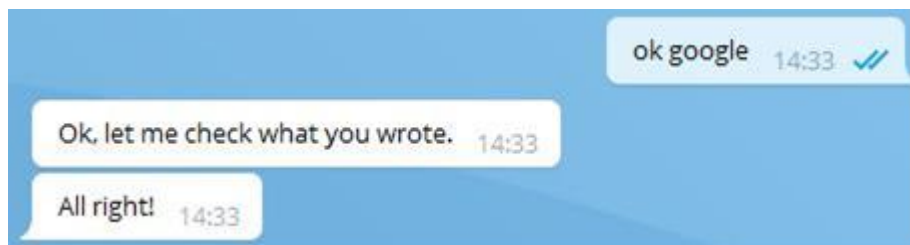


Рисунок А.1 – Початок діалогу

Як можна побачити на рис. А.2. бот відповідає на питання та вітається, але також здатний відповідати на звичайні фрази (рис.А.3).



Рисунок А.2 – Відповіді на питання

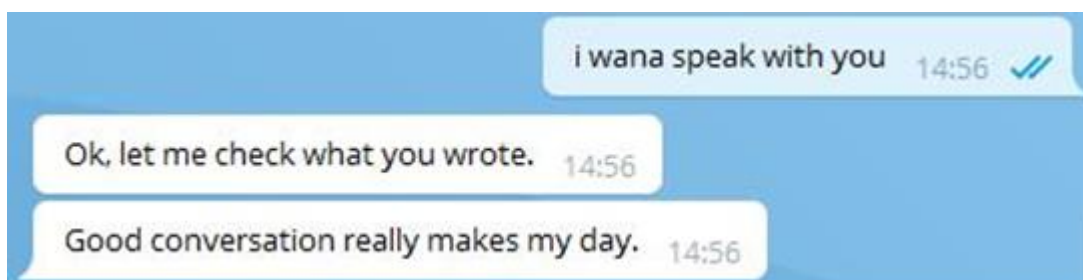


Рисунок А.3 – Підтримка контексту діалогу

Google Assistant представляє собою гарно навчену та логічно структуровану нейронну мережу. Вона генерує відповідь не лише однією фразою на одне і теж питання (рис. А.4- А.5).

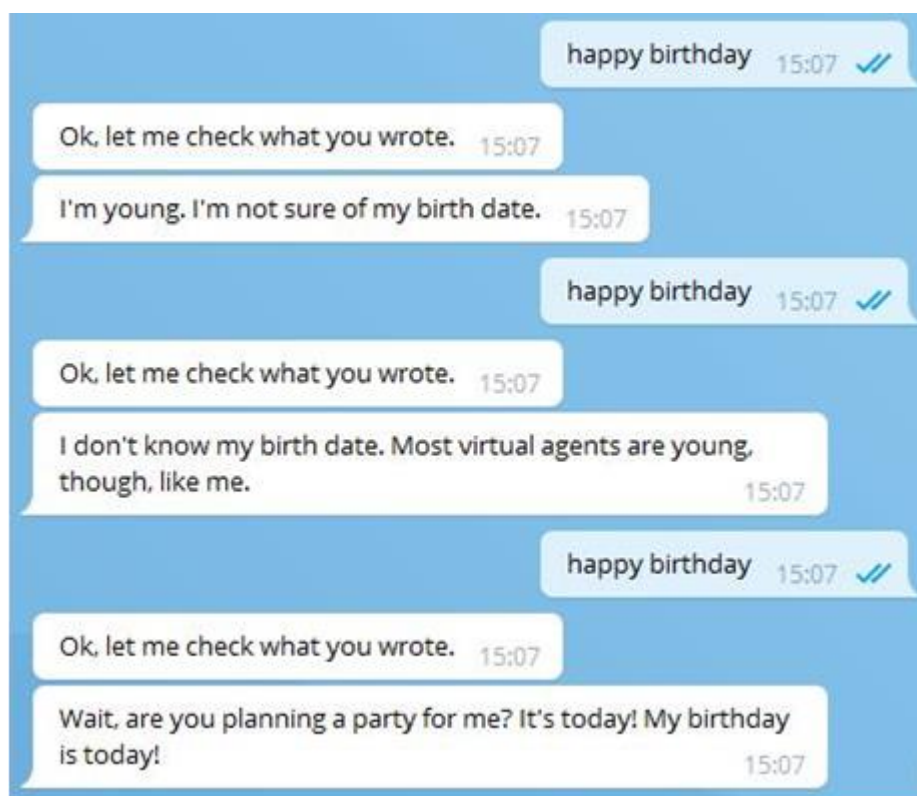


Рисунок А.3 – Приклад варіантів відповіді чат-боту



Рисунок А.4 – Приклад варіантів відповідей на питання

На рис. А.5, можна побачити що асистент, веде діалог зі зворотнім питанням. Тобто максимально наближене до людської розмови.



Рисунок А.5 – Зворотні питання від боту

Якщо асистент, не може відповісти на питання або не розуміє зміст питання, то він про це повідомляє наступним чином, результат повідомлення зображений на рис А.6.

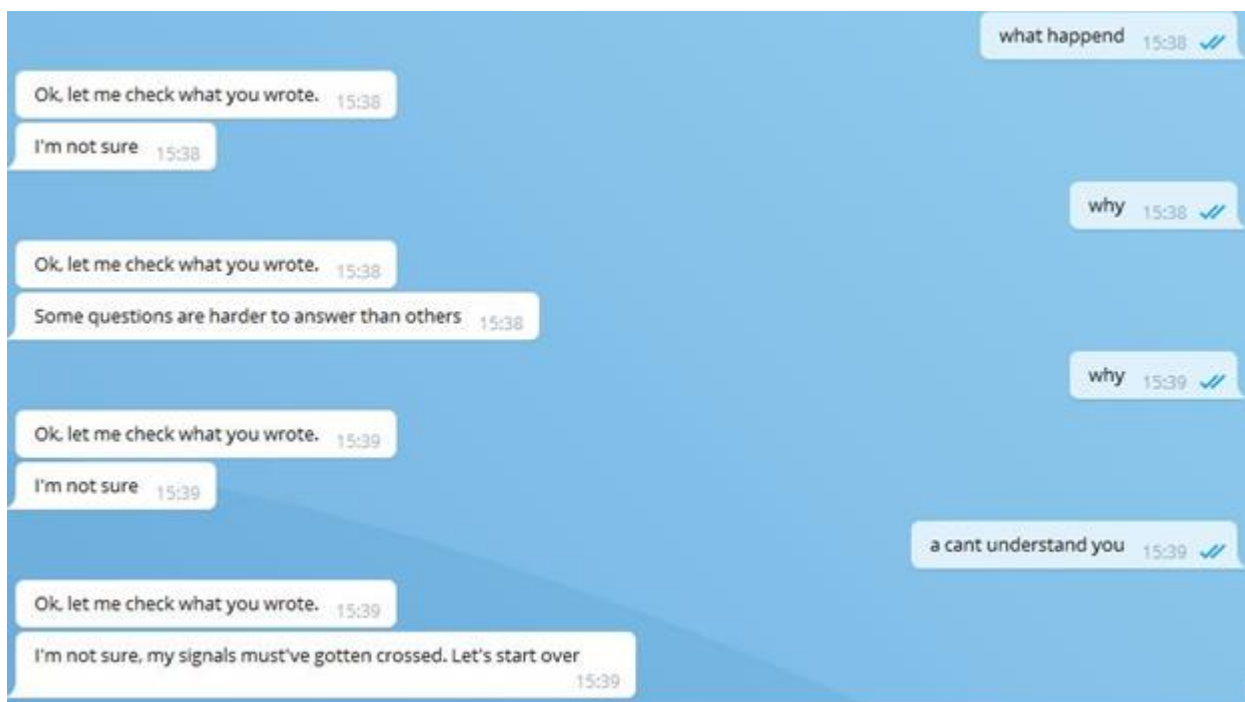


Рисунок А.6 – Приклад роботи з проблемними ситуаціями

Діалог с ботом при підтримці Google Assistant, може мати певний контекст, або вільне спілкування, у форматі чату, результати спілкування з ботом зображені на рис. А.7-А.12.



Рисунок А.7 – Спілкування з асистентом



Рисунок А.8 – Спілкування з асистентом



Рисунок А.9 – Спілкування з асистентом

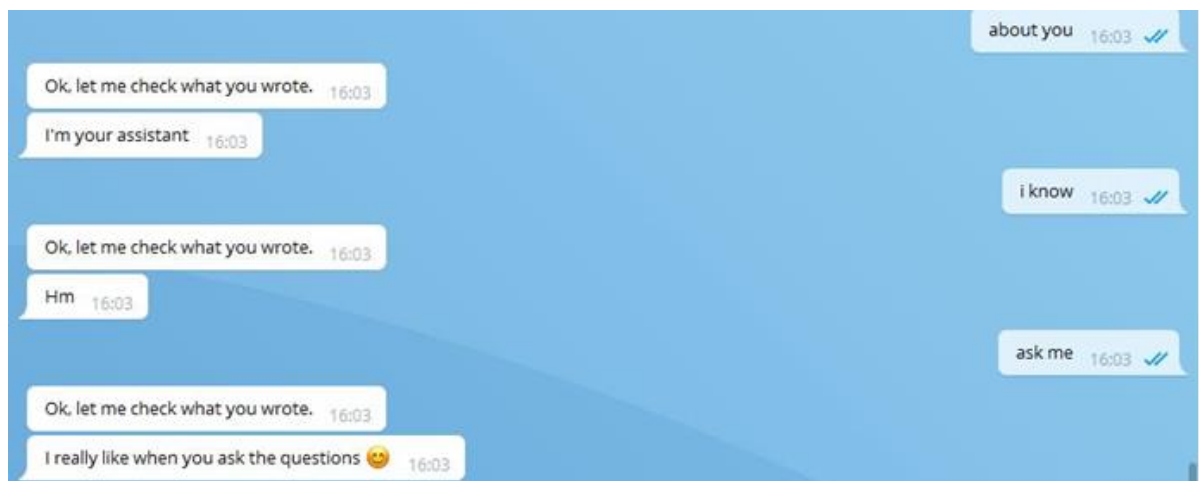


Рисунок А.10 – Спілкування з асистентом

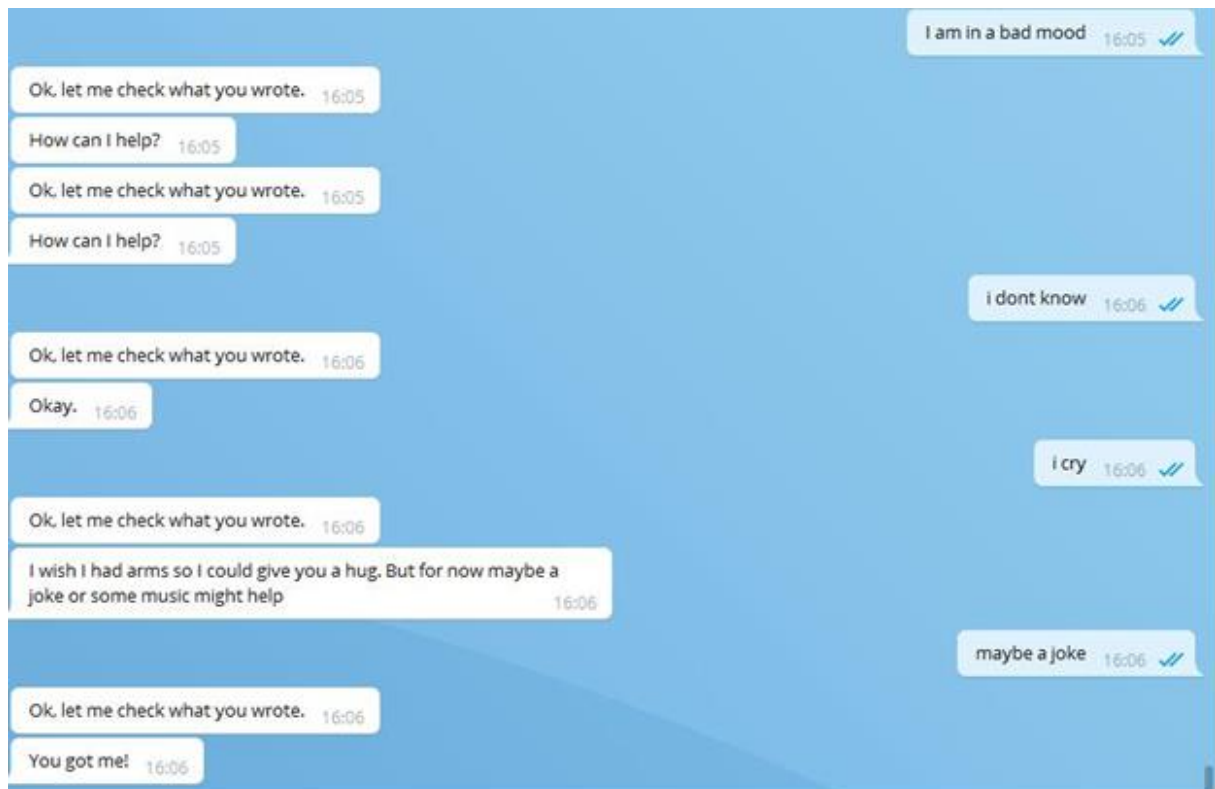


Рисунок А.11 – Спілкування з асистентом

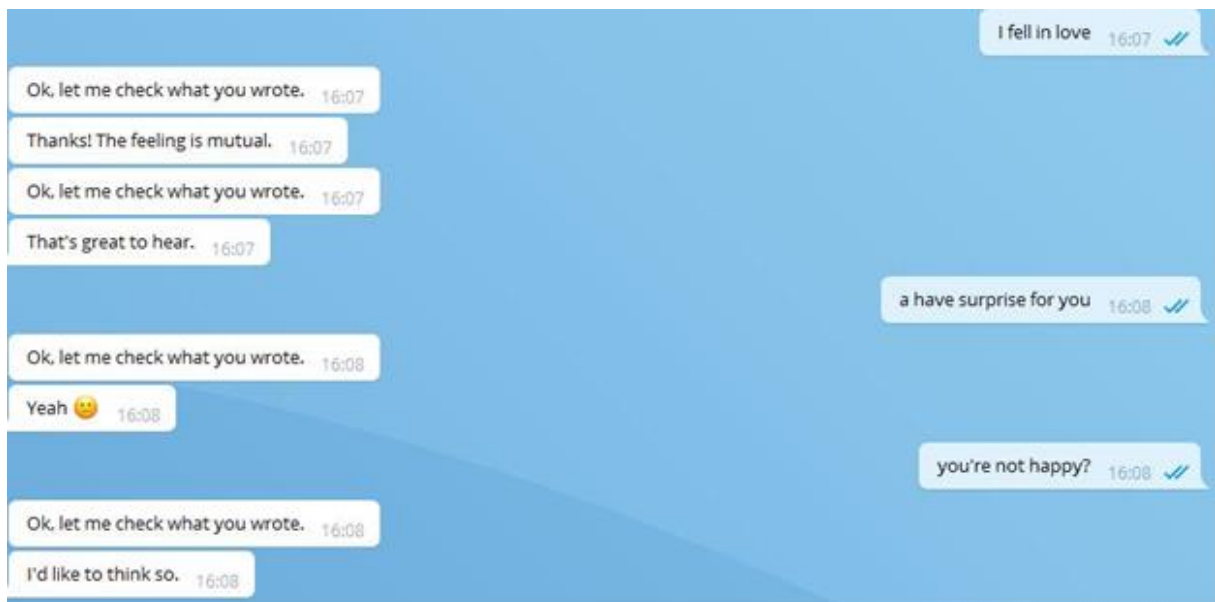


Рисунок А.12 – Спілкування з асистентом

Приклад 2

Діалог з ботом через Dialogflow

За допомогою сервісу Dialogflow була реалізоване спілкування за допомогою російської мови. Dialogflow був підключений для того, щоб бот був різностороннім та багатозадачним. В цьому випадку всі відповіді прописані раніше розробником на ті питання, які запропонував саме сервіс (А.13).

QUESTION	Я очень зол.
ANSWER	1 Успокойся! Просто расслабся и отдохни. 2 Enter a Answer variant
QUESTION	Я вернулся.
ANSWER	1 Куда ходил? 2 Агде ты был? 3 Enter a Answer variant
QUESTION	Мне скучно.
ANSWER	1 Ну так займись чем -то 2 Enter a Answer variant

Рисунок А.13 – Заповнення полів answer

Відповіді будуть генеруватись відповідно питанню. Приклади спілкування з ботом через сервіс Dialogflow зображені на А.14-А.18.

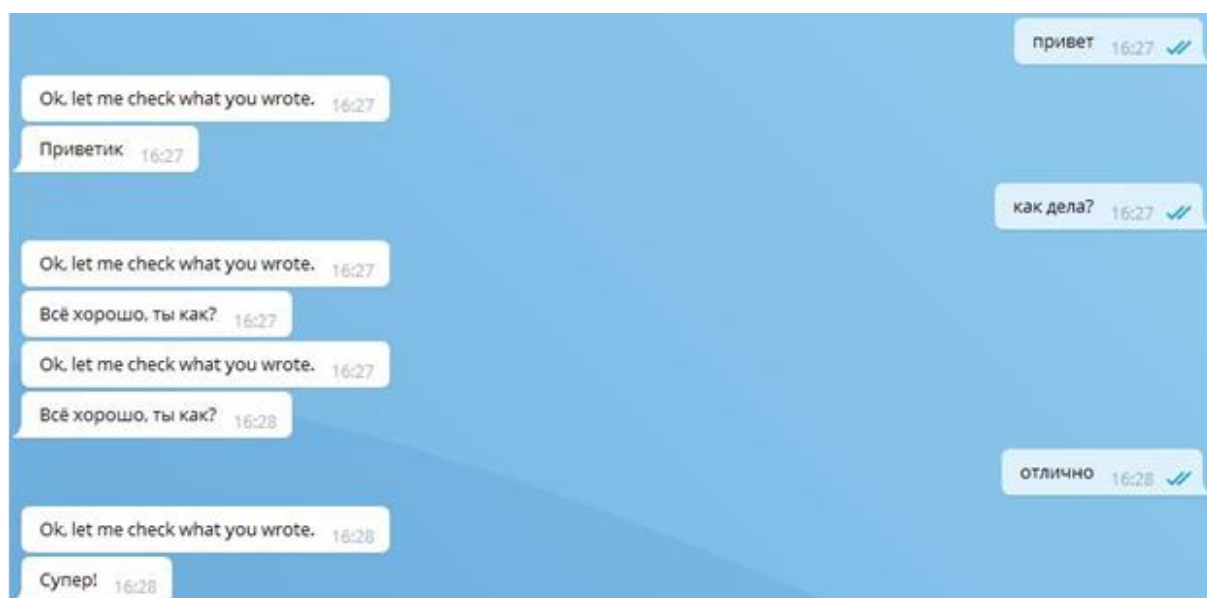


Рисунок А.14 – Спілкування з ботом через сервіс Dialogflow

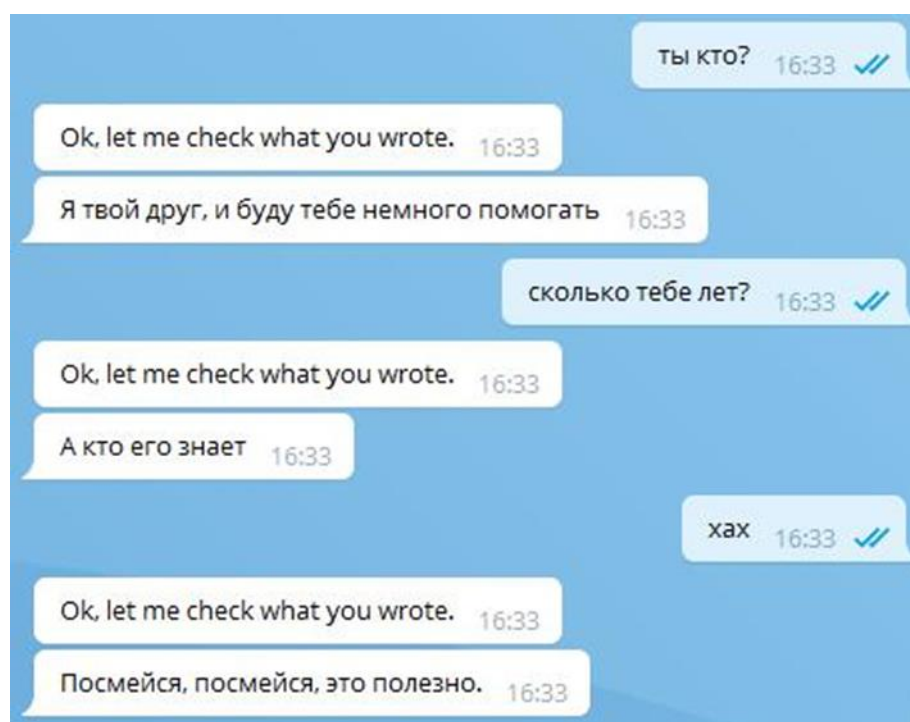


Рисунок А.15 – Спілкування з ботом через сервіс Dialogflow

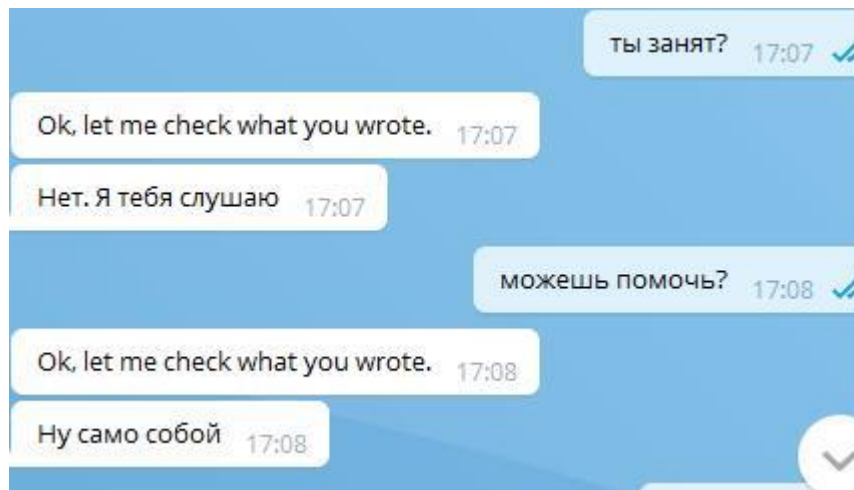


Рисунок А.16 – Спілкування з ботом через сервіс Dialogflow

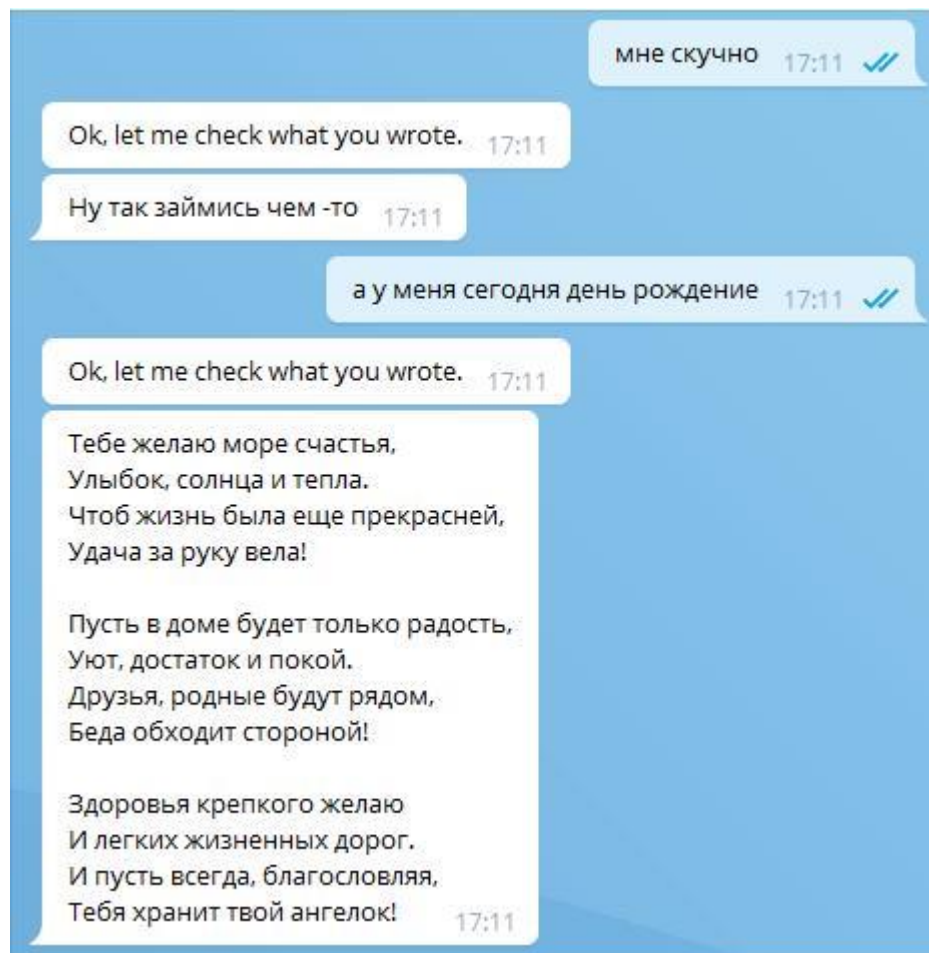


Рисунок А.17 – Спілкування з ботом через сервіс Dialogflow

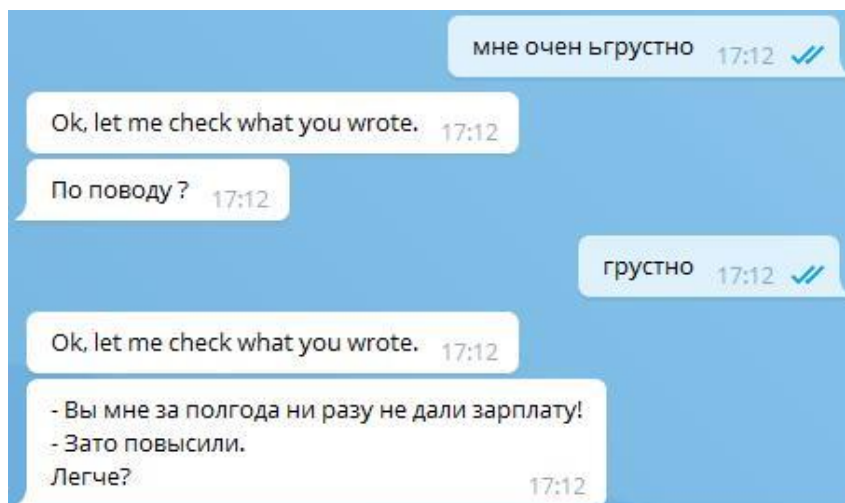


Рисунок А.18 – Спілкування з ботом через сервіс Dialogflow

Приклад 3

Пошук інформації

Пошук інформації здійснюється за допомогою голосового запиту, а також текстовим запитом. Результат залежить від того, яким чином буде задане питання. Відповідь на запит буде або від Google Search або Google Assistant. Google Assistant працює англійською мовою, як аудіо запит так і текстовий. Google Search голосовий запит англійською мовою, текстовий запит англійською та російською. Обробка запиту від Google Assistant голосові та текстові зображені на рис. А.19 - А.27.

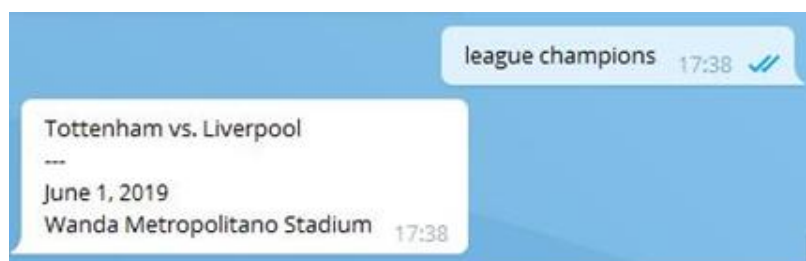


Рисунок А.19 – Текстовий пошук від Google Assistant



Рисунок А.20 – Текстовий пошук від Google Assistant



Рисунок А.21 – Текстовий пошук від Google Assistant

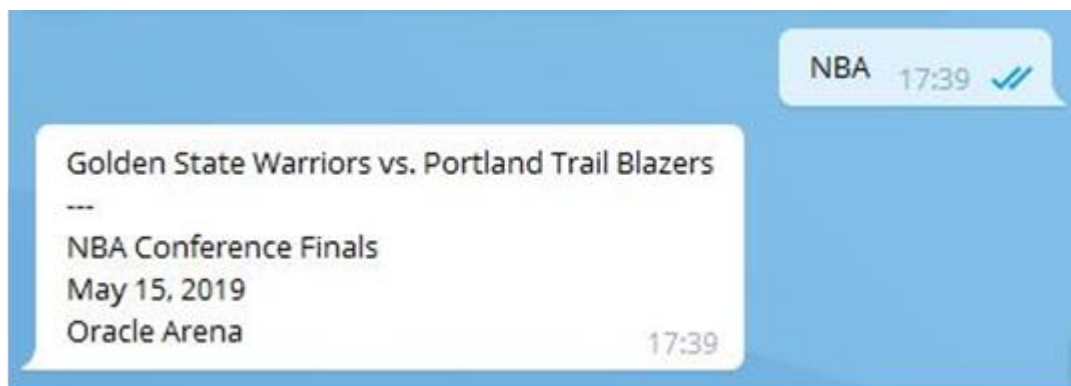


Рисунок А.22 – Текстовий пошук від Google Assistant



Рисунок А.23 – Текстовий пошук від Google Assistant



Рисунок А.24 – Голосовий пошук від Google Assistant



Рисунок А.25 – Голосовий пошук від Google Assistant

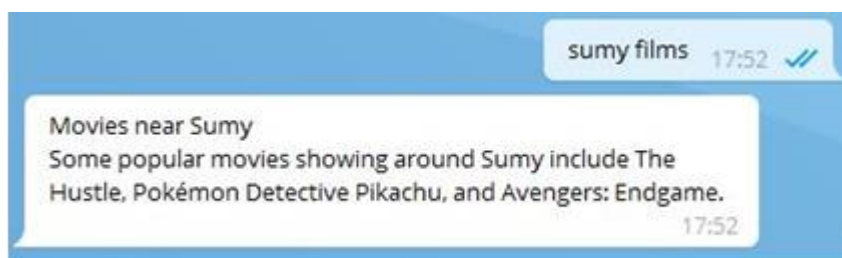


Рисунок А.26 – Текстовий пошук від Google Assistant

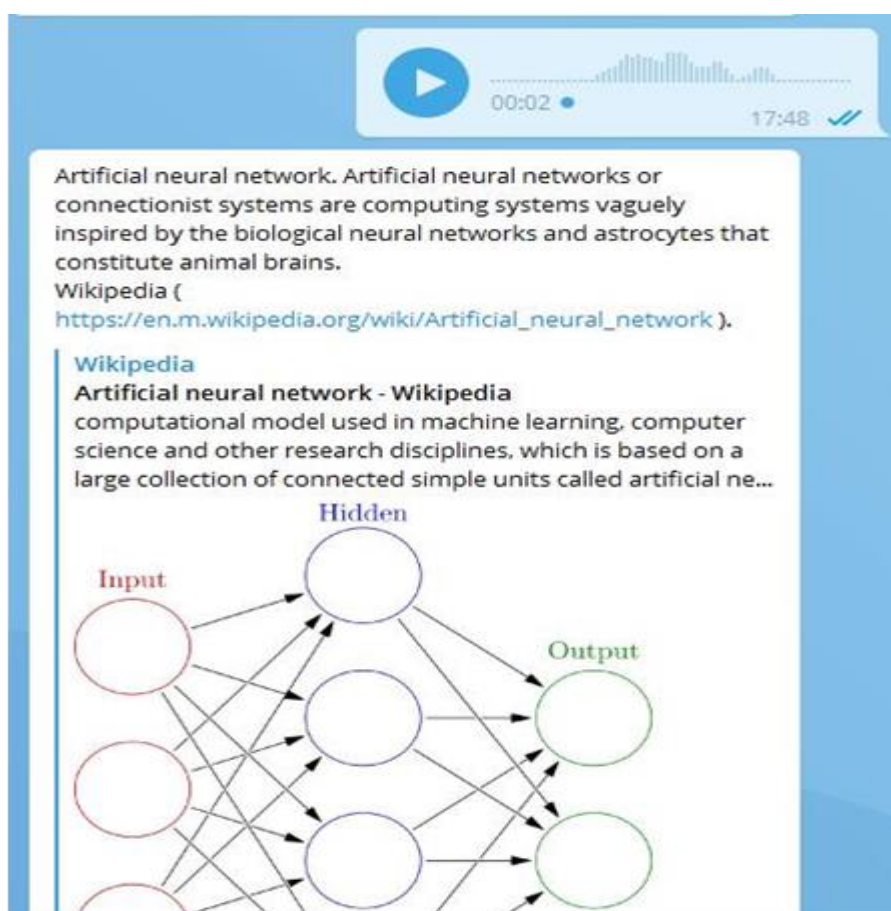


Рисунок А.27 – Голосовий пошук від Google Assistant

Обробка запиту від Google Search зображені на рис А.27 – 32.

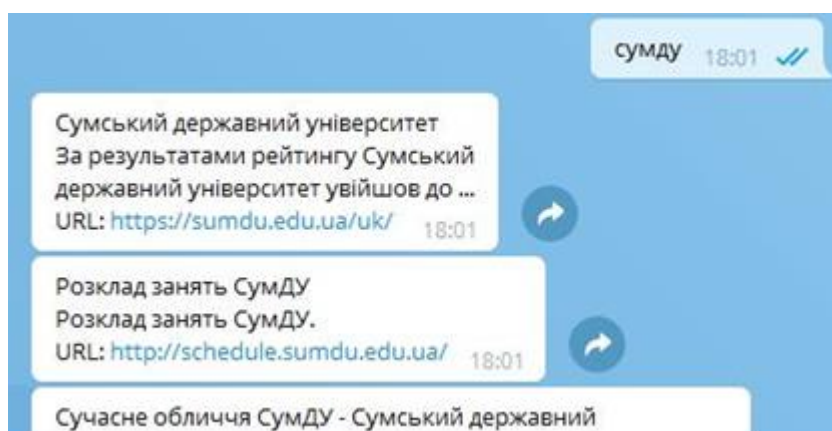


Рисунок А.28 – Текстовий пошук від Google Search

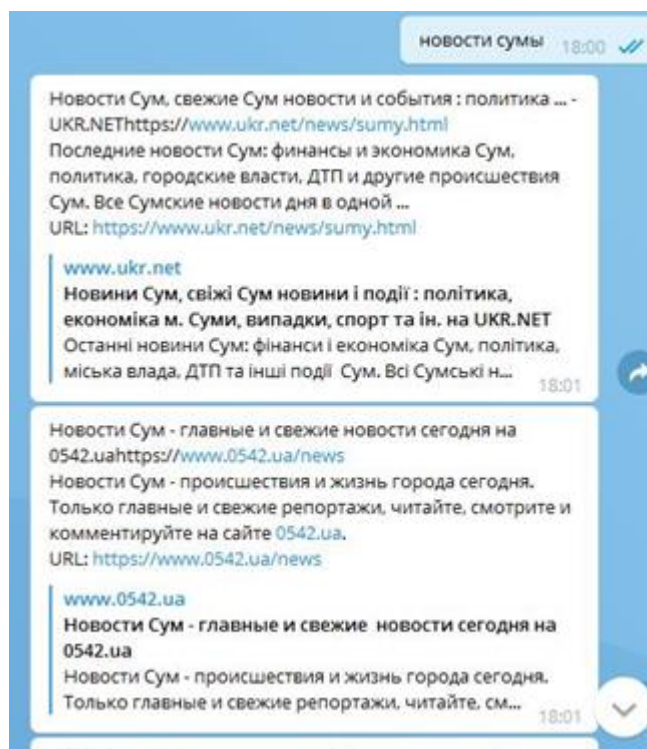


Рисунок А.29 – Текстовий пошук від Google Search



Рисунок А.30 – Голосовий пошук від Google Search

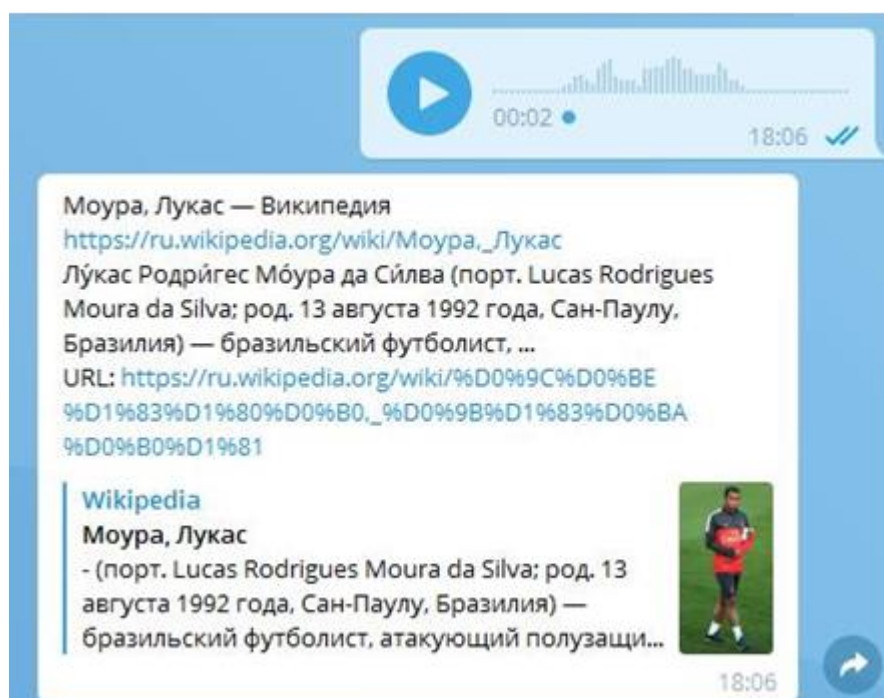


Рисунок А.31 – Голосовий пошук від Google Search

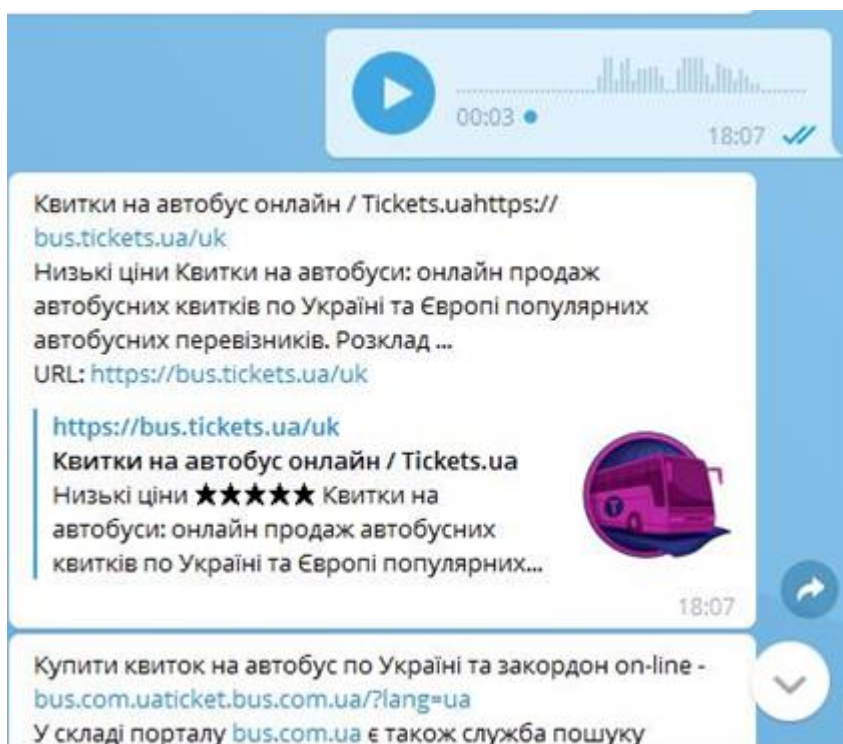


Рисунок А.32 – Голосовий пошук від Google Search

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА

Для користування даним чат-ботом необхідно перейти за посиланням @AssistantEduard_bot, в месенджері Telegram (рис.Б.1).

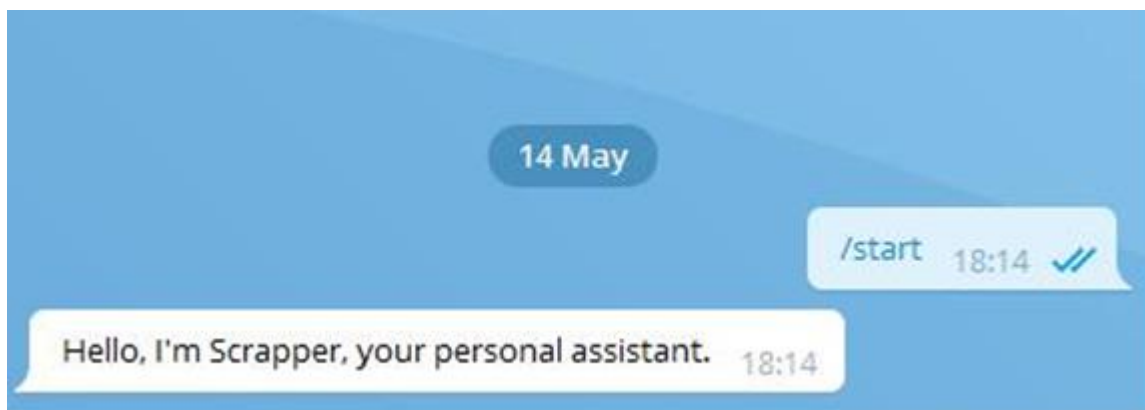


Рисунок Б.1 – Привітальне повідомлення від бота

ДОДАТОК В

КОД ПРОГРАМНОЇ СИСТЕМИ

Лістинг В.1 – Файл коду assistant.py

```

import os
import traceback
import time

import telebot
import detectlanguage
import audiotranscode
import speech_recognition as sr

from pydub import AudioSegment

from send_text import main
from g_search import search_result
from df_connect import detect_intent_texts

from gtts import gTTS

APIKEY = 'eea82a6c343247408c73423b1b2f2d5c16e'

bot =
telebot.TeleBot('632423476907230:AAEoEya4zob2ZSWgYSr7Hk6JG6nKVUv
PLk4')

detectlanguage.configuration.api_key =
"f01173453455c679cd1f096b2afre97a1 cegnhtyjfb4ed67"

CHANNELS = 1
swidth = 2

@bot.mes_handler(commands=['start'], cont_types=['text'])
def send_welcome(mes):
    bot.send_mes(mes.chat.id, "Hello, I'm Scrapper, your personal
assistant.")

@bot.mes_handler(commands=['help'])
def command_help(mes):
    help_text = "Welcome to help center"

    bot.send_mes(mes.chat.id, help_text)

```

```

@bot.mes_handler(commands=['here', 'example'])
def stop_command(mes):
    bot.send_mes(mes.chat.id, 'Who\'s Lady Gaga')

@bot.mes_handler(cont_types=['voice', 'text'])
def get_voice(mes):
    bot.send_mes(mes.chat.id, 'Ok, let me check what you wrote.')

    if mes.cont_type == 'voice':
        file = bot.get_file(mes.voice.file_id)

        download_file = bot.download_file(file.file_path)

        name = 'input'

        with open('{} .ogg'.format(name), 'wb') as new_voice:
            new_voice.write(download_file)

        at = audiotranscode.AudioTranscode()
        at.transcode('{} .ogg'.format(name), '{} .wav'.format(name))

        result = recognize_audio(name, mes.cont_type)
        elif mes.cont_type == 'text':
            result = recognize_audio(mes.text, mes.cont_type)
        else:
            result = 'Unsupported type of mes.'

        if type(result) == dict:
            print('FROM GOOGLE SEARCH')
            for respon in result['respon']:
                bot.send_mes(mes.chat.id, respon)

        bot.send_mes(mes.chat.id, 'That\'s all I found for your query.
I\'m waiting for new tasks!')
        else:
            bot.send_mes(mes.chat.id, result)

def recognize_audio(name, cont_type):
    if cont_type == 'voice':
        try:
            r = sr.Recognizer()
            with sr.AudioFile('{} .wav'.format(name)) as source:
                audio = r.record(source)

            text = r.recognize_google(audio)
            text = text.lower()
        except sr.UnknownValueError:
            return "I can't understand what are you talking about, sorry.
Try again."

```

```

except sr.RequestError as e:
    return "Could not request results from Google Speech
Recognition service; {0}".format(e)
else:
    text = name.lower()

print(text)

lang_code = detectlanguage.detect(text)
print(lang_code)

# try:
if lang_code[0]['language'] == 'en':
    df_answer = detect_intent_texts(text, 'en')
else:
    df_answer = detect_intent_texts(text, 'ru')

if df_answer != None:
    return df_answer
respon = None

if lang_code[0]['language'] == 'en':
    respon = main(query=text)

if respon != None:
    return respon
return search_respon(text)

def telegram_polling():
    try:
        bot.polling(none_stop=True, timeout=60) #
    except:
        traceback_error_string=traceback.format_exc()
        with open("Error.Log", "a") as myfile:
            myfile.write("\r\n\r\n" + time.strftime("%c")+"\r\n<<ERROR
polling>>\r\n"+ traceback_error_string + "\r\n<<ERROR polling>>")
        bot.stop_polling()
        time.sleep(10)
        telegram_polling()

if __name__ == '__main__':
    telegram_polling()

```

Лістинг В.2 – Файл коду df_connect.py

```

import os

os.environ["GOOGLE_APPLICATION_CREDENTIALS"]="./assistant-
e723c-a130aa935622.json"

def detect_intent_texts(text, lang_code):
    import dialogflow_v2 as dialogflow

```

```

    sesion_client = dialogflow. SesionClient()

    sesion = session_client.sesion_path('assistant-e723c',
    'assistant-e723c')

    text_input = dialogflow.types.TextInput(
    text=text, language_code=lang_code)

    quer_input = dialogflow.types.QuerInput(text=text_input)

    respon = sesion_client.detect_intent(
    sesion=sesion, quer_input=quer_input)

    print(respon)
    if respon.quer_rezult.intent.display_name == 'Default Fallback
Intent':
        return None

    print('=' * 20)
    print('Quer text: {}'.format(respon.quer_rezult. quer_text))
    print('Detected intent: {} (confidence: {}) \n'.format(
    respon.quer_rezult.intent.display_name,
    respon.quer_rezult.intent_detection_confidence))
    print('Fulfillment text: {} \n'.format(
    respon.quer_rezult.fulfillment_text))
    print('=' * 20)

    return response.query_rezult.fulfillment_text

```

Лістинг В.3 – Файл коду g_search.py

```

import sys
sys.path.append('D:/diplom/Google-Search-API-master')

from google import google

num_page = 1

def search_rezult(query):
    print('GOT QUERY: ', query)
    search_results = google.search(query, num_page)

    return_results = []

    for result in search_results:
        return_results.append('{} \n {} \n URL:      {}'.format(result.name,
result.description, result.link))
        print(result.name, result.link, result.description)

    result = {'response': return_results}

    return result

```

Лістинг В.4 – Файл коду send_text.py

```

import os
import logging
import json

import click
import google.auth.transp.grpc
import google.auth.transp.zaproq
import google.oauth2.credentials

from google.assist.embedded.v1alpha2 import (
    embedded_assist_pb2,
    embedded_assist_pb2_grpc
)

ASSIST_API_ENDPOINT = 'embeddedassist.googleapis.com'
DEFAULT_GRPC_DEADLINE = 60 * 3 + 5
PLAYING = embedded_assist_pb2.ScreenOutConf.PLAYING

class SampleTxtAssistant(object):

    def __init__(ego, leng_coding, dece_model_id, dece_id,
        ekr, channel, deadline_sec):
        ego.leng_coding = leng_coding
        ego.dece_model_id = dece_model_id
        ego.dece_id = dece_id
        ego.conv_state = None
        ego.is_new_conv = True
        ego.ekr = ekr
        ego.assist = embedded_assist_pb2_grpc.EmbeddedAssistStub(
            channel)

    ego.deadline = deadline_sec

    def __enter__(ego):
        return ego

    def __exit__(ego, etype, e, traceback):
        if e:
            return False

    def assist(ego, txt_mas):
        def iter_assist_zaproq():
            conf = embedded_assist_pb2.AssistConf(
                audio_out_conf=embedded_assist_pb2.AudioOutConf(
                    encoding='LINEAR16',
                    sample_rate_hertz=16000,
                    volume_percentage=0,
                ),
                dialog_state_in=embedded_assist_pb2.DialogStateIn(
                    leng_coding=ego.leng_coding,
                    conv_state=ego.conv_state,

```

```

is_new_conv=ego .is_new_conv,
),
dece_conf=embedded_assist_pb2.DeceConf(
dece_id=ego .dece_id,
dece_model_id=ego .dece_model_id,
),
txt_mas=txt_mas,
)

ego.is_new_conv = False
if ego .ekr:
conf.screen_out_conf.screen_mode = PLAYING
zap = embedded_assist_pb2.AssistZaproq(conf=conf)

yield zap

txt_quar = None
html_vidp = None
for resp in ego .assist.Assist(iter_assist_zaproq(),
ego.deadline):
if resp.screen_out.data:
html_vidp = resp.screen_out.data
if resp.dialog_state_out.conv_state:
conv_state = resp.dialog_state_out.conv_state
ego.conv_state = conv_state
if resp.dialog_state_out.supplemental_ekr_txt:
txt_vidp = resp.dialog_state_out.supplemental_ekr_txt
return txt_vidp, html_vidp
def main(mas,api_endpoint=ASSIST_API_ENDPOINT,
credentials='/home/kirillkiyko/.
conf/google-oauthlib-tool/credentials.json',
ece_model_id='yantoassist-yantoproduct-huxtov',
dece_id='ED82E54B6A31113A5CFB2578F908D807',
ekr=False,
grpc_deadline=DEFAULT_GRPC_DEADLINE,
verbose=False,
lang='en-US'):

logging.basicConfig(level=logging.DEBUG
if verbose else logging.INFO)
try:
with open(credentials, 'r') as f:
credentials = google.oauth2.credentials.
Credentials(token=None, **json.load(f))

http_zaproq = google.auth.transp.zaproq.
Zaproq()
credentials.refresh(http_zaproq)
except Exception as e:
logging.err('Err loading credentials: %s', e)
logging.err('Run google-oauthlib-tool to
initialize ' 'new OAuth 2.0 credentials.')
return

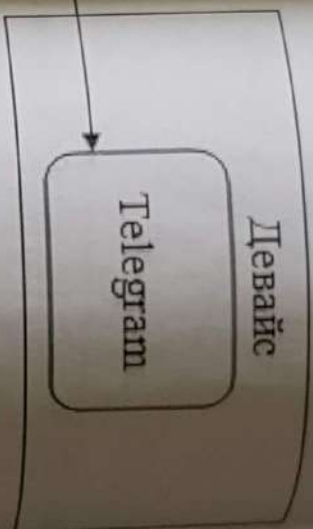
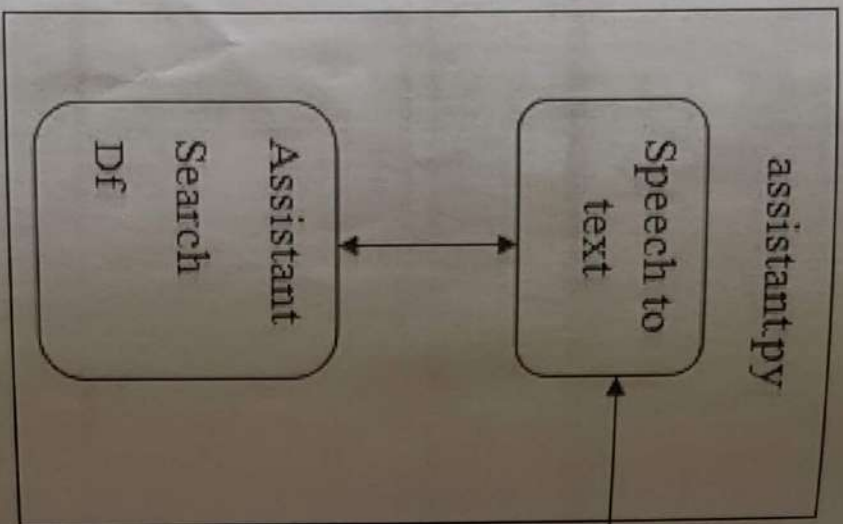
```

```

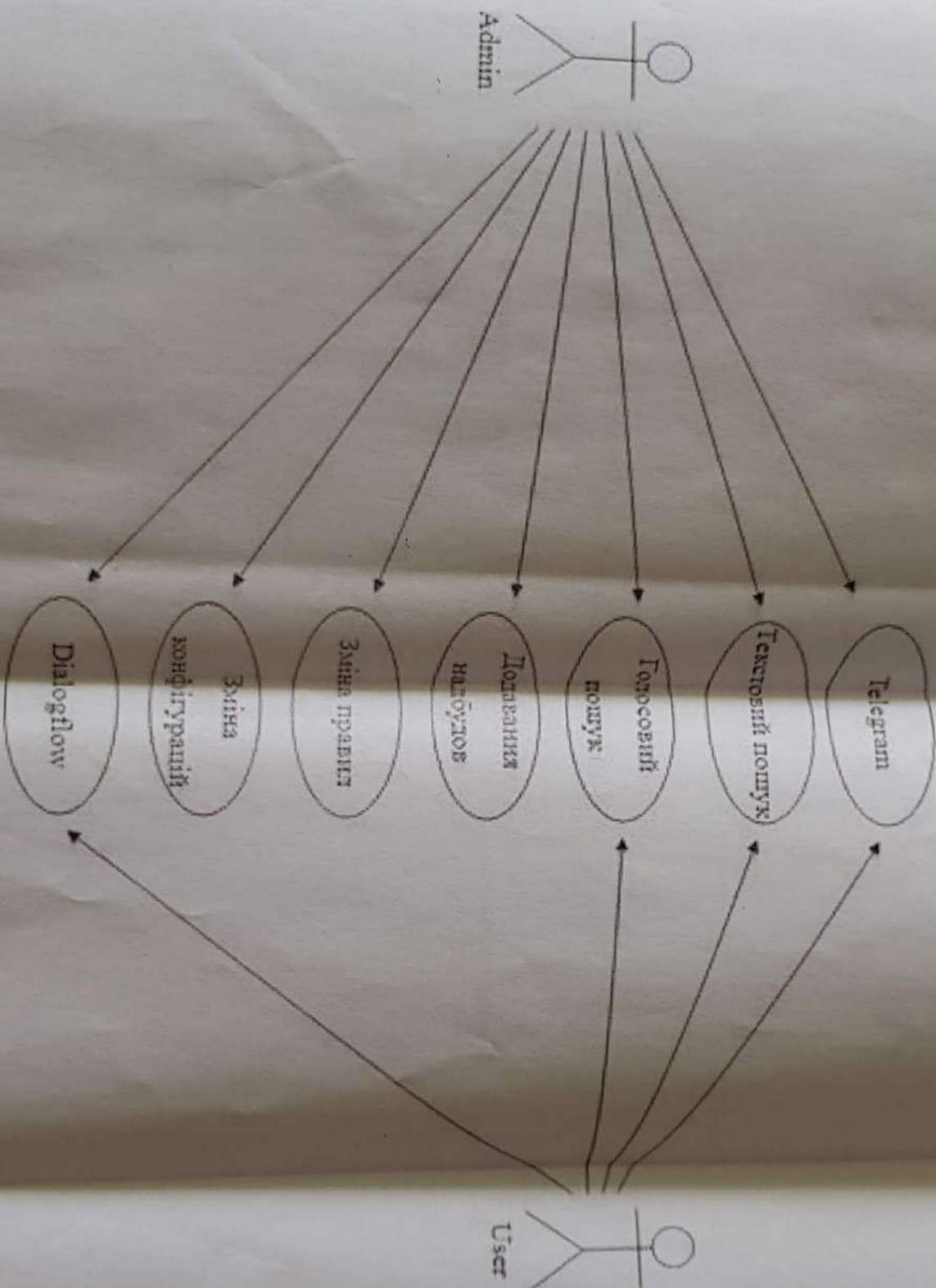
    grpc_channel = google.auth.transp.grpc.
        secure Authorized Channel( credentials,
http_zaproq,api_endpoint)

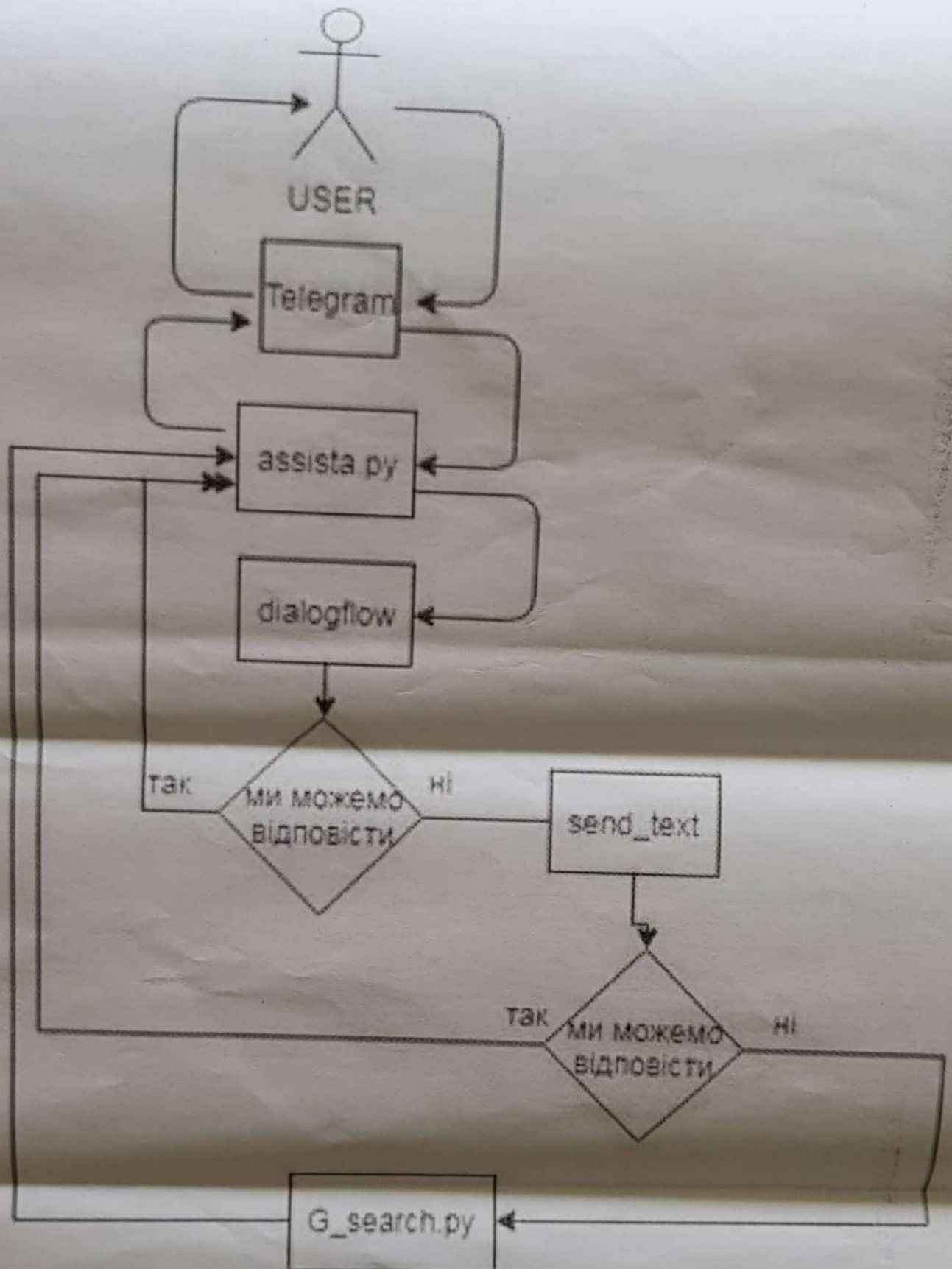
    logging.info('Connecting to %s', api_endpoint)
    with SampleTextAssist(lang, dece_model_id, dece_id,
        ekr, grpc_channel, grpc_deadline) as assist:
        while True:
            mas = mas
            vidp_txt, vidp_html = assist.assist(txt_mas=mas)
            return vidp_txt

```



13.02070849.00062 ПЛ1			
Розробка чат бота з використанням системи розпізнавання тексту			
(структура сай-бота)			
Зав. проєктом	№ документа	Місце	Дата
Розроб.	Літман В.О.	Місце	Дата
Перев.	Івашченко М.Б.	Місце	Дата
Технік		Місце	Дата
Начальн.	Земко О.В.	Місце	Дата
Зам.	Квашенко Р.Б.	Місце	Дата
ІНУ «Запорізька політехніка»		КМТ-51809	

[illegible]



Зач.	Лист.	М. докум.	Підп.	Дата
Розроб.	Латвиш В.О.			
Перев.	Ільчишченко М.Б.			
Текстур.				
Н. контр.	Зелін О.В.			
Затв.	Будьничев Р.К.			

13.02070849.00062 ПЛЗ

Розробка чат бота з
використання системи
розпізнавання тексту

Алгоритм роботи чат
– бота

Лист	Маса	Масштаб
Лист 3	Лист 4	
НУ «Запорізька політехніка КНТ-518сн		

Створення сервісного акаунта

Напаштування категорій Dialogflow

Особистий сервісний акаунт

Результат роботи Dialogflow

13.02070849.00062 ПП4