МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»
В.о. завідувача кафедри

_____Світлана ВАЩЕНКО

____ _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»_____,

освітньо-професійної програми «Інформаційні технології проектування»_____
на тему: «Телеграм–бот для перекладу текстів іноземною мовою»_____

_____

Здобувача(ки) групи_ІТ-92___Чабаненко Родіона Артемовича_____
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Родіон ЧАБАНЕНКО
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник__старший викладач кафедри «Інформаційні технології»
к.т.н. Едуард КУЗНЄЦОВ_____
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ
В.о. зав. кафедри ІТ

_____ С.М. Ващенко
«__» _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Чабаненко Родіону Артемовичу

1 Тема роботи *Телеграм-бот для переклу текстів іноземною мовою*
керівник роботи *Едуард Геннадійович Кузнєцов, к.т.н.*

затверджені наказом по університету № 0588-VI від «*29* »травня 2023 р.

2 Строк подання студентом роботи « » _____ 2023 р.

3 Вхідні дані до роботи технічне завдання на розробку Телеграм – боту для перекладу текстів іноземною мовою

4 Зміст розрахунково-пояснювальної записки (перелік питань, які **потрібно розробити)** аналіз предметної області, проектування Телеграм-боту, розробка Телеграм-боту

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) вступ, актуальність, аналіз предметної області, огляд останніх досліджень і публікацій, аналіз програмних продуктів, постановка задачі, проектування Телеграм-боту, структурно-функціональне моделювання боту, моделювання варіантів використання, архітектура, реалізація, висновки.

6. Консультанти розділів роботи:

Розділ	**Консультант**	**Підпис, дата**	
		Завдання видав	**Завдання прийняв**

7.Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення планування робіт	24.02.2023 – 03.03.2023	
2	Оформлення технічного завдання	04.03.2023 – 08.03.2023	
3	Аналіз предметної області	09.03.2023 – 23.03.2023	
4	Проектування архітектури системи	24.03.2023 – 31.03.2023	
5	Вибір технологій та інструментів розробки	01.04.2023 – 15.05.2023	
6	Реалізація функціоналу	16.05.2023 – 23.05.2023	
7	Тестування та налагодження	24.05.2023 – 31.05.2023	
8	Оформлення пояснювальної записки	01.06.2023 – 10.06.2022	

Студент _____ Чабаненко Р.А.
(підпис)

Керівник роботи _____ к.т.н., Кузнєцов Е.Г.
(підпис)

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Телеграм-бот для перекладу текстів іноземною мовою».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із 20 найменувань та трьох додатків. Загальний обсяг пояснювальної записки складає 68 сторінок, у тому числі 40 сторінок основного тексту, 3 сторінки списку використаних джерел, 25 сторінок додатків.

Розроблюваний бот – програма, яка працює в середині месенджера – шляхом автоматичного перекладу повідомлень та чатів з однієї мови на іншу буде сприяти спрощенню комунікації з іноземними партнерами, спілкуванню з іншомовними друзями, та пришвидшить пошук нових контактів, адже переклад відбуватиметься безпосередньо в самому месенджері без переривання бесіди чи переговорів.

Перший розділ складається з огляду досліджень за тематикою роботи, порівняльного аналізу аналогів розроблюваного боту, висунення функціональних вимог до розроблюваного бота. Сформульовано мету проекту та задачі для її досягнення.

Другий розділ включає в себе структурно-функціональне моделювання, побудову діаграм взаємодії з користувачем. Представлено контекстну діаграму в нотації IDEF0 та її декомпозицію, діаграму варіантів використання, застосування логіки MVC-архітектури.

У третьому розділі визначено внутрішню будову боту, його ієрархію. Розроблено програмну реалізацію, приведено детальний опис процеса розробки, показано роботу створюваного бота.

Результатом проведеної роботи є працездатний Телеграм-бот для багатостороннього застосування в якості автоматизованого перекладача набраних текстів, текстів з зображень та голосових повідомлень.

Ключові слова: бот, Телеграм, Python, API ключ, бібліотеки Python, текст в зображенні, голосове повідомлення.

ЗМІСТ

ВСТУП

У сучасному світі зростає необхідність в миттєвому перекладі тексту на різні мови, а також у зручному перекладі тексту з картинок та голосових повідомлень. Це стає особливо актуальним у сфері міжнародних комунікацій, подорожей та бізнесу. З метою забезпечення зручності та ефективності використання перекладу, необхідно розробити Телеграм-бота, який забезпечує переклад тексту на інші мови, переклад тексту з картинок та голосових повідомлень.

Телеграм-боти наразі стають ще більш популярними та гнучкими аніж звичайні додатки. Месенджери Телеграм доступні через мобільні пристрої та настільні комп'ютери, що робить їх зручними для користувачів із будь-яких пристроїв. Крім того, створення та використання ботів у Телеграмі нескладне і не потребує спеціальних технічних навичок.

Телеграм-боти можуть бути легко інтегровані з іншими сервісами та додатками, такими як поштові сервіси, соціальні мережі, календарі і т.д. Це дозволяє використовувати ботів для автоматизації різних процесів і спрощення вашого повсякденного життя.

Телеграм покладає великий акцент на безпеку та приватність користувачів. Всі повідомлення у Телеграм шифруються та захищаються протоколом шифрування TLS. Крім того, Телеграм-боти можуть бути налаштовані на захист приватності, наприклад, можуть працювати в режимі конфіденційних чатів або видаляти повідомлення автоматично після певного часу. Такі функції забезпечують додатковий рівень безпеки та конфіденційності.

Тому задача розроблення та впровадження бота-перекладача наразі вкрай актуальна. Такий бот буде затребуваний серед сьогоднішніх студентів, фрилансерів, які працюють з закордонними замовниками та людей, які полюбляють подорожі та знайомства з мешканцями інших країн.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд новітніх досягнень по тематиці роботи

Останні роки визнані визначними для технологій машинного перекладу, особливо завдяки застосуванню глибокого навчання та нейромереж. Методи машинного перекладу на основі нейромереж, відомі як моделі Transformer, виявилися дуже успішними в досягненні вражаючої якості перекладу. Застосування таких моделей дозволяє здійснювати більш точні та природні переклади.

Крім того, останні дослідження також фокусуються на поєднанні машинного перекладу з іншими технологіями, наприклад, комп'ютерним зором для перекладу тексту зображень або розпізнаванням мови для перекладу голосових повідомлень. Ці поєднання допомагають розширити можливості перекладачів і зробити їх більш універсальними та зручними для користувачів.

Також з'являються нові методи та алгоритми для покращення якості перекладу, такі як самооцінка перекладу, генеративні моделі зі змінною довжиною послідовності та використання контекстуальних векторів слів. Всі ці техніки спрямовані на зменшення помилок та покращення зрозумілості та природності перекладу.

Завдяки комп'ютерному зору та розпізнаванню образів, технології можуть виявляти та виділяти текст на зображеннях, що дозволяє перекладати написи на табличках, банерах, фотографіях меню ресторанів та інших джерелах інформації.

Технології розпізнавання та синтезу голосу дають змогу перекладачу розпізнавати голосові повідомлення користувачів та перекладати їх на обрану мову, а також створювати голосові відповіді з перекладом.

Збереження конфіденційності та захист даних користувачів є важливим аспектом розробки Телеграм-бота. Важливо використовувати надійні методи шифрування та забезпечувати безпеку передачі даних між користувачем і

ботом. Також необхідно дотримуватися політики конфіденційності, яка забезпечує захист особистої інформації користувачів.

Сьогодні будь-який розроблюваний бот можна інтегрувати в сторонні сервіси перекладу, такі як Google Translate, Microsoft Translator або іншими відомими перекладачами. Це дозволяє використовувати вже існуючі рішення з машинного перекладу та забезпечити широкий спектр мов для перекладу.

Для зручного використання Телеграм-бота важливо розробити зручний та інтуїтивно зрозумілий інтерфейс користувача. Це може включати текстове поле для введення тексту, кнопки вибору мови перекладу, кнопки завантаження зображень та інші елементи, які спрощують взаємодію з ботом.

Важливим аспектом розробки Телеграм-бота для перекладу є перевірка якості перекладів. Це можна здійснити за допомогою оцінювання якості перекладів або використанням спеціалізованих метрик, таких як BLEU (Bilingual Evaluation Understudy) або METEOR (Metric for Evaluation of Translation with Explicit Ordering).

1.2 Порівняльний аналіз аналогів розроблюваного програмного продукту

Існує кілька аналогічних Телеграм-ботів для перекладу текстів іноземною мовою, за текстом, за картинкою та за голосовим повідомленням. Один з прикладів такого бота – DeepL Translator Bot.

DeepL Translator Bot – це бот, який використовує сервіс перекладу DeepL для забезпечення точного та якісного перекладу текстів (рис. 1.1). Він підтримує переклад з багатьох мов, включаючи англійську, німецьку, французьку, іспанську та інші. Бот може перекладати текстові повідомлення, а також виявляти та перекладати текст зображень та голосові повідомлення.

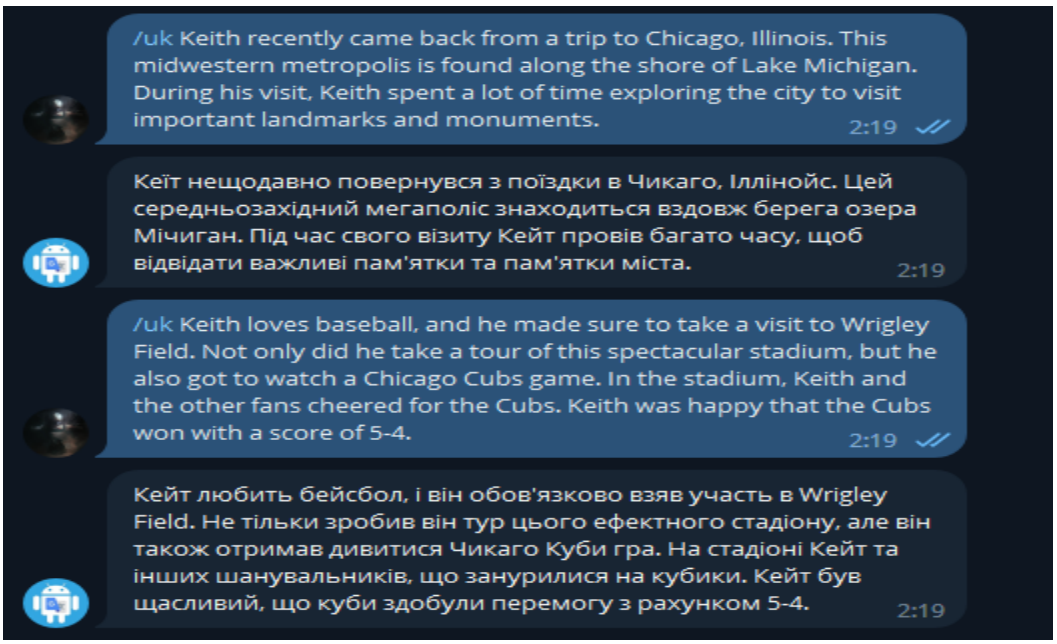

Рисунок 1.1 – приклад використання DeepL

Порівняння між розглянутим ботом та іншими аналогами може бути засноване на таких факторах:

– Доступність мов. Варто перевірити, які мови підтримуються кожним ботом для перекладу. Деякі боти можуть мати більш широкий вибір мов для перекладу, що може бути важливим фактором для користувачів з різними мовними потребами.

– Точність перекладу. Різні боти можуть мати різний рівень точності перекладу. Варто оцінити, наскільки точні та надійні є переклади, здійснені кожним ботом.

– Функціональність. Різні боти можуть мати різні функціональні можливості. Наприклад, деякі боти можуть підтримувати тільки текстові переклади, тоді як інші можуть мати можливість перекладу зображень або голосових повідомлень. Вибір бота може залежати від конкретних потреб користувача.

– Інтерфейс користувача. Важливим фактором є зручність та легкість використання інтерфейсу користувача бота. Деякі боти можуть мати простий та інтуїтивно зрозумілий інтерфейс, який дозволяє швидко та легко здійснювати

переклад. Інші боти можуть мати додаткові функції, такі як команди, кнопки або взаємодія з елементами інтерфейсу для зручного використання.

– Вартість та обмеження. Деякі боти можуть бути безкоштовними для використання, а інші можуть мати платну підписку або обмеження на кількість перекладів. Варто оцінити вартість та обмеження кожного бота, особливо якщо ви плануєте використовувати його на постійній основі або для великого обсягу перекладів.

– Швидкість та продуктивність. Різні боти можуть мати різний рівень швидкості та продуктивності при здійсненні перекладу. Це може бути важливим фактором для користувачів, які шукають швидкі та миттєві результати перекладу.

– Репутація та відгуки користувачів. Перегляд відгуків та репутації ботів може дати вам уявлення про їхню якість та задоволення користувачів. Рекомендації від інших користувачів можуть бути корисними при виборі найкращого бота для перекладу.

Google Translate Bot є Телеграм-ботом, який використовує сервіс Google Translate для перекладу текстів іноземною мовою. Цей бот пропонує широкий спектр функціоналу та може бути використаний для перекладу текстових повідомлень, зображень та голосових повідомлень (рис. 1.2).

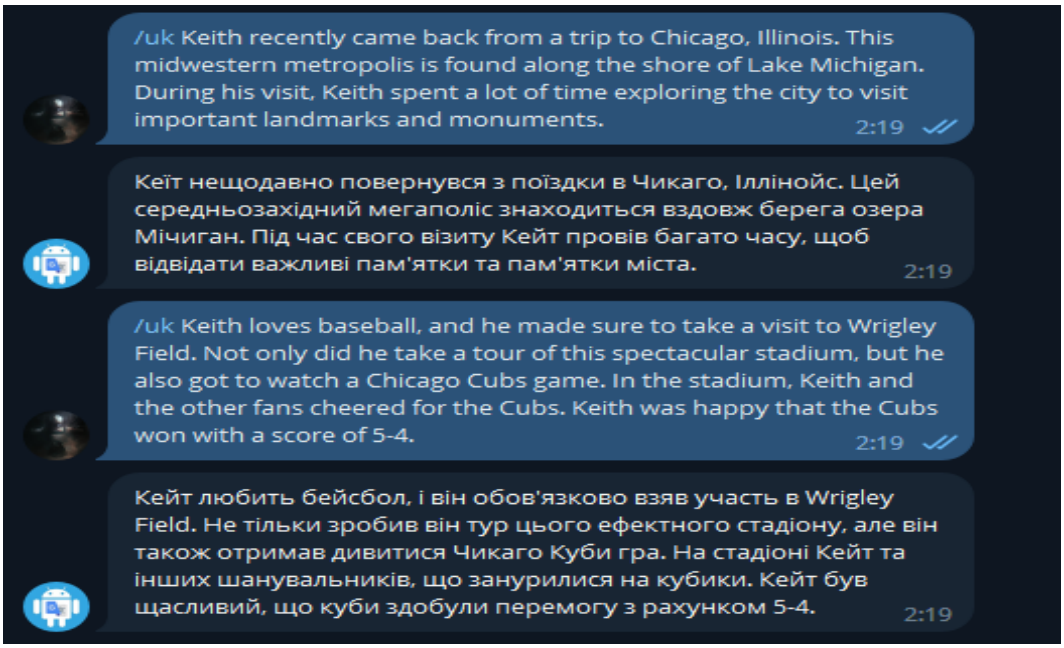

Рисунок 1.2 – приклад використання Google Translate Bot

Основні особливості Google Translate Bot:

– Переклад текстових повідомлень. Ви можете відправити боту текстове повідомлення на будь-якій мові, а він надішле вам переклад на мову, яку ви вкажете. Бот підтримує багатомовний переклад, що дозволяє перекладати тексти між багатьма мовами.

– Переклад зображень. Google Translate Bot має функціонал перекладу зображень. Ви можете надіслати зображення з текстом, і бот розпізнає текст на зображенні та надасть переклад на бажану мову.

– Переклад голосових повідомлень. Бот також підтримує переклад голосових повідомлень. Ви можете надіслати голосове повідомлення, а бот розпізнає його та надасть переклад у текстовій формі.

– Висока якість перекладу. Сервіс Google Translate відомий своєю високою якістю перекладу. Бот використовує потужні механізми машинного навчання та нейронних мереж для забезпечення точності та розуміння тексту.

– Взаємодія з іншими сервісами: Google Translate Bot може взаємодіяти з іншими сервісами або ботами, що дозволяє розширити його функціональні можливості. Наприклад, він може інтегруватись з сервісами для розпізнавання мови, перекладу спеціалізованих термінів або виконання інших додаткових операцій.

– Наявність додаткових налаштувань: Google Translate Bot надає користувачам можливість налаштувати деякі параметри перекладу, наприклад, вибрати специфічні мовні моделі або налаштувати формат виводу перекладу.

– Велика база мов. Google Translate Bot має широкий вибір підтримуваних мов. Ви можете перекладати текст на різних мовах світу, включаючи популярні мови та менш поширені мови.

1.3 Постановка задачі, формулювання функціональних вимог

Метою роботи є створення Телеграм-боту для автоматичного перекладу з однієї мови на іншу набраних текстових фрагментів, текстів на зображеннях, голосових повідомлень в чатах месенджера та листуванні.

Для досягнення поставленої мети треба вирішити наступні задачі:

1. Проведення аналізу предметної області, дослідження існуючих сервісів інтернет - перекладу та формулювання вимог до розроблюваного боту;

2. Вибір програмних засобів та середовища розробки;

3. Моделювання роботи боту;

4. Розроблення боту, інтеграція програмних пакетів, підключення ключів API;

5. Тестування боту.

Функціональні задачі для розроблюваного Телеграм-бота включають наступне:

– Реєстрація користувача. Бот повинен мати можливість реєстрації нових користувачів. Це може включати створення унікального ідентифікатора для кожного користувача та збереження їхньої особистої інформації, якщо це необхідно.

– Переклад тексту на іноземну мову. Бот повинен мати можливість приймати текстові повідомлення від користувача та перекладати їх на вибрану мову. Користувач повинен мати можливість вибрати мову перекладу і введення тексту, який потрібно перекласти. Результат перекладу повинен бути надісланий користувачеві.

– Переклад тексту з картинки. Бот повинен мати можливість обробляти зображення, які надсилаються користувачем, та використовувати розпізнавання тексту (OCR) для виділення тексту з картинки. Після цього текст повинен бути перекладений на вибрану мову та відправлений користувачеві.

– Переклад голосових повідомлень. Бот повинен мати можливість приймати голосові повідомлення від користувача та перекладати їх на вибрану

мову. Для цього можуть використовуватися технології розпізнавання та синтезу мови. Перекладений текст повинен бути надісланий користувачеві.

– Взаємодія з користувачем. Бот повинен надавати зрозумілі інструкції та відповіді на запитання користувачів. Він може мати функції, такі як автоматичне визначення мови тексту, відображення альтернативних перекладів або збереження історії перекладів для користувача.

– Обробка помилок. Бот повинен мати механізми для виявлення та обробки помилок під час перекладу, таких як нерозпізнаний текст на зображенні або нерозпізнавання мови голосового повідомлення. Він може повідомляти користувача про помилки та надавати додаткові інструкції для вирішення проблеми.

– Безпека. Бот повинен забезпечувати безпеку обробки та зберігання даних користувачів. Це може включати шифрування передачі даних, захист доступу до особистої інформації та забезпечення конфіденційності перекладених повідомлень.

– Масштабованість. Бот повинен мати можливість працювати з великою кількістю користувачів та ефективно обробляти багато запитів одночасно. Це може вимагати використання потоків, розподіленого обчислення або інших методів оптимізації продуктивності.

– Підтримка мовних моделей. Бот повинен мати можливість оновлювати та розширювати мовні моделі для поліпшення якості перекладу. Це може включати використання навчальних даних, моделей глибокого навчання або інших методів машинного навчання для покращення точності та розпізнавання мови.

– Підтримка та оновлення. Забезпечення підтримки та оновлень бота включає в себе відповідь на запити користувачів, вирішення проблем та недоліків, які можуть виникати під час використання бота. Крім того, оновлення бота можуть включати вдосконалення алгоритмів перекладу, впровадження нових технологій та функцій.

– Аналіз використання та зворотний зв'язок. Бот повинен мати механізми для аналізу використання, які дозволяють збирати дані про активність користувачів, їхні вподобання та поведінку. Це може допомогти у виявленні проблем, покращенні функціональності та наданні більш персоналізованого досвіду користувачам.

2 ПРОЕКТУВАННЯ ТЕЛЕГРАМ – БОТУ

Метою створення Телеграм-боту є надання можливості користувачам перекладати тексти, зображення та голосові повідомлення на іноземні мови. Бот повинен бути здатний ефективно та точно перекладати введений текст безпосередньо при спілкуванні месенджером, розпізнавати текст зображень, а також розпізнавати та перекладати голосові повідомлення.

Обґрунтування вибору методів дослідження та засобів реалізації:

1. Методи дослідження:

– Аналіз існуючих інтернет-сервісів перекладу, таких як Google Translate, для визначення їх можливостей, обмежень та особливостей.

– Вивчення технологій розпізнавання тексту (OCR) для обробки зображень та виділення текстової інформації з них.

– Дослідження технологій розпізнавання голосу для перетворення голосових повідомлень на текст.

– Оцінка швидкості, точності та якості перекладу різних сервісів та технологій.

2. Засоби реалізації:

– Використання мови програмування Python для реалізації логіки бота та взаємодії з іншими сервісами.

– Використання API Телеграм для комунікації з користувачами та обробки їхніх запитів.

– Інтеграція з зовнішніми сервісами перекладу, такими як Google Translate, для отримання точних та надійних перекладів.

– Використання бібліотек або сервісів для розпізнавання тексту зображень та голосових повідомлень, наприклад, Tesseract OCR для зображень і Google Cloud Speech-to-Text для голосу.

– Телеграм-бот буде забезпечувати можливість вводу тексту для перекладу користувачем.

– Для перекладу тексту буде використовуватися інтеграція зі зовнішнім сервісом перекладу Google Translate API, що забезпечить високу якість та точність перекладу.

– Для перекладу зображень бот буде використовувати технологію розпізнавання тексту Tesseract OCR. Вона дозволить отримати текстову інформацію з зображень та передати її на сервіс перекладу для отримання перекладу.

– Переклад голосових повідомлень буде здійснюватися за допомогою технології розпізнавання голосу Google Cloud Speech-to-Text. Бот розпізнає голосове повідомлення, перетворює його на текст та виконує переклад.

Загальна архітектура бота буде базуватися на використанні фреймворку або бібліотеки для роботи з API Телеграм, наприклад, python-telegram-bot. Це дозволить зручно обробляти вхідні повідомлення, реалізувати функціональність навігації та взаємодії з користувачем.

Також, можливі такі додаткові функції:

– Можливість вибору мови перекладу.

– Збереження історії перекладів для подальшого перегляду.

– Надання додаткової інформації про перекладене слово або фразу, таку як синоніми, вимова або приклади вживання.

Цей план дозволить реалізувати потужний та зручний Телеграм-бот для перекладу текстів іноземною мовою, незалежно від формату введення (текст, зображення або голосове повідомлення).

2.1 Структурно-функціональне моделювання процесу розробки боту

IDEF0 (Integrated DEFinition for Function Modeling) – це методологія моделювання функцій та процесів в рамках системного аналізу і проектування. Вона використовується для структуризації, опису та аналізу функцій системи, а також для визначення взаємозв'язків між цими функціями.

IDEF0 використовує графічну нотацію для представлення функцій, потоків даних, учасників та їх взаємозв'язків у системі. Основна мета IDEF0 – забезпечити зрозуміле та однозначне представлення функціональної структури системи та процесів, що в ній відбуваються (рис. 2.1).

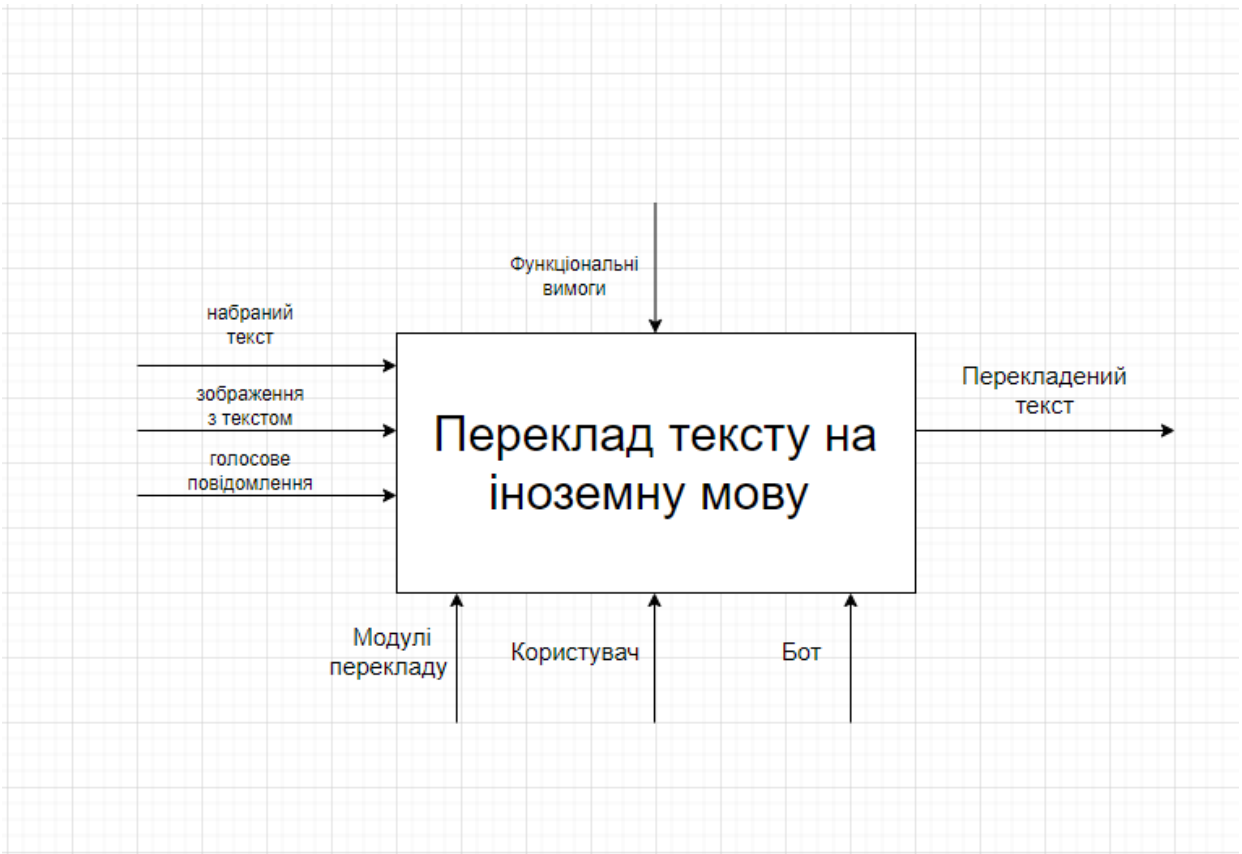

Рисунок 2.1 – Контекстна діаграма

Головні елементи IDEF0 включають:

– Функції (Functions): це дії або операції, які виконуються системою.

– Потоки даних (Data Flows): це інформація, яка пересувається між функціями та учасниками системи.

– Учасники (Participants): це ролі або агенти, які здійснюють функції та взаємодіють у системі.

– Керуючі відносини (Control Relations): це залежності та взаємозв'язки між функціями, що регулюють хід виконання процесу.

IDEF0 може бути застосований для аналізу та проектування різноманітних систем, включаючи інформаційні системи, процеси виробництва, управління проектами та бізнес-процеси. Використання IDEF0 допомагає зрозуміти функціональність системи, виявити недоліки та здійснити оптимізацію процесів.

На діаграмі декомпозиції показані різні функціональні модулі системи, які забезпечують функціональність перекладу тексту, зображень та голосових повідомлень. Користувач взаємодіє з Телеграм-ботом та вибирає опцію перекладу. Телеграм-бот отримує команду перекладу і передає її до відповідного модуля.

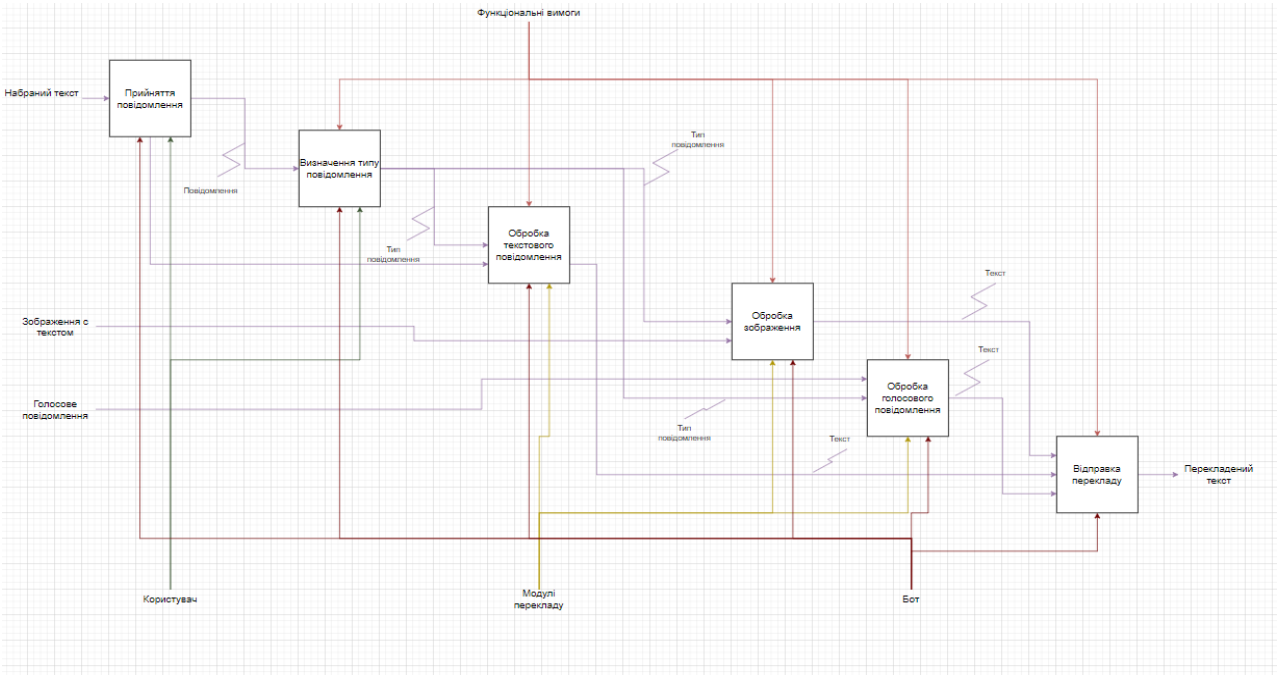

Рисунок 2.2 – Діаграма декомпозиції

Аналізуючи діаграму декомпозиції можна виділити такі підпроцеси:

Таблиця 2.1 – Дані для діаграми декомпозиції

Підпроцес	Вхід	Управління	Механізм	Вихід
Прийняття повідомлення	Набраний текст	Функціональні вимоги	Користувач, бот	Повідомлення
Визначення типу повідомлення	Повідомлення	Функціональні вимоги	Користувач, бот	Тип повідомлення
Обробка текстового повідомлення	Тип повідомлення	Функціональні вимоги	Модулі перекладу	Відправка перекладу
Обробка зображення	Тип повідомлення	Функціональні вимоги	Модулі перекладу, бот	Відправка перекладу
Обробка голосового повідомлення	Тип повідомлення	Функціональні вимоги	Модулі перекладу, бот	Відправка перекладу
Відправка перекладу	Тип повідомлення	Функціональні вимоги	Бот	Перекладений текст

2.2 Моделювання варіантів використання боту

При моделюванні варіантів використання боту для перекладу текстів іноземною мовою, можна розглядати наступні сценарії.

1. Варіант використання – переклад тексту за допомогою повідомлення:

– Користувач надсилає текстове повідомлення зі словом чи фразою, яку потрібно перекласти.

– Бот отримує повідомлення та виконує переклад за допомогою відповідного сервісу або алгоритму.

– Бот надсилає переклад користувачеві у відповідь.

2. Варіант використання – переклад тексту, який міститься на зображенні:

– Користувач надсилає зображення з текстом, який потрібно перекласти.

– Бот отримує зображення та виконує розпізнавання тексту на ньому.

– Бот отриманий текст передає на переклад за допомогою відповідного сервісу або алгоритму.

– Бот надсилає переклад користувачеві у відповідь.

3. Варіант використання – переклад тексту з голосового повідомлення:

– Користувач надсилає голосове повідомлення зі словом чи фразою, яку потрібно перекласти.

– Бот отримує голосове повідомлення та виконує його перетворення на текст за допомогою розпізнавання мови.

– Бот отриманий текст передає на переклад за допомогою відповідного сервісу або алгоритму.

– Бот надсилає переклад користувачеві у відповідь.

Таблиця 2.2 – Опис варіантів використання (рис. 2.3)

№	Назва	Опис
1.	Вибір головного меню	Користувач переходить до головного меню
2.	Вибір мови	Користувач обирає необхідну йому мову
3.	Введення тексту для перекладу	Вводить текст
4.	Відправка зображення для перекладу	Користувач відправляє зображення с текстом на ньому для перекладу
5.	Відправка голосового повідомлення для перекладу	Користувач відправляє голосове повідомлення для перекладу
6.	Перегляду запитів	Можливість переглядати запити ,які відправляв користувач

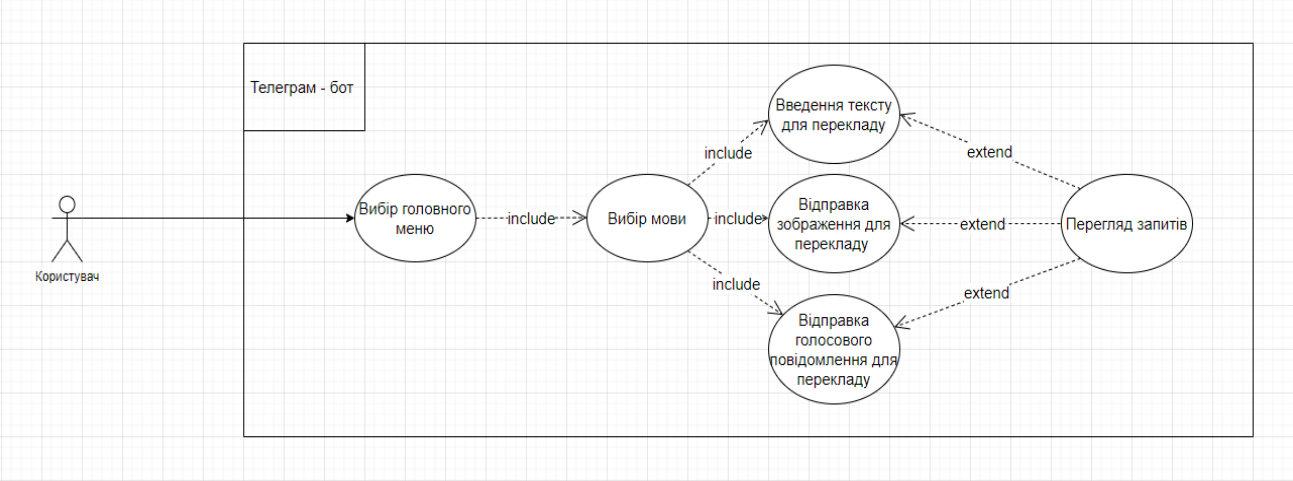

Рисунок 2.3 – Діаграма варіантів використання боту

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ – БОТУ

3.1 Архітектура боту

Архітектура Телеграм-боту для перекладу текстів іноземною мовою, за набраним текстом, за текстом з картинки та за голосовим повідомленням має бути реалізована наступним чином:

Клієнтський інтерфейс:

– Телеграм-клієнт, через який користувачі взаємодіють з ботом.

– Користувачі можуть надсилати текстові повідомлення, зображення або голосові повідомлення.

Телеграм-бот:

– Модуль, що приймає повідомлення від користувачів через API Телеграм.

– Аналізує тип отриманого повідомлення (текст, зображення, голосове повідомлення).

– Направляє отримані дані до відповідних модулів для обробки.

Модуль перекладу тексту:

– Отримує текстове повідомлення з бота.

– Виконує переклад тексту за допомогою відповідного сервісу або алгоритму перекладу.

– Повертає перекладений текст.

– Модуль розпізнавання тексту на зображенні:

– Отримує зображення з бота.

– Виконує розпізнавання тексту на зображенні за допомогою відповідного алгоритму або сервісу розпізнавання.

– Повертає отриманий текст для подальшого перекладу.

Модуль розпізнавання мови у голосових повідомленнях:

– Отримує голосове повідомлення з бота.

– Виконує розпізнавання мови у голосовому повідомленні за допомогою відповідного сервісу або алгоритму розпізнавання мови.

– Повертає розпізнану мову для подальшого перекладу.

Модуль перекладу:

– Отримує дані для перекладу (текст, розпізнаний текст з зображення, розпізнана мова у голосовому повідомлення).

– Виконує переклад отриманих даних на вибрану мову за допомогою відповідного сервісу або алгоритму перекладу.

– Повертає перекладані дані користувачу через Телеграм-бота.

Модуль взаємодії з користувачем:

– Приймає перекладані дані від модуля перекладу.

– Формує відповідь для користувача, яка містить перекладаний текст, повідомлення про помилки або іншу інформацію.

– Відправляє відповідь користувачу через Телеграм-бота.

Зовнішні сервіси та API:

– Модуль використовує зовнішні сервіси або API для здійснення перекладу, розпізнавання тексту на зображенні або розпізнавання мови у голосових повідомленнях.

– Отримує голосове повідомлення з бота.

– Виконує розпізнавання мови у голосовому повідомленні за допомогою відповідного сервісу або алгоритму розпізнавання мови.

– Повертає розпізнану мову для подальшого перекладу.

Загальна структура боту передбачає отримання повідомлень від користувачів через Телеграм-клієнт, подальшу обробку отриманих даних за допомогою відповідних модулів перекладу, розпізнавання та взаємодії з користувачем, а також використання зовнішніх сервісів для виконання специфічних завдань. Можна скористатись MVC архітектурою та використати її логіку для побудови бота.

MVC (Model-View-Controller) архітектура є популярним підходом до розробки програмного забезпечення, включаючи і Телеграм-ботів. Розглянемо, як вона може бути застосована в контексті Телеграм-бота для перекладу текстів іноземною мовою.

1. Модель (Model). Модель представляє логіку та дані системи. У випадку Телеграм-бота для перекладу, модель може включати компоненти, які відповідають за переклад текстів, обробку картинок та голосових повідомлень. Ці компоненти взаємодіють зі зовнішніми сервісами перекладу, аналізу зображень та розпізнавання голосу, щоб забезпечити потрібний функціонал.

2. Представлення (View). Представлення відповідає за візуальне відображення інтерфейсу користувача. В контексті Телеграм-бота, представлення може бути реалізоване за допомогою шаблонів повідомлень або веб-сторінок, які дозволяють користувачу взаємодіяти з ботом. Наприклад, користувач може вводити текстові повідомлення, надсилати картинки або голосові повідомлення через інтерфейс Телеграм-бота.

3. Контролер (Controller). Контролер відповідає за обробку вхідних даних від користувача та керування виконанням відповідних дій. У випадку Телеграм-бота, контролер може слухати вхідні повідомлення від користувачів, визначати їх тип (текст, картинка, голосове повідомлення) і відправляти відповідний запит до моделі для обробки та перекладу.

MVC архітектура дозволяє розділити логіку, представлення та управління в системі, що полегшує розробку та розширення Телеграм-бота для перекладу текстів іноземною мовою. Основні переваги MVC архітектури такі:

– Розділення відповідальностей – кожен компонент (модель, представлення, контролер) виконує свої функції, що дозволяє легко управляти логікою, інтерфейсом та обробкою даних.

– Модульність – компоненти можуть бути розроблені та модифіковані незалежно один від одного, що спрощує підтримку та розширення системи.

– Повторне використання – компоненти можуть бути повторно використані в інших проектах або контекстах, що забезпечує ефективне використання ресурсів та прискорює розробку.

– Тестування – кожен компонент може бути легко тестований окремо, що полегшує виявлення та виправлення помилок.

Застосування MVC архітектури в Телеграм-боті для перекладу текстів іноземною мовою дозволяє ефективно управляти логікою роботи перекладу, взаємодії з користувачами та обробкою різних типів повідомлень. Кожен компонент виконує свою роль, що сприяє кращій структуризації та підтримці проекту.

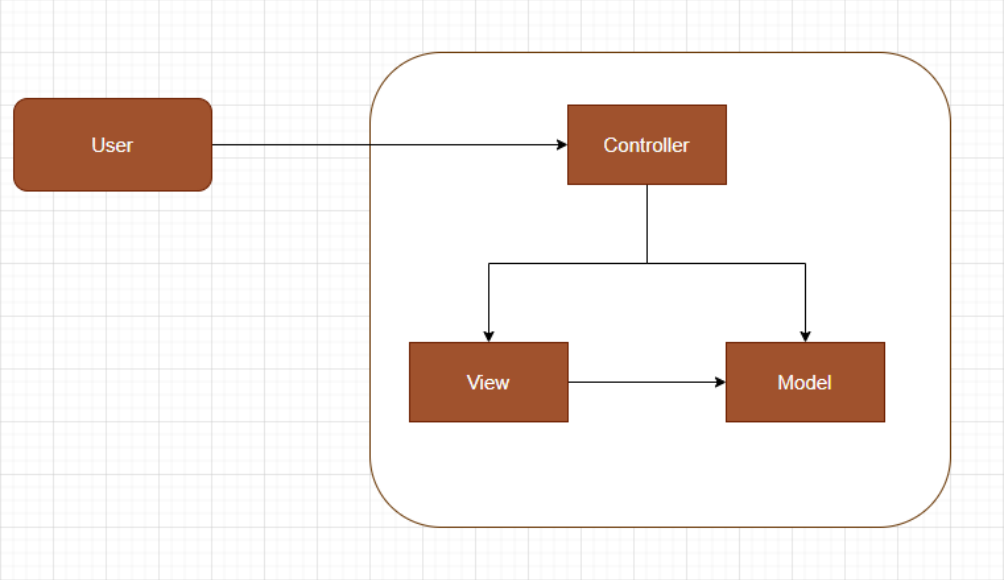

Рисунок 3.1 – Схема архітектури MVC

Загальна структура MVC діаграми для Телеграм-бота виглядає так:

1. Модель (Model):

– Відповідає за обробку та збереження даних, пов'язаних з перекладами.

– Може містити класи та функції, що виконують операції з перекладами, доступ до бази даних або зовнішніх сервісів перекладу.

2. Представлення (View):

– Відповідає за відображення інтерфейсу користувача та взаємодію з ним.

– Може включати шаблони для відображення текстових повідомлень, зображень та інших елементів інтерфейсу.

3. Контролер (Controller):

– Приймає вхідні дані від користувача та відповідає за управління взаємодією між моделлю та представленням.

– Обробляє вхідні дані, виконує відповідні операції перекладу та оновлює відображення для користувача.

4. Користувач (User):

– Взаємодіє з ботом через інтерфейс Телеграму.

– Надсилає текстові повідомлення, картинки або голосові повідомлення для перекладу.

3.2 Реалізація боту

Python є однією з найпопулярніших мов програмування для розробки ботів, особливо для месенджерів, таких як Телеграм.

Простота використання: Python має простий і зрозумілий синтаксис, що робить його доступним для новачків у програмуванні. Це дозволяє швидко розробляти ботів і зосередитися на функціональності, а не на складних деталях мови програмування.

Python має широкий вибір сторонніх бібліотек і модулів, які полегшують розробку ботів. Наприклад, для роботи з месенджером Телеграм існує популярна бібліотека Telebot, яка надає зручний інтерфейс для взаємодії з API Телеграм. Якщо поділити всю її структуру на блоки, отримаємо:

1. Налаштування середовища розробки:

– Встановлення Python та необхідних пакетів для роботи з API Телеграм та перекладом текстів.

– Створення власного бота в Телеграм та отримання API-ключа.

2. Розробка модулів та функціональності:

– Реалізація модуля для отримання та обробки текстових повідомлень від користувачів.

– Реалізація модуля для отримання та обробки зображень та голосових повідомлень.

– Інтеграція з сервісами перекладу, наприклад, Google Translate API або іншими доступними сервісами.

– Реалізація функцій перекладу тексту, зображень та голосових повідомлень.

– Обробка помилок та надання відповідних повідомлень користувачу.

3. Налаштування комунікації з Телеграмом:

– Встановлення зв'язку з API Телеграм для обробки вхідних та вихідних повідомлень.

– Налаштування обробки вхідних повідомлень, викликів та подій від Телеграму.

4. Тестування та налагодження:

– Виконання різноманітних тестів для перевірки правильності роботи боту.

– Виявлення та виправлення помилок та недоліків.

– Впровадження засобів логування для відстеження дії боту та виявлення проблем.

5. Розгортання боту – підготовка середовища для запуску боту.

Список використаних бібліотек:

– Для перекладу тексту використовуємо бібліотеку mtranslate. Бібліотека Telebot необхідна для створення та керування ботом на платформі Телеграм. Код використовує клас TeleBot для створення бота і обробки повідомлень.

– Бібліотека PIL (Python Imaging Library) надає функціонал для роботи з зображеннями (рис. 3.2). У коді from PIL import Image використовується для імпорту класу Image з бібліотеки PIL. Цей клас дозволяє мені створювати об'єкти зображень та виконувати різноманітні операції з ними. Наприклад, я використовую цей клас для відкриття та обробки зображень у функції

handle_photo, де за допомогою Image.open завантажує зображення з файлу та подальше виконує операції розпізнавання тексту на зображенні за допомогою іншої бібліотеки pytesseract.

```
# распознавание текста на фото
img = Image.open(filename)
text = pytesseract.image_to_string(img)
```

Рисунок 3.2 – Приклад використання бібліотеки PIL

– pytesseract – бібліотека є обгорткою навколо програми розпізнавання тексту Tesseract OCR (рис. 3.3). Код використовує pytesseract для розпізнавання тексту на зображеннях. Приміром використання pytesseract є функція image_to_string, яку використовую для отримання тексту з зображення у функції handle_photo. Ця функція приймає зображення як вхідний параметр та повертає розпізнаний текст. Перед використанням pytesseract потрібно налаштувати шлях до виконуваного файлу Tesseract OCR за допомогою pytesseract.pytesseract.tesseract_cmd. Встановлюю цей шлях у змінну tesseract_path у коді.

```
tesseract_path = os.path.join(current_dir, r'Tesseract-OCR\tesseract.exe')
pytesseract.pytesseract.tesseract_cmd = tesseract_path
```

Рисунок 3.3 – Приклад використання бібліотеки pytesseract

– uuid – це модуль в стандартній бібліотеці Python, який надає можливості роботи з унікальними ідентифікаторами (UUID). UUID – це 128-бітне число, яке гарантовано унікальне в межах всього світу. У коді import uuid використовується для імпорту модулю uuid. Цей модуль надає різноманітні функції для генерації, представлення та роботи з UUID. Прикладом використання uuid є виклик функції uuid.uuid4(), який генерує новий випадковий UUID типу 4 (рис. 3.4). Використовую цю функцію для створення

унікального ідентифікатора для назви файлу, коли зберігаю завантажене фото в функції handle_photo.

```
# сохранение фото
filename = str(uuid.uuid4()) + '.jpg'
with open(filename, 'wb') as f:
    f.write(file.content)
```

Рисунок 3.4 – Приклад використання модулю uuid

– speech_recognition – це модуль в Python, який надає можливості розпізнавання мовлення (speech recognition). Він залежить від зовнішніх бібліотек, таких як Google Speech Recognition, Sphinx і Microsoft Bing Voice Recognition, і дозволяє легко інтегрувати функціонал розпізнавання мовлення в свої програми (рис. 3.5). Цей модуль дозволяє отримувати аудіо дані з різних джерел, таких як мікрофон або аудіофайл, і використовувати один із розпізнавачів мовлення для отримання текстового представлення розпізнаного мовлення. Приміром використання speech_recognition є виклик функції sr.Recognizer() для створення об'єкта розпізнавача мовлення, а також використання методів цього об'єкта, наприклад recognize_google(), для розпізнавання мовлення з використанням сервісу Google Speech Recognition.

```
# Распознавание речи в голосовом сообщении
r = sr.Recognizer()
with sr.AudioFile('converted_audio.wav') as source:
    audio = r.record(source)
text = r.recognize_google(audio, language='en-US')
```

Рисунок 3.5 – Приклад використання модулю speech_recognition

– requests є модулем в Python, який дозволяє виконувати HTTP-запити (наприклад, отримувати дані з веб-сторінок або взаємодіяти з API) шляхом надсилання HTTP-запитів до сервера і отримання відповідей. Він є одним з найпопулярніших модулів для роботи з мережевими запитами в Python. Модуль

requests надає простий та зрозумілий інтерфейс для виконання різних типів запитів, таких як GET, POST, PUT, DELETE і багатьох інших. Він також підтримує роботу з різними параметрами запитів, заголовками, файлами та іншими атрибутами HTTP-протоколу. В коді вона використовується для завантаження файлів з API Телеграм (рис. 3.6).

```
# Скачивание голосового сообщения
file_info = bot.get_file(message.voice.file_id)
file = requests.get('https://api.telegram.org/file/bot{0}/{1}'.format(bot.token, file_info.file_path))
```

Рисунок 3.6 – Приклад використання модулю requests

– os є вбудованим модулем в Python, який надає функції для взаємодії з операційною системою. Цей модуль дозволяє виконувати різноманітні операції пов'язані з файловою системою, оточенням та процесами. Основні функції та можливості модуля os включають:

= Робота з файловою системою: створення, перейменування та видалення файлів та каталогів, отримання інформації про файли (розмір, час модифікації тощо), навігація по структурі каталогів.
= Взаємодія з операційною системою: запуск зовнішніх програм, отримання інформації про оточення (змінні середовища, поточна робоча директорія тощо), виконання команд терміналу.
= Робота з процесами: створення та керування процесами, отримання інформації про процеси (ідентифікатор процесу, статус, час виконання тощо).
= Робота зі шляхами файлів: об'єднання шляхів, розбиття шляху на складові частини, перевірка наявності файлу або каталогу.
= Маніпуляції зі строками шляхів: нормалізація, розширення, скорочення, перетворення стилів шляхів. Модуль os дозволяє створювати переносимий код, оскільки він надає абстракцію над операційною системою, дозволяючи виконувати різні операції незалежно від

платформи, на якій працює програма. Він є потужним інструментом для взаємодії з операційною системою, файловою системою та процесами.

– pydub – ця бібліотека надає функціонал для роботи з аудіофайлами. Код використовує її для конвертації голосового повідомлення з формату OGG в WAV перед подальшим розпізнаванням мови. from pydub import AudioSegment – цей рядок коду імпортує клас AudioSegment з бібліотеки pydub. Клас AudioSegment є основним класом у бібліотеці pydub і використовується для представлення аудіофайлів і виконання різних операцій з ними.

Далі розглянуто реалізацію програми крок за кроком.

Імпортовано необхідні бібліотеки:

```
import mtranslate
import telebot
from PIL import Image
import pytesseract
import uuid
from mtranslate import translate
import speech_recognition as sr
import requests
import os
from pydub import AudioSegment
```

Рисунок 3.7 – Імпорт необхідних бібліотек

Створення бота з використанням власного токену:

```
# создание бота
bot = telebot.TeleBot('6117148001:AAGx__Kg-hUMZ50NKF8YrkGykF62Jsx8NKQ')
```

Рисунок 3.8 – Власний токен

Щоб отримати власний API використаний BotFather. Це бот, який використовується для створення та керування ботами в Телеграм. Він дозволяє користувачам створювати нові боти в Телеграм. При взаємодії з BotFather можна надати деталі про свого бота, такі як його назву, ім'я користувача та опис. Після створення бота BotFather надає API-токен, який можна використовувати для керування ботом.

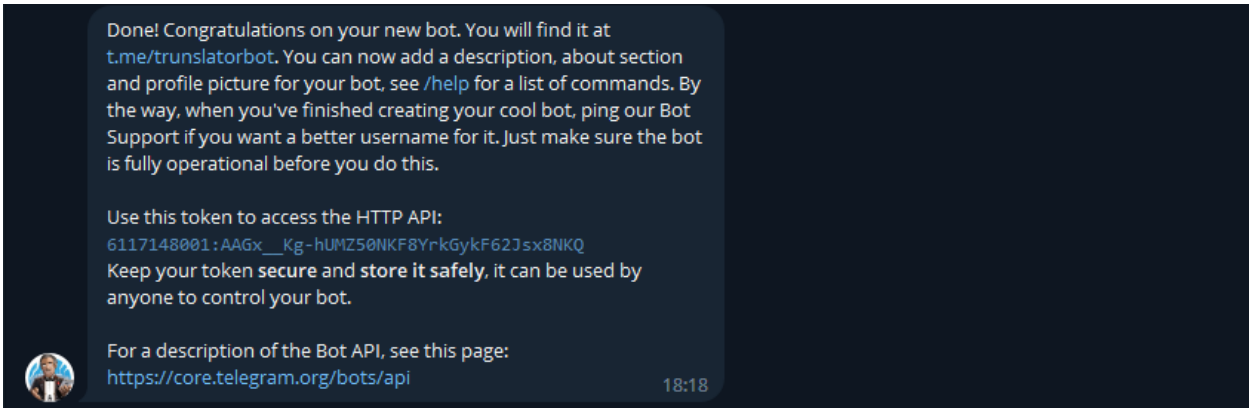

Рисунок 3.9 – Отримано власний токен

Ініціалізація OCR-движка Tesseract шляхом вказання шляху до виконуваного файлу в Python дозволяє використовувати Tesseract OCR для розпізнавання тексту у зображеннях або сканованих документах безпосередньо з вашої програми на Python (рис. 3.10). Використання Tesseract OCR в Python дозволяє автоматизувати процес розпізнавання тексту та отримувати результати у текстовому форматі.

```
# инициализация OCR-движка
current_dir = os.path.dirname(os.path.abspath(__file__))

tesseract_path = os.path.join(current_dir, r'Tesseract-OCR\tesseract.exe')
pytesseract.pytesseract.tesseract_cmd = tesseract_path
```

Рисунок 3.10 – Ініціалізація OCR Tesseract

Створено обробник повідомлень з фотографіями. @bot.message_handler(content_types=['photo']) – це декоратор, який вказує, що функція handle_photo буде викликатися тільки при отриманні повідомлення з фото.

Скачування фото. file_info = bot.get_file(message.photo[-1].file_id) – отримує інформацію про файл з фото, яке надіслано. file = requests.get('https://api.Телеграм.org/file/bot{0}/{1}'.format(bot.token, file_info.file_path)) – отримує фактичне фото, використовуючи URL, отриманий з file_info.

Збереження фото. filename = str(uuid.uuid4()) + '.jpg' – генерує унікальне ім'я для збереження фото. with open(filename, 'wb') as f: – відкриває файл для запису фото. f.write(file.content) – записує вміст фото у файл.

Розпізнавання тексту на фото. img = Image.open(filename) – відкриває збережене фото за допомогою PIL (Python Imaging Library). text = pytesseract.image_to_string(img) – використовує pytesseract для розпізнавання тексту на фото.

Переклад тексту на українську мову. translation = translate(text, 'uk') – передбачається, що є функція translate, яка виконує переклад тексту на українську мову. Замість цього коду треба мати власну функцію перекладу або використовувати відповідний сервіс перекладу.

Відправка перекладеного тексту. bot.send_message(message.chat.id, translation) – відправляє перекладений текст у чат, з якого отримано фото.

Видалення фото. os.remove(filename) – видаляє збережене фото після того, як воно було оброблене та надіслане. Це допомагає звільнити простір на диску і забезпечує чистоту обробки для кожного нового фото.

```python
# обработчик сообщений с изображениями
@bot.message_handler(content_types=['photo'])
def handle_photo(message):
    # скачивание фото
    file_info = bot.get_file(message.photo[-1].file_id)
    file = requests.get('https://api.telegram.org/file/bot{0}/{1}'.format(bot.token, file_info.file_path))

    # сохранение фото
    filename = str(uuid.uuid4()) + '.jpg'
    with open(filename, 'wb') as f:
        f.write(file.content)

    # распознавание текста на фото
    img = Image.open(filename)
    text = pytesseract.image_to_string(img)

    # перевод текста на украинский язык
    translation = translate(text, 'uk')

    # отправка переведенного текста
    bot.send_message(message.chat.id, translation)

    # удаление фото
    os.remove(filename)
```

Рисунок 3.11 – Блок оброблення повідомлень з зображеннями

Створено обробник повідомлень з текстом. @bot.message_handler(content_types=['text']) – цей декоратор вказує, що функція user_text буде викликатися тільки при отриманні текстових повідомлень.

Збереження тексту. text_1 = message.text – отримує текст повідомлення, яке надіслано від користувача.

Переклад тексту на українську. transtext = translate(text_1, 'uk') – передбачається, що є функція translate, яка виконує переклад тексту на українську мову. Замість цього коду ви повинні мати власну функцію перекладу або використовувати відповідний сервіс перекладу.

Відправка перекладеного тексту. bot.send_message(message.chat.id, transtext) – відправляє перекладений текст у чат, з якого отримано повідомлення.

```
# обработка сообщений с текстом
@bot.message_handler(content_types=['text'])
def user_text(message):
    #сохранение текста
    text_1 = message.text
    #перевод текста на украинский
    transtext = translate(text_1, 'uk')
    #отправка переведенного текста
    bot.send_message(message.chat.id, transtext)
```

Рисунок 3.12 – Блок оброблення повідомлень з текстом

@bot.message_handler(content_types=['voice']) – цей декоратор вказує, що функція voice_handler буде викликатися тільки при отриманні голосових повідомлень.

Скачування голосового повідомлення. file_info = bot.get_file(message.voice.file_id) – отримує інформацію про файл з голосовим повідомленням. file = requests.get('https://api.Телеграм.org/file/bot{0}/{1}'.format(bot.token, file_info.file_path)) – отримує фактичне голосове повідомлення, використовуючи URL, отриманий з file_info. with open('voice.ogg', 'wb') as f: – відкриває файл для запису голосового повідомлення. f.write(file.content) – записуємо вміст голосового повідомлення у файл.

Конвертування в необхідний формат. audio = AudioSegment.from_file('voice.ogg', format='ogg') – використовуючи бібліотеку

pydub, ми завантажує голосовий файл та конвертує його у формат WAV. audio.export('converted_audio.wav', format='wav') – зберігає конвертований аудіофайл у форматі WAV.

Розпізнавання мовлення в голосовому повідомленні. text = r.recognize_google(audio, language='en-US') – використовуючи recognize_google з бібліотеки SpeechRecognition, розпізнає мовлення в голосовому повідомленні і отримуємо текстовий результат.

Переклад тексту з англійської на українську. translated_text = mtranslate.translate(text, 'uk', 'en') – передбачається, що є функція translate, яка виконує переклад тексту з англійської на українську. Замість цього коду треба мати власну функцію перекладу або використовувати відповідний сервіс перекладу.

Відправка перекладеного тексту користувачу:. bot.reply_to(message, translated_text) – відповідає користувачеві з перекладеним текстом, використовуючи функцію bot.reply_to.

```python
# обработка сообщений с голосовым сообщением
@bot.message_handler(content_types=['voice'])
def voice_handler(message):
    try:
        # Скачивание голосового сообщения
        file_info = bot.get_file(message.voice.file_id)
        file = requests.get('https://api.telegram.org/file/bot{0}/{1}'.format(bot.token, file_info.file_path))

        print('Response: ', file)
        with open('voice.ogg', 'wb') as f:
            f.write(file.content)
        #конвертирование в нужный формат wav
        audio = AudioSegment.from_file('voice.ogg', format='ogg')
        audio.export('converted_audio.wav', format='wav')

        # Распознавание речи в голосовом сообщении
        r = sr.Recognizer()
        with sr.AudioFile('converted_audio.wav') as source:
            audio = r.record(source)
        text = r.recognize_google(audio, language='en-US')

        print('audio: ', r, text)

        # Перевод текста с английского на украинский язык
        translated_text = mtranslate.translate(text, 'uk', 'en')

        # Отправка переведенного текста пользователю
        bot.reply_to(message, translated_text)

    except Exception as e:
        print('Error!', e)
```

Рисунок 3.13 – Блок оброблення голосових повідомлень

Для початку отримання та обробки повідомлень від користувачів необхідно викликати функцію bot.polling(none_stop=True).

```
bot.polling(none_stop=True)
```

Рисунок 3.14 – Обробка повідомлень від користувачів

Приклади використання та результат роботи

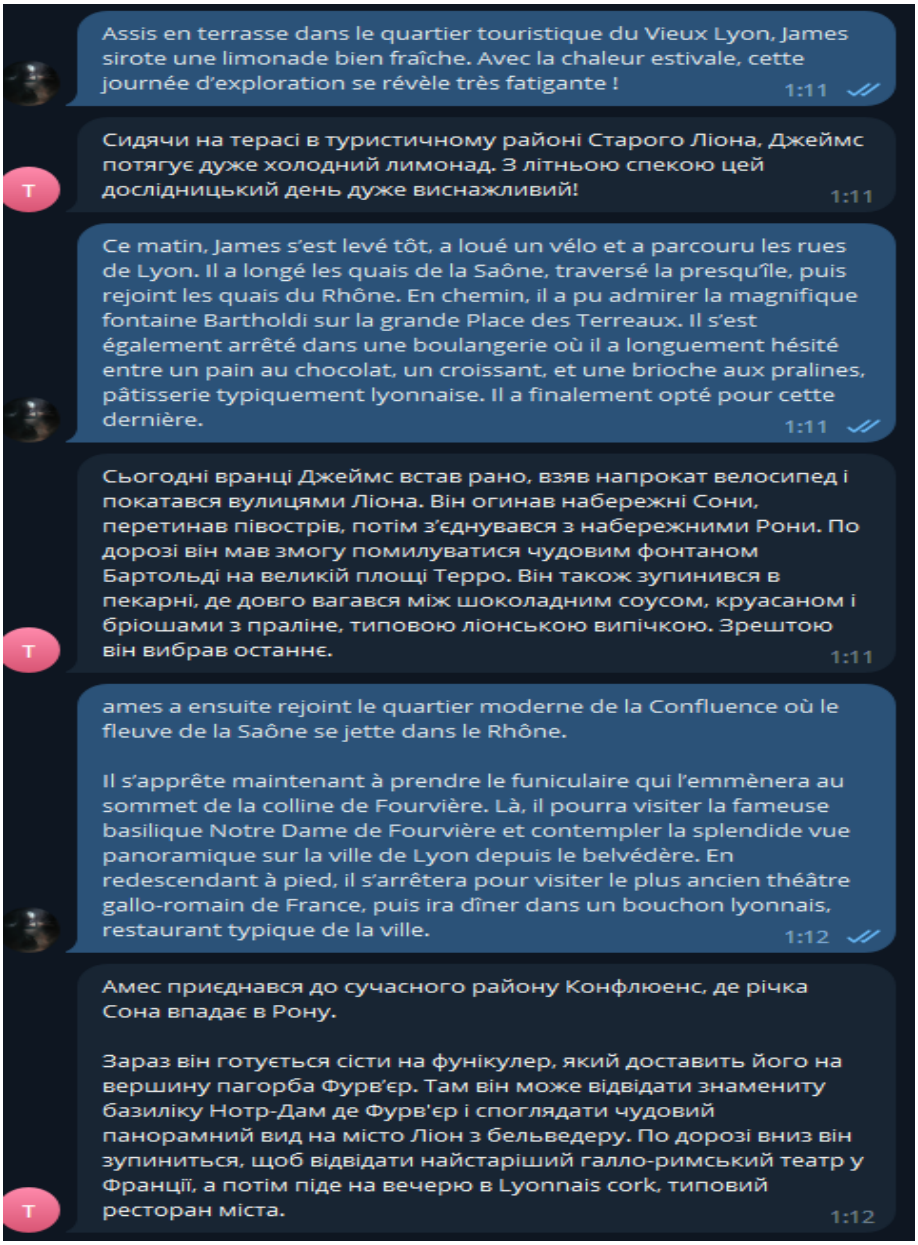

Рисунок 3.15 – Приклад перекладу звичайного тексту.

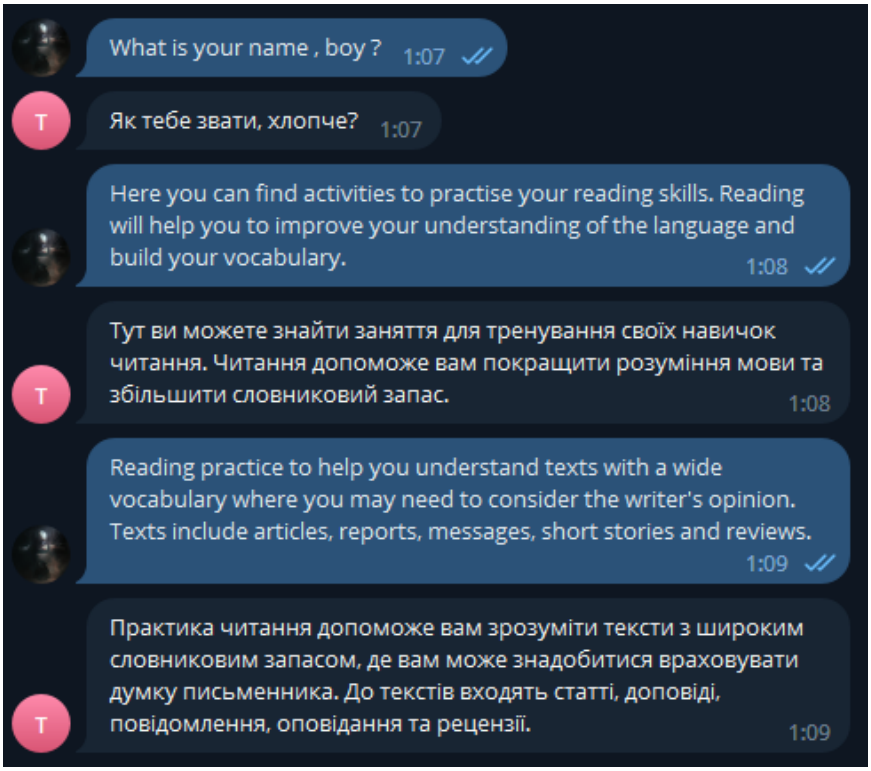

Рисунок 3.16 – Приклад перекладу звичайного тексту.

Функція перекладу тексту с зображення або картинки (рис. 3.17 – 3.18).

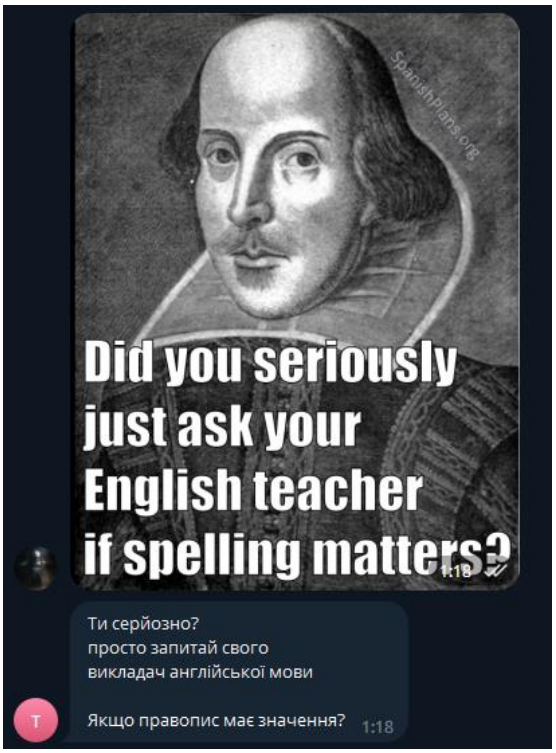

Рисунок 3.17 – Переклад тексту с зображення.

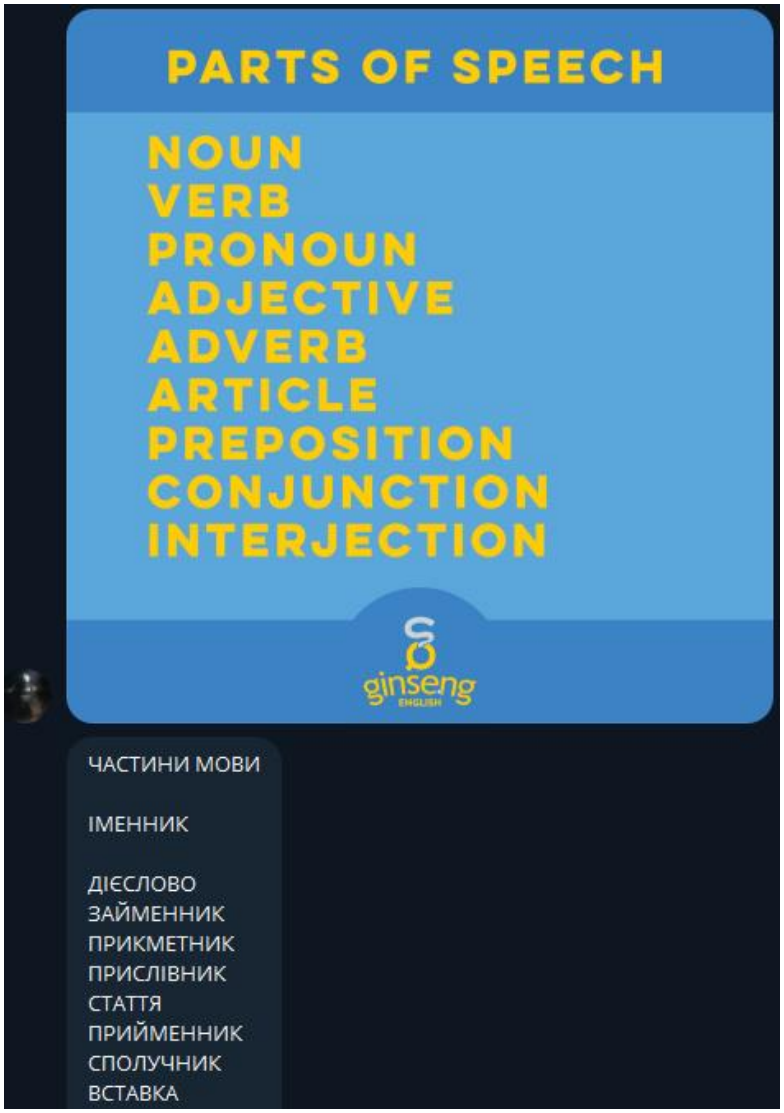

Рисунок 3.18 – Переклад тексту с зображення.

Функція розпізнавання та перекладу голосового повідомлення (рис. 3.19).

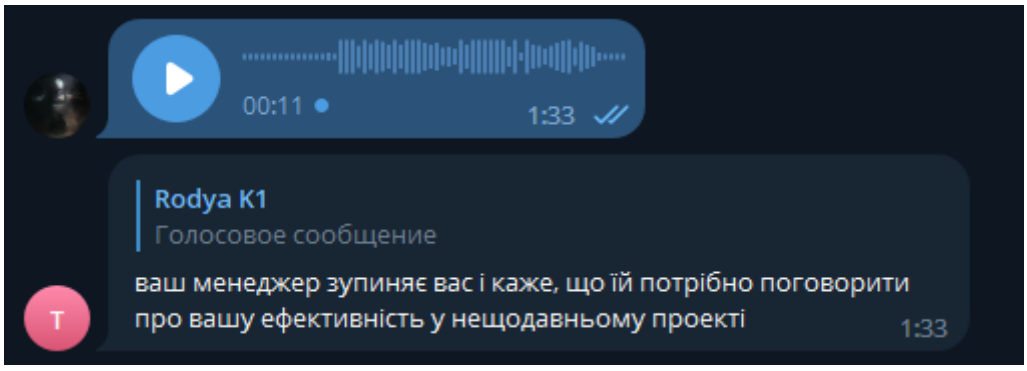

Рисунок 3.19 – Переклад голосового повідомлення.

За допомогою sr.Recognizer() можна створити об'єкт, який здатен слухати аудіо з мікрофону та розпізнавати мовлення. За допомогою методу recognize_speech_from_mic() можна отримати розпізнаний текст з мовлення. Голосове повідомлення також відображається у самому терміналі, де його можна прослухати та конвертований текст.

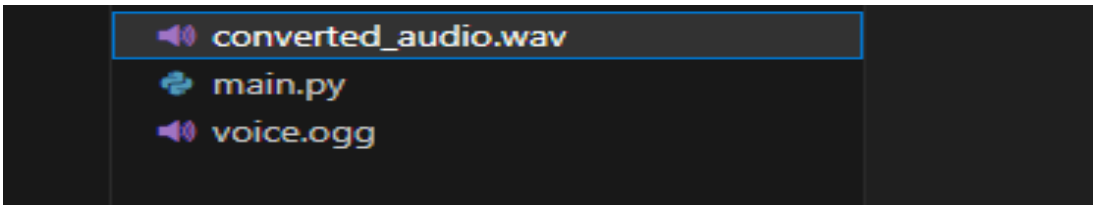

Рисунок 3.20 – Голосове повідомлення.

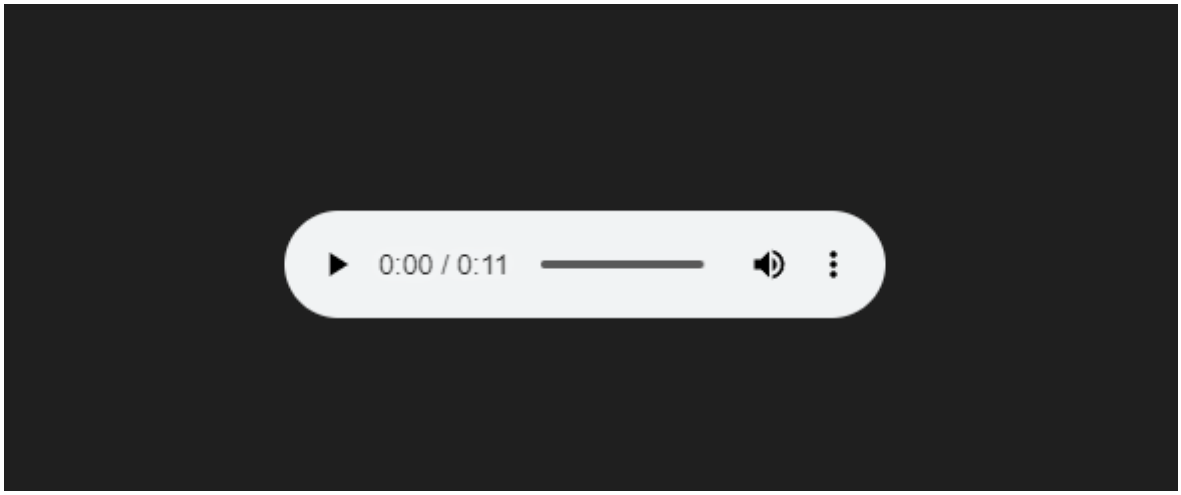

Рисунок 3.21 – Голосове повідомлення в терміналі.

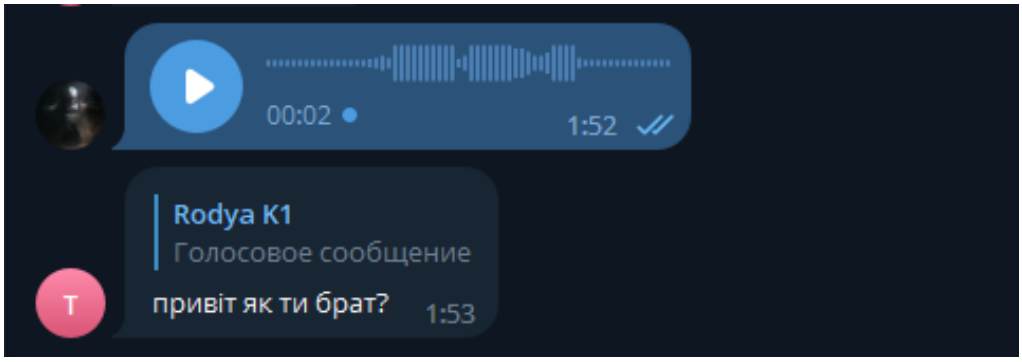

Рисунок 3.22 – Голосове повідомлення.

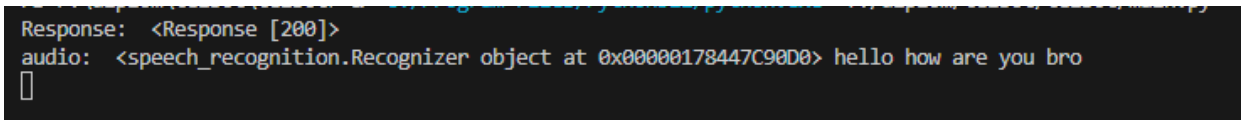

Рисунок 3.23 – Розпізнане голосове повідомлення.

ВИСНОВКИ

Аналіз предметної області та поширених інтернет-сервісів перекладу показав, що даний програмний продукт є затребуваний в області ділового та навчального спілкування та доступу до інформації різними мовами.

Моделювання роботи бота за допомогою діаграм дозволило чітко відобразити взаємодію з користувачем, модулів перекладу тексту, обробки зображень та обробки голосових повідомлень. Це спростило процес розробки та розуміння логіки системи.

Вдалий вибір програмного забезпечення та середовища розробки, таких як Python, бібліотека python-telegram-bot та інші відповідні інструменти, дозволив швидко реалізувати модель бота. Python забезпечує зручну синтаксичну структуру та багатий набір бібліотек для роботи з текстом, зображеннями та звуком.

Розроблений бот використовує API Телеграм для взаємодії з користувачами та інтегровані сервіси перекладу текстів, обробки зображень та перекладу голосових повідомлень. Це дозволяє користувачам зручно використовувати бота для отримання перекладів різноманітних типів контенту.

Тестування бота дозволило виявити високу якість та швидкість перекладу на різні мови. Бот успішно перекладає текстові повідомлення, розпізнає та перекладає тексти з зображень, а також здатен обробляти та перекладати голосові повідомлення. Тести підтвердили функціональність та ефективність бота у реальних умовах використання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. How to Translate Languages in Python [Електронний ресурс] – Режим доступу до ресурсу: https://www.thepythoncode.com/article/translate-text-in-python (дата звернення: 03.06.2023).

2. ffmpeg-python [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/kkroening/ffmpeg-python (дата звернення: 03.06.2023).

3. PhoenixNAP [Електронний ресурс] – Режим доступу до ресурсу: https://phoenixnap.com/kb/ffmpeg-windows (дата звернення: 26.05.2023).

4. Design Patterns – MVC Pattern [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm (дата звернення: 06.05.2023).

5. MVC Framework Tutorial for Beginners: What is, Architecture & Example [Електронний ресурс] – Режим доступу до ресурсу: https://www.guru99.com/mvc-tutorial.html (дата звернення: 06.05.2023).

6. Installing Packages [Електронний ресурс] – Режим доступу до ресурсу: https://packaging.python.org/en/latest/tutorials/installing-packages/%20 (дата звернення: 15.05.2023).

7. Simplilearn – Online Certification Training Course Provider [Електронний ресурс] – Режим доступу до ресурсу: https://www.simplilearn.com/tutorials/python-tutorial/speech-recognition-in-python (дата звернення: 20.05.2023).

8. Python [Електронний ресурс] – Режим доступу до ресурсу: https://www.python.org/downloads/ (дата звернення: 05.05.2023).

9. Speech Recognition examples with Python [Електронний ресурс] – Режим доступу до ресурсу: https://pythonprogramminglanguage.com/speech-recognition/ (дата звернення: 15.05.2023).

10. Image Module [Електронний ресурс] – Режим доступу до ресурсу: https://pillow.readthedocs.io/en/stable/reference/Image.html (дата звернення: 19.05.2023).

11. Python PIL [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/python-pil-image-open-method/ (дата звернення: 24.05.2023).

12. Working with way files [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/working-with-wav-files-in-python-using-pydub/ (дата звернення: 21.05.2023).

13. Os Module [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/os-module-python-examples/ (дата звернення: 17.05.2023).

14. A simple python Implemenation [Електронний ресурс] – Режим доступу до ресурсу: https://morioh.com/p/e52ea3fd1198 (дата звернення: 17.05.2023).

15. Telegram APIs [Електронний ресурс] – Режим доступу до ресурсу: https://core.telegram.org/ (дата звернення: 10.05.2023).

16. How To Get Started With the Requests Library in Python [Електронний ресурс] – Режим доступу до ресурсу: https://www.digitalocean.com/community/tutorials/how-to-get-started-with-the-requests-library-in-python (дата звернення: 15.05.2023).

17. deepl-telegram [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/wkpn/deepl-telegram (дата звернення: 15.05.2023).

18. Telegram bot future [Електронний ресурс] – Режим доступу до ресурсу: https://core.telegram.org/bots/features (дата звернення: 10.05.2023).

19. What is telegram bot [Електронний ресурс] – Режим доступу до ресурсу: https://www.opc-router.com/what-is-a-telegram-bot/ (дата звернення: 10.05.2023).

20. Python Requests Module [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/python/module_requests.asp (дата звернення: 25.05.2023).

ДОДАТОК А. Технічне завдання

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку інформаційної системи
«Телеграм-бот для перекладу текстів іноземною мовою»

ПОГОДЖЕНО:

Старший викладач кафедри «Інформаційні технології»

_____ Кузнєцов Е. Г.

Студент групи ІТ-92

_____ Чабаненко Р. А.

Суми 2023

1. Призначення й мета створення веб-додатку

1.1 Призначення Телеграм-боту

Телеграм-бот дозволяє користувачам вводити текстові повідомлення або отримувати текстові файли та отримувати переклад на вибрану мову. Це дозволяє забезпечити комунікацію з користувачами, що володіють різними мовами, без необхідності вручну перекладати кожне повідомлення.

1.2 Мета створення Телеграм-боту

Створення Телеграм-боту для автоматичного перекладу з однієї мови на іншу текстових фрагментів в повідомленнях та чатах месенджера.

1.3 Цільова аудиторія

Даний продукт буде актуальний для використання в Україні. Цільовою аудиторією є люди будь-якого віку.

2 Вимоги до Телеграм-боту

2.1 Вимоги до Телеграм-боту в цілому

2.1.1 Вимоги до структури й функціонування Телеграм-боту

Телеграм-бот повинен бути реалізований за допомогою python на платформі Телеграм та забезпечувати визначений набір функціональних можливостей.

Кінцевий продукт даного проекту повинен мати увесь необхідний функціонал для забезпечення вимог користувача.

2.1.2 Вимоги до персоналу

Адміністратор боту не повинен мати особливих технічних навичок як для роботи з ним, так і його підтримки. Єдиною вимогою є наявність базового вміння користування персональним комп’ютером та Телеграм.

2.1.3 Вимоги до збереження інформації

Уся інформація по запитам надходить на сервер розробника, далі сервер Телеграм виконує обробку та шифрування ,кінцева дія – зворотній зв'язок між ботом та користувачем .

2.1.4 Вимоги до розмежування доступу

Розроблюваний Телеграм-бот має бути загальнодоступним. Права доступу до інформації розмежовані за групами користувачів: адміністратор, який підтримує бота та користувач, який використовує функції бота.

2.2 Структура боту

2.2.1 Загальна інформація про структуру

Клієнтський інтерфейс:

Телеграм-клієнт – користувачі взаємодіють з ботом через інтерфейс Телеграм Messenger. Вони можуть надсилати повідомлення, зображення та голосові записи.

Бот-сервер:

API Телеграм – бот взаємодіє з API Телеграм для отримання та відправки повідомлень, зображень та голосових повідомлень.

Web-сервер – бот може мати вбудований web-сервер, який взаємодіє з API Телеграм та обробляє отримані повідомлення та дані.

Модуль перекладу:

Текстовий переклад – цей модуль обробляє текстові повідомлення та здійснює їх переклад на вибрану мову.

Модуль обробки зображень – цей модуль отримує зображення та використовує розпізнавання тексту для виділення текстової інформації. Після цього отриманий текст перекладається на вибрану мову.

Модуль обробки голосових повідомлень – цей модуль отримує голосові повідомлення та використовує розпізнавання мови або інші алгоритми для перетворення голосу на текст. Отриманий текст перекладається на вибрану мову.

Обробка помилок та винятків – бот може мати модуль, який відповідає за обробку помилок та винятків, що можуть виникнути під час роботи, наприклад, проблеми з підключенням до API перекладу або розпізнаванням тексту.

2.2.2 Навігація

Для зручної навігації повинно бути створене меню, яке забезпечить швидке знайомство та розуміння користувача з усіма можливими функціями.

2.2.3 Наповнення боту

Мовою розроблення обрано Python, тому що вона має велику кількість довідкової документації та прикладів застосування, має великий набір актуальних бібліотек та зручний API для взаємодії з популярними месенджерами. Сервіс ,який приймає та відправляє повідомлення, через API Телеграм.

2.2.4 Дизайн та структура боту

Дизайн та структура Телеграм-боту для перекладу текстів іноземною мовою, за текстом, за картинкою, за голосовим повідомленням має бути наступним:

Користувацький інтерфейс:

– Введення тексту для перекладу.

– Вибір опції перекладу за текстом, картинкою або голосовим повідомленням.

– Відображення результату перекладу.

Модуль перекладу тексту:

– Обробка введеного тексту.

– Взаємодія з зовнішнім сервісом перекладу для отримання перекладу.

– Показ результату перекладу користувачу.

Модуль перекладу картинок:

– Завантаження та обробка зображень.

– Використання технологій розпізнавання тексту (OCR) для виділення тексту з зображень.

– Взаємодія з модулем перекладу тексту для перекладу отриманого тексту з зображення.

– Відображення результату перекладу користувачу.

Модуль перекладу голосових повідомлень:

– Приймання та обробка голосових повідомлень.

– Використання технологій розпізнавання голосу для перетворення голосових повідомлень на текст.

– Взаємодія з модулем перекладу тексту для перекладу отриманого тексту з голосового повідомлення.

– Відображення результату перекладу користувачу.

Керування станами та обробка помилок:

– Визначення можливих станів боту, наприклад, очікування введення тексту, завантаження зображень або голосових повідомлень.

– Обробка некоректних введених даних або помилок під час взаємодії з сервісами перекладу.

2.2.5 Система навігації

Головне меню:

– Представлення основних функцій боту (переклад за текстом, картинкою, голосом).

– Вибір режиму перекладу:

– Користувач може обрати режим перекладу за текстом, картинкою або голосовим повідомленням.

– Кожен режим може мати окрему команду або кнопку для активації.

Повернення до головного меню – в будь-якому режимі перекладу користувач може повернутись до головного меню, використовуючи відповідну команду або кнопку.

Обробка введених даних:

– Під час введення тексту, завантаження зображення або голосового повідомлення, бот повинен проводити перевірку на правильність та наявність необхідних даних.

– В разі помилки або неправильного формату даних, бот може повідомити користувача про необхідність виправлення або повторного введення.

Повідомлення результату перекладу:

– Після виконання перекладу, бот повинен відображати результат користувачу.

– Результат може бути представлений у вигляді текстового повідомлення, зображення або аудіофайлу, залежно від обраного режиму перекладу.

Додаткові функції – бот може мати додаткові функції, такі як вибір мови перекладу, збереження історії перекладів, надання додаткової інформації про перекладене слово або фразу.

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Пошук, фільтрація інформації	Відвідувач
UN-02	Перегляд контактів авторів	Відвідувач
UN-03	Перегляд звітності роботи боту	Адміністратор
UN-04	Створення, редагування та видалення необхідного тексту	Відвідувач
UN-05	Перегляд інформації про бота та його діяльність	Відвідувач
UN-06	Перегляд інформації про загальну кількість користувачів	Адміністратор
UN-07	Блокування користувачів за спам	Адміністратор
UN-08	Обробка повідомлень	Адміністратор

2.3.2 Функціональні вимоги

Бот повинен забезпечувати можливість введення текстових повідомлень користувачами для перекладу на обрану мову, надавати переклад тексту у відповідь користувачам шляхом відправлення текстового повідомлення з перекладом, підтримувати можливість відправлення зображень користувачами для розпізнавання тексту на них та його перекладу, можливість отримувати голосові повідомлення від користувачів та перетворювати їх на текст для подальшого перекладу, мати певний спектр мов, мати зручний та зрозумілий інтерфейс для взаємодії з користувачами, надавати чітку зворотну інформацію про стан перекладу та результати перекладу.

2.3.3 Системні вимоги

Проаналізувавши потреби користувачів Телеграм боту було визначено наступні вимоги:

– наявність Телеграм(web версія чи desktop);

– можливість в зручному та зрозумілому вигляді ознайомитися з функціоналом боту;

– пошук та фільтрація за мовою, або переклад тексту з картинки;

– наявність панелі адміністратора

звітування кількості користувачів та перекладів

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація боту відбувається з використанням:

– API Телеграм;

– Python;

2.4.2 Вимоги до лінгвістичного забезпечення

Увесь текст у боті має бути виконаний українською мовою, а зміст та наповнення з застосуванням мови програмування Python.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам – веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

3 Склад і зміст робіт зі створення веб-додатку

Докладний опис етапів роботи зі створення боту наведено в таблиці А.2.

Таблиця А.2 – Етапи створення боту

№	Склад і зміст робіт	Строк розробки
1	Розробка шаблону боту	5 дні
2	Розробка сторінок боту	14 днів
3	Розробка модулю зв’язку між користувачами	4 днів
4	Розробка зв’язку Телеграм з сервером	6 днів
5	Розробка адміністративної панелі	7 днів
6	Beta-тестування	8 днів
7	Alpha-тестування	5 днів
8	Перевірка працездатності	2 дні
9	Написання супровідної документації	5 дні
10	Реліз	1 день
	Загальна тривалість робіт	57 днів

4 Вимоги до складу й змісту робіт із введення боту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного боту вимогам технічного завдання (ТЗ) та можливість його ефективної роботи, в організації замовника повинен бути проведений певний комплекс заходів.

ДОДАТОК Б. Планування робіт

Б.1. Планування дипломного проекту

Мета проекту "Телеграм-бот для перекладу тексту на іноземну мову, перекладу тексту з картинки та голосового повідомлення" полягає у створенні функціонального і ефективного бота, який надасть користувачам можливість зручно та швидко перекладати текст на різні мови, використовуючи різні вхідні формати, такі як текстові повідомлення, зображення та голосові повідомлення. Має наступні етапи:

Початок:

– Визначення контексту проекту:

– Формулювання мети та завдань дипломної роботи.

– Визначення обмежень та обсягу проекту.

Аналіз предметної області:

– Вивчення існуючих інтернет-сервісів перекладу текстів, зображень та голосових повідомлень:

– Аналіз їх функціональності, особливостей та переваг.

– Визначення основних вимог до розроблюваного бота.

Проектування архітектури бота:

– Визначення модулів та їх функціональності.

– Розроблення структурної схеми бота.

– Визначення інтерфейсів та взаємодії між модулями.

Реалізація бота:

– Вибір мови програмування (наприклад, Python) та використовуваних технологій.

– Розроблення коду для кожного модуля бота.

– Інтеграція зі зовнішніми сервісами перекладу та розпізнавання тексту/голосу.

Тестування та налагодження:

– Виконання модульних та інтеграційних тестів для перевірки роботи кожного модуля та їх взаємодії.

– Виявлення та усунення помилок та недоліків.

– Оцінка швидкості, точності та якості перекладу.

Документування – Підготовка технічної документації, включаючи опис функціональності кожного модуля, інструкції з встановлення та використання бота, пояснювальну записку до дипломної роботи.

Б.2. Деталізація мети проекту методом SMART

Використання методу SMART (Specific, Measurable, Achievable, Relevant, Time-bound) є доцільним для уточнення та конкретизації мети проекту. Розглянемо метод SMART для опису мети проекту:

Результат деталізації методом SMART розміщено у таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Розробити Телеграм-бота, який здатен перекладати текст на іноземну мову, перекладати текст з картинок та голосових повідомлень.
Measurable (вимірювана)	Вимірювання досягнення мети можна визначити через показники, такі як: Кількість користувачів, які використовують бота для перекладу тексту. Кількість успішно перекладених повідомлень (текстових, зображень, голосових). Відгуки та оцінки користувачів щодо якості перекладу та функціональності бота.

Продовження табл. Б.1

Achievable (досяжна, узгоджена)	Розробка Телеграм-бота для перекладу тексту на іноземну мову, перекладу тексту з картинки та голосового повідомлення є досяжною завдяки наявним технологіям та сервісам машинного перекладу.
Relevant (актуальна)	В сучасному світі зростає потреба в швидкому та зручному перекладі тексту на іноземні мови. Такий Телеграм-бот може бути корисним для студентів, мандрівників, бізнесменів та будь-яких осіб, які мають потребу в перекладі.
Time-framed (обмежена в часі)	Є конкретний термін – до 10 червня 2023 р.

Б.3. Планування змісту робіт.

Планування змісту робіт для проекту "Телеграм-бот для перекладу тексту на іноземну мову, перекладу тексту з картинки та голосового повідомлення" може включати наступні етапи:

Аналіз вимог та дослідження – ретельно проаналізуйте вимоги проекту та зробіть необхідні дослідження щодо технологій, API та інструментів, що можуть бути використані для реалізації функціональності перекладу тексту, зображень та голосових повідомлень.

Визначення функціональності – сформулюйте конкретні функції, які має виконувати бот, такі як переклад тексту на іноземну мову, переклад тексту з картинки та голосового повідомлення.

Розробка архітектури – визначте архітектуру Телеграм-бота, включаючи взаємодію з API перекладу тексту та розпізнавання зображень, забезпечення обробки голосових повідомлень та зв'язку з платформою Телеграм.

Реалізація базової функціональності – розробіть основну функціональність бота, що дозволяє користувачам перекладати текст на

іноземну мову, перекладати текст з картинки та голосового повідомлення. Забезпечте зворотний зв'язок користувача та обробку запитів.

Інтеграція зі сторонніми сервісами – підключіть відповідні API для перекладу тексту та розпізнавання зображень для забезпечення потрібної функціональності бота.

Розробка функціональності перекладу зображень – реалізуйте можливість перекладу тексту, який міститься на зображенні. Використовуйте методи розпізнавання зображень та перекладу тексту для забезпечення цієї функціональності.

Розробка функціональності перекладу голосових повідомлень – забезпечте можливість перекладу тексту, який міститься у голосовому повідомленні. Використовуйте методи розпізнавання голосу та перекладу тексту для реалізації цієї функціональності.

Тестування та усунення помилок – виконайте ретельне тестування функціональності бота, включаючи переклад тексту на різні мови, переклад тексту з картинки та голосового повідомлення. Виправте будь-які помилки та недоліки, які виявляться під час тестування.

Розгортання та публікація – після успішного тестування та вдосконалення бота, розгорніть його на потрібному сервері та опублікуйте в мережі Телеграм для загального доступу користувачів.

Діаграма WBS зображені на рисунку Б.1.

OBS діаграма використовується для розподілу роботи між учасниками. У цьому випадку вся діяльність, пов'язана з пошуком ідей та визначенням роботи. Всі інші етапи повинні бути виконані студентом.Діаграма OBS показана на рисунку Б.2.

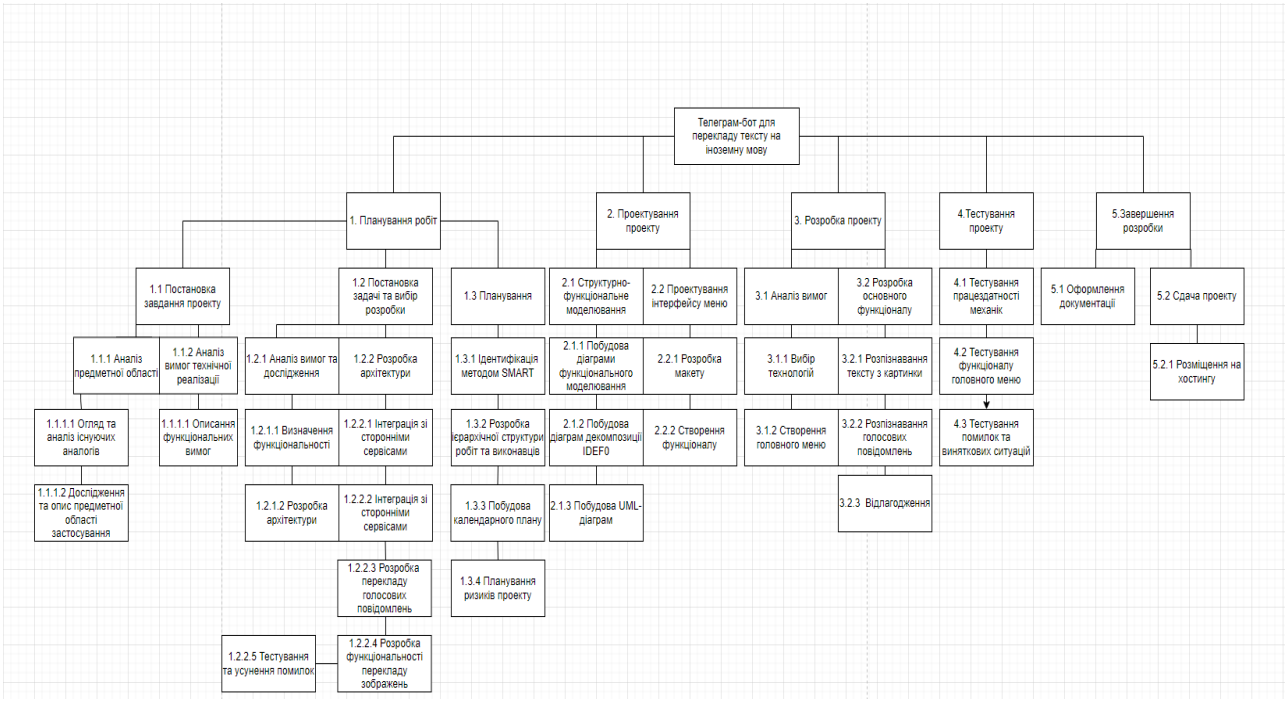

Рисунок Б.1 – WBS-структура робіт проекту

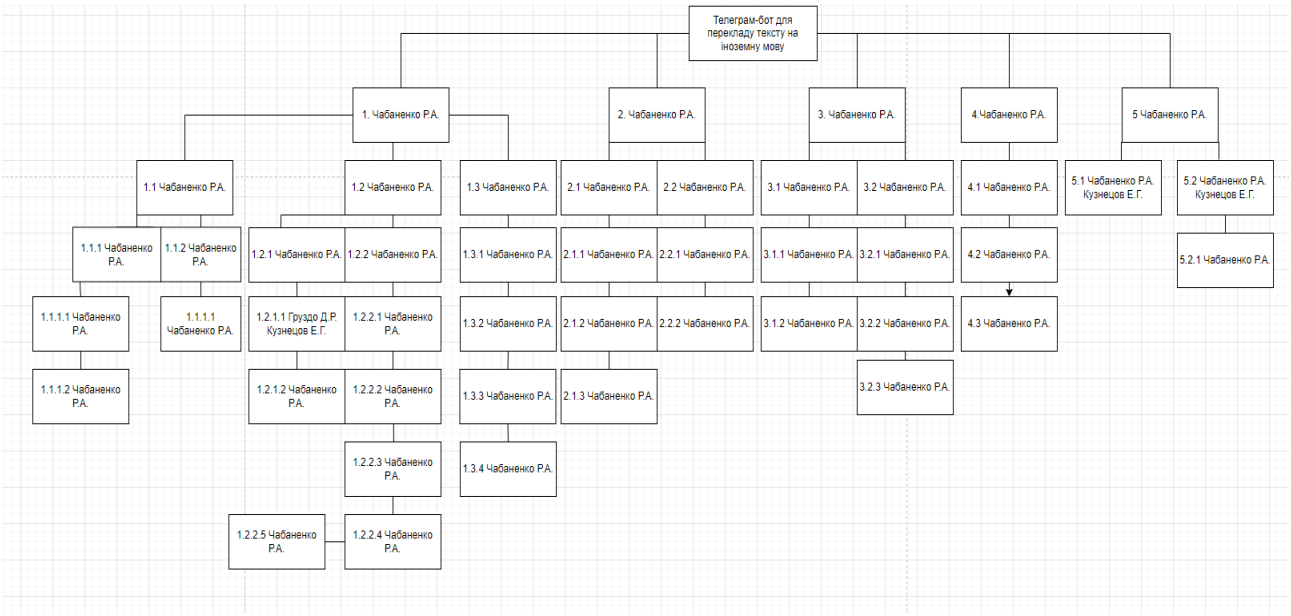

Рисунок Б.2 – Діаграма засобами OBS

Як показано на Рисунку Б.2, було створено діаграму Ганта для відстеження ключових показників, таких як цілі, підцілі, терміни їх досягнення, послідовність виконання завдань і відповідальні за досягнення цих цілей.

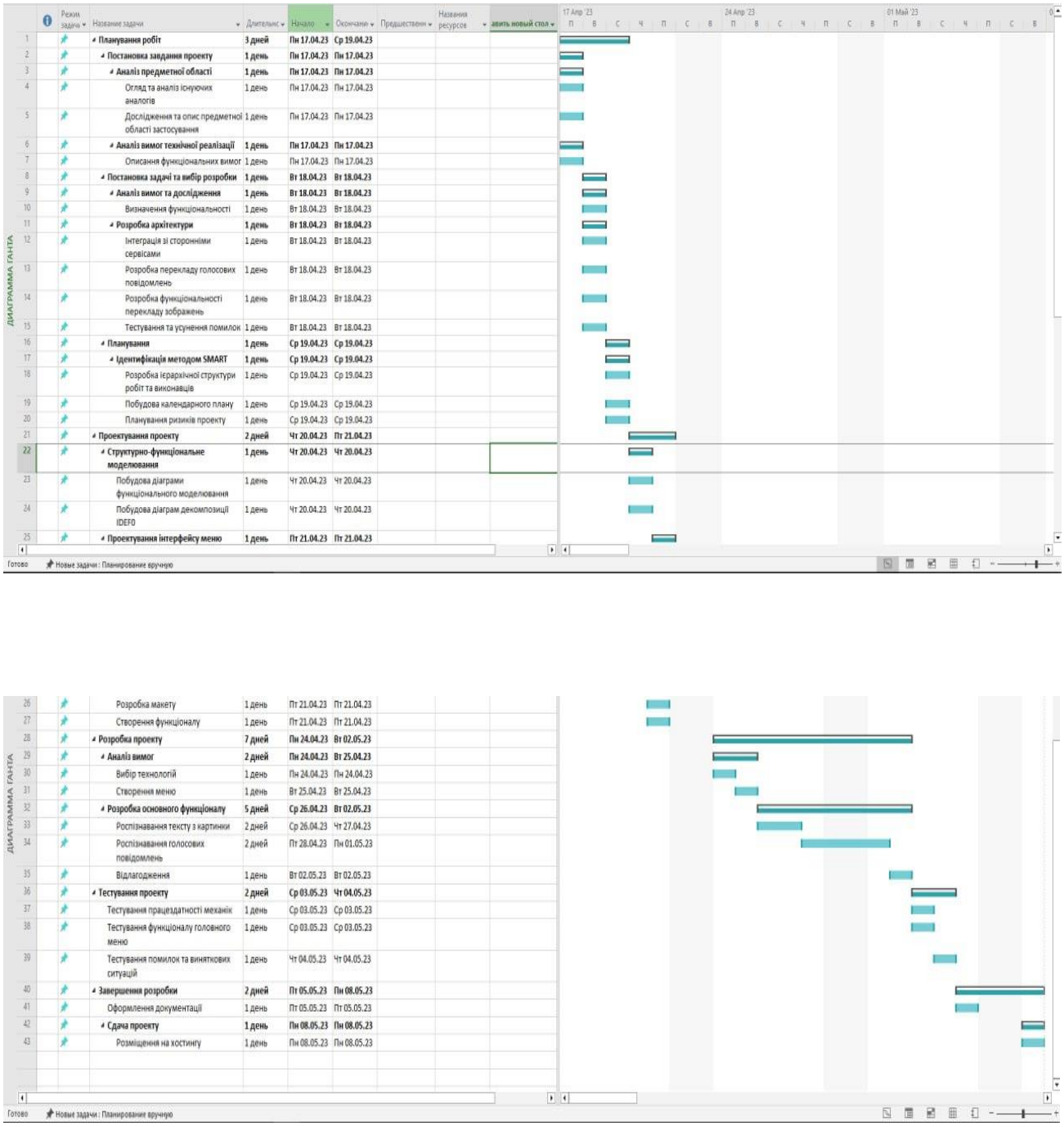

Рисунок Б.3 – Загальний вигляд діаграми всього проекту

Управління проектними ризиками – одна з найважливіших частин проекту. Виявлення потенційних проблем у розробці, тестуванні та дизайні може вберегти вас від непотрібних дій. Виявлені дефекти також можуть допомогти прискорити процес розробки та знайти альтернативи на етапі планування. Інформація про виявлені ризики представлена в таблиці Б.4.

Таблиця Б.4 – Ідентифікація ризиків

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Недостатня точність перекладу	0,3	0,8	0,24
2	Поява альтернативного продукту	0,3	0,2	0,06
3	Недостатня якість розпізнавання голосу	0,5	0,8	0,2
4	Залежність від зовнішніх сервісів	0,1	0,4	0,08
5	Погана взаємодія з платформою Телеграм	0,3	0,8	0,12

Для оцінки проблеми використовується матриця ймовірності та ризику. Всі несправності в зеленій зоні оцінюються як прийнятні, в жовтій – як виправдані, а в червоній – як неприйнятні.

Зелена оцінка означає, що проблема має лише незначний вплив на загальний бізнес.

Жовта оцінка означає, що ризик існує, але додаткові стратегії можуть запобігти проблемі.

Червоні ризики, однак, повинні бути усунені якомога швидше або ймовірність шкоди для проекту повинна бути значно знижена.

Розподіл ризиків за впливом та ймовірністю наведено в Таблиці Б.5. Шкала оцінки ризиків також наведена в Таблиці Б.6.

Заходи для ризиків, ідентифікованих у проекті, наведені в Таблиці Б.7.

Таблиця Б.5 – Матриця імовірності та впливу

Ймовірність ризику (Й)	Вплив загрози (ризику)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9					
0,7					
0,5					R3(0,20)
0,3			R2(0,06)	R5(0,12)	R1(0,24)
0,1				R4(0,08)	

Таблиця Б.6 – Шкала оцінювання за рівнем ризику

	Назва	Межі	Ризики, які входять (номера)
	Прийнятні	$0,005 \leq R \leq 0,05$	4
	Виправдані	$0,05 < R \leq 0,14$	2, 5
	Недопустимі	$0,14 < R \leq 0,72$	1, 3

Таблиця Б.7 – Заходи реагування на виявлені ризики проекту

ID ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратег іїреагування	План Б (заходи усунення наслідків ризику)
RS_1	Відкритий	Недостатня точність перекладу	Середня	Високий	0,24	Використання автоматичних систем перекладу може призводити до неточностей або неправильного розуміння контексту.	Попередження	Нейромережі швидко вчаться, згодом проблема упаде до мінімума

Продовження таблиці Б.7

ID ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратегії реагування	План Б (заходи усунення наслідків ризику)
RS_2	Відкритий	Поява альтернативного продукту	Низька	Середній	0,06	Розпізнавання тексту на зображеннях може бути складним завданням, особливо якщо зображення мають низьку якість або складну структуру. Це може призвести до неточностей або невдалого розпізнавання тексту.	Прийняття	Вибрати унікальну стратегію створення проекту

Продовження таблиці Б.7

ID ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратегії реагування	План Б (заходи усунення наслідків ризику)
RS_3	Відкритий	Недостатня якість розпізнавання голосу	Середня	Високий	0,2	Розпізнавання голосу може бути чутливим до шуму, акцентів або інших викликів. Це може призвести до помилок у розпізнаванні та неправильного перекладу голосових повідомлень.	Попередження	Нейромережі швидко вчаться, згодом проблема упаде до мінімума

Продовження таблиці Б.7

ID ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратегії реагування	План Б (заходи усунення наслідків ризику)
RS_4	Відкритий	Залежність від зовнішніх сервісів	Низька	Високий	0,08	Збереження конфіденційності користувачів та їхніх даних є важливою проблемою. Неправильна обробка або збереження персональної інформації може призвести до порушення конфіденційності.	Прийняття	Нейромережі швидко вчаться, згодом проблема упаде до мінімума

Продовження таблиці Б.7

ID ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратегії реагування	План Б (заходи усунення наслідків ризику)
RS_5	Відкритий	Погана взаємодія з платформою Телеграм	Середня	Високий	0,12	Бот може залежати від API та обмежень платформи Телеграм.	Попередження	При розробці бота необхідно враховувати правила та обмеження платформи Телеграм

ДОДАТОК В

Лістинг програмного коду

```
import mtranslate
import telebot
from PIL import Image
import pytesseract
import uuid
from mtranslate import translate
import speech_recognition as sr
import requests
import os
from pydub import AudioSegment

# создание бота
bot = telebot.TeleBot('6117148001:AAGx__Kg-
hUMZ50NKF8YrkGykF62Jsx8NKQ')

# инициализация OCR-движка
current_dir = os.path.dirname(os.path.abspath(__file__))

tesseract_path = os.path.join(current_dir, r'Tesseract-OCR\tesseract.exe')
pytesseract.pytesseract.tesseract_cmd = tesseract_path

# обработчик сообщений с изображениями
@bot.message_handler(content_types=['photo'])
def handle_photo(message):
    # скачивание фото
    file_info = bot.get_file(message.photo[-1].file_id)
    file = requests.get('https://api.Телеграм.org/file/bot{0}/{1}'.format(bot.token,
file_info.file_path))
```

```
    # сохранение фото
    filename = str(uuid.uuid4()) + '.jpg'
    with open(filename, 'wb') as f:
        f.write(file.content)

    # распознавание текста на фото
    img = Image.open(filename)
    text = pytesseract.image_to_string(img)

    # перевод текста на украинский язык
    translation = translate(text, 'uk')

    # отправка переведенного текста
    bot.send_message(message.chat.id, translation)

    # удаление фото
    os.remove(filename)

# обработка сообщений с текстом
@bot.message_handler(content_types=['text'])
def user_text(message):
    #сохранение текста
    text_1 = message.text
    #перевод текста на украинский
    transtext = translate(text_1, 'uk')
    #отправка переведенного текста
    bot.send_message(message.chat.id, transtext)

# обработка сообщений с голосовым сообщением
@bot.message_handler(content_types=['voice'])
```

```
def voice_handler(message):
    try:
        file_info = bot.get_file(message.voice.file_id)
        file = requests.get('https://api.Телеграм.org/file/bot{0}/{1}'.format(bot.token,
file_info.file_path))
        print('Response: ', file)
        with open('voice.ogg', 'wb') as f:
            f.write(file.content)
        #конвертирование в нужный формат wav
        audio = AudioSegment.from_file('voice.ogg', format='ogg')
        audio.export('converted_audio.wav', format='wav')

        # Распознавание речи в голосовом сообщении
        r = sr.Recognizer()
        with sr.AudioFile('converted_audio.wav') as source:
            audio = r.record(source)
        text = r.recognize_google(audio, language='en-US')

        print('audio: ', r, text)

        # Перевод текста с английского на украинский язык
        translated_text = mtranslate.translate(text, 'uk', 'en')

        # Отправка переведенного текста пользователю
        bot.reply_to(message, translated_text)

    except Exception as e:
        print('Error!', e)

bot.polling(none_stop=True)
```