

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ЗІ ЗВУКОВИМ
СУПРОВОДЖЕННЯМ ДЛЯ ЛЮДЕЙ З ОБМЕЖЕНИМИ
МОЖЛИВОСТЯМИ МОВОЮ PYTHON»**

Виконав: студент 4 курсу, групи ПД-44

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Сільченко В.О

(прізвище та ініціали)

Керівник

Бондарчук А.П.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

Київ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

«____» _____ 2021 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

СІЛЬЧЕНКО ВЛАДИСЛАВУ ОЛЕКСАНДРОВИЧУ

1. Тема роботи: «Розробка мобільного додатку зі звуковим супроводженням для людей з обмеженими можливостями мовою Python»

Керівник роботи Бондарчук Андрій Петрович, професор, доктор технічних наук затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки PyCharm Community Edition 2020.3.3

3.2. Алгоритм дії бота

3.3. Aiogram Bot API

3.4. Науково-технічна література, пов'язана з розробкою ботів на основі спеціальних API

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

4.1. Аналіз обов'язків розроблюваного бота

4.2. Аналіз та порівняння існуючих прототипів

4.3. Дослідження програмних засобів для розробки бота

4.4. Розробити функціонал створеного бота

5. Перелік графічного матеріалу

5.1.1. Популярність месенджерів

5.1.2. Переваги та недоліки найвідоміших месенджерів

5.1.3. Програмні засоби реалізації

5.1.4 Апробація результатів досліджень

6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих прототипів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту проектування	01.05.21 – 05.05.21	
5	Розробка функціоналу бота	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації застосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 40 ст. 22 рис. 18 джерел.

Ключеві слова: PyCharm, Python, Aiogram, Telegram, синтез мови, чат-бот.

Об'єкт дослідження – створення чат боту, що полегшує спілкування людей, з обмеженими можливостями

Предмет дослідження – чат-бот, для синтезу мови.

Мета роботи – розробка Telegram-бота з функцією «синтез мови», для людей с обмеженими можливостями на мові програмування Python.

Наукова новизна даної роботи полягає в наступному:

1. Розроблено чат бот, для поліпшення методів спілкування людей.
2. Встановлено, що зв'язка API Aiogram з мовою програмування Python, показує найкращі результати по швидкодії та використанню ресурсів комп'ютера.
3. На основі результатів виконаних порівнянь та досліджень, був розроблений авторський бот для спілкування між людьми, за допомогою функції «синтез мови».

В роботі виконано аналіз існуючих ботів аналогів. Встановлено переваги та недоліки існуючих ботів. В результаті аналізу було визначено основні потреби користувачів. Проаналізовано можливості середовища розробки PyCharm. Розроблено логіку практичних завдань та загальну концепцію представлення інформації для користувачів.

Галузь використання – бота може використовувати будь-яка людина, яка є користувачем месенджера, та є в чаті, куди був добавлений даний бот.

ЗМІСТ

ВСТУП.....	8
1 ПРОЕКТУВАННЯ ТА АНАЛІЗ ВИМОГ	9
1.1 Проектування.....	9
1.2 Аналіз вимог	9
2 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕНЬ.....	11
2.1 Месенджери.....	11
2.2 Telegram	14
2.3 Чат-боти	17
2.4 Огляд чат-ботів аналогів, по роботі з текстом.....	19
2.4.1 Чат-бот Grammar Checker	19
2.4.2 Чат-бот StenographerBot	20
2.4.3 Чат-бот VoiceMsgBot	21
2.4.4 Чат-бот OrfoBot	22
2.4.5 Чат-бот Plugy	22
3 ВИБІР ТЕХНОЛОГІЇ ТА СЕРЕДОВИЩА РОЗРОБКИ.....	23
3.1 Середовище розробки PyCharm Community Edition.....	23
3.2 Фреймворк Aiogram	24
3.3 Speech Recognition	24
3.3.1 Google Speech Api.....	25
3.4 PyDub Python	25
3.5 PySox Python	26
4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	27
4.1 Реєстрація чат-бота в Telegram.....	27
4.2 Реалізація чат-боту.....	29
4.2 Огляд можливостей інформаційної системи	35
4.3 Повний код чат-боту у Python	36
ВИСНОВОК	39
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	41

ВСТУП

Як ми бачимо, сучасний світ, не стоїть на місці. Технології стрімко розвиваються з кожним днем, з'являються нові ідеї та тренди, які впливають на повсякденне життя людей та суспільства. Ще не так давно, люди мали можливість спілкуватися тільки за допомогою листів, з тих пір було багато змін.

SMS – повідомлення, Електронна пошта, а згодом і популярні месенджери. В даний час існує багато засобів та форм «спілкування», де велика частина пов'язана з технічними можливостями, які залежать від глобальної комп'ютерної мережі.

Зв'язок з друзями та родичами, колегами, нові знайомства, обговорювання цікавих новин, з іншими користувачами – все це є частиною життя «сучасної людини».

Якщо до недавнього часу популярними були різні комп'ютерні додатки, пошта, то тепер перше місце займають месенджери. В яких є можливість не тільки спілкуватися з іншими людьми, а й користуватися перевагами чат-ботів, які значно полегшують життя.

Цим можна пояснити актуальність даної роботи, адже чат-боти дозволяють спростити щоденні рутинні справи, такі як пошук інформації про погоду, карта з заторами на дорогах, замовлення їжі, конвертація валют, керування фінансовими рахунками, акаунтами в соціальних мережах, отримання передових новин та інші.

Метою випускної роботи є розробка чат-боту Telegram мовою Python, який буде допомагати людям з вадами зору, або тим, у кого не має можливості прочитати текст повідомлення.

Базуючись на встановленій меті, були поставлені наступні завдання :

- проведення аналізу обраної предметної області;
- порівняння з іншими аналогами чат-ботів;
- вибір технологій і середовища розробки;
- розробка чат-бота на платформі Telegram.

1 ПРОЕКТУВАННЯ ТА АНАЛІЗ ВИМОГ

1.1 Проектування

Проектування – це перша частина розробки будь-якого проекту. Умовно, цей процес можна розділити на декілька етапів, а саме:

- а) збір вимог;
- б) аналіз області досліджень;
- в) створення прототипу системи.

Аналіз предметної області передбачає більш поглиблене вивчення вимог, що відносяться до функціоналу системи, обробка сценарію взаємодії, та алгоритмів.

Чітко сформовані цілі, та завдання, якісний аналіз вимог, опрацювання сценарію взаємодії с системою – формує успіх в досягненні необхідного результату.

Таким чином, проектування допомагає збільшити шанси на успішне виконання проекту, дає правильний напрямок у створенні.

1.2 Аналіз вимог

Завданням дипломного проекту було визначенно створення телеграмм-боту зі звуковим супроводженням для людей, з обмеженими можливостями.

Перед початком створення телеграмм боту, потрібно визначити основні вимоги, та описати їх. Також потрібно провести аналіз предметної області, та розглянути існуючі рішення, щоб переконатися в правильній моделі розробки.

Оскільки проект, який ми розробляємо потрібно буде впроваджувати у вже існуючу програму, це накладає певні обмеження на технології, які можна буде використовувати.

Виходячи з поставлених цілей, ми розробляємо телеграм бот, який:

- а) буде розпізнавати написаний текст;
- б) буде певним чином перетворювати його в звуковий запис, у вигляді голосового повідомлення;

- в) дасть змогу прослуховувати запис;
- г) буде розпізнавати відправлений запис;
- д) буде перетворювати звуковий запис в текст.

2 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕНЬ

2.1 Месенджери

Не багато додатків, вплинули на психологію користувачів так само сильно, як месенджери. По суті, вони повністю переписали принципові правила спілкування і змінили наші емоції під час розмов.

Початок історії месенджерів приходить на 90х роки 20 століття, проте сама для спілкування великої кількості користувачів між собою за допомогою електронних пристроїв, поєднаних в уніфіковану мережу, з'явилась ще в 60х. Так система обміну даними в режимі, наближеному до реального часу, під назвою Compatible Time-Sharing System (CTSS) була створена у Масачусетському Технологічному інституті (MIT) ще 1961 року (більшість наших читачів на той час ще навіть не народилися). В межах цього експериментального проекту до 30 користувачів могли одночасно увійти в спільне інформаційне середовище та обмінюватися повідомленнями між собою. До 1965 року вона стала доволі популярним внутрішнім інструментом комунікації у MIT. Поява peer-to-peer протоколу у 1970-х дала змогу підключати різних користувачів до одного і того ж комп'ютера та обмінюватися даними — і перемикатися між різними користувачами одного і того ж комп'ютера. На той час, месенджери були виключно текстовими, які працювали звертаючи увагу на дуже повільний інтернет, тільки зі стаціонарними ПК.

Історія месенджерів у вигляді повноцінних додатків, а не систем з обмеженим доступом, почалась в 1996 році. В той час, ізраїльська компанія Mirabilis запустила всім відомий ICQ, оновлення якого виходять по наші дні.

Ця програма відрізнялася своїми багатокористувацькими чатами, пошуком по базі користувачів, можливістю обміну файлами, та іншими корисними опціями, яких не було у месенджерів 80х років

Після показу можливостей, та популярності цієї програми, з великою швидкістю, на ринку почали з'являтися інші месенджери. Власні розробки були у

таких велетнів, як Yahoo(Yahoo!Messenger), MSN messenger Білла Гейтса, у якого було більше 330 млн. активних користувачів, до 2009 року, що місяця. Їх відмінною рисою, було створення власних платформ і чатів, але була проблема з синхронізацією історії повідомлень, між пристроями, оскільки на той час, ще не було хмарних сервісів, та теперішніх можливостей роботи з файлами, для мобільних користувачів.

Саме тоді, при обмеженнях трафіку, та популярності культури спілкування, з'явилися засоби вираження емоцій, за допомогою аббревіатур на певні типи повідомлень, таких як: LOL (Laugh(ing) Out Loud), BRB (Be Right Back), OMG (Oh My God/Gosh). Згодом, коли швидкість почала дозволяти відправляти різні графічні елементи, користувачі перейшли на «смайлики» та «стикери».

Справжній вибух популярності цього доповнення, був на 2014-2015 роки, після вдосконалення можливостей спілкування в таких месенджерах, як Viber, Telegram, Facebook Messenger. Не в останню чергу, після появи сучасного 3G інтернету, який дозволяв їх відправляти.

Зміни у сприйнятті месенджерів як засобу комунікації настали в момент, коли швидкості та стабільність роботи бездротових мереж дали змогу людям спілкуватися не лише за монітором комп'ютера чи екраном ноутбуку. Після низки тестових спроб у 2010-х майже всі основні месенджери (від Skype та Facebook до більш екзотичних у наших краях Line та WeChat) запустили власні версії для мобільних браузерів чи окремі додатки для смартфонів.

А все це сталося завдяки появі iPhone від Apple та смартфонів на ОС Android від Google. Справжню революцію здійснив WhatsApp, створений нащадком емігрантів з України Яном Кумом та экс-інженером Yahoo Брайаном Актоном. Цей месенджер першим мав прив'язку до номеру мобільного телефону та першим отримав підтримку push-сповіщень для смартфонів Apple. Згодом цей месенджер викупила компанія Марка Цукерберга.

Експерти вважають, що зростання популярності програм для швидкого обміну повідомлень з кожним роком будуть тільки рости, вже зараз месенджерами користуються не менше 2 млрд користувачів по всьому світу. Прогнози

спеціалістів показані на рисунку 1.

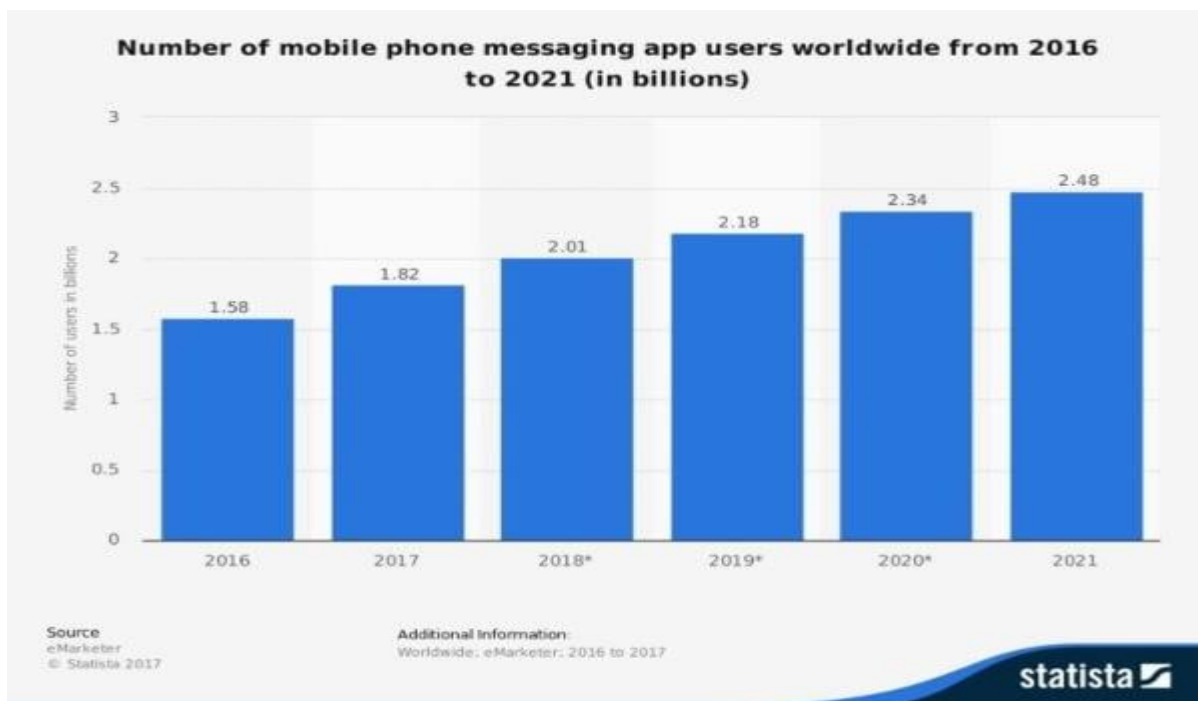


Рисунок 2.1 – Прогноз кількості користувачів месенджерів 2016-2021 рік в млд.

Також було розглянуто рейтинг різних месенджерів на 2020 рік. Ця статистика була представлена на рисунку 2.

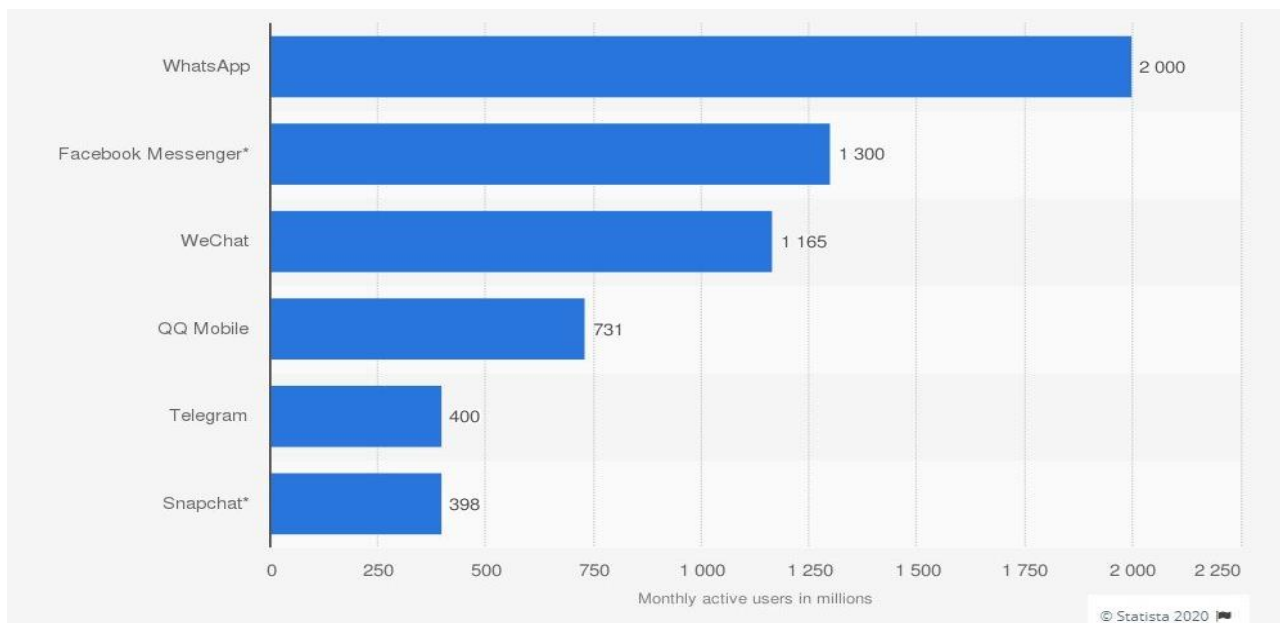


Рисунок 2.2 – Рейтинг месенджерів на момент 2020 року.

Якщо порівнювати месенджери між собою, то можна побачити, що ці додатки

мають схожі функції, такі як відправка повідомлень, зображень, аудіо- та відео-файлів. Але звісно в кожному з них, є свої переваги та недоліки.

2.2 Telegram

Telegram – це безкоштовний додаток, який був розроблений на мові програмування C++, він дозволяє користувачам обмінюватися повідомленнями і файлами більшості форматів. Месенджер використовує спеціальну серверну частину з закритим кодом, яка працює на серверах США.

Telegram має ряд переваг:

- а) конфіденційність – всі повідомлення зашифровуються, і видаляються через зазначений час;
- б) оптимізація – сервера Telegram розташовані по всьому світу, що підвищує ефективність роботи;
- в) реклама – її повна відсутність в функціоналі додатку;
- г) швидкість – швидкість доставки на багато більше, ніж у аналогів додатку.
- д) ніяких обмежень на розмір повідомлення, або файлів;
- е) є можливість зберігати особисті данні в архівах акаунту.

Засновниками месенджера Telegram є Павло та Микола Дуров. Павло займався фінансовим та ідеологічним аспектами проекту, а Микола технічними.

Однієї з найвідоміших особливостей є введення секретних чатів в Telegram, та спеціально розроблений протокол шифрування MTProto, для них. Зображено на рисунку 3.

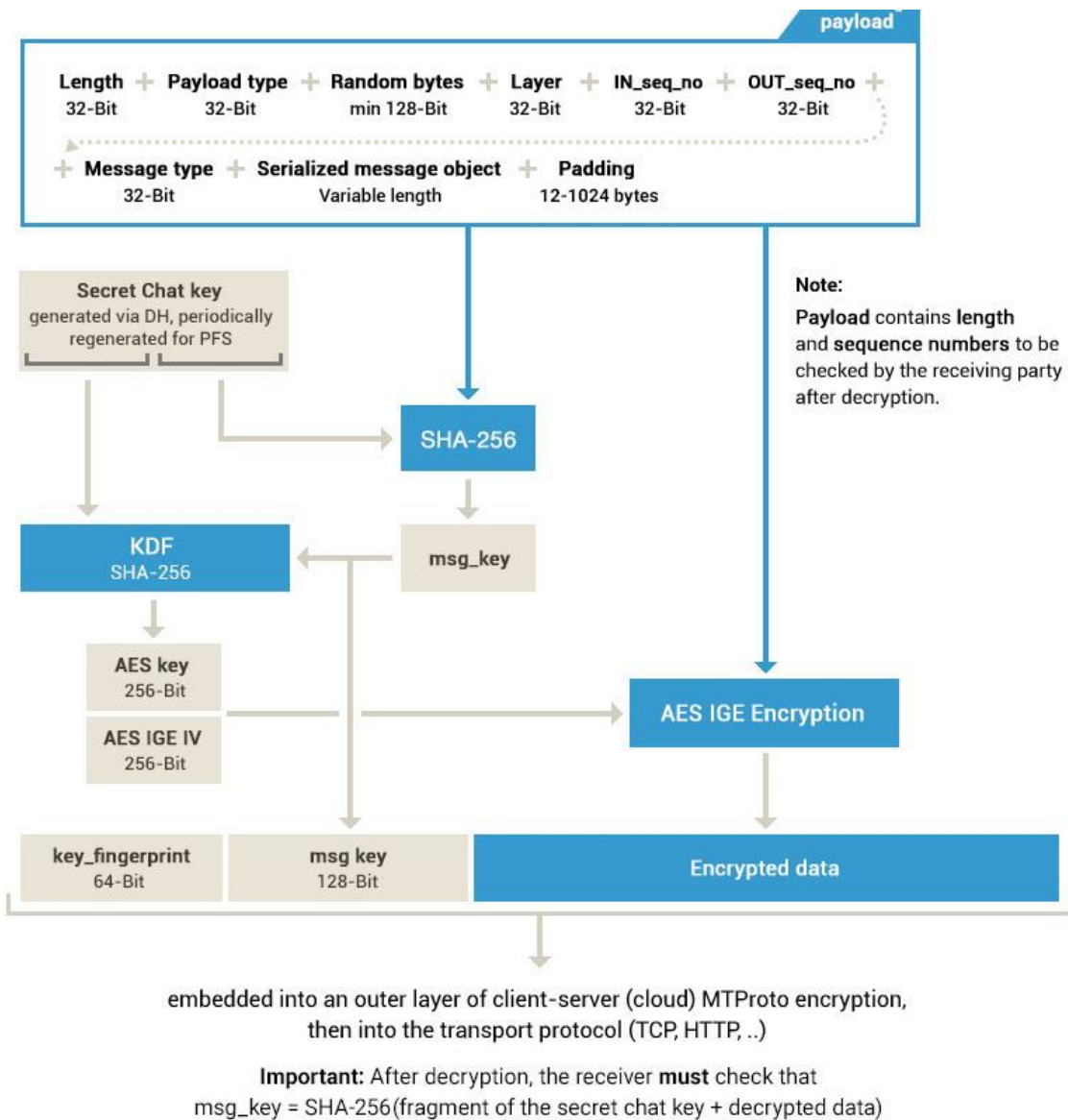


Рисунок 2.3 – Принцип роботи протоколу шифрування MTProto.

Секретні чати створювались для максимально безпечного, та приватного спілкування користувачів. Окрім використання протоколу зображеного вище, всі повідомлення шифруються ключами за принципом «пристрій до пристроя». Ніхто не зможе перехопити, та розшифрувати повідомлення, це відноситься також і до робітників Telegram, у яких також не має доступу, до вашої особистої інформації. Повідомлення з таких чатів неможливо переслати, та вони не зберігаються на серверах Telegram. Секретні чати доступні тільки на тих пристроях, з яких вони створювались.

Месенджер може працювати на всіх відомих платформах, таких як Windows, Android, IOS, Linux, та інші.

Також ми розглянемо інтерфейс додатку, на рисунках 4, 5 і 6.

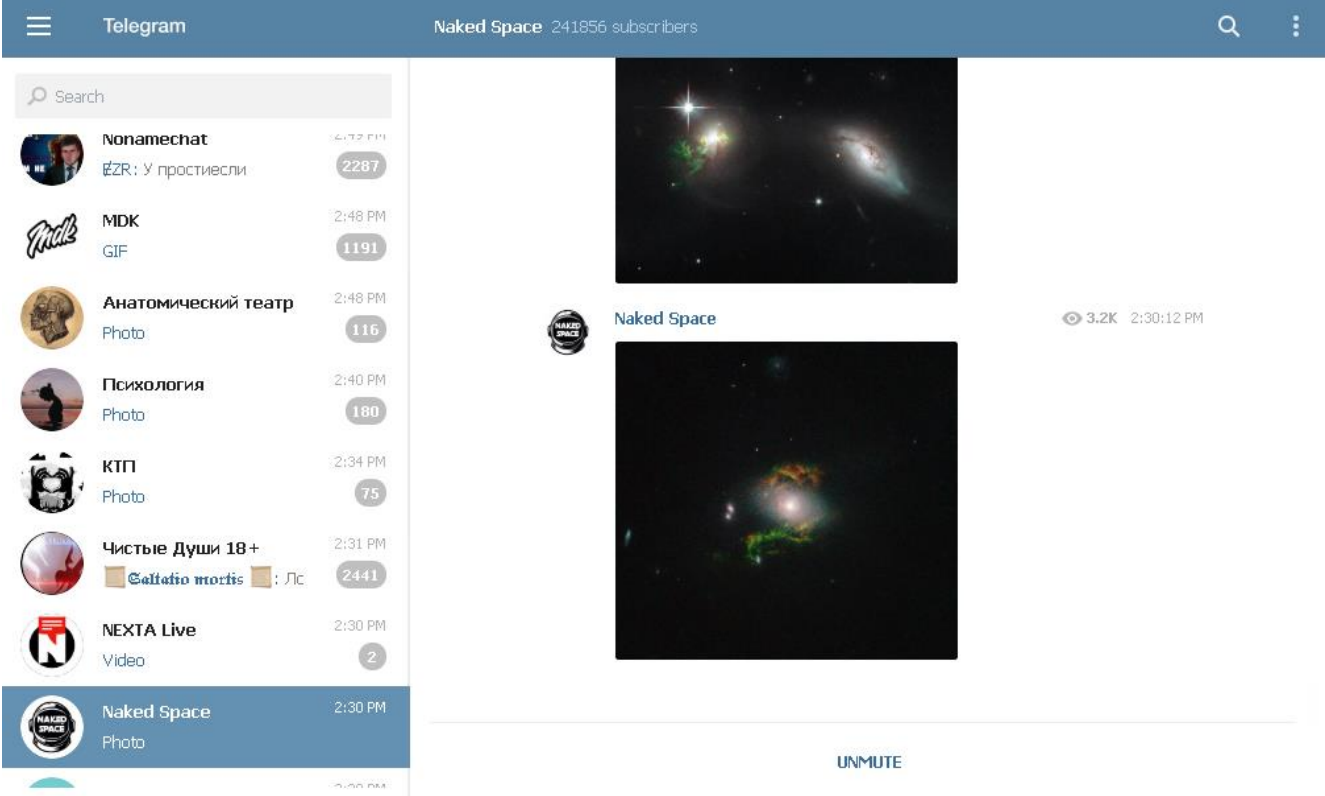


Рисунок 2.4 – Інтерфейс телеграм в web версії браузера Google Chrome.

WEB версія додатку Telegram, доступна в будь якому популярному браузері на ПК та мобільних пристроях.



Рисунок 2.5 – Інтерфейс в мобільній додатку Telegram на Android 11

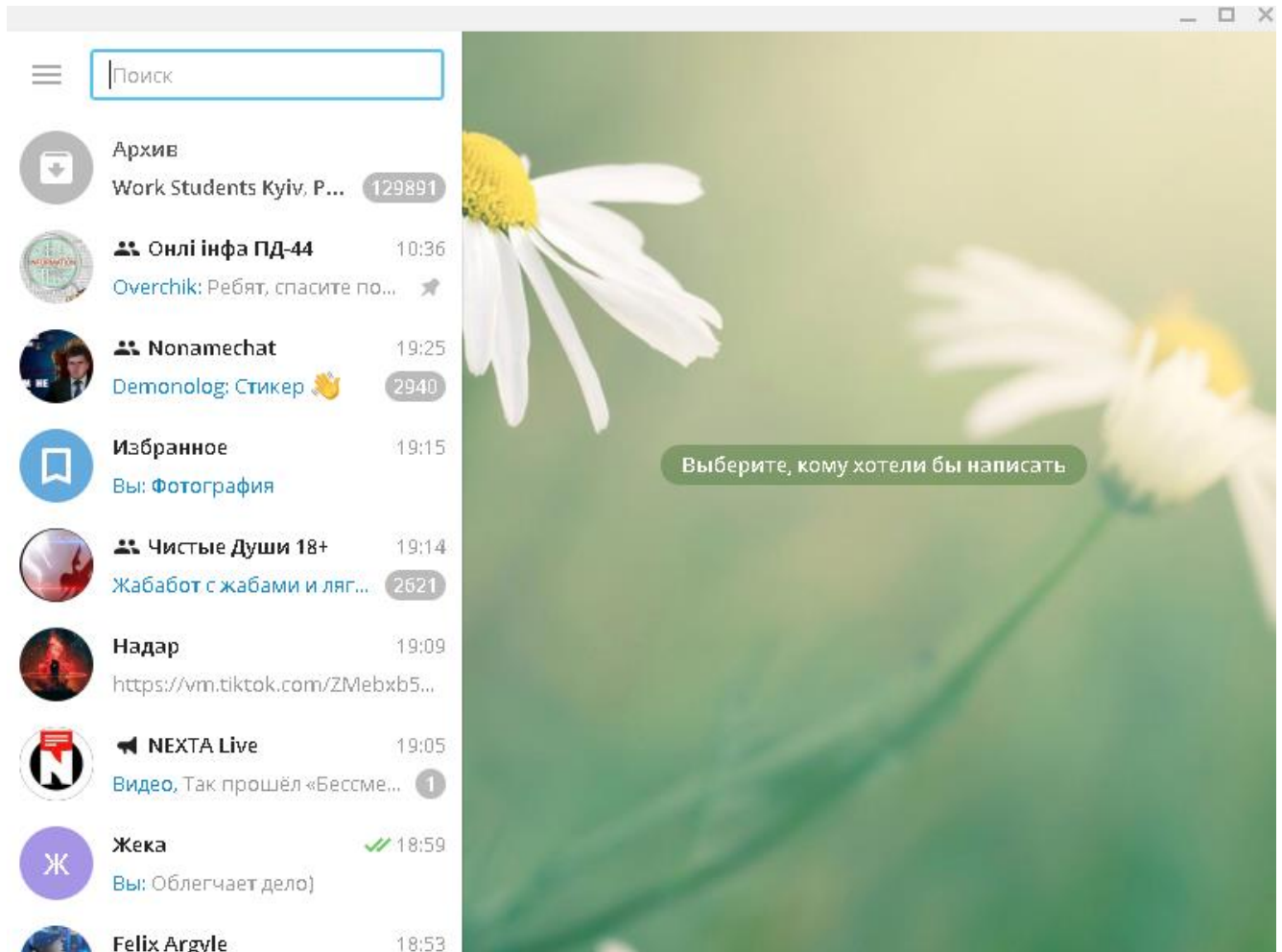


Рисунок 2.6 – Інтерфейс Telegram на Windows 10.

Додаток доступний майже на всіх сучасних мовах світу, таких як: Українська, Англійська, Російська, Німецька, Французька, Корейська, Китайська, тощо.

2.3 Чат-боти

Чат-боти - це спеціальні акаунти, за якими не закріплено якоїсь людини, а повідомлення, відправлені з них або на них, обробляються зовнішньою системою. Крім того, для користувача спілкування з ботом виглядає як звичайна листування, з реальною людиною.

Чат-бот - це розумна програма, яка живе в месенджерах і виконує різні,

вкладені до неї функції.

Функції та переваги чат-ботів:

1) Підтримка клієнтів

Чат-бот допоможе замінити незручний FAQ на сайті, зможе відповісти на типові питання клієнта. На відміну від людей, бот буде працювати 24 години на добу.

2) Клієнтський сервіс

За допомогою чат-бота можна робити різні покупки і замовляти послуги. В роздрібної торгівлі, з постійним розширенням асортименту, важче шукати конкретні товари. Після невеликого аналізу бот зрозуміє, що цікавить клієнта, і відправить пряме посилання

3) Маркетинг

Чат-бот - це ще один маркетинговий інструмент, який допоможе поширювати контент, підтримувати лояльність клієнтів і збирати аналітику. За допомогою нього, можна робити розсилки, інформувати клієнтів про акції, збирати коментарі про товари чи послуги, якість обслуговування

4) Праця в середині компанії

Чат-боти допоможуть оптимізувати в роботі такі процеси як: бронювання переговорок, інформування співробітників про дати відпустки, розклад корпоративного транспорту, терміни зарплати і багато іншого.

5) Підбір кандидатів або пошук роботи.

6) Розсилка повідомлень.

7) Розвага користувачів

В телеграмі на даний момент існує більше декількох сотен ігор, для роботи яких, потрібно просто увімкнути чат-бот.

Основний недолік чат-ботів - помилки в нестандартних ситуаціях: боти реагують тільки на певні ключові слова, тому обов'язково повинна бути переадресація на живих співробітників.

2.4 Огляд чат-ботів аналогів, по роботі з текстом

2.4.1 Чат-бот Grammar Checker

Основна задача цього чат-боту, є перевірка граматики, в англійській мові. @fixmebot - виправляє помилки і перефразовує текст англійською мовою. Буде корисний тим, хто вивчає цю мову або веде на ньому спілкування, але має прогалини в знанні англійської граматики.

Для початку роботи, потрібно ввести будь який текст, на прохання бота.

Приклад використання чат-боту представлений на рисунках 7 та 8.

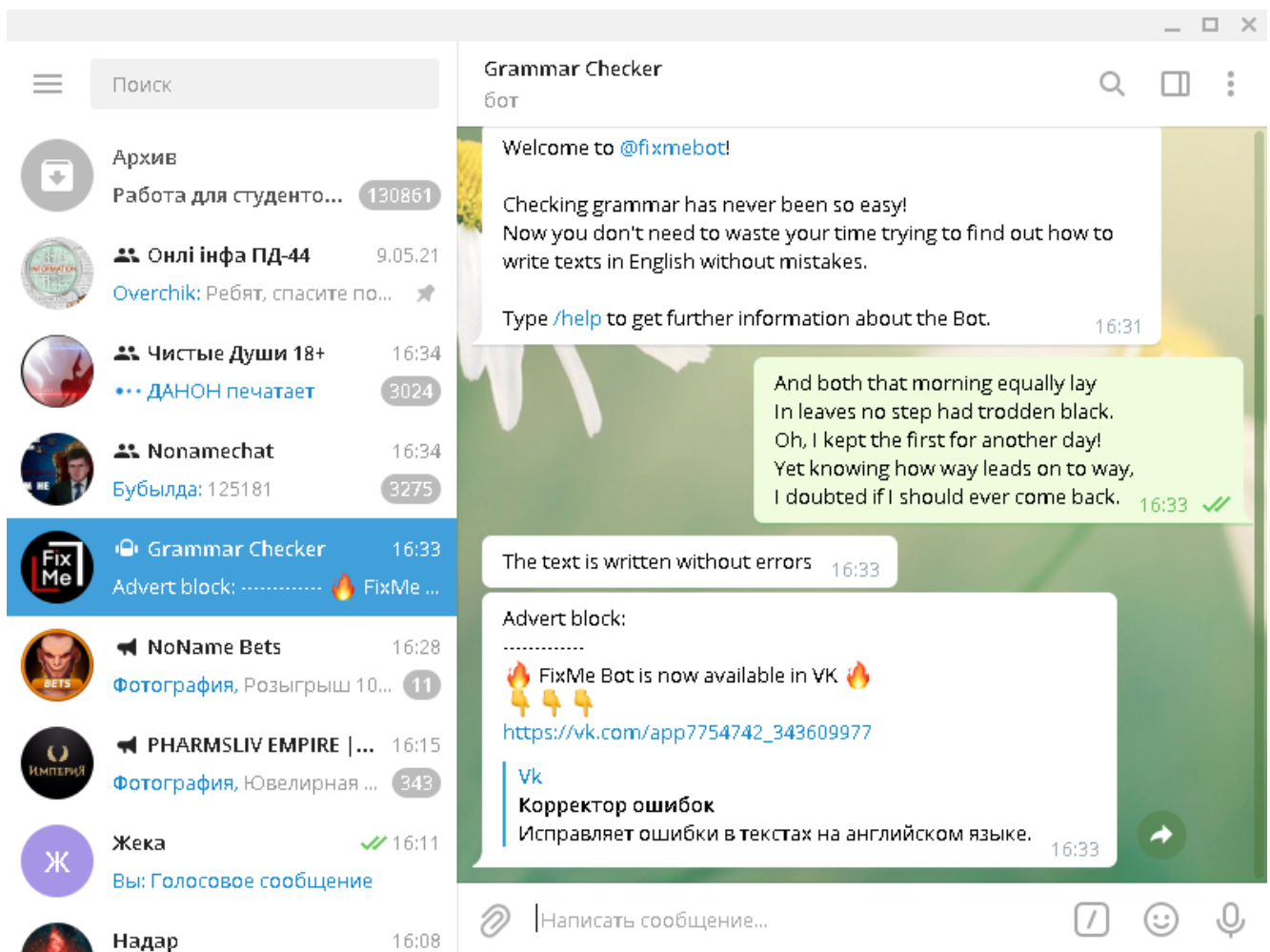


Рисунок 2.7 – Приклад реакції чат-боту на граматично правильно сформований текст.

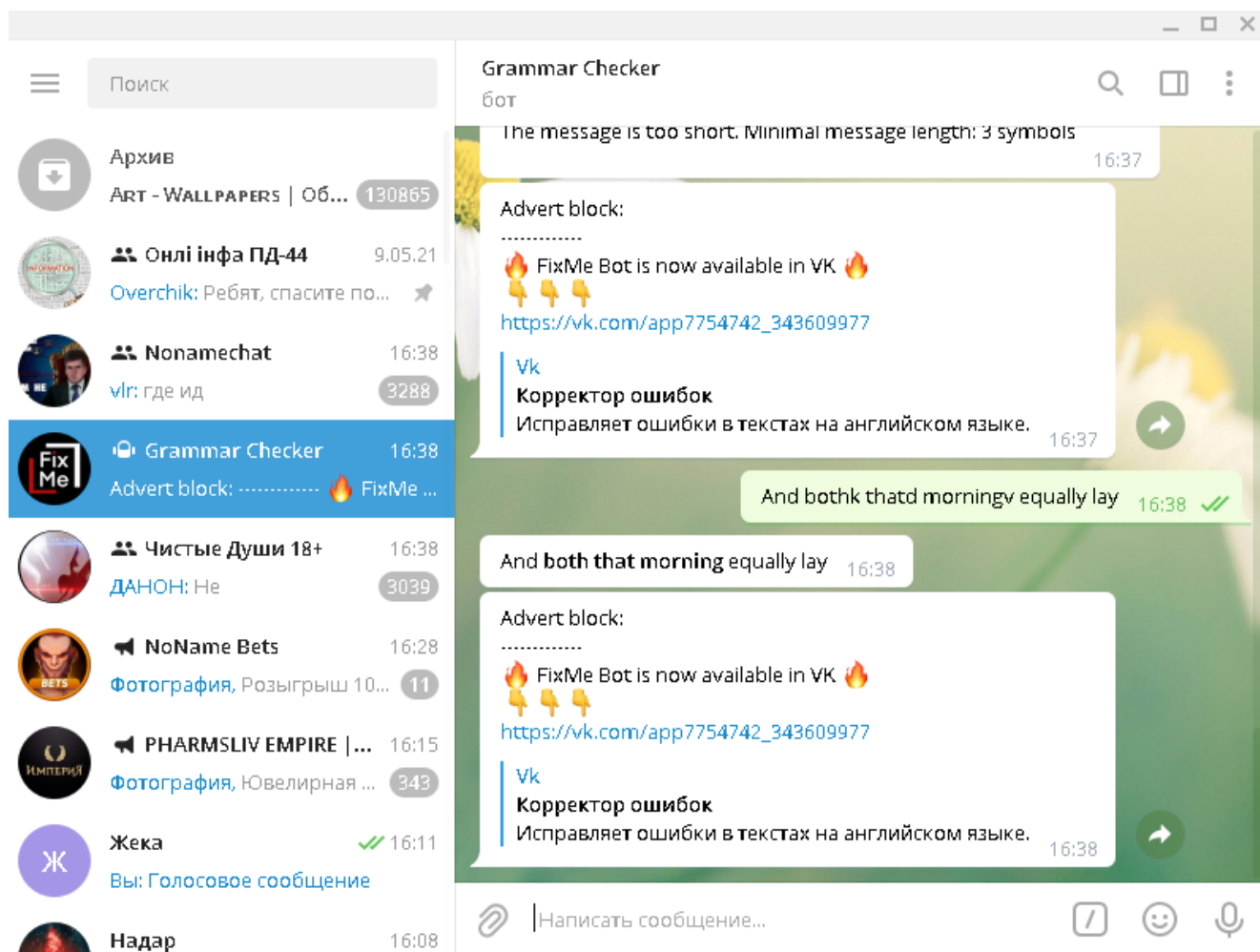


Рисунок 2.8 – Приклад реакції чат-бота на граматично не правильно сформований текст.

2.4.2 Чат-бот StenographerBot

Основною задачею цього чат-бота, є переведення голосового повідомлення користувача, в текст.

Корисно для тих людей, хто не має багато часу для написання великого повідомлення, в тих ситуаціях, де людина яка його отримає, не має можливості прослуховувати голосові.

Для початку роботи, потрібно записати, або переслати голосове повідомлення, в чат.

На жаль права на цей чат бот продали компанії Яндекс, тому ми не зможемо опробувати його на території України.

2.4.3 Чат-бот VoiceMsgBot

VoiceMsgBot – це безкоштовний чат бот, який переводить голосове повідомлення, в текст. Працює без обмежень на кількість слів, або ж тривалість голосового повідомлення.

Приклад роботи чат-боту представлений в рисунку 9.

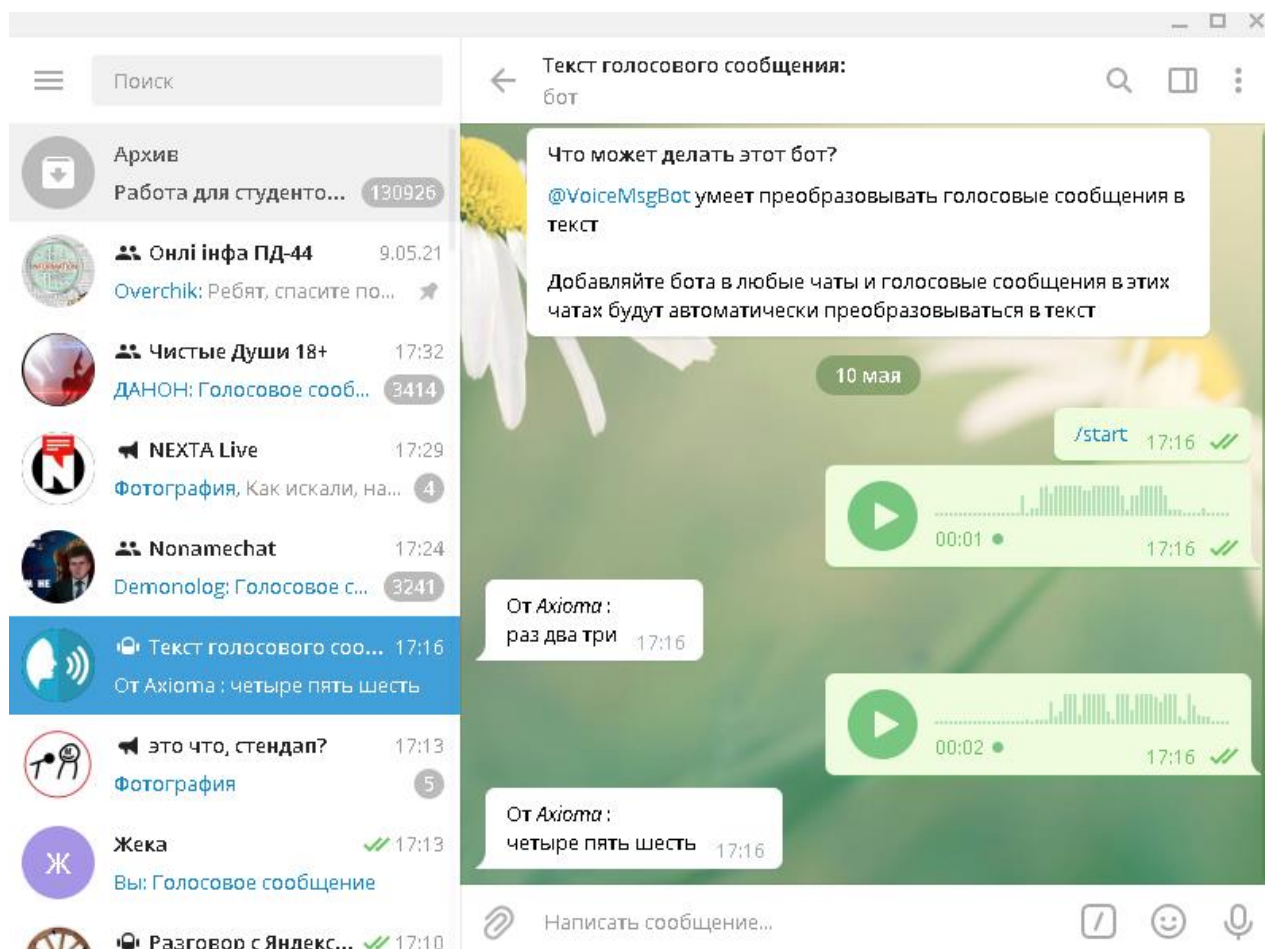


Рисунок 2.9 – Приклад перетворення голосового повідомлення в текст.

Є досить багато чат-ботів аналогів нашому, але всі вони мають досить мало функцій. На відміну від них, наш чат бот зможе конвертувати як текстові повідомлення в голосові, так і голосові в текстові.

2.4.4 Чат-бот OrfoBot

Основною задачею цього чат-боту є перевірка тексту, на орфографічні, або граматичні помилки. Приклад роботи чат-бота зображено на рисунку 10.

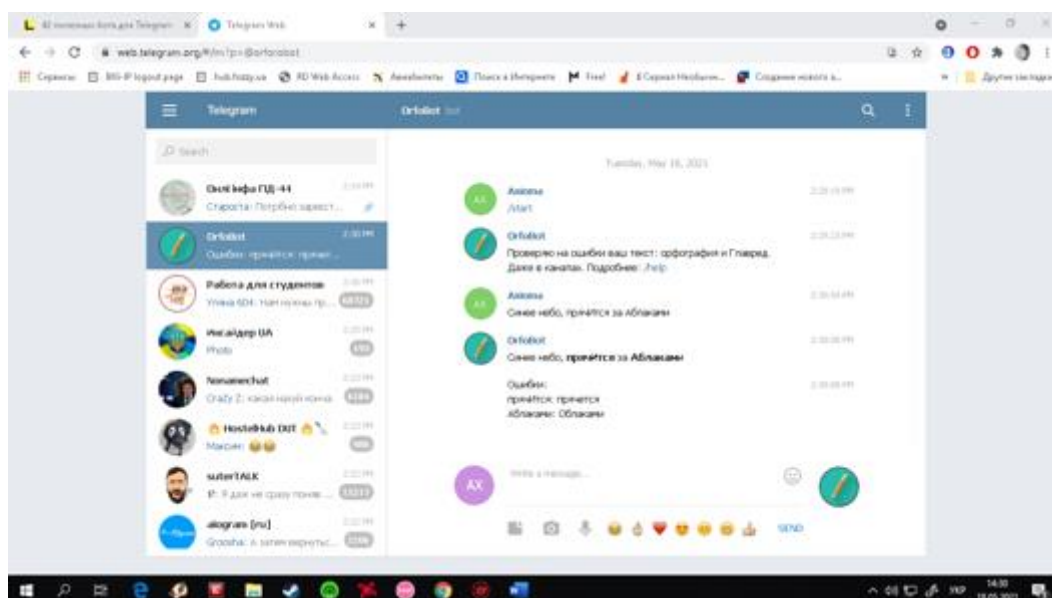


Рисунок 2.10 – Приклад роботи чат-бота OrfoBot

Цей чат-бот знаходить помилки у тексті, а потім позначає їх наявність у тексті.

2.4.5 Чат-бот Plugy

Основна функція цього чат-боту, це перевірка будь-яких текстів на унікальність в інтернеті. Якщо текст не проходить перевірку, бот відправляє користувачу адресу сайту, де його можна знайти.

3 ВИБІР ТЕХНОЛОГІЇ ТА СЕРЕДОВИЩА РОЗРОБКИ

3.1 Середовище розробки PyCharm Community Edition

PyCharm Community Edition – легкий та потужний редактор коду, розроблений компанією JetBrains, сумісний з Windows, MacOS та Linux.

PyCharm був випущений на ринок інтегрованих середовищ розробки для Python щоб створити конкуренцію з PyDev і поширенішим середовищем розробки Komodo IDE.

Можливості PyCharm:

- Статичний аналіз коду, підсвічування синтаксису і помилок;
- Навігація серед проектів і початкового коду: відображення файлової структури проекту, швидкий перехід між файлами, класами і методами;
- Рефакторинг: перейменування, витяг методу, введення змінної, введення константи, підняття і опускання методу тощо;
- Інструменти для веб-розробки з використанням фреймворку Django;
- Вбудований «відлагоджувач» для Python, який використовується для тестування, та виправлення помилок в коді;
- Вбудовані інструменти для юніт-тестування;
- Розробка з використанням Google App Engine.

З цього можна зрозуміти, що PyCharm одна з кращих, якщо не найкраща, повнофункціональна, спеціалізована та універсальна IDE для розробки на Python. Він має масу можливостей, які економлять час, допомагаючи вам з рутинними завданнями.

Інтерфейс PyCharm показаний на рисунку 11.

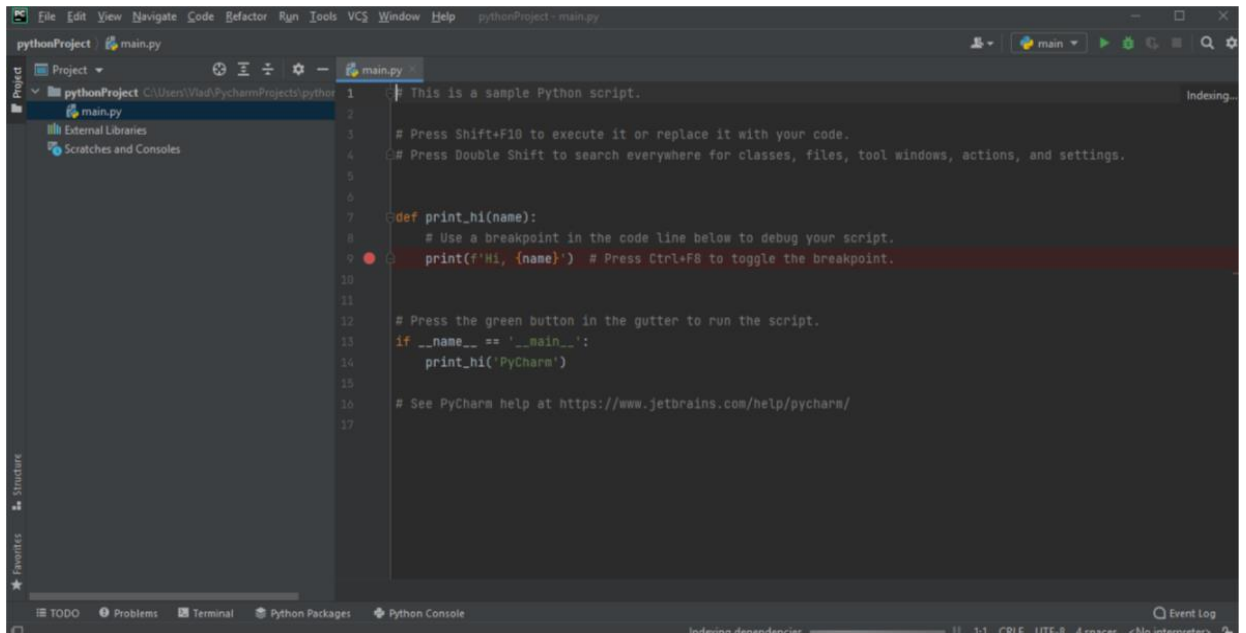


Рисунок 3.11 – Інтерфейс програми PyCharm.

3.2 Фреймворк Aiogram

Фреймворк Aiogram - це технологія, що була викристана під час розробки чат-боту. Досить простий і повністю асинхронний фреймворк для API Telegram Bot, написаний на Python 3.7 з стандартні бібліотеки «asyncio» та іншого асинхронного веб-фреймворку «aiohttp». Його найголовнішою особливістю, є можливість витримувати дуже великі навантаження, при мінімальних ресурсах.

Ця бібліотека вважається найкращою в розробці ботів на Python.

3.3 Speech Recognition

Все частіше в світі, застосовується технологія розпізнавання мови (через API Speech Recognition), що дозволяє обчислювальним системам розпізнавати та реагувати на людську мову. Ця новаторська технологія змінила розуміння праці людини з комп'ютером.

Сучасне розпізнавання мовлення використовує алгоритми глибоких нейронних мереж і може розуміти більше ста мов.

Технологія використовується світовими компаніями для реалізації таких

функцій, як пошук «по голосу», керування різними пристроями через команди, обробці голосових повідомлень.

Список найкращих API, на теперішній момент:

- Google Speech API;
- IBM Watson API;
- Speech API;
- Speech to Text API;
- Text-to-Speech API;
- Rev.AI API;
- ReadSpeaker API;
- Siri API;
- Yandex Speech API;
- Read Speaker API.

3.3.1 Google Speech Api

Для реалізації функції «синтеза мови» в боті, було використано технологію Google speech, простий мультиплатформенний інструмент CLI для читання тексту за допомогою API Google Translate TTS (Text To Speech).

Його особливостями є:

- Підтримка 64 різних мов;
- Можливість считування текст без обмеження по довжині;
- Може считування тексту із стандартного введення користувачем;
- Автоматичне збереження завантажених даних в локальному кеші.

3.4 PyDub Python

PyDub - це бібліотека Python для роботи лише з файлами «.wav». За допомогою цієї бібліотеки ми можемо відтворювати, розділяти, об'єднувати, редагувати наші «.wav» аудіо файли.

Бібліотека PyDub, дозволила перевести створенні голосові повідомлення, в потрібний для боту формат звукового запису. Функції, що може використовувати PyDub:

- Відтворення аудіофайлів;
- Збільшення / зменшення обсягу файлів;
- Об'єднання двох або більше аудіофайлів;
- Експорт аудіофайлів;
- Переведення аудіофайла в інші формати.

3.5 PySox Python

SoX - це популярний інструмент командного рядка для обробки звуку. Серед багатьох інших процесів це дозволяє користувачам оброблювати аудіофайли великих розмірів, в один рядок коду. Бібліотека досить потужна та стабільна, за рахунок чого може будувати довгі командні рядки.

Функціонал, що підтримується в pysox, включає перетворення окремих аудіофайлів, поєднання декількох аудіофайлів та отримання інформації про аудіофайл.

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Реєстрація чат-бота в Telegram

Для того, щоб створити чат бот в телеграмі, потрібно спочатку пройти реєстрацію в спеціальному додатку «@BotFather».

Перший крок починається з команди «/newbot», після цього, вам запропонують ввести назву свого бота. Обовязковою умовою є те, щоб в кінці назви, було вказано приставку «Bot», або ж «_bot». Якщо користувач виконав умови, бот відсилає індивідуальний токен (спеціальний набір символів для доступу), який є «ключем» до вашого особистого бота. Приклад зображений на рисунку 12.

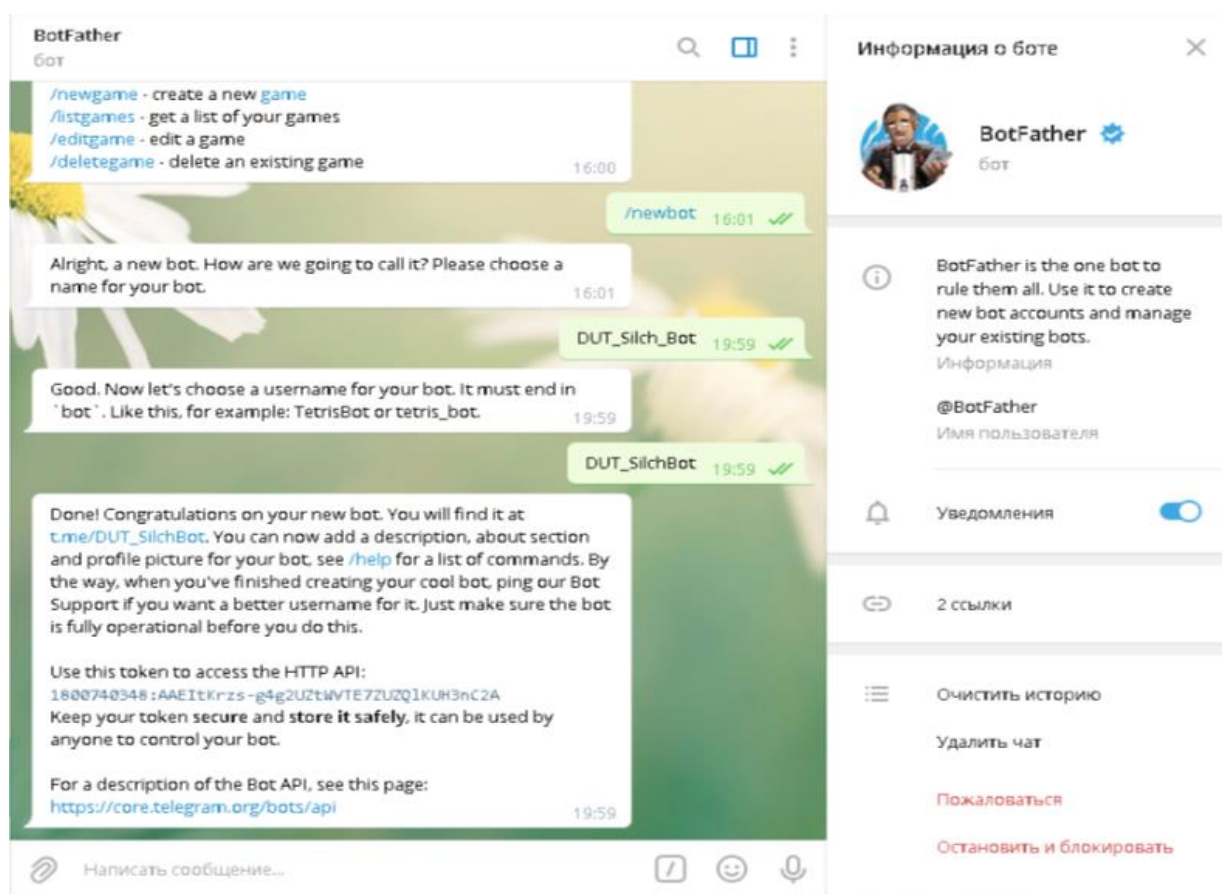


Рисунок 4.12 – Реєстрація чат-боту та отримання особистого токена

Для налаштування параметрів боту, та встановлення таких речей, як іконка для боту, опис чат-боту, повідомлення на початку праці с ботом, є такі команди (таблиця 1).

Таблиця 1- Команди для зміни основної інформації.

Команда	Опис
/setname	Заміна існуючого імені
/setdescription	Встановлення тексту, що відображається при першому відкритті чат-боту
/setabouttext	Встановлення тексту, для опису чат-боту
/setuserpic	Привласнює чат-боту обрану «іконку»
/setcommands	Дає можливість створити список доступних команд
/deletebot	Видаляє ваш чат-бот

Окрім команд для зміни основних параметрів, є додаткові, для контролю самого чат-боту (таблиця 2).

Таблиця 2 – Команди для додаткового налаштування чат-боту

Команда	Опис
/token	Генерує токен для авторизації
/revok	Відзиває ваш токен доступу до бота
/setinline	Вмикає або вимикає можливість виклику бота з інших чатів
/setinlinegeo	Отримання дозволу на запит геолокації у користувачів
/setinlinefeedback	Дозволяє отримувати інформацію про кількість обраних користувачами команд
/setjoingroup	Дозволяє додавати бот в різні чат-групи.
/setprivacy	Вмикає режим конфіденційності

Після налаштування бота за допомогою команд, та отримання токена, починаємо переходити до програмної частини створення.

4.2 Реалізація чат-боту

Взаємодія користувача з ботом починається командою «/start».

Після цього, програма має реагувати на написаний текст, або відправлене голосове повідомлення. Ця функція була реалізована на Aiogram, за допомогою перехоплювачів (message_handler). На рисунку 13, був використаний перехоплювач, що реагує на команду «/start», від користувача.

```

47 # Стартовое сообщение
48 @dp.message_handler(commands=['START'])
49 async def start(message: types.Message):
50     await message.answer('***Здравствуй***\n' +
51                          'Отправьте текст, чтобы получить аудио\n'+
52                          'или аудио, чтобы получить текст\n' +
53                          'на английском или русском языках',
54                          parse_mode=ParseMode.MARKDOWN)
55

```

Рисунок 4.13 – Приклад перехоплювача Aiogram

У відповідь на команду, чат-бот пересилає вказане повідомлення, з інструкцією, для користування. Це зображено на рисунку 14.

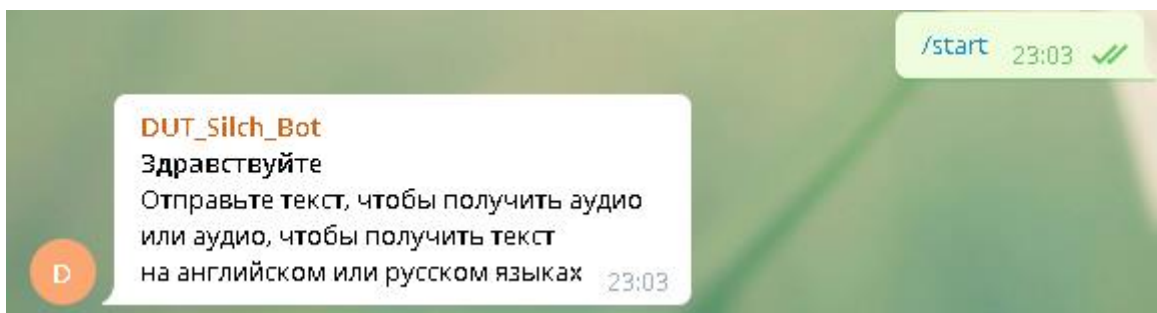


Рисунок 4.14 – Початок роботи з чат-ботом

Далі йде реалізація основних функцій чат-боту. Після того, як користувач відправить будь-який текст, перехоплювач прийме його на опрацювання. Функція в коді, зображена на рисунку 15.

```

56     # обработка текстового сообщения
57     @dp.message_handler(content_types=['text'])
58     async def check(message: types.Message):
59         text = message.text
60         lang = get_lang(text)
61         file = text_to_mp3(text, lang)
62         with open(file[0], 'rb') as audio:
63             await message.answer_audio(audio.read(), f'Язык: {lang}')
64

```

Рисунок 4.15 – Перехоплювач тексту, від користувача.

Після цього, програма має визначити мову, на якій був написаний текст, та перевести його в голосове повідомлення в форматі «.mp3». Ця функція зображена на Рисунку 16.

```

33     # получение языка текста
34     def get_lang(text):
35         return Translator().detect(text).lang
36
37     # преобразование текста в аудио
38     def text_to_mp3(text, lang):
39         speech = Speech(text, lang)
40         speech.save("output.mp3")
41         return ["output.mp3", lang]
42

```

Рисунок 4.16 – Функція переведення тексту в аудіо

Сама функція реалізується за рахунок однієї з підключених нами бібліотек, а саме API Google Speech. Технологія «синтезу мовлення», що підтримує десятки різних мов. Приклад переведення тексту в аудіо, зображений на рисунку 17.

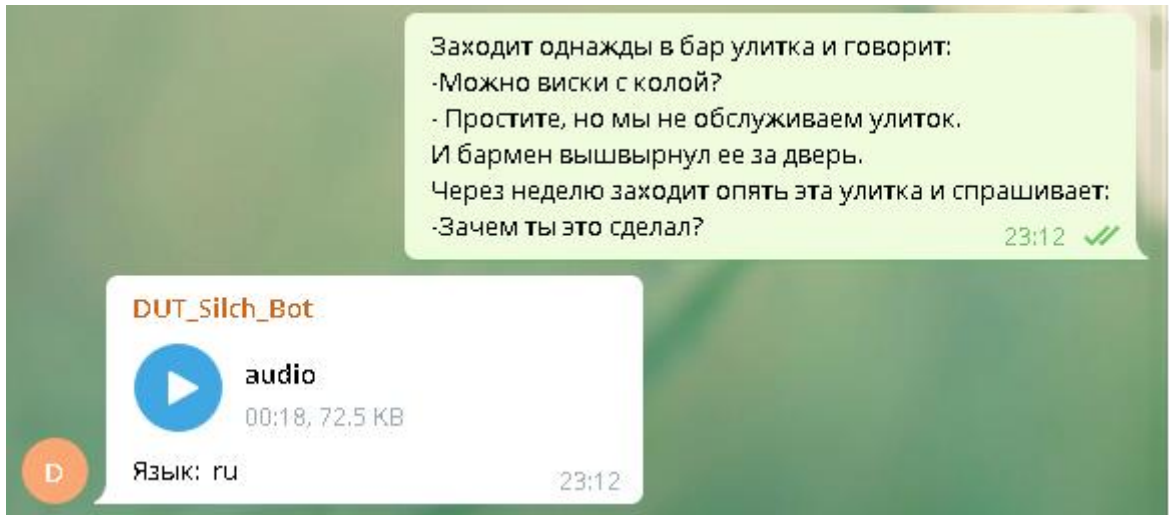


Рисунок 4.17 – Пример перевода текста в аудио

Наступною йде реалізація функція переведення аудіо в текстове повідомлення. В досягненні цієї мети, допомагає спеціалізована бібліотека Python Speech Recognition API. Це технологія, що допомагає боту в розпізнаванні слів користувача, для подальшої обробки, та переведення в текстове повідомлення.

На даний момент, чат може працювати на російській та англійській мовах.

Функція в кодї, зображена на рисунку 18.

```

19
20 # преобразование аудио файла в текст
21 def wav_to_text(audio, lang='ru'):
22     dst = 'output.wav'
23     sound = AudioSegment.from_file(audio)
24     sound.export(dst, format="wav")
25
26     r = sr.Recognizer()
27
28     with sr.AudioFile('output.wav') as source:
29         audio_data = r.record(source)
30         text = r.recognize_google(audio_data, language=lang)
31     return text
32

```

Рисунок 4.18 – Функція переведення аудіо в текст

Оскільки бібліотека працює лише з файлами у форматі «.wav», який не підходить для використання месенджером Telegram, була підключена бібліотека, що дозволяє нам перевести файл в потрібний формат «.mp3», а саме PyDub API. Використання цієї бібліотеки в коді, та обробка аудіо, зображена на Рисунок 19.

```
65 # обробка голосового сообщения или mp3 файла
66 @dp.message_handler(content_types=['audio', 'voice'])
67 async def check(message: types.Message):
68     try:
69         file_id = message.audio['file_id']
70     except:
71         file_id = message.voice['file_id']
72     file_info = await bot.get_file(file_id)
73     file_path = file_info.file_path
74     try:
75         await bot.download_file(file_path, 'input.mp3')
76         text = wav_to_text('input.mp3')
77     except:
78         await bot.download_file(file_path, 'voice.ogg')
79         text = wav_to_text('voice.ogg')
80     await message.answer(text)
81
```

Рисунок 4.19 – Обробка аудіо файлів в коді

Це дозволяє на отримувати текстове повідомлення, що було сформоване на основі голосового, що ми відправили до чат-боту. Приклад використання в чат-боті зображений на рисунку 20.

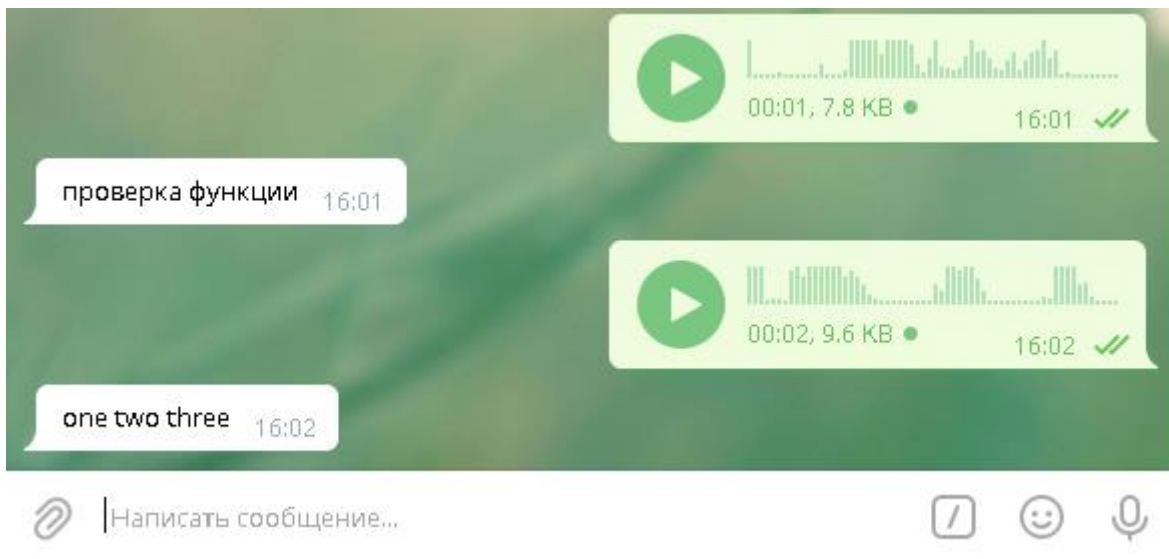


Рисунок 4.20 – Приклад використання функції переведення аудіо в текст.

Для стабільної праці, бот було розташовано, на віддалених серверах FirstVds. VDS (Virtual Dedicated Server) – це хостинг-послуга, де користувачеві надається віртуальний сервер з максимальним привілеями.

VDS емулює роботу реального фізичного сервера - є root-доступ, можлива установка своїх операційних систем і програмного забезпечення. На одному фізичному сервері зазвичай працює кілька незалежних віртуальних серверів. Робота чат-боту на сервері, зображена на рисунку 21.

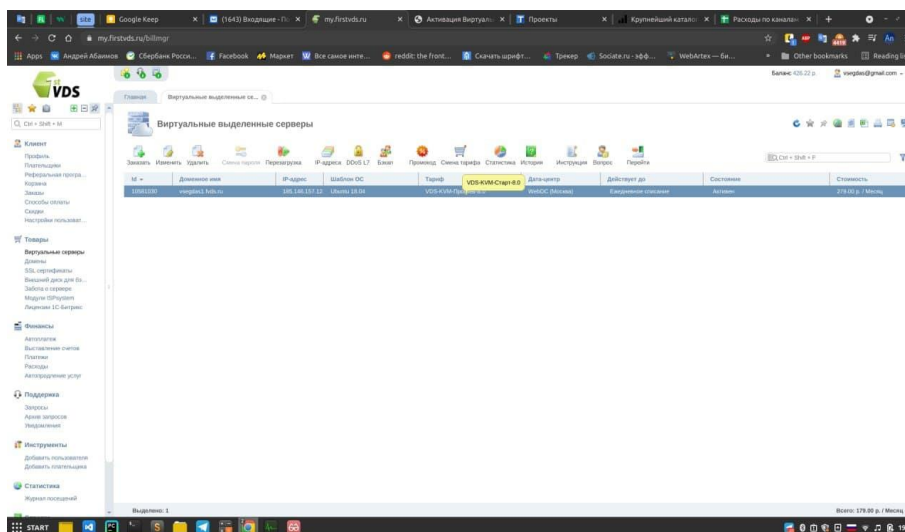


Рисунок 4.21 – Работа чат-боту на сервері FirstVds

4.2 Огляд можливостей інформаційної системи

Для покращення спілкування між користувачами, та вирішення проблеми з обмеженими можливостями, був розроблений чат-бот.

В його функціонал входить:

- Розпізнавання мови користувача;
- Вільне переведення тексту в голосове повідомлення, на більш 100 мовах світу;
- Вільне переведення голосових повідомлень в текстові, на російській та англійській мовах.
- Можливість додавати чат-бота в групи або чати, для подальшого користування.

Приклад використання чат-бота в групі зображений на рисунку 22.

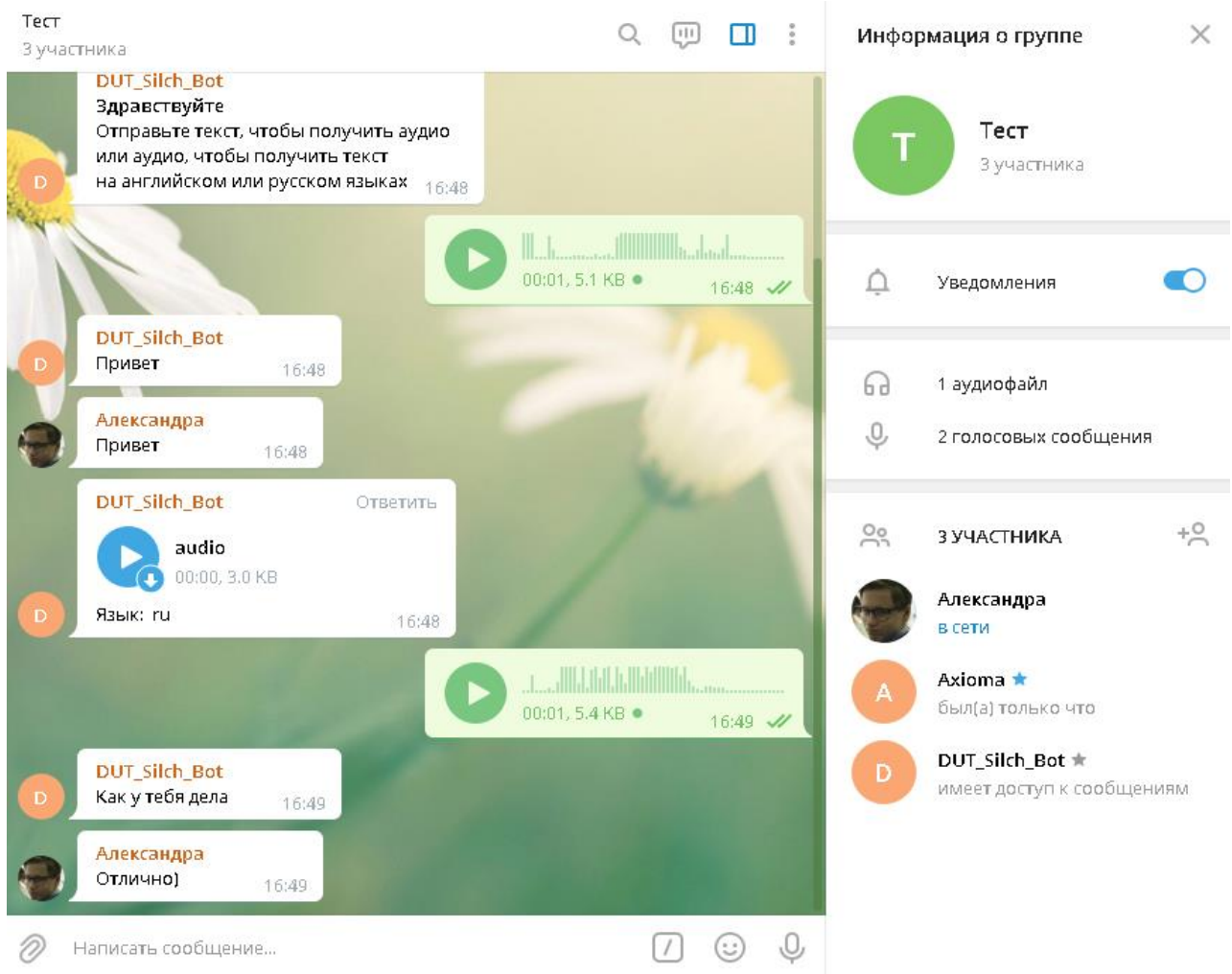


Рисунок 4.22 – Приклад використання чат-бота в групових розмовах.

Обов'язковою умовою, для використання, є надання боту доступу до повідомлень в чаті. Для цього потрібно надати йому права адміністратора.

Як ми бачимо, бот вільно виконує свої основні функції.

4.3 Повний код чат-боту у Python

```
import asyncio
from aiogram import Bot, Dispatcher, executor, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters import Text
from aiogram.dispatcher.filters.state import State, StatesGroup
from aiogram.types import ReplyKeyboardRemove, ReplyKeyboardMarkup,
KeyboardButton, InlineKeyboardMarkup, \
    InlineKeyboardButton, ParseMode
from aiogram import md
from google_speech import Speech
import google_speech
```

```

import googletrans
from googletrans import Translator
import speech_recognition as sr
from os import path
from pydub import AudioSegment

# преобразование аудио файла в текст
def wav_to_text(audio, lang='ru'):
    dst = 'output.wav'
    sound = AudioSegment.from_file(audio)
    sound.export(dst, format="wav")

    r = sr.Recognizer()

    with sr.AudioFile('output.wav') as source:
        audio_data = r.record(source)
        text = r.recognize_google(audio_data, language=lang)
        return text

# получение языка текста
def get_lang(text):
    return Translator().detect(text).lang

# преобразование текста в аудио
def text_to_mp3(text, lang):
    speech = Speech(text, lang)
    speech.save("output.mp3")
    return ["output.mp3", lang]

# инициализируем бота
bot = Bot(token='1800740348:AAEItKrzS-g4g2UZtWVTE7ZUZQlKUH3nC2A') # токен бота
dp = Dispatcher(bot, storage=MemoryStorage())

# Стартовое сообщение
@dp.message_handler(commands=['START'])
async def start(message: types.Message):
    await message.answer('***Здравствуйте***\n' +
        'Отправьте текст, чтобы получить аудио\n'+
        'или аудио, чтобы получить текст\n' +
        'на английском или русском языках',
        parse_mode=ParseMode.MARKDOWN)

# обработка текстового сообщения
@dp.message_handler(content_types=['text'])
async def check(message: types.Message):
    text = message.text
    lang = get_lang(text)
    file = text_to_mp3(text, lang)
    with open(file[0], 'rb') as audio:
        await message.answer_audio(audio.read(), f'Язык: {lang}')

# обработка голосового сообщения сообщения или mp3 файла
@dp.message_handler(content_types=['audio', 'voice'])
async def check(message: types.Message):
    try:
        file_id = message.audio['file_id']
    except:
        file_id = message.voice['file_id']
    file_info = await bot.get_file(file_id)
    file_path = file_info.file_path
    try:
        await bot.download_file(file_path, 'input.mp3')
        text = wav_to_text('input.mp3')

```

```
except:
    await bot.download_file(file_path, 'voice.ogg')
    text = wav_to_text('voice.ogg')
    await message.answer(text)

# запуск бота
if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True)
```

ВИСНОВОК

В сучасний час популярність чат-ботів продовжує зростати. Цих помічників використовують в різних цілях, від звичайного спілкування, до проведення фінансових операцій. Потрібно відзначити, що свій розвиток вони отримали саме завдяки месенджерам. З'явилися різні види ботів, для самих різних цілей.

Основна задача чат-ботів – вести діалог з користувачем, відповідати на його запити, орієнтуючись на особисту базу, закладену під час створення.

Зараз технології дають можливість даже для створювати ботів, які будуть імітувати штучний інтелект. Або ж розпізнавати людську мову, в спілкуванні з «машиною».

Проаналізовано, що чат-боти можуть виконувати різноманітні задачі: проводити опитування, консультувати людей, надавати інформацію по потрібним об'єктам, повідомляти про новини, замовляти їжу, слідкувати за заторами на дорогах, проводити синтез мови і все це в рамках одного додатку, на платформі якого, реалізують бот. Це зменшує потребу в встановленні десятків різних додатків, за причиною того, що майже всі їх функції, можуть виконувати чат-боти в одному месенджері.

В теперішній час, всі великі компанії намагаються повністю перевести спілкування з клієнтом в формат відправки повідомлення, користуючись месенджерами.

В межах випускної роботи, були виконані поставлені задачі.

В першу чергу, ми детально дослідили месенджери, якими користуються більшість користувачів в інтернеті. Було проведено дослідження різних чат-ботів аналогів, що мали схожі функції.

На основі цього, ми виявили основні вимоги, для розробки чат-боту з функцією «синтез мови», для людей з обмеженими можливостями.

Після виявлення вимог, та проектування, були обрані технології, та середовище для розробки.

Таким чином, результатом цієї випускної роботи, є створення чат-боту, що

допомагає користувачам у спілкуванні між собою. Якщо відправити боту текстове повідомлення, або якщо воно з'явиться під час розмови, в групі з чат-ботом, він переведе його в голосове повідомлення, що дуже корисно для людей з вадами зору, або ж для тих, хто не має часу та можливості для того, щоб читати такі повідомлення. Якщо ж відправити боту голосове повідомлення, він переведе його в зручний текстовий формат. Це в свою чергу також буде використовуватись людьми, у яких не має можливості прослуховувати голосові повідомлення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1) Welcome to aioqram's documentation! [Електронний ресурс]: <https://docs.aioqram.dev/en/latest/>
- 2) PyCharm Wiki [Електронний ресурс]: - Режим доступу: <https://uk.wikipedia.org/wiki/PyCharm>
- 3) Google Speech [Електронний ресурс]: - Режим доступу: <https://pypi.org/project/google-speech/>
- 4) Pydub [Електронний ресурс]: - Режим доступу: <https://github.com/jiaaro/pydub>
- 5) Lanet VPS [Електронний ресурс]: - Режим доступу: <https://lanet.business/cloud/vps/>
- 6) Офіційний сайт Aioqram [Електронний ресурс]: Документація – Режим доступу: <https://docs.aioqram.dev/en/latest>.
- 7) Найкорисніші боти у Telegram [Електронний ресурс]: – Режим доступу: <https://www.vodafone.ua/shop/ua/blog/ne-lishe-dlja-spilkuvannja-najkorisnishi-botiu-telegram.html>
- 8) Most popular global mobile messenger apps [Електронний ресурс]: – Режим доступу: <https://www.statista.com/statistics/258749/most-popular-globalmobilemessenger-apps/>
- 9) Еволюція месенджерів [Електронний ресурс]: – Режим доступу: <https://nachasi.com/2017/11/13/evolyutsiya-mesendzheriv/>
- 10) Welcome to python.org [Електронний ресурс]: – Режим доступу: <https://www.python.org/>
- 11) Chatbots: Are they Really Useful? [Електронний ресурс]: – Режим доступу: https://jlcl.org/content/2-allissues/20-Heft1-2007/Bayan_Abu-Shawar_and_Eric_Atwell.pdf
- 12) Telegram Revenue and Usage Statistics (2020) [Електронний ресурс]: – Режим доступу: <https://www.businessofapps.com/data/telegram-statistics/>
- 13) Офіційний сайт Aioqram [Електронний ресурс]: <https://docs.aioqram.dev/en/latest>

- 14) Welcome to pysox's documentation! [Электронный ресурс]: – Режим доступа: <https://pysox.readthedocs.io/en/latest/#welcome-to-pysox-s-documentation>
- 15) A history of Instant Messaging and Chat [Электронный ресурс]: – Режим доступа: <https://www.maize.io/news/lizshemaria-historyof-instant-messaging/>
- 16) История мессенджеров: первая волна [Электронный ресурс]: – Режим доступа: <https://www.astrosoft.ru/articles/unified-communications/istoriya-messenzherov-pervaya-volna/>
- 17) Working with wav files in Python using Pydub [Электронный ресурс]: – Режим доступа: <https://www.geeksforgeeks.org/working-with-wav-files-in-python-using-pydub/>
- 18) PYSOX: LEVERAGING THE AUDIO SIGNAL PROCESSING POWER OF SOX IN PYTHON [Электронный ресурс]: – Режим доступа: <https://s18798.pcdn.co/ismir2016/wpcontent/uploads/sites/2294/2016/08/bittner-pysox.pdf>