

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота
на здобуття освітнього рівня бакалавра**

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

**РОЗРОБКА АДАПТИВНОГО ПОШУКОВОГО СЕРВІСУ
МУЗИЧНИХ КОМПОЗИЦІЙ НА БАЗІ ПЛАТФОРМИ TELEGRAM**

Виконав студент 4-го курсу
Дмитро ШВАБ

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Максим ВЕРЕС

(підпис)

Засвідчую, що в цій курсовій роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до
захисту
на засіданні кафедри інтелектуальних
програмних систем

« ____ » _____ 2021 р.,

протокол № ____

Завідувач кафедри

Олександр ПРОВОТАР

(підпис)

Київ – 2021

РЕФЕРАТ

Обсяг роботи 53 сторінки, 26 ілюстрацій, 4 таблиці, 21 джерело посилання.

КЛЮЧОВІ СЛОВА: БОТ, ЧАТ-БОТ, ТЕЛЕГРАМ-БОТ, ІНТЕРФЕЙС ПРОГРАМНОГО ПРОДУКТУ, МЕСЕНДЖЕР, TELEGRAM, МУЗИКА, МУЗИЧНА КОМПОЗИЦІЯ, ПОШУК.

Об'єктом роботи є полегшення доступу до світової бібліотеки музичних композицій через платформу Telegram, зручний пошук та отримання пісень. Предметом роботи є багатопоточний чат-бот на базі платформи Telegram для завантаження потрібних пісень за прийнятний для цього час.

Метою роботи є створення та розробка Telegram-бота для полегшеного доступу до світової бібліотеки музичних композицій.

Методи розробки: проектування програмного забезпечення за допомогою засобів автоматизованої розробки програмного забезпечення та використання стандартів описів формату, такі як Уніфікована Мова Моделювання (UML). Інструменти розроблення: безкоштовне, вільно поширюване інтегроване середовище розробки Microsoft Visual Studio 2019, мова програмування C#.

Результат роботи: виконано поглиблений огляд механізмів та інструментів для організації багатопотокового додатку, проаналізовано недоліки та переваги використання платформи Telegram для реалізації зручного доступу до музичних композицій, проаналізовано аналоги чат-ботів зі схожим функціоналом, поглиблено знання у застосуванні інструментів Telegram Bot Api та Entity Framework мови програмування C#, розроблено програмний продукт «Muz-z-z», який дозволяє досить швидко завантажити улюблену пісню на пристрій для майбутнього прослуховування її в он-лайн та оф-лайн режимі.

Програмний продукт «Muz-z-z» може використовуватися великою кількістю користувачів через механізми відмовостійкості для зручного та швидкого пошуку та завантаження будь-яких музичних композицій на пристрій користувача для майбутнього її прослуховування та розповсюдження.

Ця програмна система може бути доповнена механізмами впровадження рекламних інтеграцій для монетизації використання чат-бота великою кількістю користувачів та механізмами більш швидкого поширення серед людей, наприклад заохочення користувачів до реклами чат-бота через надання привілейованого режиму.

ЗМІСТ

	С.
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	6
ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ	10
1.1 Месенджери	10
1.2 Чат-боти	13
РОЗДІЛ 2 ЗАСОБИ РОЗРОБКИ	15
2.1 Мова програмування C#	15
2.2 IDE Visual Studio 2019	16
2.3 Telegram Bot Api	21
2.4 Microsoft Sql Server	24
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	26
3.1 Реєстрація чат-боту на платформі Telegram	26
3.2 Програмна реалізація додатку	28
3.2.1 Реалізація основних компонентів чат-боту	28
3.2.2 Реалізація вітання чат-боту	30
3.2.3 Реалізація пошуку та завантаження музичних композицій	32
3.2.4 Реалізація взаємодії з плейлістами	37
3.2.5 Реалізація завантаження аудіоряду з кліпів платформи YouTube	40
3.2.6 Реалізація підтримки In-line mode	41
3.2.7 Реалізація додаткового функціоналу	43
3.2.7.1 Реалізація збереження історії запитів користувачів	43
3.2.7.2 Реалізація зміни мови користувача	43
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47

ЗМІСТ

	С.
ДОДАТОК А. Порівняння існуючих месенджерів	49
ДОДАТОК Б. Порівняння існуючих пошукових сервісів	51
ДОДАТОК В. Список існуючих команд чат-боту Muz-z-z	53

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

AI – Artificial intelligence, штучний інтелект

IDE – Integrated Development Environment, інтегроване середовище розробки

ООП – Об'єктно-орієнтоване програмування

МП – Мова програмування

VS – Visual Studio

СКБД – Система керування базою даних

SSMS – Sql Server Management Studio

MSS – Microsoft Sql Server

ОС – Операційна система

ВСТУП

Оцінка сучасного стану об'єкта розробки. Так як сучасний світ не стоїть на місці, технології стрімко розвиваються та кожного дня з'являються нові тренди та ідеї, багато з них мають можливість впливати на ІТ-галузь, яка в свою чергу впливає на інші галузі, а вони впливають на наше повсякденне життя. До недавнього часу популярними були додатки та комп'ютерні програми, але зараз лідерство займають месенджери та чат-боти. Останні мають великі перспективи в майбутньому.

Для початку треба розібратися, що являє собою чат-бот. Чат-бот – це спеціалізований додаток-помічник, що дозволяє користувачам дуже зручно взаємодіяти зі сторонніми сервісами через інтерфейс чату заданого месенджеру.

Зазвичай чат-бот спілкується з користувачами через повідомлення і має перелік певних функцій. Тобто, можна отримати конкретну інформацію, написавши чат-боту спеціальну команду, яку в свою чергу останній обробить певним чином та дасть відповідь. Таким чином можна дуже швидко взаємодіяти з іншими сервісами і платформами, які розширили свій інтерфейс взаємодії на інші месенджери.

AI (Artificial intelligence) в загальному використовується для скорочення витрат і підвищення якості послуг. Найбільш популярними напрямками розвитку AI є розпізнавання голосових записів та переведення їх в текстовий вигляд і текстове спілкування за допомогою різних пошукових ботів. Наприклад можна згадати про всім відомі спливаючі віконця чатів "консультантів" на багатьох сайтах, які використовують штучний інтелект. Також яскравим прикладом використання AI можна назвати величезну кількість ботів для прийому заявок на

доставку їжі, бронювання місць в ресторанах, швидке звернення до служби підтримки і багато іншого.

Такі боти дозволяють збільшити прибутки компаній, які інтегрували свої сервіси на платформи відомих месенджерів, оскільки бот може обробляти заявки швидше ніж людина, і зменшити навантаження на робочий персонал.

Деякі люди відчувають потребу у прослуховуванні музичних композицій. Наприклад під час перебування у громадському транспорті, при подорожі на великі відстані, де не має доступу до мережі Інтернет або просто для морального відпочинку після важкого дня.

Тому, для задоволення цих потреб було вирішено створити адаптивний пошуковий сервіс музичних композицій на базі платформи Telegram. Він являтиме собою чат-бот зі зручним інтерфейсом взаємодії з користувачами.

Telegram – це багатоплатформний додаток, розроблений на мові програмування C++, що дозволяє обмінюватися повідомленнями і файлами невеликого розміру більшості форматів між користувачами. Месенджер використовує спеціально розроблену серверну частину з закритим кодом, які працюють на серверах багатьох країн. Telegram відомий своєю конфіденційністю, дуже великою швидкістю доставки повідомлень, можливістю кешування та завантаження отриманих файлів собі на пристрій, відсутністю реклами.

Актуальність роботи та підстави для її виконання. Актуальність бакалаврської кваліфікаційної роботи зумовлена високою популярністю месенджерів і таких засобів автоматизації як чат-боти серед користувачів мережі Інтернет. Головною перевагою перед класичними додатками є можливість об'єднання інтерфейсів всіх сервісів в один інтерфейс месенджеру.

Мета й завдання роботи. За мету було взято проектування та розробку чат-бота на платформі Telegram, який зможе за запитом користувачів швидко

знаходити та завантажувати музичні композиції на пристрій для майбутнього прослуховування. Також потрібно було створити додатковий функціонал, такий як можливість завантаження аудіоряду з кліпів відеоплатформи YouTube, можливість створення плейлістів, позначення пісень як «Улюблені», підтримка чат-ботом in-line моду, отримання списку популярних пісень на поточний час, зміна мови, запис історії користувача.

Об'єкт, методи й засоби розроблення. Об'єктом роботи є полегшення доступу до світової бібліотеки музичних композицій через платформу Telegram, зручний пошук та отримання пісень.

Засобами розробки була обрана мова програмування C#, IDE Microsoft Visual Studio 2019, технологія Telegram Bot Api, Entity Framework та система керування базою даних Microsoft SQL Server.

Взаємозв'язок з іншими роботами. За методами розробки та інструментальними засобами робота виконувалася сумісно з роботами по розробці чат-ботів на платформі Telegram.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Месенджери

Свій розвиток Інтернет-сервіси для спілкування почали з чатів, потім месенджери, потім перейшли на соціальні мережі, але нещодавно вони знову очолили список найбільш перспективних сервісів. Це підтвердило дослідження компанії «Білайн» в 2017 році: кількість користувачів месенджерів в країнах СНГ збільшилася на 7%. Одночасно з цим середня кількість месенджерів, які використовуються одним користувачем, збільшилася за 2017 рік з 1,73 до 3 штук. Месенджери використовуються переважно дорослими, так на частку осіб до 18 років припадає 7,7% від загальної кількості аудиторії, користувачі у віці від 18 до 25 років складають 9,8%, від 25 до 35 років – 32,3%, від 35 до 45 років – 26,8%. При цьому 13,1% користувачів месенджерів знаходяться у віці від 45 до 55 років, а 7,6% – від 55 до 64 років. І лише 2,7% аудиторії месенджерів знаходиться в віці більше 64 років [1].

Причина повторної хвилі популярності месенджерів – зміни в області мобільного Інтернету: високі швидкості обміну даних, широке поширення мобільних пристроїв. Кріс Мессіна в статті «2016 will be the year of conversational commerce» [2] назвав 2016 рік – роком діалогової або розмовної комерції. Розуміючи під цим поняттям чати, месенджери та інші інтерфейси, які забезпечують взаємодію людей з сервісами різних компаній. Кінцевим результатом він бачить те, що користувачі будуть спілкуватися з компаніями через месенджери. При цьому, на його думку, неважливо, з ким або з чим будуть спілкуватися люди – зі співробітником компанії, з ботом або з їх комбінацією.

Всі сучасні месенджери мають безліч схожих функцій, таких як відправка повідомлень, медіафайлів, аудіо- та відео спілкування. В даний час існує велика

їх кількість та всі вони мають свої переваги та недоліки (Додаток А). На особливу увагу заслуговує месенджер Telegram.

Засновниками месенджеру Telegram є брати Павло та Микола Дурови. Павло є фінансовою і ідеологічною опорою проекту, а Микола зайнятий технічними аспектами. Особливостями даного месенджеру є секретні чати і спеціально розроблений протокол шифрування MTProto.

Секретні чати призначені для людей, які хочуть забезпечити себе найбільш безпечним спілкуванням. Окрім використання основного протоколу, усі повідомлення шифруються ключами за принципом від пристрою до пристроїв. Ніхто не перехоплює і не розшифровує повідомлення, в тому числі співробітники Telegram. Повідомлення з таких чатів не можна пересилати і вони не можуть зберігатися на серверах Telegram. Але секретні чати мають два недоліки – це неможливість відкрити чат на пристрої, який не брав участі при його створенні, і ніхто не захищений від знімку екрану інтерфейсу Telegram з відкритим таємним чатом.

Месенджер Telegram доступний для всіх популярних платформ, наприклад, ОС Android, iOS, Windows Phone, Linux та інші. Також у більшості версій для різних операційних систем доступні англійська, російська, польська та німецька мови. Приклади інтерфейсу настільного додатку в Windows 10 показано на рисунку 1.

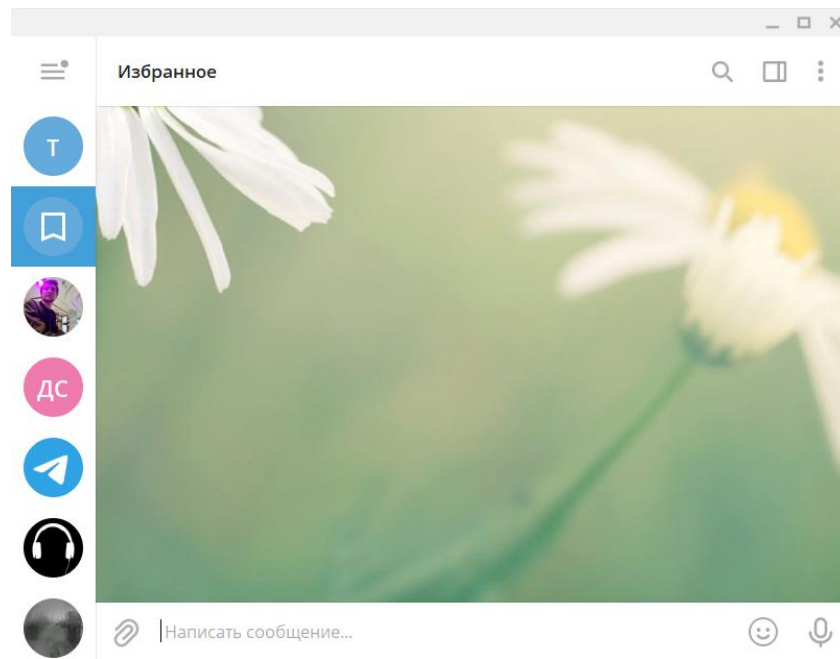


Рисунок 1. Інтерфейс клієнту Telegram на ОС Windows 10

Останнім часом компанія Telegram активно розробляє свою блокчейн-платформу Telegram Open Network та криптовалюту Gram на її основі.

1.2 Чат-боти

Останнім часом набирає популярності в месенджерах таке явище, як чат-боти. Сам термін "чат-бот" був введений Майклом Молдіном у 1994 р. для опису розмовних програм. Проаналізувавши тенденцію до використання чат-ботів у месенджерах, ми можемо дійти висновку, що вони є універсальними інструментами, здатними вирішувати різноманітні завдання – від спілкування до розваг, від надання медичної консультації до замовлень товарів та послуг за допомогою спеціалізованих прикладних додатків, від визнання вподобань користувача до вирішення складних консультаційних проблем у службах підтримки клієнтоорієнтованими системами. У будь-якому випадку, незалежно від платформи, чат-бот являє собою прикладну програму, яка, отримуючи

інформацію від користувача, генерує правильні, логічні відповіді, закладені розробником.

Сьогодні існує безліч різноманітних чат-ботів. Деякі з можливих варіантів представлені нижче:

- ігрові чат-боти (квести, рольові ігри);
- реклама чат-ботів (Hrusteam, Coca-Cola);
- чат-боти новин ("Коммерсант", "Медуза");
- чат-боти для доставки, магазинів, послуг (таксі "Максим", піцерія "Papa John's", бронювання поїздів і літаків "Tutu.rf");
- чат-боти для консультацій та підтримки клієнтів (банк Точка, каршерінг "UrentCar") та інші.

Останнім часом великі можливості відкривають інформаційні технології, пов'язані з чат-ботами. Вони міцно увійшли в життя людини тим, що стали використовуватися у всіх сферах людської діяльності, і їхня роль з кожним днем зростає. Перш за все це пояснюється тим, що люди проводять більшу частину свого часу в своєму мобільному пристрої, у поштових клієнтах та месенджерах.

Чат-боти – це програми, які виконують різні запити користувачів, які, в свою чергу, здатен обробляти бот. Він виглядає як звичайний чат, однак спілкування відбувається не з людиною, а з програмою, яка, наприклад, може обробляти замовлення на таксі, якщо це каршерінговий бот, або відправить останнє зведення новин, якщо це новинний бот, або замовить вам доставку їжі додому, якщо це – ресторанний бот. Не так давно чат-боти стали дуже популярними, перевтілилися з розваги в більш серйозну річ, оскільки їх, головним чином, стали використовувати для вирішення серйозних ділових проблем по спрощенню взаємодії компаній та користувачів. В епоху інформаційних технологій – це нормальне явище, і тим більше – в мережі Інтернет, бо суспільство

вже перейшло на мережевий бізнес. І чат-боти стали дуже непоганою платформою для вирішення бізнес-проблем. Адже вони підтримують діалог з користувачем, який вибирає відповіді з бази даних, заздалегідь відомі: ви запитаете, де поведеряти і миттєво отримаєте відповідь. Крім того, чат-боти виконують багато рутинних операцій, які можна автоматизувати, наприклад: пошук інформації, поверхнева обробка даних, обслуговування клієнтів.

Чат-бот, як віртуальний співрозмовник, має базу знань – це набір можливих запитів користувача та відповіді на них. Найпоширеніші варіанти для отримання бажаної відповіді – це ключові слова, фрази або контекст. Завжди є якесь просте і легке програмне забезпечення, яке здатне робити речі, на які не хочеться витратити час. Тут в нагоді стали чат-боти. Вік друкарських машинок, прогулянок по бібліотеках і нескінченні черги в книжкові магазини відійшли в забуття. Тепер щодо збору інформації можна використовувати чат-боти. Звичайно, це зроблено для спілкування з людьми, але не тільки. Наприклад, в межах будь-якої події чат-боти можуть інформувати всіх учасників новин і надавати довідкову інформацію не спілкуючись з ними.

В межах цієї роботи були також проаналізовані переваги та недоліки існуючих музичних чат-ботів. Порівняння пошукових сервісів на платформі Telegram було наведено у додатку Б.

РОЗДІЛ 2 ЗАСОБИ РОЗРОБКИ

2.1 Мова програмування C#

На сьогоднішній момент МП C# одна з найпотужніших і затребуваних мов в IT-галузі. Сьогодні на ній пишуться найрізноманітніші програми: від невеликих десктопних до великих веб-порталів і веб-сервісів, які обслуговують щодня мільйони користувачів.

C# вже не молода мова, як і вся платформа .NET, вона вже пройшла великий шлях. Перша версія мови вийшла разом з релізом Microsoft Visual Studio .NET в лютому 2002 року. Поточною версією мови є версія C# 9.0, яка вийшла 10 листопада 2020 року разом з релізом .NET 5.

C# є мовою з Сі-подібним синтаксисом і близька щодо цього до C++ і Java. Це значно облегшує опанування цієї мови іншими розробниками, які володіють даними або схожими мовами.

C# є об'єктно-орієнтованою і в цьому плані багато перейняла у Java і C++. Наприклад, C# підтримує поліморфізм, успадкування, перевантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків. C# продовжує активно розвиватися, і з кожною новою версією з'являється все більше цікавих функціональних можливостей, як, наприклад, лямбда, динамічне зв'язування, асинхронні методи і т.д.

Код на C# має двоетапну компіляцію. Спочатку він компілюється в додаток або збірку з розширеннями exe або dll на мові CIL. Далі при запуску виконання подібної програми відбувається JIT-компіляція (Just-In-Time) в машинний код, який потім виконується комп'ютером. При цьому, оскільки наш додаток може бути великим і містити купу інструкцій, в поточний момент часу компілюватиметься лише та частина програми, до якої безпосередньо йде

звернення. Якщо ми звернемося до іншої частини коду, то вона буде скомпільована з CIL в машинний код. При цьому вже скомпільована частина програми зберігається до завершення роботи програми. У підсумку це підвищує продуктивність.

2.2 Інтегроване середовище розробки Visual Studio 2019

IDE Visual Studio – це стартовий майданчик для написання, налагодження і складання коду, а також подальшої публікації додатків. IDE являє собою багатофункціональну програму, яку можна використовувати для різних аспектів розробки програмного забезпечення. Крім стандартного редактора і відладчика, які існують в більшості середовищ IDE, Visual Studio включає в себе компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для спрощення процесу розробки.

Нижче перераховані деякі популярні можливості Visual Studio, які допоможуть вам підвищити продуктивність розробки програмного забезпечення.

- Хвилясті лінії і швидкі дії

Хвилясті лінії позначають помилки або потенційні проблеми коду прямо під час введення. Ці візуальні підказки дозволяють усувати проблеми негайно і не чекати, поки помилка буде виявлена під час збирання або запуску програми. Якщо навести курсор миші на хвилясту лінію, на екран будуть виведені додаткові відомості про помилку. Крім того, в поле зліва може з'явитися значок лампочки зі швидкими діями щодо усунення помилки.

- Очищення коду

Ви можете одним натисканням кнопки відформатувати код і застосувати до нього виправлення, запропоновані параметрами стилю коду, угодами в файлі EditorConfig і (або) аналізаторами Roslyn. Очищення коду допомагає усунути

багато проблем в кодї ще до перевірки коду. (Зараз ця можливість доступна тільки для коду на C #.)

- Рефакторинг

Рефакторинг включає в себе такі операції, як інтелектуальне перейменування змінних, витягування однієї або декількох рядків коду в новий метод, зміна порядку параметрів методів і багато іншого.

- IntelliSense

IntelliSense – це набір функцій, що відображають відомості про код безпосередньо в редакторі і в деяких випадках автоматично створюють невеликі уривки коду. По суті, це базова документація, вбудована в редактор, з якою вам не доводиться шукати інформацію десь ще. Функції IntelliSense залежать від мови. Додаткові відомості можна знайти у посібниках з IntelliSense для C #, IntelliSense для Visual C ++, IntelliSense для JavaScript і IntelliSense для Visual Basic.

- Пошук в Visual Studio

Середовище Visual Studio може здатися складним, адже там стільки різних меню, параметрів і властивостей. Щоб швидко знаходити функції інтегрованого середовища розробки та елементи коду, в Visual Studio представлений єдиний компонент пошуку (CTRL + Q).

- Live Share

Надає можливість спільного редагування та налагодження в реальному часі незалежно від типу додатка або мови програмування. Ви можете миттєво і безпечно поділитися своїм проектом і, при необхідності, сеансами налагодження, екземплярами терміналів, веб-додатками localhost, голосовими дзвінками і багатьом іншим.

- Ієрархія викликів

У вікні Ієрархія викликів показані методи, що викликають обраний метод. Це може бути корисно, якщо ви збираєтеся змінити або видалити метод або хочете відстежити помилку.

- CodeLens

CodeLens допомагає знаходити посилання на код, зміни коду, пов'язані помилки, робочі елементи, перевірки коду та модульні тести – все це не виходячи з редактора.

- Перейти до визначення

З функцією "Перейти до визначення" ви безпосередньо переходите туди, де визначена функція або тип.

- Показати до визначення

У вікні Перегляд визначень показано визначення методу або типу, при цьому не потрібно відкривати окремий файл.

Також Visual Studio підтримує інші МП, а саме:

- C++
- Python
- JavaScript
- TypeScript
- F#
- Visual Basic

2.3 Технологія Telegram Bot Api

Bot API представляє собою HTTP-інтерфейс для роботи з ботами в Telegram. Кожен бот – це спеціальний обліковий запис, створений для

автоматичного оброблення та відправлення повідомлень. Існує два протилежних за логікою способу отримання оновлень від бота:

- long pulling – додаток автоматично опитує сервера Telegram на наявність будь-яких оновлень для бота. За замовчуванням 100мс.
- webhook – сервера Telegram самі сповіщають додаток на сервері як тільки з'являться будь-які оновлення.

Вхідні оновлення будуть зберігатися на сервері до тих пір, поки їх не оброблять, але не довше 24 годин. Незалежно від способу отримання оновлень, у відповідь відправляється об'єкт Update, серіалізований в форматі JSON.

Всі запити до Telegram Bot API повинні здійснюватися через HTTPS в наступному вигляді: https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ.

Принцип роботи взаємодії чат-бота і користувача зображений на рисунку 2.

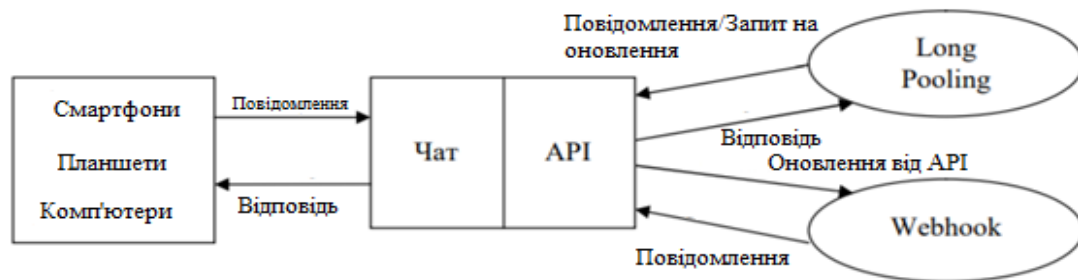


Рисунок 2. Принцип роботи чат-бота на платформі Telegram

Для того щоб створити та отримати токен для зв'язування додатку з ботом, необхідно написати спеціальному боту @BotFather (рисунок 3).



Рисунок 3. Приклад створення Telegram боту через спеціального бота @BotFather

Приклади доступних для API методів описані нижче:

- getUpdates – цей метод використовується для отримання оновлень по технології long polling;
- setWebhook – метод прив'язує до боту url домену, де міститься запущений бот;
- sendMessage – метод відправляє текстове повідомлення в клієнт Telegram користувачеві;
- sendLocation – метод відправляє повідомлення з координатами в клієнт Telegram користувачеві;
- getFile – метод повертає завантажений файл по його імені або id.

Допускаються POST і GET запити. Для передачі параметрів в Bot API є 4 способи:

- запит в URL форматі;
- application / x-www-form-urlencoded;

- application / json (не прийнятний для завантаження файлів);
- multipart / form-data (для завантаження файлів).

Але для більш зручної та швидкої розробки додатку було прийнято рішення використовувати створену бібліотеку від розробників Telegram Bot, яку можна зручно завантажити через диспетчер пакетів NuGet. Підключення бібліотеки відображено на рисунку 4.

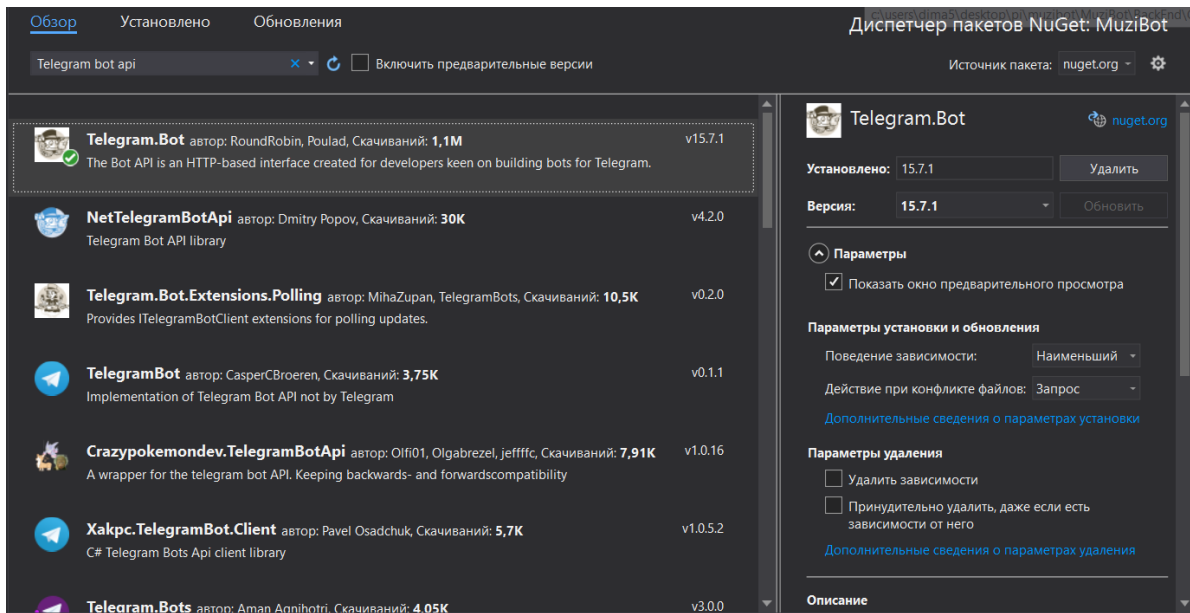


Рисунок 4. Завантаження бібліотеки Telegram.Bot через диспетчер пакетів NuGet

2.4 Microsoft SQL Server

Microsoft SQL Server є однією з найбільш популярних систем управління базами даних (СКБД) в світі. Дана СКБД підходить для різноманітних проєктів: від невеликих додатків до великих високонавантажених проєктів.

MSS був створений компанією Microsoft. Перша версія вийшла в 1987 році. А поточною версією є версія 2019, яка вийшла в 2019 році і яка буде використовуватися в поточній роботі.

SQL Server довгий час був винятково системою управління базами даних для Windows, проте починаючи з версії 16 ця система доступна і на Linux.

SQL Server характеризується такими особливостями як:

- Продуктивність. MSS працює дуже швидко.
- Надійність і безпека. MSS надає шифрування даних.
- Простота. З даної СКБД відносно легко працювати і вести адміністрування.

Центральним аспектом в MS SQL Server, як і в будь-якій СКБД, є база даних. База даних являє собою сховище даних, організованих певним способом. Нерідко фізично база даних представляє файл на жорсткому диску, хоча це не обов'язково. Для зберігання і адміністрування баз даних застосовуються системи керування базами даних або СКБД.

Для організації баз даних MS SQL Server використовує реляційну модель. Ця модель баз даних була розроблена ще в 1970 році Едгаром Коддом. А на сьогоднішній день вона фактично є стандартом для організації баз даних.

Реляційна модель передбачає зберігання даних у вигляді таблиць, кожна з яких складається з рядків і стовпців. Кожен рядок зберігає окремий об'єкт, а в стовпчиках розміщуються атрибути цього об'єкта.

Для взаємодії з базою даних застосовується мова SQL (Structured Query Language). Клієнт (наприклад, зовнішня програма) відправляє запит на мові SQL за допомогою спеціального API. СКБД належним чином інтерпретує і виконує запит, а потім посилає клієнту результат виконання.

Виділяються два різновиди мови SQL: PL-SQL і T-SQL. PL-SQL використовується в таких СКБД як Oracle і MySQL. T-SQL (Transact-SQL) застосовується в MSS.

Інтерфейс SSMS наведено на ілюстрації 5.

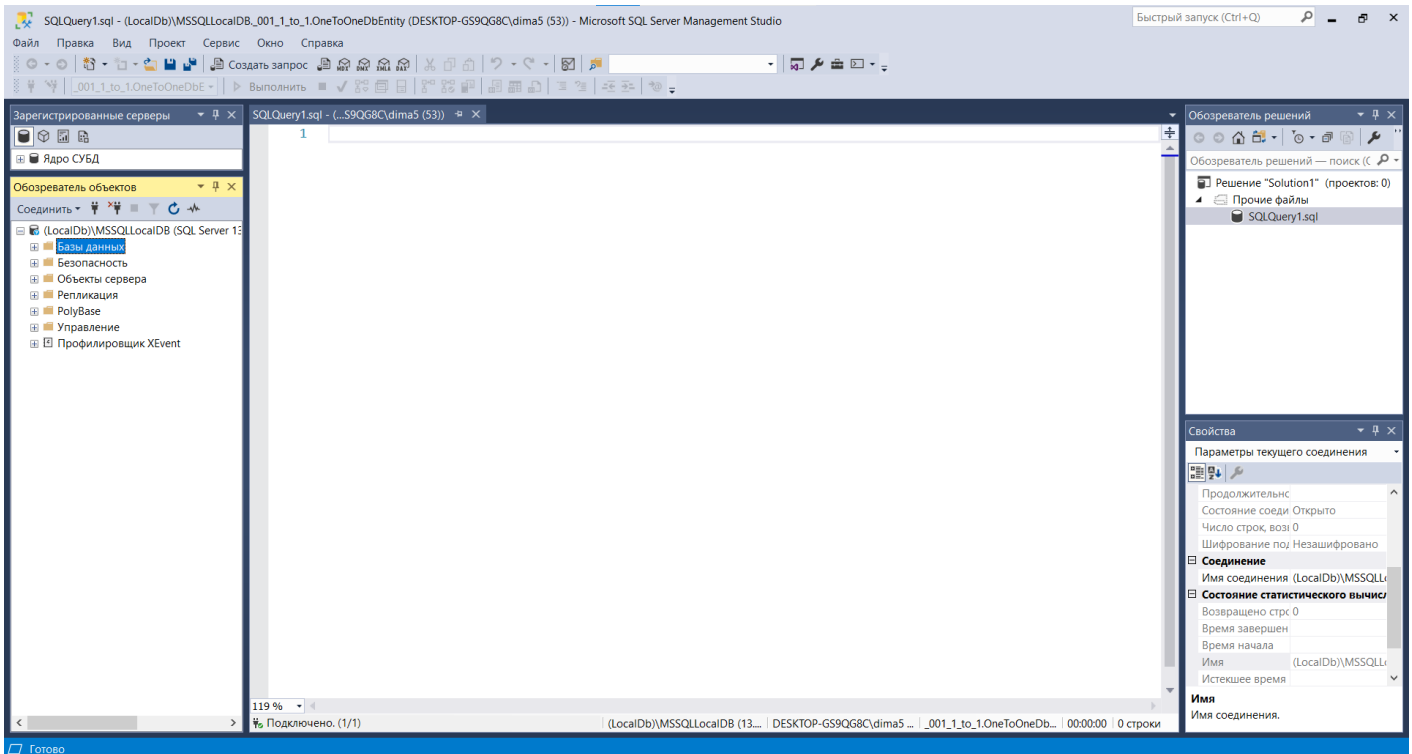


Рисунок 5. Интерфейс среды разработки Sql Server Management Studio

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1 Реєстрація чат-боту на платформі Telegram

Першим кроком розробки додатку є реєстрація у спеціального чат-бота «BotFather». Реєстрація починається з команди «/newbot», після чого пропонується ввести назву чат-бота з обов'язковим умовою: в кінці назви має бути зазначено «Bot» або «_bot». Якщо всі умови були виконані, то «BotFather» видає токен (спеціальний набір символів для доступу до HTTP API Telegram Bot) і URL-адресу для доступу з чат-боту. Приклад реєстрації чат-бота представлений на рисунку 6.

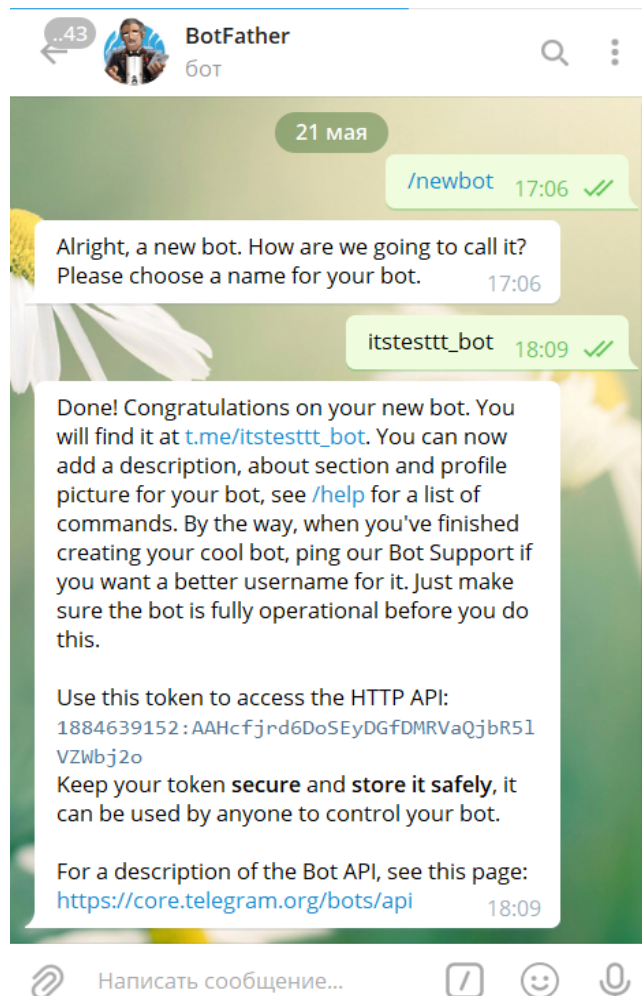


Рисунок 6. Приклад реєстрації чат боту

Для встановлення додаткових параметрів, таких як іконка чат-боту, вітальне повідомлення, опис чат-бота, а так само видалення наявних чат-ботів, існують такі команди (таблиця 1).

Таблиця 1 – Доступні команди для зміни чат-ботів

Команда	Опис
/setname	Змінює або встановлює існуюче ім'я.
/setdescription	Присвоює текст, який буде відображатися при першому відкритті чат-боту.
/setabouttext	Задає текст у полі «Про бота».
/setuserpic	Задає іконку чат-боту.
/setcommands	Задає список доступних команд чат-боту, які будуть відображатися у вигляді підказки вище чату.
/deletebot	Видаляє обраного чат-бота.

Крім команд для зміни основних параметрів чат-бота існує ряд інших команд, які дозволяють виводити незмінні параметри (наприклад токен), а також привласнювати значення, які представлені в таблиці 2.

Таблиця 2. – Доступні команди для додаткового налаштування чат-бота

Команда	Опис
/token	Повертає раніше отриманий токен обраного бота.
/revoke	Генерує новий токен для обраного бота, видаляє старий.

/setinline	Вмикає або вимикає можливість звертатися до боту з інших чатів.
/setinlinegeo	Вмикає або вимикає можливість передавати локацію до бота з інших чатів.
/setinlinefeedback	Дозволяє отримувати чат-боту інформацію про обрані команди користувачем.
/setjoingroup	Встановлює, чи може чат-бот бути доданий до групових чатів.
/setprivacy	Вмикає режим конфіденційності. В цьому режимі чат-бот отримує, обробляє і відсилає назад інформацію окремо для кожного користувача в груповому чаті

Після реєстрації чат-боту були проведені вище описані налаштування в зв'язку з специфікою додатку (наприклад неможливість додавання до групових чатів). Далі, після всіх налаштувань на стороні Telegram і отриманні токена можна приступити до розробки програмної частини чат-бота.

3.2 Програмна реалізація додатку

3.2.1 Реалізація основних компонентів чат-боту

Для зручної розробки наголос робився на об'єктно-орієнтованому програмуванні. Тобто для всіх об'єктів, задіяних у процесі використання чат-боту,

були створенні відповідні класи для структурування їх можливостей. Основні з них будуть перелічені нижче.

Клас `TelegramChat` представляє собою чат з користувачем. Він зберігає таку інформацію, як історію пошуку, плейлісти користувача, улюблені пісні, ідентифікаційний номер чату для зв'язку, інформацію про користувача, обрану мову та ін. У ньому реалізовані безпечний доступ до інформації, що зберігається, та механізми обміну інформацією з базою даних.

Клас `PlayList` являє собою представлення конкретного плейліста користувача в програмному коді. У ньому зберігається інформація про додані пісні та назва. Містить механізми завантаження та зчитування інформації з бази даних.

Клас `MusikLinks` являє собою список музичних композицій зі зручним механізмом їх перебору.

Клас `MusikLinkObject` являє собою одну конкретну музичну композицію, яка містить інформацію про виконавця, назву, посилання на завантаження та музичний ресурс, якому ця музична композиція належить.

Клас `LanguageText` являє собою клас з текстом для повідомлень бота. Він потрібен для зручної зміни мови користувача. При зміні мови цей клас буде оновлювати текст повідомлень завантажуючи його з відповідних файлів. Наприклад для англійсько мови файл з текстом має назву `eng.txt`.

Клас `CommandText` представляє собою текст повідомлення для конкретної команди. За допомогою цього класу можна змінювати текст конкретної команди або додавати текст для нової не змінюючи текст інших.

Клас `Vot` представляє собою сам чат-бот. У ньому міститься інформація про бота, токен, список діючих команд, список підключених ресурсів, поточну мову користувача, методи-обробники запитів, механізми багатопоточної обробки повідомлень, механізми ідентифікування запиту користувача як команди.

Клас `Buttons` – клас, який бере участь у створенні кнопок при відповіді користувачеві. Він створений для полегшеної розробки візуальної частини додатку.

За допомогою цих класів можна переходити до реалізації конкретних функцій чат-боту.

3.2.2 Реалізація вітання чат-боту

Перша команда, яку обов'язково повинен надіслати користувач чат-бот – це «`/start`». Вона потрібна для запуску бота для поточного користувача. Через цю команду користувач «дозволяє» боту надсилати йому повідомлення. Після цього бот повинен надіслати привітання користувачеві та інформацію про себе.

Для організації відповіді додатку на запити користувача було використано шаблон проектування `Command`, що значно пришвидшує розробку додаткових команд для чат-боту.

Спочатку був реалізований базовий абстрактний клас `Command` з функціоналом зміни текстових повідомлень в залежності від користувача (метод `ChangeText`) та ідентифікації запиту користувача як конкретної команди (метод `Contains`).

```

public abstract class Command
{
    Ссылка: 14
    public abstract string Name { get; }

    Ссылка: 99+
    static public string[] Text { get; private set; }
    ссылка: 1
    static public Type LastCommand { get; set; }

    ссылка: 1
    public Language ChangeText(Message message) =>
        ChangeText(message, this.GetType(), message.From);

    Ссылка: 36
    static public Language ChangeText(Message message, Type type, User user)...

    Ссылка: 21
    public abstract int Execute(Message message, TelegramBotClient client, TelegramChat chat, string msgText = null);

    Ссылка: 8
    public virtual bool Contains(string command) =>
        command.ToLower().Contains(this.Name.ToLower());
}

```

Рисунок 7. Реалізація базового класу Command

Далі був створений клас StartCommand, який успадковується від вище згаданого, та реалізує обробку запиту та надсилання відповіді користувачеві. Відповідь містить коротку інформацію про функціонал чат бота.

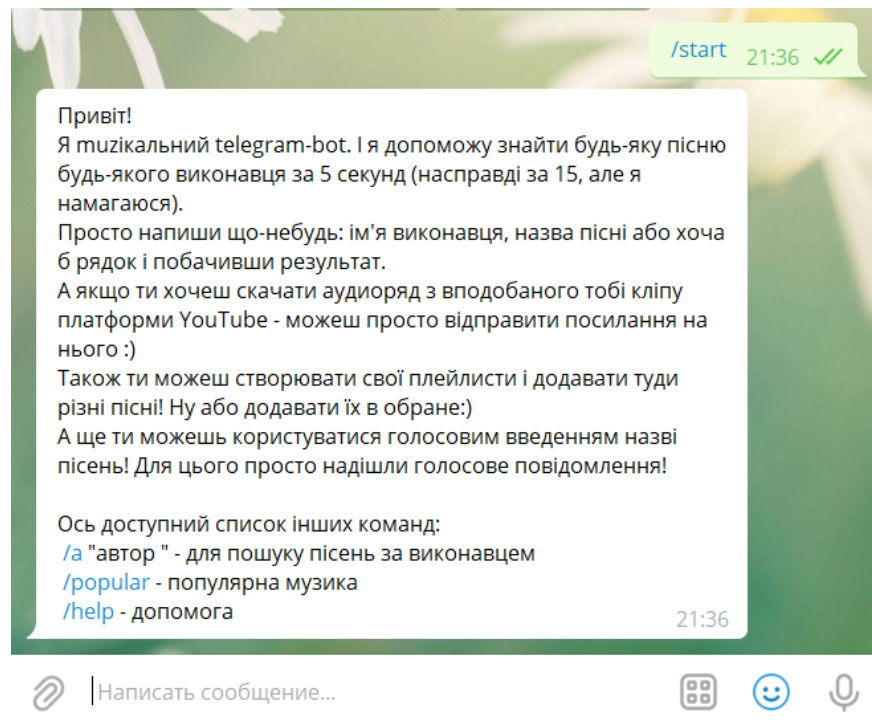


Рисунок 8. Приклад вітання чат-бота

Також при привітанні користувачеві буде надіслана панель швидких команд, які будуть знаходитись знизу. Ця панель значно полегшить введення команд, адже тепер користувач має можливість просто натиснути на кнопку замість введення. Зображення панелі наведено на рисунку 9.

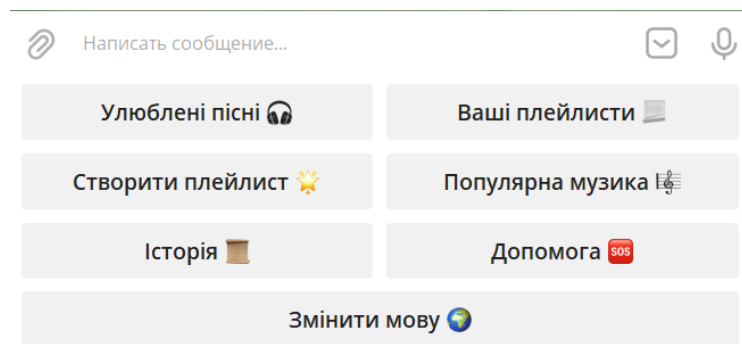


Рисунок 9. Нижня панель швидких команд

Також було додано команду «/help», яка в точності повторює роботу команди «/start» для інтуїтивно зрозумілого отримання допомоги.

3.2.3 Реалізація пошуку та завантаження музичних композицій

Основним завданням даного програмного продукту є можливість шукати за запитами користувачів музичні композиції та завантажувати їх на пристрій. Для розпізнавання чат-ботом запиту на пошук та завантаження пісень було обрано реалізувати команду «/search», також ця команда має псевдоніми, такі як «/s», «/getMusic». Сценарій завантаження користувачем аудіофайлів наведено у вигляді мережі Перті на рисунку 10.

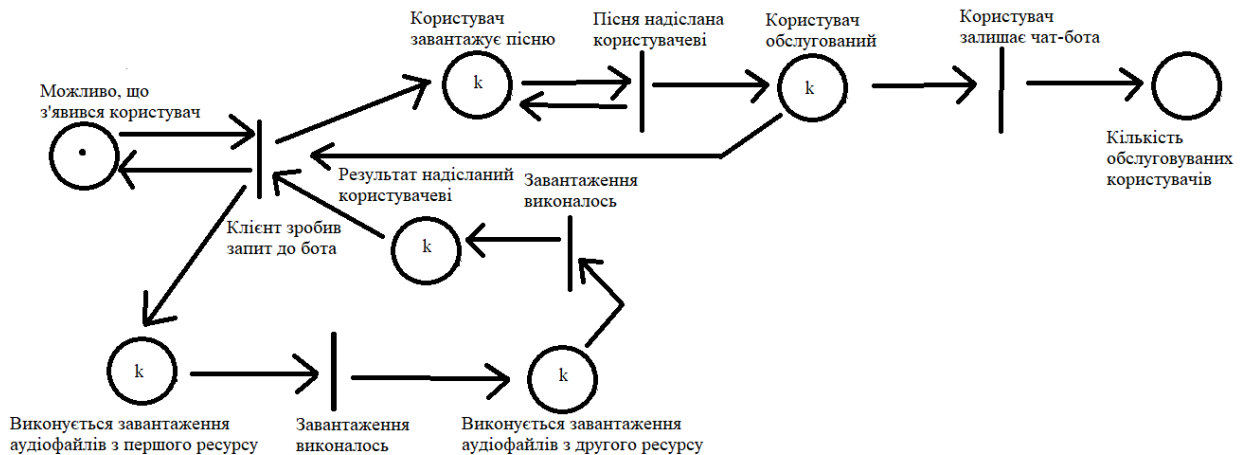


Рисунок 10. Сценарій виконання запиту користувача на завантаження пісень у вигляді мережі Петрі

Цей функціонал був реалізований у вигляді роботи команди «/search». Для пошуку та завантаження музичних композицій були обрані безкоштовні музичні портали, які можна легко підключати до бота через створення додаткових класів-обробників музичних ресурсів. Ці класи містять в собі заздалегідь прописаний HTML-парсер для конкретного ресурсу та обов'язково наслідуються від базового абстрактного класу всіх музичних ресурсів – BaseMusicResource. Цей клас задає структуру для класів-наслідників, що значно спрощує взаємодію з ними, та збирає і об'єднує отримані посилання на пісні від всіх класів-обробників музичних порталів.

Наразі бот має два підключених музичних портали для великої бази пісень та високої відмовостійкості, в разі коли один з ресурсів буде недоступний для пошуку та завантаження музичних композицій. Для обробки HTML-коду музичних ресурсів була завантажена через диспетчер пакетів NuGet бібліотека CQ, яка дозволяє швидко та зручно пересуватися та зчитувати частини HTML-коду. У майбутньому цей список планується розширяти.

```

public abstract class BaseMusicResource
{
    Ссылка: 3
    abstract public string ClassNameForUrls { get; }
    Ссылка: 4
    abstract public string ClassNameForNames { get; }
    Ссылка: 4
    abstract public string MainUrl { get; }
    Ссылка: 4
    abstract public Resource ResourceType { get; }

    ссылка: 1
    protected CQ GetCQ(string search)...

    Ссылка: 4
    public virtual MusicLinks GetUrlAudiosFromResource(string search)...

    Ссылка: 9
    static public MusicLinks GetUrls(string search, params BaseMusicResource[] baseResources)...

    ссылка: 1
    static private MusicLinks GetUrls(List<MusicLinks> arrayResources)...

    Ссылка: 3
    abstract protected MusicLinks GetMusLinks(CQ cq);

    Ссылка: 7
    abstract public Uri GetDownloadUrl(Uri url);
}

```

Рисунок 11. Реалізація базового абстрактного класу BaseMusicResource

Підключені ресурси представлені класами в папці Resources на рисунку 12.

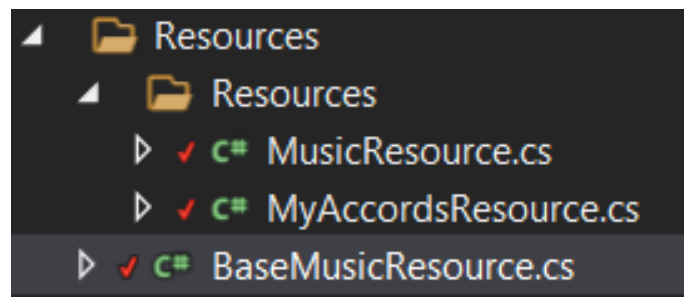


Рисунок 12. Класи-обробники музичних ресурсів, підключені до чат-боту

Далі був реалізований зручний інтерфейс для швидкого пошуку та завантаження пісень: був покращений AI для розпізнавання команд користувача, тобто було прибрано обов'язкове написання ключової команди «/search» для пошуку музичних композицій та реалізовано відображення списку музичних композицій через InlineKeyboardButton, на які треба натискати для завантаження обраної пісні. Від тепер можна просто вписати назву пісні, автора або і те і інше

та отримати список композицій. Приклад інтерфейсу команди «/search» можна побачити на рисунку 13.

Результатом команди «/search» є список пісень з усіх підключених музичних ресурсів розбитий на блоки по 10 пісень для зручного представлення. Щоб завантажити пісню, треба натиснути на кнопку з її назвою. Кнопки «Вперед» та «Назад» дозволяють пересуватись по списку, кнопка з червоним хрестиком видаляє запит користувача та список пісень.

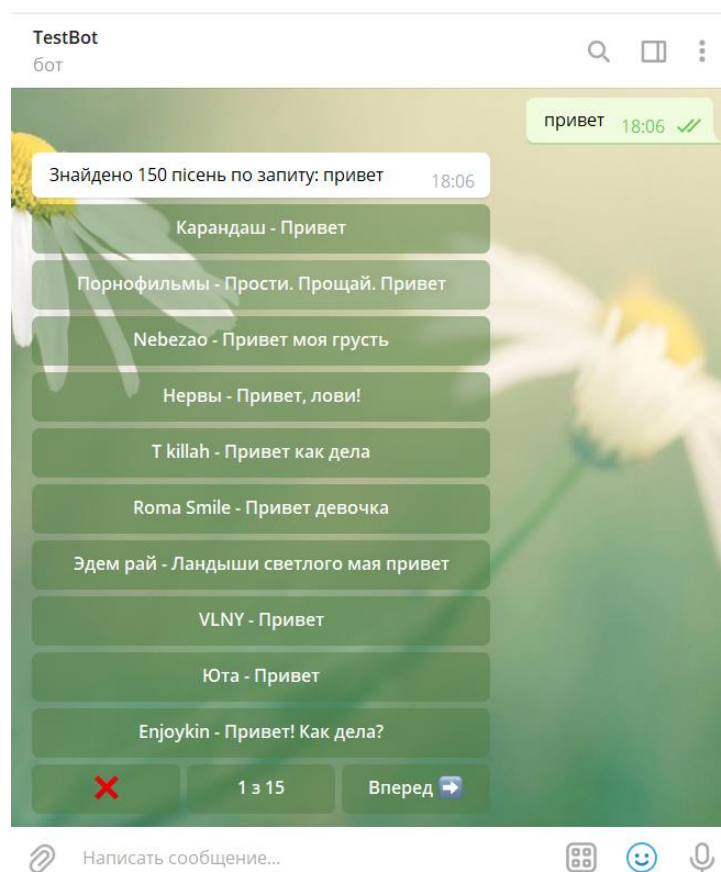


Рисунок 13. Приклад інтерфейсу команди «/search»

Також було створено додаткову команду «/artist», або «/a», яка дозволяє шукати музичні композиції через ім'я виконавця або назву групи. Формат цієї команди: /a «ІМ'Я_ВИКОНАВЦЯ».

Так як платформа Telegram підтримує можливість обміну голосовими повідомленнями між користувачами, було прийнято рішення додати підтримку голосової обробки у чат-бот. Для вирішення цієї задачі можна скористатися різними сервісами обробки голосу, наприклад Google Voice Api, YandexSpeechKit або Microsoft Azure Speech to text.

При використанні YandexSpeechKit у роботі чат-боту будуть виникати проблеми у зв'язку з блокуванням ресурсів цієї компанії на території України, тому порівнювалися Google Voice Api та Microsoft Azure Speech to text.

В цілому ці дві технології розпізнавання мови у використанні дуже схожі та значних відмінностей не мають. Але при роботі з VS значно легше працювати та під'єднувати технології компанії Microsoft, адже у VS є вбудовані механізми для цього. Тому було прийнято рішення використовувати Microsoft Azure Speech to text.

Для використання цієї технології потрібно зареєструвати обліковий запис на сайті Microsoft Azure та придбати платну підписку на обрану технологію. Після цього можна працювати з Api цієї технології за допомогою отриманого при придбанні унікального токена. Таким чином цей функціонал був впроваджений в чат-бот для полегшення вводу запитів користувачів. Щоб відправити голосове повідомлення до бота потрібно затиснути кнопку з зображенням мікрофону справа від чату та говорити запит, після чого відпустити кнопку з мікрофоном.

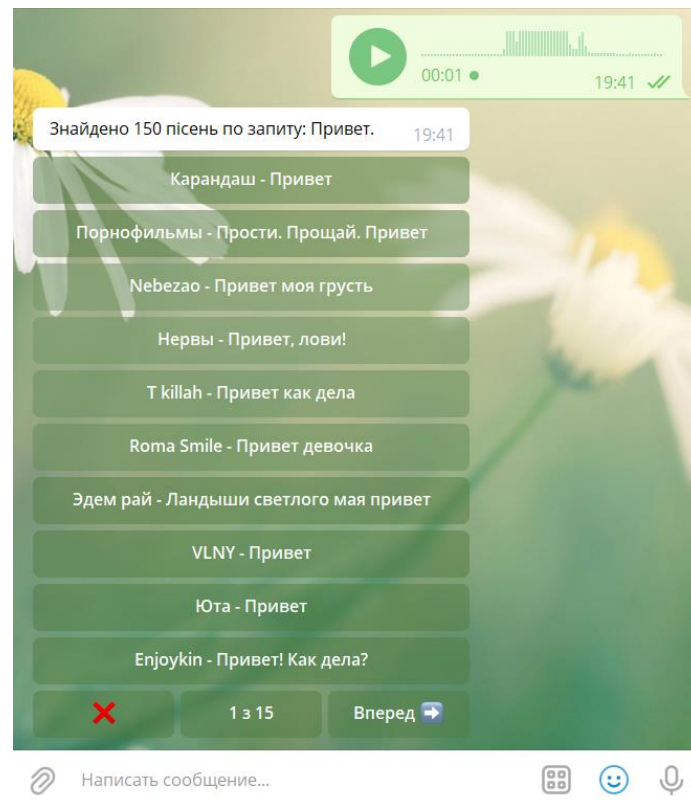


Рисунок 14. Приклад обробки запиту через голосове повідомлення

Також до функціоналу чат-боту була додана можливість отримання списку популярних на поточний час музичних композицій через команду «/popular». При обробці цієї команди чат-бот збирає всі популярні пісні за версіями підключених ресурсів, комбінує їх, та надсилає користувачеві. Для того, щоб отримати цей список, можна надіслати боту команду «/popular» або натиснути на кнопку «Популярна музика» на швидкій панелі команд.

При завантаженні будь якої пісні, до неї буде прикріплена можливість додавання цієї пісні до «Улюблених» або в будь який плейліст.

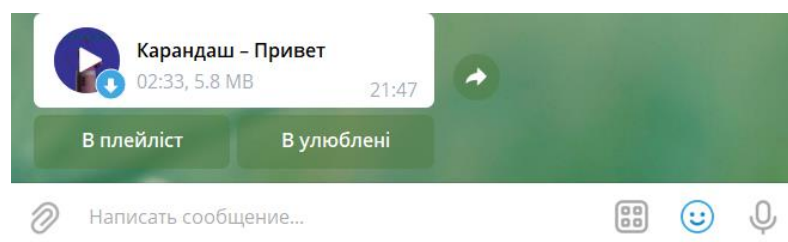


Рисунок 15. Приклад завантаженого аудіофайлу

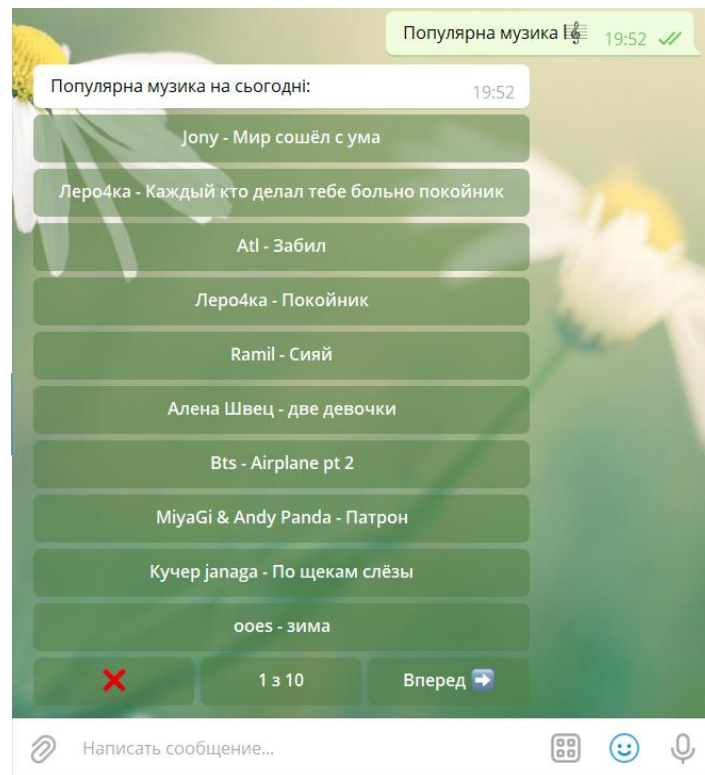


Рисунок 16. Приклад роботи команди «/popular»

3.2.4 Реалізація взаємодії з плейлістами

Для створення плейлістів була реалізована нова команда «/createPlaylist» з відповідним класом. Ця команда надсилає користувачеві повідомлення з проханням відправити назву нового плейлісту. Після відправки назви бот створює плейліст для користувача, записуючи всі зміни у БД. Усі плейлісти записуються в відповідну таблицю Playlist, яка посилається на таблицю користувачів для зв'язування їх з плейлістами. Надалі цей плейліст буде доступний для запису до нього пісень. У разі, якщо користувач випадково натиснув на кнопку, створення плейлісту можна скасувати, натиснувши на відповідну кнопку «Скасування». Ця команда може бути викликана з панелі швидких повідомлень, тому прописувати її не обов'язково. Приклад виконання цієї команди наведено на рисунку 17.

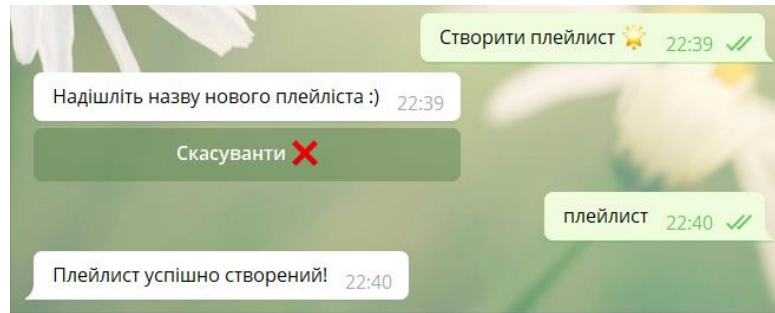


Рисунок 17. Приклад створення плейлісту

Також при необхідності новий плейліст можна створити прямо під завантаженою піснею, або пісню можна додати вже в існуючий. При додаванні до плейлісту буде виведено повідомлення про успішність або не успішність операції. Додавання пісні в плейліст може не відбутися, якщо користувач перед додаванням видалив плейліст, куди хотів додати пісню. Спроба додавання пісні до плейлісту проілюстрована на рисунку 18.

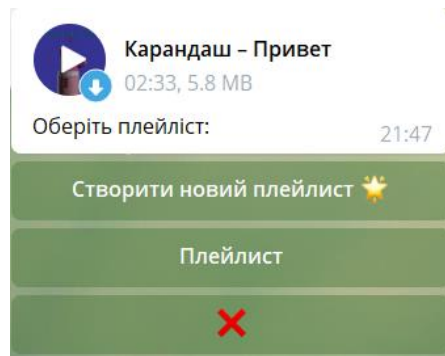


Рисунок 18. Спроба додати пісню в плейліст

Для доступу до плейлістів була реалізована команда «/getPlaylists», яка надсилає користувачеві список всіх створених плейлістів. Через цю команду також можна створити новий або видалити старий. Видалення плейлісту відбувається разом з піснями у ньому. Кнопка з червоним хрестиком видаляє запит користувача та відповідь бота з чату. Ця команда може бути викликана з панелі швидких повідомлень, тому прописувати її не обов'язково.

Якщо натиснути на кнопку з назвою плейлісту, повідомлення оновиться і буде містити інформацію про музичні композиції, що знаходяться в конкретному плейлісті. У цьому повідомленні ви можете вибірково завантажити пісні з цього плейлісту, натиснувши на назву пісні, завантажити весь плейліст, натиснувши на кнопку «завантажити все», переглянути пісні цього плейлісту та видалити непотрібні.

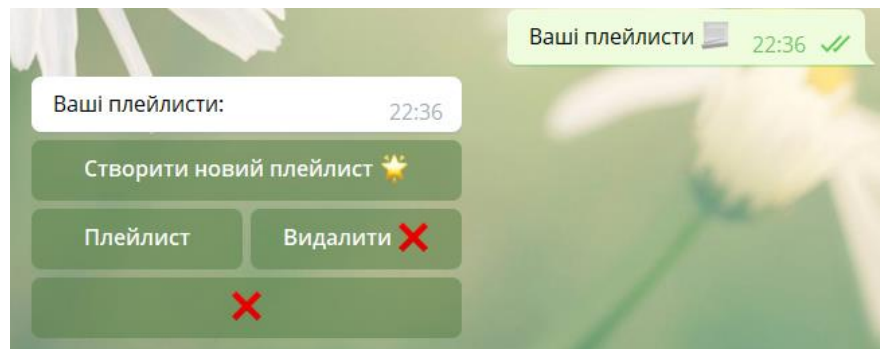


Рисунок 19. Результат роботи команди «/getPlaylists»

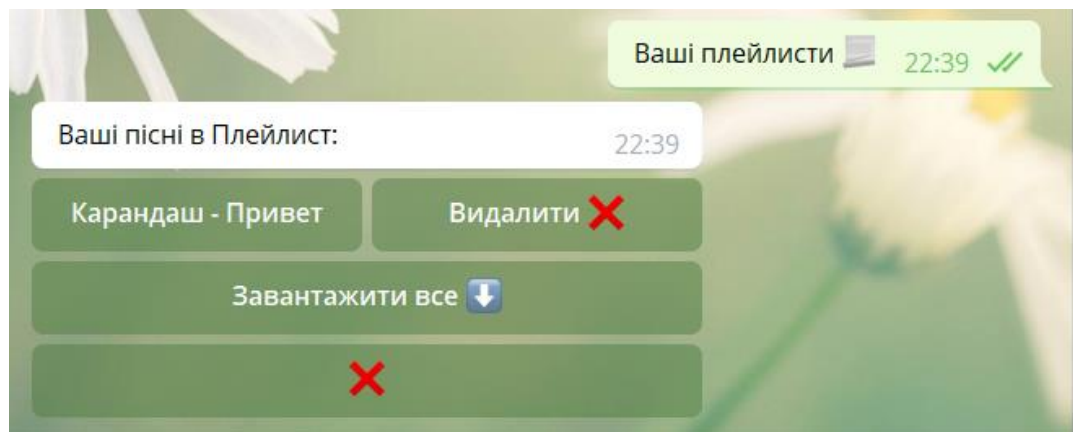


Рисунок 20. Результат роботи команди «/getPlaylists» при переході до конкретного плейлісту

Також в цьому чат-боті було реалізовано додавання пісень до «Улюблених».

Улюблені пісні – це особливий плейліст, який є у кожного користувача за замовчуванням та який не можна видалити. Додавання до «Улюблених» відбувається при натисканні кнопки «До улюблених» під аудіофайлом.

Щоб отримати список улюблених пісень була реалізована команда «/getLikeMusic», яка також присутня на панелі швидких команд та не потребує обов'язкового введення. У цій команді можна переглянути, завантажити та видалити ваші улюблені пісні. Для завантаження пісні треба натиснути на кнопку з її назвою. Інтерфейс команди зображений на рисунку 21.

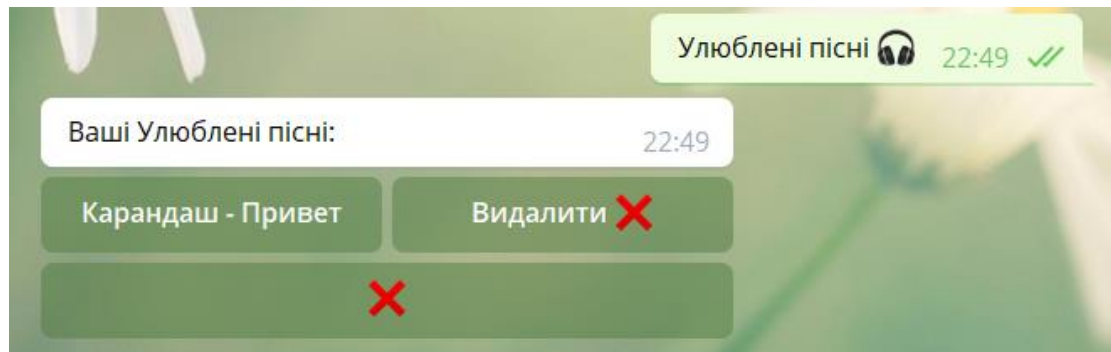


Рисунок 21. Інтерфейс команди «/getLikeMusic»

3.2.5 Реалізація завантаження аудіоряду з кліпів платформи YouTube

Для приваблення більшої кількості користувачів було прийнято рішення додати функціонал завантаження аудіофайлів з платформи YouTube. Щоб вирішити поставлену задачу була підключена бібліотека VideoLibrary, яка спеціалізується на завантаженні відео з платформи YouTube та бібліотека Xabe.FFmpeg, яка спеціалізується на конвертування медіафайлів в різні формати.

Для додання такого функціоналу до чат-бота було реалізовано команду «/youtubeDownload». Для того, щоб звернутися до цієї команди також не обов'язково прописувати її, достатньо надіслати боту посилання на кліп і він відправить вам через деякий час аудіофайл з цього кліпу. Результат виконання команди наведено на рисунку 22. При виконанні команди спочатку завантажуються кліп з платформи YouTube через бібліотеку VideoLibrary на сервер, де працює додаток. Далі йде конвертація відеофайлу (.mp4) у аудіофайл (.mp3)

через бібліотеку `Xabre.FFmpeg`. Потім отриманий аудіофайл надсилається на пристрій користувачеві та йде видалення створених файлів.

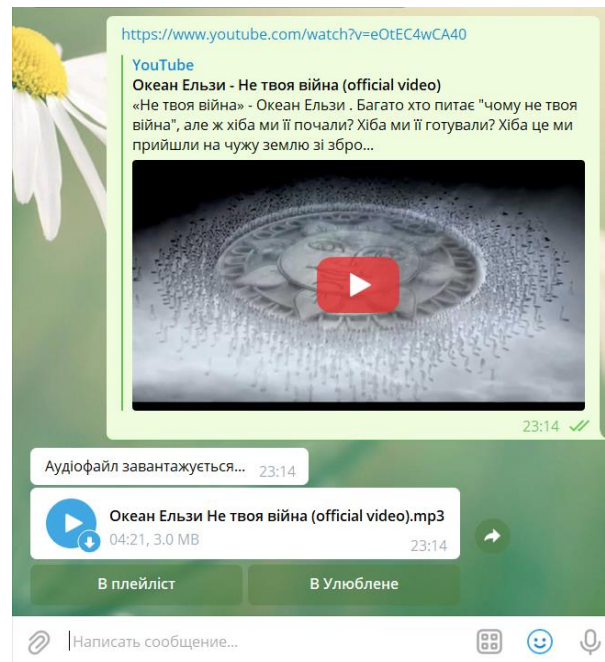


Рисунок 22. Результат виконання команди «/youtubeDownload»

3.2.6 Реалізація підтримки In-line mode

Платформа Telegram дозволяє взаємодіяти з ботами не тільки в чаті з ним, а ще й в інших чатах, які не мають відношення до чат-ботів. Це відбувається через In-line mode. Щоб наш чат-бот мав підтримку цього режиму, його треба ввімкнути надіславши FatherBot команду `/setinline`. Це дасть змогу взаємодіяти з ботом в інших чатах через його ім'я. Тобто, що б звернутися до бота, треба в чаті написати `@НАЗВА_БОТА АРГУМЕНТИ`. Приклад використання In-line mode наведено на рисунку 23.

Для організації підтримки такого режиму в програмному коді був розроблений спеціальний метод-обробник події, який спрацьовує при події `OnInlineQuery`. Вона спрацьовує при звертанні з будь якого чату до чат-боту через ім'я, як це проілюстровано на рисунку 23. Також цей метод-обробник приймає

текстовий аргумент після пробілу від назви бота та може, відштовхуючись від нього, повертати розрахований результат користувачеві, який звернувся до бота через ім'я, яке було отримано методом-обробником разом з текстовим аргументом. Для надсилання в чат аудіофайлу, треба натиснути на обрану пісню зі списку.

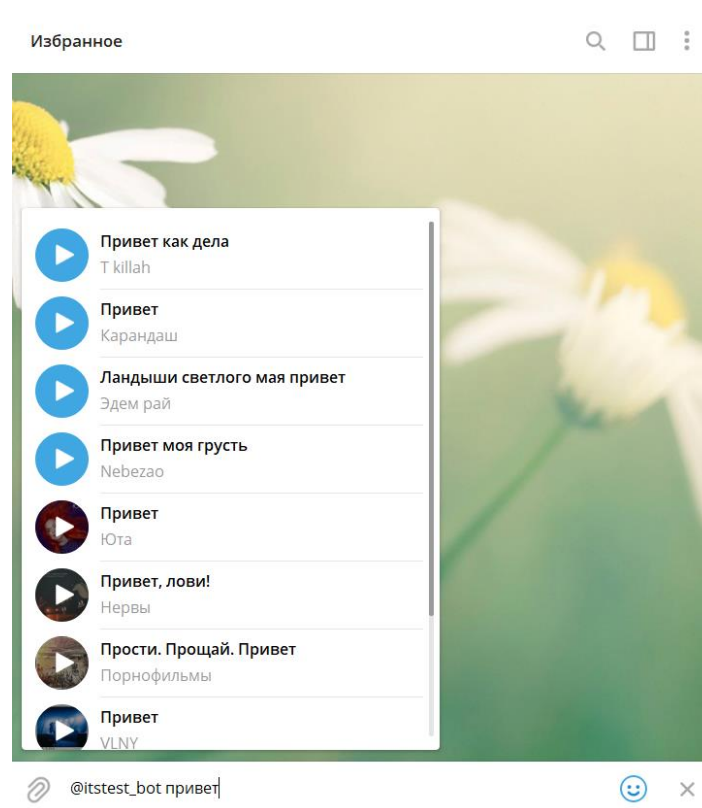


Рисунок 23. Приклад використання ботом In-line mode

В даному випадку чат-бот Muz-z-z на основі отриманого аргументу робить пошук по підключеним ресурсам, вибирає перші 10 аудіофайлів, асинхронно витягує посилання на ці пісні та надсилає користувачеві у вигляді списку пісень.

Це може бути дуже корисно для швидкого розповсюдження між близькими людьми та друзями, адже не потрібно буде заходити в чат з ботом, а можна буде на пряму надіслати знайомому пісню разом з цим прорекламувавши чат-бота.

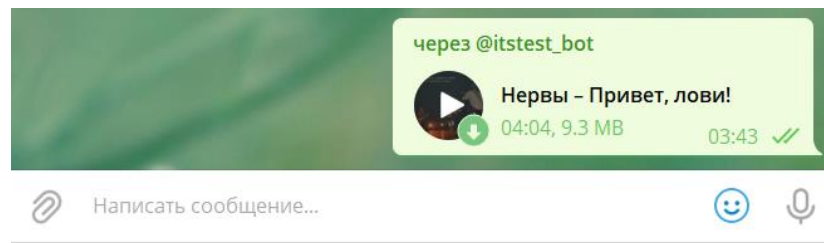


Рисунок 24. Приклад надісланої пісні в інший чат за допомогою In-line mode

3.2.7 Реалізація додаткового функціоналу

3.2.7.1 Реалізація збереження історії запитів користувачів

Для цієї задачі була реалізована команда «/getHistory». Кожен раз при будь-якому запиті користувача всі його повідомлення, надіслані чат-боту, будуть зберігатись в базі даних. Для отримання історії запитів також не обов'язково використовувати команду, а можна скористатися панеллю швидких команд. Інтерфейс команди наведений на рисунку 25.

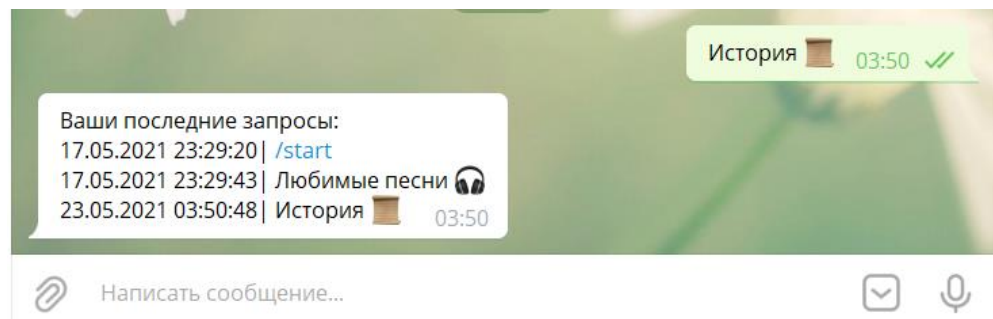


Рисунок 25. Результат роботи команди «/getHistory»

3.2.7.2 Реалізація зміни мови користувача

Для цієї задачі була реалізована команда «/changeLanguage». Ця команда викликається під час першого запиту користувача для початкового встановлення мови. Чат-бот буде насилати відповіді обраною мовою користувачем. Механізм зміни мови реалізований через розділ тексту між відповідями різних команд, тому

дуже зручно додавати нові команди до чат-боту та одразу ж додавати до відповідних файлів переклади відповідей нової команди.

Наразі чат-бот підтримує три мови: українську, англійську та російську. Весь текст, звідки додаток бере переклад – файли з відповідними назвами: ukr.txt, eng.txt, rus.txt. Цей функціонал допоможе збільшити аудиторію користувачів задовольнивши їхні вподобання на рахунок мови спілкування. Ця команда також не обов’язкова для введення та доступна на панелі швидких команд. Приклад роботи команди проілюстровано на рисунку 26.

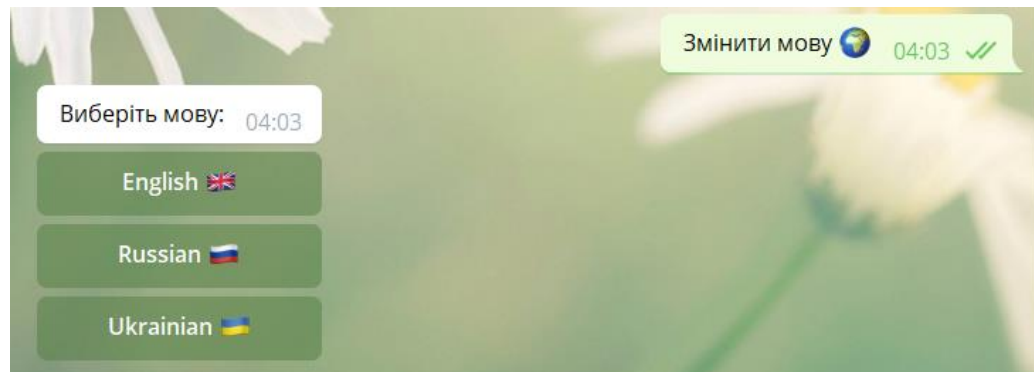


Рисунок 26. Інтерфейс команди «/changeLanguage»

Список всіх команд чат-боту та їх короткий опис наведено в додатку В.

Висновки

В даний час популярність месенджерів як засобів спілкування незмінно зростає. Компанії, сім'ї, друзі щодня користуються можливостями обміну повідомленнями та медіаконтентом на відстані. Так само варто відзначити зростання популярності такого виду програмних продуктів як чат-боти, які працюють на платформах месенджерів.

Цілодобова служба підтримки користувачів, конвертація документів і медіафайлів, замовлення таксі, пошук необхідних даних та багато іншого в даний час може бути реалізовано в рамках лише одного месенджера. Користувачам не доведеться завантажувати безліч додатків для вирішення вузьконаправлених завдань, тому що досить мати лише один месенджер і необхідний набір чат-ботів, які не займають місце в пам'яті смартфона.

В межах випускної кваліфікаційної роботи були виконані поставлені завдання. Були вивчені месенджери та чат-боти. Було проведено порівняння і аналіз переваг і недоліків існуючих месенджерів, внаслідок чого був обраний Telegram як найзручніший і доступний в плані документації Telegram Bot API. Також було порівняно існуючі пошукові сервіси музичних композицій.

Було розглянуто всі можливості чат-ботів на платформі Telegram та їх реалізація у боті Muz-z-z, була організована та налагоджена багатопоточність для можливості обслуговування різної кількості користувачів в даний момент часу та зроблені значні успіхи в організації розпізнавання голосових повідомлень і інтерпретації їх у текстовий вигляд.

В межах виконання роботи було обрано технології та середовище для розробки чат-бота серед яких C#, SSMS, VS, Telegram Bot Api та ін.

Таким чином, результатом випускної кваліфікаційної роботи є повністю реалізований чат-бот для зручного пошуку музичних композицій, який буде у нагоді користувачам при прослуховуванні пісень та розповсюджені їх своїм родичам або друзям завдяки підтримки In-line моду. Чат-бот наділений великою кількістю різноманітних команд для задоволення всіх потреб користувачів.

Цей бот дуже зручний для швидкого пошуку музичних пісень, також його реалізація на платформі Telegram дозволяє прослуховувати пісні в режимі офлайн, наприклад, під час далеких подорожей в інші міста. У майбутньому планується додати більшу кількість музичних ресурсів та механізми впровадження рекламних інтеграцій в роботу чат-бота для монетизації.

Перелік використаних джерел

- 1) Messina, C. 2016 will be the year of conversational commerce [Електронний ресурс] / С. Messina // Medium. – 2016. – Режим доступу до сайту: <https://medium.com/chris-messina/2016-will-be-the-year-of-conversationalcommerce-1586e85e3991>
- 2) Рейтинг месенджерів [Електронний ресурс] : Сервіс порівняння та вибору додатків та сервісів у сфері ІТ. – 2018. – Режим доступу до сайту: <https://coba.tools/compilation/reiting-messendzherov-2018>
- 3) Microsoft Docs [Електронний ресурс] : Інтегрована середа розробки Visual Studio. – Режим доступу до сайту: <https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- 4) Telegram Bot API [Електронний ресурс] : Telegram Documents. – Режим доступу до сайту: <https://core.telegram.org/bots/api>
- 5) Metanit [Електронний ресурс] : Введення в MS Sql Server та T-Sql. – Режим доступу до сайту: <https://metanit.com/sql/sqlserver/1.1.php>
- 6) Козлов А. А., Телеграм бот як простий та зручний спосіб отримання інформації [Електронний ресурс] / А. А. Козлов, А. В. Батищев // Територія науки. – 2017. – №5. – с. 55-64. – Режим доступу до сайту: <https://cyberleninka.ru/article/v/telegram-bot-kak-prostoy-i-udobnyy-sposobpolucheniya-informatsii>
- 7) Microsoft Azure [Електронний ресурс] : Speech to text. – Режим доступу до сайту: <https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>
- 8) Telegram Bot API [Електронний ресурс] : Довідник по Bot Api. – Режим доступу до сайту: <https://tlgrm.ru/docs/bots/api>

- 9) Матвеева Н. Ю. Технології створення та застосування чат-ботів [Електронний ресурс] / Н. Ю. Матвеева, А. В Золотарюк . – Режим доступу до сайту: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov>
- 10) Microsoft Docs [Електронний ресурс] : C# Documentation. – Режим доступу до сайту: <https://docs.microsoft.com/en-us/dotnet/csharp/>
- 11) Microsoft Docs [Електронний ресурс] : C# programing guid. – Режим доступу до сайту: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
- 12) Microsoft Docs [Електронний ресурс] : .Net documentation. – Режим доступу до сайту: <https://docs.microsoft.com/en-us/dotnet/>
- 13) GitHub [Електронний ресурс] : Telegram.Bot documentation. – Режим доступу до сайту: <https://github.com/TelegramBots/Telegram.Bot>
- 14) Хабр [Електронний ресурс] : Боты на .Net Core для Telegram, Slack и Facebook. – Режим доступу до сайту: <https://habr.com/ru/company/alfa/blog/343846/>
- 15) FFmpeg [Електронний ресурс] : FFmpeg documentation. – Режим доступу до сайту: <https://www.ffmpeg.org/documentation.html>
- 16) GitHub [Електронний ресурс] : VidelLibrary documentation. – Режим доступу до сайту: <https://github.com/omansak/libvideo>
- 17) Шилдт Г. Полное руководство C# 4 / Г. Шилдт – К. : «Вильямс», 2015. – 1056 с.
- 18) Троелсен Е. Язык программирования C# 5 и платформа .net 4.5. / Е. Троелсен – К. : «Вильямс», 2016. – 1440 с.
- 19) Рихтер Д. CLR via C#. / Д. Рихтер – К. : «Питер», 2019. – 896 с.

20) Фрімен Е. Паттерны проектирования / Е. Фрімен – К. : «Питер», 2016. – 656 с.

21) Клири Ст. Конкурентность в С#. Асинхронное, параллельное и многопоточное программирование. 2-е межд. изд. / Клири Ст. – К. : «Питер», 2020. – 304 с.

Додаток А

Порівняння існуючих месенджерів

Назва	Переваги	Недоліки
WhatsApp	Доступ по протоколу HTTPS	Значне зниження якості переданих медіафайлів
	Групові чати до 256 учасників	
	Доступ до чатів в режимі офлайн	Велика затримка у відеозвінків
	Наявність голосових та відеозвінків	
Viber	Push-повідомлення в офлайн режимі	Наявність вбудованих покупок
	Вбудовані сервіси для оплати в мережі інтернет	Наявність великої кількості реклами
	Наявність public-акаунтів	
Skype	Демонстрація екрану під час дзвінку або відеотрансляції	Сильна залежність від швидкості з'єднання мережі Інтернет
	Зручні групові дзвінки та відеоконференції	Нестабільність з'єднання Відсутність шифрування
iMessage	Великий набір стандартних анімацій чату	Велика затримка доставки повідомлень
	Висока ступінь інтеграцій з іншими додатками Apple	Працює тільки на ОС iOS
		Неможливість створення чат-ботів
Hangouts	Особливий протокол передачі голосових повідомлень Google Voice	Не зручний інтерфейс
	Історія повідомлень зберігається на віддаленому сервері	Сильне зниження якості при передачі медіафайлів
	Інтегровані сервіси Google+ та Gmail	
Facebook Messenger	Підтримка чат-ботів	Повідомлення неможливо редагувати або видалити
	Шифрування повідомлень	
Facebook Messenger	Можливість підтвердження прочитання з Push-повідомлень	Немає підтримки десктопних ОС
	Функція «Секретний чат»	
	Автоматичне видалення повідомлень через вказаний час	Слабка конфіденційність даних

Telegram	Найзручніша платформа для чат-ботів	Десктопні версії не підтримують можливість створення секретних чатів
	Відкритий початковий код	
	Вбудований редактор фото	
	Спеціально створений протокол шифрування MTProto	Десктопні версії мають низьку стабільність при роботі.
	Секретні чати та самовидалення повідомлень	
	Наявність голосових та відеозвінків	

Додаток Б

Порівняння існуючих пошукових сервісів

Назва/Ім'я користувача	Переваги	Недоліки
FinderMusic/@fmsbot	Підтримка великої кількості мов світу	Немає підтримки завантаження аудіофайлів з платформи YouTube
	Дуже велика швидкість завантаження аудіофайлів (< 1 с)	Немає підтримки In-line моду
		Немає можливості створення плейлістів
		Неможливість зміни мови інтерфейсу
LosslessBot/@losslessBot	Дуже велика швидкість завантаження аудіофайлів (< 1 с)	Немає можливості завантажувати аудіофайли з платформи YouTube
	Підтримка In-line моду	
	Підтримка створення плейлістів	Наявність рекламної розсилки
	Доступний пошук по альбомах	
VKMusic/@ivksound_bot	Підтримка створення плейлістів	Немає можливості завантажувати аудіофайли з платформи YouTube
	Можливість синхронізації з обліковим записом мережі ВКонтакте	Наявність рекламної розсилки
		Є груповий чат для обговорення тем, пов'язаних з цим чат-ботом
Muz-z-z/ @ItsMuzzBot	Підтримка створення плейлістів	Низька швидкість завантаження аудіофайлів (>>1 с)
	Підтримка голосового введення запитів	
	Підтримка In-line моду	

Muz-z-z/ @ItsMuzzBot	Можливість завантаження кліпів з платформи YouTube	
	Позначення пісень як «Улюблені»	

Додаток В

Список команд чат-боту Muz-z-z

Назва	Опис
/s, /search	Пошук музичних композицій по назві пісні або імені виконавця. Формат: /s «НАЗВА_ПІСНІ», «НАЗВА_ПІСНІ»
/a, /artist	Пошук музичних композицій по імені виконавця. Формат: /a «ІМ'Я ВИКОНАВЦЯ»
/youtubeDownload	Завантаження аудіоряду з кліпів з платформи Youtube. Формат: /youtubeDownload «ПОСИЛАННЯ_НА_КЛІП», «ПОСИЛАННЯ_НА_КЛІП»
/createPlaylist	Створює плейліст з заданим ім'ям. Є у нижній панелі швидких команд. Формат: /createPlaylist
/getPlaylists	Надає користувачеві список плейлістів для подальшої взаємодії з ним. Є у нижній панелі швидких команд. Формат: /getPlaylists
/getLikeMusic	Надає користувачеві список пісень, помічені як «Улюблені», для подальшої взаємодії з ним. Є у нижній панелі швидких команд. Формат: /getLikeMusic
/popular	Надає користувачеві список популярних пісень на сьогодні. Є у нижній панелі швидких команд. Формат: /popular
/start, /help	Виводить вітання користувачеві та загальну інформацію по функціоналу чат-боту. Є у нижній панелі швидких команд. Формат: /start, /help
/changeLanguage	Змінює мову інтерфейсу чат-боту поточному користувачу. Є у нижній панелі швидких команд. Формат: /changeLanguage
/getHistory	Надсилає користувачеві його історію запитів. Є у нижній панелі швидких команд. Формат: /getHistory