

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

**ЩЕРБАК ВЛАДИСЛАВ ВОЛОДИМИРОВИЧ**

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

канд. техн. наук, доцент

\_\_\_\_\_ Т. В. Нескородева

« \_\_\_\_\_ » \_\_\_\_\_ 2021р.

**РОЗРОБКА ТЕЛЕГРАМ-БОТУ ДЛЯ ФОРМУВАННЯ РОЗКЛАДУ  
ЗАНЯТЬ**

Спеціальність 122 «Комп'ютерні науки»

**Кваліфікаційна (бакалаврська) робота**

Керівник:

Римар П. В., старший викладач

кафедри інформаційних технологій

Оцінка: \_\_\_\_/\_\_\_\_/\_\_\_\_

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: \_\_\_\_\_  
(підпис)

Вінниця – 2021

## АННОТАЦІЯ

Щербак В. В. **Розробка телеграм-боту для формування розкладу занять.** Спеціальність 122 «Комп'ютерні науки», освітня програма «Сучасні інформаційні технології та програмування». Донецький національний університет імені Василя Стуса, Вінниця, 2021.

Кваліфікаційна (бакалаврська) робота присвячена розробці телеграм-боту для формування розкладу занять. За допомогою мови програмування C# та середовища .Net був розроблений телеграм-бот, за допомогою якого можна викладачам та студентам дізнатися про свій розклад.

Ключові слова: інформаційні технології, телеграм бот, бот.

Сторінок 39, рис. 22, джерел 27.

## ABSTRACT

Shcherbak Vladislav. **Development of a telegram-bot for the formation of the schedule of classes.** Specialty 122 «Computer Science», Educational program «Modern information technologies and programming», Vasyl' Stus Donetsk National University, Vinnytsia, 2021.

Qualification (bachelor's) work is devoted to the development of a telegram-bot for the formation of classes. Using the C # programming language and the .Net environment, a telegram bot was developed that allows teachers and students to learn about their schedule.

Keywords: telegram bot, ASP.NET, C#, web-application.

Pages 39, Figure 22, sources 27.

## ЗМІСТ

ВСТУП .....	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯДІ ІСНУЮЧИХ АНАЛОГІВ.....	6
1.1 Постановка задачі.....	6
1.2 Огляд існуючих Telegram–ботів за даною тематикою .....	6
Висновок до розділу 1 .....	8
РОЗДІЛ 2. ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ .....	9
2.1 ASP.NET Core.....	9
2.2 Мова програмування C#.....	11
2.3 Microsoft SQL Server.....	13
2.4 Visual Studio 2019 .....	15
2.5 Docker .....	15
2.6 HTTPS.....	17
Висновок до розділу 2.....	17
РОЗДІЛ 3. РОЗРОБКА ТЕЛЕГРАМ-БОТА.....	18
3.1 Огляд BotFather .....	18
3.2 Ознайомлення з Telegram API: .....	25
3.3 Вибір мови програмування та середовища розробки.....	27
3.4 База даних.....	29
3.5 Команди для бота .....	30
3.6 Тестування програми та результати її виконання .....	31
Висновок до розділу 3.....	34
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	36

## ВСТУП

У сьогоднішній день у світі спостерігається стрімкий розвиток комп'ютерних інформаційних технологій. За останні роки в цій області відбувся певний ряд змін, які особливо проявилися в процесі формування та функціонування Інтернету. В даний час різноманітні гаджети та додатки націлені на те, щоб максимально спростити життя звичайних користувачів у вирішенні повсякденних завдань, а також забезпечити досить швидкий доступ до будь-якої інформації.

Згідно зі статистикою, найчисленніша група людей активно користуються як інтернетом, так і можливостями електронних пристроїв і різноманітних додатків – це школярі та студенти. Будь-який сучасний навчальний заклад прагне до розвитку та підвищення якості процесу навчання, його доступності та інтерактивності. На даний момент більшість закладів вищої освіти мають своє інтернет-представництво на просторах всесвітньої мережі, де розміщена найактуальніша інформація про навчальний заклад, його історії, співробітників і викладачів, висвітлюється студентська життя і різні заходи. Як правило, основна мета таких веб-сервісів – привернути увагу майбутніх студентів, їхніх батьків, потенційних роботодавців, надавши, в першу чергу, відомості про актуальні проекти, дослідження та інших наукових і позанавчальних заходах, що проводяться в тому чи іншому освітньому закладі. Однак самі студенти не завжди можуть отримувати швидкий і зручний доступ до інформації, пов'язаної з їх навчальною діяльністю, наприклад, актуальні зміни в розкладі занять. Є кілька варіантів вирішення такого роду завдань. Одним із них є створення Telegram-бота.

Telegram-бот – це програма співрозмовник, головна мета якої полягає у оперативному наданні необхідної інформації користувачеві та відповідям на його запити. Можливості Telegram-бота припускають не тільки автоматизацію певних процесів, а також налагодження персоналізованого спілкування з користувачем через месенджери. Бот може вирішувати певний ряд завдань:



виконання стандартних завдань, що повторюються, пошук та акумулювання даних (збір матеріалів для подальшої аналітики), технічна підтримка.

Кожен з нас є активним користувачем соціальних мереж, особливо популярною мережею серед студентів є Telegram. Перевагами використання даного месенджера є: можливість обмінюватися текстовими, голосовими та відео-повідомленнями, надсилати стікери та фотографії, файли багатьох форматів. Також Telegram надає можливість здійснювати аудіо- та відео-дзвінки, організовувати різноманітні конференції, групи та канали, що розраховані на багатьох користувачів. Додатковою та корисною функцією є Telegram-бот, це програма, головним призначенням якої є – в текстовому або голосовому діалоговому режимі допомогти користувачеві з вирішенням його проблеми, пошуком необхідної інформації і т.д. Спілкування з ботом здійснюється через діалог, так само як зі звичайним користувачем.

**Мета бакалаврської роботи** полягає у вивченні методики проектування telegram-боту з використанням мови програмування C#.

**Задачами дослідження є:**

- огляді існуючих ботів за даною тематикою, виявленні їх переваг та недоліків;
- вивченні необхідних інструментів та технологій для розробки telegram-боту;
- розробці telegram-боту для формування розкладу занять.

**Об'єктом** для дослідження є мова програмування «C#», а саме чи можна на її базі розробити власний телеграм-бот.

**Предметом** дослідження є механізм розробки телеграм-боту на базі «ASP.NET Core».

В ході роботи належить вивчити роботу платформи «ASP.NET Core», розібратися в її особливостях, можливостях і доповненнях.

Робота складається зі вступу, трьох розділів зі своїми підрозділами, висновки та списку літературних джерел кількістю 27 найменувань та 21 рисунка. Загальний об'єм роботи 39 сторінок.

## **РОЗДІЛ 1**

### **ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ІСНУЮЧИХ АНАЛОГІВ**

В даному розділі описано постановку задачі з розробки телеграм-бота, а також вибір та опис всіх необхідних інструментів та технологій для розробки телеграм-бота.

#### **1.1 Постановка задачі**

Необхідно розробити телеграм-бот для отримання розкладу занять викладача або групи та отримання графіку сесії. Бот повинен містити наступні ключові елементи:

- Меню, за допомогою якого можна обирати необхідний розклад.
- Кнопки для отримання розкладу окремої групи або викладача.
- Розміри шрифтів повинні бути зручними для читання.
- Всі данні повинні зберігатися в базі даних.
- Спілкування з Telegram API повинно відбуватися по протоколу HTTPS.

#### **1.2 Огляд існуючих Telegram-ботів за даною тематикою**

Для огляду було знайдено декілька варіантів. Проте, оглядаючи функціональні можливості аналогічних телеграм-ботів, було виявлено, що вони працюють неналежним чином.

Перший варіант – «Львівський транспорт» – це Telegram-бот, завдання якого полягає у наданні допомоги користувачам в онлайн-режимі. Завдяки цьому додатку на мобільному телефоні жителі міста можуть відслідковувати розклад прибуття трамваїв, тролейбусів та автобусів міста на зупинку. Для того, щоб розпочати роботу з ботом, користувачу необхідно натиснути на кнопку «Розпочати», далі слід ввести код зупинки, що вказаний на знаку або надіслати власне місцезнаходження[1].

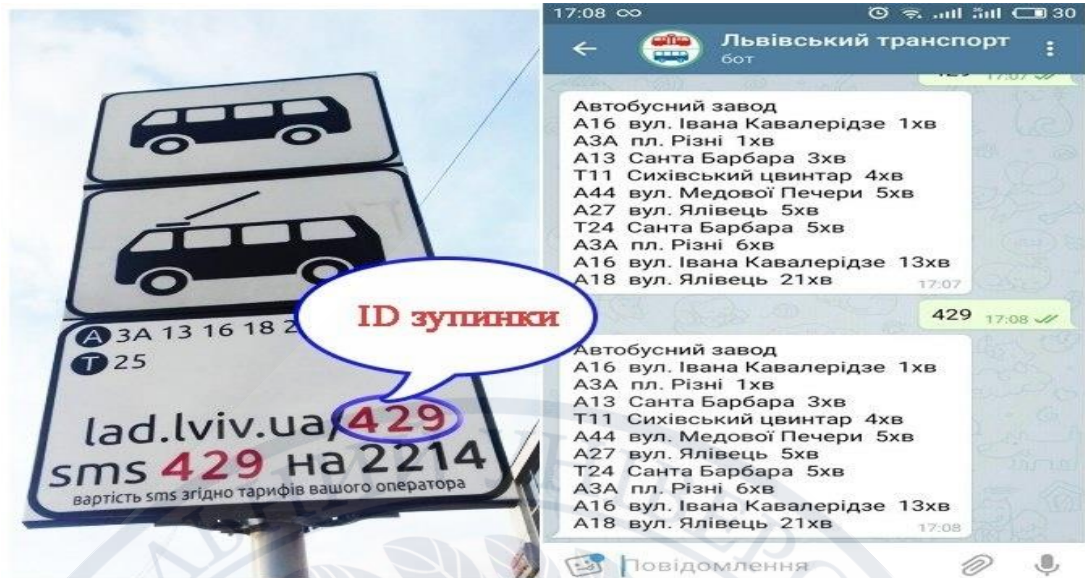


Рисунок 1.1 – Telegram-бот «Львівський транспорт»

Другий варіант – «Tracker»

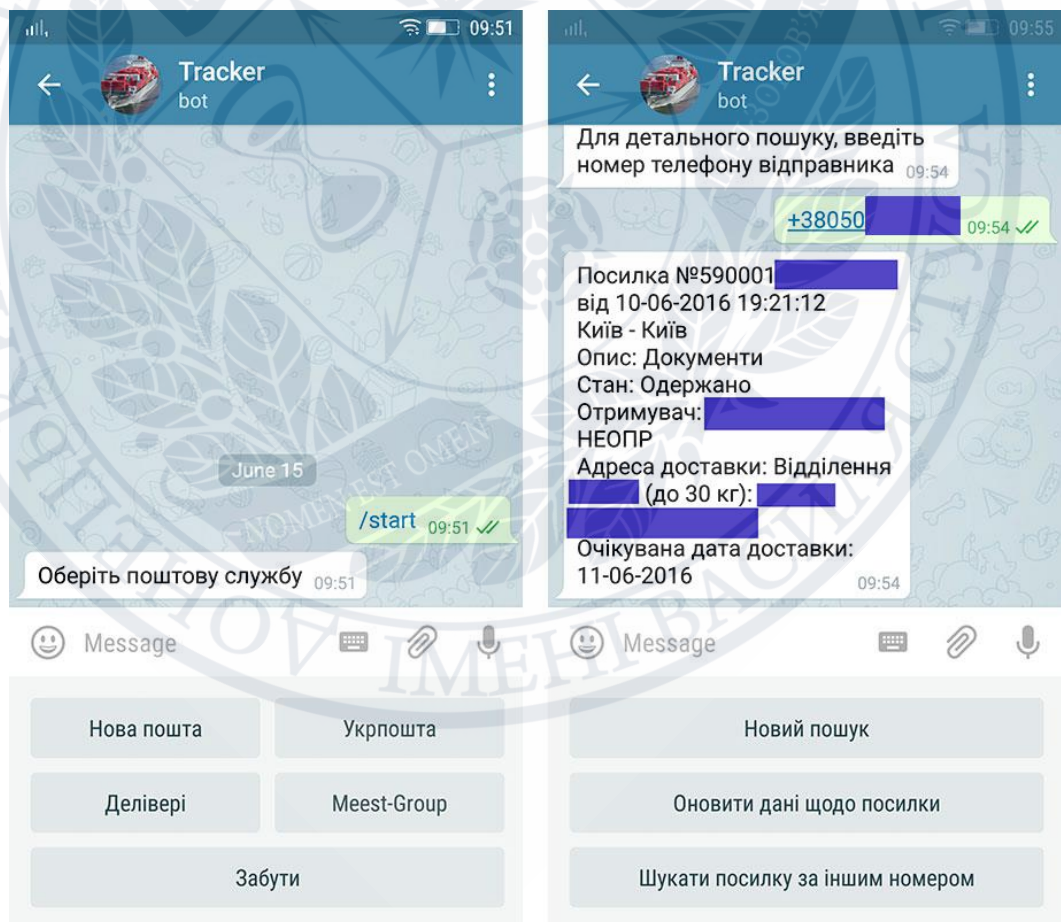


Рисунок 1.2 – Telegram-бот «Tracker»

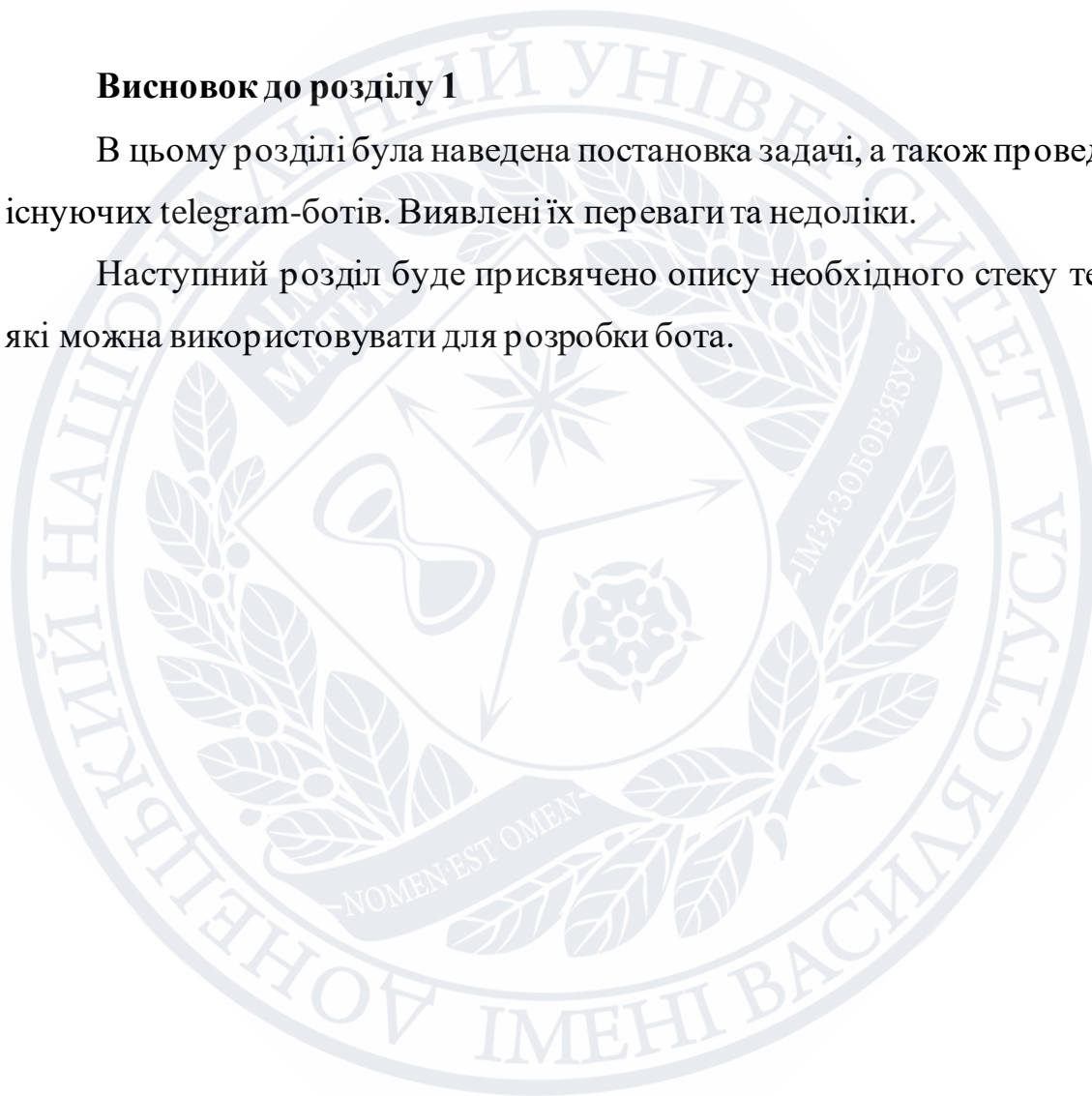


Цей бот відстежує посилки різних операторів. У даному випадку користувачі мають можливість отримувати повідомлення про стан замовлення посилки різних операторів. Щоб почати користуватися ботом, потрібно ввести в пошуку месенджера Telegram його назву – @QTrackerbot і натиснути Start. Потім вибрати поштового оператора, який обслуговує шукане відправлення, і ввести номер посилки[2].

### **Висновок до розділу 1**

В цьому розділі була наведена постановка задачі, а також проведено огляд існуючих telegram-ботів. Виявлені їх переваги та недоліки.

Наступний розділ буде присвячено опису необхідного стеку технологій, які можна використовувати для розробки бота.





## РОЗДІЛ 2

### ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ

В даному розділі описані технології, які були обрані для вирішення поставленої задачі. Існує дуже велика кількість інструментів та технологій для розробки Telegram-ботів, виходячи з їх характеристик були обрані наступні.

#### 2.1 ASP.NET Core

ASP.NET Core – це новий загальнодоступний і багатоплатформовий фреймворк для створення сучасної хмари додатків, пов'язаних з підключенням до інтернету, таких як веб-додатки, додатки для інтернет речей і мобільних серверів.[17] Фреймворк спроектовано таким чином, щоб забезпечити оптимізовану платформу розробки для додатків, які переміщуються в хмару або виконуються локально. Він складається з модульних компонентів з мінімальною перевантаженням, тому ви зберігаєте гнучкість при побудові своїх рішень. Існує можливість розробляти і запускати кросплатформні ASP.NET Core додатки на Windows, Mac і Linux. Фреймворк ASP.NET Core загальнодоступний на GitHub[4].

Перша попередня версія ASP.NET з'явилася майже 15 років тому як частина платформи .NET Framework. З плином часу дуже багато розробників використовують цю технологію для створення і запуску відмінних веб-додатків. За ці роки вдалося додати і розробити безліч можливостей.

ASP.NET Core отримав певні зміни, які призвели до більш компактної та модульної структури. В свої роботі використовує пакети NuGet. Це рішення дозволяє оптимізувати та пришвидшити програму. Переваги меншої площі поверхні програми включають: більш суворий захист, знижений рівень обслуговування, поліпшену продуктивність і зменшення витрат в моделі «плати за те, що використовуєш». [23]

Схематично роботу ASP.NET можна представити наступним чином:

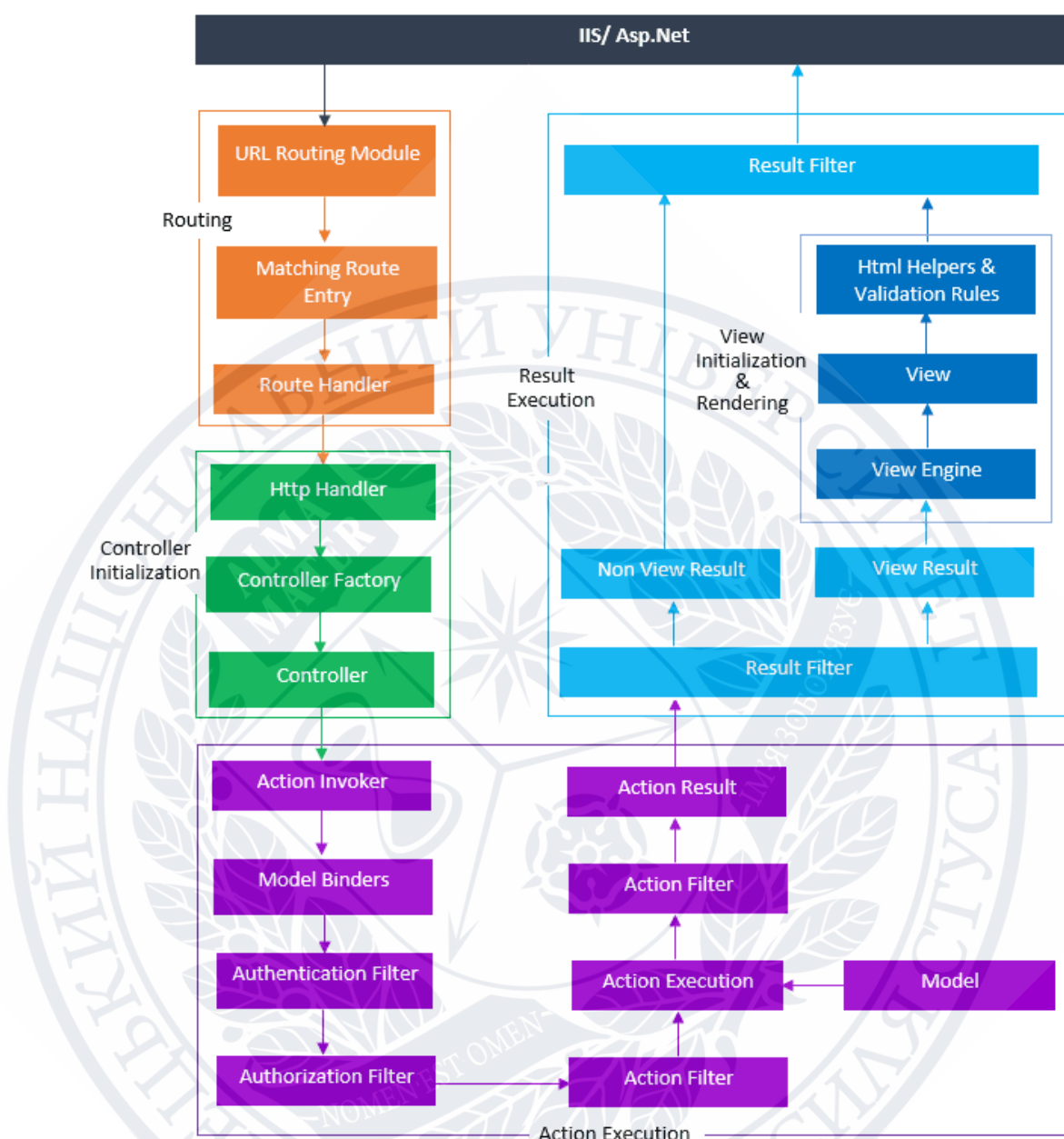


Рисунок 2.1 – Принцип роботи ASP.NET

ASP.NET Core надає наступні переваги:

1. Єдине рішення для користувача та розробника web арі.
2. Розроблено для тестування додатків.
3. Razor Pages зменшує написання коду для сценаріїв сторінок і підвищує його ефективність.
4. Можна використовувати мову C# замість JavaScript для роботи браузера.

5. Відкритий вихідний код і орієнтація на співтовариство.
6. Інтеграція сучасних клієнтських платформ і робочих процесів розробки.
7. Підтримка процедур (RPC) за допомогою gRPC.
8. Хмарна система конфігурації на основі середовища.
9. Вбудоване введення залежностей.
10. Спрощений високопродуктивний модульний конвеєр HTTP-запитів.
11. Наступні можливості розміщення:
12. Kestrel
13. Служби IIS
14. HTTP.sys
15. Nginx
16. Apache
17. Docker
18. Управління паралельними версіями.
19. Інструментарій, що спрощує процес сучасної веб-розробки.

## 2.2 Мова програмування C#

C# - це сучасна, об'єктно-орієнтована та безпечна мова програмування. C# дозволяє розробникам створювати багато типів безпечних та надійних додатків, що працюють в екосистемі .NET. C# належить до сімейства мов C і буде одразу знайомий програмістам на C, C++, Java.

C# - це об'єктно-орієнтована, орієнтована на компоненти мова програмування. C# надає мовні конструкції для прямої підтримки цих концепцій, роблячи C# природною мовою, на якій можна створювати та використовувати програмні компоненти. З моменту свого зародження C# додав функції для підтримки нових навантажень та нових практик проектування програмного забезпечення.

Кілька функцій C# допомагають створювати надійні та довговічні програми. Збір сміття автоматично вилучає пам'ять, зайняту недосяжними невикористаними об'єктами. Допустимі типи захищають від змінних, які не



посилаються на виділені об'єкти. Обробка винятків забезпечує структурований та розширюваний підхід до виявлення та відновлення помилок. Лямбда-вирази підтримують методи функціонального програмування. Синтаксис інтегрованого мовного запиту (LINQ) створює загальний шаблон роботи з даними з будь-якого джерела. Мовна підтримка асинхронних операцій забезпечує синтаксис для побудови розподілених систем. C# має уніфіковану систему типів. Усі типи C#, включаючи примітивні типи, такі як `int` і `double`, успадковуються від одного кореневого типу об'єкта. Усі типи мають спільний набір загальних операцій. Цінності будь-якого типу можна зберігати, транспортувати та використовувати послідовно. Крім того, C# підтримує як визначені користувачем типи посилань, так і типи значень. C# дозволяє динамічно розподіляти об'єкти та зберігати в лінійці легкі конструкції. C# підтримує загальні методи та типи, які забезпечують підвищену безпеку та продуктивність типу. C# надає ітератори, які дозволяють реалізаторам класів колекцій визначати власну поведінку для клієнтського коду.

Програми C# працюють на .NET, віртуальній системі виконання, яка називається середовищем виконання загальної мови (CLR) та набором бібліотек класів. CLR - це впровадження корпорацією Майкрософт загальномовної інфраструктури (CLI), міжнародного стандарту. CLI - це основа для створення середовищ виконання та розробки, в яких мови та бібліотеки працюють безперебійно.

Принцип роботи CLR:

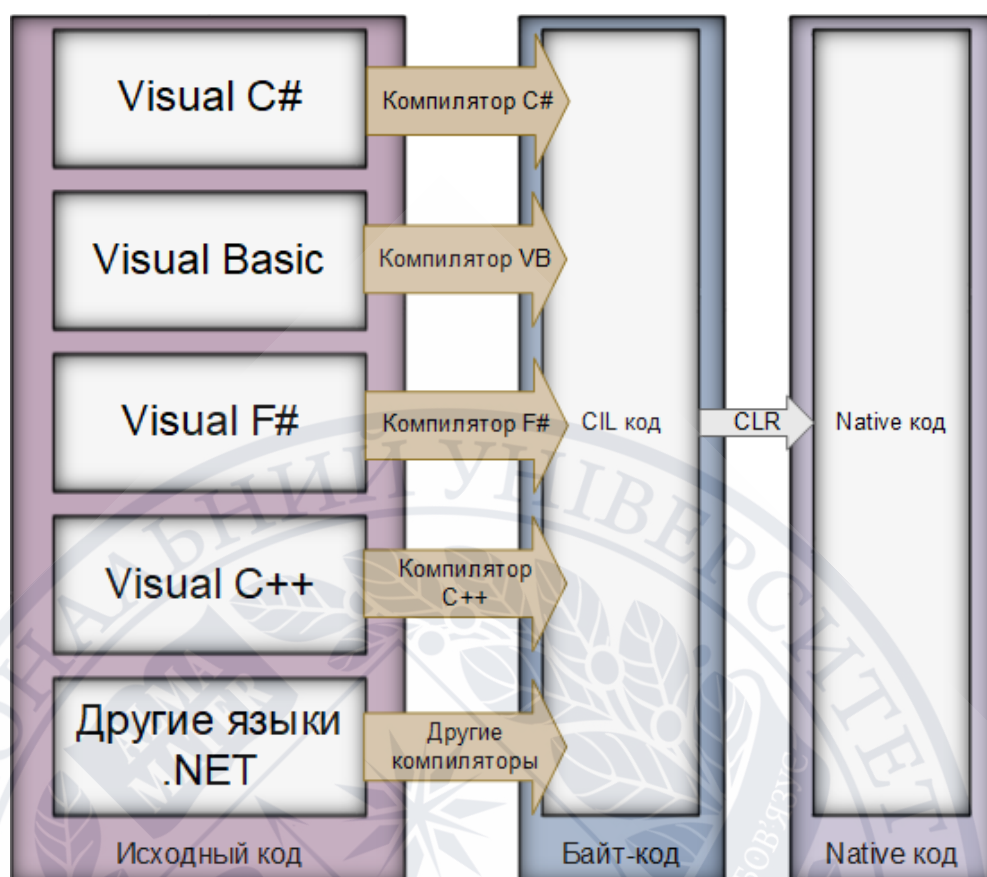


Рисунок 2.2 – Принцип роботи CLR

До вагомих переваг C# належить:

- Один API для розробки на різних мовах.
- Велика кількість NuGet пакетів.
- Реалізовані класи, що використовують різні алгоритми, зменшують терміни розробки програм і підвищують надійність.

### 2.3 Microsoft SQL Server

SQL Server - це реляційна система управління базами даних (СУБД), розроблена та продана корпорацією Майкрософт.

Подібно до іншого програмного забезпечення СУБД, SQL Server побудований поверх SQL, стандартної мови програмування для взаємодії з реляційними базами даних. Сервер SQL пов'язаний з Transact-SQL, або T-SQL, реалізацією Microsoft SQL, яка додає набір власних конструкцій програмування.

SQL Server працює виключно в середовищі Windows більше 20 років. У 2016 році Microsoft зробила це доступним на Linux. SQL Server 2017 став загальнодоступним у жовтні 2016 року для Windows і Linux.

Нижче наведена архітектура SQL Server(Рис. 2.3):

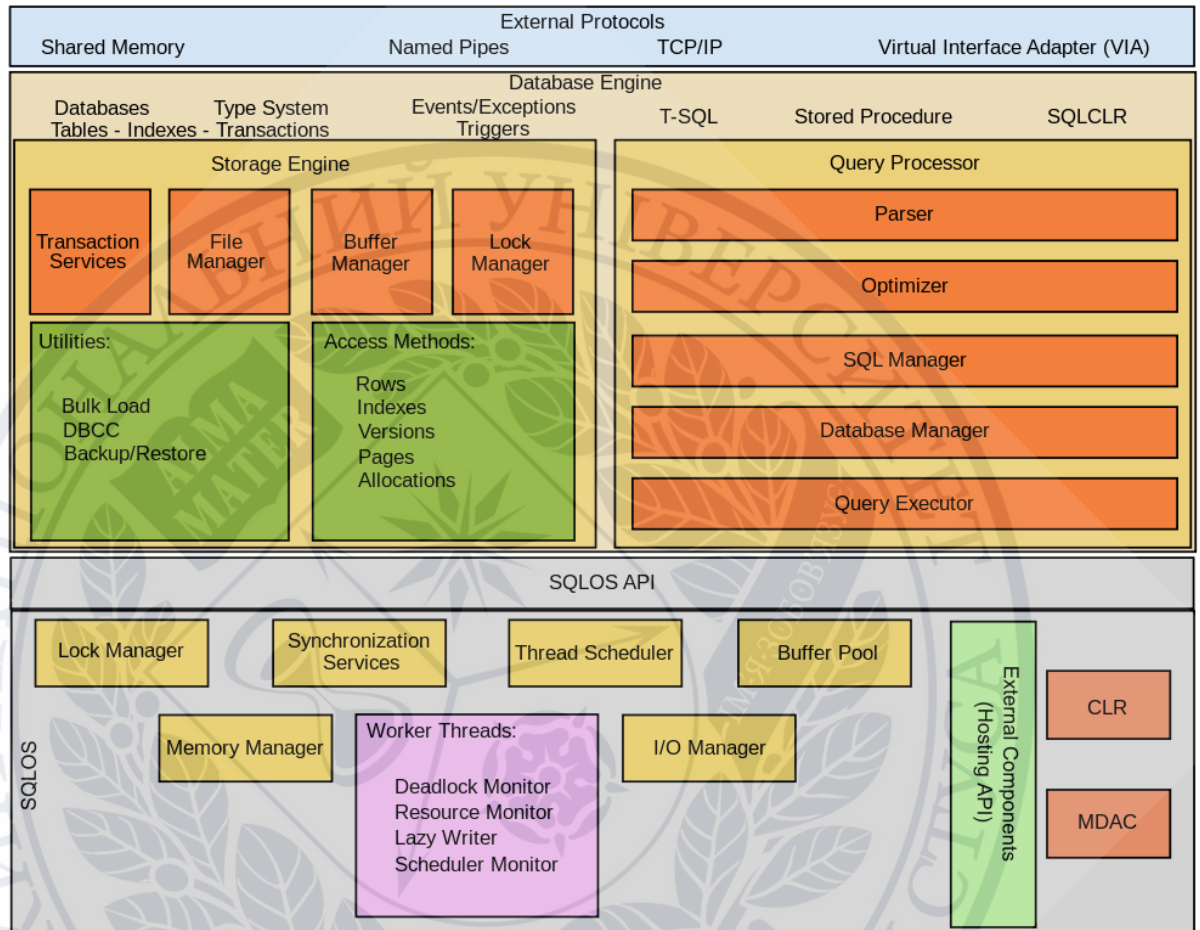


Рисунок 2.3 – Архітектура SQL Server

Основним компонентом SQL Server є Database Engine. Механізм баз даних складається з реляційного механізму, який обробляє запити, та механізму зберігання, який управляє файлами бази даних, сторінками, сторінками, індексом тощо. Об'єкти бази даних, такі як збережені процедури, подання та тригери, також створюються та виконуються механізмом баз даних. Реляційний двигун містить компоненти, які визначають найкращий спосіб виконання запиту. Реляційний механізм також відомий як процесор запитів.



## 2.4 Visual Studio 2019

Вбудоване середовище розробки Visual Studio - це креативна стартова площа, яка використовується для редагування, налагодження та побудови коду, а потім публікування програми. Інтегроване середовище розробки (IDE) - це багатофункціональна програма, яка може бути використана для багатьох аспектів розробки програмного забезпечення. Крім стандартного редактора та налагоджувача, які надає більшість середовищ розробки, Visual Studio включає компілятори, засоби доробки коду, графічні дизайнери та багато інших функцій для полегшення процесу розробки програмного забезпечення.

## 2.5 Docker

Docker Desktop - це проста у встановленні програма для вашого Mac або Windows, яка дозволяє створювати та обмінюватися контейнерними програмами та мікросервісами. До робочого столу Docker входять Docker Engine, клієнт Docker CLI, Docker Compose, нотаріус, Kubernetes та Credential Helper.

Docker Desktop працює з вашим вибором засобів розробки та мов і надає вам доступ до величезної бібліотеки сертифікованих зображень та шаблонів у Docker Hub. Це дозволяє командам розробників розширити своє середовище для швидкої автоматичної збірки, постійної інтеграції та співпраці за допомогою безпечного сховища.

Деякі ключові особливості Docker Desktop:

- Можливість вмістити та поділитися будь-якою програмою на будь-якій хмарній платформі різними мовами та структурами
- Проста установка та налаштування повного середовища розробки Docker
- Використання останньої версії Kubernetes.
- Автоматичні оновлення для постійного оновлення та захисту
- У Windows можливість перемикання середовищ Linux та Windows Server для створення додатків
- Швидка та надійна робота завдяки власній віртуалізації Windows Hyper-V

- Коренева файлова система для контейнерів створюється з використанням механізму сору-on-write (окремо зберігаються тільки змінені і нові дані), що дозволяє прискорити розгортання, знижує витрату пам'яті і економить дисковий простір.
- Можливість власної роботи на Linux через WSL 2 на машинах Windows
- Змінена файлова система одного контейнера може використовуватися як основа для формування нових базових образів і створення інших контейнерів, без необхідності оформлення шаблонів або ручного налаштування складу образів.
- Монтаж томів для коду та даних, включаючи сповіщення про зміну файлів та легкий доступ до запущених контейнерів у локальній мережі
- Можливість створення контейнерів, що містять складні програмні стеки, через зв'язування між собою вже існуючих контейнерів, що містять складові частини формованого стека. [24]

Нижче наведена структура Docker та принцип його функціонування.

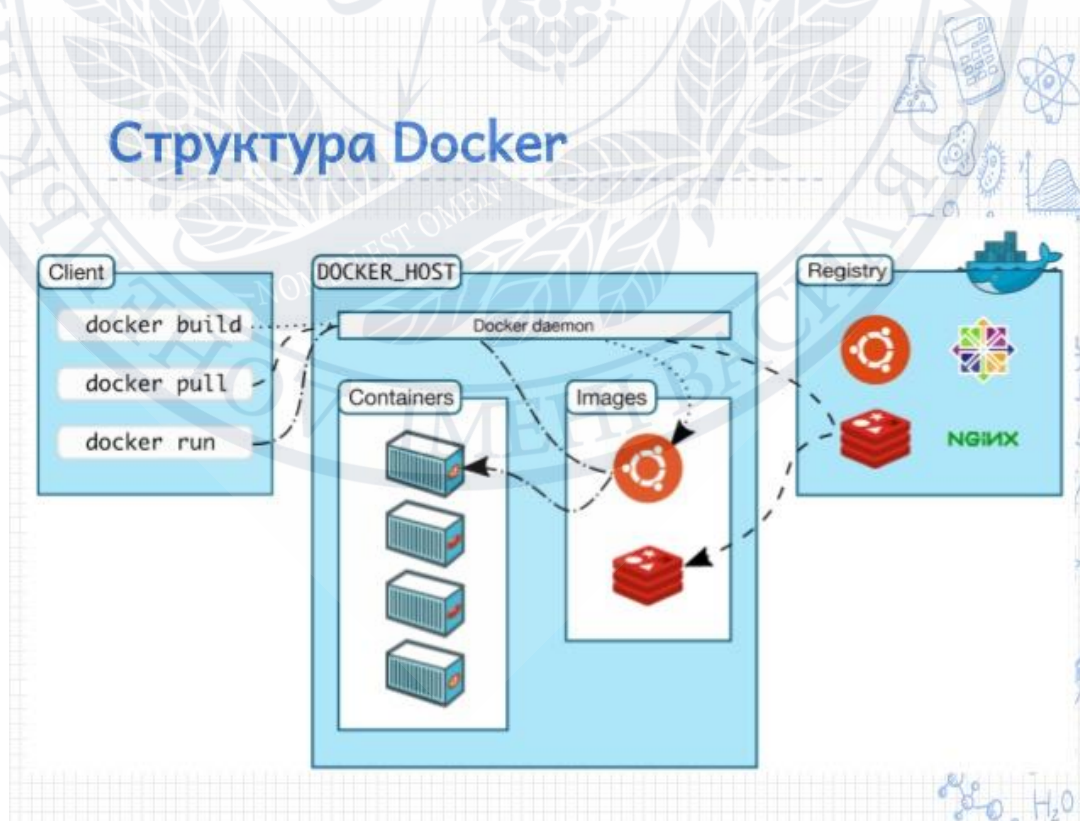


Рисунок 2.4 – Структура Docker та принцип його функціонування

## 2.6 HTTPS

HTTPS (аббр. Від англ. HyperText Transfer Protocol Secure) – покращена версія протоколу HTTP, яка підтримує шифрування для покращення безпеки. Дані в HTTPS передаються поверх криптографічних протоколів TLS або застарілого в 2015 році SSL. Якщо порівняти з HTTP, то HTTPS працює на 443 порту, а HTTP на 80. Цей протокол був розроблений в 1994 році компанією Netscape Communications для свого браузера, але зараз він доступний в усіх браузерах.

### Висновок до розділу 2

У цьому розділі наведені характеристики інструментів, що використовувалися для проектування телеграм-бота. У наступному розділі буде розглядатися сам процес розробки.



## РОЗДІЛ 3

### РОЗРОБКА ТЕЛЕГРАМ-БОТА

В даному розділі наведено опис розробки телеграм-бота, проведено аналіз вибору засобів та інструментів та зразок розроблення телеграм-боту за їх допомогою.

#### 3.1 Огляд BotFather

На початку розроблення програми слід визначити засоби та інструменти, які будуть використовуватися для подальшої роботи. До них ми можемо віднести:

1. Мова програмування.
2. Середовище розробки.
3. Засоби хостингу застосунків.
4. Система контролю версій.
5. Додаткові засоби та інструменти.

На рис. 3.1 представлений бот – BotFather, за допомогою якого можна створити та здійснювати управління іншими Telegram-ботами. Для початку роботи необхідно розпочати його роботу, натиснувши /start. Головна мета полягає у тому, щоб отримати токен авторизації бота. Токен необхідний для роботи з Bot API за допомогою http або https-протоколу, та використання програмою при виникненні будь яких запитів до серверу.[3]

Далі необхідно відправити в чат команду /newbot та слідувати наступним крокам:

1. Для бота необхідно обрати ім'я, воно буде відображатись користувачам. Потім бота можна перейменувати.
2. Ім'я повинно бути унікальним і закінчуватися на «bot», воно може включати лише латинські символи, цифри та підкреслення (наприклад – «My\_Test\_1\_Bot»). Загальна кількість символів складає від 5 до 32.

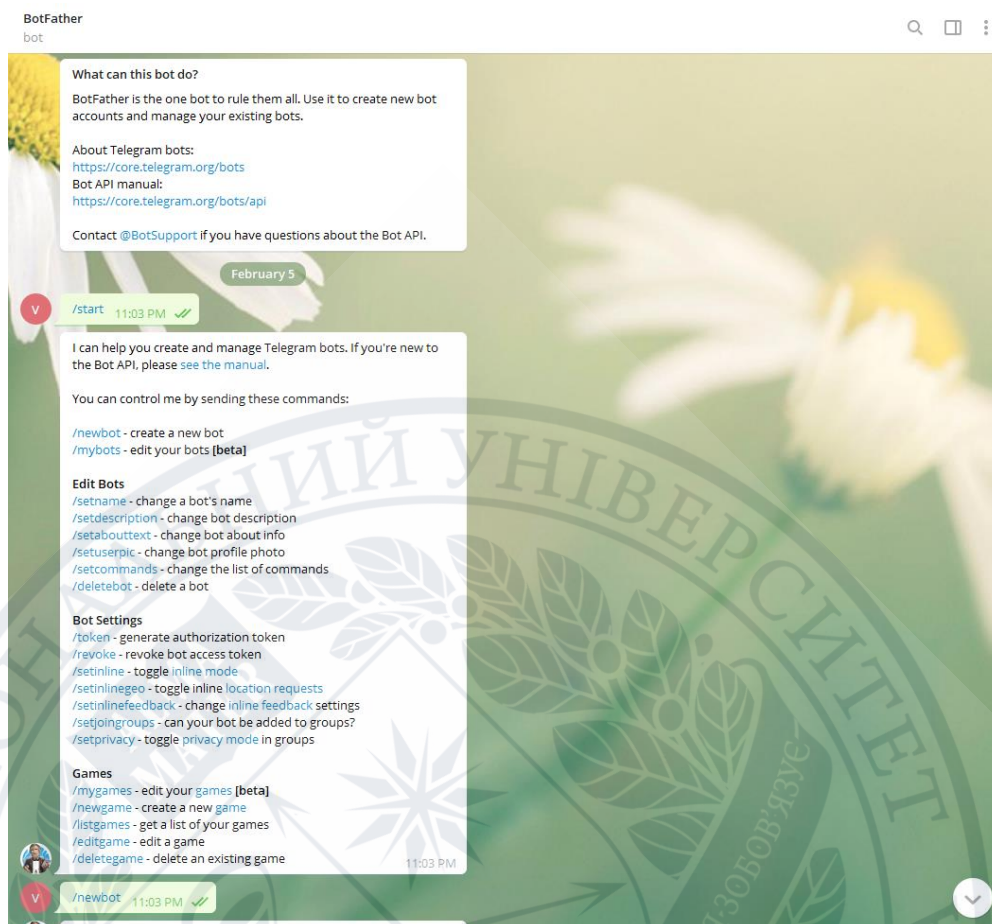


Рисунок 3.1 – BotFather

3. Якщо все виконано правильно, то у відповідь прийде токен. Його не можна передавати іншим і бажано не втрачати. Наприклад, можна скопіювати його, зберегти в текстовий файл та покласти у хмарне сховище.

BotFather – це бот для керування своїми ботами. Команди для редагування (рис. 3.2):

- 1) /newbot – використовується для створення ботів;
- 2) /mybots – повертає своїх ботів та дозволяє налаштовувати їх;
- 3) /mygames – повертає свої ігри та дозволяє редагувати їх налаштування;
- 4) /setname – для зміни імені бота;
- 5) /setdescription – для зміни опису бота;

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

[/newbot](#) - create a new bot  
[/mybots](#) - edit your bots [beta]

#### Edit Bots

[/setname](#) - change a bot's name  
[/setdescription](#) - change bot description  
[/setabouttext](#) - change bot about info  
[/setuserpic](#) - change bot profile photo  
[/setcommands](#) - change the list of commands  
[/deletebot](#) - delete a bot

#### Bot Settings

[/token](#) - generate authorization token  
[/revoke](#) - revoke bot access token  
[/setinline](#) - toggle inline mode  
[/setinlinegeo](#) - toggle inline location requests  
[/setinlinefeedback](#) - change inline feedback settings  
[/setjoiningroups](#) - can your bot be added to groups?  
[/setprivacy](#) - toggle privacy mode in groups

#### Games

[/mygames](#) - edit your games [beta]  
[/newgame](#) - create a new game  
[/listgames](#) - get a list of your games  
[/editgame](#) - edit a game  
[/deletegame](#) - delete an existing game

11:03 PM

Рисунок 3.2 – Команди BotFather

6) [/setabouttext](#) – для зміни інформації про бота. Цей текст видно на сторінці профілю бота (рис. 3.3);



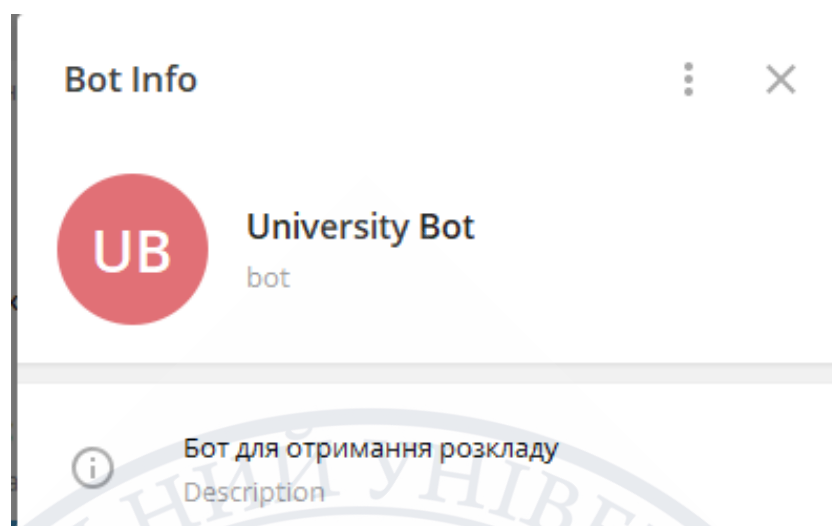


Рисунок 3.3 – Опис бота

- 7) /setuserpic – для зміни зображення профілю бота;
- 8) /setcommands – для зміни списку команд, що підтримуються ботом;
- 9) /deletebot – дозволяє видалити бота;
- 10) /setinline – налаштування використання бота в будь-якому чаті;
- 11) /setinlinegeo – дозволяє боту отримувати значення про місцезнаходження користувача;
- 12) /setjoiningroups – налаштування можливості додавання до груп. Якщо бот розроблений для особистих повідомлень і не повинен обробляти повідомлення з груп, то цю опцію можна вимкнути;
- 13) /setprivacy – налаштування команд, які бот буде отримуватиме в групах. Якщо ця команда вимкнена, то бот отримуватиме всі повідомлення.

В документації Telegram Bot описано, два способи отримання оновлень:

Перший спосіб – періодичні запити, це простий варіант отримати оновлення від серверів Telegram. Його суть полягає в тому, щоб періодично опитувати сервер на наявність нових запитів. Цей зв'язок називається Long Polling.

Long Polling – спеціальний метод, за допомогою якого можна отримати інформацію про нові події з використанням так званих «довгих запитів».[10] На сервер приходить HTTP-запит, проте відповідь надсилає не одразу, а лише після

того, як було виконано якусь дію (наприклад, отримано нове повідомлення) або після закінчення часу на очікування дії (рис. 3.4)[9].

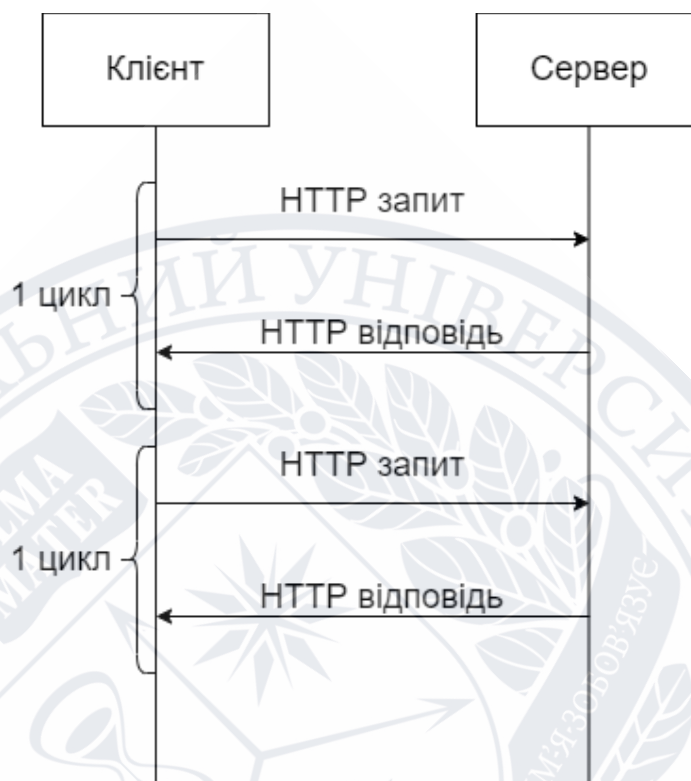


Рисунок 3.4 – принцип роботи Long Polling

Другий спосіб отримання оновлень – це установка веб-хуків (рис. 3.5).

Веб-хуки працюють інакше, якщо приходить запит, то телеграм сам повідомляє. В цьому разі відпадає необхідність опитувати сервера телеграм, тим самим зменшити на них навантаження.[13] Проте необхідно встановити веб-сервер на пристрій, щоб була можливість запустити бота.

Також для коректної роботи необхідно використовувати SSL (Secure Sockets Layer) сертифікат, бо Telegram Web-hooks працюють лише бо HTTPS-сертифікату.

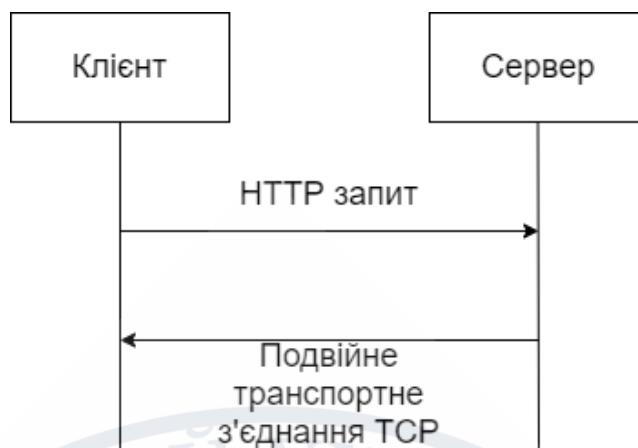


Рисунок 3.5 – Принцип роботи веб-хуків

Був обраний другий спосіб, тому що веб-хуки зменшують навантаження на сервери телеграм та використовують. Нижче розглянуто принцип роботи HTTPS та переваги над HTTP.

Вхідні оновлення зберігаються, поки сервер їх не отримає або поки не пройде 24 години. У відповідь отримуємо серіалізований JSON[14] об'єкт Update.

```

{
  "scores": [
    {
      "Away_Score": 2,
      "Away_Team": "Newcastle",
      "Home_Score": 2,
      "Home_Team": "Arsenal"
    },
    {
      "Away_Score": 2,
      "Away_Team": "Napoli",
      "Home_Score": 4,
      "Home_Team": "Liverpool"
    }
  ]
}

```

Рисунок 3.6 – Приклад JSON об'єкту

### Принцип роботи HTTPS:

HTTPS – це не окремий протокол. Це звичайний HTTP, який шифрує через транспортні механізми SSL і TLS. Він гарантує захист від атак, заснованих на прослуховуванні мережевого з'єднання – від сніфферських атак і атак типу man-in-the-middle, це тільки тоді, коли будуть використовувати шифрувальні засоби і сертифікат сервера перевірений і йому довіряють.

Якщо не встановлювати порт для HTTPS, то буде використаний 443 TCP-порт (для HTTP за замовчуванням буде використаний – 80). Для підготовки HTTPS веб-сервера, адміністратор повинен встановити сертифікат відкритого і закритого ключа для цього веб-сервера. Для TLS використовується асиметрична схема шифрування (для вироблення загального секретного ключа), і симетрична (для обміну даними, зашифрованими загальним ключем).

Існує можливість створити такий сертифікат, не звертаючись до центру сертифікації. Підписуються такі сертифікати цим же сертифікатом і називаються самопідписаними (self-signed).

Ця система також може використовуватися для аутентифікації клієнта, щоб забезпечити доступ до сервера тільки авторизованим користувачам. Для цього адміністратор зазвичай створює сертифікати для кожного користувача і завантажує їх в браузер кожного користувача. Також будуть прийматися всі сертифікати, підписані організаціями, яким довіряє сервер.

Деякі старі версії браузерів використовують довжину ключа 40 біт (приклад тому - ІЕ версій до 4.0), що пов'язано з експортними обмеженнями в США. Довжина ключа 40 біт не є надійною. Багато сучасних сайтів вимагають використання нових версій браузерів, що підтримують шифрування з довжиною ключа 128 біт, з метою забезпечити достатній рівень безпеки. Шифрування з довжиною ключа 128 біт значно ускладнює підбір паролів і доступ до особистої інформації [15].

Традиційно на одну IP-адресу може працювати тільки один HTTPS-сайт.

На рисунку 3.7 показаний принцип роботи HTTPS та HTTP.



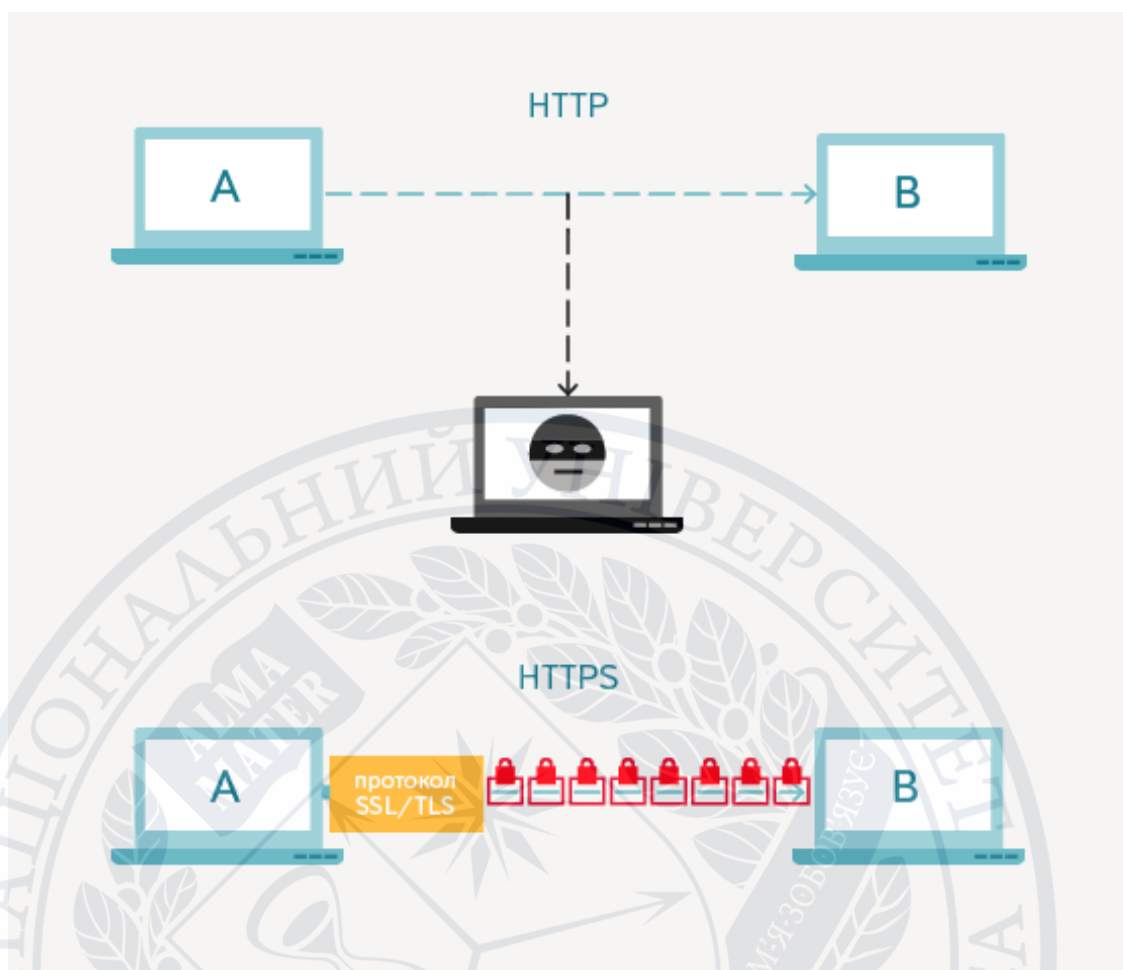


Рисунок 3.7 – Принцип роботи HTTPS та HTTP

### 3.2 Ознайомлення з Telegram API:

Наступний крок – ознайомлення із прикладним програмним інтерфейсом (англ. Application Programming Interface, API), який необхідно для розробників.

API - це набір визначень та протоколів для побудови та інтеграції прикладного програмного забезпечення. API означає інтерфейс прикладного програмування.

API дозволяють вашому товару або послугі взаємодіяти з іншими продуктами та послугами, не знаючи, як вони впроваджені. Це може спростити розробку додатків, заощадити час і гроші. Коли ви розробляєте нові інструменти та продукти - або керуєте існуючими - API надають вам гнучкість; спростити дизайн, адміністрування та використання; та надають можливості для інновацій.

API іноді розглядають як контракти з документацією, що представляє домовленість між сторонами: якщо сторона 1 надсилає віддалений запит, структурований певним чином, це те, як реагує програмне забезпечення сторони

Оскільки API спрощують те, як розробники інтегрують нові компоненти програми в існуючу архітектуру, вони допомагають бізнесу та IT-командам співпрацювати. Потреби бізнесу часто швидко змінюються у відповідь на постійно мінливі цифрові ринки, де нові конкуренти можуть змінити цілу галузь за допомогою нового додатка. Щоб залишатися конкурентоспроможними, важливо підтримувати швидкий розвиток та впровадження інноваційних послуг. Розробка власних хмарних додатків - це ідентифікований спосіб збільшити швидкість розробки, і він покладається на підключення архітектури додатків мікросервісів через API.

Telegram Bot API – це API для створення та керування телеграм ботом. Цей API дозволяє підключати ботів до системи Телеграм. Телеграм використовує протокол MTProto [27] для шифрування. MTProto – криптографічний протокол, що використовується в системі обміну повідомленнями Telegram для шифрування переписок користувачів. В основі протоколу лежить оригінальна комбінація симетричного алгоритму шифрування AES (в режимі IGE), протокол Диффі-Хеллмана для обміну 2048-бітними RSA-ключами між двома пристроями та ряд хеш-функцій. Протокол допускає використання шифрування наскрізних з неонов’язковою парою ключів. Також на основі цього протоколу був створений MTProху. [21]

Схематично роботу MTProto можна зобразити так:

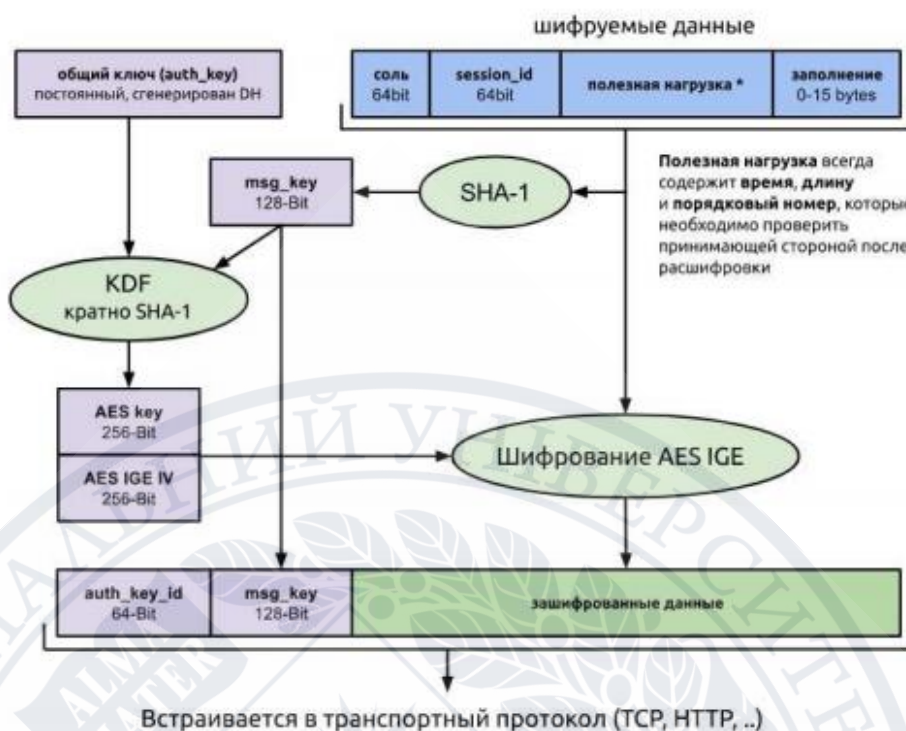


Рисунок 3.8 – Принцип работы MTProto

Для спілкування з API Telegram немає необхідності знати про принцип дії MTProto, сервер телеграм сам буде робити всю роботу. Всі запити до Телеграм будуть передаватись через HTTPS-протокол та мати такий вигляд: <https://api.telegram.org/bot/token/method>. Замість токена має міститися токен авторизації бота, а замість method – метод який використовується в Telegram API.

### 3.3 Вибір мови програмування та середовища розробки

Мову програмування можна обрати любую, але потрібно звернути увагу чи є бібліотека для роботи з Telegram API. Якщо немає, то потрібно обрати іншу мову програмування або написати свою реалізацію бібліотеки для спілкування з Telegram API. Для цього була обрана мова C# від компанії Microsoft.[25]

Для .NET платформи є декілька бібліотек для роботи з Telegram API. Наприклад TelegramBotApi, TeleBot, TelegramBots. Для роботи було обрано TelegramBots, тому що це найпопулярніша бібліотека, яка надає найбільший функціонал для роботи з Telegram API, а також має гарну документацію.

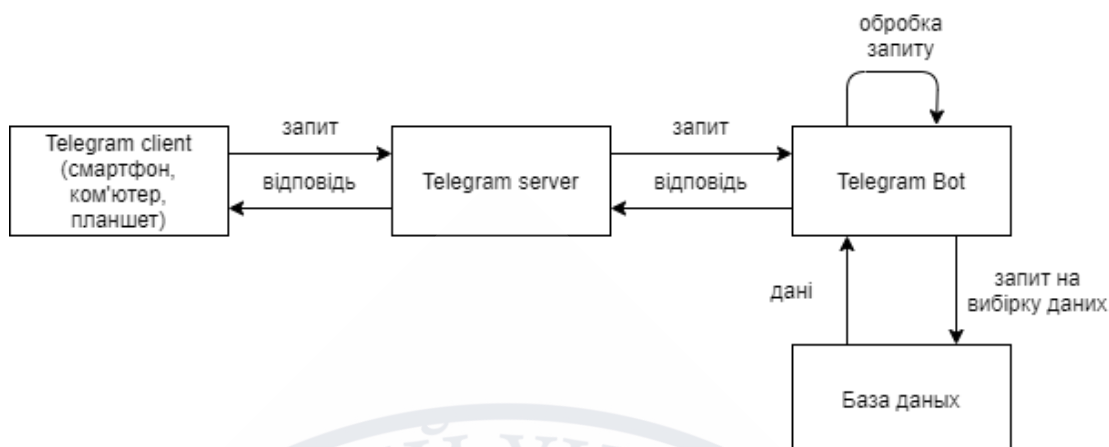


Рисунок 3.9 – Принцип роботи Telegram бота

Спочатку було створено порожній ASP.NET Core, потім підключено бібліотеку TelegramBots та створено тестовий роут для відправки повідомлення, яке надіслав користувач. Лістинг тестового роуту наведений нижче:

```

[HttpPost]
[Route("api/message/update")]
public async Task<IActionResult> Post([FromBody] Update update)
{
    await telegramBotClient.SendTextMessageAsync(update.Message.Chat.Id,
        update.Message.Text);
    return Ok();
}
  
```

Далі було розроблено структуру проекту та поділено її на 3 шари:

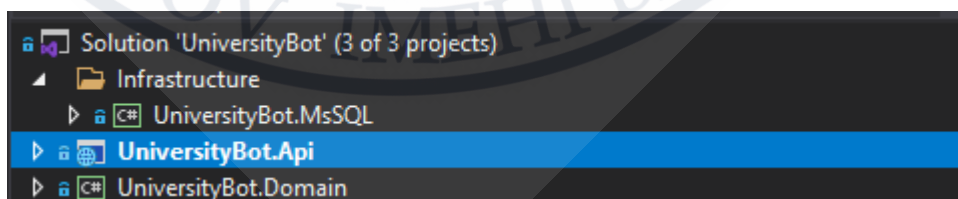


Рисунок 3.9 – Структура проекту

- UniversityBot.Api – це HTTPS сервер на який Telegram кидає запити від користувачів.



- UniversityBot.MsSQL – запити до бази даних.
- UniversityBot.Domain – бізнес логіка бота.

Далі потрібно вибрати базу даних та спроектувати таблиці.

### 3.4 База даних

База даних (БД) – це організована структура, призначена для зберігання, зміни та обробка взаємозв’язаної інформації, переважно більших об’єктів. [16]

Для збереження даних про розклад обрано Microsoft SQL Server. SQL Server – це система управління базами даних, в системі запитів використовується SQL.

Специфіка роботи сервера бази даних SQL Server полягає в транзакційній обробці даних. Тобто, після кожного запиту до бази, буде зберігатись певна інформація про запит.

Використання SQL Server дозволяє автоматизувати рішення різних бізнес-задач, робити аналітику даних в режимі онлайн, відслідковувати всі запити до ресурсів СУБД, управляти транзакціями (операціями з обробки даних).

Нижче наведена схема бази даних (рис. 3.10). Вона складається з 7-ти таблиць.

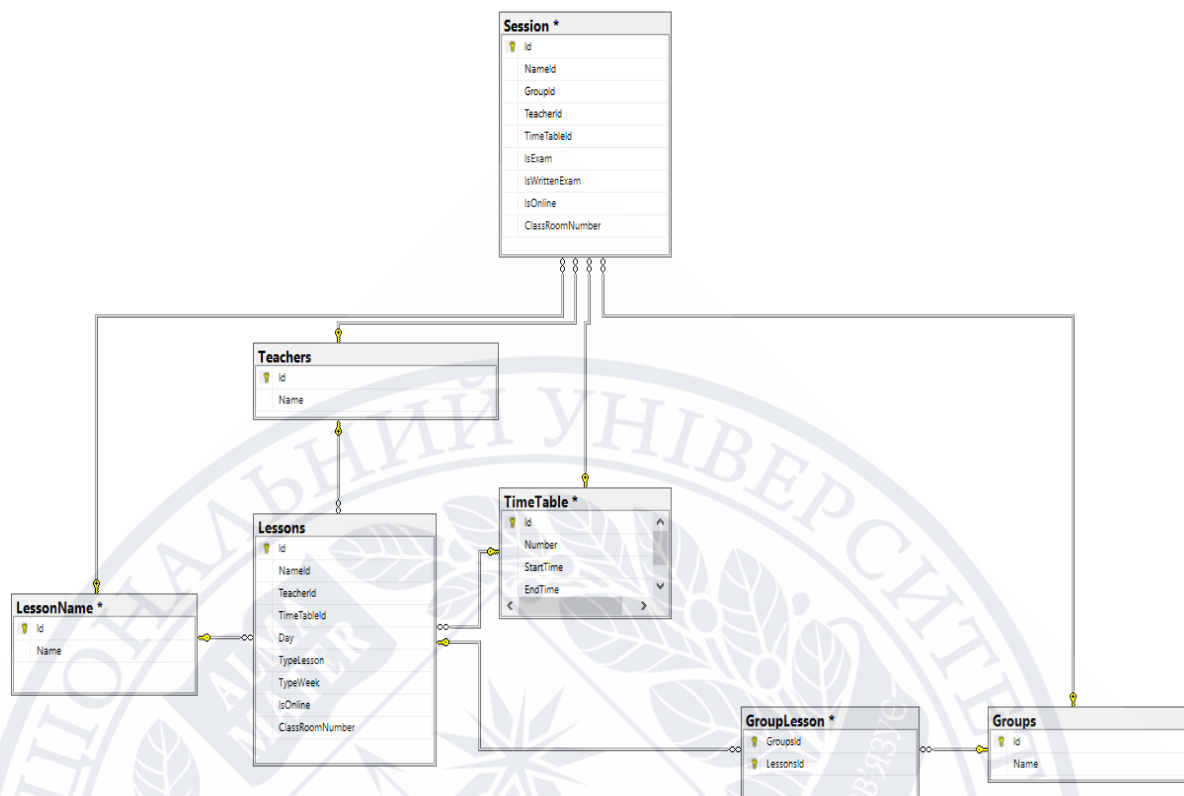


Рисунок 3.10 - Схема бази даних

### 3.5 Команди для бота

Всі команди, які вміє виконувати бот, наслідуються від класу `Command`:

```

public abstract class Command
{
    public abstract string Name { get; }
    public abstract Task Execute(Message message, ITelegramBotClient client);
    public abstract bool Contains(Message message);
}
  
```

Приклад реалізації команди для отримання розкладу пар:

```

public class LessonCommand : Command
{
    public override string Name => "\U0001F4C5 Розклад пар";
    public override async Task Execute(Message message, ITelegramBotClient
client)
  
```

```

{
    var chatId = message.Chat.Id;
    var keyBoards = new InlineKeyboardMarkup(new[]
    {
        new []
        {
            InlineKeyboardButton.WithCallbackData("Розклад вчителя"),
            InlineKeyboardButton.WithCallbackData("Розклад групи"),
        }
    });
    await client.SendTextMessageAsync(chatId, "Виберіть розклад який вам потрібен", replyMarkup: keyBoards);
}

public override bool Contains(Message message)
{
    return message.Type == MessageType.Text &&
    message.Text.Contains(Name);
}
}

```

### 3.6 Тестування програми та результати її виконання

Для початку роботи з ботом, користувач повинен написати /start (Рис. 3.11).

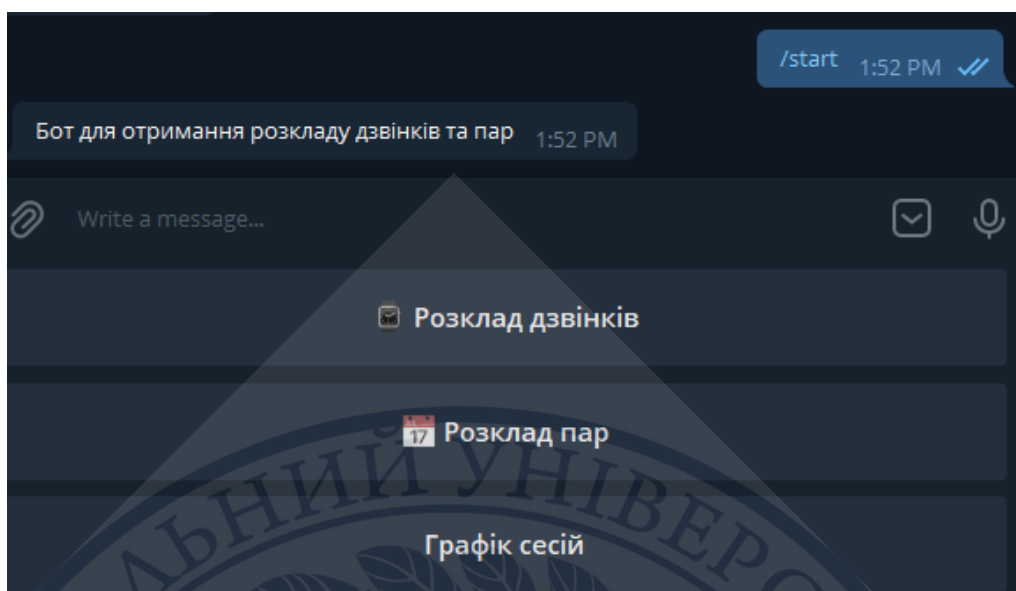


Рисунок 3.11 – Початок роботи з ботом

Далі потрібно вибрати, що користувач хоче дізнатись від бота.

1. Розклад дзвінків – ця команда надсилає користувачу розклад дзвінків (Рис 3.12).

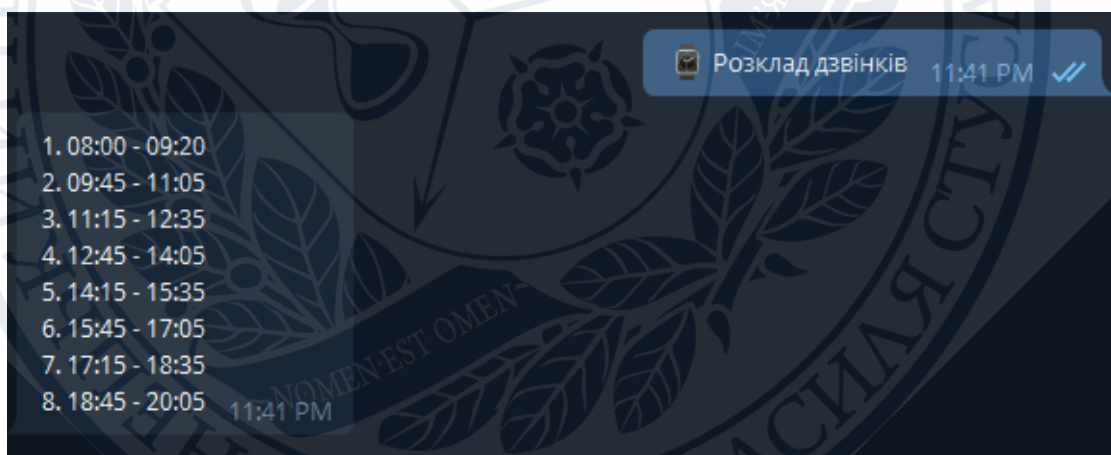


Рисунок 3.12 – Відповідь бота на команду «Розклад дзвінків»

2. Розклад пар – це команда для отримання розкладу пар, яка має 2 сценарії(Рисунок 3.13). Перший сценарій – розклад вчителя(Рис. 3.14), другий сценарій - розклад групи(Рис. 3.15).



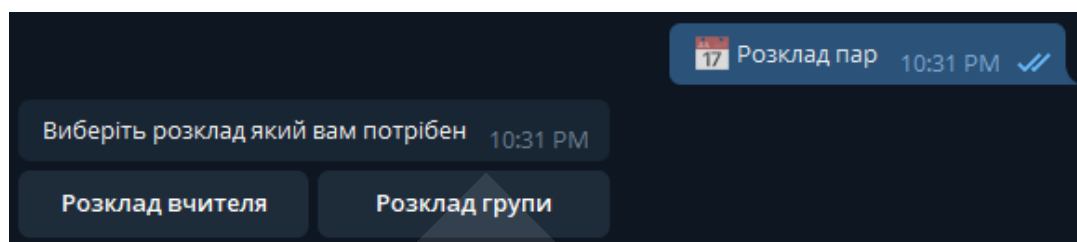


Рисунок 3.13 – Види розкладу

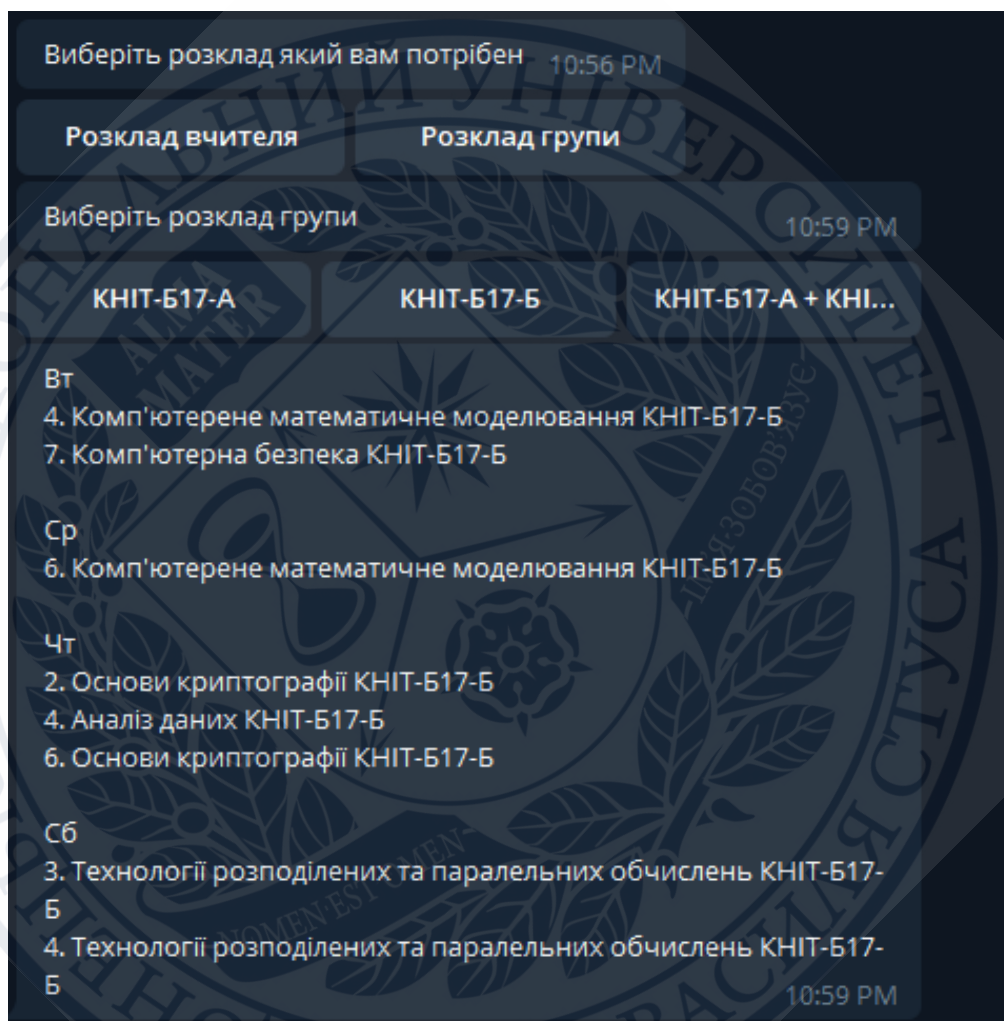


Рисунок 3.14 – Розклад групи

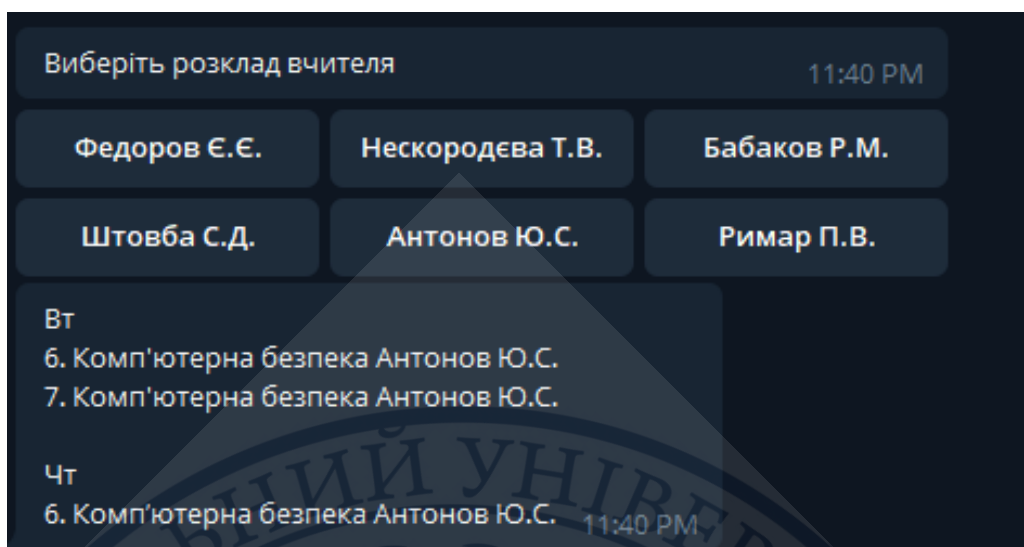


Рисунок 3.15 – Розклад вчителя

3. Графік сесії – команда для отримання графіку сесії вибраної групи.  
(Рис. 3.16).

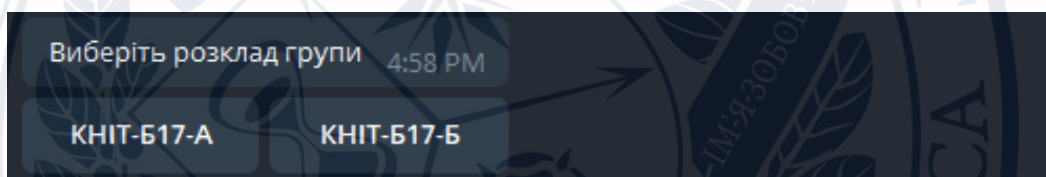


Рисунок 3.16 – Вибір розкладу сесії.

### Висновок до розділу 3

У даному розділі описано розробку Telegram боту, а також опис протоколів та принципи функціонування інструментів за допомогою яких створювався Telegram Bot.

## ВИСНОВКИ

У сучасному світі спостерігається зростання популярності месенджерів як незамінних засобів спілкування. Люди щодня активно використовують можливості та переваги, які їм надає онлайн спілкування. Проте, також, слід відзначити чат-боти, які сьогодні набирають все більшої популярності серед користувачів різноманітних месенджерів.

Усі необхідні для користувачів речі, можуть бути представлені в рамках месенджерів. Наприклад: відслідковування посилок, прогноз погоди, пошук розкладу громадського транспорту, служба підтримки та багато іншого, усе це відтепер знаходиться в одному месенджері, завдяки чому користувач може економити свій час, та зручно використовувати всі додатки в одному місці. Отже, користувачам більше не потрібно шукати та завантажувати всі додатки у свій смартфон, достатньо просто завантажити лише один месенджер та необхідні чат-боти, які не будуть займати так багато місця в пам'яті смартфона.

В рамках бакалаврської роботи були виконані поставлені завдання. По-перше, були вивчені різноманітні Telegram-боти, принципи їх роботи, переваги та недоліки їхнього використання. По-друге, було досліджено аналоги Telegram-боту для студентів та викладачів, у якому можна знайти розклад занять та дзвінків. У процесі дослідження було з'ясовано, що вони працюють неналежним чином. Виходячи з останнього виконаного завдання було обрано технології для розробки чат-бота серед яких ASP.NET Core, SQL Server, Docker і Visual Studio.

Таким чином, результатом роботи є повністю реалізований чат-бот для допомоги студентам і викладачам університету під час навчання.



## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Новини міста Львів. URL: <https://lviv.vgorode.ua/news/sobytyia/392054-zapytai-u-chat-bota-yak-vidslidkovuvaty-lvivskyi-transport-u-Telegram> (дата звернення: 24.04.2021)
2. Інтернет-бізнес в Україні. URL: <https://ain.ua/2016/06/15/ukraincy-zapustili-telegram-bota-kotoryj-otslezhivaet-posylki-na-ukrpochte-i-drugix-krupnyx-operatorax/> (дата звернення: 24.04.2021)
3. Вікіпедія – вільна енциклопедія. URL: <https://ru.wikipedia.org/wiki/Telegram> (дата звернення: 24.04.2021)
4. ІТ-курси онлайн. URL: [https://itvdn.com/ru/blog/article/aspnet\\_core](https://itvdn.com/ru/blog/article/aspnet_core) (дата звернення: 25.04.2021)
5. Вікіпедія – вільна енциклопедія. URL: [https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B8%D0%B9\\_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) (дата звернення: 25.04.2021)
6. Вікіпедія – вільна енциклопедія. URL: [https://uk.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server) (дата звернення: 27.04.2021)
7. Вікіпедія – вільна енциклопедія. URL: [https://uk.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio) (дата звернення: 27.04.2021)
8. Yvision – комунікаційний центр. URL: <https://yvision.kz/post/70339> (дата звернення: 01.05.2021)
9. Навчання мові JavaScript. URL: <https://javascript.ru/ajax/comet/long-poll> (дата звернення: 01.05.2021)
10. Навчання мові JavaScript. URL: <https://learn.javascript.ru/long-polling> (дата звернення: 01.05.2021)

11. Tproger – сайт про програмування. URL: <https://tproger.ru/translations/what-are-web-sockets/> (дата звернення: 01.05.2021)
12. Вікіпедія – вільна енциклопедія. URL: <https://ru.wikipedia.org/wiki/WebSocket> (дата звернення: 01.05.2021)
13. Вікіпедія – вільна енциклопедія. URL: <https://ru.wikipedia.org/wiki/Webhook> (дата звернення: 01.05.2021)
14. Вікіпедія – вільна енциклопедія. URL: <https://ru.wikipedia.org/wiki/JSON> (дата звернення: 02.05.2021)
15. Вікіпедія – вільна енциклопедія. URL: <https://ru.wikipedia.org/wiki/HTTPS> (дата звернення: 02.05.2021)
16. Хостинг сайтів. URL: <https://hostiq.ua/wiki/database/> (дата звернення: 02.05.2021)
17. Dotnet platform. URL: <https://dotnet.microsoft.com/apps/aspnet> (дата звернення: 02.05.2021)
18. Visual Studio. URL: <https://visualstudio.microsoft.com/ru/> (дата звернення: 03.05.2021)
19. Amazon. URL: <https://aws.amazon.com/ru/docker/> (дата звернення: 03.05.2021)
20. Telegram API: <https://core.telegram.org/bots/api> (дата звернення: 03.05.2021)
21. Telegram API: <https://core.telegram.org/mtproto> (дата звернення: 03.05.2021)
22. Microsoft документація. URL: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019> (дата звернення: 04.05.2021)
23. <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>
24. Бібліотека програміста. URL: <https://proglib.io/p/docker> (дата звернення: 04.05.2021)
25. Microsoft документація. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (дата звернення: 04.05.2021)
26. Киреева Н.А., Родіонов А.С., Фархутдінов Р.И., Хусаїнов І.Р. Чат-бот як новий засіб інтерактивного навчання с.149-150 (дата звернення: 04.05.2021)

27. Вікіпедія – вільна енциклопедія. URL: <https://ru.wikipedia.org/wiki/MTPROTO>  
(дата звернення: 04.05.2021)



Декларація щодо унікальності текстів роботи  
та невикористання матеріалів інших авторів без посилань

Щербак Владислав Володимирович

Прізвище, ім'я, по батькові

Інформаційних і прикладних технологій

Факультет

122 «Комп'ютерні науки»

Шифр і назва спеціальності

«Сучасні інформаційні технології та програмування»

Освітня програма

**ДЕКЛАРАЦІЯ**

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «Розробка телеграм-боту для формування розкладу занять» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

\_\_\_\_\_  
дата

\_\_\_\_\_  
підпис