**Комунальний заклад освіти**

**«Дніпровський ліцей інформаційних технологій**

**при Дніпровському національному університеті**

**імені Олеся Гончара»**

**Випускна робота**

**на тему:**

**Сбір даних засобами «Telegram» та їх візуалізація, засобами С#**

**Виконавець:**

**ліцеїст 11-Б-II класу**

**Волошин Микита**

**Керівники роботи:**

**Лавренюк І.В. \_\_\_\_\_\_\_\_\_\_\_\_\_**

Зміст

[ВСТУП 3](#_Toc90981574)

[Тема 3](#_Toc90981575)

[Мета 3](#_Toc90981576)

[Задачі 3](#_Toc90981577)

[РОЗДІЛ І ТЕОРІЯ 4](#_Toc90981578)

[Обґрунтування використання бібліотек та платформ 4](#_Toc90981579)

[C# 4](#_Toc90981580)

[Windows Forms (.net Framework 4.8) 4](#_Toc90981581)

[Telegram 4](#_Toc90981582)

[ASP .net Core Web-API 4](#_Toc90981583)

[БД (MSSQL) 4](#_Toc90981584)

[Entity Framework 6 (для Forms) та пакет Entity Framework Core v8 (для WebAPI) 5](#_Toc90981585)

[Live Chart 5](#_Toc90981586)

[Алгоритми 5](#_Toc90981587)

[БД 5](#_Toc90981588)

[Telegram бот 6](#_Toc90981589)

[Клієнт для перегляда 6](#_Toc90981590)

[Розділ ІІ Можливості роботи 8](#_Toc90981591)

[Структурна схема 8](#_Toc90981592)

[Зовнішній вигляд та приклади взаємодії 8](#_Toc90981593)

[Застосування 11](#_Toc90981594)

[Розділ ІІІ Висновки та специфікації 13](#_Toc90981595)

[Системні вимоги та комплектація 13](#_Toc90981596)

[Telegram бот 13](#_Toc90981597)

[Клієнт для перегляду 13](#_Toc90981598)

[Використані засоби 14](#_Toc90981599)

[Висновки 15](#_Toc90981600)

[Джерела 16](#_Toc90981601)

ВСТУП

## Тема

Збір та аналіз даних – це задача що, на мою думку, одна з основних для усієї науки, яку сьогодні звуть інформатикою. Моя робота уявляє собою повний цикл обробки інформації з моменту вводу даних респондентами до їх візуалізації.

Сам проект уявляє собою доданок для аналізу показників руху міського транспорту. За допомогою бота у Telegram водій буду вказувати дані щодо швидкості свого пересування та динаміки кількості пасажирів у своєму транспорті.

А менеджер за допомогою доданка Windows Forms зможе проаналізувавши інформацію у вигляді графіків оптимізувати маршрути та кількість транспортних одиниць на ньому цим, поліпшивши якість послуг.

## Мета

Створити платформи для збору і візуалізації даних руху та завантаженості міського транспорту. Дослідження Telegram, як засобу для сбіру даних та створення ботів на базі цього месенджеру. Створити механізм передачі зібраних даних та доданок для їх візуалізації

## Задачі

* Створення платформи для збору і візуалізації даних руху та завантаженості міського транспорту. Дослідження Telegram, як засобу для сбіру даних та створення ботів на базі цього месенджеру. Проаналізувати засоби передачі даних, та навчити застосувати.
* інтернет-доданок на платформі .net Core 6, із архітектурою. WebApi, для взаємодії із Telegram, а через нього з користувачем;
* створення та публікація база даних на базі MSSQL та подальше її використання у клієнтах проекту
* створення клієнту для перегляду і редагування даних на базі Windows Forms;
* дослідження бібліотек для створення інфографік та після вибору, вдале використовування її у проекті;
* поєднання у суцільний проект.

# РОЗДІЛ І ТЕОРІЯ

## Обґрунтування використання бібліотек та платформ

### C#

Сучасна мова програмування, одним з основних плюсів якої є її багатогранність якою, на мою думку, не може сперечатися будь-яка інша мова, що є дуже важливим пунктом для мого проекту, адже синхронізувати програми на різних мовах набагато складніше.

### Windows Forms (.net Framework 4.8)

Ця платформа досить легка для освоєння, а також надає змогу створювати інтерфейс лише засобами С#, до того ж ця платформа, хоч вже і не підтримується але за час існування зібрала великий арсенал бібліотек, що без сумніву, мені допоможу у роботі із візуалізацією даних. До того ж ця платформа вивчається у ЛІТі, а тому я зможу отримати допомогу у разі виникнення проблем.

### Telegram

Зручний та сучасний месенджер, що відповідає стандартам безпеки та просто є зручним. Це робить цю платформу дуже привабливою, адже користувачу не потрібно навіть завантажувати додаток, або використовувати сайт – усе буде у його улюбленому месенджері.

Витікає із попереднього пункту, ця бібліотека хоч розроблена не офіційними представниками Telegram, але сама її рекомендують використовувати розробники Telegram у своєму керівництві по створенню ботів, а отже сама її слід використовувати для створення найгарнішого доданку.

### ASP .net Core Web-API

Також випливає із попереднього пункту, сама цей типи проекту рекомендують використовувати розробники бібліотеки, а також додали приклад використання своєї бібліотеку саме в такому варіанті проекту. Також у своїй курсовій роботі я використовував консольний доданок як основу, що теж є прикладом у цій бібліотеці, але, мою думку, програмний продукт, а саме цим має бути кваліфікаційна робота, має відповідати світовим стандартам якості, а отже програмне забезпечення, що обробляє інформацію та знаходиться на сервері має бути Web-API.

БД (MSSQL)

База даних у цьому проекті життєво необхідна, бо дані потрібно кудись збирати, а потім з чогось брати, це можна було б реалізувати за допомогою запитів до бота, але тоді, на мою думку, на цю частину буде надзвичайно велике навантаження, що недопустимо для стабільної системи. MSSQL же був обраний як мова, що розробляється Microsoft а отже проблем із сумісністю з‘явиться на порядки менше, якщо узагалі з‘являться.

### Entity Framework 6 (для Forms) та пакет Entity Framework Core v8 (для WebAPI)

Після остаточного рішення щодо використання повноцінної бази даних, стало питання як із нею взаємодіяти тут я, не довгого вагаючись, я обрав Entity Framework, як саму стабільну швидку та офіційну бібліотеку для роботи.

### Live Chart

Найгарніша бібліотека, на мою думку, для створення інфографік із багатим простором для творчості та різноманіттям функцій.

## Алгоритми

### БД

Створення БД за допомогою Entity Framework- надзвичайно цікавий процес. По-перше, існує декілька підходів для використання БД.

* Code-First підхід котрий полягає у тому, що ми створюємо таблицю БД із використанням принципів ООП у С# тобто, ми створюємо клас, наповнення якого і будуть полями таблиці, а набор класів і буде свого чину таблицею. Цей підхід я застосував при створенні бази даних у рамках проекту Web-API.
* Database-First підхід котрий полягає у створенні ООП моделі після підключення до БД, свого роду реверс-інжиніринг. Цей підхід був використаний мною під час розробки клієнта для перегляду даних, бо база даних має бути однією для усіх проектів, а цей підхід автоматично побудує моделі для вдалого підключення.
* Існують також інші підходи для створення але вони уявляють собою, або комбінацію попередніх, або використання них же але наприклад із інтерфейсом як у MS Access. Їх я також частково використовував, але не вважаю досить важливими для згадування, адже суттєвого впливу на проект вони не мали.

Існують декілька методів завантаження даних із бази даних допомогою Entity Framework.

* Explicit Loading полягає у явному завантаженні у конкретного об‘єкта властивості чи колекції.
* Eager Loading полягає у використанні метода .Include, у я кому ми за допомогою лямбда виразів вказуємо яку властивість або колекцію маємо завантажити для **колекції об‘єктів**.
* Lazy Loading зараз не дуже часто використаний підхід через те, що він полягає у завантаженні тільки того, що ми використовуємо під час використанні програми, як можна зрозуміти такий вид завантаження не дуже гарно впливає на швидкість роботи доданку.

Усі ці методи гарні по своєму але треба розуміти, що серед них нема жодного поганого чи доброго і це я довів власноруч у своїй кваліфікаційній роботі де я використовую усі три вида. Усі це методи мають різну спеціалізацію якщо ми маємо завантажити малі об‘єми даних, то ми використовуємо Explicit loading, якщо колекцію даних, то Eager Loading, щодо Lazy Loading хоч він і може бути замінений комбінацією Eager і Explicit Loading, але у десктопних програмах де не так важливо використовувати мінімум оперативної пам‘яті і навантаження процесора цей підхід має свою нішу, бо це суттєво спрощує розробку.

### Telegram бот

Основна складність написанням алгоритмів для бота є у тому, що в нього відсутня оперативна пам’ять. Звісно не буквально, але кожне нове повідомлення розглядається як перше і це значно ускладнює не стільки розробку корисної логіки, скільки творення життєздатної системи, де користувач не зможе випадково користуватися програмою не так я було продумано, а злодій не зможе видалити чи переглянути дані, що можуть бути комерційною таємницею.

### Клієнт для перегляда

Щодо клієнту основна складність алгоритмів звісно полягає у готуванні даних для їх представлення.

Тут я частково застосовували як LINQ так і звичайні цикл для більш специфічних операцій, навіть, я б сказав, що LINQ я використовував тільки коли лез нього були б значні падіння у швидкодії, або операція легко описується LINQ, циклами ж набагато складніше. В інших випадках і використовував сдебільше цикли, бо вважаю їх більш читабельними, а також, не враховуючи специфічні операції, цикли працюють швидше.

Також слід зазначити, що уся статистика показується лише за останні 30 днів, на мою думку, це оптимальний час для того щоб була повна картина аналізу, а об‘єми даних не завеликі для обри бік на звичайному персональному комп‘ютері, до того ж якщо використовувати більші періоди місця на формі не достатньо для комфортного перегляду.

Також були створені форми для зміни та додавання маршрутів або водіїв нічого складного в них нема, але завдяки тому що бот *не* має оперативної пам‘яті усі зміни що робить адміністратор, відображаються майже моментально.

Розділ ІІ Можливості роботи

## Структурна схема

Telegram бот

Клієнт для перегляду даних

## Зовнішній вигляд та приклади взаємодії

Telegram бот

Зображення, що містить текст, монітор, комп’ютер, електроніка

Автоматично згенерований описБудь-яка неправильна команда або /usage викликає допоміжне повідомлення

Рисунок 1Приклад /usage

Рисунок 1Приклад /usage

Команда /routes викликає меню із усіма маршрутам у яких показана початкова і кінцева станція. Клік по кнопці видає Id маршруту та список усіх станцій

Зображення, що містить текст, монітор, комп’ютер, знімок екрана

Автоматично згенерований опис

Рисунок 2 Приклад /routes

Команда /start – єдина команда яка приймає аргументи, а саме Id маршруту який треба почати та стартову кількість пасажирів.

Зображення, що містить текст, монітор, знімок екрана, комп’ютер

Автоматично згенерований опис

Рисунок 3 Приклад /start

/continue створює клавіатуру із швидким вводом чисел для позначення зміни кількості пасажирів.

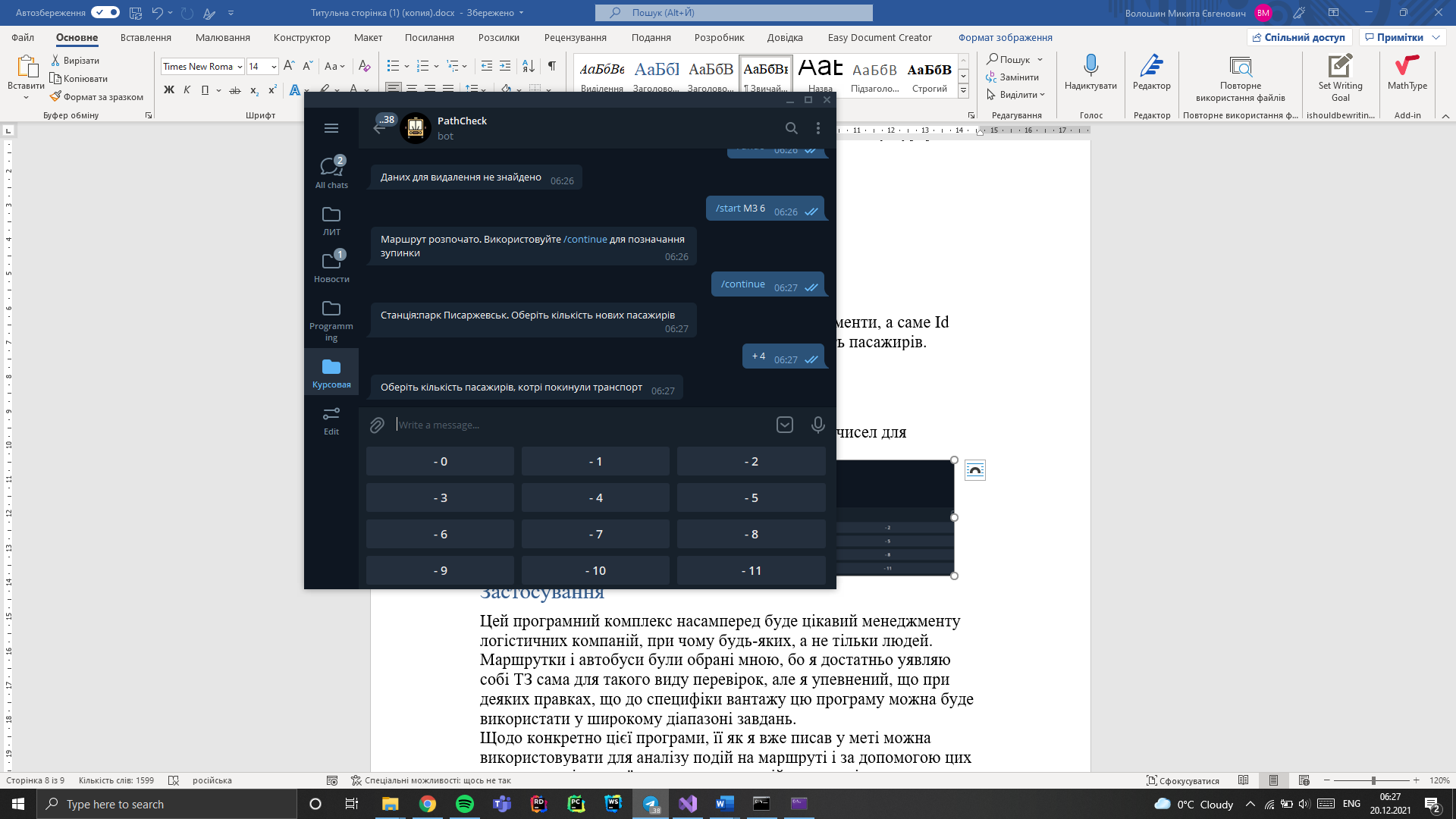


Рисунок 4 Приклад /continue

Команда /undo видаляє останні введені дані

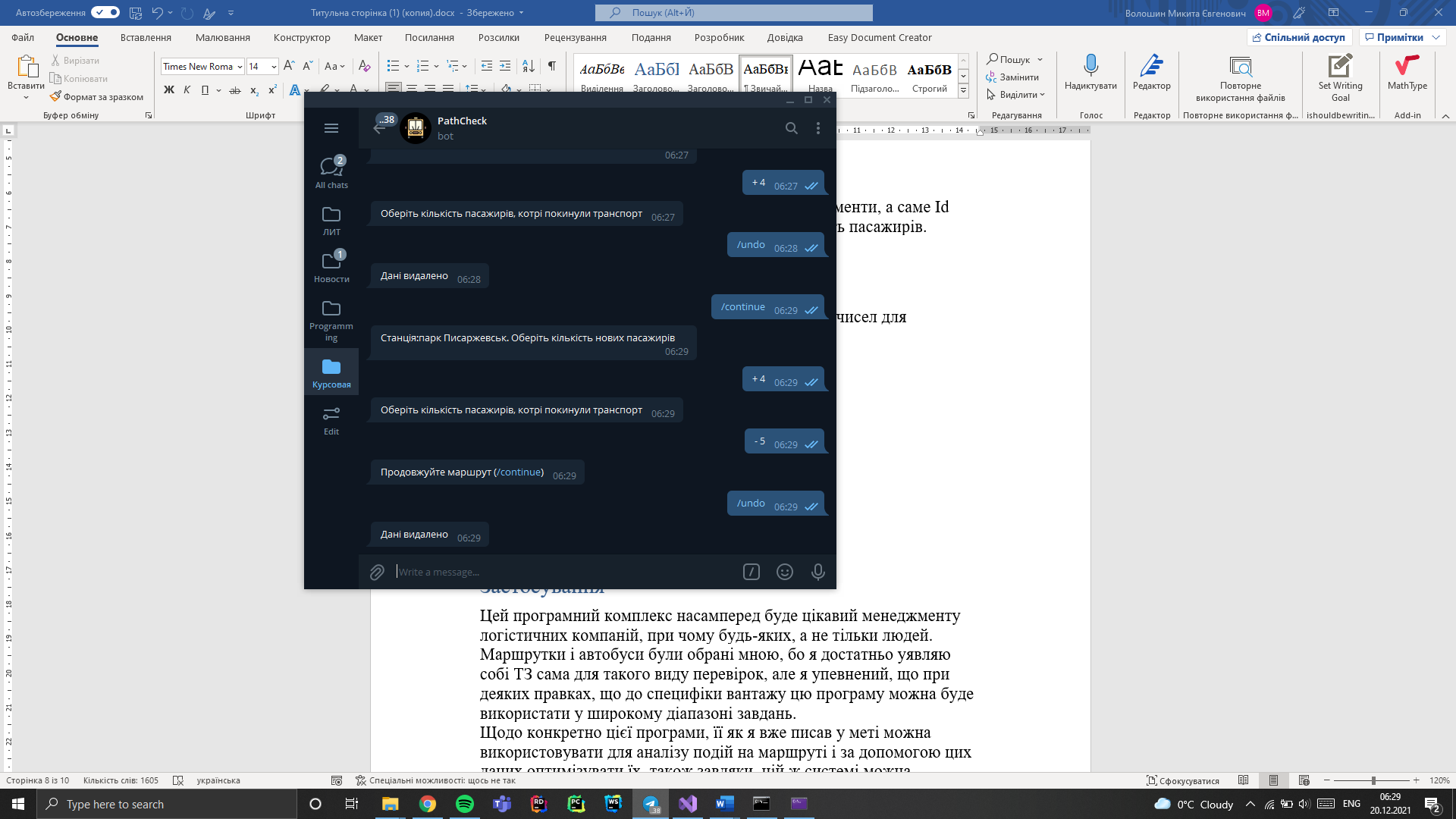


Рисунок 5 Приклад /undo

/end видаляє початий але незавершений маршрут та всі дані по ньому.

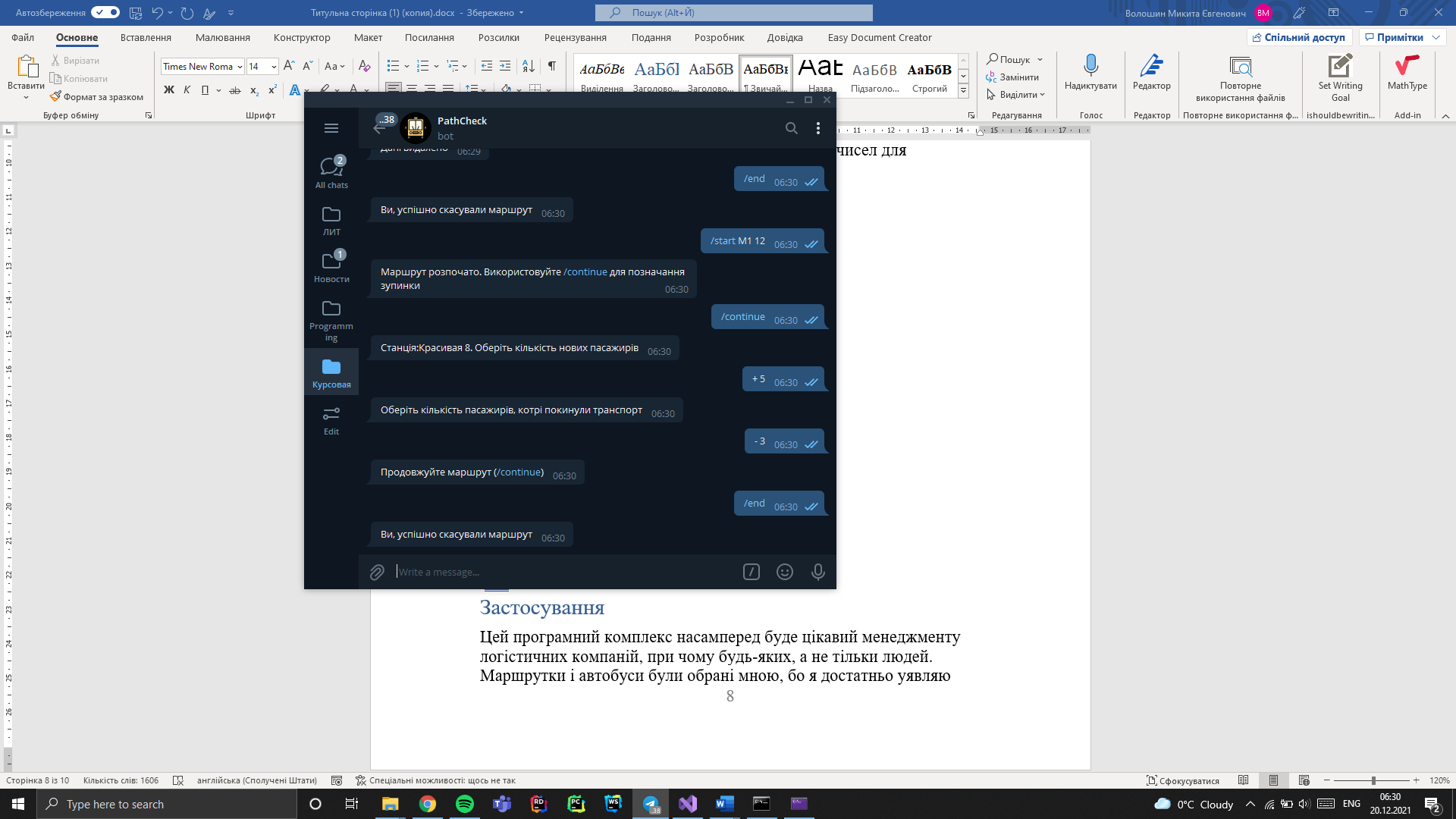


Рисунок 6 Приклад /end

Клієнт для перегляду даних

Після вітальної сторінку ми попадаємо у головне меню. Де можемо обрати один з чотирьох пунктів

* Перший показує статистику щодо маршрутів, спочатку загальну діаграму кількості. Після цього можна обрати тип маршрута та створити графік із середніми значеннями на кожній ділянці маршрута, у наступному меню можна обрати детальний аналіз вибраного маршрута

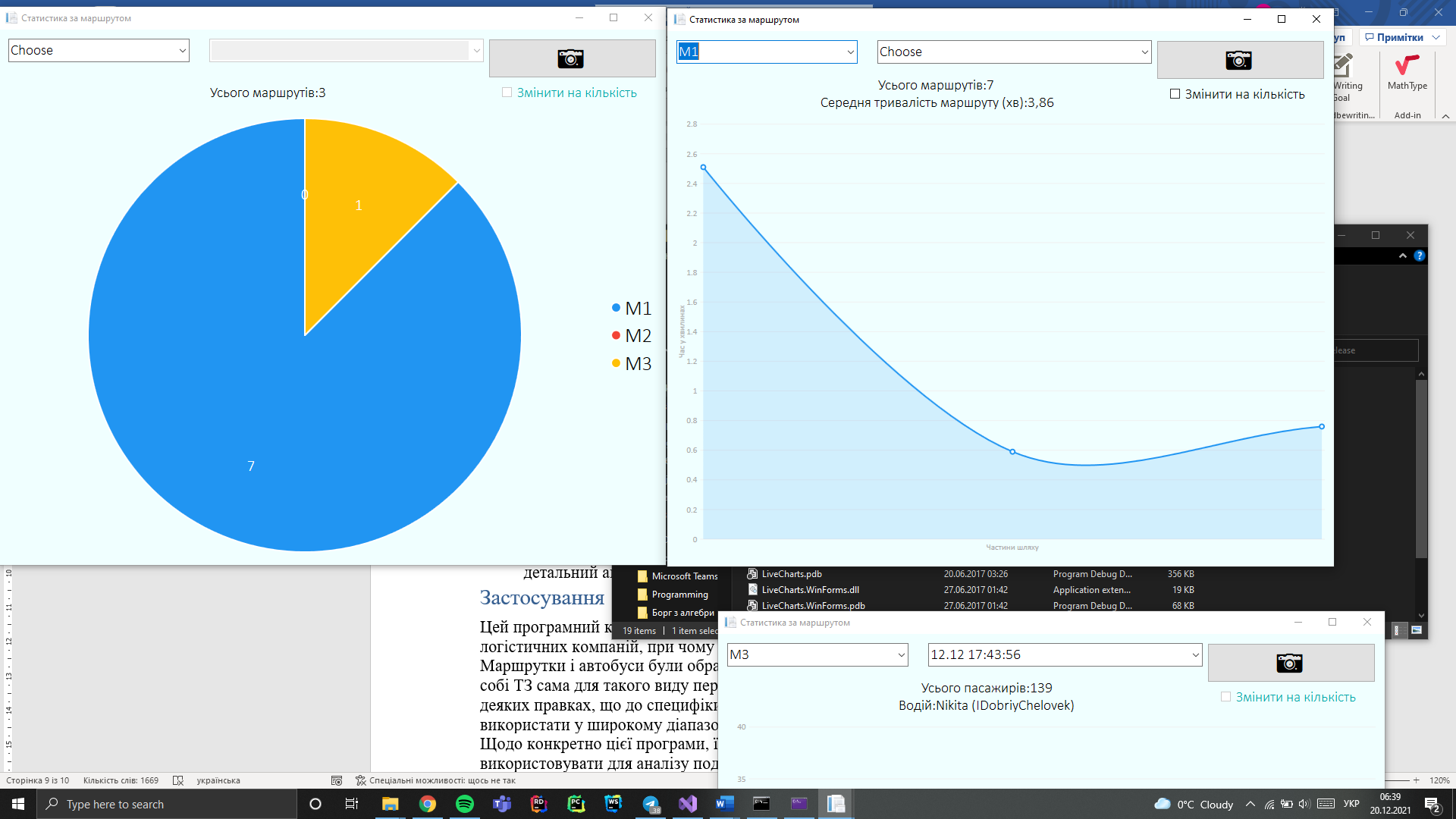


Рисунок 7 Приклад перших двух інфографік

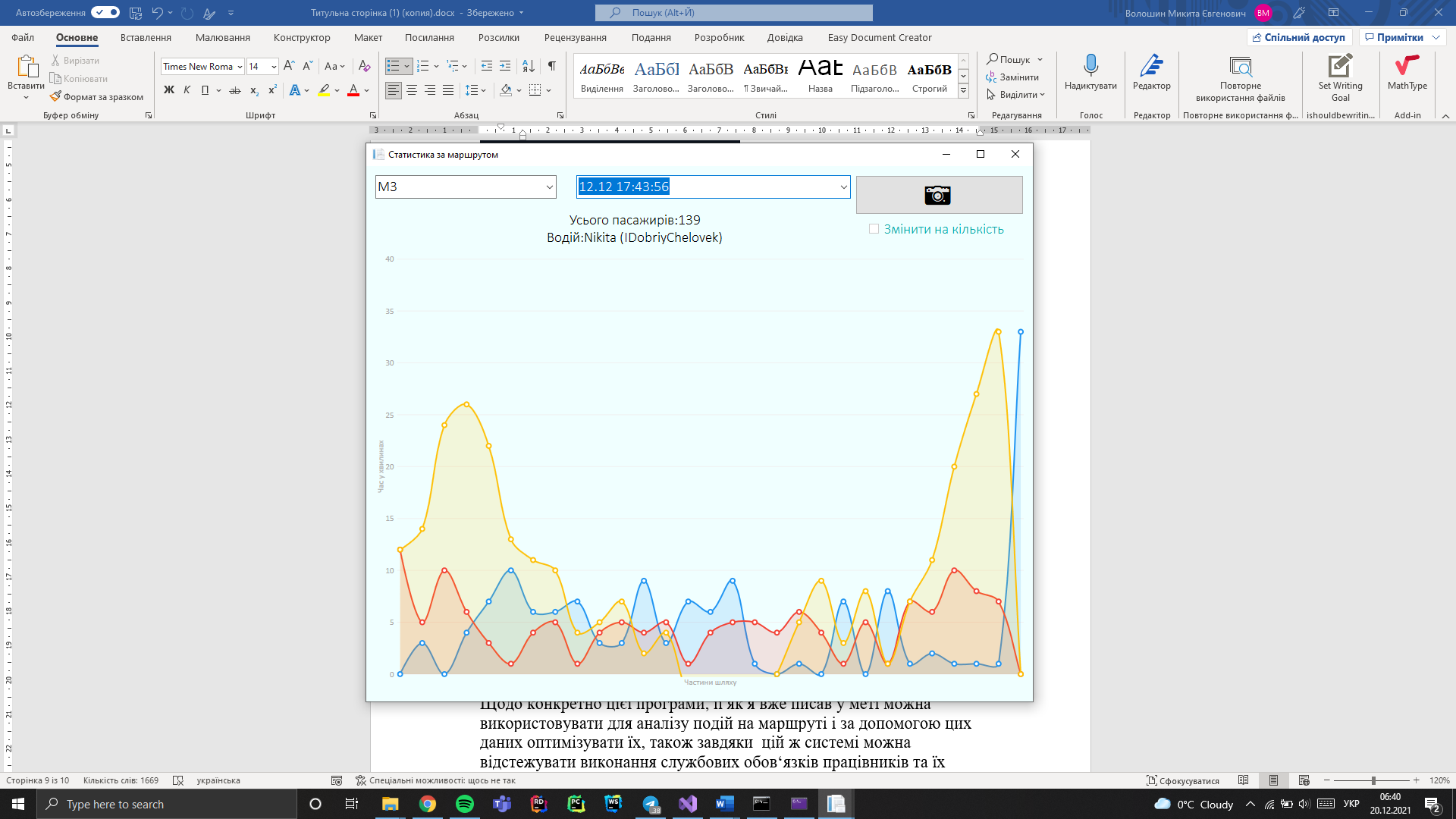


Рисунок 8 Приклад третьої інфографіки

* У другому пункті після обрання водія ми побачимо діаграму та графік із статистикою його виїздів за останні 30 днів

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 9 Інфорграфіки з другого вікна

* Третій пункт дозволяє створювати та видаляти маршрути. Усі зміни відбуваються тільки після натискання кнопки «Зберегти».

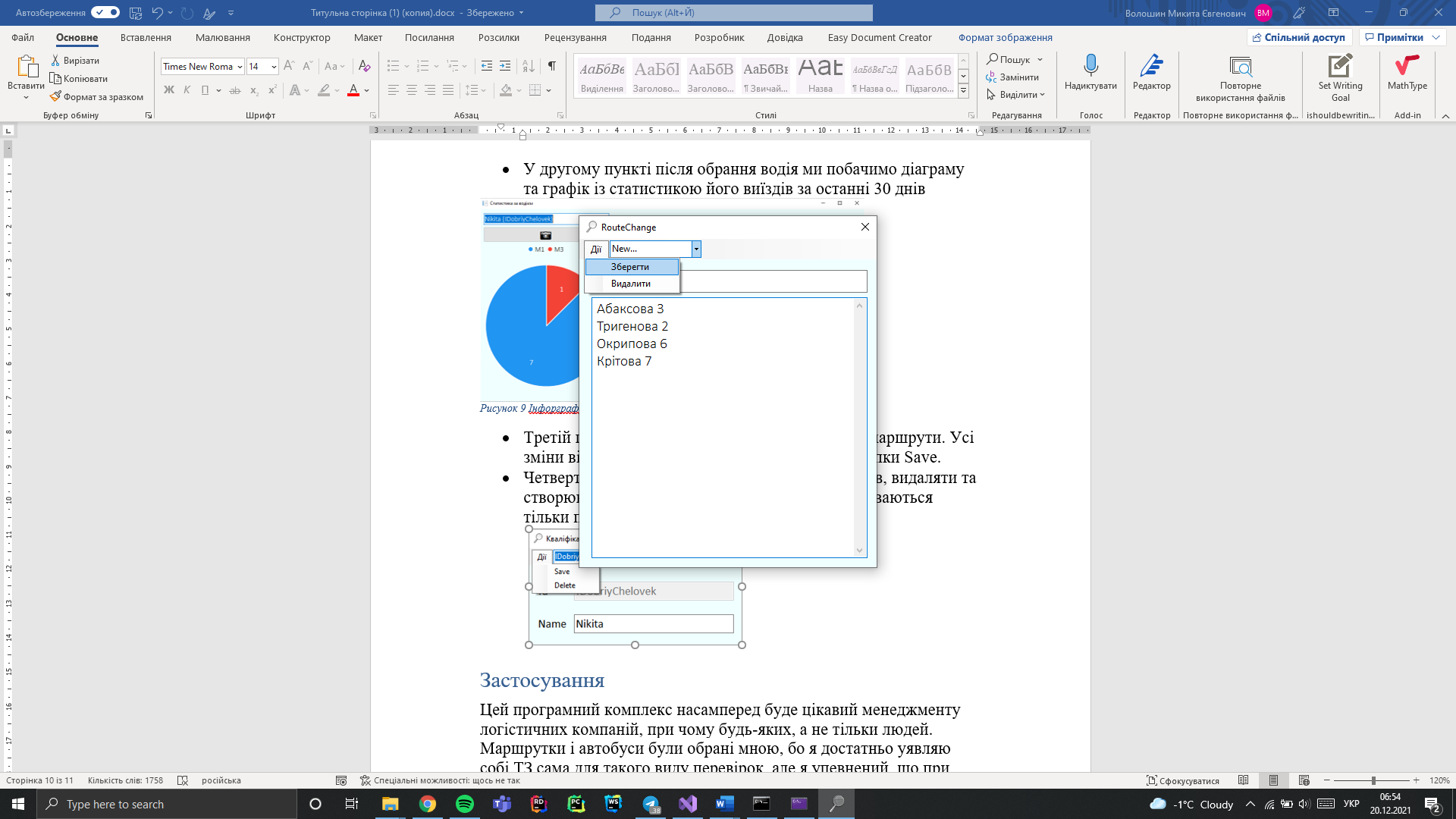


Рисунок 10 Редактор маршрутів

* Четвертий дозволяє редагувати ім’я користувачів, видаляти та створювати нових користувачів. Усі зміни відбуваються тільки після натискання кнопки кнопки «Зберегти».

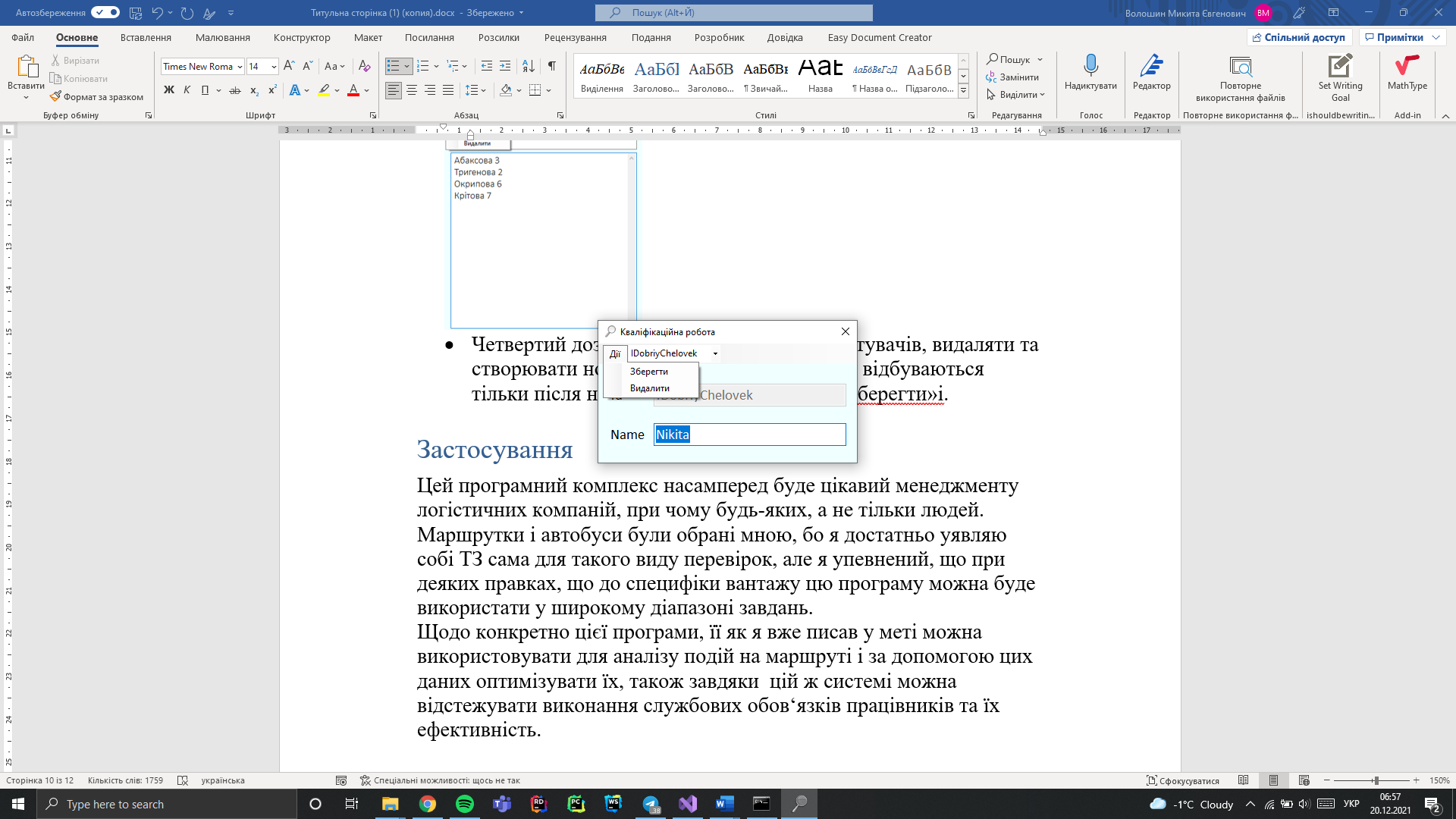


Рисунок 11 Редактор водіїв

## Застосування

Цей програмний комплекс насамперед буде цікавий менеджменту логістичних компаній, при чому будь-яких, а не тільки людей. Маршрутки і автобуси були обрані мною, бо я достатньо уявляю собі ТЗ сама для такого виду перевірок, але я упевнений, що при деяких правках, що до специфіки вантажу цю програму можна буде використати у широкому діапазоні завдань.

Щодо конкретно цієї програми, її як я вже писав у меті можна використовувати для аналізу подій на маршруті і за допомогою цих даних оптимізувати їх, також завдяки цій ж системі можна відстежувати виконання службових обов‘язків працівників та їх ефективність.

Розділ ІІІ Висновки та специфікації

## Системні вимоги та комплектація

### Telegram бот

По-перше слід зазначити, що запустити бота можна двома шляхами. Перший на localhost для цього потрібна програма хостер ngrok, а у ній увести *“ngrok http -host-header=localhost 31623* або використати ngrok.bat, що введе автоматично усю необхідну інормацію*.* Після цього у файлі appsettings потрібно змінити існуючу адресу на ту що відобразилася у програмі.

Другий ж передбачає публікування бота на сервер, але перед цим слід перевірити, що сайт підтримує https, а хостинг підтримує .net 6, після цього потрібно вказати доменне ім‘я у файлі appsettings і опублікувати бота із використанням спеціальної функції у visual studio.

* Підключення до інтернету
* Вільний простір 200MB

**Комплектація**:

* Exe файл із супутніми DLL
* Appsettings.json
* ngrok.exe
* ngrok.bat
* web.config
* TgBot.pdb
* TgBot.XML

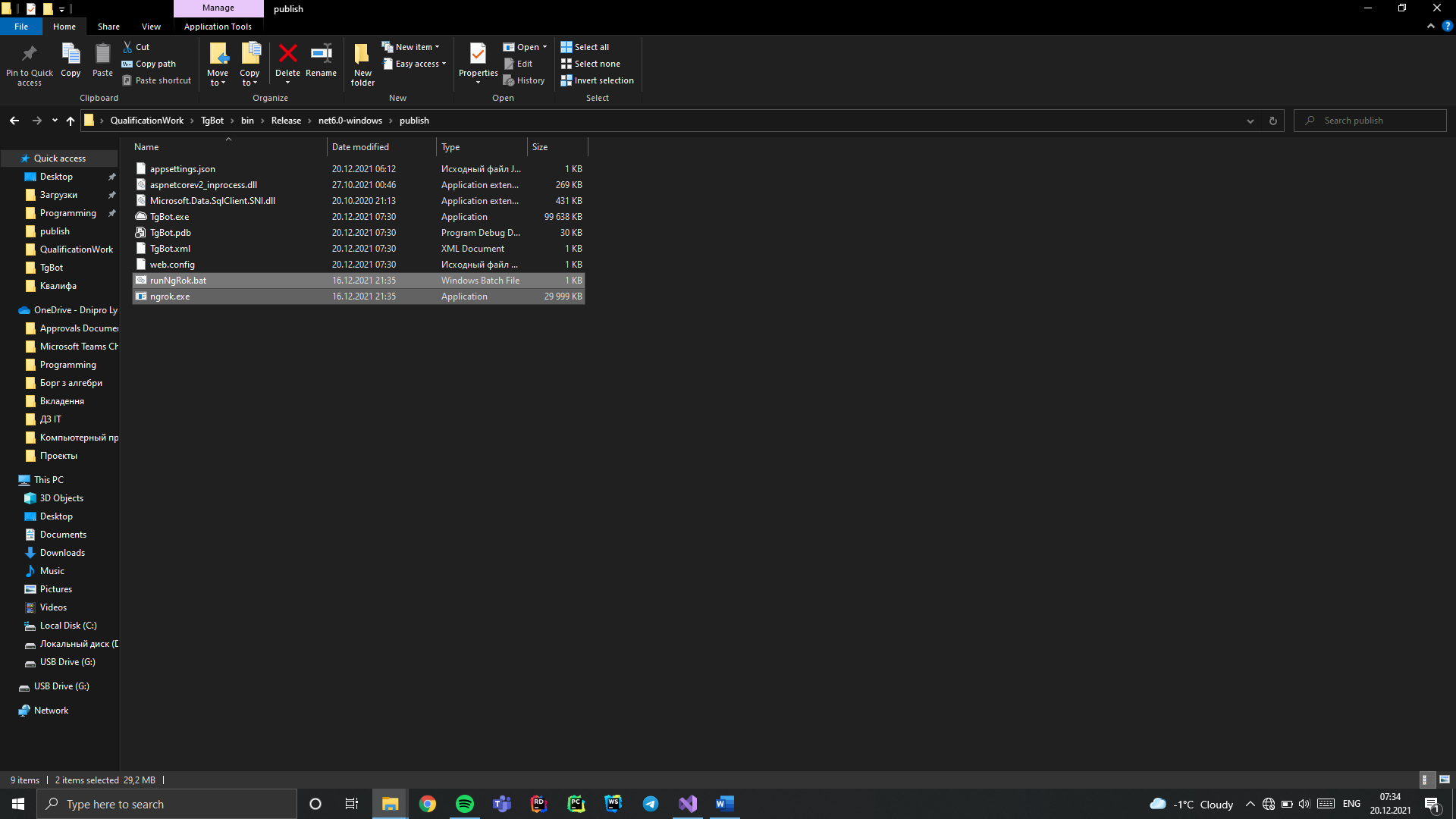


Рисунок 12 Повиний список

### Клієнт для перегляду

Вільний простір: 200MB

**Комплектація:**

* Setup.exe
* Папка Application Files/DataClient\_1\_2\_1\_3 із необхідними для інсталяції файлами

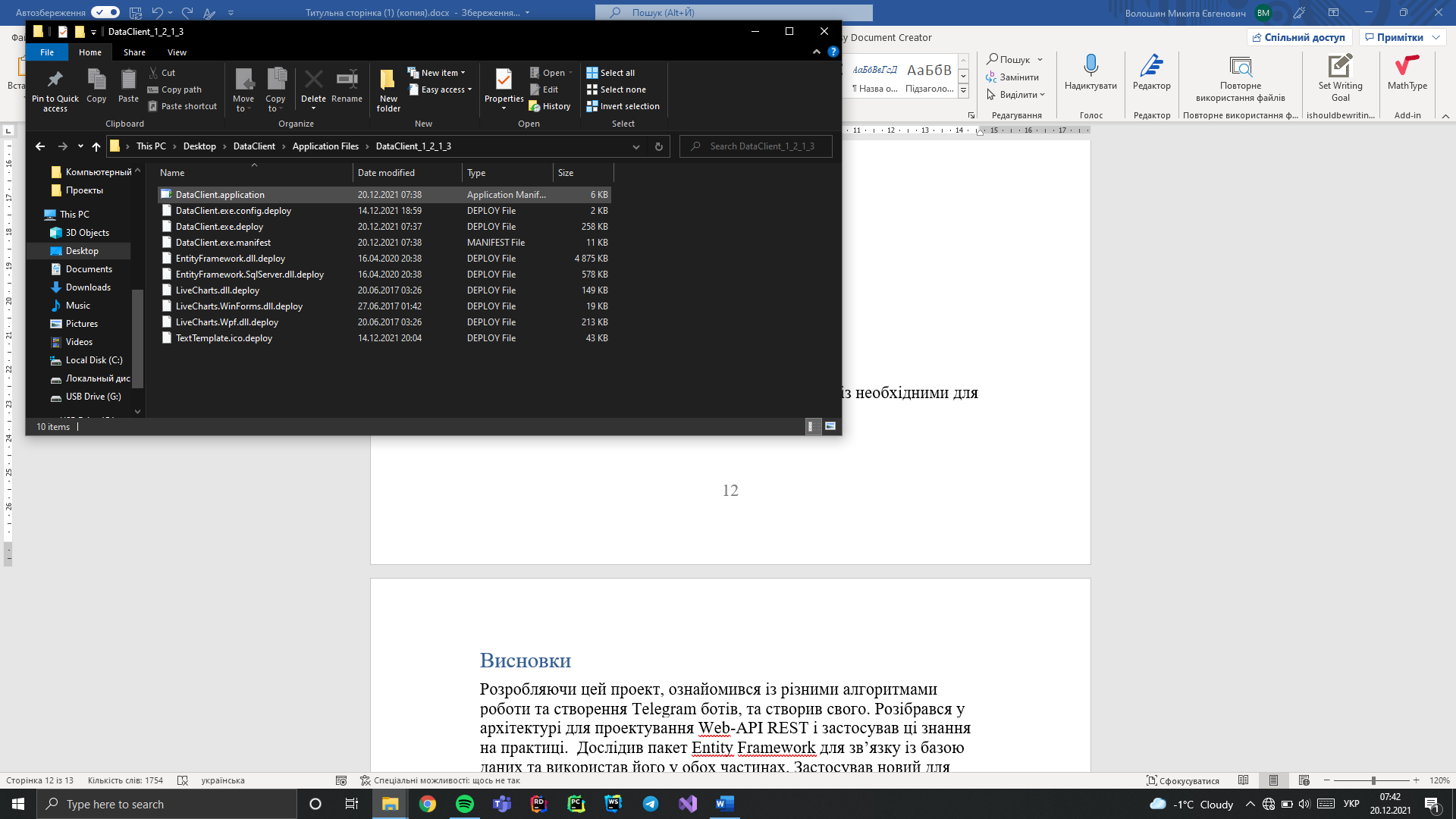


Рисунок 13 Необхідні для інсталяції клієнту даних файли

## Використані засоби

* Visual Studio 2021
* Rider
* Access
* Azure

Висновки

Розробляючи цей проект, ознайомився із різними алгоритмами роботи та створення Telegram ботів, та створив свого. Розібрався у архітектурі для проектування Web-API REST і застосував ці знання на практиці. Дослідив пакет Entity Framework для зв’язку із базою даних та використав його у обох частинах. Застосував новий для себе елемент для створення інтерактивних графіків у формі.

Джерела

1. <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
2. <https://www.youtube.com/watch?v=OTKJfGbgfxM&ab_channel=AtlassDev>
3. <https://www.youtube.com/watch?v=yubASf8mNZw&ab_channel=Skillbox%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5>
4. <https://habr.com/ru/company/alfa/blog/343846/>
5. <https://ru.stackoverflow.com/questions/784321/telegram-bot-c-%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3>
6. <https://metanit.com/sharp/entityframeworkcore/>
7. <https://metanit.com/sharp/entityframework/>
8. <https://lvcharts.net/App/examples/v1/WinForms/start>
9. <https://github.com/TelegramBots/Telegram.Bot>
10. <https://core.telegram.org/bots>
11. <https://telegrambots.github.io/book/>
12. <https://github.com/TelegramBots/telegram.bot.examples>