

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка чат-боту Telegram для замовлення таксі

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Гричка Владислава
Олеговича

підпис студента

Науковий керівник
науковий ступінь
вчене звання

Рассамакін Володимир
Якович

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

підпис керівника

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Зав. кафедри інженерії
програмного забезпечення
кібербезпеки
Криворучко О.В.
"20" жовтня 2020 р.

на випускн кваліфікаційну роботу студентів

Гричку Владиславу Олеговичу

(прізвище, ім'я, по батькові)

- Тема випускної кваліфікаційної роботи Розробка чат боту TELEGRAM для замовлення таксі

Затверджена наказом ректора від "30" жовтня 2020 р. № 3225

2. Строк здачі студентом закінченої роботи

3. Цільова установка та вихідні дані до роботи

Мета роботи та описати та розробити чат бот таксі для платформи телеграм

Об'єкт дослідження інформаційні технології автосервісу

Предмет дослідження чат бот Таксі

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ОПИС ТА ОГЛЯД ЧАТ БОТІВ

1.1 Месенджери

1.2 Чат боти

1.3 Огляд аналогів чат ботів

1.4 Висновки до першого розділу

РОЗДІЛ 2 Огляд інструменталу використаного при розробці

2.1 HTTP-інтерфейс Bot API

2.2 База даних Sql Lite

2.3 Платформа розробки . Net Core

2.4 Середовище розробки Visual Studio

2.5 Висновки до другого розділу

РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ

3.1 Розробка інтерфейсу адміністратора

3.2 Акробатія розробки

3.3 Інтерфейс користувача

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2020	21.09.2020
2.	<i>Вступ та перелік літературних джерел</i>	14.12.2020	14.12.2020
3.	<i>Розділ 1. «ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ СТВОРЕННЯ ТА АНАЛІЗУ ЧАТУ»</i>	19.02.2021	19.02.2021
4.	<i>Розділ 2. «ОСНОВНІ ЕТАПИ ПРОЕКТУВАННЯ ТА СТВОРЕННЯ МЕРЕЖЕВОГО ЧАТУ»</i>	05.03.2021	05.03.2021
5.	<i>Розділ 3. «РЕАЛІЗАЦІЯ ПРОЕКТУ “HOUSE CHAT”»</i>	10.04.2021	10.04.2021
6.	<i>Висновки</i>	24.04.2021	24.04.2021
7.	<i>Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)</i>	14.05.2021	14.05.2021
8.	<i>Підготовка автореферату та презентації доповіді</i>	17.05.2021	17.05.2021
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	18.05.2021 - 21.05.2021	
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.06.2021	
11.	<i>Здача прошого випускної кваліфікаційної роботи на кафедрі</i>	02.06.2021	
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «20» жовтня 2020 р.

8. Науковий керівник випускної кваліфікаційної роботи

Рассамакін В. Я.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Цензура М. О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Гричок В. О.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист _____ Рассамакін В. Я.
(ПІБ, підпис, дата)

Випускна кваліфікаційна робота студента Гричок В. О.
(прізвище, ініціали)
може бути допущена до захисту екзаменаційній комісії.

« _____ » 2021 p.

АНОТАЦІЯ

Дипломна робота на тему «Розробка чат боту Telegram для замовлення таксі» містить 39 сторінку та 40 рисунків. Перелік списку використаних джерел налічує 6 посилань.

Метою роботи є описати та розробити чат бот таксі для платформи Телеграм.

Під **об'єктом дослідження** випускної кваліфікаційної роботи розуміємо інформаційні технології автосервісу.

Предметом дослідження випускної кваліфікаційної роботи є чат бот Таксі.

В першому розділі розглянуті історичні та технологічні аспекти, сервісів обслуговування населення що пов'язані с таксі, існуючі новітні технології з використанням чат ботів та месенджерів які найбільш актуальні на сьогоднішній день для розробки програмних додатків і програм для таксі.

У другому розділі було описано інструментал який був використаний при розробці. Для зв'язку із сервером було обрано Bot API який представляє собою HTTP-інтерфейс для роботи з ботами в Telegram. Далі було обрано платформу розробки .NET Core вона працює на різних платформах і є швидкою, гнучкою і сучасною платформою. Підключено це все до внутріпроцесної бібліотеки SQLite, яка реалізує автономний, безсерверний, з нульовою конфігурацією, транзакційний движок бази даних SQL. Та визначилися з середовищем розробки це Visual Studio Code яка є легким , але потужним редактором вихідного коду, який сумісний з Windows, MacOS і Linux.

В третьому розділі була описана робота над створенням додатку, описаний інтерфейс адміністратора та користувача, також представлений код основних функцій програми.

ANOTATION

The diploma of the robot on the topic "**Development of the chat bot Telegram for the change of taxis**" revenge 39 sides and 40 drawings. Changed to the list of vicarious dzherel nalichu 6 posilan.

Meta robots € describe how to develop a chat bot taxi for Telegram platforms.

Pid about the **advancement of high quality robots** and information technology for car service.

The subject of advanced quality robots is chat bot Taxi.

For the first time, the historical and technological aspects, services for servicing the population, are connected with taxis, and new technologies for the current chatbots and messengers for the most current

In another rozdili bulo described instrumental like buv vikoristany when rozrobsi. For the connection from the server, the Bot API was used, which is an HTTP interface for robots with bots in Telegram. Now, the development platform .NET Core has been successfully developed on other platforms and is fast, nasty, and fun. Everything is connected to the internal process library SQLite, as it implements an autonomous, serverless, zero-configuration, transactional SQL database engine. That was the middle of the distribution of the Visual Studio Code like € an easy, tedious editor of the output code, like the one for Windows, MacOS and Linux.

In the third section, the robot is described above the doors of the add-on, descriptions of the interface of the administrator and the keystroke, as well as representations of the code of the main functions of the programs.

ЗМІСТ

ВСТУП.....	2
РОЗДІЛ 1 ОПИС ТА ОГЛЯД ЧАТ БОТІВ	4
1.1 Месенджери.....	4
1.2 Чат боти	6
1.3 Огляд аналогів чат ботів.....	8
1.4 Висновки до першого розділу	12
РОЗДІЛ 2 Огляд інструменталу використаного при розробці	13
2.1 HTTP-інтерфейс Bot API	13
2.2 База даних Sql Lite	14
2.3 Платформа розробки . Net Core.....	15
2.4 Середовище розробки Visual Studio	17
2.5 Висновки до другого розділу	18
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ	19
3.1 Розробка інтерфейса адміністратора	19
3.2 Акробатія розробки	26
3.3 Інтерфейс користувача	37
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39

					<i>КНТЕУ 121 06-07.БР</i>		
Зм.	Аркуш	№ докум	Підпис	Дата			
Зав. каф.		Криворучко О.В.			<i>Розробка чат боту TELEGRAM для замовлення таксі</i> <i>Зміст</i>		
Керівник		Рассамакін В.Я.					
Гарант		Цензура М.О.					
Розроб.		Гричок В.О.					
					<i>Зміст</i> Факультет інформаційних технологій, 4 курс, 6 група		
						Стадія	Аркуш
						Зміст	2
							39

ВСТУП

У процесі вивчення цієї предметної області я виявив, що користувачам потрібен чат бот Таксі, який міг би доставити їх куди потрібно. Останнім часом зростає популярність месенджерів, вони стали не тільки засобами для спілкування та ще й засобами для отримання інформації. Велику роль у цьому відіграли чат-боти. На даний час існує багато їх варіацій – від ботів для знаходження музики до ботів для замовлення таксі. Таким важливим інструментом цікавляться більшість населення. І хоча боти уже не малий час посилено використовуються закордоном, в Україні вони не так популярні, а отже є прибутковою нішею для розробників, тому було вирішено розробити чат бота таксі на базі месенджеру Telegram.

У роботі описуються загальні принципи створення бота для месенджера, його можлива архітектура, а також розглядається можливість введення функціоналу самого таксі. Аналізуючи цю тему, можна зробити висновок, що чат боти займають важливе місце в житті людей, тобто є актуальною темою в суспільстві.

Актуальність теми: полягає у необхідності розробки чат бота Таксі.

Мета дослідження: описати та розробити чат бот таксі для платформи Телеграм.

Об'єкт дослідження: інформаційні технології автосервісу.

Предмет дослідження: чат бот Таксі.

Завдання випускної кваліфікаційної роботи: визначити середовище розробки та програмних засобів, апробація розробки інструкції користувача.

Методи дослідження: аналіз і узагальнення спеціальної літератури та публікацій.

Аналіз предметної області: служби замовлення таксі.

					КНТЕУ 121 06-07.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка чат боту TELEGRAM для замовлення таксі	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.				В	3	39
Керівник		Рассамакін В.Я.				Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розроб.		Гричок В.О.						
					Вступ			

РОЗДІЛ 1

ІСТОРИЧНІ ТА ТЕХНОЛОГОЧНІ АСПЕКТИ

1.1 Месенджери

Свій розвиток інтернет-сервер для спілкування почали з чатів, потім месенджери, потім соціальні мережі, але з недавніх пір месенджери знову очолили список найбільш перспективних сервісів. Месенджери використовуються переважно дорослими, так на частку осіб до 18 років припадає 7,7% від загальної кількості аудиторії.

Всі сучасні месенджери мають безліч схожих функцій, таких як відправка повідомлень, медіа файлів, аудіо та відео спілкування.

В даний час існує велика різноманітність месенджерів, серед яких варто відзначити деякі з них, а також їх переваги та недоліки. На особливу увагу заслуговує месенджер Telegram. Telegram - це кроссплатформне додаток, розроблене на мові програмування C++, що дозволяє обмінюватися повідомленнями і файлами більшості форматів. Месенджер використовує спеціально розроблену серверну частину з закритим кодом, які працюють на серверах Німеччини і США. Telegram має ряд переваг:

приватність - все чати зашифровуються, а повідомлення знищуються через зазначений час;

- швидкість - швидкість доставки повідомлень вище, ніж у аналогів;
- розподіл - сервера Telegram розташовані по всьому світу, що підвищує відмовостійкість;
- відкритість - використання відкритого протоколу MTProto і API, безкоштовних для всіх;

					КНТЕУ 121 06-07.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка чат боту TELEGRAM для замовлення таксі Загальні відомості про приймальну комісію	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.				Р1	4	39
Керівник		Рассамакін В.Я.				Факультет інформаційних технологій, 4 курс, 6 група		
Гарант		Цензура М.О.						
Розроб.		Гричок В.О.						

Секретні чати потрібні для людей, які хочуть забезпечити себе максимально безпечним спілкуванням. Крім використання основного протоколу всі повідомлення шифруються ключами за принципом від пристрою до пристрою. Ніхто не перехопить і не розшифрує повідомлення, у тому числі працівникам Telegram. Повідомлення з таких чатів пересилати не можна і вони не зберігаються на серверах додатки. Але у секретних чатів є два недоліки - це неможливість відкрити чат на пристрої, яка не брала участі в його створенні, і ніхто не захищений від скриншота інтерфейсу Telegram з відкритим секретним чатом.

Мессенджер Telegram доступний для всіх популярних платформ, наприклад, Android OS, iOS, Windows Phone, Linux та інші. Так само в більшості версія для різних операційних систем доступні як мінімум англійську, російську, польську та німецьку мови.



Рис. 1.2 - Інтерфейс Telegram на Windows 10

Зм.	Аркуш	№ докум	Підпис	Дата



Рис. 1.2 - Інтерфейс Telegram на iOS 11

1.2 Чат боти

Останнім часом в месенджерах набирають популярність такі сервіси як чат-боти. Сам термін «чат-бот» був придуманий Майклом Молдін в 1994 році для опису розмовних програм.

Вивчивши сучасний стан використання чат-ботів в месенджерах, можна прийти до висновку, що чат-боти є універсальними засобами, здатними до вирішення різноманітних завдань - від спілкування до розваг, від надання медичної консультації до замовлення товарів і послуг за допомогою спеціалізованих прикладних рішень, від розпізнавання емоцій до вирішення складних консалтингових завдань в службах підтримки клієнтоорієнтованих інформаційних систем.

У будь-якому випадку, в незалежності від платформи, чат-бот - це прикладна програма, яка, отримуючи інформацію від користувача, формує коректні, логічно обґрунтовані відповіді.

На сьогоднішній момент існує величезна різноманітність чат-ботів. Деякі з можливих варіантів представлені нижче:

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата					6

- ігрові чат-боти (квести, рольові ігри);
- рекламні чат-боти (Хрустteam, Coca-Cola);
- новинні чат-боти (Комерсант, Meduza);
- чат-боти для доставлення, магазинів, сервісів (таксі «Максим»,
- піцерія «Папа Джонс», бронювання поїздів і літаків «Туту.рф»);
- чат-боти для консультації й підтримку клієнтів (банк «Точка»,
- каршерінг «UrentCar») та інші.

Останнім часом великі можливості відкривають інформаційні технології, пов'язані з чат-ботами. Вони настільки міцно увійшли в життя людей, що застосовуються в усіх сферах людської діяльності, і з кожним днем їх роль все більше збільшується. В першу чергу це пояснюється тим, що основну частину свого часу люди проводять за смартфоном в email-клієнтах і месенджерах.

Боти - це програми, які виконують різні завдання для користувача, що знаходиться в месенджері. Бот виглядає як звичайний чат, однак спілкування відбувається не з людиною, а з програмою, яка може прийняти замовлення на виклик машини, якщо це - бот таксі, або надіслати свіжі статті, якщо це - новинний бот, або замовити доставлення їжі додому, якщо це - ресторанний бот.

Не так давно чат-боти здобули велику популярність, перевтілившись з розваги в більш серйозну річ, бо вони, в основному, стали використовуватися для вирішення серйозних бізнес-завдань. В епоху інформаційних технологій - це нормальне явище, а тим більше - мережі Інтернет, адже суспільство давно перейшло на «нове» і ділове, і неділові спілкування. По-перше, чат-боти - це «платформи» для вирішення бізнес-завдань. По-друге, чат-бот - це програма, яка підтримує діалог з користувачем, вибираючи відповіді з бази даних: ви запитуєте, де пообідати і тут же отримуєте миттєву відповідь. Крім того, чат-боти виконують безліч корисних функцій у виконанні рутинних операцій, пошуку інформації, об'єднання даних, роботі з клієнтурою.

Чат-бот як віртуальний співрозмовник має базу знань, яка являє собою набори можливих питань користувача і відповідних їм відповідей. Завжди існують якісь прості й легкі по виконання справи, на які не хочеться витратити час. Тут на допомогу завжди можуть прийти чат-боти. Століття друкарських машинок, ходьби по бібліотеках і нескінченних черг в книгарні канув в Лету. Тепер для збору інформації можна використовувати чат-ботів. Звичайно, це робиться за допомогою спілкування з людьми, але аж ніяк не завжди. Наприклад, в рамках якого-небудь заходу чат-боти можуть повідомляти всім учасникам новини і надавати довідкову інформацію.

1.3 Огляд аналогів чат ботів.

1.3.1 Чат бот YouDrive.

Даний бот був розроблений програмістами каршерінг-сервісу YouDrive. Сценарії і функції були розроблені на основі аналізу звернень клієнтів через інші канали зв'язку - телефон, соцмережі, електронну пошту.

Мета бота полягала в тому, щоб зробити процес повідомлення про проблеми з орендованим транспортним засобом максимально простими і зручним для користувачів, а також підвищити швидкість обробки повідомлень співробітниками сервісу, тим самим знизивши час очікування для клієнтів. Для зручності користувачів бот має стандартний набір найбільш часто запитань і відповідних відповідей на них. Також є форма зворотного зв'язку, яка автоматично зв'яже вас з фахівцем по питання, що цікавить. Інтерфейс чат-бота зображений на.



Рис. 1.3 - Функціонал чат-бота YouDriverBot
на платформі Telegram для Window 10

1.3.2 Чат-бот «Спорт-Експрес»

Чат-бот Спорт-Експрес являє собою новинного бота з ухилом в спортивну тематику. Головними його завданнями є:

- ☐ розсилка головних новин в області спорту;
- ☐ висновок списку трансляцій на поточний день;
- ☐ висновок списку результатів на поточний день.



Рис. 1.4 - Приклад інтерфейсу чат-бота Спорт-Експрес

З плюсів можна відзначити:

- максимально спрощений інтерфейс;
- повна зведення про результати всіх подій на поточний день.

Мінусами даного чат-бота є:

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

- можливість виведення результатів тільки на поточний день;
- відсутність посилань на трансляції;
- обмежена кількість спортивних дисциплін.

1.3.3 Бот для замовлення таксі Maxim

Даний чат-бот дозволяє замовити таксі без використання спеціального додатку. У бота є всього 4 команди:

- / start - стандартна обов'язкова команда для активації чат-бота;
- / order - команда для замовлення таксі;
- / cancel - скасувати поточне замовлення;
- / forget - видалити наявну інформацію про користувача.

Після активації бота і відправлення команди для замовлення таксі буде запропоновано ввести номер мобільного телефону. Це необхідно для захисту бота від атак і ідентифікації користувачів. Після чого на мобільний пристрій приходить SMS з перевірочним кодом. Далі бот пропонує вибрати місто і адресу зі списку його частих поїздки або вказати інший. Якщо в налаштуваннях мобільного пристрою включено передача поточних геоданих, то користувач може переслати свої координати, а бот сам спробує знайти відповідний адресу користувача.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

10



Рис. 1.5 - Функціонал чат-бота TaxіMaximBot
на платформі Telegram для iOS 11

1.3.4 Чат-бот «Який рахунок?»

Даний чат-бот надсилає результати футбольних матчів колективів, на які підписаний клієнт. Після вибору команд чат-бот буде в автоматичному режимі надсилати сповіщення після чергового події.



Рис. 1.6 - Інтерфейс чат-бота «Який рахунок?»

Плюсами даного чат-бота є:

- повний список футбольних колективів світу (футбольні клуби і збірні країн);
- можливість отримувати оновлення в режимі реального часу.

Однак існують і мінуси

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

11

- неможливо відредагувати список обраних футбольних колективів;
- інформація обмежена тільки результатами футбольних матчів.

1.4 Висновки до розділу

В цьому розділі розглянуті історичні та технологічні аспекти, сервісів обслуговування населення що пов'язані с таксі, існуючі новітні технології з використанням чат ботів та месенджерів які найбільш актуальні на сьогоднішній день для розробки програмних додатків і програм для таксі.

						КНТЕУ 121 06-07.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			12

РОЗДІЛ 2

ОГЛЯД ІНСТРУМЕНТАЛУ ВИКОРИСТАНОГО ПРИ РОЗРОБЦІ

2.1 HTTP-інтерфейс Bot API

Bot API представляє собою HTTP-інтерфейс для роботи з ботами в Telegram. Кожен бот - це спеціальний аккаунт, 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, серіалізовані в JSON. Всі запити до Telegram Bot API повинні здійснюватися через.

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 запуск бота;
- ☐ sendMessage - метод відправляє текстові повідомлення в клієнт Telegram;
- ☐ sendLocation - метод відправляє повідомлення з координатами в клієнт .

					<i>КНТЕУ 121 06-07.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата	<i>Розробка чат боту TELEGRAM для замовлення таксі</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>				<i>Р2</i>	<i>13</i>	<i>39</i>
<i>Керівник</i>		<i>Рассамакін В.Я.</i>				<i>Факультет інформаційних технологій, 4 курс, 6 група</i>		
<i>Гарант</i>		<i>Цензура М.О.</i>						
<i>Розроб.</i>		<i>Гричок В.О.</i>			<i>Основні етапи проектування та створення веб сайту</i>			

2.2 База даних Sql lite

SQLite - це внутріпроцесна бібліотека, яка реалізує автономний, безсерверний, з нульовою конфігурацією, транзакційний движок бази даних SQL. Код для SQLite є суспільним надбанням і, таким чином, вільний для використання в будь-яких цілях, комерційних або приватних. SQLite - це найпоширеніша в світі база даних з великою кількістю додатків, ніж ми можемо порахувати, включаючи кілька гучних проєктів.

SQLite - це вбудований движок бази даних SQL. На відміну від більшості інших баз даних SQL, у SQLite немає окремого серверного процесу. SQLite читає і записує файли безпосередньо на звичайний диск. Повна база даних SQL з декількома таблицями, індексами, тригерами і уявленнями міститься в одному дисковому файлі. Формат файлу бази даних крос-платформний - ви можете вільно копіювати БД між 32-бітними і 64-бітними системами або між біг-бендінговою і мало-бендінговою архітектурою. Ці можливості роблять SQLite популярним вибором в якості формату файлу програми. Файли баз даних SQLite є рекомендованим Бібліотекою Конгресу США форматом зберігання. Думайте про SQLite не як про заміну Oracle, а як про заміну fopen ().

SQLite - це компактна бібліотека. При включених функціях розмір бібліотеки може бути менш 600KiB, в залежності від цільової платформи і налаштувань оптимізації компілятора. (64-бітний код більше. А деякі оптимізації компілятора, такі як агресивне вбудовування функцій і розгортання циклів, можуть привести до того, що об'єктний код буде значно більше). Існує компроміс між використанням пам'яті і швидкістю. Як правило, SQLite працює швидше, чим більше пам'яті ви йому дасте. Проте, продуктивність, як правило, досить хороша навіть в середовищах з низьким обсягом пам'яті. Залежно від того, як він використовується, SQLite може бути швидше, ніж пряме введення / виведення файлової системи.

SQLite дуже ретельно тестується перед кожним релізом і має репутацію дуже надійною. Велика частина вихідного коду SQLite присвячена виключно

використовується, SQLite може бути швидше, ніж пряме введення / виведення файлової системи.

SQLite дуже ретельно тестується перед кожним релізом і має репутацію дуже надійною. Велика частина вихідного коду SQLite присвячена виключно тестування і верифікації. Автоматичний набір тестів запускає мільйони і мільйони тестових випадків, що включають сотні мільйонів окремих виразів SQL, і досягає 100% охоплення тестуванням філій. SQLite витончено реагує на збої в розподілі пам'яті і помилки дискового введення / виводу. Транзакції є ACID, навіть якщо вони перервані збоєм системи або перебоями в електроживленні. Все це перевіряється автоматизованими тестами з використанням спеціальних тестових джгутів, що імітують збої в роботі системи. Звичайно, навіть при такому тестуванні все одно виникають помилки. Але на відміну від деяких схожих проєктів (особливо комерційних конкурентів) SQLite відкритий і чесний у всіх помилках і надає списки помилок і похвилинні хронології зміни коду.

Кодова база SQLite підтримується міжнародною командою розробників, які працюють над SQLite повний робочий день. Розробники продовжують розширювати можливості SQLite і підвищувати його надійність і продуктивність, зберігаючи при цьому зворотну сумісність з опублікованими специфікацією інтерфейсу, синтаксисом SQL і форматом файлів бази даних. Вихідний код абсолютно безкоштовний для всіх бажаючих, але професійна підтримка також доступна.

2.3 Платформа розробки .NET Core

.NET Core - новітня універсальна платформа розробки, підтримувана Microsoft. Він працює на різних платформах і був перероблений таким чином, щоб зробити .NET швидким, гнучким і сучасним. Це є одним з основних вкладів Microsoft. Тепер розробники можуть створювати додатки для Android, iOS, Linux, Mac і Windows за допомогою .NET, все з відкритим вихідним кодом.

Зм.	Аркуш	№ докум	Підпис	Дата

Характеристики .NET Core

Нижче наведені основні характеристики .NET Core -

Відкритий вихідний код

- .NET Core - це реалізація з відкритим вихідним кодом, яка використовує ліцензії MIT і Apache 2.
- .NET Core є проектом .NET Foundation і доступний на GitHub.
- Як проект з відкритим вихідним кодом, він сприяє більш прозорого процесу розробки і сприяє активному і зацікавленій спільноті.

Крос-платформна

1. Додаток, реалізоване в .NET Core, може бути запущено, і його код може бути повторно використаний незалежно від мети вашого платформи.
2. В даний час він підтримує три основні операційні системи (ОС)
 - Windows
 - Linux
 - MacOS
3. Операційні системи (ОС), ЦП і сценарії додатків з часом будуть розширюватися, надані Microsoft, іншими компаніями і приватними особами.

Гнучке розгортання

1. Для додатків .NET Core може бути два типи розгортання:
 - Залежить від платформи
 - автономне розгортання
2. При розгортанні, що залежить від інфраструктури, ваше додаток залежить від загальносистемної версії .NET Core, на якій встановлені ваше додаток і сторонні залежності.
3. При автономному розгортанні версія .NET Core, яка використовується для створення додатка, також розгортається разом з вашим додатком і сторонніми залежностями і може працювати паралельно з іншими версіями.

Інструменти командного рядка

Всі сценарії продукту можуть бути виконані в командному рядку.

сумісний

- .NET Core сумісний з .NET Framework, Xamarin і Mono через стандартну бібліотеку .NET

модульна

- .NET Core випускається через NuGet в невеликих складальних пакетах.
- .NET Framework - це одна велика збірка, яка містить більшість основних функцій.
- .NET Core доступний у вигляді невеликих функціонально-орієнтованих пакетів.
- Такий модульний підхід дозволяє розробникам оптимізувати свій додаток, включаючи в нього тільки ті пакети NuGet, які їм необхідні.
- Переваги меншої площі програми включають більш сувору безпеку, зниження рівня обслуговування, підвищення продуктивності і зниження витрат в моделі з платою за використання.

2.4 Середовище розробки Visual Studio

Visual Studio Code - це легкий, але потужний редактор вихідного коду, який сумісний з Windows, MacOS і Linux.

IDE підтримує велику кількість популярних мов програмування. Підтримувані розширення і середовища виконання представлені нижче:

- ☐ JavaScript;
- ☐ TypeScript;
- ☐ Node.js;
- ☐ C ++;
- ☐ C #;
- ☐ Java;
- ☐ Python;

- ☐ PHP;
- ☐ Go;
- ☐ .NET;
- ☐ Unity.

Для зручності розробки програмних продуктів VS Code включає в себе вбудований відладчик, інструменти для роботи з Git-репозиторіями, підсвічування синтаксису, засоби для рефакторинга і IntelliSense (технологія автодоповнення, яка пропонує команду за першими літерами). IDE VS Code була анонсована 29 квітня 2015 року компанією Microsoft в рамках конференції Blink. 18 листопада 2015 року редактор був випущений в реліз під ліцензії Массачусетського технологічного інституту. Visual Studio Code заснований на Electron - фреймворк, що дозволяє використовуючи Node.js розробляти настільні додатки, що працюють на движку Blink. Незважаючи на те, що редактор заснований на Electron, він не використовує редактор Atom. Замість нього був реалізований веб-редактор Monaco, розроблений для Visual Studio Online. Інтерфейс IDE Visual Studio Code представлений на малюнку 10.

2.5 Висновки до другого розділу

У цьому розділі було описано інструментал який був використаний при розробці. Для зв'язку із сервером було обрано Bot API який представляє собою HTTP-інтерфейс для роботи з ботами в Telegram. Далі було обрано платформу розробки .NET Core вона працює на різних платформах і є швидкою, гнучкою і сучасною платформою. Підключено це все до внутріпроцесної бібліотеки SQLite, яка реалізує автономний, безсерверний, з нульовою конфігурацією, транзакційний движок бази даних SQL. Та визначилися з середовищем розробки це Visual Studio Code яка є легким , але потужним редактором вихідного коду, який сумісний з Windows, MacOS і Linux.

					<i>КНТЕУ 121 06-07.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ

3.1 Розробка інтерфейса адміністратора

Рішення представляє собою два інтерфейси взаємодії - один для користувача, та один для адміністратора.

Інтерфейс адміністратора призначений для менеджменту серверу. Виконаний у двох реалізаціях.

Одна спроектована на .Net 5.0, на базі архітектури WinForms (з використанням бібліотеки MaterialSkin) для виконання безпосередньо у середовищі CLR (Windows), інша на .Net Core 3.1 у вигляді консольного додатку для виконання на будь-якій платформі.

3.1.1 Додаток адміністратора на базі .Net Core

BotConsole (назва проекту) передбачає найпростіший механізм потенціального менеджменту серверу (наразі являє собою моніторинг подій), та створений для старту рішення на будь-якій платформі.

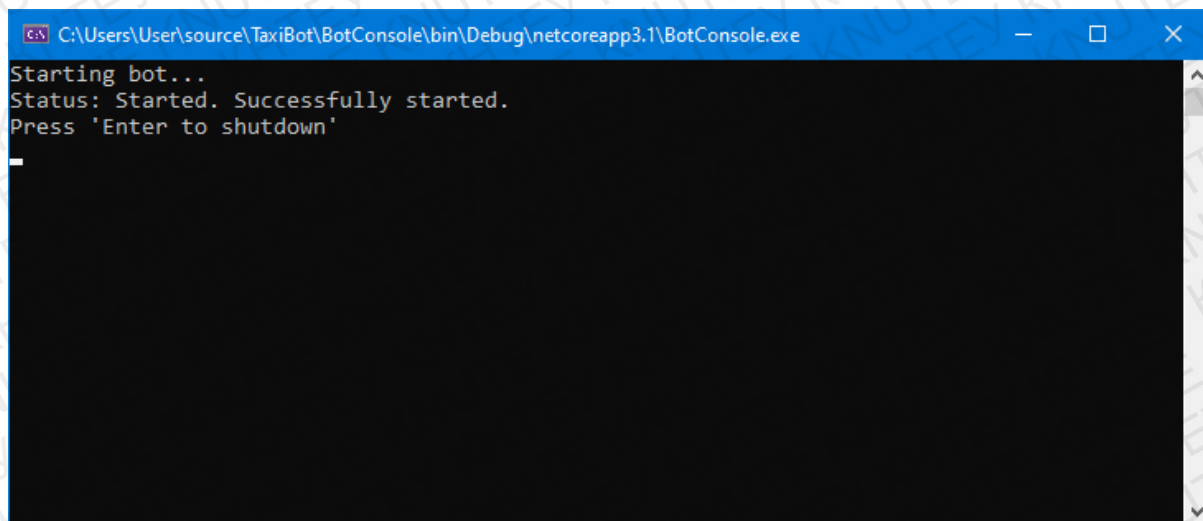


Рис. 3.1 Додаток адміністратора

					<i>КНТЕУ 121 06-07.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О.В.			<i>Розробка чат боту TELEGRAM для замовлення таксі</i>	Стадія	Аркуш	Аркушів
Керівник		Рассамакін В.Я.				РЗ	19	39
Гарант		Цензура М.О.				Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.		Гричок В.О.						
					<i>Реалізація веб сайту та функціоналу онлайн черги</i>			

По своїй суті являє собою оболонку над головною бібліотекою рішення (Core), яка містить основну масу функціоналу.

```
0 references
9  static void Main()
10 {
11     Bot.OnEvent += (IBotStatus status) => Console.WriteLine(status.Message);
12
13     Console.WriteLine(Bot.Init());
14
15     Console.WriteLine("Press 'Enter to shutdown'");
16
17     Console.ReadLine();
18
19     Console.WriteLine(Bot.Shutdown());
20
21     Console.ReadLine();
22 }
```

Рис. 3.2 Підписка на події боту

В цілому підписка на події боту з подальшим відображенням на консолі, команда “стоп” для зупинки серверу.

3.1.2 Додаток адміністратора на базі .Net Framework

Наряду з попередньою реалізацією даний додаток збудований на архітектурі WinForms, що значить більш зрозумілий для користувача (адміністратора) інтерфейс взаємодії. Збудований на фреймворку .Net Framework не він не дає можливості виконуватися на будь-якій платформі, окрім Windows.

Зм.	Аркуш	№ докум	Підпис	Дата

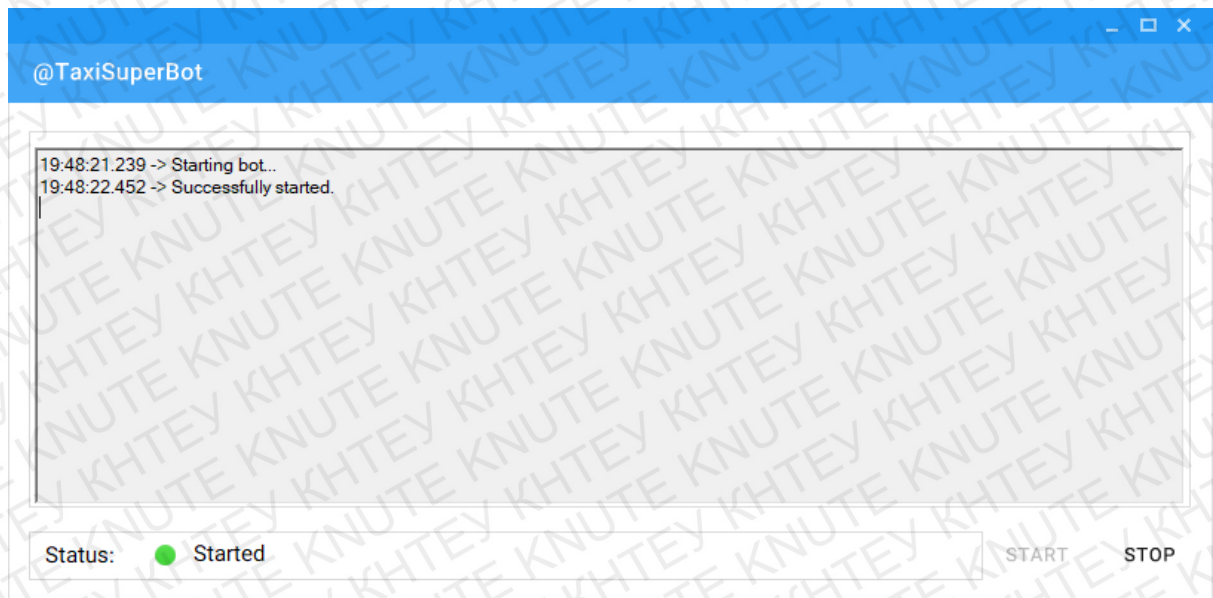


Рис. 3.3 Інтерфейс

Інтерфейс передбачає команди старту/зупинення серверу, лог подій (будь-які події, про які повідомляє бот) та індикацію статусу серверу.

```

5  static class Program
6  {
7      [System.STAThread]
8      static void Main()
9      {
10         Application.SetHighDpiMode(HighDpiMode.SystemAware);
11         Application.EnableVisualStyles();
12         Application.SetCompatibleTextRenderingDefault(false);
13         Application.Run(new MainForm());
14     }
15 }
16

```

Рис. 3.4 Статичний клас Program

Статичний клас Program, який викликає головну форму (MainForm).

```

10 3 references
11  --- public partial class MainForm : MaterialForm
12  --- {
13  ---     1 reference
14  ---     public MainForm()
15  ---     {
16  ---         InitializeComponent();
17  ---         InitStyles();
18  ---         SetStatus(Status.Waiting);
19  ---     }
20

```

Рис. 3.5 Клас типу System.Windows.Forms.Form

Клас типу System.Windows.Forms.Form (MaterialSkin наслідується від вище зазначеного) має наступні методи:

```

21 1 reference
22  --- private void startButton_Click(object sender, EventArgs e)
23  --- {
24  ---     Bot.OnEvent += LogEvent;
25  ---     LogEvent(Bot.Init());
26  --- }

```

Рис.3.6 startButton_Click

startButton_Click - обробник натискання кнопки Start. Виконує підписку методу LogEvent на події бота, та вікликає ініціалізацію бота.

```

27 3 references
28  --- void LogEvent(IBotStatus botStatus)
29  --- {
30  ---     SetStatus(botStatus.Status);
31  ---     if (!string.IsNullOrEmpty(botStatus.Message))
32  ---     {
33  ---         string log = $"{DateTime.Now.ToString("HH:mm:ss.fff")} -> {botStatus.Message}{Environment.NewLine}";
34  ---         if (InvokeRequired)
35  ---             BeginInvoke((Action)(() => logTextBox.AppendText(log)));
36  ---         else
37  ---             logTextBox.AppendText(log);
38  ---     }
39  --- }
40

```

Рис. 3.7 Метод LogEvent

Метод LogEvent - представляє собою обробник подій бота. Встановлює візуальне відображення статусу та виконує запис події у лог подій.

```

2 references
42 void setStatus(Status status)
43 {
44     if (InvokeRequired)
45     {
46         BeginInvoke((Action)(() =>
47             statusLabel.Text = status.ToString();
48             SetAvailability(status);
49             statusPicture.Refresh();
50         )));
51     }
52     else
53     {
54         statusLabel.Text = status.ToString();
55         SetAvailability(status);
56         statusPicture.Refresh();
57     }
58 }

```

Рис. 3.8 Метод setStatus

Метод setStatus - реалізація встановлення візуального відображення статусу, а також активність кнопок start/stop.

```

2 references
59 void SetAvailability(Status status)
60 {
61     switch (status)
62     {
63         case Status.Started:
64         {
65             statusPicture.BackgroundImage = Resources.statusGreen;
66             SetControlsAvailability(true);
67             break;
68         }
69         case Status.Initializing:
70         {
71             statusPicture.BackgroundImage = Resources.statusYellow;
72             SetControlsAvailability(true);
73             break;
74         }
75         case Status.Error:
76         {
77             statusPicture.BackgroundImage = Resources.statusRed;
78             SetControlsAvailability(false);
79             break;
80         }
81         case Status.Waiting:
82         {
83             statusPicture.BackgroundImage = Resources.StatusBlue;
84             SetControlsAvailability(false);
85             break;
86         }
87     }
88 }
89

```

Рис. 3.9 Метод SetAvailability

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

28

Метод SetAvailability - Встановлює активність кнопок start/stop через виклик методу SetControlsAvailability.

```
4 references
90 void SetControlsAvailability(bool connected)
91 {
92     startButton.Enabled = !connected;
93     stopButton.Enabled = connected;
94 }
```

Рис. 3.10 Метод SetControlsAvailability

Метод SetControlsAvailability - Встановлює активність кнопок start/stop.

```
1 reference
96 void InitStyles()
97 {
98     MaterialSkinManager materialSkinManager = MaterialSkinManager.Instance;
99     materialSkinManager.AddFormToManage(this);
100    materialSkinManager.Theme = MaterialSkinManager.Themes.LIGHT;
101
102    materialSkinManager.ColorScheme = new ColorScheme(
103        Primary.Blue400, Primary.Blue500,
104        Primary.Blue500, Accent.LightBlue200,
105        TextShade.WHITE);
106 }
```

Рис. 3.11 Метод InitStyles

Метод InitStyles - встановлює стиль форми (частина функціоналу MaterialSkin).

```
1 reference
108 private void stopButton_Click(object sender, EventArgs e) => LogEvent(Bot.Shutdown());
```

Рис. 3.12 Метод stopButton_Click

Метод stopButton_Click - обробник натискання кнопки Stop, зупиняє сервер.

```
1 reference
110 private void MainForm_FormClosed(object sender, System.Windows.Forms.FormClosedEventArgs e) => Bot.Shutdown();
```

Рис. 3.13 Метод MainForm_FormClosed

Зм.	Аркуш	№ докум	Підпис	Дата

Метод MainForm_FormClosed - обробник закриття форми, призначений для зупинки серверу автоматично при закритті форми.

3.2 Акробация розробки

Основний функціонал реалізовано у бібліотеці Core, збудованой на базі .Net Standart, що дозволяє її імпорт у збірку будь-якої реалізації .Net.

```
8 public static class Bot
9 {
10     public static Action<IBotStatus> OnEvent;
11
12     internal static Context Context { get; set; }
13
14     static DialogsCache cash;
15     static ITelegramBotClient client;
16
17     public static Status Status { get; private set; }
18 }
```

Рис. 3.14 Основний клас - Bot.

Основний клас - Bot

Поля:

- OnEvent - поле типу event. Даний івент виконується на будь-які події боту.
- Context - поле типу Context - представляє клас для роботи з БД (реалізований у бібліотеці DBAccessor).
- cash - поле типу DialogCache - колекція активних діалогів бота.
- client - поле типу інтертфейсу ITelegramBotClient - клієнт телеграм боту. Являє собою оболонку над функціоналом для роботи з API телеграму.
- Status - публічна властивість типу перечислення Status - відображає статус боту у даний момент часу.

Зм.	Аркуш	№ докум	Підпис	Дата

```

2 references
19 .....public static IBotStatus Init()
20 .....{
21 .....    string message = null;
22 .....    try
23 .....    {
24 .....        Status = Status.Initializing;
25 .....        OnEvent(new BotStatus(Status, "Starting bot..."));
26 .....
27 .....        Context = new Context();
28 .....
29 .....        cash = new DialogsCache();
30 .....
31 .....        client = new TelegramBotClient(Resources.APIKey);
32 .....
33 .....        client.OnMessage += OnMessage;
34 .....        client.StartReceiving();
35 .....
36 .....        Status = Status.Started;
37 .....    }
38 .....    catch (Exception ex)
39 .....    {
40 .....        message = ex.Message;
41 .....    }
42 .....
43 .....    return new BotStatus(Status, "Successfully started.");
44 .....}
45

```

Рис. 3.15 Метод Init

Метод Init - метод ініціалізації Боту. Призначений для ініціалізації полей класу.

```

3 references
46 .....public static IBotStatus Shutdown()
47 .....{
48 .....    Status = Status.Initializing;
49 .....    OnEvent(new BotStatus(Status, "Trying to stop..."));
50 .....
51 .....    client.StopReceiving();
52 .....
53 .....    Status = Status.Waiting;
54 .....    return new BotStatus(Status, "Successfully stoped");
55 .....}
56

```

Рис. 3.16 Метод Shutdown

Метод Shutdown - метод зупинки бота.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

28

```

1 reference
57 private static void OnMessage(object sender, MessageEventArgs e)
58 {
59     OnEvent(new BotStatus(Status, $"Message received: '{e.Message.Text}'"));
60
61     cash.TakeMessage(e.Message.Chat.Id, e.Message);
62 }
63

```

Рис. 3.17 Метод OnMessage

Метод OnMessage - обробник події телеграм-клієнта (client). Ініціює подію бота та обробку отриманого повідомлення.

```

6 references
64 internal static async void SendMessage(MessageData data) =>
65     await client.SendTextMessageAsync(
66         data.ChatID,
67         data.Message,
68         replyMarkup: data.ReplyMarkup);
69 }
70 }

```

Рис. 3.18 Метод SendMessage

Метод SendMessage - виконує відправлення повідомлення користувачу.
Допоміжний клас DialogsCache

```

2 references
6 internal class DialogsCache : Dictionary<long, Dialog>
7 {
8     1 reference
9     public void TakeMessage(long chatID, Message message)
10    {
11        if (ContainsKey(chatID))
12            this[chatID].ReceiveMessage(message);
13        else
14            Add(chatID, new Dialog(message));
15    }
16 }

```

Рис. 3.19 Допоміжний клас DialogsCache

Клас являє собою колекцію діалогів з користувачами та їх менеджмент.

Допоміжний клас MessageData

```
3  internal class MessageData
4  {
5      public long ChatID { get; set; }
6
7      public string Message { get; set; }
8
9      public Telegram.Bot.Types.ReplyMarkups.IReplyMarkup ReplyMarkup { get; set; }
10 }
11
```

Рис. 3.20 Допоміжний клас MessageData

Клас являє собою оболонку для інкапсуляції у собі тексту повідомлення для користувача та розмітку кнопок (IReplyMarkup)

Клас Dialog - екземплярний клас, який містить алгоритми діалогу бота з користувачем.

```
11 internal class Dialog
12 {
13     private Order order;
14
15     public long ChatID { get; }
16
17     Dictionary<string, Action<Message>> actions;
18 }
```

Рис. 3.21 Клас Dialog

Поля:

- order - клас типу Order - об'єкт, що представляє замовлення таксі.
- ChatID - публічна властивість, що містить ідентифікатор чату з користувачем (встановлюється сервером Telegram).

- actions - поле типу “словник”, який містить реакції на ключові запити користувача.

```

19 1 reference
20  .... public Dialog(Message message)
21  .... {
22  ....  .... actions = new Dictionary<string, Action<Message>>
23  ....  .... {
24  ....  ....  .... ["/start"] = Start,
25  ....  ....  .... [Resources.Back] = Start,
26  ....  ....  .... [Resources.CreateOrder] = CreateOrder,
27  ....  ....  .... [Resources.Templates] = Templates,
28  ....  ....  .... [Resources.History] = History,
29  ....  ....  .... [Resources.SpecifySupplyAddress] = SpecifySupplyAddress,
30  ....  ....  .... [Resources.SpecifyDestinationAddress] = SpecifyDestinationAddress,
31  ....  ....  .... [Resources.DoOrder] = DoOrder
32  ....  .... };
33  ....  .... ChatID = message.Chat.Id;
34  ....  .... ReceiveMessage(message);
35  .... }
36

```

Рис. 3.22 Конструктор

Конструктор ініціалізує поля класу.

```

37 2 references
38  .... public void ReceiveMessage(Message message)
39  .... {
40  ....  .... if (!string.IsNullOrEmpty(message.Text))
41  ....  .... {
42  ....  ....  .... if (actions.ContainsKey(message.Text))
43  ....  ....  .... actions[message.Text](message);
44  ....  .... }
45  ....  .... if (message.Location != null)
46  ....  ....  .... SetLocation(message);
47  ....  .... }
48

```

Рис. 3.23 Метод ReceiveMessage

Метод ReceiveMessage - обробляє вхідне повідомлення користувача через виклик делегату з колекції actions, при чому у ролі ключа виступає текст, отриманий від користувача.

```

51     private bool greeted;
52     void Start(Message message)
53     {
54         Helpers.RegisterUser(new Models.User
55         {
56             ChatID = message.Chat.Id,
57             Name = message.Chat.Username
58         });
59
60         order = null;
61
62         Bot.SendMessage(new MessageData
63         {
64             ChatID = message.Chat.Id,
65             Message = greeted ? "Выберите действие" : Resources.Greeting,
66             ReplyMarkup = new ReplyKeyboardMarkup(new KeyboardButton[3]
67             {
68                 new KeyboardButton(Resources.CreateOrder),
69                 new KeyboardButton(Resources.Templates),
70                 new KeyboardButton(Resources.History)
71             },
72             resizeKeyboard: true)
73         });
74
75         greeted = true;
76     }

```

Рис. 3.24 Метод Start

Метод Start - метод, що відображає користувачу головне меню інтерфейсу користувача (у додатку Telegram).

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

31

```

84 3 references
85 private void CreateOrder(Message message)
86 {
87     if (order == null)
88     {
89         order = new Order
90         {
91             ChatID = message.Chat.Id,
92             TripDate = DateTime.Now
93         };
94     }
95     var buttons = new List<KeyboardButton>
96     {
97         new KeyboardButton(Resources.Back),
98     };
99     string textMessage;
100     if (order.ReadyToGo)
101     {
102         textMessage = $"Заказ оформлен!{Environment.NewLine}{order.SupplyCoords} -> {order.DestinationCoords}";
103         buttons.Add(new KeyboardButton(Resources.DoOrder));
104     }
105     else
106     {
107         textMessage = Resources.SpecifyTripData;
108         buttons.Add(new KeyboardButton(!order.ValidSupply ? Resources.SpecifySupplyAddress : string.Format(Resources.SpecifySupplyAddress, order.SupplyCoords)));
109         buttons.Add(new KeyboardButton(!order.ValidDestination ? Resources.SpecifyDestinationAddress : string.Format(Resources.SpecifyDestinationAddress, order.DestinationCoords)));
110     }
111     Bot.SendMessage(new MessageData
112     {
113         ChatID = message.Chat.Id,
114         Message = textMessage,
115         ReplyMarkup = new ReplyKeyboardMarkup(buttons, resizeKeyboard: true, oneTimeKeyboard: true)
116     });
117 }
118

```

Рис. 3. 25 Метод CreatOrder

Метод CreatOrder - метод призначений для ініціалізації об'єкту замовлення для подальшого збереження у базу даних.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

32

```

120 1 reference
121  private void SpecifySupplyAddress(Message message)
122  {
123      destinationEdit = false;
124      supplyEdit = true;
125      SaySendLocation(message.Chat.Id);
126  }
127
128 1 reference
129  private void SpecifyDestinationAddress(Message message)
130  {
131      supplyEdit = false;
132      destinationEdit = true;
133      SaySendLocation(message.Chat.Id);
134  }
135
136 2 references
137  private void SaySendLocation(long chatID) => Bot.SendMessage(new MessageData
138  {
139      ChatID = chatID,
140      Message = "Отправьте геолокацію"
141  });
142
143 1 reference
144  private void SetLocation(Message message)
145  {
146      if (supplyEdit)
147      {
148          order.SupplyAddressX = message.Location.Longitude;
149          order.SupplyAddressY = message.Location.Latitude;
150      }
151      else if (destinationEdit)
152      {
153          order.DestinationAddressX = message.Location.Longitude;
154          order.DestinationAddressY = message.Location.Latitude;
155      }
156      CreateOrder(message);
157  }

```

Рис. 3.26 Методи SupplyAddress, SpecifyDestinationAddress, SaySendLocation, SetLocation

Методи SpecifySupplyAddress, SpecifyDestinationAddress, SaySendLocation, SetLocation призначені для зазначення користувачем інформації для підтвердження замовлення.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

34

```

180     1 reference
181     private void Templates(Message message)
182     {
183         Bot.SendMessage(new MessageData
184         {
185             ChatID = message.Chat.Id,
186             Message = "В розробці"
187         });
188         Start(message);
189     }
190
191     #endregion Templates
192
193     #region History
194
195     1 reference
196     private void History(Message message)
197     {
198         Bot.SendMessage(new MessageData
199         {
200             ChatID = message.Chat.Id,
201             Message = Helpers.GetTripsHistory(message.Chat.Id),
202             ReplyMarkup = new ReplyKeyboardMarkup(new KeyboardButton[1]
203             {
204                 new KeyboardButton(Resources.Back)
205             },
206             resizeKeyboard: true)
207         });
208     }

```

Рис. 3.27 Методи Templates та History

Методи Templates та History призначені для відображення меню шаблонів а історії відповідно.

3.3 Інтерфейс користувача

Коли ви заходите в телеграм бота потрібно його запустити, запуск проводиться кнопкою “/start”.



Рис. 3.28 Старт бота

Щоб зробити замовлення потрібно натиснути кнопку “/Создать заказ”.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-07.БР

Аркуш

34

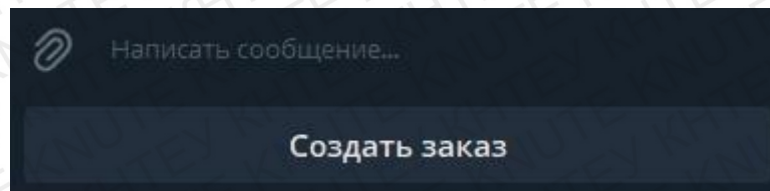


Рис. 3.29 Створення замовлення

Далі потрібно вказати адрес подачі та пункт призначення, це робиться за допомогою геолокації в самому телеграмі.

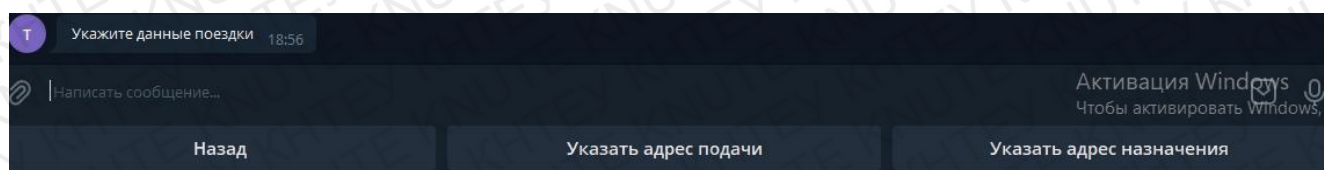


Рис. 3.30 Кнопки для вказання адреси подачі та адреси призначення

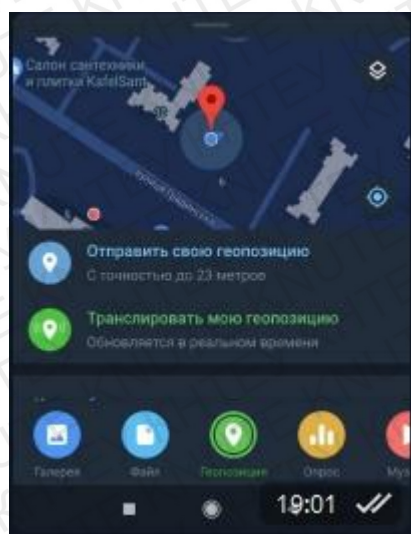


Рис 3.31 Кнопка Геолокації в самому телеграмі.

Після підтвердження замовлення бот його прийме та покаже ваші геоданні та розрахункову вартість замовлення.

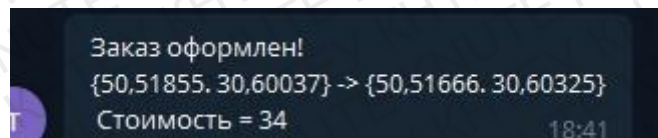


Рис. 3.32 Ваше замовлення.

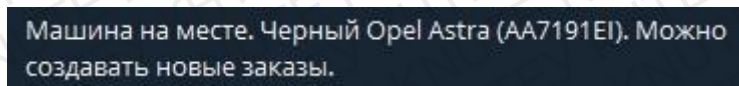


Рис. 3.32 Оповіщення що таксі прибуло.

Через певний час бот оповістить вас що таксі на місці.

Також Бот має такі функції як:

“/Шаблон”.

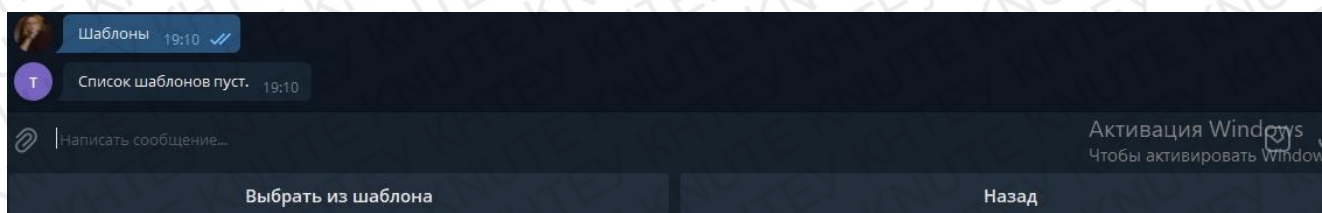


Рис. 3.33 Кнопка шаблон.

Ви можете створити шаблон з любими геоданими по типу Робота-Дім і вам не доведеться кожен раз робити одне і теж замовлення та вказувати геоданні а лише натиснути одну кнопку та підтвердити замовлення.

“/История”

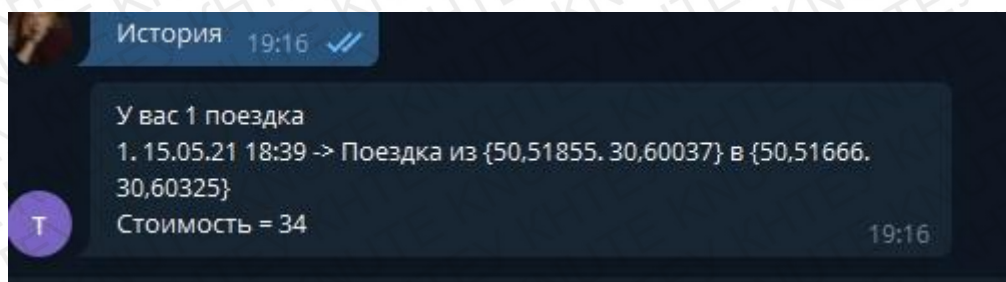


Рис. 3.34 Кнопка історії замовлень.

Кнопка історії замовлень показує ваші попередні замовлення.

3.3 Висновок до розділу 3

В третьому розділі була описана робота над створенням додатку, описаний інтерфейс адміністратора та користувача, також представлений код основних функцій програми.

					<i>КНТЕУ 121 06-07.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання випускної кваліфікаційної роботи був розроблений чат бот Таксі на базі месенджера Telegram, бот призначений для виклику самого таксі за допомогою месенджера. Бот написаний на мові C# та має свій власний інтерфейс адміністратора.

Також були набуті навички розробки Telegram чат-ботів та вдосконаленні знання HTTP протоколу і мови програмування C#.

					КНТЕУ 121 06-07.БР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.	Криворучко О.В.				Розробка чат боту TELEGRAM для замовлення таксі	Стадія	Аркуш	Аркушів
Керівник	Рассамокін В.Я.					ВП	38	39
Гарант	Цензура М.О.					Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.	Гричок В.О							
					Висновки та пропозиції			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Адам Фримен. ASP.NET MVC 5 с примерами на C#5.0 для профессионалов. : пер. с англ. / 2015. -736 стр.
2. Барт де Смет. C# 5 Unleashed. : пер. с англ. / 2013. -1700 стр.
3. Прайс М. C# 8 и .NET Core. Разработка и оптимизация : пер. с англ. / 2003. - 816 стр.

Електронні ресурси

1. https://uk.wikipedia.org/wiki/Миттєві_повідомлення – Месенджер Вікіпедія
2. <https://uk.wikipedia.org/wiki/Чат-бот> – Чат бот Вікіпедія
3. https://uk.wikipedia.org/wiki/C_Sharp – C Sharp Вікіпедія

					КНТЕУ 121 06-07.БР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.	Криворучко О.В.				Розробка чат боту TELEGRAM для замовлення таксі	Стадія	Аркуш	Аркушів
Керівник	Рассамокін В.Я.					СВД	39	39
Гарант	Цензура М.О.					Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.	Гричок В.О.							
					Список використаних джерел			