

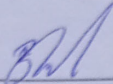
Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»  
Кафедра «Комп'ютерні інформаційні технології»

### Пояснювальна записка

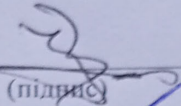
до кваліфікаційної роботи бакалавра

на тему: «Розробка програмного забезпечення для чат-боту таксі»  
за освітньою програмою: «12 Інженерія програмного забезпечення»  
зі спеціальності: «121 Інженерія програмного забезпечення»  
Виконав: студент групи «ПЗ1912»

  
\_\_\_\_\_  
(підпис студента)

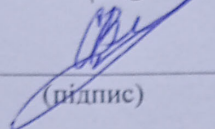
/Кирило ЗАВЕРТКІН/  
(Ім'я ПРІЗВИЩЕ)

Керівник:

  
\_\_\_\_\_  
(підпис)

/доц. Іван КЛИМЕНКО/  
(посада, Ім'я ПРІЗВИЩЕ)

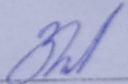
Нормоконтролер:

  
\_\_\_\_\_  
(підпис)

/доц. Світлана ВОЛКОВА/  
(посада, Ім'я ПРІЗВИЩЕ)

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

Студент

  
\_\_\_\_\_  
(підпис)

Ministry of Education and Science of Ukraine  
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»  
Department «Computer information technology»

Explanatory Note  
to Bachelor's Thesis

on the topic: «Software development for a taxi chatbot»  
according to educational curriculum «Software engineering»  
in the Speciality: «121 Software engineering»

Done by the student of the group PZ1912: /Kyrylo ZAVERTKIN/  
Scientific Supervisor: /Ivan KLYMENKO /  
Normative controller: /Svitlana VOLKOVA/

Dnipro – 2023

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет: «Комп'ютерні технології і системи»  
Кафедра: «Комп'ютерні інформаційні технології»  
Рівень вищої освіти: бакалавр  
Освітня програма: «Інженерія програмного забезпечення»  
Спеціальність: «121 Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ  
Завідувач кафедри КІТ  
/Вадим ГОРЯЧКІН/  
(підпис)  
Дата \_\_\_\_\_

## ЗАВДАННЯ

на кваліфікаційну роботу бакалавра  
студенту Заверткіну Кирилу Юрійовичу

1. Тема роботи: «Розробка програмного забезпечення для чат-боту таксі»

Керівник роботи: Клименко Іван Вікторович, доцент  
затверджені наказом № 1209 ст від 07.12.2022

2. Строк подання студентом роботи: 12.06.2023 р.

3. Вихідні дані до роботи: \_\_\_\_\_

4. Пояснювальна записка, зміст (питання, для опрацювання):

4.1 Аналітична частина:

Для отримання якісної програми треба зібрати вимоги та ознайомитися з предметною областю, знайти та оглянути аналоги і визначити яку програму потрібно розробити.

4.2 Основна частина:

Спроекувати систему, яка буде виконувати зазначену задачу, визначити сутності необхідні для реалізації програмного додатка, розробити інтерфейс користувача. Обрати мову програмування на котрій буде реалізовано додаток та розробити програму, що реалізує проект системи.

4.3 Тестування:

Протестувати та налагодити розроблену програму.  
Висновки щодо роботи.

5. Перелік графічного матеріалу:

Скріншоти роботи програми.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
	Збір та аналіз вимог		
	Зовнішнє проектування		
	Внутрішнє проектування		
	Розробка алгоритмів		
	Розробка програмного забезпечення		
	Тестування програмного забезпечення		
	Подання кваліфікаційної роботи до кафедри		
	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії		

Студент

\_\_\_\_\_

(підпис)

Кирило ЗАВЕРТКІН

(Ім'я ПРІЗВИЩЕ)

К  
е  
р  
і  
в  
н  
и  
к

(підпис)

(Ім'я ПРІЗВИЩЕ)

р  
о  
б  
о  
т  
и

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра:

Об'єктом розробки є чат-бот, призначений для водіїв таксі. Його основна функція - отримувати замовлення, створювати реквізити для оплати, управляти тарифами та надавати додаткову інформацію водіям.

Метою даної роботи є створення чат-бота, який надає зручний інтерфейс для водіїв таксі з можливістю приймати замовлення, керувати реквізитами оплати та тарифами, а також отримувати додаткову інформацію. Для досягнення поставленої мети були застосовані наступні методи: аналіз вимог користувачів, розробка архітектури чат-боту, програмування та інтеграція з необхідними сервісами, тестування та оптимізація роботи чат-боту.

Для розробки чат-бота було використано мову програмування C# та Telegram Bot API. Була створена класова структура, що дозволяє виконувати різні функції, такі як приймання замовлень, створення реквізитів для оплати, управління тарифами та надання додаткової інформації водіям. Для зручного взаємодії з користувачами використовується Inline Keyboard з кнопками.

Під час розробки чат-бота для водіїв таксі було створено функціонал, який дозволяє водіям приймати замовлення, створювати реквізити для оплати, керувати тарифами та отримувати необхідну інформацію. Чат-бот забезпечує зручну взаємодію водіїв з системою та полегшує їхню роботу, дозволяючи ефективно керувати замовленнями і забезпечувати якісний сервіс для клієнтів.

Значення роботи полягає в тому, що розробка чат-бота для водіїв таксі має велике значення для покращення процесу замовлення та обробки заявок. Чат-бот допомагає водіям швидко і зручно приймати та виконувати замовлення, а також забезпечує їм доступ до необхідної інформації про тарифи, реквізити оплати та інші деталі. Це сприяє підвищенню якості обслуговування пасажирів та збільшенню задоволення від користування послугами таксі.

Пояснювальна записка складається зі вступу, п'яти розділів, висновків, бібліографічного списку та додатків.

Вступ описує суть, мету та актуальність роботи (3 сторінки).

Перший розділ: збір та аналіз вимог (11 сторінок).

Другий розділ: проектування системи (10 сторінок).

Третій розділ: розробка програми (13 сторінок).

Четвертий розділ: тестування та налагодження (22 сторінки).

П'ятий розділ: аналіз та висновки (3 сторінок).

Додатки: робочий проект. Таблиць – 3 , рисунків – 30, бібліографія – 23 джерела.

Ключові слова: таксі, бот, чат, комунікація, клієнт, система, розробка, інтеграція, сервіс, водій, пасажир, замовлення, обробка, автоматизація, інтерфейс, інтелект, натуральна мова, оптимізація, рекомендації, аналіз даних, додаток.

# ЗМІСТ

<b>Пояснювальна записка .....</b>	<b>1</b>
<b>ЗАВДАННЯ .....</b>	<b>3</b>
<b>КАЛЕНДАРНИЙ ПЛАН.....</b>	<b>4</b>
<b>ВСТУП .....</b>	<b>8</b>
<b>1 ЗБІР ТА АНАЛІЗ ВИМОГ .....</b>	<b>10</b>
1.1 Анкетування .....	10
1.2 Аналіз конкурентів.....	12
2.1. Зовнішнє проектування .....	15
2.2. Внутрішнє проектування.....	19
<b>3 РОЗРОБКА ПРОГРАМИ.....</b>	<b>25</b>
<b>4 ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ.....</b>	<b>33</b>
<b>АНАЛІЗ ТА ВИСНОВКИ .....</b>	<b>46</b>
<b>Список використаних джерел.....</b>	<b>48</b>

## ВСТУП

**Актуальність роботи.** В сучасному світі спостерігається стрімкий розвиток технологій, що суттєво впливає на різні сфери життя, включаючи галузь таксі. Зростання популярності онлайн-сервісів таксі, таких як Uber, Bolt, DiDi та інші, відкрило нові можливості та створило великий попит на зручні та ефективні інструменти для водіїв таксі.

Традиційні методи замовлення таксі за допомогою дзвінка або особистої зустрічі поступово стають застарілими і менш ефективними. У цьому контексті розробка чат-бота для водіїв таксі набуває особливої актуальності, оскільки вона відповідає сучасним вимогам і вирішує проблеми, з якими стикаються водії.

Використання чат-бота для водіїв таксі дозволить автоматизувати процес прийому та обробки замовлень, що сприятиме поліпшенню ефективності роботи водіїв та підвищенню рівня обслуговування пасажирів. Водії зможуть швидко та зручно приймати замовлення, створювати реквізити для оплати, управляти тарифами та отримувати необхідну інформацію про маршрути та пасажирів, що сприятиме високоякісній та ефективній роботі таксі-служби.

Більшість сучасних водіїв таксі володіють смартфонами та користуються месенджерами, такими як Telegram. Розробка чат-бота, який буде інтегрований з месенджером Telegram, відповідає актуальним потребам водіїв та сприяє поліпшенню комунікації з клієнтами.

Крім того, розвиток інтернет-технологій та швидкий доступ до інформації робить чат-боти важливим інструментом для автоматизації рутинних завдань водіїв таксі. Використання чат-бота дозволить водіям швидко отримувати необхідну інформацію про замовлення, пасажирів та маршрути, що сприятиме їхній продуктивності та забезпечить більш точне та ефективне виконання замовлень.

**Мета роботи.** Метою даного дослідження є розробка чат-бота для водіїв таксі з метою полегшення їхньої роботи та покращення процесу прийому та обробки замовлень. Основною метою розробки є забезпечення зручного та ефективного інструменту, який водії таксі зможуть використовувати для керування своїми замовленнями, отримання необхідної інформації та забезпечення якісного обслуговування пасажирів.

Цей чат-бот дозволить водіям легко та швидко приймати замовлення через месенджер Telegram, створювати реквізити для оплати, встановлювати тарифи, а також отримувати важливу інформацію про пасажирів та маршрути. Завдяки цьому водії зможуть оптимізувати свій час та ресурси, забезпечуючи більш швидку та точну обробку замовлень.

Крім того, метою розробки чат-бота є покращення комунікації між водіями та клієнтами. Чат-бот забезпечить зручний канал зв'язку, де водії та клієнти зможуть

обмінюватися необхідною інформацією, такою як деталі маршруту, додаткові побажання або зміни в замовленні. Це підвищить рівень задоволеності клієнтів та сприятиме формуванню позитивного іміджу таксі-служби.

**Експлуатаційне призначення.** Експлуатаційне призначення розробленого чат-бота для водіїв таксі передбачає його практичне застосування та користування водіями, які працюють в таксі-службах. Кінцевими користувачами цього розробленого ПЗ є самі водії таксі, які є активними учасниками процесу перевезення пасажирів.

Розроблений чат-бот буде використовуватися водіями таксі різних компаній, незалежно від їхнього досвіду та стажу. Це означає, що він буде доступним та корисним для новачків, які тільки починають працювати в сфері таксі, а також для досвідчених водіїв, які мають значний досвід роботи [4].

Користувачі розробленого ПЗ, тобто водії таксі, зможуть легко та зручно користуватися чат-ботом з месенджера Telegram. Вони зможуть приймати замовлення, отримувати інформацію про пасажирів та маршрути, керувати тарифами та створювати реквізити для оплати. Цей чат-бот стане незамінним інструментом для організації та управління роботою водіїв, сприяючи підвищенню їхньої продуктивності та ефективності.

Крім водіїв та клієнтів, іншими зацікавленими сторонами, які скористаються розробленим ПЗ, є таксі-компанії та адміністратори таксі-системи. Вони зможуть використовувати чат-бота для моніторингу та аналізу роботи водіїв, отримання статистичної інформації щодо виконаних замовлень та фінансових показників. Це надасть їм зручний інструмент для керування та планування діяльності таксі-служби

# 1 ЗБІР ТА АНАЛІЗ ВИМОГ

## 1.1 Анкетування

Для підвищення точності даних, отриманих у результаті анкетування, їх необхідно повторити декілька разів; при цьому необхідна величина вибірки даних встановлюється методами математичної статистики, виходячи з встановленого рівня статистичної похибки.

Дані питання були використанні для того, що найчастіше використовують користувачі під час поїздки на таксі.

Таблиця 1. Приклади питань для збору вимог

№ з/п	Тип запитання	Опис запитання	Приклад відповіді на питання
1	Як часто ви користуєтесь послугами таксі?	Визначення частоти використання таксі.	Кілька разів на тиждень. Точна кількість разів виклику таксі за певний проміжок часу (проміжок часу теж варто вказати).
2	Які основні функції ви очікуєте від чат-бота таксі?	Встановлення головних функцій, які користувачі хотіли б бачити у чат-боті таксі.	Розгорнута відповідь на кшталт: замовлення таксі, розрахунок вартості поїздки, стеження за статусом замовлення.
3	Які додаткові можливості ви б хотіли бачити в чат-боті таксі?	Визначення додаткових функцій, які користувачі бажають мати в чат-боті таксі.	Розгорнута відповідь на кшталт: відгуки та оцінки водіїв, підтримка кількох мов, сповіщення про прибуття таксі.

4	Яка мова ви б воліли використовувати для взаємодії з чат-ботом таксі?	Встановлення переважної мови для інтерфейсу чат-бота таксі.	Мова та/або перелік мов.
5	Які критерії безпеки ви вважаєте важливими при користуванні чат-ботом таксі?	Визначення основних аспектів безпеки, які користувачі хочуть бачити в чат-боті таксі.	Розгорнута відповідь на кшталт: захищеність особистих даних, безпека платежів.
6	Як багато часу ви готові витратити на очікування таксі?	Встановлення максимально прийнятнього часу очікування замовленої таксі.	Приблизна кількість хвилин та/або годин очікування
7	Чи бажаєте ви мати можливість відстежувати маршрут таксі в реальному часі?	Визначення потреби користувачів у відстеженні маршруту автомобіля в режимі реального часу.	Так або ні.
8	Яка форма оплати вам зручна при користуванні таксі через чат-бот?	Встановлення переважної форми оплати для поїздок замовлених через чат-бот таксі.	Розгорнута відповідь на кшталт: кредитна/дебетова картка, готівка, електронний платіжний сервіс.
9	Чи важливо для вас	Визначення потреби користувачів у	Так або ні.

	мати можливість оцінювати якість обслуговування таксі та водіїв?	можливості залишати відгуки та оцінювати якість послуг таксі та водіїв.	
10	Які додаткові функції чат-бота таксі ви б хотіли бачити для поліпшення комунікації з операторами?	Встановлення додаткових функцій, які сприяють поліпшенню комунікації з водіями таксі.	Розгорнута відповідь на кшталт: можливість зв'язатися з оператором безпосередньо через чат-бот, надсилання спеціальних інструкцій.

## 1.2 Аналіз конкурентів

В рамках цієї роботи було проведено аналіз конкурентів у сфері чат-ботів таксі з метою визначення функціональних можливостей, переваг та недоліків і встановлення ключових вимог до розробленого чат-боту таксі. Аналіз конкурентів відіграє важливу роль у процесі розробки програмного продукту, оскільки він дозволяє отримати цінну інформацію про вже існуючі рішення на ринку та врахувати їх при формулюванні вимог.

Під час проведення аналізу було обрано кілька популярних чат-ботів таксі в Україні, а саме: Uklon, Bolt, Let's Go та Uber. Вибір цих конкурентів зумовлений їх широкою популярністю, значним покриттям ринку та репутацією вже утверджених гравців у даній галузі. Дослідження цих конкурентів дозволить нам зрозуміти, які функції та можливості можуть бути вимогами для нашого проекту, а також виявити їх переваги та недоліки [21].

Отримані результати аналізу конкурентів стануть основою для формулювання ключових вимог до нашого розробленого чат-боту таксі.

1. Uklon є одним з найпопулярніших сервісів таксі в Україні. Вони надають чат-бота для водіїв, який має широкий функціонал, включаючи прийом та виконання замовлень, навігацію, обмін повідомленнями з клієнтами та інші корисні функції. Інтерфейс є зручним та інтуїтивно зрозумілим. Uber також надає підтримку для водіїв і регулярно оновлює свій чат-бот з новими функціями та покращеннями.

2. Volt є ще одним популярним сервісом таксі в Україні. Їх чат-бот для водіїв також має розширений функціонал, включаючи прийом замовлень, навігацію, спілкування з клієнтами та інші корисні можливості. Інтерфейс Volt є простим та зрозумілим. Вони також забезпечують підтримку водіям та оновлюють свій чат-бот для поліпшення користувацького досвіду.
3. Uber є одним з найпопулярніших сервісів таксі в Україні. Вони надають чат-бота для водіїв, який має широкий функціонал, включаючи прийом та виконання замовлень, навігацію, обмін повідомленнями з клієнтами та інші корисні функції. Інтерфейс є зручним та інтуїтивно зрозумілим. Uber також надає підтримку для водіїв і регулярно оновлює свій чат-бот з новими функціями та покращеннями.
4. Let's Go є ще одним чат-ботом для водіїв таксі в Україні. Вони пропонують широкий спектр функцій, включаючи прийом та виконання замовлень, навігацію, взаємодію з клієнтами та можливість створювати реквізити для оплати. Інтерфейс Let's Go відзначається своєю зручністю та простотою використання. Команда розробників постійно вдосконалює функціонал чат-бота та забезпечує підтримку для водіїв.

При порівнянні цих чат-ботів за розглянутими критеріями можна зробити такі спостереження:

- Всі вищезгадані чат-боти для водіїв таксі в Україні надають розширений функціонал, включаючи обробку замовлень, навігацію та взаємодію з клієнтами.
- Інтерфейси цих чат-ботів є зрозумілими та простими у використанні, що сприяє зручній роботі водіїв та покращує їх користувацький досвід.
- Всі розглянуті чат-боти надають підтримку своїм користувачам, а також регулярно вносять оновлення та покращення до своїх систем.

Загалом, конкурентна ситуація в галузі чат-ботів для водіїв таксі в Україні досить активна. Різноманітні сервіси пропонують широкий функціонал та забезпечують зручний користувацький досвід. Вирішенням цієї конкуренції може стати постійне

вдосконалення функціоналу, інтерфейсу та надання якісної підтримки користувачам чат-ботів.

## Висновки до першого розділу

В даному розділі були підняті питання збору та аналізу вимог. Було здійснено пошук та огляд програмних аналогів, що дозволило виявити їх обмеження та визначити недоліки, які ми плануємо уникнути у нашому додатку. Проведений аналіз вимог дозволив чітко сформулювати завдання та напрямки розробки програмного додатку чат-боту таксі. Отримані вимоги стануть основою для подальшої реалізації та розвитку додатку з метою досягнення поставлених цілей та задоволення потреб користувачів.

## ПРОЄКТУВАННЯ

### 2.1. Зовнішнє проєктування

#### 2.1.1 Функціональне призначення

Функціональне призначення чат-бота для водіїв таксі полягає у забезпеченні основної дії, а саме управління та координації процесу отримання та виконання замовлень. Це включає такі основні функції:

- **Прийом замовлень:** Чат-бот дозволяє водіям таксі отримувати замовлення в режимі реального часу. Водій отримує повідомлення про нове замовлення, включаючи інформацію про місце виклику, пункт призначення та інші деталі, необхідні для виконання поїздки.
- **Навігація:** Чат-бот надає можливість водіям таксі використовувати навігаційні функції для зручного переміщення до пункту виклику та призначення. Це може включати інтеграцію з картографічними сервісами, побудову оптимального маршруту та надання вказівок під час поїздки.
- **Створення реквізитів для оплати:** Чат-бот може допомагати водіям таксі створювати реквізити для оплати поїздок. Це може включати генерацію кодів, створення електронних рахунків або інших методів оплати.

#### 2.1.2 Експлуатаційне призначення

- **Експлуатаційне призначення чат-бота для водіїв таксі** полягає у нематеріальній вигоді, яку можна отримати при його використанні. Нижче наведено декілька прикладів такого призначення:
- **Підвищення ефективності:** Використання чат-бота дозволяє водіям таксі знизити часові витрати та збільшити продуктивність. Швидкий та зручний доступ до замовлень, навігаційних інструментів та інших функцій допомагає водіям ефективно виконувати поїздки, зменшуючи затримки та марнотратство часу.
- **Точність розрахунків:** Чат-бот забезпечує точність розрахунків та фінансову прозорість для водіїв таксі. Він автоматично обраховує тарифи за поїздки, враховуючи відстань, час та інші фактори. Це дозволяє водіям отримувати

справедливу винагороду за свою роботу і уникати спорів з клієнтами або компанією таксі.

- Зручність у взаємодії з клієнтами: Використання чат-бота спрощує взаємодію водіїв таксі з клієнтами. Заовлення, комунікація та обмін даними відбуваються швидко та зручно через текстові повідомлення. Це забезпечує зручність та ефективність обслуговування клієнтів.
- Підвищення надійності: Чат-бот може покращити надійність процесу прийому та виконання замовлень. Він автоматично сповіщає водіїв про нові замовлення та надає важливі деталі, такі як адреса виклику та призначення.
- Забезпечення зручності та доступності: Використання чат-бота робить процес замовлення та керування таксі більш зручним та доступним для водіїв. Вони можуть отримувати замовлення в режимі реального часу без необхідності телефонувати або спілкуватися особисто. Це дозволяє водіям ефективно планувати свій робочий час та займатися виконанням поїздок без зайвих перешкод.
- Покращення задоволеності клієнтів: Використання чат-бота сприяє покращенню задоволеності клієнтів таксі. Швидка обробка замовлень, точність розрахунків та зручна комунікація з водієм допомагають забезпечити позитивний досвід поїздок для клієнтів. Це може призвести до збільшення популярності та впевненості клієнтів у використанні таксі з чат-ботом.
- Оптимізація робочих процесів: Чат-бот допомагає водіям таксі оптимізувати свої робочі процеси. Він забезпечує швидкий доступ до необхідних інформаційних ресурсів, таких як мапи, тарифи та реквізити оплати. Це сприяє ефективному плануванню маршрутів, зменшенню помилок та скороченню часу на адміністративні операції

### 2.1.3 Функціональні вимоги

Функціональні вимоги можуть включати такі функції, як:

- Реєстрація водіїв: Система повинна забезпечувати можливість реєстрації нових водіїв таксі. Це включає збір необхідної інформації, такої як особисті дані, контактна інформація та документи, що підтверджують право керування транспортним засобом.
- Прийом та обробка замовлень: Система повинна дозволяти водіям таксі отримувати та обробляти замовлення в режимі реального часу. Це включає отримання деталей замовлення, таких як місце виклику, призначення, час початку та інші важливі відомості.
- Навігація та маршрутизація: Система повинна забезпечувати навігаційні функції для водіїв таксі, включаючи визначення оптимального маршруту до пункту виклику та призначення. Вона може використовувати картографічні сервіси та алгоритми маршрутизації для покращення точності та ефективності переміщення.
- Генерація розрахунків: Система повинна автоматично генерувати розрахунки за виконані поїздки. Вона повинна враховувати відстань, час початку та завершення поїздки, тарифи, знижки або націнки, які можуть бути застосовані, а також інші фактори, визначені в правилах тарифікації. Генерація розрахунків повинна бути точною і надійною, забезпечуючи правильну винагороду для водіїв.

#### 2.1.4 Вхідні дані

Вхідними даними програмного додатку є:

- програмний код на мові програмування C#;
- файли з розширенням .cs із програмним кодом на мові програмування C# (кожен файл відповідає за окрему сутність);

#### 2.1.5 Вихідні дані

Вхідними даними програмного додатку є:

- текстові повідомлення для користувачів щодо інформацію про поїздку;

- текстові повідомлення для користувачів щодо налаштувань картки для платежів;
- кнопки для меню;
- кнопки для навігації між підпунктами меню;

## 2.1.6 Опис зовнішнього інформаційного середовища, в якому буде експлуатуватися ПЗ.

### 2.1.6.1 Канали вводу-виводу:

Користувачі: взаємодія з чат-ботом може відбуватися через мобільний додаток, веб-інтерфейс або популярні месенджери, такі як Telegram.

Водії: водії таксі можуть мати окремий доступ до системи через мобільний додаток або веб-інтерфейс, щоб отримувати та виконувати замовлення.

Адміністратори: адміністратори системи можуть мати доступ до панелі управління, де вони можуть керувати користувачами, водіями та налаштуваннями системи.

### 2.1.6.2 Інформаційні об'єкти:

Замовлення: інформація про замовлення, включаючи початкову та кінцеву точки, деталі клієнта та статус замовлення.

Користувачі: дані про користувачів, такі як ім'я, контактна інформація, історія поїздок та платіжні дані.

Водії: профілі водіїв, включаючи особисті дані, технічні характеристики автомобілів, статус доступності та історію поїздок.

Оплата: інформація про оплату поїздок, включаючи дані про кредитні картки та транзакції.

Зовнішнє інформаційне середовище, в якому буде експлуатуватися ПЗ чат-бота для водіїв таксі, включає наступні елементи:

- Смартфони та комп'ютери водіїв: Водії таксі будуть використовувати смартфони або комп'ютери з підтримкою веб-браузера для доступу до чат-бота. Ці пристрої слугують як канал вводу-виводу для взаємодії з програмним забезпеченням.
- Мобільні мережі: Чат-бот буде працювати в режимі онлайн і взаємодіяти з водіями таксі через мобільні мережі з доступом до Інтернету. Це забезпечує постійну зв'язок і передачу даних між ПЗ та водіями.

- База даних: Інформація про водіїв, замовлення, розрахунки та інші деталі будуть зберігатися у відповідній базі даних. Ця база даних слугує як інформаційний об'єкт, з яким взаємодіє ПЗ для отримання, зберігання та оновлення необхідної інформації.
- API (Application Programming Interface): ПЗ може використовувати API сторонніх сервісів, наприклад, для отримання картографічних даних або розрахунку вартості поїздок. Ці API слугують як джерела додаткової інформації та функціональності для ПЗ [23].
- Операційна система: Чат-бот може бути розроблений для різних операційних систем, таких як Android, iOS або Windows. Відповідні програми або додатки, які підтримують ці операційні системи, можуть бути необхідні для функціонування ПЗ.

#### 2.1.6.3 Програмне забезпечення:

Серверна частина: ПЗ для обробки та керування замовленнями, базою даних користувачів та водіїв, маршрутизації замовлень та оплати.

Клієнтська частина: мобільні додатки для користувачів та водіїв, веб-інтерфейс та панель управління.

Взаємодія ПЗ чат-бота таксі з іншими програмними продуктами включає роботу із популярними месенджерами такими, як Telegram, для зручної комунікації з користувачами та водіями через чат-інтерфейс [4]

## 2.2. Внутрішнє проєктування

Для реалізації вищезазначених функціональних характеристик програмного додатку, було спроектовано та розроблено класи і форми, які відповідають за користувацький інтерфейс користувача та визначення відповідності тексту програми графічному представленню алгоритму.

### 2.2.1 Проєктування класів системи

Розроблені класи відобразимо на діаграмах класів, рис. 2.1

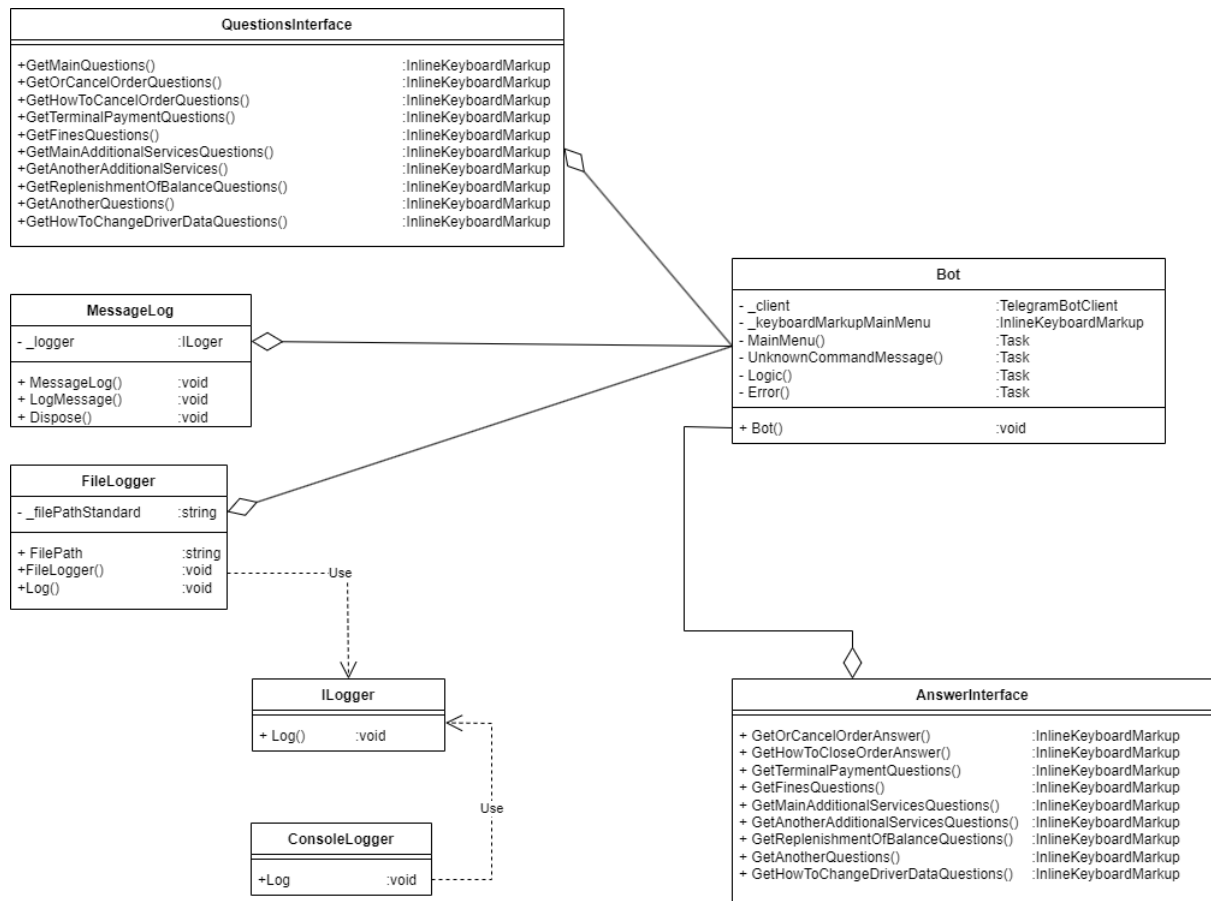


Рисунок 2.1 – Діаграма класів рівня логіки

Для розроблених діаграм класів було виділено наступні рівні:

- логіки – включає класи, що відповідають за логіку програми, а саме за визначення відповідності тексту програми графічному представленню алгоритмів;

Розглянемо кожний розроблений клас більш детально:

- Bot – клас рівня логіки, який відповідає за клієнт чат-боту та нові повідомлення (кліки на кнопки) від користувача;
- ConsoleLogger – клас рівня логіки, який відповідає за забезпечення можливості виводити повідомлення про помилки у консоль для подальшого моніторингу та аналізу.
- FileLogger – клас рівня логіки, який відповідає за можливість записувати повідомлення про помилки до файлу, і використовується для збереження логів помилок для подальшого аналізу і відлагодження.
- ILogger – клас рівня логіки, який визначає прототип для функціональності логування помилок. Конкретні реалізації логгерів, які реалізують цей інтерфейс, повинні надати свою власну реалізацію методу **Log**, відповідно до

своїх потреб. Використання цього інтерфейсу дозволяє використовувати принцип Dependency Injection для логгерів. Замість того, щоб жорстко залежати від конкретної реалізації логгера, класи, які потребують функціональності логування, можуть використовувати цей інтерфейс як залежність. Це дозволяє легко замінювати реалізацію логгера іншою, встановленою за допомогою DI-контейнера.

- MessageLog – клас рівня логіки, який відповідає за забезпечення виклику методу логування на об'єкті логгера, що надає гнучкість і можливість легко замінити реалізацію логування через використання принципу Dependency Injection. Клас MessageLog реалізує інтерфейс IDisposable. Це означає, що клас може мати ресурси, які потрібно звільнити, і для цього використовується метод Dispose(). У даному коді метод Dispose() не має реалізації, але його можна використовувати для звільнення ресурсів, якщо вони присутні у класі MessageLog.
- AnswerInterface – клас рівня логіки, який надає зручний спосіб створення інтерфейсу користувача з кнопками для відповідей на питання в чат-боті.
- QuestionsInterface – клас рівня логіки, який надає зручний спосіб створення інтерфейсу користувача з кнопками для стандартних меню в чат-боті. Кнопки допомагають користувачам взаємодіяти з ботом та вибирати відповідні опції.

Спроекований інтерфейс користувача відображено на рис. 2.2 – 2.3.

Інтерфейс користувача містить наступні елементи:

- віконце з текстовою інформацією та відповідями на запити користувача;
- кнопка навігації назад;
- кнопка навігації до чату із оператором таксі;
- кнопка навігації до головного меню;
- кнопки навігації між пунктами меню;

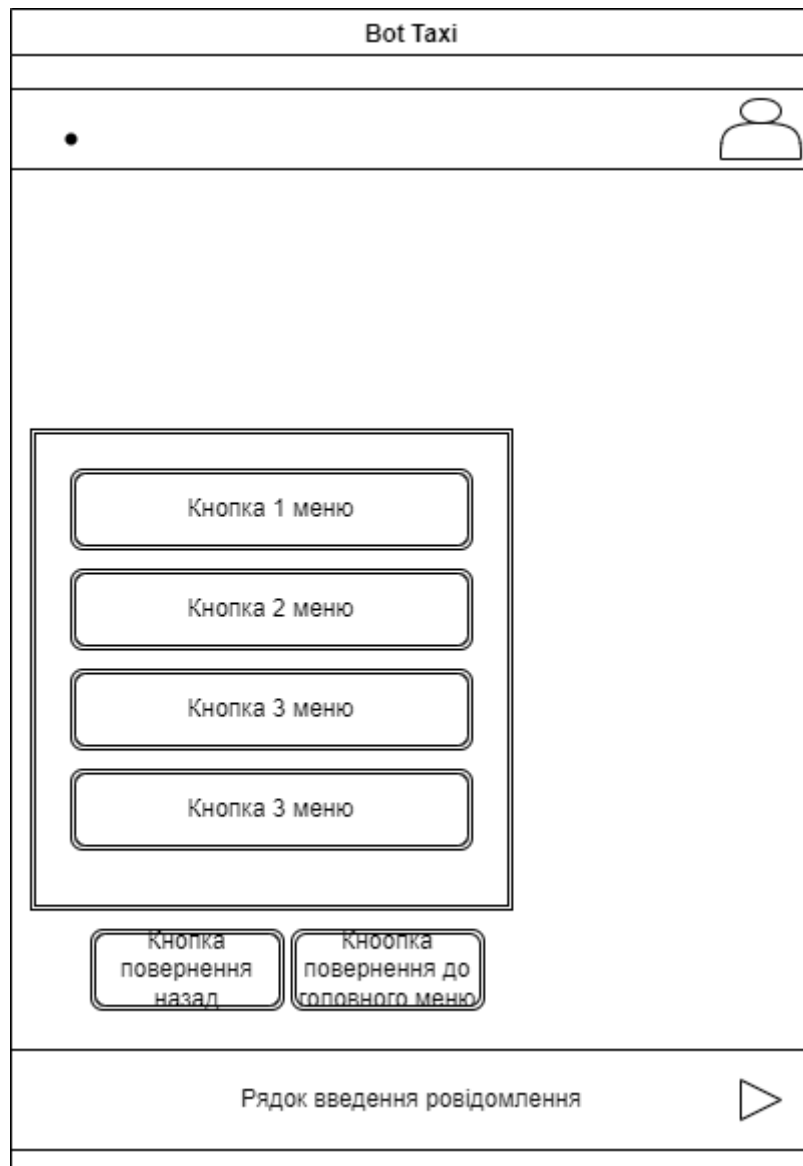


Рисунок 2.2 – Ескіз екранної форми ChatBot Taxi

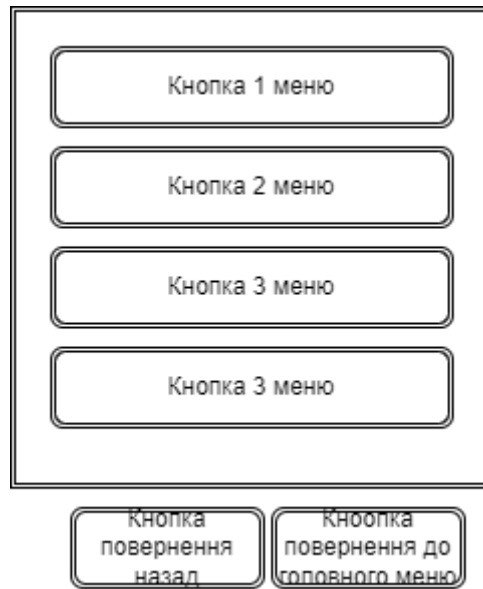


Рисунок 2.3 – Ескіз головного меню

Взаємодія користувача з графічним інтерфейсом відображена на діаграмі станів (рис.

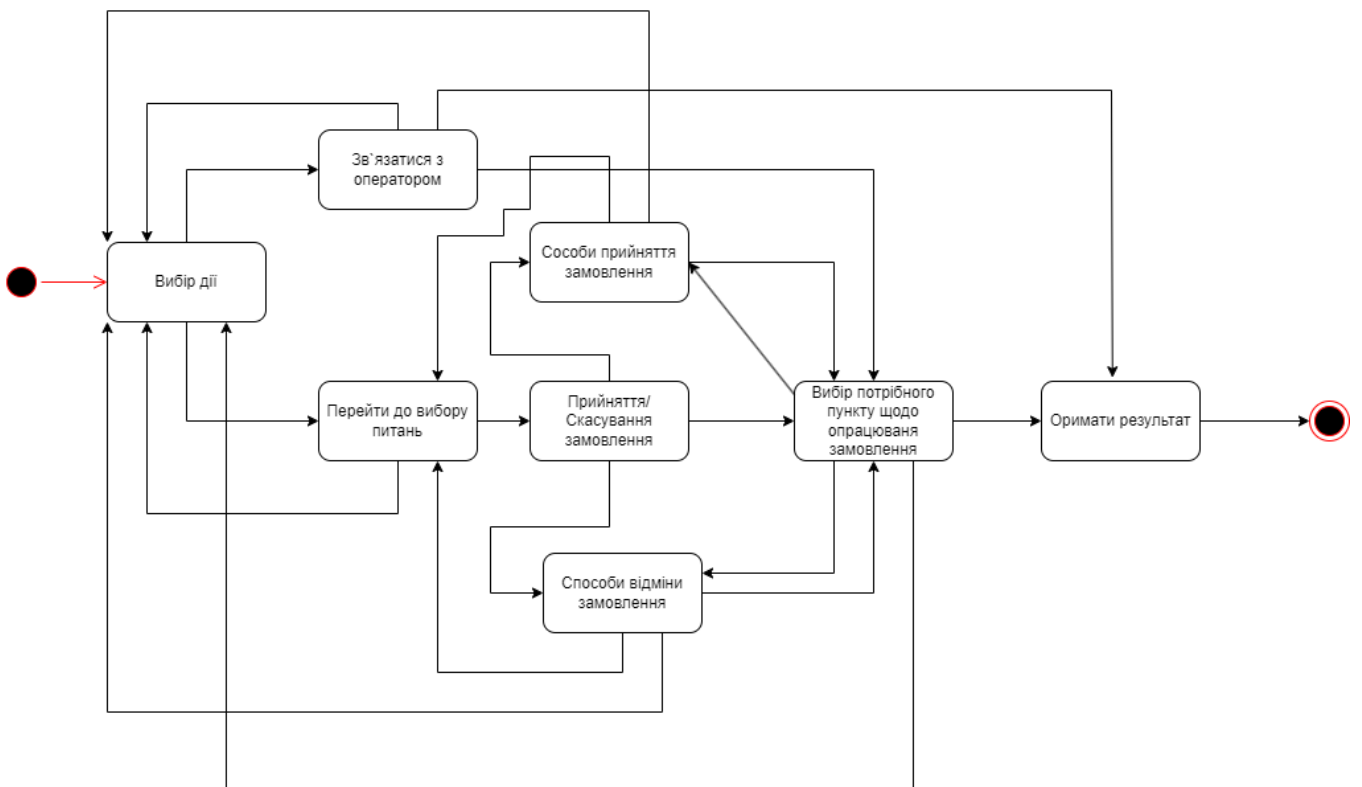


Рисунок 3.4 – Діаграма станів взаємодії користувача з вікном меню

Висновки до другого розділу

В даному розділі було виконано проектування та розробку інструментального забезпечення для визначення відповідності алгоритму та його графічного зображення.

Було визначено вхідні та вихідні дані програмного додатку, а також його функціональні характеристики.

Також було визначено архітектуру додатку, яка складається з трьох рівнів: представлення, логіки та даних було спроектовано основні класи системи та їх взаємодію між собою.

Було обрано модель інтерфейсу користувача, а саме GUI, яка складається з діалогу на основі меню та діалогу на основі екранних форм. Також було спроектовано та реалізовано екранні форми програмного додатку.

## 3 РОЗРОБКА ПРОГРАМИ

Вибір мови C# є відмінним варіантом для програмування такого типу системи. C# є потужною та розширюваною мовою програмування з великим набором функціональностей та бібліотек, які полегшують розробку [2].

У програмній реалізації чат-боту таксі, методи можуть бути використані для реалізації різних функцій, таких як обробка замовлень, взаємодія з користувачем, маршрутизація водіїв, оплата тощо. Методи можуть бути створені для кожного модуля системи та виконувати певні завдання.

Алгоритми визначають послідовність дій, які система повинна виконати для досягнення певних результатів. В програмній реалізації чат-боту таксі алгоритми використані для обробки замовлень, керування потоком даних, розподілу ресурсів.

### 3.1 Метод покрокової деталізації

Метод покрокової деталізації дозволяє розбити складну задачу на менші, більш керовані кроки. Це сприяє кращому розумінню задачі і полегшує розробку програми.

Метод розбиття на етапи передбачає поділ задачі на підзадачі і деталізацію кожної з них, це дає змогу створити чітку структуру програми і зробити її більш організованою і керованою.

Також під час роботи з цим методом можна зосередитись на кожному кроці окремо і перевіряти правильність його виконання. Це полегшує процес розробки, виявлення помилок і налагодження програми [5].

Запишемо алгоритм за методом покрокової деталізації. Перепишемо подану задачу у вигляді специфікації вхідних та вихідних даних.

Вхід: запит користувача – текстовий рядок, який містить питання від користувача.

Вихід: відповідь чат-бота – текстовий рядок, який містить відповідь на запит користувача.

Розіб'ємо задачу на підзадачі:

#### 1. Прийняття запиту:

1.1. Отримання текстового повідомлення від користувача.

1.2. Аналіз введеного тексту для визначення наміру користувача.

#### 2. Розпізнавання команд:

2.1. Порівняння наміру користувача зі списком підтримуваних команд.

2.2. Визначення необхідної дії на основі розпізнаної команди.

3. Обробка команди:
  - 3.1. Виконання відповідних операцій, пов'язаних з командою.
  - 3.2. Передача необхідних даних та параметрів для виконання дії.
4. Генерація відповіді:
  - 4.1. Створення текстової відповіді на основі виконаної дії.
  - 4.2. Форматування відповіді для відображення користувачу.
5. Відправка відповіді:
  - 5.1. Надсилання сформованої відповіді користувачеві через відповідний канал.
6. Завершення запиту:
  - 6.1. Очікування нового запиту від користувача або завершення роботи чат-бота.

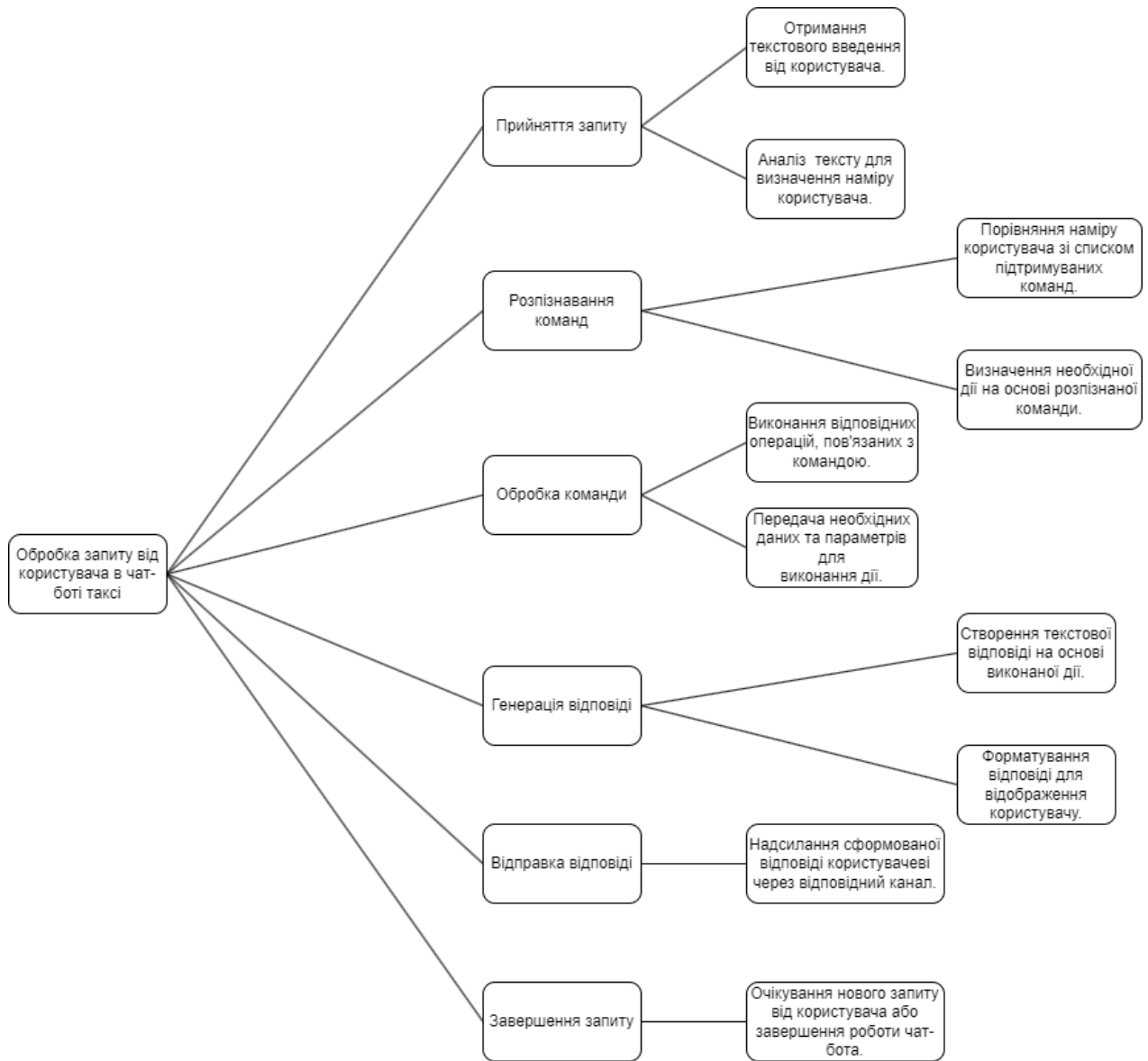
Кожна з під задач є значно простішою, ніж задача в цілому. Підзадачі можуть бути реалізовані окремими функціями (наприклад, 2.1, 2.2., 2.3), або бути частиною однієї

Цю ж задачу представимо деревом рис. 3.1 та таблицею каскадів (табл. 2). У каскадному представленні зробимо на один рівень деталізації більше, щоб наблизитися до безпосередньої реалізації.

Таблиця 2 – Каскадне представлення алгоритму

Задача	Підзадачі	
Обробка запиту ввід користувача в чат-боті таксі	Прийняття запиту	Отримання текстового введення від користувача.
		Аналіз тексту для визначення наміру користувача.
	Розпізнавання команд	Порівняння наміру користувача зі списком підтримуваних команд.
		Визначення необхідної дії на основі розпізнаної команди.

	Обробка команди	Виконання відповідних операцій, пов'язаних з командою.
		Передача необхідних даних та параметрів для виконання дії.
	Генерація відповіді	Створення текстової відповіді на основі виконаної дії.
		Форматування відповіді для відображення користувачу.
	Відправка відповіді	Надсилання сформованої відповіді користувачеві через відповідний канал.
	Завершення запиту	Очікування нового запиту від користувача або завершення роботи чат-бота.



### 3.1 Дерево каскадів

Блок-схеми дозволяють візуально подати послідовність кроків алгоритму, умови прийняття рішень, цикли, функції та інші елементи. Кожен блок представляє окремий крок або операцію, а стрілки вказують напрямок потоку виконання.

Графічне представлення за допомогою блок-схем має кілька переваг, зокрема:

1. Блок-схеми використовують стандартні символи та зв'язки, що дозволяє легко сприймати та розуміти структуру алгоритму без необхідності детального знання програмування.
2. Блок-схеми дозволяють логічно розділити алгоритм на окремі блоки та зв'язати їх за допомогою стрілок. Це допомагає зрозуміти послідовність виконання кожного кроку та взаємозв'язки між ними.

3. Графічне представлення за допомогою блок-схем дає можливість швидко оцінити структуру алгоритму, виявити можливі помилки або складні місця та зробити відповідні зміни.
4. Блок-схеми можуть бути легко відредаговані та модифіковані, що дозволяє швидко вносити зміни в алгоритми без необхідності переписування повного коду.

Визначмо види алгоритмів та їх реалізацій:

- лінійний алгоритм;
- алгоритм розгалуження;
- циклічний алгоритм з постумовою;
- вкладеність алгоритмів розгалуження;

До представлених тем не було додано лістинг програми, оскільки дані алгоритми були складені природньою мовою для більшого розуміння користувачам, які не знаються на програмних мовах.

1. Лінійний алгоритм (рис.3.1):



Рисунок 3.2 – Блок-схема лінійного алгоритму

2. Алгоритм розгалуження (рис 3.3):

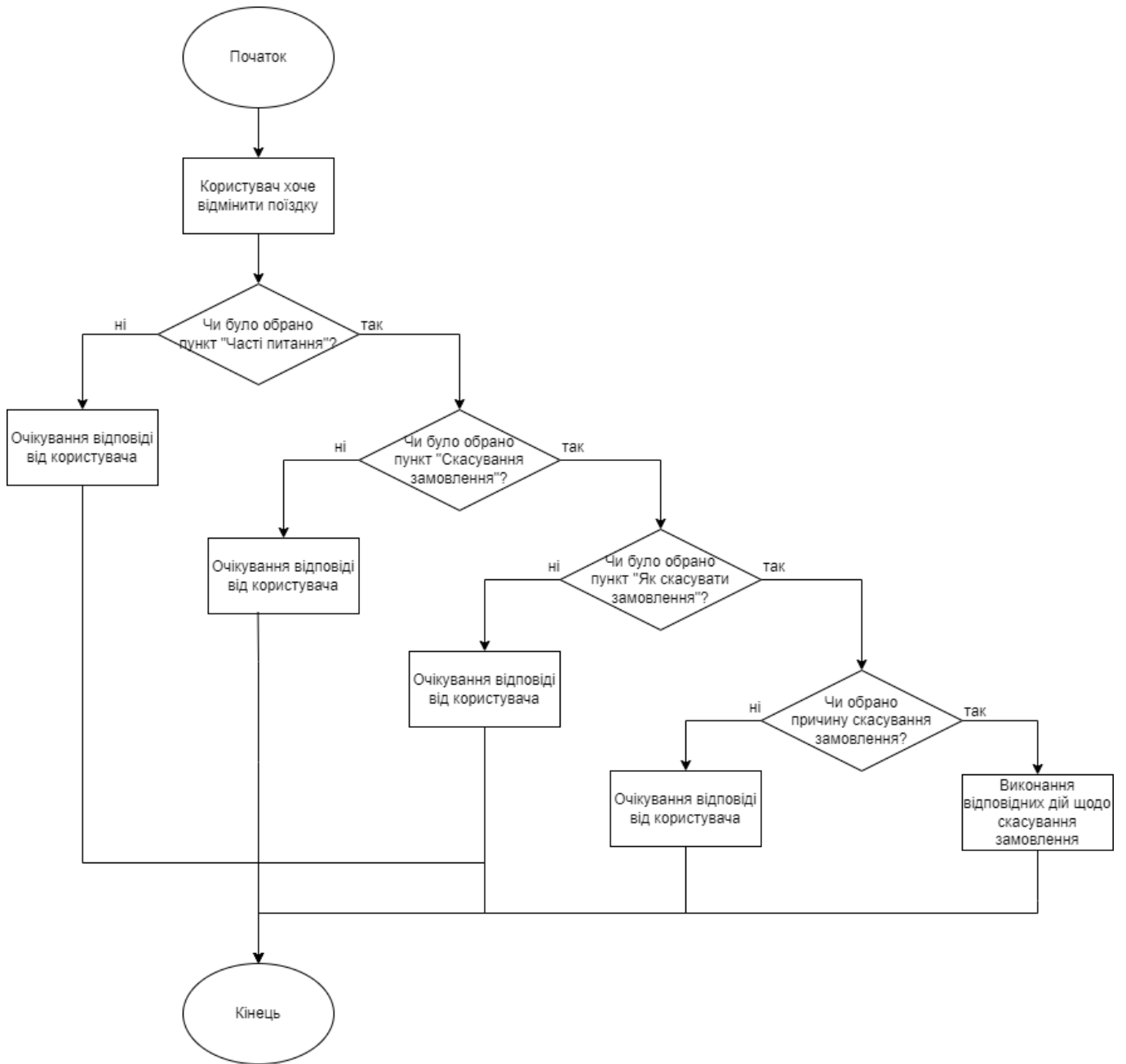


Рисунок 3.3 – Блок-схема алгоритму розгалуження з вкладеними конструкціями

### 3. Циклічний алгоритм з пост умовою (рис 3.4):



Рисунок 3.4 – Блок-схема циклічного алгоритму з пост умовою

#### Висновок до третього розділу

Далі було проведено визначення сутності методів та конструювання алгоритмів. Завдяки методу покрокової деталізації були розкриті деталі кожного кроку виконання програми, розглянуті взаємозв'язки та відповідальність модулів системи. Цей підхід дозволив забезпечити чіткість та структурованість програмної реалізації.

В результаті третього розділу пояснювальної записки було зроблено важливий крок у програмному проектуванні, забезпечено чітке розуміння вибору мови програмування, методів та алгоритмів. Це стало основою для подальшої успішної реалізації проекту чат-боту таксі.

## 4 ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ

Однією з ключових фаз в процесі розробки програмного забезпечення є налагодження та тестування. Цей етап має на меті перевірити функціональність, надійність та відповідність програмного забезпечення вимогам та очікуванням користувачів. Нашою ціллю під час тестування чат-бота для водіїв таксі є забезпечення якісної та безперебійної роботи програми, щоб забезпечити найкращий досвід для користувачів.

Основні аспекти, на які ми зосереджуємося під час тестування, включають:

1. **Функціональність:** Ми перевіряємо, чи виконує чат-бот всі свої функції відповідно до вимог. Це охоплює обробку запитів, навігацію за допомогою меню та інші функції, які були визначені в попередніх етапах розробки.
2. **Надійність:** Ми перевіряємо стійкість та надійність чат-бота в умовах реального використання. Це означає тестування на витривалість, відновлення після збоїв та оптимальну обробку помилок.
3. **Взаємодія з користувачами:** Ми перевіряємо, наскільки легко користувачі можуть взаємодіяти з чат-ботом. Це включає тестування інтерфейсу, зрозумілість повідомлень та зручність використання для водіїв таксі.
4. **Стабільність та продуктивність:** Ми тестуємо чат-бот на стабільність його роботи та продуктивність в умовах реального завантаження. Це допомагає забезпечити, що програма працює швидко та ефективно, навіть при великій кількості одночасних користувачів.
5. **Сумісність:** Ми перевіряємо, як чат-бот взаємодіє з різними пристроями та операційними системами. Це включає тестування на різних моделях смартфонів, планшетах та комп'ютерах.

Нашою метою під час тестування є виявлення та усунення будь-яких помилок, аномалій та недоліків у програмному забезпеченні, щоб забезпечити надійну та задовільну роботу чат-бота для водіїв таксі (табл.3).

Таблиця 3 – Тестування та налагодження.

Номер тесту	Опис ситуації	Очікуваний результат
-------------	---------------	----------------------

1	Запуск системи для початку тестування (рис. 4.1)	Виведення меню.
2	Вибір пункту меню «Часті питання» (рис. 4.2)	Виведення можливих частих питань, з якими найчастіше стикаються водії під час роботи.
3	Вибір питання про прийняття/скасування замовлень (рис. 4.3)	Виведення можливих варіантів опрацювання ситуації.
4	Вибір пунктів про принципи скасування та прийняття замовлень (рис.4.4)	При виборі цих двох пунктів буде показане повідомлення про те, що детальну інформацію можна знайти на сайті компанії. Оскільки ця система є тестовим варіантом, сайт компанії таксі не було розміщено.
5	Вибір пункту «Як скасувати замовлення?» (рис. 4.5)	Виведення умов, за якими можна скасувати замовлення.
6	Вибір пункту «Клієнт не виходить на зв'язок» (рис. 4.6)	Виведення інформації для користувача, що робити в такому випадку.
7	Вибір пункту «Клієнт не виходить до авто» (рис. 4.7)	Виведення інформації для користувача, що робити в такому випадку.

8	Вибір пункту «Поломка авто в процесі поїздки з клієнтами» (рис. 4.8)	Виведення інформації для користувача, що робити в такому випадку.
9	Вибір пункту «Не поміщається багаж (вантаж)» (рис. 4.9)	Виведення інформації для користувача, що робити в такому випадку.
10	Вибір пункту «Некоректна поведінка пасажирів» (рис. 4.10)	Виведення інформації для користувача, що робити в такому випадку.
11	Вибір пункту «Оплата безготівкових замовлень» (рис. 4.11)	Виведення варіантів щодо зарахувань безготівкових замовлень.
12	Вибір пункту «Куди надходять гроші по БГ замовленням?» (рис. 4.12)	Виведення інформації для користувача.
13	Вибір пункту «Чи можна регулювати зарахування?» (рис. 4.13)	Виведення інформації для користувача.
14	Вибір пункту «Штрафи/компенсації» (рис. 4.14)	Виведення підменю.
15	Вибір пункту «За що знімаються штрафи» (рис. 4.15)	Виведення інформації для користувача.
16	Вибір пункту «Тарифи/додаткові послуги» (рис. 4.16)	Виведення підменю.
17	Вибір пункту «Додатковий багаж в салоні» (рис. 4.17)	Виведення інформації для користувача.
18	Вибір пункту «Їду з твариною» (рис.4.18)	Виведення інформації для користувача.
19	Вибір пункту «Кондиціонер» (рис. 4.19)	Виведення інформації для користувача.

20	Вибір пункту «Інші додаткові послуги» (рис 4.20)	Виведення підменю.
21	Вибір пункту «Водій, який не палить» (рис. 4.21)	Виведення інформації для користувача.
22	Вибір пункту «Англомовний водій» (4.22)	Виведення інформації для користувача.
23	Вибір пункту «Зустріч з табличкою» (рис.4.23)	Виведення інформації для користувача.
24	Вибір пункту «Тиша в салоні» (рис. 4.24)	Виведення інформації для користувача.

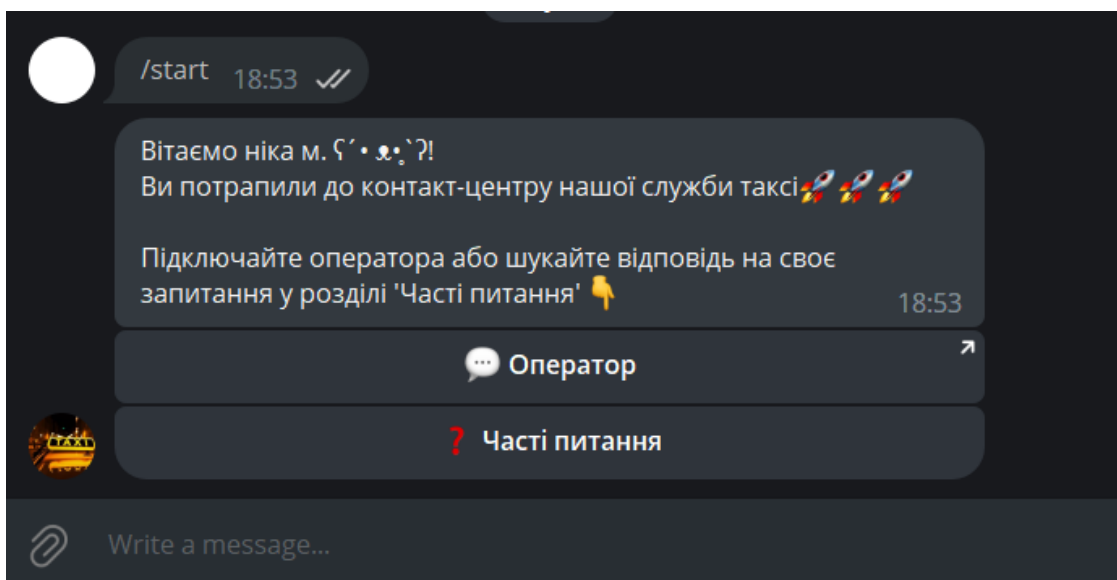


Рисунок 4.1 – Тест 1.

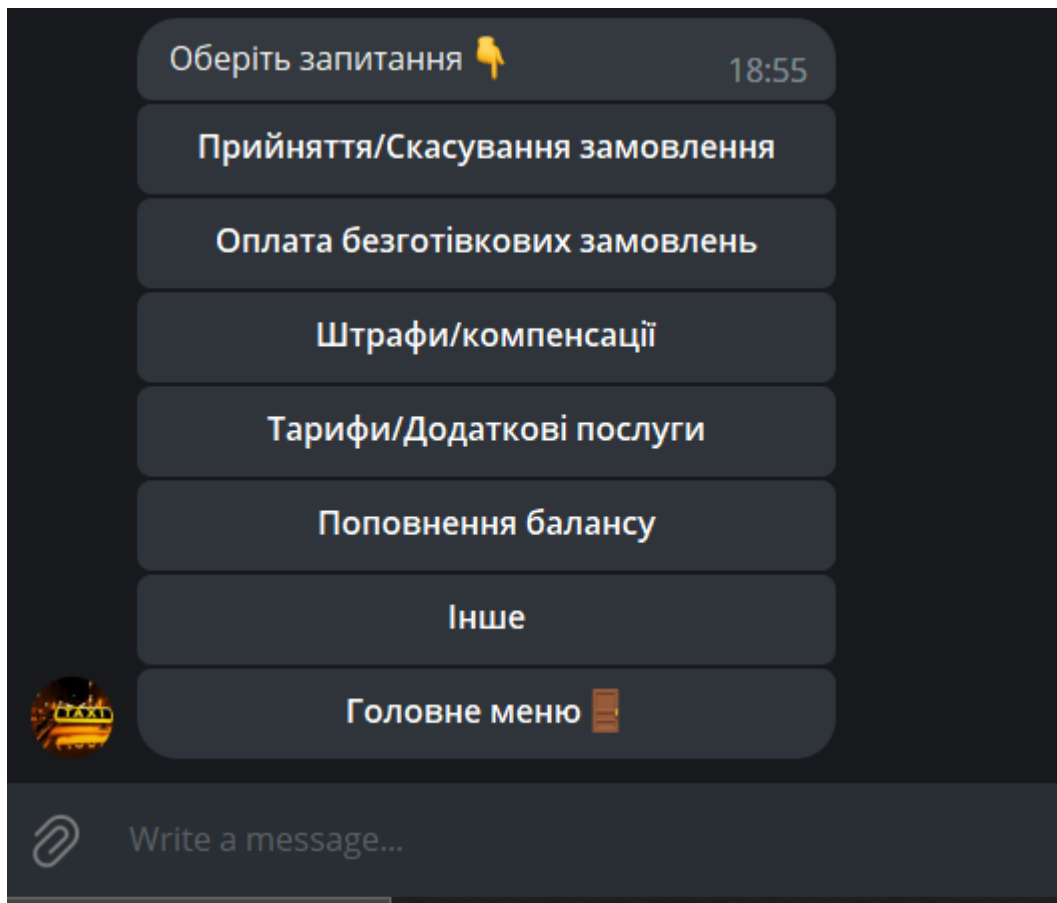


Рисунок 4.2 – Тест 2.

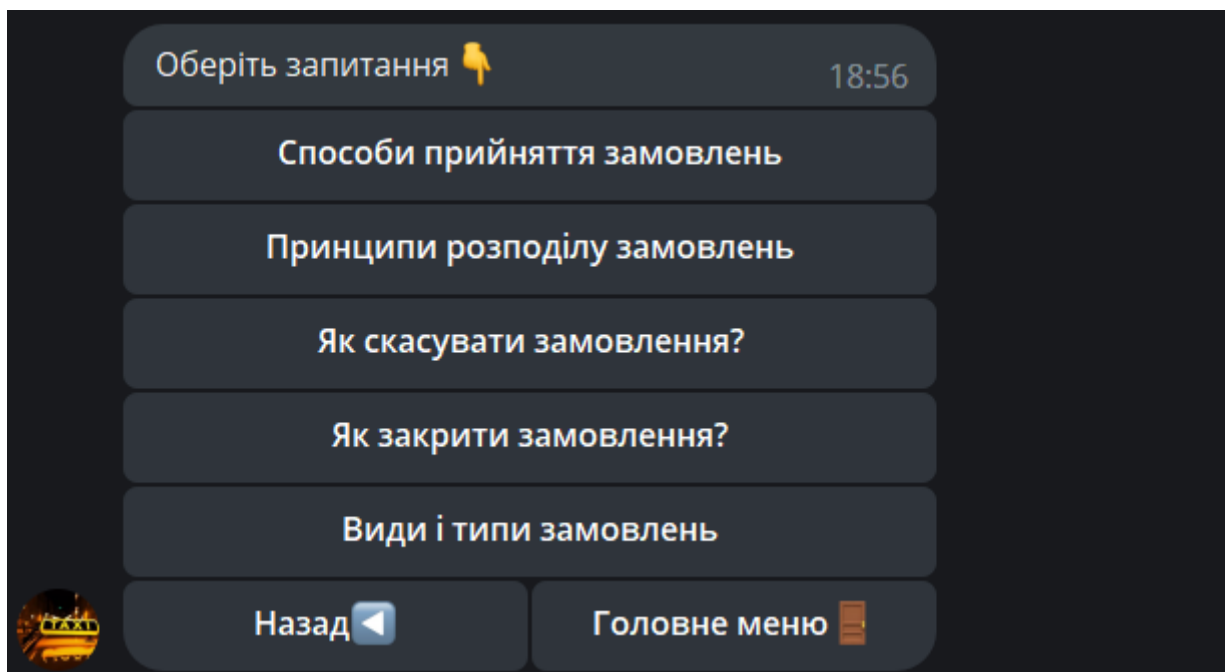


Рисунок 4.3 – Тест 3

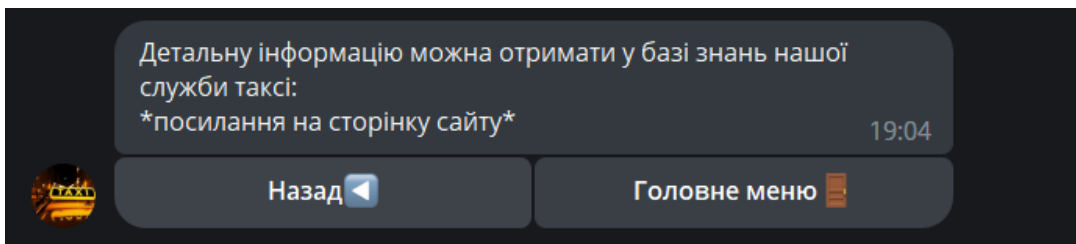


Рисунок 4.4 – Тест 4

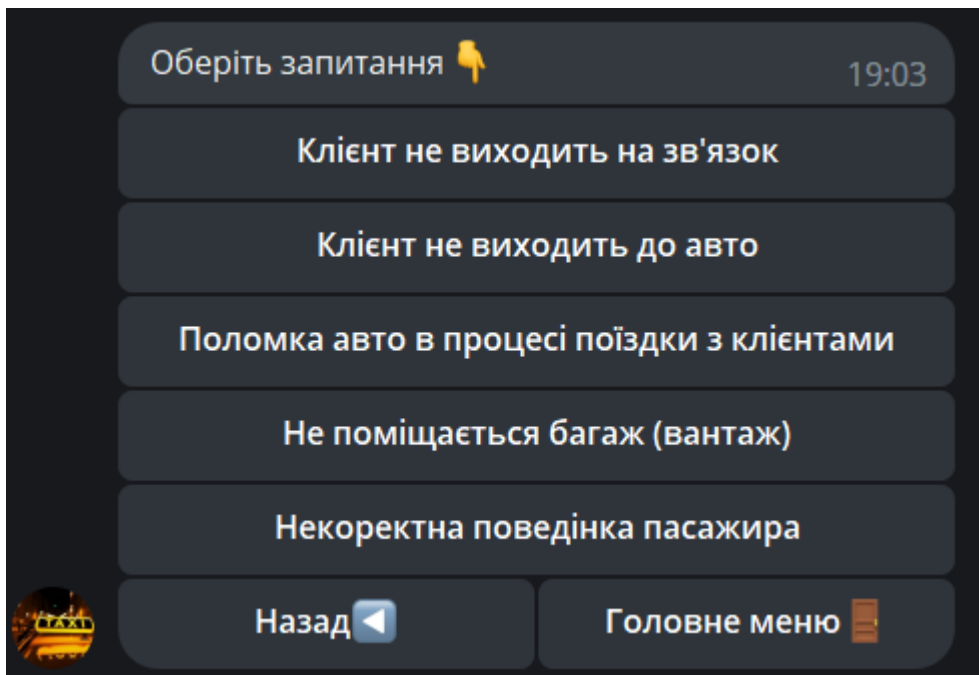


Рисунок 4.5 – Тест 5

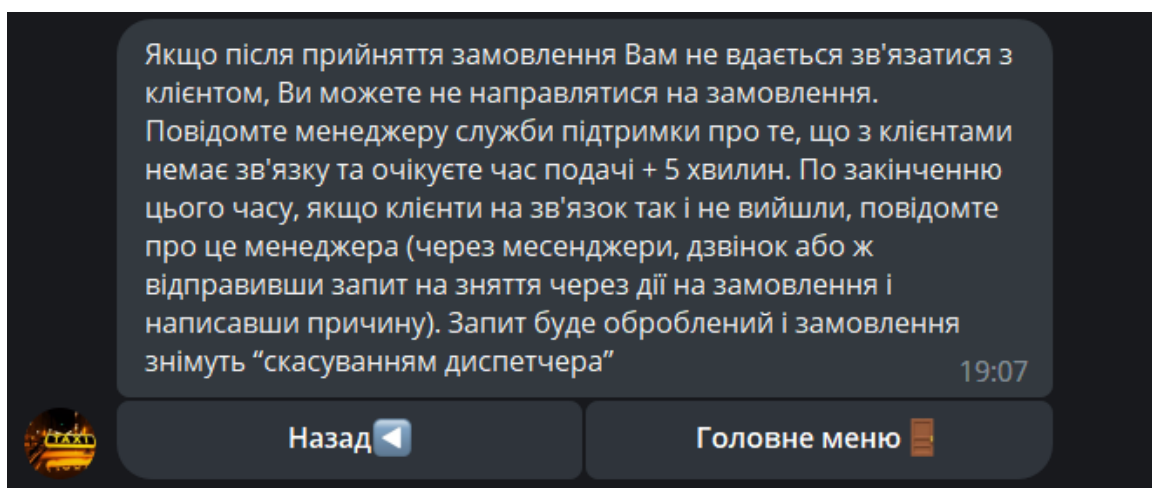


Рисунок 4.6 – Тест 6

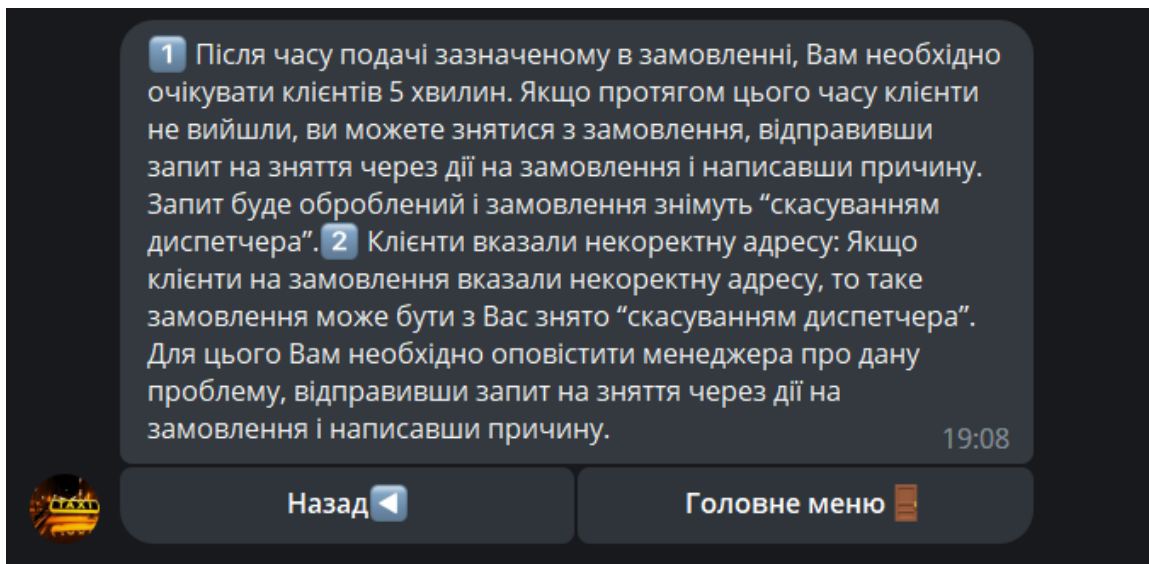


Рисунок 4.7 – Тест 7

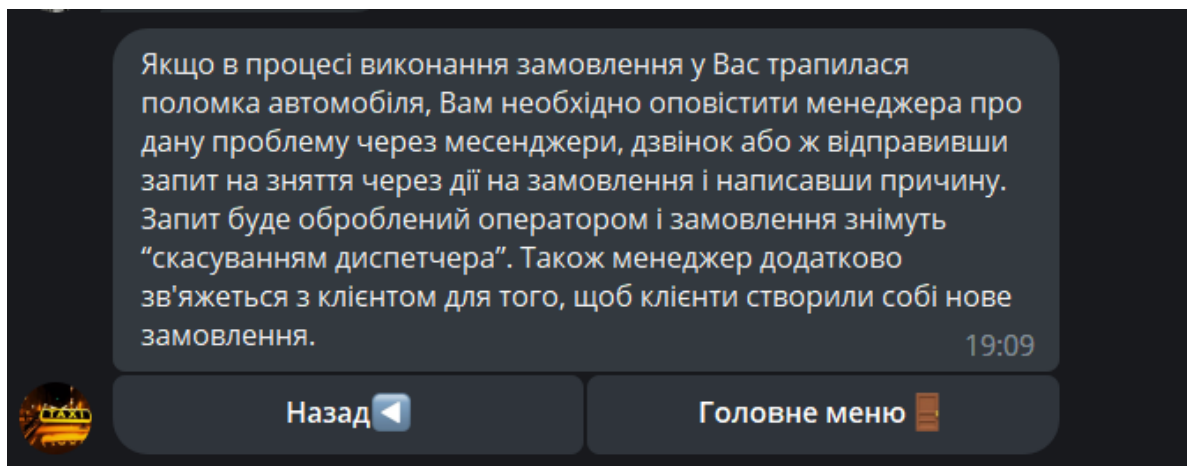


Рисунок 4.8 – Тест 8

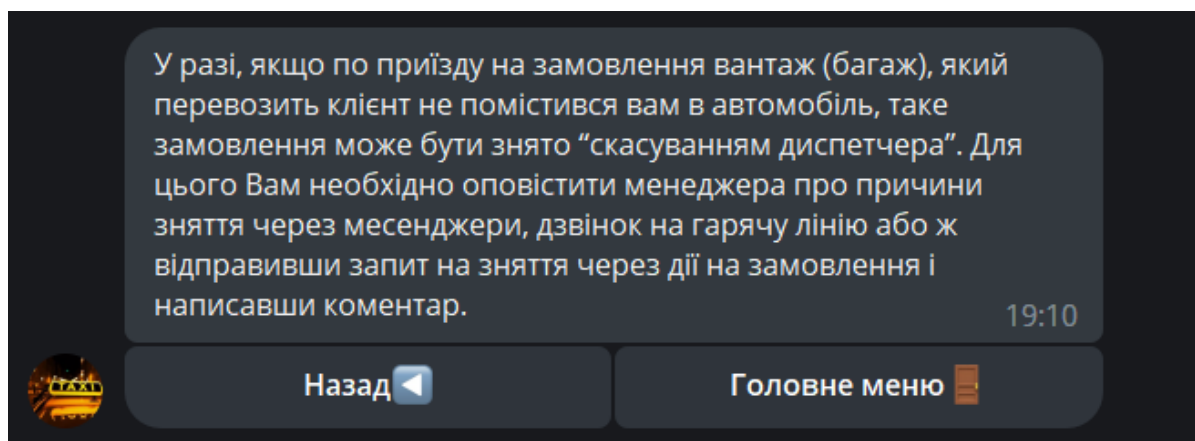


Рисунок 4.9 – Тест 9

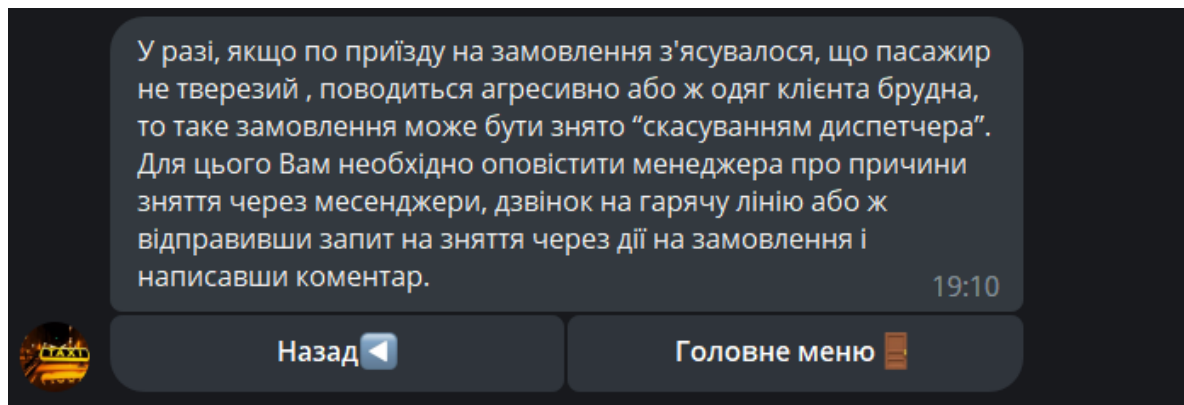


Рисунок 4.10 – Тест 10

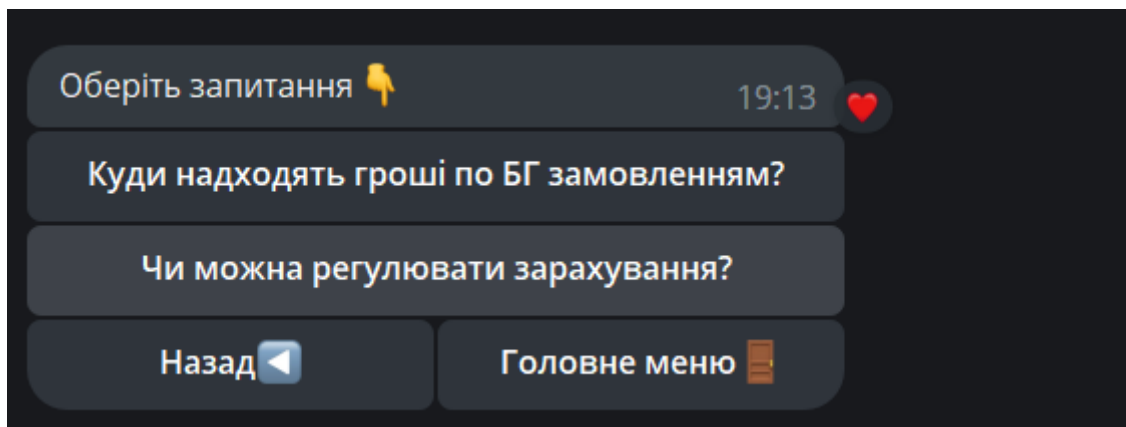


Рисунок 4.11 – Тест 11

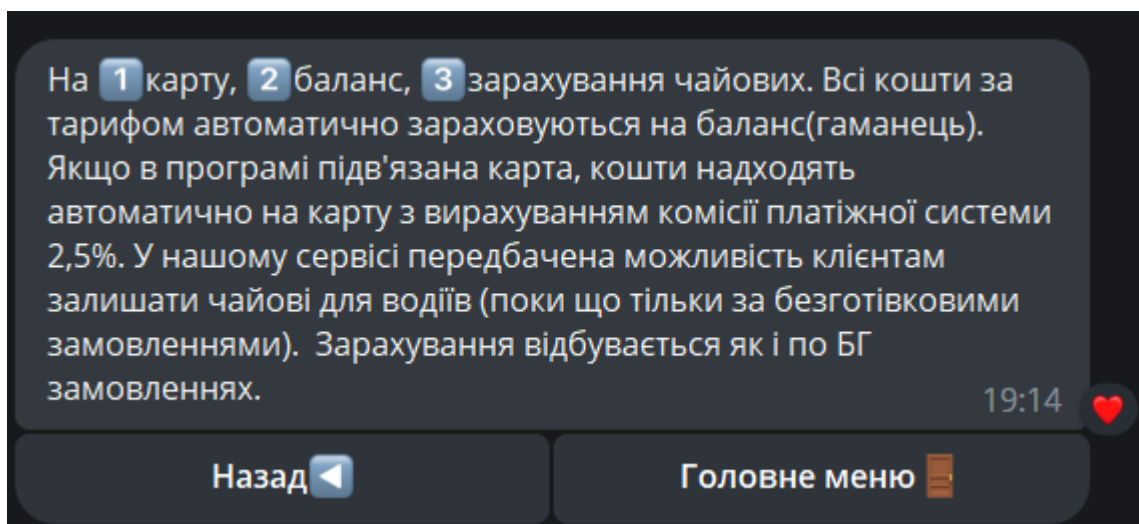


Рисунок 4.12 – Тест 12

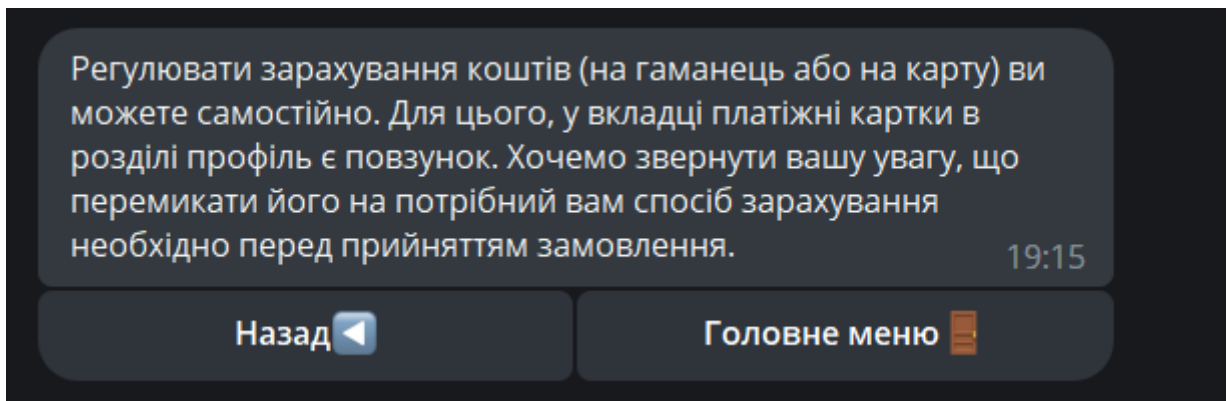


Рисунок 4.14 – Тест 14

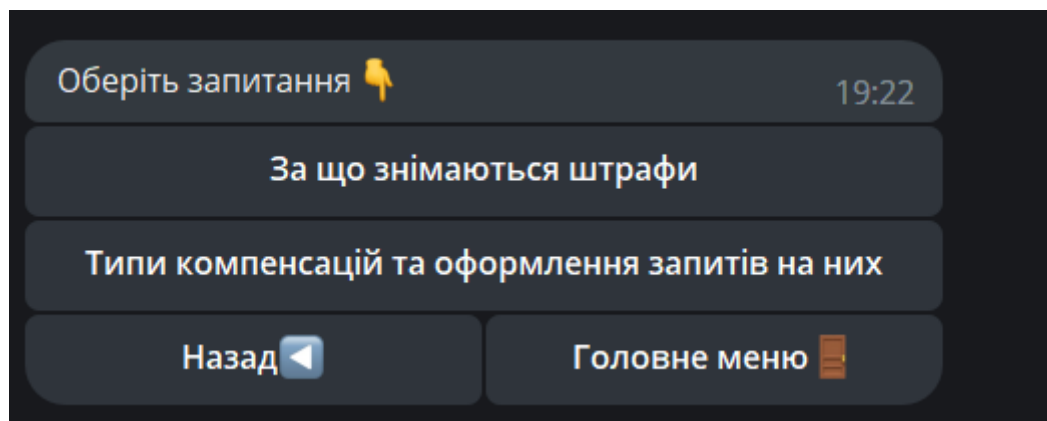


Рисунок 4.15- Тест 15

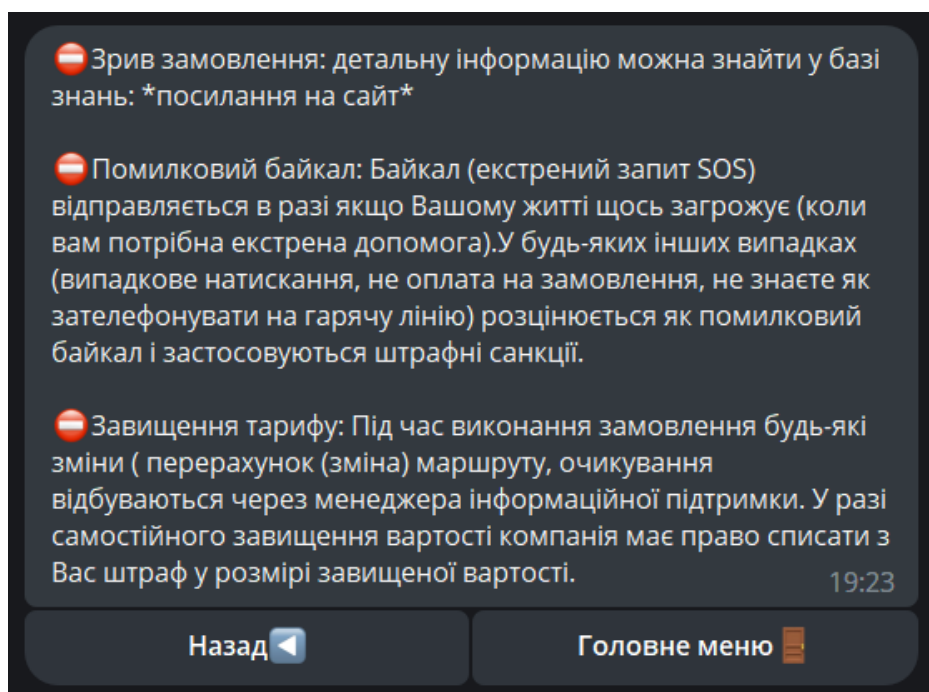


Рисунок 4.16 – Тест 16

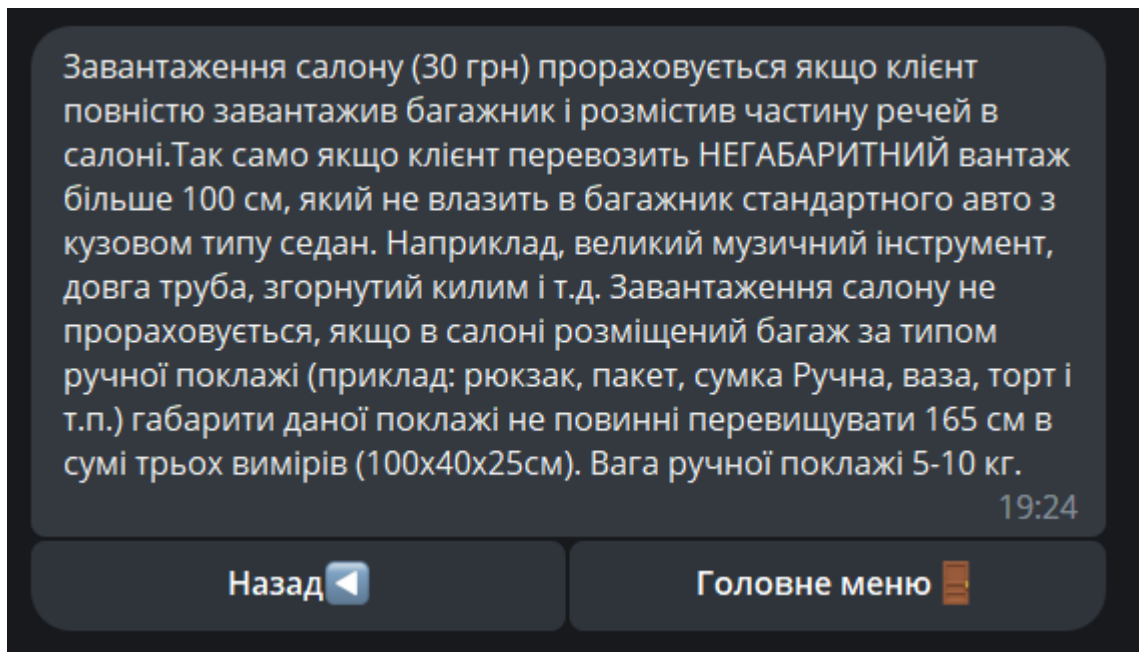


Рисунок 4.17 – Тест 17

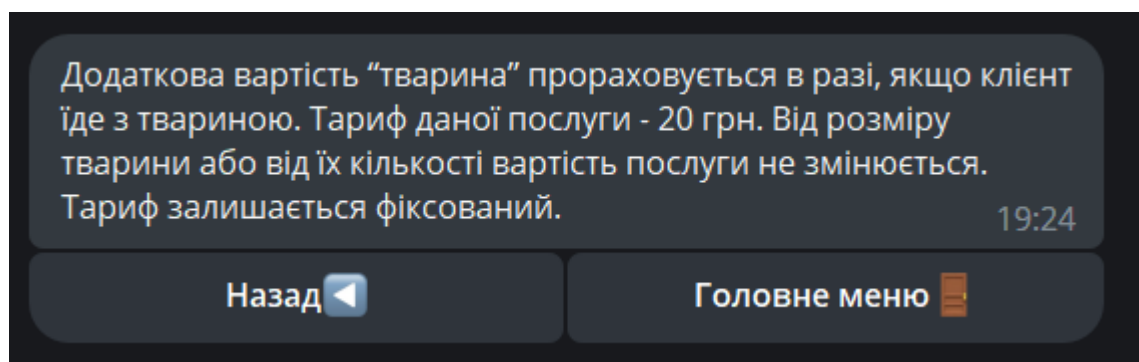


Рисунок 4.18 – Тест 18

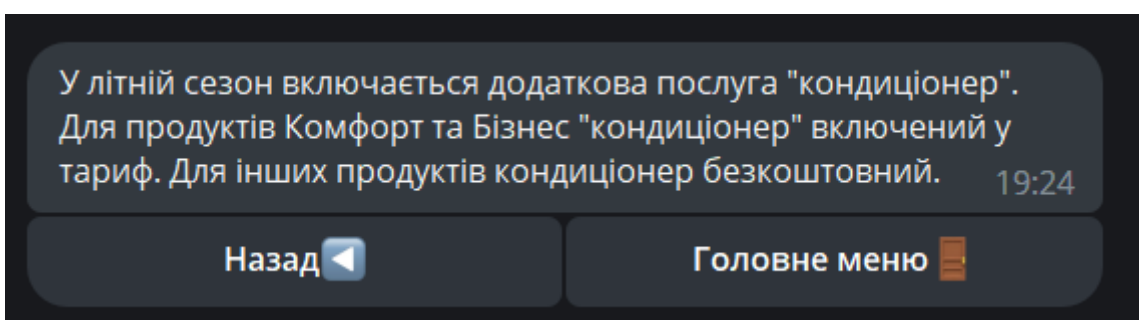


Рисунок 4.19 – Тест 19

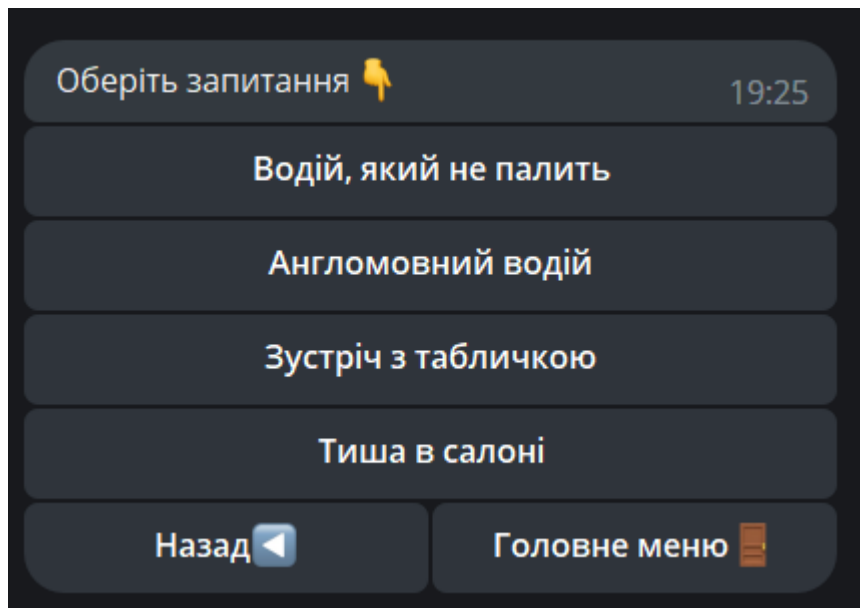


Рисунок 4.20 – Тест 20

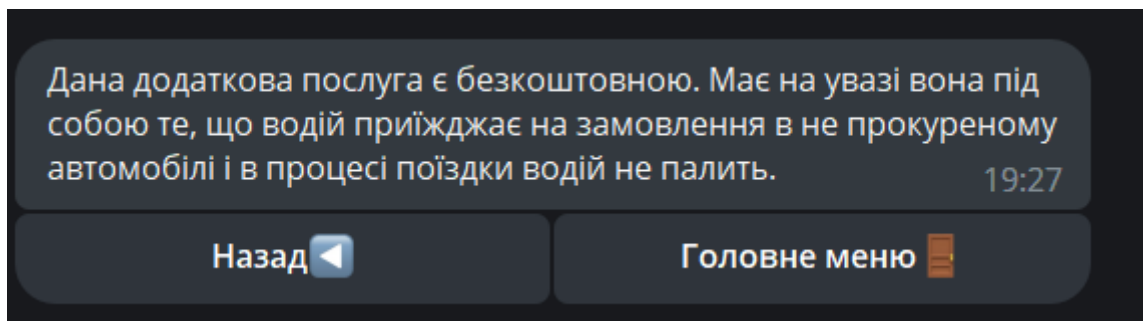


Рисунок 4.21 – Тест 21

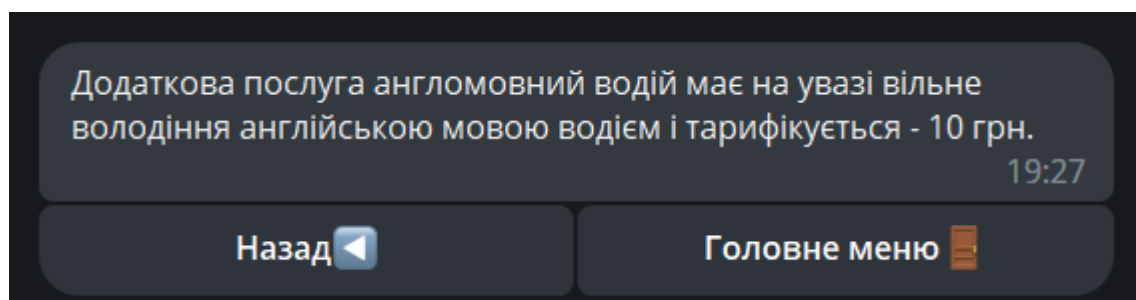


Рисунок 4.22 – Тест 22

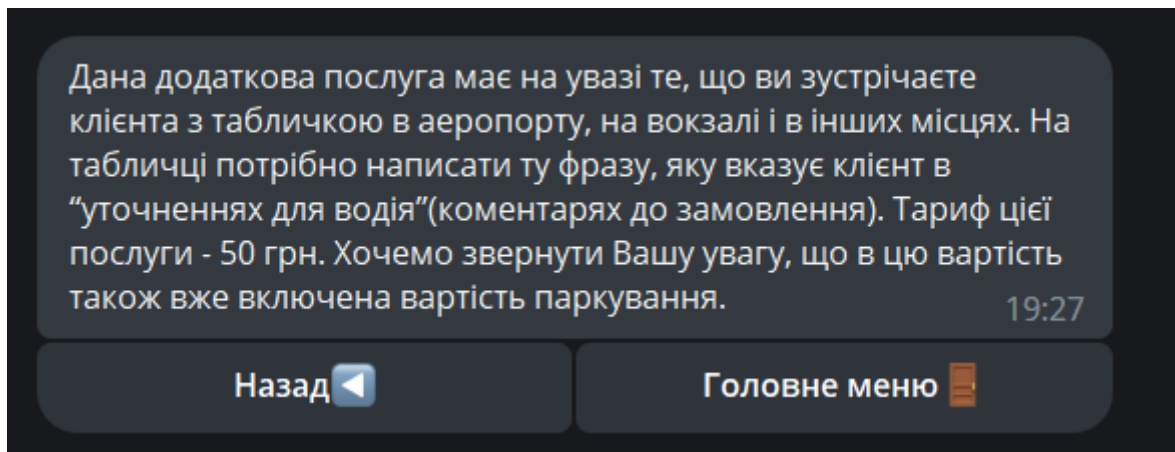


Рисунок 4.23 – Тест 23

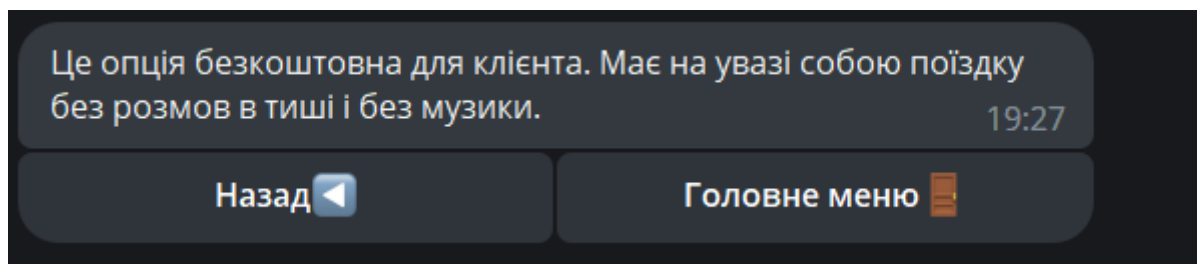


Рисунок 4.24 – Тест 24

В системі кнопка «Оператор» повинна бути прив'язаною до акаунту оператора, чат з яким буде відкриватися при натисканні на кнопку. Для користування нею компанія, що буде користуватися цим додатком, повинна буде створити базу даних операторів з можливим відстеженням зайнятості операторів.

Тестування системи проводилось на різних пристроях, з різними операційними системами, щоб впевнитись, що в незалежності від цього, програма буде працювати правильно і видавати коректні значення на запити користувача.

Висновки до розділу 4

Після проведення тестування програми можна зробити наступні висновки:

1. В результаті тестування було підтверджено, що програма виконує свої основні функції, пов'язані з комунікацією водіїв та сервісу таксі. Вона забезпечує авторизацію водіїв, прийняття та керування замовленнями, навігацію до пункту початку поїздки та інші необхідні функції.
2. Програма продемонструвала стабільну роботу та відсутність критичних помилок або збоїв під час тестування. Вона вдало впоралася з різними сценаріями використання та коректно обробляла дані вводу.

3. Програма успішно виконала встановлені функціональні вимоги. Вона надала необхідну функціональність для водіїв та забезпечила їх зв'язок з сервісом таксі. Всі заплановані функції працюють належним чином.
4. В процесі тестування було виявлено, що програма має зручний і інтуїтивно зрозумілий інтерфейс для водіїв. Навігація та взаємодія з додатком є зручними та ефективними. Це сприяє поліпшенню користувацького досвіду та забезпечує зручність у використанні програми.
6. На основі результатів тестування були запропоновані деякі покращення, а також вирішення виявлених помилок. Також були рекомендовані додаткові функціональні можливості, які поліпшать досвід користувачів.

Загалом, тестування програми підтвердило її працездатність та відповідність вимогам. Виявлені помилки та можливі покращення вказують на те, що програма може бути ще більш ефективною та функціональною з подальшою роботою над її вдосконаленням [1].

## АНАЛІЗ ТА ВИСНОВКИ

В результаті роботи над дипломною роботою було розроблено онлайн чат-бот таксі. Програмний застосунок, що спрямований на поліпшення комунікації водіїв таксі з сервісом таксі. Цей додаток забезпечує зручну та ефективну взаємодію між водіями та сервісом, сприяючи поліпшенню якості та швидкості обслуговування пасажирів.

Було ретельно визначено функціональні вимоги до чат-боту, такі як прийом та обробка замовлень, надсилання повідомлень водіям та пасажирам, керування статусами замовлень та інше. Кожна функція була реалізована з урахуванням потреб та очікувань користувачів.

Програмний застосунок пройшов успішне тестування, під час якого були виявлені та виправлені деякі незначні помилки та недоліки. Це гарантує стабільну та безперебійну роботу застосунку, що є важливим фактором для забезпечення заданих функцій та задоволення потреб користувачів.

Застосунок відповідає вимогам та потребам сучасного ринку таксі в Україні, де популярність електронних платформ для замовлення таксі постійно зростає. Він забезпечує конкурентоспроможність водіїв та їх можливість ефективно конкурувати на ринку таксі.

Використання мови програмування C# для розробки чат-боту для платформи Telegram має значний вплив на процес створення та функціональність програмного рішення. Основні висновки щодо використання C# в цьому контексті включають наступне:

1. **Продуктивність:** C# є мовою програмування, яка забезпечує високу продуктивність та швидкість виконання програмного коду. Це особливо важливо для реалізації чат-боту, оскільки він повинен бути здатний швидко обробляти та відповідати на повідомлення користувачів у реальному часі.
2. **Розширюваність:** C# має широкі можливості для розширення функціональності програми за допомогою сторонніх бібліотек та компонентів. Це дозволяє розробникам легко інтегрувати додаткові функції, такі як обробка платежів, геолокація, бази даних та інші, що поліпшують функціональність та користувацький досвід чат-боту.
3. **Велика спільнота розробників:** C# є однією з найпопулярніших мов програмування, що має велику та активну спільноту розробників. Це означає, що є велика кількість ресурсів, документації, форумів та інших джерел, які

допомагають розробникам вирішувати проблеми та отримувати підтримку під час роботи над проектом.

4. Інтеграція з платформою Telegram: C# має добре підтримувану бібліотеку Telegram.Bot, яка спрощує взаємодію з API Telegram та надає зручний інтерфейс для розробки чат-ботів. Це дозволяє легко створювати та налаштовувати функціональність бота, включаючи обробку повідомлень, команд, відправку зображень, посилань та іншого.

Завдяки вищезгаданим підходам та методам було досягнуто експлуатаційного призначення програмного застосунку. Водії отримали зручний та надійний інструмент для зв'язку із системою таксі для вирішення своїх питань.

## Список використаних джерел

1. M
2. G
3. Y. Wang, H. Li, X. & Lu, Z. (2021). Design and implementation of a chatbot  
Kaval, S. (2019). Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot
- 4.
- 5.
6. do-Stephan, S., Fafalios, P., & Holzinger, A. (2021). Chatbots in recommender
7. ng, P. (2020). Python Machine Learning By Example: The easiest way to get into
8. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. Pearson.
- 9.
- 10.
11. Аналіз існуючих рішень для чат-ботів у транспортній сфері"
- 12.
13. B. Pal, R., & Tripathi, S. (2019). A review on natural language processing  
Використання чат-ботів для покращення обслуговування водіїв таксі
14. Телеграм-чатботи: огляд платформи та функцій"
15. Розробка чат-бота для Facebook Messenger" (<https://habr.com/ru/post/404395/>)

16. "Розробка чат-бота для Viber" (<https://habr.com/ru/post/418919/>)

17."

18. "Технології розробки чат-ботів на основі природньої мови"

а

в

ч

19."

а

20. "Методи та алгоритми розробки чат-ботів"

а

а

я

21. "Використання машинного навчання в розробці чат-ботів"

л

і

ч

22. "Розробка чат-ботів на основі Telegram API"

а

т

23. "Процес розробки та впровадження чат-ботів у бізнесі"

і

н

о

с

о

т

т

р

і

у

в

м

е

н

н

а

т

і

о

в

с

н

р

о

о

в

з

і

р

о

м

б

б

## Додатки

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Проректор Українського державного  
університету науки і технологій

Анатолій РАДКЕВИЧ

18.02.22

### МОБІЛЬНИЙ ДОДАТОК “ЧАТ-БОТ ТАКСІ”

Технічне завдання

ЛИСТ ЗАТВЕРДЖЕННЯ

44165850.01223-01-ЛЗ

Представники

підприємства-розробника

Завідувач кафедри КІТ

Вадим ГОРЯЧКІН

18.06.23

Керівник розробки

Іван КЛИМЕНКО

18.06.23

Виконавець

Кирило ЗАВЕРТКІН

18.06.23

Норм-контролер

Олена КУРОП'ЯТНИК

18.06.23

ЗАТВЕРДЖЕНО

44165850.01223-01-ЛЗ

**МОБІЛЬНИЙ ДОДАТОК “ ЧАТ-БОТ ТАКСІ ”**

Технічне завдання

44165850.01223-01

Листів 16

2023

## ЗМІСТ

Додатки	50	
1	ВВЕДЕННЯ	53
2	ПІДСТАВИ ДЛЯ РОЗРОБКИ	54
3	ПРИЗНАЧЕННЯ РОЗРОБКИ	55
4	ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	56
4.1	Вимоги до функціональних характеристик	56
4.2	Вимоги до надійності	57
4.3	Вимоги експлуатації	57
4.4	Вимоги до складу та параметрів технічних засобів	59
4.5	Вимоги до інформаційної та програмної сумісності	59
4.6	Вимоги до маркування і упаковки	59
4.7	Вимоги до транспортування та зберігання	60
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	61
6	СТАДІЇ ТА ЕТАПИ РОЗРОБКИ	62
7	ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ	63
8	БІБЛЮГРАФІЧНИЙ СПИСОК	64

## 1 ВВЕДЕННЯ

Мобільний додаток «Чат-бот таксі», що розробляється, має виконувати функції боту для відповіді на питання користувачів (водіїв таксі) для полегшення їх роботи.

В залежності від вибраних користувачем питань або функції системи - мобільний додаток має генерувати різні відповіді на питання, які націлені на вирішення проблеми користувача.

Користувач, використовуючи цей мобільний додаток, може обрати питання із списку запропонованих та, за своїм бажанням, може написати адміністратору служби таксі. Після отримання відповіді на поставлене питання, користувач може повернутись на початковий екран або розпочати пошук відповіді на свою проблему з самого початку, зберігаючи усі вибрані до цього запити.

Програмний продукт є безкоштовний, що робить його більш цікавим на фоні інших аналогів, але варто розуміти, що це рекламний продукт.

## 2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є наказ від 08.12.21 №77ст ректора Українського державного університету науки і технологій “Про призначення наукових керівників та затвердження тем бакалаврських робіт” за спеціальністю 121 “Інженерія програмного забезпечення» факультету “Комп’ютерних технологій і систем” по кафедрі “Комп’ютерні інформаційні технології”.

Тема дипломної роботи - “ЧАТ-БОТ ТАКСІ”. Керівник - доцент Клименко І. В.

### 3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення – програмний продукт має генерувати відповіді на питання водіїв таксі для отримання оперативної відповіді на питання, пов'язані з роботою, такими як оптимальні маршрути, тарифи, дорожні обставини тощо.

Експлуатаційне призначення – додаток розроблений з метою надання зручного та швидкого способу отримання необхідної інформації для водіїв таксі під час їх роботи. Він призначений для полегшення комунікації та підтримки водіїв, допомагаючи їм вирішувати питання, що виникають під час перевезення пасажирів.

## 4 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

### 4.1 Вимоги до функціональних характеристик

- Питання та відповіді: Додаток повинен надавати можливість водіям задавати запитання через чат-бота та отримувати оперативні та точні відповіді на них.
- Інформація про маршрути: Додаток має забезпечувати доступ до інформації про оптимальні маршрути, дорожні обставини та будь-які інші важливі дані, що допоможуть водіям при плануванні та виконанні поїздок.
- Інформація про тарифи: Додаток повинен надавати водіям доступ до інформації про тарифи, збори та інші витрати, пов'язані з таксі. Це дозволить водіям зручно розраховувати вартість поїздок та вести облік фінансових аспектів.
- Система оцінювання: Додаток повинен мати можливість збирати та відображати відгуки та оцінки від пасажирів про роботу водіїв. Це дозволить водіям вдосконалювати свою роботу та підтримувати високу якість обслуговування.

Вимоги до вхідних даних:

- Запитання користувача: Додаток повинен забезпечити можливість вводу запитань від водіїв таксі. Вхідні дані повинні бути зрозумілими та коректними.
- Маршрутні дані: Додаток може вимагати введення даних про поточне місцезнаходження водія, початкову та кінцеву точки маршруту, якщо це потрібно для надання оптимальних маршрутних рекомендацій.
- Питання та коментарі: Користувачі можуть вводити будь-які питання, коментарі або додаткову інформацію, що стосується їх запитань або проблем, пов'язаних з роботою в таксі.

Вхідні дані повинні бути зрозумілими, правильними та відповідати вимогам, що ставляться до функціональності та цілей додатку чат-боту таксі.

Вимоги до вихідних даних:

Вихідними даними є:

- Відповіді на запитання: Додаток повинен надавати чіткі та зрозумілі відповіді на запитання водіїв таксі. Вихідні дані повинні бути лаконічними, повними та інформативними.
- Інформація про маршрути: Додаток має надавати інформацію про оптимальні маршрути, дорожні обставини та інші важливі дані, необхідні водіям для успішного виконання поїздок. Вихідні дані повинні бути точними та оновлюватися в режимі реального часу.
- Інформація про тарифи: Додаток повинен надавати водіям інформацію про тарифи, збори та інші фінансові аспекти, пов'язані з таксі. Вихідні дані повинні бути чіткими та зрозумілими, допомагати водіям розраховувати вартість поїздок та уникати непорозумінь.

#### 4.2 Вимоги до надійності

Вимоги до надійності наступні:

- при оновленні даних забезпечити показ відповідних повідомлень про стан роботи програми та результати виконання операцій;
- наявність архівної копії тексту програми на зовнішньому носії;
- наявність резервної копії бази даних на зовнішньому носії.

#### 4.3 Вимоги експлуатації

Програмний продукт повинен використовуватись у приміщеннях які відповідають умовам роботи ЕОМ, а саме мають такі кліматичні, санітарні та гігієнічні умови, які відповідають ДНАОП 0.00-1.13-99 (див. табл. 1).

Таблиця 1. Кліматичні умови

Пора року	Категорія робіт згідно з ГОСТ 12.01-005-88	Температура повітря, град.С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
		Оптимальна	Оптимальна	Оптимальна
Холодна	легка-1-а	22-24	40-60	0,1
	легка-1-б	21-23	40-60	0,1
Тепла	легка-1-а	23-25	40-60	0,1
	легка-1-б	22-24	40-60	0,2

Працювати з програмою може людина, що має навички роботи з мобільними пристроями та ознайомена з керівництвом користувача програмного продукту.

#### 4.4 Вимоги до складу та параметрів технічних засобів

Продукт, що розробляється повинен використовуватись на мобільних пристроях, що мають наступні характеристики:

- діагональ екрану – 5.5;
- роздільна здатність дисплею – HD (1280x720);
- оперативна пам'ять – 2 ГБ;
- вбудована пам'ять – 16 ГБ;
- операційна система – ANDROID;
- частота процесора – 1.6 ГГц;
- комунікації – microUSB.

#### 4.5 Вимоги до інформаційної та програмної сумісності

Програмний продукт розробляється для всіх видів операційних систем сімейства “ANDROID” починаючи від версії 6 та наступні версії.

#### 4.6 Вимоги до маркування і упаковки

Упаковка програмного продукту, включаючи документацію повинна бути захищена від пошкоджень різного роду (механічних, кліматичних).

На упаковці повинно бути вказана назва продукту, номер версії, мінімальні системні вимоги.

На зворотній стороні упаковки вказується розробник та його юридична адреса.

<p>Програмний продукт <b>МОБІЛЬНИЙ ДОДАТОК</b> <b>«ЧАТ-БОТ ТАКСІ»</b></p> <ul style="list-style-type: none"><li>• діагональ екрану – 5.5;</li><li>• роздільна здатність дисплею – HD (1280x720);</li><li>• оперативна пам'ять – 2 ГБ;</li><li>• вбудована пам'ять – 16 ГБ;</li><li>• операційна система – ANDROID;</li><li>• частота процесора – 1.6 ГГц;</li><li>• комунікації – microUSB.</li></ul>	<p>Програмний продукт <b>МОБІЛЬНИЙ ДОДАТОК</b> <b>«ЧАТ-БОТ ТАКСІ»</b></p> <p><b>Розробник: Заверткін К. Ю.</b> <b>Кафедра «КІТ», УДУНТ</b></p> <p><b>м. Дніпро, вул. Лазаряна 2</b></p> <p><b>2023</b></p>
---	--

#### 4.7 Вимоги до транспортування та зберігання

Транспортування повинне забезпечувати збереження програмного продукту його цілісність і запобігання несанкціонованого доступу до нього. Програмний виріб міститься на фізичному носії та переданий через microUSB порт.

## 5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документації мають входити:

- специфікація;
- текст програми;
- опис програми;
- керівництво користувача.

Вся документація програмного додатку повинна задовольняти вимоги до програмної документації.

## 6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Таблиці 1. – Стадії та етапи розробки

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, вибір та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	31.01.23 - 18.06.23
Робочий проект	Програмування та відлагодження програми.	19.02.23 - 20.05.23
	Тестування програми	20.05.23 - 27.05.23
	Розробка, узгодження і затвердження програмної документації.	27.05.23 - 12.06.23

## 7 ПОРЯДОК І КОНТРОЛЬ ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник розробки доц.  
Клименко І. В.

Прийом здійснюється комісією у складі:

- Горячкін В. М. (керівник підрозділу);
- Клименко І. В. (керівник розробки).

## 8 БІБЛОГРАФІЧНИЙ СПИСОК

1. Івченко, Ю.М. Основи стандартизації програмних систем: методичні вказівки до дипломного проектування та лабораторних робіт/уклад.: Ю.М. Івченко, В. І. Шинкаренко, В. Г. Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В. Лазаряна, 2009. - 38 с

```

using CSharpTaxiChatBot.Loggers;
using CSharpTaxiChatBot.MenuInterfaces;
using Telegram.Bot;
using Telegram.Bot.Types;
using Telegram.Bot.Types.Enums;
using Telegram.Bot.Types.ReplyMarkups;

namespace CSharpTaxiChatBot.Bot
{
    //Чат-бот
    internal class Bot
    {
        //Клієнт боту
        private readonly TelegramBotClient _client;

        //Головне меню боту
        private readonly InlineKeyboardMarkup _keyboardMarkupMainMenu;

        //Параметричний конструктор
        public Bot(string token)
        {
            _keyboardMarkupMainMenu = new InlineKeyboardMarkup(new[]
            {
                new[]
                {
                    InlineKeyboardButton.WithUrl("☎ Оператор", "tg://resolve?domain=rest17921") //Замість
                    TELEGRAM_USERNAME ввести ім'я акаунту оператора в телеграмі без @
                },
                new[]
                {
                    InlineKeyboardButton.WithCallbackData(" ? Часті питання", "button_faq")
                }
            });

            //Виділення пам'яті під клієнт боту
            _client = new TelegramBotClient(token);
        }
    }
}

```

```
//Запуск боту
_client.StartReceiving(Logic, Error);
```

```
//Бот працює, доки працює сервер
Console.ReadLine();
```

```
//Асинхронний метод виведення головного меню програми
message - отримане повідомлення
botClient - клієнт боту
cancellationToken - токен для відміни дії, якщо вона довго обробляється
private async Task MainMenu(Message message, ITelegramBotClient botClient, CancellationToken cancellationToken)
{
    string userName = message.From.FirstName + "\t" + message.From.LastName;
    string helloMessage = $"Вітаємо {userName}!\n" +
        $"Ви потрапили до контакт-центру нашої служби такі 🌀 🌀 🌀\n\n" +
        $"Підключайте оператора або шукайте відповідь на своє запитання у розділі 'Часті питання' 🗨 ";

    await botClient.SendTextMessageAsync(message.Chat.Id, helloMessage, replyMarkup: _keyboardMarkupMainMenu,
        cancellationToken: cancellationToken);
```

```
//Асинхронний метод виведення помилки у програмі(якщо спамити боту стікерами, словами та іншим - то
виводиться передана фраза)
chatId - айді чату з користувачем
unknownCommandMessage - повідомлення з помилкою
botClient - клієнт боту
cancellationToken - токен для відміни дії, якщо вона довго обробляється
private static async Task UnknownCommandMessage(long chatId, string unknownCommandMessage, ITelegramBotClient
botClient, CancellationToken cancellationToken)
n {
    await botClient.SendTextMessageAsync(chatId, unknownCommandMessage, cancellationToken: cancellationToken);

    return;
}

//Логіка боту
```

```

//botClient - клієнт боту

cancellationToken - токен для відміни дії, якщо вона довго обробляється

private async Task Logic(ITelegramBotClient botClient, Update update, CancellationToken cancellationToken)
{
    //Отримання повідомлення від користувача
    update - нові повідомлення(кліки на кнопки) від користувача
    var message = update.Message;

    //Це точно повідомлення та саме текст?
    if (message != null && message.Type == MessageType.Text)
    {
        //Запуск
        if (message.Text.Equals("/start"))
        {
            //Вивід головного меню
            await MainMenu(message, botClient, cancellationToken: cancellationToken);
        }
        else
        {
            //Вивід помилки, про невідому команду
            string unknownCommandMessage = "Невідома команда 🤖 😊";
            await UnknownCommandMessage(message.Chat.Id, unknownCommandMessage, botClient, cancellationToken:
            cancellationToken);
        }
        else // - якщо це не повідомлення, а клік

            //Отримуємо клік
            var
            r
            c
            a
            l
            l
            b
            a
            c
            k
            e
            d
            C
            l
            i
            c
            k
            e
            d
            M
            e
            s
            s
            a
            g
            e
            =
            u
            p
            d
            a
            t
            e
            .
            M
            e
            s
            s
            a
            g
            e
            .
            C
            l
            i
            c
            k
            e
            d
            M
            e
            s
            s
            a
            g
            e
            ;

            //Якщо це саме клік

            //Видаляємо попереднє повідомлення

            //Текст для обирання запитання
            questionMessage = "Оберіть запитання 🗣️";

```

```

//Відповідь на запитання(спочатку пуста, потім міняється в залежності від запитання)

s
t
r
i
n
g
    //Отримуємо кнопку, на яку був зроблений клік
    //Кнопка "Часті питання"
case "button_faq":
    {
        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetMainQuestions(), cancellationToken: cancellationToken);
    }

a
s
k
Q
u
e
r
y
D
a
t
a
    //----- 1 блок запитань
    //Кнопка "Прийняття/скасування замовлення"
case "button_getOrCancelOrder":
    {
        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetOrCancelOrderQuestions(), cancellationToken: cancellationToken);

        //##### Блок запитань "Прийняття/скасування замовлення"

        //Кнопка "Способи прийняття замовлень"

c
a
s
e
    answer = "Детальну інформацію можна отримати у базі знань нашої служби таксі: \n*посилання на сторінку сайту*";
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswersInterface.GetOrCancelOrderAnswer(), cancellationToken: cancellationToken);
n
h
o
w
T
o
G
e
t
O
r
d

```

//Кнопка "Принципи розподілу замовлень"

c  
a  
s  
e  
b  
u  
t  
n  
p  
r  
i  
n  
c  
i  
p  
e  
O  
f  
i  
s  
t  
r  
i  
b  
u  
t  
i  
o  
n  
s  
O  
f  
f  
e  
r  
a  
t  
i  
o  
n  
s  
p  
r  
o  
p  
o  
s  
e  
r  
i  
e  
s  
o  
f  
c  
o  
n  
t  
e  
n  
t  
s  
o  
n  
t  
h  
e  
p  
a  
g  
e

answer = "Детальну інформацію можна отримати у базі знань нашої служби такі: \n\*посилання на сторінку сайту\*";

await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetOrCancelOrderAnswer(), cancellationToken: cancellationToken);

//Кнопка "Як скасувати замовлення?"

case "button\_howToCancelOrder":

{

await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:  
QuestionsInterface.GetHowToCancelOrderQuestions(), cancellationToken: cancellationToken);

\$ Блок запитань "Як скасувати замовлення?"

//Кнопка "Клієнт не виходить на зв'язок"

answer = "Якщо після прийняття замовлення Вам не вдається зв'язатися з клієнтом, Ви можете не направлятися на замовлення." +

" Повідомте менеджеру служби підтримки про те, що з клієнтами немає зв'язку та очікуєте час подачі + 5 хвилин." +

" По закінченню цього часу, якщо клієнти на зв'язок так і не вийшли, повідомте про це менеджера через месенджери, дзвінок або ж відправивши запит на зняття через дії на замовлення і написавши причину)." +

" Запит буде оброблений і замовлення знімуть "скасуванням диспетчера";

//Кнопка "Клієнт не виходить до авто"

Після часу подачі зазначеному в замовленні, Вам необхідно очікувати клієнтів 5 хвилин." +

" Якщо протягом цього часу клієнти не вийшли, ви можете знятися з замовлення, відправивши запит на зняття через дії на замовлення і написавши причину." +

" Запит буде оброблений і замовлення знімуть “скасуванням диспетчера”." +

Клієнти вказали некоректну адресу:" +

" Якщо клієнти на замовлення вказали некоректну адресу, то таке замовлення може бути з Вас знято “скасуванням диспетчера”." +

" Для цього Вам необхідно оповістити менеджера про дану проблему, відправивши запит на зняття через дії на замовлення і написавши причину.";

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetHowToCloseOrderAnswer(), cancellationToken: cancellationToken);
```

//Кнопка "Поломка авто в процесі поїздки з клієнтами"

```
с
а
s
е
b
answer = "Якщо в процесі виконання замовлення у Вас трапилася поломка автомобіля," +
u
t
t
o
n
с
а
r
В
r
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetHowToCloseOrderAnswer(), cancellationToken: cancellationToken);
е
n
W
і
t
h
C
l
c
і
a
e
n
с
b
і
a
k
t
o
n
l
u
g
g
e
D
o
e
s
N
o
t
```

" Вам необхідно оповістити менеджера про дану проблему через месенджери," +  
" дзвінок або ж відправивши запит на зняття через дії на замовлення і написавши причину." +  
" Запит буде оброблений оператором і замовлення знімуть “скасуванням диспетчера”." +  
" Також менеджер додатково зв'яжеться з клієнтом для того, щоб клієнти створили собі нове замовлення.";

```
В
r
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetHowToCloseOrderAnswer(), cancellationToken: cancellationToken);
```

//Кнопка "Не поміщається багаж(вантаж)"

```
l
c
і
a
e
n
с
b
і
a
k
t
o
n
l
u
g
g
e
D
o
e
s
N
o
t
```

"У разі, якщо по приїзду на замовлення вантаж (багаж)," +  
" який перевозить клієнт не помістився вам в автомобіль, таке замовлення може бути знято “скасуванням диспетчера”." +  
" Для цього Вам необхідно оповістити менеджера про причини зняття через месенджери," +  
" дзвінок на гарячу лінію або ж відправивши запит на зняття через дії на замовлення і написавши коментар.";

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetHowToCloseOrderAnswer(), cancellationToken: cancellationToken);
```



```

//Кнопка "Назад", для повернення до питань "Прийняття/скасування замовлень"
case "button_goBackToHowToCancelOrder":
    {
        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetHowToCancelOrderQuestions(), cancellationToken: cancellationToken);

//##### Кінець блоку запитань "Прийняття/скасування замовлення"

//Кнопка "Назад", для того, щоб повернутись до меню "Прийняття/скасування замовлення"
case "button_goBackToGetOrCancelOrder":
    {
        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetOrCancelOrderQuestions(), cancellationToken: cancellationToken);

//----- кінець 1 блоку запитань

//----- 2 блок запитань

//Кнопка "Оплата безготівкових замовлень"
b
a
{
e
a
k
QuestionsInterface.GetTerminalPaymentQuestions(), cancellationToken: cancellationToken);
u
t
t
o
n
p
a
//Кнопка "Куди надходять зарахування по БГ замовленням?"
y
m
a
в
e
answer = "На 1 карту, 2 баланс, 3 зарахування чайових." +
B
" Всі кошти за тарифом автоматично зараховуються на баланс(гаманець)." +
F
" Якщо в програмі підв'язана карта, кошти надходять автоматично на карту з вирахуванням комісії
€
платіжної системи 2,5%." +
D
D

```



```
//Кнопка "За що знімаються штрафи?"
```

с  
а  
с  
е  
а  
b  
n  
y  
t  
e  
o  
g  
w  
h  
z  
r  
f  
a  
m  
g  
b  
n  
n  
я  
:  
д  
с  
а  
т  
а  
с  
е  
в  
н  
t  
y  
r  
н  
ф  
о  
р  
м  
а  
ц  
і  
я  
t  
i  
n  
o  
ж  
т  
н  
у  
н  
с  
ю  
с  
в  
а  
т  
н  
е  
ч  
а  
н  
к

```
● Помилковий байкал: Байкал (екстрений запит SOS) відправляється в разі якщо Вашому житті щось загрожує (коли вам потрібна екстрена допомога)." +
```

```
● "У будь-яких інших випадках (випадкове натискання, не оплата на замовлення, не знаєте як зателефонувати на гарячу лінію)" +
```

```
● Завищення тарифу: Під час виконання замовлення будь-які зміни ( перерахунок (зміна) маршруту, очікування відбуваються через менеджера інформаційної підтримки." +
```

```
● " У разі самостійного завищення вартості компанія має право списати з Вас штраф у розмірі завищеної вартості.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetFinesQuestions(), cancellationToken: cancellationToken);
```

```
//Кнопка "Типи компенсацій та оформлення запитів на них"
```

```
answer = "Детальну інформацію можна знайти у базі знань: *посилання на сторінку сайту*";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetFinesQuestions(), cancellationToken: cancellationToken);
```

```
//Кнопка "Назад", для того, щоб повернутись до питань меню "Штрафи/коспенсації"
```

```
case "button_goBackToFinesQuestions":
```

```
{
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:  
QuestionsInterface.GetFinesQuestions(), cancellationToken: cancellationToken);
```

```
//----- кінець 3 блоку запитань
```

у  
к  
б  
я

```
//----- 4 блок запитань
```

```
//Кнопка "Тарифи/додаткові послуги"
```

```
case "button_ratesAdditionalServices":
```

```
{
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup: QuestionsInterface.GetMainAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
    break;
```

```
}
```

```
//Кнопка "Тарифи і комісія"
```

```
case "button_tariffsAndCommission":
```

```
{
```

```
    answer = "Актуальні тарифи: *посилання на сторінку сайту*," +  
            " Комісія: *посилання на сторінку сайту*";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup: AnswerInterface.GetMainAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
//Кнопка "Додатковий багаж в салоні"
```

```
с  
а  
с  
е  
b answer = "Завантаження салону (30 грн) прораховується якщо клієнт повністю завантажив багажник" +  
u " і розмістив частину речей в салоні.Так само якщо клієнт перевозить НЕГАБАРИТНИЙ вантаж більше  
t 100 см," +  
o " який не влезить в багажник стандартного авто з кузовом типу седан." +  
n " Наприклад, великий музичний інструмент, довга труба, згорнутий килим і т.д." +  
a " Завантаження салону не прораховується, якщо в салоні розміщений багаж за типом ручної  
d поклажі" +  
i " (приклад: рюкзак, пакет, сумка Ручна, ваза, торт і т.п.)" +  
t " габарити даної поклажі не повинні перевищувати 165 см в сумі трьох вимірів (100x40x25см)." +  
i " Вага ручної поклажі 5-10 кг.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup: AnswerInterface.GetMainAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
g
```

```
a
```

```
g
```

```
e
```

```
p
```

```
h
```

```
т
```

```
h
```

```
l
```

```

        break;
    }

    //Кнопка "Їду з твариною"
    case "button_driveWithAnimal":
    {
        answer = "Додаткова вартість "тварина" прораховується в разі, якщо клієнт їде з твариною." +
            " Тариф даної послуги - 20 грн." +

" Тариф залишається фіксований.";

        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetMainAdditionalServicesQuestions(), cancellationToken: cancellationToken);

        break;
    }

    //Кнопка "Кондиціонер"
    case "button_aitConditioner":

answer = "У літній сезон включається додаткова послуга \"кондиціонер\"." +
        " Для продуктів Комфорт та Бізнес \"кондиціонер\" включений у тариф." +
        " Для інших продуктів кондиціонер безкоштовний.";

a
w
a
:
b
t
b
e
o
a
t
k
C
l
i
Від розміру тварини або від їх кількості вартість послуги не змінюється." +
#Кнопка "Інші додаткові послуги"
t
S
case "button_anotherAdditionalServices":
e
{
n
d
        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetAnotherAdditionalServices(), cancellationToken: cancellationToken);
e
x
t
M
e
s
s
a
g
e
A
s
s
y
n
c
        //Кнопка "Водій, який не палить"

```

```
answer = "Дана додаткова послуга є безкоштовною." +  
        " Має на увазі вона під собою те, що водій приїжджає на замовлення в не прокуреному автомобілі"  
  
        " і в процесі поїздки водій не палить.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
        break;  
    }  
    //Кнопка "Англомовний водій"  
    case "button_engDriver":
```

```
answer = "Додаткова послуга англомовний водій має на увазі" +  
        " вільне володіння англійською мовою водієм і тарифікується - 10 грн.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
        break;  
    }  
    //Кнопка "Зустріч з табличкою"  
    case "button_meetingWithASign":
```

```
    {  
        answer = "Дана додаткова послуга має на увазі те," +  
                " що ви зустрічаєте клієнта з табличкою в аеропорту, на вокзалі і в інших місцях." +  
                " На табличці потрібно написати ту фразу, яку вказує клієнт в “уточненнях для водія”(коментарях до  
замовлення)." +  
                " Тариф цієї послуги - 50 грн. Хочемо звернути Вашу увагу, що в цю вартість також вже включена  
вартість паркування.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
    //Кнопка "Тиша в салоні"
```

```
s  
a  
s  
e  
b  
u  
t  
t
```

= "Це опція безкоштовна для клієнта." +

" Має на увазі собою поїздку без розмов в тиші і без музики.";

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetAnotherAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

//Кнопка "Назад", для повернення до інших додаткових послуг у меню "Тарифи/додаткові послуги"

case "button\_goBackToAnotherAdditionalServicesQuestions":

{

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetAnotherAdditionalServices(), cancellationToken: cancellationToken);
```

//Кнопка "Назад", для повернення до іншого меню з запитаннями до головного меню з запитанням у "Тарифи/додаткові послуги"

case "button\_goBackToMainAdditionalServicesQuestions":

{

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetMainAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
b
r
e
a
k
//##### Кінець блоку з іншими додатковими послугами
```

//Кнопка "Назад", для повернення до головного меню запитань у "Тарифи/додаткові послуги"

case "button\_goBackToAdditionalServicesQuestions":

{

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetMainAdditionalServicesQuestions(), cancellationToken: cancellationToken);
```

```
b
r
e
a
k
```

//----- кінець 4 блоку запитань

//----- 5 блок запитань

```
b
r
```

```
//Кнопка "Поповнення балансу"
```

```
case "button_replenishmentOfBalance":
```

```
{
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:  
QuestionsInterface.GetReplenishmentOfBalanceQuestions(), cancellationToken: cancellationToken);
```

```
//Кнопка "Як поповнити баланс в нашому додатку?"
```

```
с  
а  
с  
е  
b  
u  
t  
o  
n  
h  
o  
w
```

```
answer = "Для поповнення балансу через наш додаток" +
```

```
    " Вам необхідно зайти в профіль → вкладка “гаманець” → вгорі в правому куті натискаєте на знак  
долара " +
```

```
    "і вибираєте пункт “Поповнити гаманець”. " +
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetReplenishmentOfBalanceQuestions(), cancellationToken: cancellationToken);
```

```
R  
e  
p  
l  
e  
n  
i
```

```
//Кнопка "Що робити, якщо поповнив не свій позивний?"
```

```
s  
h  
i  
f  
t  
h  
e  
n  
i  
f  
t  
h  
e  
n  
e  
l  
s  
e
```

```
answer = "У разі, якщо ви поповнили не свій позивний, " +
```

```
    "Вам необхідно зробити фото чека про поповнення (якщо поповнювалися через термінал) " +
```

```
    "або завантажити виписку про поповнення (якщо поповнення було через додаток) " +
```

```
    "і написати лист на пошту *пошта служби таксі* з описом виниклої проблеми " +
```

```
    "(прописуєте я (ваш позивний) поповнив не той позивний, або не те місто і прикріплюєте фото або  
виписку. " +
```

```
    "У темі так само вказуєте Ваш контактний номер і позивний для ідентифікації. Заявки  
обробляються в робочий час.");
```

```
U  
p  
d  
a  
t  
e
```

```
//Кнопка "Поповнив позивний, а гроші не прийшли"
```

```
answer = "У разі, якщо ви поповнили позивний, кошти з картки списалися, " +  
    "але на позивний не зарахувалися, Вам необхідно звернутися на гарячу лінію банку через " +  
        "який ви поповнювалися для уточнення інформації. Якщо ж з боку банку ніяких неполадок не  
виявлено, " +  
    "рекомендуємо вам зробити фото чека про поповнення (якщо поповнювалися через термінал) " +  
        "або завантажити виписку про поповнення (якщо поповнення було через додаток) і написати лист  
на пошту *пошта служби таксі*" +  
    " з описом виниклої проблеми (прописуєте я (ваш позивний) поповнював баланс програми, " +  
        "в банк звертався з їх боку проблем не виявлено і прикріпити фото і виписку. " +  
        "У темі так само вказуєте Ваш контактний номер і позивний для ідентифікації.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetReplenishmentOfBalanceQuestions(), cancellationToken: cancellationToken);
```

```
//Кнопка "Як вивести кошти з балансу на карту в нашому додатку?"
```

```
с  
а  
с  
е  
b  
u  
t  
t  
о  
n  
h  
о  
w  
т  
рн.;"  
о  
W  
і  
а  
t  
w  
h  
a  
d  
t  
b  
r  
o  
w  
o  
M  
i  
s  
C  
base "button_goBackToReplenishmentOfBalanceQuestions":  
{  
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:  
QuestionsInterface.GetReplenishmentOfBalanceQuestions(), cancellationToken: cancellationToken);
```

```
//Кнопка "Назад", для повернення до меню запитань "Поповнення балансу"
```

```
base "button_goBackToReplenishmentOfBalanceQuestions":  
{  
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:  
QuestionsInterface.GetReplenishmentOfBalanceQuestions(), cancellationToken: cancellationToken);
```

```
//----- кінець 5 блоку запитань
```

```
//----- 6 блок запитань
```

```
//Кнопка "Некоректний відгук/оцінка про поїздки"
```

```
с  
а  
с  
е  
а  
n  
u  
s  
t  
w  
a  
s  
r  
n  
b  
а  
d  
G  
r  
а  
d  
е
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);  
"Ми не можемо скоригувати оцінку райдера, тому що будь-який відгук - це суб'єктивне враження від поїздки.";
```

```
//----- кінець 6 блоку запитань
```

```
//----- 7 блок запитань
```

```
//Кнопка "Що робити, якщо ви виявили забуті речі в салоні?"
```

```
с  
а  
с  
е  
а  
n  
u  
s  
t  
w  
a  
s  
r  
n  
b  
а  
d  
G  
r  
а  
d  
е  
answer = "У разі, якщо ви знайшли речі, які у вас в авто забули клієнти," +  
" і ви знаєте за яким замовленням, то дану інформацію ви можете повідомити нам вибравши в  
архіві" +  
" замовлень потрібний вам замовлення і на ньому в пункті “проблеми із замовленням”” +  
" натискаєте “знайдені речі” → прописуєте коментар (вказуєте яку річ знайшли) і відправляєте  
запит." +  
" Якщо ж Ви не впевнені в тому, які з клієнтів забули річ, то інформацію про знахідку ви можете  
повідомити" +  
" по смс через програму УД або зв'язатися з оператором.";
```

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);  
е  
а  
к
```

```
//----- кінець 7 блоку запитань
```

```
//----- 8 блок запитань
```

```
//Кнопка "Як змінити особисті дані в нашій службі таксі?"
```

```
case "button_howToChangeDriverData":
```

```
{
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup: QuestionsInterface.GetHowToChangeDriverDataQuestions(), cancellationToken: cancellationToken);
```

```
    break;
```

```
}
```

```
//Кнопка "Мова"
```

```
case "button_language":
```

```
{
```

```
    answer = "Налаштування мови залишилася тільки для Android версії 7." +
```

```
        " Якщо версія Android вище, то мова виставляється відповідно до мови системи телефону." +
```

```
        " На IOS в останніх версіях також мова залежить від налаштувань телефону." +
```

```
        " Якщо мова телефону іноземна (Турецька і т.д.) за замовчуванням мова програми - українська.";
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup: AnswerInterface.GetHowToChangeDriverDataQuestions(), cancellationToken: cancellationToken);
```

```
    break;
```

```
}
```

```
//Кнопка "Телефон"
```

```
case "button_phone":
```

```
    answer = "Для того, щоб замінити номер телефону необхідно в додатку УД зайти у вкладку" +
```

```
        " Профіль, натиснути на номер телефону, вам потрібно ввести код, який прийде на старий номер телефону," +
```

```
        " щоб підтвердити свою особу, потім ввести новий номер телефону і код, який буде на нього відправлений." +
```

```
        " У разі, якщо у вас немає можливості ввести код, відправлений на старий номер, Вам необхідно звернутися в офіс" +
```

```
        " в робочий час у відповідному місті, де ви з нами співпрацюєте. Контактний номер може бути вказаний тільки один";
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup: AnswerInterface.GetHowToChangeDriverDataQuestions(), cancellationToken: cancellationToken);
```

```

        break;
    }

    //Кнопка "Авто"
    case "button_car":

a
n
s        " вам необхідно в робочий час (з пн по пт з 09:00 до 18:00)" +
w
e        " звернутися до офісу у відповідному місті. Або ви можете зробити це самостійно" +
r        " через додаток у вкладці \"Ваше авто\". Після підтвердження заміни даних вам надійде" +
"Для зміни даних автомобілем (марка, колір, номер)" +
        " відповідне повідомлення.";

a
w
a
i
t
break;
b
o
t
C
l
i
e
answer = "Для того, щоб змінити карту в додатку вам необхідно в профілі" +
t
S        " в розділі банківська карта видалити стару карту і вказати нову." +
e
n        " Видалити карту ви можете свайпом вліво і натиснути на значок кошика.";
d
T
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetHowToChangeDriverDataQuestions(), cancellationToken: cancellationToken);
t
M
e
s
s
a
g
e
A        //Кнопка "Назад", для повернення до меню запитань "Як змінити особисті дані в нашій службі таксі?"
case "button_goBackToHowToChangeDriverDataQuestions":
y
n
c        {
c
        await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetHowToChangeDriverDataQuestions(), cancellationToken: cancellationToken);
l
l
b
a
c
k
Q
u
e
r
y
M
e
s
s
a

```

//----- 9 блок запитань

//Кнопка "Умови брендуння авто та програми лояльності"

с  
а  
с  
е  
b  
u

answer = "Умови щодо брендуння автомобілів: \*посилання на сторінку сайту\*\n\nБонусні програми: \*посилання на сторінку сайту\*";

t  
t

await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);

t  
e  
r  
m  
s

O  
f  
C

//----- кінець 9 блоку запитань

a  
r  
b

//----- 10 блок запитань

r  
e  
n

//Кнопка "В якому класі мій автомобіль"

d  
i  
в  
g  
A

answer = "Щоб зрозуміти в якому класі ваш автомобіль, ви можете перейти за посиланням: \*посилання на сторінку сайту\*";

е  
у

await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:  
AnswerInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);

т  
н  
р

е  
R

с  
н

//----- кінець 10 блоку запитань

р  
f  
a

//----- 11 блок запитань

к

//Кнопка "Адреса офісів - всі міста"

с  
а  
с

answer = "Наші контакти за посиланням: \*посилання на контакти\*";

u  
t

t  
o

n  
g

e  
t

A

```
await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, answer, replyMarkup:
AnswerInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);
```

```
//----- кінець 11 блоку запитань
```

```
//Кнопка "Назад", для повернення до запитань "Інше" в розділі всіх запитань
```

```
case "button_goBackToAnotherQuestions":
```

```
{
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);
```

```
    break;
```

```
}
```

```
//Кнопка "Інше"
```

```
case "button_anotherQuestions":
```

```
{
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetAnotherQuestions(), cancellationToken: cancellationToken);
```

```
b  
r  
e  
a  
k
```

```
//Кнопка "Назад", для повернення до головних запитань
```

```
case "button_goBackToMainQuestions":
```

```
{
```

```
    await botClient.SendTextMessageAsync(callbackQuery.Message.Chat.Id, questionMessage, replyMarkup:
QuestionsInterface.GetMainQuestions(), cancellationToken: cancellationToken);
```

```
    break;
```

```
}
```

```
//Кнопка "Головне меню 📖"
```

```
case "button_mainMenu":
```

```
{
```

```
    await MainMenu(callbackQuery.Message, botClient, cancellationToken: cancellationToken);
```

```
b  
r
```

```

        //Метод обробки помилок у боті
private Task Error(ITelegramBotClient botClient, Exception e, CancellationToken cancellationToken)
    {
        using MessageLog messageLog = new(new ConsoleLogger());
        messageLog.LogMessage(e.Message);

        return Task.CompletedTask;
    }
}

```

## 2) ConsoleLogger.cs

```

namespace CSharpTaxiChatBot.Loggers
{
    //Логгер, який призначений для того, щоб виводити помилки до консолі
    і
    n
    t/Метод Log, який виводить помилку до консолі
    #Вхідні дані
    message - повідомлення помилки
    #Вихідні дані
    #Повідомлення у консолі з текстом помилки та датою цієї помилки
    public void Log(string message) => Console.WriteLine($"Error - {DateTime.Now}\n{message}\n");
    c }
}
a
s
3) FileLogger.cs
C
namespace CSharpTaxiChatBot.Loggers
{
    //Логгер, який призначений для того, щоб записувати помилки до файлу
    b
    n
    #Шлях до файлу з помилками за замовченням
    private readonly string _filePathStandard;
    b
    n
    #Шлях до файлу, який можна встановлювати
    public string FilePath { get; set; }
    a
    e
    c
    //Параметричний конструктор
    public FileLogger(string filePath)
    і
    Logger
    {
        s
        s
        s
        F
        i
        #Метод Log, який виводить помилку до консолі
        #Вхідні дані
        #message - повідомлення помилки
        #Вихідні дані
        #Текст у текстовому файлі з помилкою та датою цієї помилки
        public void Log(string message)
        e
        r
        і
        Logger

```

```

//Файл існує? - записати у вказаний користувачем шлях
if (File.Exists(filePath))
    {
        File.AppendAllText(filePath, $"Error - {DateTime.Now}\n{message}\n");

// - записати у шлях вказаний за замовченням
    {
        File.AppendAllText(_filePathStandard, $"Error - {DateTime.Now}\n{message}\n");
    }
    }
}
}
}
}

```

#### 4) ILogger.cs

```

namespace CSharpTaxiChatBot.Loggers
{
    //Інтерфейс, який призначений для використання Dependency Injection у логгерах
    interface ILogger
    {
        //Прототипу методу логування помилки
        void Log(string message);
    }
}

```

#### 5) MessageLog.cs

```

namespace CSharpTaxiChatBot.Loggers
{
    //Сервіс, який приймає різні логери і викликає їх метод логування
    internal class MessageLog : IDisposable
    {
        //Логгер
        private readonly ILogger _logger;

        //Параметричний конструктор
        public MessageLog(ILogger logger)
        {
            _logger = logger;
        }

        //Метод для виклику логування з логгеру
        //Вхідні дані
        //messageForLog - текст помилки для логування
        //Вихідні дані
        //Помилка в консолі, або у файлі, в залежності від логеру
        public void LogMessage(string messageForLog) => _logger.Log(messageForLog);

        public void Dispose() { }
    }
}

```

#### 6) AnswerInterface.cs

```

using Telegram.Bot.Types.ReplyMarkups;

namespace CSharpTaxiChatBot.MenuInterfaces
{
    //Клас, який відповідає за інтерфейс користувача, коли він доходить до відповіді на якесь питання
    internal class AnswerInterface
    {
        //Повернутися до меню запитань "Прийняття/скасування замовлення"
        public static InlineKeyboardMarkup GetOrCancelOrderAnswer()
        {
            InlineKeyboardMarkup keyboardMarkupQuestionsHowToGetOrder = new(new[]
            {
                new[]
                {
                    InlineKeyboardButton.WithCallbackData("Назад ⏪", "button_goBackToGetOrCancelOrder"),
                    InlineKeyboardButton.WithCallbackData("Головне меню 📖", "button_mainMenu")
                }
            });
        }
    }
}

```

```

    }
    });

    return keyboardMarkUpQuestionsHowToGetOrder;
}

//Повернутися до меню запитань "Як скасувати замовлення?"
public static InlineKeyboardMarkup GetHowToCloseOrderAnswer()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsHowToGetOrder = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад◀", "button_goBackToHowToCancelOrder"),
            InlineKeyboardButton.WithCallbackData("Головне меню 📄", "button_mainMenu")
        }
    });

    return keyboardMarkUpQuestionsHowToGetOrder;
}

//Повернутися до меню запитань "Оплата безготівкових замовлень"
public static InlineKeyboardMarkup GetTerminalPaymentQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsTerminalPayment = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад◀", "button_goBackToTerminalPaymentQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню 📄", "button_mainMenu")
        }
    });

    return keyboardMarkUpQuestionsTerminalPayment;
}

//Повернутися до меню запитань "Штрафи/компенсації"
public static InlineKeyboardMarkup GetFinesQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsTerminalPayment = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад◀", "button_goBackToFinesQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню 📄", "button_mainMenu")
        }
    });

    return keyboardMarkUpQuestionsTerminalPayment;
}

//Повернутися до головного меню запитань "Тарифи/додаткові послуги"
public static InlineKeyboardMarkup GetMainAdditionalServicesQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsAdditionalServices = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад◀",
"button_goBackToAdditionalServicesQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню 📄", "button_mainMenu")
        }
    });


    return keyboardMarkUpQuestionsAdditionalServices;
}

//Повернутися до іншого меню запитань у "Тарифи/додаткові послуги"
public static InlineKeyboardMarkup GetAnotherAdditionalServicesQuestions()

```



```

{
//Клас, який відповідає за інтерфейс користувача у стандартних меню
internal class QuestionsInterface
{
//Меню головних запитань
public static InlineKeyboardMarkup GetMainQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsMain = new(new[]
        {
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Прийняття/Скасування замовлення",
"button_getOrCancelOrder")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Оплата безготівкових замовлень",
"button_paymentByTerminal")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Штрафи/компенсації", "button_finesCompensations")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Тарифи/Додаткові послуги",
"button_ratesAdditionalServices")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Поповнення балансу", "button_replenishmentOfBalance")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Інше", "button_anotherQuestions")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Головне меню  ", "button_mainMenu")
            }
        }
    )
}
}

```

```

    }
    });

    return keyboardMarkUpQuestionsMain;
}

//Меню запитань "Прийняття/скасування замовлення"
public static InlineKeyboardMarkup GetOrCancelOrderQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsGetOrCancelOrder = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Способи прийняття замовлень",
"button_howToGetOrder")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Принципи розподілу замовлень",
"button_principleOfDistributionsOfOrders")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Як скасувати замовлення?", "button_howToCancelOrder")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Як закрити замовлення?", "button_howToCloseOrder")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Види і типи замовлень", "button_typesOfOrder")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToMainQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
        }
    });
}

```

```

return keyboardMarkUpQuestionsGetOrCancelOrder;
}

//Меню запитань "Як скасувати замовлення?"
public static InlineKeyboardMarkup GetHowToCancelOrderQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsHowToCancelOrder = new(new[]
        {
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Клієнт не виходить на зв'язок",
"button_clientNotContacting")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Клієнт не виходить до авто",
"button_clientDoesNotGoesToTheCar")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Поломка авто в процесі поїздки з клієнтами",
"button_carBrokenWithClients")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Не поміщається багаж (вантаж)",
"button_luggageDoesNotFit")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Некоректна поведінка пасажирів",
"button_unacceptableClientBehavior")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToGetOrCancelOrder"),
                InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
            }
        });
}

```

```

return keyboardMarkUpQuestionsHowToCancelOrder;
}

//Меню запитань "Оплата безготівкових замовлень"
public static InlineKeyboardMarkup GetTerminalPaymentQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsTerminalPayment = new(new[]
        {
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Куди надходять гроші по БГ замовленням?",
"button_whereDoesTheMoneyGoPaymentByTerminal")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Чи можна регулювати зарахування?",
"button_canPaymentsBeRegulated")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToMainQuestions"),
                InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
            }
        });

return keyboardMarkUpQuestionsTerminalPayment;
}

//Меню запитань "Штрафи/компенсації"
public static InlineKeyboardMarkup GetFinesQuestions()
{
    InlineKeyboardMarkup keyboardMarkupQuestionsFines = new(new[]
        {
            new[]
            {
                InlineKeyboardButton.WithCallbackData("За що знімаються штрафи", "button_whatAreFinesFor")
            },
            new[]

```

```

        {
            InlineKeyboardButton.WithCallbackData("Типи компенсацій та оформлення запитів на них",
"button_compensationTypes")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToMainQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
        }
    });

```

```

    return keyboardMarkupQuestionsFines;
}

```

//Головне меню запитань "Тарифи/додаткові послуги"

```

public static InlineKeyboardMarkup GetMainAdditionalServicesQuestions()
{
    InlineKeyboardMarkup keyboardMarkupQuestionsAdditionalServices = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Тарифи і комісія", "button_tariffsAndCommission")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Додатковий багаж в салоні",
"button_additionalBaggageInTheCabin")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Їду з твариною", "button_driveWithAnimal")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Кондиціонер", "button_aitConditioner")
        },
        new[]
        {

```

```

        InlineKeyboardButton.WithCallbackData("Інші додаткові послуги",
"button_anotherAdditionalServices")
    },
    new[]
    {
        InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToMainQuestions"),
        InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
    }
});

return keyboardMarkupQuestionsAdditionalServices;
}

//Меню запитань з іншими додатковими послугами у "Тарифи/додаткові послуги"
public static InlineKeyboardMarkup GetAnotherAdditionalServices()
{
    InlineKeyboardMarkup keyboardMarkupQuestionsAnotherAdditionalServices = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Водій, який не палить", "button_driverNotSmoking")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Англомовний водій", "button_engDriver")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Зустріч з табличкою", "button_meetingWithASign")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Тиша в салоні", "button_silenceInTheCabin")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад ◀",
"button_goBackToMainAdditionalServicesQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
        }
    }
);
}

```

```

        }
    });

    return keyboardMarkupQuestionsAnotherAdditionalServices;
}

//Меню запитань "Поповнення балансу"
public static InlineKeyboardMarkup GetReplenishmentOfBalanceQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsReplenishmentOfBalance = new(new[]
    {
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Як поповнити баланс в нашому додатку?",
"button_howToReplenishTheBalance")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Що робити, якщо поповнив не свій позивний?",
"button_filledUpNotMyCallSign")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Поповнив позивний, а гроші не прийшли",
"button_toppedUpMyCallsignButMoneyDidNotCome")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Як вивести кошти з балансу на карту в нашому
додатку?", "button_howToWithdrawMoneyInTheApp")
        },
        new[]
        {
            InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToMainQuestions"),
            InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
        }
    });

    return keyboardMarkUpQuestionsReplenishmentOfBalance;
}

```

```

//Меню інших запитань
public static InlineKeyboardMarkup GetAnotherQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsAdditional = new(new[]
        {
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Некоректний відгук/Оцінка про поїзду",
"button_badGrade")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Що робити, якщо ви виявили забуті речі в салоні?",
"button_lostThings")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Як змінити особисті дані в нашій службі таксі?",
"button_howToChangeDriverData")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Умови брендування авто та програми лояльності",
"button_termsOfCarbrendingAndLoyaltyPrograms")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("В якому класі мій автомобіль?", "button_checkCarClass")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Адреса офісів - всі міста", "button_getAllAddresses")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToMainQuestions"),
                InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
            }
        });
}

```

```

return keyboardMarkUpQuestionsAdditional;
}

//Меню запитань "Як змінити особисті дані в нашій службі таксі?"
public static InlineKeyboardMarkup GetHowToChangeDriverDataQuestions()
{
    InlineKeyboardMarkup keyboardMarkUpQuestionsHowToChangeDriverData = new(new[]
        {
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Мова", "button_language")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Телефон", "button_phone")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Авто", "button_car")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Картка", "button_creditCard")
            },
            new[]
            {
                InlineKeyboardButton.WithCallbackData("Назад ◀", "button_goBackToAnotherQuestions"),
                InlineKeyboardButton.WithCallbackData("Головне меню ☰", "button_mainMenu")
            }
        });

    return keyboardMarkUpQuestionsHowToChangeDriverData;
} }

```

9) Program.cs

```
using CSharpTaxiChatBot.Bot;
```

```
Bot bot = new("6187176482:AAFF5tA_vXm_9ijMBRLCqJJzIXre6KrdC_g")
```

