

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ  
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

## КУРСОВА РОБОТА

з дисципліни  
«СИСТЕМИ БАЗ ДАНИХ»

на тему: «БД служби таксі»

Студентки 3 курсу, групи КА-98  
спеціальності 122 «Комп'ютерні науки»  
*Супрунук Юлії Володимирівни*

Керівник: Афанасьєва І.В.

Національна оцінка: \_\_\_\_\_

Кількість балів: \_\_\_\_ Оцінка: ECTS \_\_\_\_

|               |          |  |
|---------------|----------|--|
| Члени комісії | _____    | _____  |
|               | (підпис) | (науковий ступінь, вчене звання, прізвище та ініціали) |
|               | _____    | _____  |
|               | (підпис) | (науковий ступінь, вчене звання, прізвище та ініціали) |
|               | _____    | _____  |
|               | (підпис) | (науковий ступінь, вчене звання, прізвище та ініціали) |

Київ 2021

## **ЗМІСТ**

|  |           |
|--|-----------|
| <b>ПЕРЕЛІК СКОРОЧЕНЬ</b>   | <b>3</b>  |
| <b>ВСТУП</b>   | <b>4</b>  |
| <b>РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ</b>                                 | <b>6</b>  |
| 1. 1. Опис предметної області.                                     | 6         |
| 1. 2. Аналіз аналогічних продуктів.                                | 7         |
| 1. 3. Вибір інструментів для розробки.                             | 8         |
| 1. 4. Постановка задачі проектування                               | 9         |
| <b>РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ.</b>             | <b>10</b> |
| 2. 1. Концептуальна модель бази даних. Вимоги користувача.         | 10        |
| 2. 2. Діаграма потоків даних.                                      | 11        |
| 2. 3. Інфологічна модель бази даних. Діаграма «сутність— зв’язок». | 12        |
| 2. 4. Створення привілейованого користувача.                       | 13        |
| <b>РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ</b>                         | <b>14</b> |
| 3. 1. Реалізація зв'язків.   | 14        |
| 3. 2. Інструкція користувача.                                      | 16        |
| <b>ВИСНОВКИ</b>  | <b>24</b> |
| <b>СПИСОК ЛІТЕРАТУРИ</b>   | <b>25</b> |
| <b>ДОДАТОК А. ER-ДІАГРАМА</b>                                      | <b>26</b> |

## **ПЕРЕЛІК СКОРОЧЕНЬ**

БД – База Даних

СКБД - Система Керування Базами Даних

DFD - Data Flow Diagrams

ER - Entity-Relationship

ID – Identifier

SQL - Structured Query Language

## ВСТУП

**Актуальність.** У наш час бази даних є невід’ємним механізмом збереження інформації, а системи керування базами даних надають можливість швидко й ефективно маніпулювати нею, представляти у зрозумілому для користувача вигляді. У курсовій роботі спроектовано базу даних служби таксі. Таксі користується широким попитом серед людей, і мені, як майбутньому розробнику, необхідно зрозуміти на прикладі такої системи, як користувачі взаємодіють із сервером та базою даних, розібрати бізнес логіку процесів, які є основою для створення програмного продукту.

**Мета.** Розробка системи замовлення таксі, яка дозволить зберігати всю інформацію про замовлення та користувачів, та яка забезпечить клієнтам та робітникам потрібний функціонал.

**Завдання.** Спроектувати базу даних, реалізувати всі необхідні запити та процедури для роботи з нею, створити зручний веб застосунок для користувачів системи.

**Практичне значення.** Розроблений навчальний продукт створений задля поглиблення і вдосконалення теоретичних знань проектування баз даних, та закріплення здобутих навичок на прикладі вирішення конкретної задачі.

**Програмне забезпечення.** При розробці даного програмного продукту було використано систему керування базами даних PostgreSQL, веб-додаток реалізовано з використанням фреймворку Angular мови програмування TypeScript, сервер за допомогою платформи Node.js. При виконанні роботи було використано таке програмне забезпечення: редактор

коду Visual Studio Code, середовище розробки PostgreSQL pgAdmin4; операційна система Ubuntu 20.04.3 LTS.

**Структура роботи.** Робота складається зі переліку скорочень, вступу, трьох розділів, висновку, списку використаних джерел та додатку А.

## РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Онлайн сервіси виклику таксі є полегшенням процесу замовлення поїздки. Вони дозволяють робити оплату за допомогою прив'язаної в особистому кабінеті банківської картки. У час, коли майже у всіх людей є телефони з можливістю виходу в інтернет, такі додатки стають все популярнішими і популярнішими, користуючись великим попитом серед жителів мегаполісів, та потрохи перебираюся до маленьких містечок.

### 1. 1. Опис предметної області.

Служба таксі надає послуги перевезення. Для того, щоб її викликати, клієнт має розмістити своє замовлення на платформі певного таксі(веб сайт чи мобільний додаток), або ж зателефонувати до неї, і передати диспетчеру дані, які той введе в систему. Під час процесу подання заявки на виклик машини, клієнт має передати інформацію про своє місцезнаходження, цільову адресу, кількість пасажирів, багаж, вид оплати та деякі опціональні деталі, наприклад перевезення тварин.

Після того як клієнт тим чи іншим чином зробив замовлення, диспетчер має повідомити його про вартість поїздки. Якщо пасажир вона влаштовує, заявка прямує до наступної дійової ролі.

Коли всі дані внесені до системи, інформацію про замовлення має можливість переглянути водій. Він може як відмовитись, так і прийняти його. Якщо водій підтверджує виконання, пасажир на своїй стороні може побачити інформацію про перевізника та його машину.

Машина приїжджає до клієнта, і водій везе його до пункту призначення. Процес оплати відбувається залежно від його форми, якщо клієнт обрав розрахунок одразу в додатку, то платіж проходить до поїздки, якщо ж готівкою, чи карткою в машині, то в кінці поїздки.

Крім того, після завершення замовлення, водій має позначити його як виконане. Клієнт, в свою чергу, має можливість оцінити водія та залишити

відгук про поїздки.

Відгуки можуть переглядати всі користувачі системи.

Крім того, в системі служби таксі є адміністратор. Він призначає роль диспетчера у додатку, бо відкрита реєстрація можлива тільки для водіїв чи клієнтів. Диспетчери відносяться до адміністрації, але не можуть призначати ролі користувачам.

## **1. 2. Аналіз аналогічних продуктів.**

На даний момент існують чимало компаній, які займаються перевезеннями. Однією з таких є Uber[1]. Uber мають сайт та мобільний додаток, через які клієнт може замовити таксі. Функціонал їх веб сайту дозволяє пасажиру на карті обирати своє місцезнаходження та пункт призначення, розраховувати оплату в спеціальному калькуляторі. Крім того, для водіїв та користувачів, є так звані “гайди”, для того, щоб вони могли з легкістю почати використовувати цей застосунок. При реєстрації водія є можливість обрати вид співпраці та спосіб отримання зарплати, можна як орендувати автомобіль у партнерського автопарку, так і працювати на своєму. Обов’язковим етапом є перевірка документів водія. Тобто, тільки після перевірки даного користувача, він отримає можливість перевозити людей.

Ще одною, не менш популярною службою є Bolt[2]. Її сайт дозволяє зареєструватися у якості водій, через спеціальну форму, у якій необхідно вказувати повну інформацію про машину, особистість перевізника, документи та дані для виплат. Пасажирів одразу перенаправляють до мобільного застосунку.

Проаналізувавши дані продукти, можу зробити висновок, що служби таксі для долучення нових клієнтів та співробітників(водіїв) користуються насамперед мобільними додатками, бо це зручно у повсякденному житті. Але у користувачів є також можливість робити замовлення через веб сайти.

### **1. 3. Вибір інструментів для розробки.**

Насамперед, після розробки архітектури додатку, необхідно було обрати, яким чином буде зберігатись інформація про користувачів та замовлення. Згідно з поставленою умовою у даній роботі була спроектована реляційна база даних. За СКБД було обрано - PostgreSQL. PostgreSQL є вільнорозповсюджуваним продуктом з відкритим вихідним кодом, який доступний на великій кількості платформ. Ця система забезпечує дуже хорошу підтримку стандарту мови SQL і також надає цікаві та практично корисні додаткові можливості[3].

Користувач не може працювати напряму з базою даних. Для того, щоб дані були захищені, оброблені, і звісно ж для зручності використання створений бекенд та фронтенд вебсайту.

Локальний сервер було реалізовано за допомогою платформи Node.js. Ця платформа, заснована на JavaScript-бібліотеці Chrome, яка дозволяє спростити створення швидких масштабованих мережевих додатків[4]. Крім того, вона має велику спільноту розробників, що значно полегшить мені, як новачку, створення веб серверу для курсової роботи.

Для побудови фронтенд частини проекту було використано фреймворк Angular. Він забезпечує розробника інструментами CLI (для створення, побудови, тестування та розгортання додатків), роблячи ці інструменти доступними для легкої інтеграції в користувацькі рішення. Angular написаний за допомогою TypeScript, який є надпростором мови JavaScript, який надає можливість примусової типізації. TypeScript може допомогти виявити багато простих синтаксичних помилок, перш ніж вони вплинуть на програму[5].



#### **1. 4. Постановка задачі проектування**

Відповідно до умов технічного завдання, необхідно створити додаток служби таксі у вигляді веб сайту, з сервером та базою даних, з можливостями замовлення таксі, обробкою замовлень, переглядом інформацію про них та в реалізації наступних пунктів:

1. Змоделювати структуру бази даних.
2. Створити користувача бази даних, у якого є права на взаємодію лише з базою таксі.
3. Створити бекенд з сервісами та контролами для можливості доступу до бази даних.
4. Підключити бекенд до бази даних.
5. Передбачити аутентифікацію/авторизацію користувача, використовуючи JSON Web Tokens.
6. Спроекувати інтерфейс веб додатку інтерфейс, який буде відповідати вимогам «дружнього» інтерфейсу користувача;
7. Розробити фронтенд та зв'язати його з бекендом за допомогою запитів та відповідей.
8. Тестувати роботу всіх компонентів веб додатку, виправляючи помилки, які виникли під час розробки.

## **РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ.**

### **2. 1. Концептуальна модель бази даних. Вимоги користувача.**

Насамперед, потрібно описати вимоги до системи, що розробляється. Для цього необхідно прописати так звані user stories. User story - це неформальне загальне пояснення функцій програмного забезпечення, написане з точки зору кінцевого користувача або замовника. Мета полягає в тому, щоб сформулювати, яку цінність для клієнта матимуть окремі функціональні одиниці програмного продукту[6].

В системі таксі є чотири основні ролі для користувачів: пасажир(клієнт), водій, диспетчер та адміністратор, тому для кожного з них потрібно описати вимоги.

Як пасажир, я хочу:

- мати можливість створювати замовлення;
- бачити інформацію про конкретне замовлення, її оновлення;
- мати змогу залишити оцінку та коментар про поїздку, після завершення поїздки;
- мати історію всіх моїх замовлень.

Як водій, я хочу:

- мати можливість додавати в систему, машини, якими я володію та на яких буду перевозити клієнтів;
- отримувати список замовлень;
- мати можливість прийняти замовлення та вказати час подання авто;
- мати змогу оцінити поїздку;
- переглядати повну інформацію про виконане замовлення;
- мати інформацію про диспетчерів, щоб мати можливість з ними зв'язатися;

Як диспетчер, я хочу:

- переглядати список замовлень, нових та тих, що вже виконані;
- підтверджувати та відхиляти замовлення.
- передавати в замовлення ціну;
- видаляти користувачів;
- бачити статистику водіїв та клієнтів.

Як адміністратор, я хочу:

- переглядати всі замовлення та інформацію про них;
- видаляти користувачів;
- надавати ролі користувачам;
- бачити статистику водіїв та клієнтів.

## 2. 2. Діаграма потоків даних.

На основі заданих вимог користувача можна проаналізувати бізнес-процес у вигляді діаграми потоків даних. Ця діаграма показує переміщення інформації з одного місця в інше як частину конкретного процесу в цілому. DFD може продемонструвати, як різні відділи системи взаємодіють — це робить речі зрозумілими та узгодженими[7].

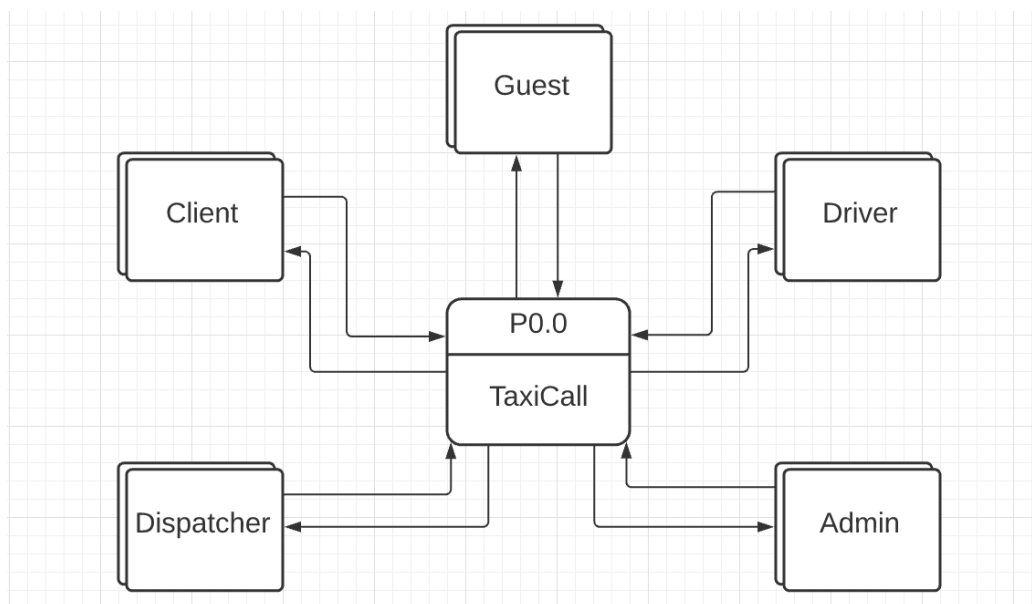


Рис. 2.1 DFD-0. P0.0 TaxiCall.

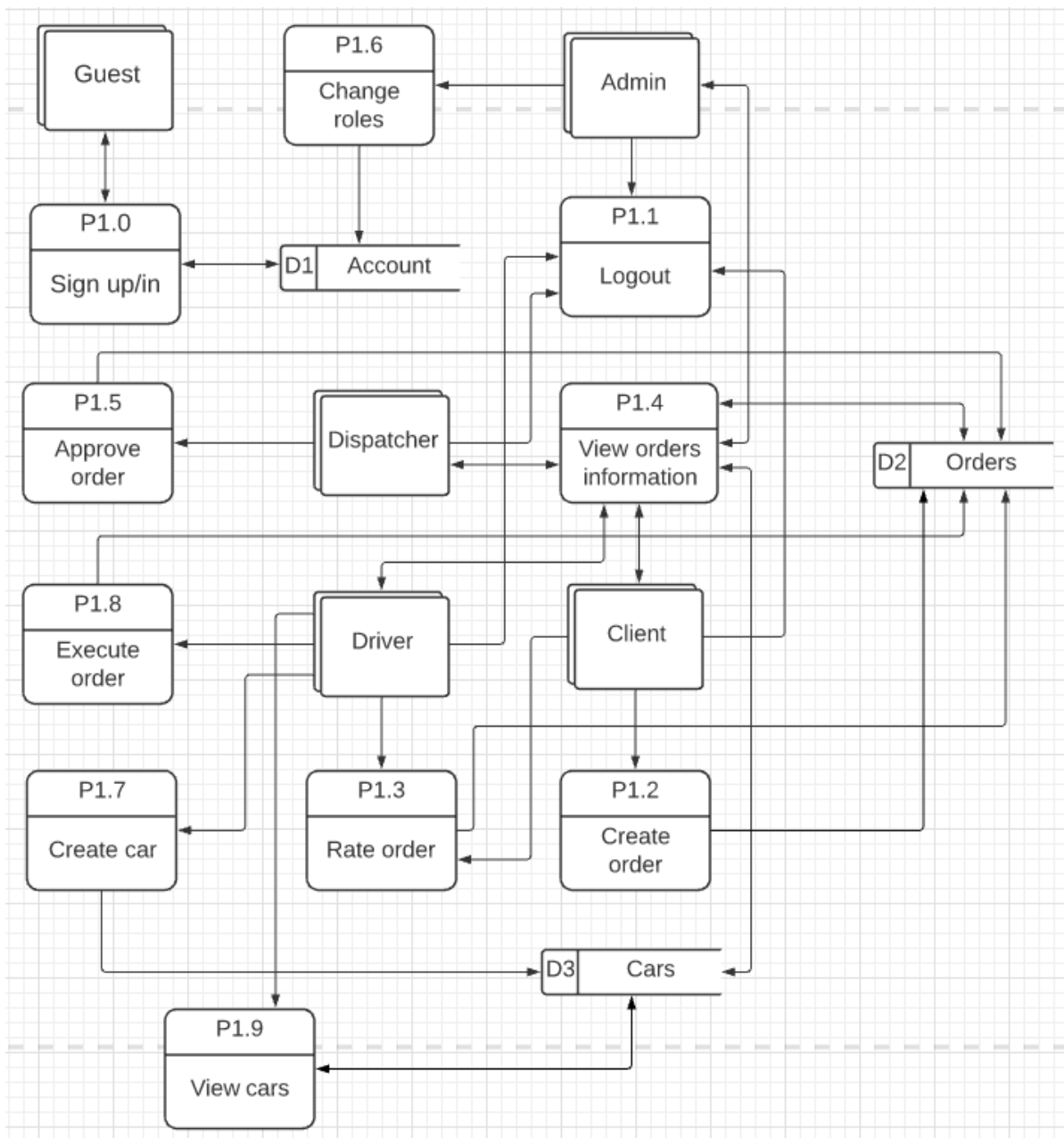


Рис. 2.2 DFD-1. P1 Деталізований процес TaxiCall.

## 2. 3. Інфологічна модель бази даних. Діаграма «сутність– зв’язок».

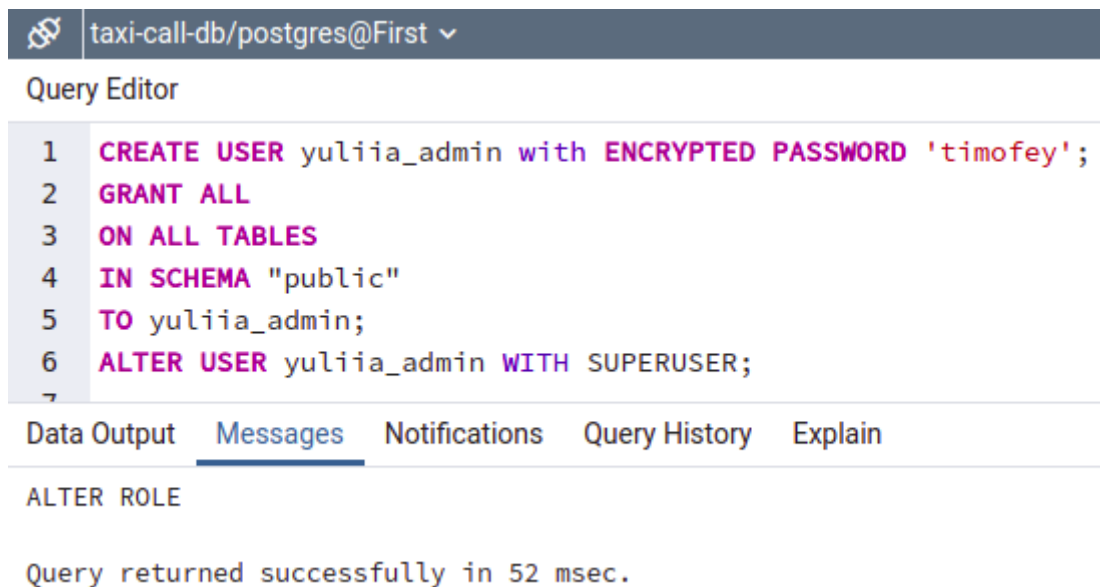
Інфологічна модель представлена у вигляді ER-діаграми, яка винесена у додаток А(Рис. 1).

База даних складається з семи таблиць: users, orders, cars, user\_role, car\_types, order\_status, roles. Для кожної з таблиць визначені первинні(primary) та зовнішні(foreign) ключі. Крім того, для таблиць cars та

user\_role було створено додатково унікальні ключі за стовпцями, щоб не було дублювання інформації.

## 2. 4. Створення привілейованого користувача.

Дефолтний користувач програми pgAdmin4 має всі привілеї, які тільки можуть бути для таблиць. Він може створювати інших користувачів та бази даних. Такий користувач отримує доступ і до інших баз даних, ніяк не пов'язаних з даним програмним продуктом. Такої ситуації можна уникнути, створивши обліковий запис, який може мати обмежені привілеї на конкретну бд, у моєму випадку, taxi-call-db.



```
1 CREATE USER yuliia_admin with ENCRYPTED PASSWORD 'timofey';
2 GRANT ALL
3 ON ALL TABLES
4 IN SCHEMA "public"
5 TO yuliia_admin;
6 ALTER USER yuliia_admin WITH SUPERUSER;
```

ALTER ROLE

Query returned successfully in 52 msec.

Рис. 2.3 Створення облікового запису та надання привілеїв.

## РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ

### 3. 1. Реалізація зв'язків.

Для реалізації зв'язку між базою даних та бекендом було встановлено пакет pg для Node.js.

Для встановлення зв'язку між фронтендом та бекендом використовувалась маршрутизація(роути) фреймворку Express. За допомогою маршрутизації описуються відповіді додатку Node.js на запити, які виконуються з клієнтської сторони на конкретну URL-адресу.

Таблиця 3.1

Опис реалізованих та використаних маршрутів серверу.

| №  | URL  | Метод  | Параметри запити            | Призначення   |
|----|--|--------|-----------------------------|---|
| 1. | http://localhost:5000/api/users/login          | post   | email,<br>password,<br>role | Перевіряє чи наявний користувач у системі і якщо так, повертає токен. |
| 2. | http://localhost:5000/api/users                | post   | email,<br>password,<br>role | Додає користувача у базу даних і повертає токен.                      |
| 3. | http://localhost:5000/api/users/role           | post   | user_id,<br>rolename        | Додає користувачу роль.   |
| 4. | http://localhost:5000/api/users/role/:rolename | get    | limit,<br>page,<br>rolename | Повертає всіх користувачів з заданою роллю.                           |
| 5. | http://localhost:5000/api/users/:id            | get    | id                          | Повертає користувача за вказаним id.                                  |
| 6. | http://localhost:5000/api/users/:id/roles      | get    | id                          | Повертає всі ролі користувача.  |
| 7. | http://localhost:5000/api/users/role           | delete | user_id,<br>rolename        | Видалення ролі користувача.   |

| №   | URL   | Метод  | Параметри запиту   | Призначення                                       |
|-----|---|--------|--|---|
| 8.  | http://localhost:5000/api/orders                  | get    | limit,<br>page,<br>userId,<br>userRole   | Повертає замовлення залежно від ролі користувача. |
| 9.  | http://localhost:5000/api/orders/:id              | get    | id   | Повертає замовлення по id.                        |
| 10. | http://localhost:5000/api/orders/:id              | delete | id   | Видалення замовлення по id.                       |
| 11. | http://localhost:5000/api/orders/statuses         | get    |  | Повертає всі статуси замовлень.                   |
| 12. | http://localhost:5000/api/orders/dispatcher/:id   | put    | id, approved,<br>payment   | Оновлення замовлення диспетчером.                 |
| 13. | http://localhost:5000/api/orders/driver/:id       | put    | id,<br>waiting_time,<br>order_status,<br>car_id  | Оновлення замовлення водієм.                      |
| 14. | http://localhost:5000/api/orders/grade/driver/:id | put    | id,<br>driver_comment,<br>driver_grade   | Додає відгук водія про замовлення.                |
| 15. | http://localhost:5000/api/orders/grade/client/:id | put    | id,<br>client_comment,<br>client_grade,  | Додає відгук клієнта до замовлення.               |
| 16. | http://localhost:5000/api/orders                  | post   | origin_address,<br>destination_address,<br>number_of_people,<br>empty_trunk,<br>animals,<br>terminal,<br>air_condition,<br>car_type_id | Створення замовлення.                             |

| №   | URL   | Метод  | Параметри запиту  | Призначення  |
|-----|---|--------|---|--|
| 17. | <a href="http://localhost:5000/api/orders/statistics">http://localhost:5000/api/orders/statistics</a> | get    | userId<br>userRole  | Повертає статистику користувача, загальну кількість замовлень, та з них - підтверджених. |
| 18. | <a href="http://localhost:5000/api/cars">http://localhost:5000/api/cars</a>                           | get    | limit, page,<br>userId  | Повертає всі машини конкретного водія.   |
| 19. | <a href="http://localhost:5000/api/cars/types">http://localhost:5000/api/cars/types</a>               | get    |   | Повертає всі типи машин.   |
| 20. | <a href="http://localhost:5000/api/cars/types/:id">http://localhost:5000/api/cars/types/:id</a>       | get    | id  | Повертає назву типу машини за унікальним id.   |
| 21. | <a href="http://localhost:5000/api/cars/:id">http://localhost:5000/api/cars/:id</a>                   | get    | id  | Повертає об'єкт машини за id.  |
| 22. | <a href="http://localhost:5000/api/cars/">http://localhost:5000/api/cars/</a>                         | post   | license_number, model, color, type_id, air_conditioning, terminal, empty_trunk, animals | Додання нової машини в систему.  |
| 23. | <a href="http://localhost:5000/api/cars/:id">http://localhost:5000/api/cars/:id</a>                   | delete | id  | Видалення машини.  |

### 3. 2. Інструкція користувача.

Для початку роботи з програмою необхідно запустити на виконання локальні сервери бекенду та фронтенду.

Бекенд: `npm run start`

Фронтенд: `npm install, npm run build, npm run start`

Після запуску потрібно перейти за посиланням <http://localhost:4200/>.



Користувач може ознайомитися зі сторінкою входу у додаток (рис.3.1). Якщо користувач ще не зареєстрований у системі виклику таксі, то він переходить до сторінки “Sign in” і вводить свої дані, за допомогою яких у подальшому зможе авторизуватись у додатку (рис. 3.2). Збереження даних відбувається після натискання кнопки “Submit”. Якщо користувач вже вніс свої дані до бази, він переходить до сторінки “Login”, і вводить у відповідні комірочки свої авторизаційні дані(рис.3.3). Ця дія підтверджується за допомогою кнопки “Login”.

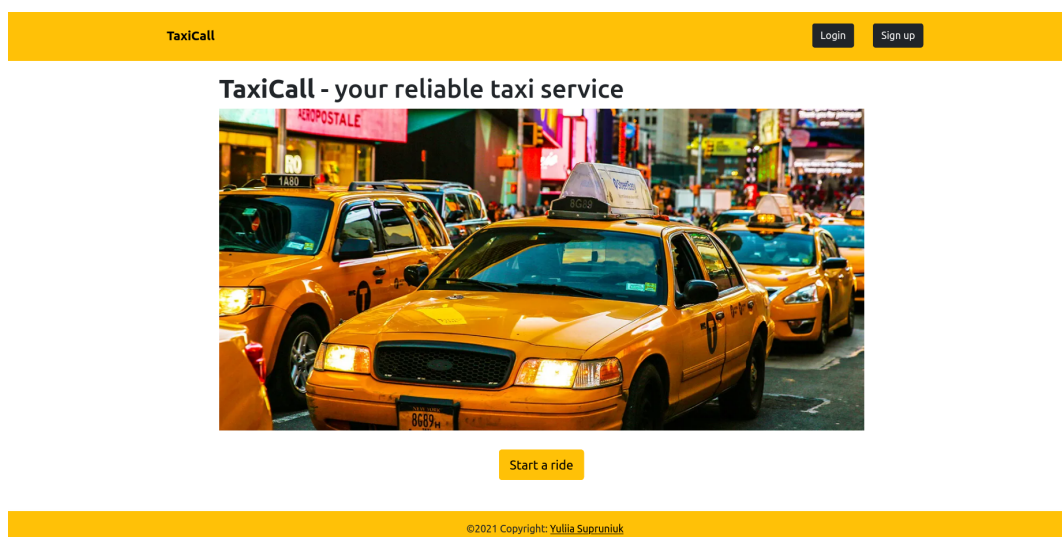


Рис.3.1 Головна сторінка додатку.

На сторінці реєстрації користувач також повинен вказати свою роль.

Рис.3.2 Створення облікового запису

The form is titled 'Email address' and contains a text input field with the email 'peyton.franklin@example.com'. Below it is a 'Password' section with a text input field showing six dots. The 'Role' section features a dropdown menu currently set to 'Passenger'. At the bottom is a dark 'Login' button.

Рис.3.3 Сторінка авторизації зареєстрованого користувача

Клієнт при вході отримує доступ до перегляду своїх замовлень, та створення нових. Натиснувши кнопку ‘New order’ пасажир може створити замовлення для виклику таксі, і вказати свої переваги(рис 3.4).


The interface shows a yellow header with 'TaxiCall Orders' and a 'Logout' button. Below is a section titled 'Orders' with a 'New order' button. A modal form is open for creating a new order. It includes three text input fields for address: '8725 W Pecan St', '381 Pockrus Page Rd', and '2'. There are four checkboxes: 'Empty trunk' (checked), 'Animal(small pet)' (checked), 'Terminal' (unchecked), and 'Air conditioning' (unchecked). A dropdown menu is set to 'universalis - large trunk'. At the bottom of the modal are 'Create' and 'Close' buttons. Below the modal, a blue bar indicates 'No orders'. The footer contains the copyright notice '©2021 Copyright: Yulia Supruniuk'.

Рис.3.4 Форма створення замовлення.

Пасажир може переглянути всі свої замовлення у таблиці(рис 3.5), та перейти на сторінку конкретного(рис 3.6).

**Orders** New order

Total number of orders: 1

| Driver  | Origin address  | Destination         | Payment | Date              | Approved |   |
|---------|-----------------|---------------------|---------|-------------------|----------|---|
| No info | 8725 W Pecan St | 381 Pockrus Page Rd | No info | 12/21/21, 5:31 PM | ×        |  |

1

Рис.3.5 Перегляд замовлень користувача.

**TaxiCall** Orders Logout

Driver:

Passenger: Dwayne Butler

Origin address: 8725 W Pecan St

Date: 12/21/21, 5:31 PM

Destination address: 381 Pockrus Page Rd

×

Approved

Passengers: 2

Car type: universalis

Empty trunk

Animals

©2021 Copyright: Yuliia Supruniuk

Рис.3.6 Перегляд конкретного замовлення.

Далі користувач повинен дочекатися підтвердження замовлення зі сторони диспетчера та водія. Якщо в системі авторизується диспетчер, він має перевірити замовлення та підтвердити його, вказавши сумму оплати за нього(рис. 3.7).

TaxiCall
Clients Drivers Orders
Logout

Driver:  
Passenger: Dwayne Butler  
Dispatcher: Mia Burton

Origin address: 8725 W Pecan St  
Destination address: 381 Pockrus Page Rd  
Passengers: 2  
Car type: universalis  
Empty trunk  
Animals

Date: 12/21/21, 5:31 PM

✓ Approved

Total payment: 124\$

©2021 Copyright: Yulia Supruniuk

Рис.3.7 Підтвердження замовлення диспетчером.

У диспетчера, окрім можливості перегляду замовлень, є можливість переглянути профілі пасажирів та водіїв(рис. 3.8).

TaxiCall
Clients Drivers Orders
Logout

Clients

Total number of clients: 16

| Name      | Surname  | Email                        | Phone          | Date of joining   |
|-----------|----------|------------------------------|----------------|-------------------|
| Dwayne    | Butler   | dwayne.butler@example.com    | (406)-848-3346 | 12/21/21, 5:26 PM |
| Peyton    | Franklin | peyton.franklin@example.com  | (681)-829-8181 | 12/17/21, 4:39 PM |
| Ricardo   | Hoffman  | ricardo.hoffman@example.com  | (701)-308-8424 | 12/17/21, 4:38 PM |
| Ben       | Hopkins  | ben.hopkins@example.com      | (179)-313-1882 | 12/17/21, 4:38 PM |
| Esther    | Cole     | esther.cole@example.com      | (121)-683-1506 | 12/17/21, 4:38 PM |
| Catherine | Willis   | catherine.willis@example.com | (539)-266-6370 | 12/17/21, 4:37 PM |
| Becky     | Sims     | becky.sims@example.com       | (678)-548-8286 | 12/17/21, 4:37 PM |
| Jessie    | Watkins  | jessie.watkins@example.com   | (659)-584-9684 | 12/17/21, 4:36 PM |
| Enrique   | Stevens  | enrique.stevens@example.com  | (997)-336-2185 | 12/17/21, 4:36 PM |
| Jesus     | Gibson   | jesus.gibson@example.com     | (383)-169-4773 | 12/17/21, 4:35 PM |

1 2

©2021 Copyright: Yulia Supruniuk

Рис.3.8 Перегляд користувачів.

Після того як диспетчер підтвердив замовлення, до нього доступ отримує водій, який має вказати час очікування та статус замовлення(рис. 3.9). Також, водій може реєструвати у системі свої машини та переглядати їх список, також видаляти(рис. 3.10).

The screenshot shows the 'My cars' section of the TaxiCall application. At the top, there's a navigation bar with 'TaxiCall', 'Orders', and 'My cars', along with a 'Logout' button. The main content area displays order details for a confirmed booking (indicated by a green checkmark and 'Approved' status). The details include:
 

- Destination address: 3011 Rockliff Tegenburg
- Passengers: 2
- Car type: universalis
- Empty trunk
- Animals
- A 'Take' button.
- Total payment: 124\$
- Waiting time: 12 minutes
- Status: executing (with a dropdown arrow)
- Select car: ChevroletCruze LT, AH43677 (with a dropdown arrow)
- A 'Take' button at the bottom.

 The footer contains the copyright notice: ©2021 Copyright: Yuliia Supruniuk.

Рис.3.9 Підтвердження замовлення.

The screenshot shows the 'Cars' section of the TaxiCall application. At the top, there's a navigation bar with 'TaxiCall', 'Orders', and 'My cars', along with a 'Logout' button. Below the navigation bar, there's a 'Cars' header with a 'New car' button. The main content area displays a list of available vehicles under the heading 'Total number of cars: 2'. The list contains two cars:
 

- 0M80M0**: Grand Prix Gtp, black, standart. Features: Air condition ✓, Terminal ✗, Empty trunk ✓, Animals ✗. Date/Time: 12/18/21, 8:49 PM. A 'Delete' button is at the bottom.
- LEH866**: VOLKSWAGENC C SPORT, white, comfort. Features: Air condition ✓, Terminal ✓, Empty trunk ✓, Animals ✗. Date/Time: 12/18/21, 5:26 PM. A 'Delete' button is at the bottom.

 The footer contains the copyright notice: ©2021 Copyright: Yuliia Supruniuk.

Рис.3.10 Список машин.

Після того, як водій виконав замовлення, він може залишити коментар та оцінку поїздки(рис. 3.11). Також, після завершення поїздки, клієнт також матиме змогу залишити відгук про водія(рис. 3.12).

**TaxiCall** Orders My cars Logout

Car type: universal

Empty trunk

Animals

Total payment: 124\$

Waiting time: 12 minutes

Status: executing

Car: ChevroletCruze LT, black(AH43677)

☐ ★ 
 ☐ ★★ 
 ☐ ★★★ 
 ☒ ★★★★ 
 ☐ ★★★★★

Everything was good

Set status

completed

Finish

©2021 Copyright: Yulia Supruniuk

Рис.3.11 Відгук водія.

**TaxiCall** Orders Logout

Animals

Grade order

Total payment: 124\$

Waiting time: 12 minutes

Status: executing

Car: ChevroletCruze LT, black(AH43677)

★★★★★

Driver's feedback: Everything was good

☐ ★ 
 ☐ ★★ 
 ☐ ★★★ 
 ☐ ★★★★ 
 ☒ ★★★★★

Driver was nice

Send

©2021 Copyright: Yulia Supruniuk

Рис.3.12 Відгук клієнта.

Щоб вийти зі свого облікового запису потрібно натиснути кнопку “Logout”. Якщо виникають якісь проблеми з доступом на сервер, то користувача перенаправить на сторінку помилки(рис. 3.13).



Рис.3.13 Сторінка помилки.

У адміністраторів є можливості переглядати інформацію про пасажирів, водіїв та диспетчерів. Також, змінювати користувачам ролі.

Весь функціонал додатку також можна протестувати за допомогою HTTP-клієнту для тестування - Postman.

## ВИСНОВКИ

В результаті виконання роботи розроблено програмне забезпечення для замовлення таксі. Під час виконання курсової роботи, згідно технічного завдання, був реалізований додаток «TaxiCall». За допомогою цього додатку, пасажери можуть замовляти таксі, диспетчери підтверджувати замовлення, водії приймати та виконувати їх.

У першому розділі виконано огляд існуючих альтернатив. Обґрунтовано вибір мови та середовища програмування, а також сформульовано постановку задачі проектування.

У другому розділі досліджено вимоги користувача та розроблено структуру бази даних сервісу таксі.

У третьому розділі представлено опис основних маршрутів до бекенду у вигляді таблиці. Також розроблено інструкцію користувача, в якій викладено проілюстровані рисунками основні вказівки, необхідні для роботи з програмою.

Під час розробки даного додатку я дослідила як реляційна база даних взаємодіє з користувачами через веб-ресурс.

У результаті виконання курсової роботи отримано програмний продукт, що відповідає технічному завданню.

У подальшому розроблене програмне забезпечення можливе бути удосконалене за рахунок покращення інтерфейсу користувача, рефакторингу коду, та розробки більш динамічних сторінок. Крім того, можливий перегляд логіки додатку та розширення функціоналу.



## СПИСОК ЛІТЕРАТУРИ

1. Uber [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uber.com>.
2. The fast, affordable way to ride. [Електронний ресурс] // Bolt – Режим доступу до ресурсу: <https://bolt.eu/>.
3. Моргунов Е. П. Язык SQL. Базовый курс: учеб.-практ. пособие. Москва : Postgres Professional, 2017. 256 с.
4. Node.js in Action / Mike Cantelon et al. Shelter Island : Manning Publications, 2014.
5. Jeremy Wilken. Angular in Action. Shelter Island : Manning Publications, 2018. 320 p.
6. Max Rehkopf. User stories with examples and a template [Електронний ресурс] / Max Rehkopf // Atlassian – Режим доступу до ресурсу: <https://www.atlassian.com/agile/project-management/user-stories>.
7. Volodymyr Bilyk. Data Flow Diagrams (DFD) Explained [Електронний ресурс] / Volodymyr Bilyk // Art of Business Analysis. – 2019. – Режим доступу до ресурсу: <https://www.artofba.com/post/data-flow-diagrams-dfd-explained>.

## ДОДАТОК А. ER-ДІАГРАМА

