

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методика організації спільних пасажирських
перевезень за допомогою хмарних технологій»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Дмитро БАЦУНОВ
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-62
_____ Дмитро БАЦУНОВ

Керівник: _____ Віктор ГРЕБЕНЮК
доктор філософії (PhD)

Рецензент: _____
науковий ступінь, _____ Ім'я, ПРІЗВИЩЕ
вчене звання

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____Ірина ЗАМРІЙ

«_____» _____2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____Бацунову Дмитру Сергійовичу

1. Тема кваліфікаційної роботи: «Методика організації спільних пасажирських перевезень за допомогою хмарних технологій»

керівник кваліфікаційної роботи Віктор ГРЕБЕНЮК, доктор філософії (PhD),
затверджені наказом Державного університету інформаційно-комунікаційних
технологій від «15» жовтня 2024 р. № 320.

2. Строк подання кваліфікаційної роботи «17» грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, алгоритми пошуку найкоротшого маршруту, алгоритм спільних пасажирських перевезень.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1. Дослідження методів побудови можливих маршрутів.
 2. Алгоритм визначення найкоротших шляхів.
 3. Розробка та валідація методу організації спільних пасажирських перевезень.
 5. Перелік ілюстративного матеріалу: *презентація*
 1. Мета, об'єкт та предмет дослідження.
 2. Актуальність роботи.
 3. Методика організації спільних пасажирських перевезень
 4. Алгоритм спільних пасажирських перевезень.
 5. Алгоритм пошуку маршруту.
 6. Математична модель організації спільних пасажирських перевезень.
 7. Програмні засоби реалізації.
 8. Практичний результат.
 9. Порівняння ефективності
 6. Дата видачі завдання «16» жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	16.10.2024	Виконано
2	Вивчення матеріалів для аналізу організації спільних пасажирських перевезень за допомогою хмарних технологій	20.10.2024	Виконано
3	Дослідження методів організації спільних пасажирських перевезень	23.10.2024	Виконано
4	Аналіз особливостей спільних пасажирських перевезень	03.11.2024	Виконано
5	Аналіз доступних хмарних технологій та перспективи їх інтеграції	20.11.2024	Виконано
6	Розробка системи для організації спільних пасажирських перевезень	06.11.2024	Виконано
7	Оформлення роботи: вступ, висновки, реферат	17.11.2024	Виконано
8	Розробка демонстраційних матеріалів	28.11.2024	Виконано
9	Попередній захист роботи	29.11.2024	Виконано

Здобувач вищої освіти

(підпис)

Дмитро БАЦУНОВ

Керівник
кваліфікаційної роботи

(підпис)

Віктор ГРЕБЕНЮК

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 70 стор., 4 табл., 22 рис., 11 джерел.

Мета – удосконалення процесу організації спільних пасажирських перевезень за допомогою хмарних технологій.

Об'єкт дослідження – процес організації спільних пасажирських перевезень.

Предмет дослідження – метод організації спільних пасажирських перевезень.

У межах магістерської роботи було розроблено низку алгоритмів, що забезпечують ефективну організацію спільних пасажирських перевезень. Зокрема, створено алгоритми для пошуку оптимальних маршрутів, які базуються на методах теорії графів, таких як формула Флойда – Уоршела. Використання сучасних технологій об'єктно-орієнтованого програмування дозволило створити модульну та масштабовану систему, що відповідає сучасним вимогам транспортної індустрії.

Особливу увагу приділено інтеграції з хмарними сервісами, які значно підвищують ефективність системи. Для отримання актуальних метеорологічних даних використовувалися сервіси OpenWeather, а для побудови маршрутів — API Sygic. Крім того, застосування хмарних технологій AWS, зокрема AWS SQS Queue для обробки запитів та AWS Aurora DB для управління базами даних, дозволило значно скоротити час обробки запитів і забезпечити стабільну роботу системи навіть за умов високого навантаження.

Проведено ґрунтовний аналіз сучасних підходів до організації спільних пасажирських перевезень, що дозволило виявити їх ключові особливості, переваги та обмеження. На основі проведених досліджень розроблено інноваційний метод, який враховує потреби користувачів та тенденції ринку транспортних послуг.

У рамках роботи реалізовано програмну систему, яка інтегрує розроблений метод для організації спільних пасажирських перевезень. Система створена із застосуванням технологій об'єктно-орієнтованого програмування, що забезпечує її модульність, гнучкість та масштабованість. У процесі розробки було особливу

увагу приділено оптимізації процесів побудови маршрутів та управління перевезеннями.

Проведені експерименти підтвердили ефективність запропонованого методу. Зокрема, було оцінено точність побудови маршрутів, продуктивність програмної системи та її здатність адаптуватися до динамічних змін у параметрах перевезень. Результати тестування порівняли з існуючими підходами, що дозволило отримати об'єктивну картину переваг нового рішення.

Результати дослідження підкреслюють значний потенціал використання розробленого методу в практичних умовах. Його ефективність у зменшенні витрат, підвищенні комфорту пасажирів та оптимізації процесів перевезень відкриває широкі можливості для впровадження у транспортній галузі. Таким чином, розроблений метод демонструє високу перспективність та може стати основою для подальших інновацій у сфері спільних пасажирських перевезень.

КЛЮЧОВІ СЛОВА: СПІЛЬНІ ПЕРЕВЕЗЕННЯ, МОЖЛИВІ МАРШРУТИ, НАЙКОРОТШІ ШЛЯХИ, ХМАРНІ ТЕХНОЛОГІЇ.

ABSTRACT

The text part of the qualification work for obtaining the master's degree: 70 pages, 4 table, 22 figures, 11 sources.

The purpose of the work is improve the methodology of organizing shared passenger transportation using cloud technologies.

The object of research is the process of organizing shared passenger transportation.

The subject of research is a method of organizing shared passenger transportation.

As part of the master's thesis, a series of algorithms were developed to ensure the efficient organization of shared passenger transportation. Specifically, algorithms for finding optimal routes were created, based on graph theory methods such as the Floyd-Warshall algorithm. The use of modern object-oriented programming technologies made it possible to develop a modular and scalable system that meets contemporary transportation industry requirements.

Particular attention was given to the integration of cloud services, which significantly enhance the system's efficiency. To obtain up-to-date meteorological data, OpenWeather services were utilized, and routes were built using the Sygic API. Additionally, the application of AWS cloud technologies, including AWS SQS Queue for request processing and AWS Aurora DB for database management, greatly reduced the request processing time and ensured the system's stable operation even under high loads.

A thorough analysis of modern approaches to organizing shared passenger transportation was conducted, allowing key features, advantages, and limitations to be identified. Based on the research, an innovative method was developed that considers user needs and market trends in transportation services.

Within the scope of the work, a software system implementing the developed method for shared passenger transportation was realized. The system was created using object-oriented programming technologies, ensuring its modularity, flexibility, and scalability. During development, particular emphasis was placed on optimizing the processes of route building and transportation management.

The experiments conducted confirmed the effectiveness of the proposed method. In particular, the accuracy of route planning, system performance, and its ability to adapt to dynamic changes in transportation parameters were evaluated. The testing results were compared with existing approaches, providing an objective view of the advantages of the new solution.

The research results underscore the significant potential for applying the developed method in practical conditions. Its efficiency in reducing costs, enhancing passenger comfort, and optimizing transportation processes opens up vast opportunities for its implementation in the transportation industry. Thus, the developed method demonstrates high prospects and could become the foundation for further innovations in the field of shared passenger transportation.

KEYWORDS: SHARED TRANSPORTATION, POSSIBLE ROUTES, SHORTEST PATHS, CLOUD TECHNOLOGIES.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1 ОСОБЛИВОСТІ ТА МЕХАНІЗМИ ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ	14
1.1 Актуальність спільних пасажирських перевезень	14
1.2 Особливості організації спільних пасажирських перевезень.....	15
1.3 Використання сучасних хмарних технологій при розробки методики спільних пасажирських перевезень.....	17
1.4 Аналіз наявних методик та їх функціональні можливості.....	22
1.5 Постановка завдання магістерського дослідження	27
2 РОЗРОБКА МЕТОДУ ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ ЗА ДОПОМОГОЮ ХМАРНИХ ТЕХНОЛОГІЙ.....	29
2.2 Вибір хмарних технологій	29
2.2.1 Отримання та передача даних до системи	29
2.2.2 Зберігання даних із можливістю масштабування з AWS Aurora DB Service.....	35
2.2.3 Використання метеорологічних даних OpeanWeather	37
2.2.4 Побудова маршруту разом з Sygic.....	42
2.3 Дослідження та визначення формул розрахунку	45
2.3.1 Вага ребра графа.....	45
2.3.2 Пошук найкоротшого шляху	47
2.4.2 Алгоритм пошуку маршруту	52
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ МЕТОДУ ОРГАНІЗВАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ	55
3.1 Вимоги до вхідних та вихідних даних додатку організації спільних пасажирських перевезень	55
3.2 Програмна реалізація та засоби реалізації	56
3.3 Дослідження ефективності запропонованої методики.....	63
ВИСНОВКИ	69
ПЕРЕЛІК ПОСИЛАНЬ	71
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GPS — глобальна система позиціонування, що дозволяє визначати місце розташування об'єкта на Землі за допомогою супутників.

gRPC — система віддаленого виклику процедур, розроблена Google, яка дозволяє клієнтам і серверам ефективно обмінюватися даними у вигляді повідомлень за допомогою протоколу HTTP/2.

REST API — інтерфейс програмування, що дозволяє обмінюватися даними між клієнтом і сервером через HTTP, використовуючи принципи REST-архітектури.

SDK — набір інструментів і бібліотек, призначений для розробки програмного забезпечення для певної платформи чи середовища.

UTC — всесвітній координований час, що використовується як стандартний час для визначення часу в усіх часових поясах.

API — інтерфейс програмування додатків, який дозволяє різним програмам взаємодіяти одна з одною.

ВСТУП

Сучасний транспорт активно розвивається завдяки впровадженню новітніх технологій. Одним із найперспективніших напрямків є впровадження платформ для організації спільних пасажирських перевезень, відомих як карпулінг або райдшаринг. Ці рішення не лише підвищують зручність для користувачів, а й допомагають оптимізувати транспортні потоки, зменшувати затори та мінімізувати негативний вплив на довкілля.

Зростання популярності цифрових платформ, зокрема хмарних технологій, відкрило нові можливості для автоматизації пошуку попутників, створення маршрутів, розрахунку вартості поїздок і управління пасажирськими перевезеннями. Сервіси, такі як Uber, BlaBlaCar та їх аналоги, здобули широке визнання в усьому світі, зокрема й в Україні. Їх головною перевагою є здатність швидко обробляти великі обсяги даних для оперативного прийняття рішень.

Проте реалізація технологій спільних перевезень супроводжується певними викликами, серед яких: забезпечення безпеки пасажирів, ефективне прокладання маршрутів, захист конфіденційних даних і адаптація систем до специфіки транспортної інфраструктури міст. Водночас ці платформи мають значний потенціал для сталого розвитку міського транспорту, зокрема через зниження кількості приватних автомобілів на дорогах.

У цій роботі аналізуються підходи до організації спільних пасажирських перевезень із застосуванням хмарних технологій. Основну увагу приділено вивченню існуючих рішень, розробці оптимізаційних алгоритмів для маршрутів та інтеграції подібних платформ у транспортну систему міст.

Сучасні транспортні системи, зокрема платформи спільних пасажирських перевезень, також зосереджені на використанні інноваційних підходів для покращення взаємодії з користувачами. Незважаючи на значні успіхи, багато таких систем орієнтовані на базові функції, як-от пошук маршрутів, оптимізація часу та розрахунок вартості поїздки.

З технічного боку вони часто обмежуються простими алгоритмами маршрутизації, мінімальними механізмами обробки даних і недостатньою адаптивністю до змінних умов.

Для ефективного впровадження платформ райдшарингу необхідно враховувати технічні виклики, такі як масштабованість, надійність обробки великих обсягів даних, забезпечення прозорості й точності роботи алгоритмів. З іншого боку, важливу роль відіграють практичні аспекти, як-от інтуїтивний інтерфейс користувача, зручність у користуванні та довіра пасажирів до системи. Для досягнення максимальної ефективності такі платформи повинні гармонійно поєднувати технічні рішення та практичні інновації.

У цьому дослідженні пропонується розробити унікальну методику спільних пасажирських перевезень, яка враховує сучасні вимоги до технологій і взаємодії з користувачами. Запропонована система вирізняється двома ключовими аспектами: впровадженням хмарних технологій для адаптивної оптимізації маршрутів і аналізу даних, а також просунутим розрахунком можливих та найкоротших шляхів.

Мета: удосконалення процесу організації спільних пасажирських перевезень за допомогою хмарних технологій.

Об'єкт дослідження: процес організації спільних пасажирських перевезень.

Предмет дослідження: метод організації спільних пасажирських перевезень.

Завдання:

1. Проаналізувати існуючі методи організації спільних пасажирських перевезень. Визначити недоліки наявних систем.
2. Визначення можливостей та обмеження методу.
3. Визначення набору вхідних даних.
4. Порівняння та вибір хмарних технологій задля розширення можливостей розробленого методу.
5. Розробка методики спільних пасажирських перевезень за допомогою хмарних технологій.
6. Створення прототипу за допомогою запропонованої методики інформаційної технології

7. Визначення напрямків подальшого удосконалення виконаного дослідження.

Методи дослідження: Розроблений метод включає інтеграцію з Sygic для картографічних даних і розрахунку оптимальних маршрутів, що забезпечує точну навігацію та стан трафіку. Для моніторингу погодних умов, які можуть впливати на вибір маршруту, вартість та час, використовується OpenWeatherMap, що дозволяє коригувати маршрути в реальному часі. Для обробки великих обсягів даних про пасажирів і перевезення застосовуються AWS (SQS для асинхронного обміну повідомленнями та Aurora для авто масштабованості бази даних, та ізоляції транзакцій на запис/читання). З метою оптимізації маршрутів розроблено власну формулу підрахунку ваги графа для точнішої оцінки доцільності та ефективності кожного з шляхів.

Наукова новизна: Розроблена методологія організації спільних пасажирських перевезень, яка базується на використанні хмарних технологій та вдосконаленому алгоритмі побудови і вибору найкоротших маршрутів.

Практичне значення: Розроблену методологію організації спільних пасажирських перевезень можна застосовувати перевізникам для оптимізації процесів управління перевезеннями та розширення доступу до нових сегментів аудиторії. З боку користувачів така система надає можливість більш економного та, потенційно, швидшого переміщення завдяки ефективному використанню транспортних ресурсів.

Апробація результатів магістерської роботи: Основні положення та результати дослідження доповідалися та обговорювалися на II міжнародної науково-практичної конференції «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії». В результаті було подано наступні тези доповідей:

- Бацунов. Д.С. Застосування хмарних технологій для покращення методу спільних пасажирських перевезень / Д.С. Бацунов // Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії : Міжнародній науково-практичній конференції. Збірник тез, чекає на

публікацію

- Бацунов. Д.С. Розробка методики організації спільних пасажирських перевезень на основі хмарних технологій / Д.С. Бацунов // Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії : Міжнародній науково-практичній конференції. Збірник тез, чекає на публікацію

1 ОСОБЛИВОСТІ ТА МЕХАНІЗМИ ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ

1.1 Актуальність спільних пасажирських перевезень

Спільні пасажирські перевезення є важливим елементом сучасних транспортних систем, що дозволяють знизити навантаження на міську інфраструктуру та забезпечити ефективне використання транспортних засобів. У багатьох великих містах, зокрема в Україні, проблема заторів на дорогах стала однією з основних причин погіршення якості життя та ефективності міської мобільності. Щоденні затори забирають значну кількість часу у пасажирів та водіїв, що призводить до збільшення витрат на паливо та підвищеного зносу транспортних засобів.

Крім того, зростання кількості автомобілів на дорогах без належної організації спільних перевезень веде до суттєвого забруднення повітря, що сприяє підвищенню рівня викидів парникових газів і шкідливих речовин. Це негативно позначається на здоров'ї мешканців міста, а також погіршує екологічну ситуацію в регіоні.

Актуальність впровадження методів спільних пасажирських перевезень зростає з кожним роком, оскільки вони сприяють значному зниженню кількості приватних автомобілів на дорогах, зменшенню заторів, а також дозволяють знизити викиди шкідливих речовин у атмосферу. Крім того, такий підхід забезпечує економічну вигоду для учасників перевезень, адже спільні поїздки дозволяють знизити витрати на транспорт, що є важливим аспектом у сучасних умовах високих цін на пальне і залежності від нафти.

Отже, організація спільних пасажирських перевезень є не тільки важливим кроком у покращенні екологічної ситуації, але й сприяє підвищенню ефективності транспортної інфраструктури, знижуючи економічні витрати на перевезення та вирішуючи низку соціальних проблем.

1.2 Особливості організації спільних пасажирських перевезень

Спільні пасажирські перевезення набувають все більшого поширення завдяки своїм унікальним особливостям, що дозволяють ефективно вирішувати багато транспортних, екологічних та економічних проблем сучасних міст. Однією з ключових особливостей такого підходу є можливість оптимального використання транспортних засобів. Замість того, щоб кожен пасажир користувався окремим автомобілем, спільні поїздки дозволяють зібрати кількох людей в одному транспортному засобі. Це знижує загальне навантаження на дороги та сприяє зменшенню заторів.

Важливим аспектом є також економічна вигода. Учасники спільних перевезень ділять витрати на паливе, технічне обслуговування та інші супутні витрати, що робить такі поїздки доступнішими. У часи, коли вартість пального постійно зростає, така економія стає суттєвим стимулом для вибору саме спільного транспорту.

Екологічна складова є ще одним вагомим аргументом на користь організації спільних перевезень. Зменшення кількості автомобілів на дорогах призводить до зниження рівня викидів парникових газів і шкідливих речовин в атмосферу. Це позитивно впливає на загальний стан екології, знижуючи вплив на зміну клімату, а також сприяє покращенню якості повітря, яким дихають мешканці міст.

Спільні пасажирські перевезення можна умовно розділити на дві основні групи: приватні та публічні. Публічні перевезення, такі як міські автобуси або приміські потяги, мають фіксовані маршрути та зупинки, які не адаптуються під індивідуальні потреби пасажирів. Це дозволяє підтримувати нижчу вартість послуг, але знижує гнучкість і комфорт для користувачів. У свою чергу, приватні перевезення орієнтовані на індивідуальний підхід, де кожен маршрут може бути налаштований під конкретного клієнта. Ця персоналізація робить приватні перевезення більш дорогими через значні витрати, пов'язані з адаптацією маршрутів, витратами на паливо та оплатою праці водіїв.

Організація пасажирських перевезень, зокрема приватних, повинна

враховувати безліч важливих аспектів для забезпечення ефективності та прибутковості системи. Одним із ключових факторів є погодні умови, які можуть впливати на швидкість пересування, безпеку та витрати на обслуговування транспорту. Наприклад, ускладнення через снігопади або дощі можуть збільшувати тривалість маршрутів і витрати на паливо. Це повинно бути враховано ще до початку поїздки.

Ще одним важливим аспектом є навантаження на дорожню інфраструктуру. Трафік може суттєво впливати на швидкість руху транспорту, а також на вартість обслуговування транспортних засобів. Для уникнення заторів система має ефективно аналізувати навантаження на дороги й оптимізувати маршрути.

Також необхідно враховувати якість дорожнього покриття, особливо якщо маршрут включає віддалені або сільські зони. Погані дороги можуть призводити до зношування транспорту, збільшення витрат на його обслуговування та незадоволення пасажирів.

Не менш важливим є аналіз попиту на маршрути. Наявність чи відсутність замовлень на конкретному маршруті впливає на доцільність його виконання. Запуск маршруту без достатнього пасажиропотоку може призвести до фінансових втрат. Нарешті, розподіл доступних ресурсів, таких як кількість водіїв, транспортних засобів та фінансів, відіграє ключову роль у підтримці стабільності системи перевезень. Недостатність ресурсів може призвести до погіршення якості послуг і втрати клієнтів, а їх надмірність — до нераціональних витрат.

Таким чином, ігнорування хоча б одного з перелічених аспектів може призвести до значних фінансових втрат для системи пасажирських перевезень і зниження її конкурентоспроможності. Ефективна організація вимагає комплексного підходу, який охоплює всі ці фактори для забезпечення надійності, зручності та економічної вигоди.

1.3 Використання сучасних хмарних технологій при розробки методики спільних пасажирських перевезень

Сучасні хмарні технології значно підвищують ефективність організації спільних пасажирських перевезень, забезпечуючи швидкий доступ до даних, масштабованість систем, адаптацію до змін і гнучку обробку різноманітних інформаційних потоків. Завдяки хмарним сервісам різні джерела даних об'єднуються в єдину платформу, що гарантує стабільну та безперебійну роботу системи в реальному часі. Це дозволяє оптимізувати маршрути, скорочувати час очікування для пасажирів і підвищувати загальний рівень обслуговування.

Одним із ключових чинників успіху є збір та аналіз даних про дорожній трафік і стан доріг. Ця інформація дозволяє створювати маршрути з урахуванням місцезнаходження водія та пасажирів, а також визначати оптимальний шлях залежно від поточних умов. Завдяки трьом основним вхідним даним — стану доріг, локації водія і пасажирів — можна швидко побудувати маршрут і оцінити його ефективність. У сучасних мегаполісах швидкість і точність рішень відіграють вирішальну роль, адже навіть кілька секунд можуть суттєво вплинути на успіх бізнесу. Швидке прокладання оптимального маршруту стає запорукою ефективності як обраного методу, так і його впровадження. Саме тому вибір хмарного постачальника даних трафіку та стану доріг є одним із найважливіших питань, в рамках огляду потенційних кандидатів на використання в методиці.

Проте є декілька проблем які потребують врахування, умовно поділивши їх на технічні критерії і нетехнічні у виборі постачальника даних з хмари. Технічні критерії стосуються функціональних можливостей сервісу та його здатності інтегруватися в систему. До них можна віднести:

1. Формат даних

Постачальники можуть використовувати різні формати даних для передачі інформації (JSON, XML, CSV тощо). Сумісність формату даних з вашою системою є критично важливою для безперебійного функціонування сервісу.

1. Вид комунікації

Вибір типу зв'язку між вашим додатком і сервісом впливає на швидкість передачі даних та стійкість до помилок. Серед основних видів

REST API — стандартний метод взаємодії через HTTP-протокол.

WebSocket — забезпечує двосторонній зв'язок для оперативного оновлення даних.

gRPC — ефективний і швидкий протокол для високопродуктивної взаємодії.

2. Development Kit

Наявність SDK спрощує інтеграцію сервісу з вашим додатком. Деякі постачальники надають готові бібліотеки для роботи з їхнім API для популярних мов програмування (JavaScript, Python, Java тощо), що суттєво прискорює розробку.

3. Масштабованість

Хмарна платформа має підтримувати динамічне збільшення обсягів оброблюваних даних та кількості запитів у реальному часі, щоб уникнути збоїв у роботі під час пікових навантажень

4. Стійкість до збоїв

Важливо враховувати, чи забезпечує постачальник відмовостійкість, зокрема резервне копіювання та автоматичне перемикання на інші сервери у разі збою.

Та нетехнічні аспекти, а саме:

1. Покриття даних про дорожню інфраструктуру

- Важливо з'ясувати, чи працює сервіс з певними географічними регіонами (наприклад, чи враховує тільки міські дороги або також приміські маршрути).

- Чи включені в покриття нові дороги, тимчасові обмеження руху або ремонти.

2. Тип дорожньої інфраструктури

Деякі сервіси працюють найкраще з магістральними дорогами, але можуть не враховувати провулки чи другорядні шляхи, що може бути критичним для точності маршрутів.

3. Врахування якості доріг

Чи аналізує платформа стан дорожнього покриття, затори або інші фактори, що впливають на швидкість пересування.

4. Юридичні обмеження

У деяких регіонах існують обмеження на використання певних даних, наприклад, картографічних або трафікових сервісів. Постачальник повинен відповідати місцевим законодавчим нормам

Базуючись на цих можливостях і повинен бути обран постачальник даних щодо стану доріг.

Окрім інформації про стан доріг важливо роздивитись використання даних про погодні умови. Вони дозволяють прогнозувати можливі ризики, забезпечувати безпеку учасників руху, оптимізувати маршрути та зменшувати час у дорозі. У реаліях сучасних мегаполісів, де кожна хвилина має значення, точність і швидкість прийняття рішень є вирішальними факторами.

Хмарні технології забезпечують надійний доступ до актуальної метеорологічної інформації, підтримку високих навантажень і зручну інтеграцію з іншими системами. Для вибору відповідного сервісу збору та обробки погодних даних важливо оцінити як технічні, так і організаційні критерії.

Прогнозування погодних умов значно впливає на планування і організацію спільних перевезень. Завдяки використанню хмарних сервісів, можна завчасно отримувати дані про можливі зміни погоди, такі як опади, туман чи екстремальні температури. Це дозволяє мінімізувати ризики, уникати потенційно небезпечних маршрутів і коригувати графіки руху. Крім того, точні прогнози сприяють ефективному розподілу ресурсів, наприклад, підготовці транспорту до роботи в несприятливих умовах або оптимізації витрат на паливо. Своєчасний доступ до цих даних забезпечує стабільність роботи та підвищує рівень довіри клієнтів до методики.

Окрім інтеграції з постачальниками даних, важливою задачею є отримання геолокації безпосередньо від користувачів методики, таких як водій та пасажир. У вирішенні цього питання нам теж допоможуть хмарні технології. Ці дані є основою для побудови оптимальних маршрутів у реальному часі, особливості збору локації водія вимагають використання неблокуючої комунікації, а саме так званих черг

повідомлення. Оскільки ця інформація не є критичною, і у випадку перевантаження системи може бути оброблена згодом. Приклад архітектури черги повідомлень зображений на Рисунку 1.1.

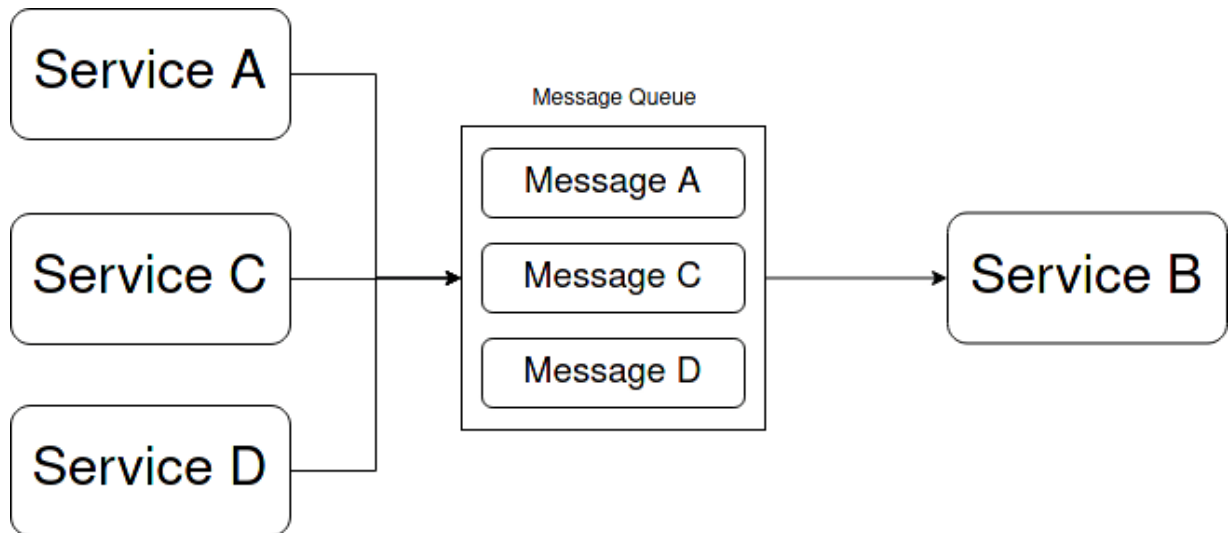


Рис. 1.1. Архітектура черги повідомлення з трьома продюсерами

Чому саме неблокуюча комунікація? Відповідь на це питання полягає у:

1. Динамічність геолокаційних даних

Локація водія змінюється постійно, іноді навіть кілька разів на секунду. Асинхронна модель дозволяє обробляти оновлення без затримок, незалежно від інших процесів. Це критично важливо для точного та своєчасного побудови маршрутів.

2. Оптимізація швидкості обробки

- У синхронній моделі обробка запиту блокує основний потік до отримання відповіді, що може створювати затримки.

- Асинхронна модель, навпаки, дозволяє надсилати запити та отримувати відповіді у фоновому режимі, забезпечуючи безперервну роботу інших частин системи.

3. Стійкість до нестабільного зв'язку

У випадку тимчасової втрати зв'язку (наприклад, якщо водій знаходиться в зоні

зі слабким покриттям мережі) асинхронна модель дає змогу повторно надсилати запит або отримувати дані, як тільки з'єднання буде відновлено. Це мінімізує ризики втрати важливої інформації.

4. Масштабованість

У системах із великою кількістю користувачів (водіїв і пасажирів) асинхронна комунікація дозволяє обробляти тисячі запитів одночасно. Це критично для забезпечення стабільної роботи сервісу під час пікових навантажень.

Використання асинхронної неблокуючої комунікації дозволяє не тільки забезпечити точність і швидкість побудови маршрутів, а й створити масштабовану, надійну й адаптивну систему. Це закладає фундамент для якісного користувацького досвіду та стабільної роботи сервісу навіть у складних умовах.

Система, яка оперує з великими обсягами даних для операцій читання і запису, вимагає використання сучасних підходів до організації цих процесів з використанням сучасних хмарних технологій. Система організацій спільних пасажирських перевезень є такою, оскільки від одного користувача ми можемо отримати і 1000 повідомлень в день. Для цього важливо застосовувати сервіс, який дозволяє розділяти транзакції на операції читання і запису, а також автоматично масштабувати ресурси в залежності від навантаження. Такий підхід має кілька важливих переваг.

По-перше, розподіл операцій читання та запису зменшує навантаження на основну базу даних. Операції запису можуть оброблятися окремо, що забезпечує їхню точність і надійність, а операції читання можна перенаправляти на репліки бази, оптимізовані для швидкого доступу.

Це дозволяє уникнути блокувань і конфліктів між одночасними транзакціями, гарантуючи безперебійну роботу системи.

По-друге, такий поділ транзакцій сприяє підвищенню продуктивності. Наприклад, реплікація даних на кілька серверів для операцій читання зменшує час доступу до інформації навіть під час високих навантажень. Водночас, транзакції запису не впливають на швидкість читання даних.

Автоматичне масштабування забезпечує гнучкість системи, дозволяючи їй адаптуватися до змін обсягів запитів. Якщо навантаження зростає, система

автоматично додає додаткові ресурси, а коли навантаження зменшується — звільняє непотрібні, що оптимізує витрати. Це критично важливо для забезпечення стабільної роботи під час пікових навантажень, наприклад, у моменти різкого збільшення кількості користувачів або обробки великих обсягів даних. Такий підхід дозволяє створити ефективну, масштабовану і надійну інфраструктуру для організації спільних пасажирських перевезень.

1.4 Аналіз наявних методик та їх функціональні можливості

На момент написання даної роботи існує кілька розроблених методик для організації спільних перевезень, які вже набули популярності на ринку. Однак, для більш детального аналізу та порівняння, доцільно сфокусуватися на двох найбільш відомих та широко використовуваних платформах у своїх напрямках: Uber та BlaBlaCar. Кожна з цих платформ має свої унікальні функціональні можливості, орієнтуючись на різні аспекти потреб користувачів.

Uber, як один із лідерів у сфері спільних поїздок, зосереджений на наданні послуг таксі та організації поїздок у межах міст. Це рішення орієнтоване на швидкість, зручність та доступність, що робить його популярним серед користувачів, які потребують миттєвого транспорту. Платформа підтримує інтеграцію з картографічними сервісами для точного визначення маршруту, а також застосовує алгоритми для оптимізації маршруту з урахуванням дорожніх умов та місця розташування водія і пасажирів.

Uber використовує комплексний набір технологій для роботи з картографією та алгоритмами, що дозволяють оптимізувати маршрути, враховувати дорожні умови та забезпечити швидке обслуговування клієнтів.

Основним джерелом картографічних даних для Uber є інтеграція з такими популярними сервісами, як Google Maps і Mapbox. Ці сервіси надають точні карти, а також інформацію про трафік і дорожні умови в реальному часі. Завдяки цьому, Uber може точно визначати місцезнаходження водіїв і пасажирів, а також враховувати затори або дорожні обмеження, щоб оптимізувати маршрути.

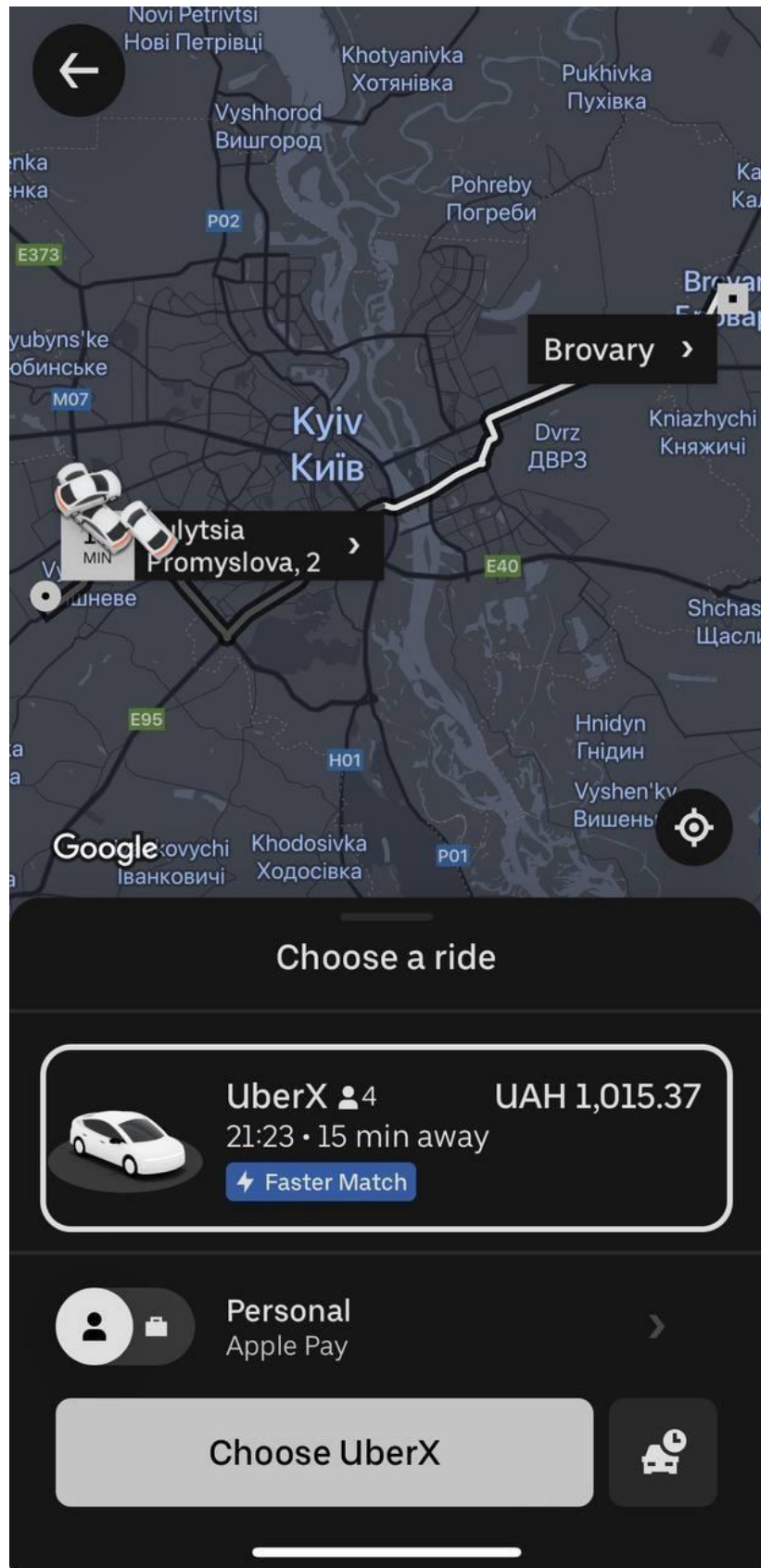


Рис. 1.2. Приклад прокладеного маршруту Uber

Для побудови маршрутів Uber використовує складні алгоритми, серед яких популярними є алгоритми пошуку найкоротшого шляху, такі як алгоритм Дейкстри. Цей алгоритм дозволяє знаходити найбільш оптимальний маршрут з точки А в точку В, враховуючи не тільки відстань, а й час, який знадобиться на подолання цього шляху, з урахуванням дорожніх умов і заторів. Принцип роботи алгоритма який використовує Uber – Дейкстри зображений на Рисунку 1.3.

В основі роботи алгоритму лежить пріоритетна обробка вузлів, де кожному вузлу надається значення найменшої знайденої вартості шляху від початкового вузла. Алгоритм повторює процес оцінки вузлів, поступово "розповсюджуючи" інформацію про найкоротші шляхи, доки всі вузли графа не будуть оброблені. Це забезпечує ефективне знаходження оптимальних маршрутів навіть у складних транспортних мережах.

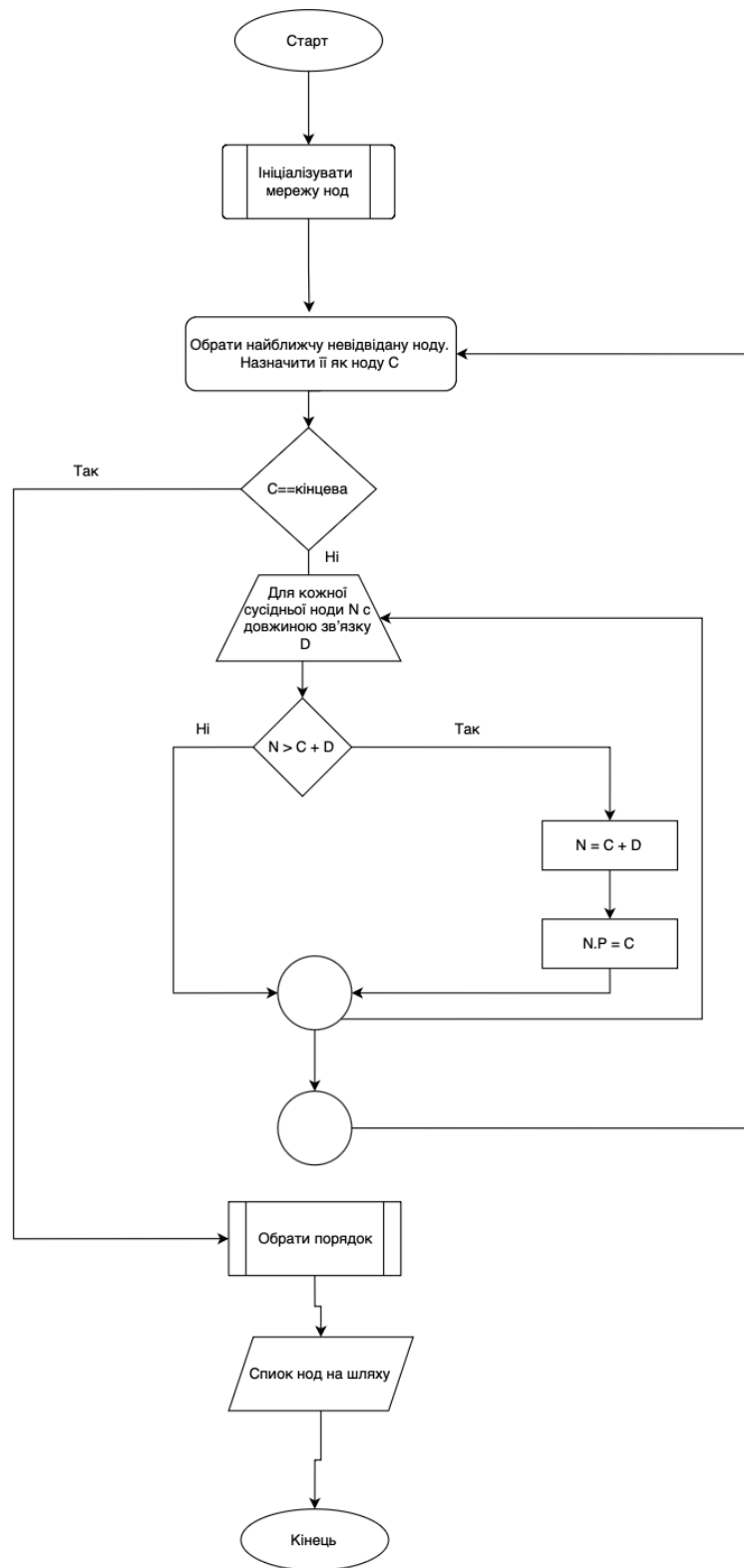


Рис. 1.3. Алгоритм Дейкстри

Оскільки ситуація на дорогах може змінюватися, Uber також активно застосовує динамічне оновлення маршруту. Це означає, що система може коригувати маршрут в реальному часі на основі нових даних про трафік або інші зміни на дорозі.

BlaBlaCar, в свою чергу, спеціалізується на міжміських поїздках, надаючи користувачам можливість знайти попутників для спільних поїздок на довгі відстані.

Ця платформа є зручним рішенням для тих, хто хоче заощадити на поїздках та готовий ділити транспорт з іншими людьми. BlaBlaCar пропонує не тільки пошук попутників, але й систему рейтингів та відгуків для забезпечення довіри між водієм та пасажиром. Приклад побудованого маршруту Одеса- Київ зображено на Рисунку 1.4.

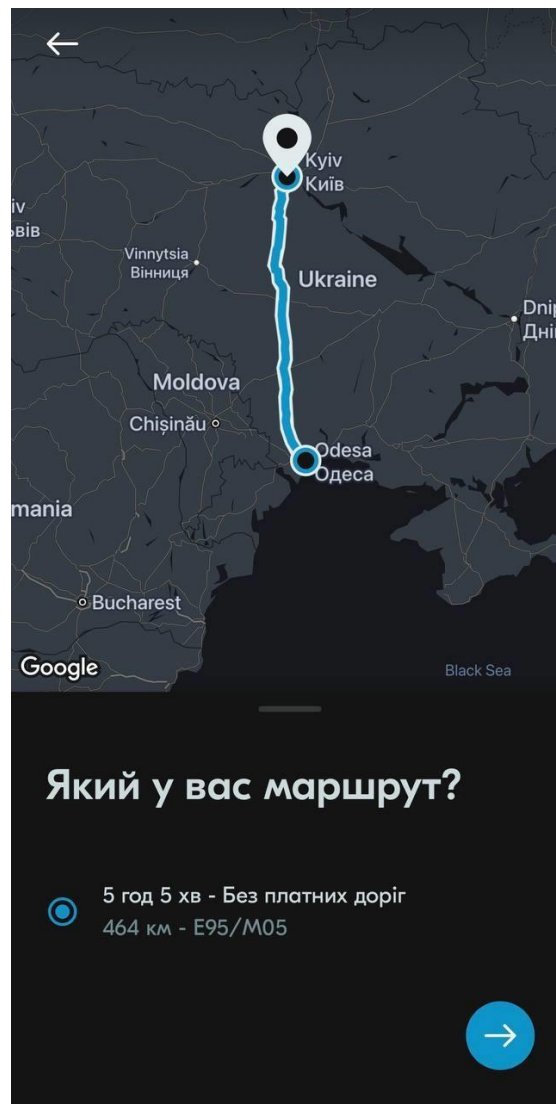


Рис. 1.4. Приклад побудованого маршруту BlaBlaCar

Для побудови оптимальних маршрутів Blablacar використовує алгоритми, схожі на алгоритми пошуку найкоротшого шляху, які дозволяють знаходити ефективні маршрути між різними містами. Вони враховують не лише відстань, але й інші важливі аспекти, як-от час прибуття в пункт призначення та зручність для всіх учасників поїздки.

Якщо ж говорити про алгоритми, то Blablacar застосовує моделі для прогнозування попиту на поїздки, що дозволяє ефективно управляти кількістю доступних місць у різних автомобілях. Алгоритми також допомагають визначати оптимальні ціни для поїздок, щоб забезпечити привабливу вартість для пасажирів, водіїв і платформу загалом.

У загальному, Blablacar спрямований на оптимізацію міжміських поїздок, використовуючи сучасні картографічні та алгоритмічні технології, що забезпечують зручність, безпеку та ефективність організації спільних подорожей.

Обидві платформи мають власні особливості та переваги, тому при розробці нової методики доцільно враховувати їхній досвід, зокрема, у таких аспектах як інтеграція з картографічними та трафіковими даними, алгоритми оптимізації маршрутів, а також методи взаємодії між користувачами платформи.

1.5 Постановка завдання магістерського дослідження

Ця магістерська робота спрямована на розробку методики організації спільних пасажирських перевезень за допомогою хмарних технологій. Метою дослідження є покращення ефективності пасажирських перевезень шляхом застосування інноваційних рішень на основі хмарних технологій, що дозволяє автоматизувати процеси планування, управління маршрутами та моніторингу перевезень. Запропонована методика орієнтована на створення гнучкої, масштабованої та ефективної системи для організації спільних поїздок, яка буде доступна для користувачів через хмарну платформу.

Основною метою цього дослідження є впровадження хмарних технологій у процес організації спільних пасажирських перевезень для підвищення ефективності та зручності для всіх учасників процесу — пасажирів і перевізників. Для досягнення зазначеної мети були встановлені такі завдання:

1. Аналіз існуючих методик організації пасажирських перевезень. Визначити основні недоліки традиційних підходів та існуючих рішень у сфері спільних перевезень, таких як Uber та BlaBlaCar.

2. Розробка вимог до системи. Визначити обмеження та вимоги до функціональності та безпеки хмарної платформи для організації спільних перевезень.

3. Оцінка можливостей використання хмарних технологій для організації спільних перевезень. Визначити, як хмарні сервіси та платформи можуть покращити процеси зберігання та обробки даних, а також оптимізацію маршрутів.

4. Розробка алгоритмів та методів для ефективного планування маршрутів і управління перевезеннями. Використання хмарних технологій для оптимізації маршрутів, управління замовленнями та комунікації між водіями та пасажирами.

5. Створення прототипу хмарної платформи для організації спільних перевезень. Розробка інтерфейсу та функціональних модулів для зручності користувачів, включаючи реєстрацію, пошук маршруту, бронювання поїздок та оцінку сервісу.

6. Оцінка ефективності запропонованої методики. Проведення тестування платформи в реальних умовах і оцінка результатів щодо зручності використання, економічної ефективності та зменшення заторів.

Запропонована методика дозволить покращити організацію спільних пасажирських перевезень, знижуючи витрати, підвищуючи зручність для пасажирів та водіїв, а також оптимізуючи використання транспортних ресурсів через інтеграцію хмарних технологій. В додаток до цього розроблена методика буде набагато універсальнішою в порівнянні з Uber та BlaBlaCar оскільки буде можливість прокладати маршрути будь-якої довжини, а не обмежуватись тільки тривалими або короткостроковими.

2 РОЗРОБКА МЕТОДУ ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ ЗА ДОПОМОГОЮ ХМАРНИХ ТЕХНОЛОГІЙ

2.1 Постановка задачі

Задача полягає в розробці повноцінного методу організації спільних пасажирських перевезень, який відповідатиме сучасним вимогам до таких систем. Цей метод має враховувати метеорологічні фактори, фактори завантаження доріг та якості дорожнього покриття, а також пропонувати більше функціональних можливостей порівняно з існуючими конкурентами. Крім того, система повинна забезпечити точні розрахунки часу та ціни, які будуть надані пасажиром.

Окрім безпосередньої розробки методу, важливим завданням є визначення хмарних технологій, які зможуть значно покращити наявні методи організації перевезень. Врахування новітніх хмарних рішень дозволить підвищити ефективність системи, забезпечити масштабованість, інтеграцію з іншими сервісами та поліпшити обробку великих обсягів даних. Також хмарні технології можуть сприяти оптимізації маршрутизації, моніторингу транспортних засобів в реальному часі, зменшенню витрат на інфраструктуру і поліпшенню взаємодії з користувачами.

Тому метою є створення інноваційної, високопродуктивної системи, яка буде надавати користувачам найкращий досвід перевезень, пропонуючи оптимізовані маршрути, точні прогнози та конкурентоспроможні ціни.

2.2 Вибір хмарних технологій

2.2.1 Отримання та передача даних до системи

Оскільки в системі спільних пасажирських перевезень є багато водіїв які постійно у русі, виникає необхідність оновлення локації кожного з водіїв, задля відстежування локації і в подальшому урахуванні при запиті на поїздку. Правильний вибір системи отримання даних та передачі даних в БД є важливим етапом у створенні ефективної платформи для організації спільних пасажирських перевезень. Від неї

залежить те, наскільки швидко та надійно система зможе отримувати інформацію про запити пасажирів, стан транспортних засобів, швидкість яку притримується водій, вся ця інформація впливає на роботу платформи.

В отриманні даних сервером необхідно мати два актора, перший це пристрій водія який буде відправляти повідомлення, зазвичай всі сучасні смартфони мають GPS тому постачальник даних не повинно бути проблемою. На стороні водія система автоматично буде відправляти повідомлення з заданими параметрами, а другий зі сторони серверу - задача правильно отримати і обробити їх, враховуючи можливі перенавантаження повідомлення повинні автоматично кластись в чергу. Черга повідомлень - це програмний механізм для асинхронного обміну повідомленнями між різними компонентами системи. Черга зберігає повідомлення, поки їх не буде оброблено, забезпечуючи надійну передачу даних навіть у разі тимчасової недоступності одного зі складників.

Таким чином, інтеграція правильної системи отримання даних робить метод спільних пасажирських перевезень більш організованим, стабільним і клієнтоорієнтованим, що є ключовим для його успіху та подальшого розвитку. Організація спільних пасажирських перевезень передбачає оновлення локація для кожного з водіїв. Кожен транспортний засіб у системі повинен відправляти свої евеннти (повідомлення) для оновлення статусу та локації. Ці евеннти дозволяють платформі в режимі реального часу отримувати інформацію про стан транспортних засобів, зокрема про їх локацію, стан на маршруті та інші важливі параметри. Таким чином, система може приймати обґрунтовані рішення, оптимізувати маршрути та забезпечувати ефективну координацію між водіями, пасажирами та диспетчерами.

Оскільки ми підтвердили важливість розуміння місцезнаходження водія, тепер саме час зрозуміти що саме повинен відправляти пристрій водія до нашої системи.

```

1  {
2      "event_type": "location_update",
3      "vehicle_id": "ABC123",
4      "driver_id": "D001",
5      "latitude": 49.2827,
6      "longitude": 123.1216,
7      "timestamp": "2024-12-09T14:30:00Z",
8      "speed": 60,
9      "status": "en_route",
10     "route_id": "R001"
11 }

```

Рис. 2.1. Необхідні дані в повідомленні оновлення локації

На Рисунку 2.1 зображено JSON повідомлення яке буде відправлено в систему водієм для оновлення локації в БД.

де:

event_type - тип події, в даному випадку - оновлення локації;

vehicle_id - унікальний ідентифікатор транспортного засобу;

driver_id - ідентифікатор водія, який управляє транспортним засобом;

latitude та **longitude** - географічні координати поточної локації транспортного засобу;

timestamp - час, коли було зафіксовано оновлення локації;

speed - поточна швидкість транспортного засобу;

status - поточний стан транспортного засобу (наприклад, "en_route" - у дорозі, "arrived" - прибув, "delayed" - затримка тощо);

route_id - ідентифікатор маршруту, по якому рухається транспортний засіб;

Важливим відступленням буде додати розмір такого повідомлення, оскільки це одна з найважливіших показників при виборі брокера повідомлень між системою та водієм.

Розмір усіх рядків: 21 (eventType) + 10 (eventId) + 20 (timestamp) + 10 (driverId) + 12 (vehicleId) + 6 (status) = 79 байт.

Розмір всіх числових значень: 4 (latitude) + 4 (longitude) + 4 (speed) + 4 (heading) + 4 (accuracy) = 20 байт.

Загальний розмір повідомлення : 79 байт (стрічки) + 20 байт (числа) = 99 байт.

Після того як ми зрозуміли об'єм та тип даних ми повинні роздивитись кілька варіантів хмарних черг повідомлень. Ми розглянемо найбільш популярні сервіси черг повідомлень такі як: **Amazon SQS, Google Cloud Pub/Sub та Azure Queue Storage.** Порівняємо їх за кількома критичними параметрами.

1. Amazon SQS (Simple Queue Service).

- **Переваги:**

- a) **Масштабованість:** Amazon SQS автоматично масштабується під навантаження без необхідності налаштовувати інфраструктуру.
- b) **Надійність:** підтримує гарантовану доставку повідомлень із повторною доставкою в разі помилки.
- c) **Легкість інтеграції:** вбудовується з іншими сервісами AWS, такими як Lambda для обробки повідомлень і DynamoDB для зберігання даних.
- d) **Простота налаштування:** проста і швидка інтеграція без необхідності складного налаштування.
- e) **Вартість:** доступні тарифи з оплатою за використання, що дозволяє зменшити витрати на інфраструктуру.

- **Недоліки:**

- a) Обмеження на максимальний розмір повідомлень — 256 КБ. Для великих повідомлень потрібно використовувати додаткові рішення.

2. Google Cloud Pub/Sub.

- **Переваги:**

- a) **Масштабованість:** Google Cloud Pub/Sub підтримує майже безмежну масштабованість завдяки своєму хмарному характеру.

- b) **Підтримка потокової обробки:** ідеально підходить для обробки великих потоків даних і передачі їх до різних підсистем.
- c) **Висока доступність:** має вбудовану реплікацію для забезпечення високої доступності.
- d) **Інтеграція з іншими сервісами Google Cloud:** легко інтегрується з іншими сервісами екосистеми Google Cloud.

- **Недоліки:**

- a) Трохи складніша інтеграція з не-Google хмарними сервісами.
- b) Потребує додаткових налаштувань для забезпечення високої пропускної здатності при великій кількості підписників.

3. Azure Queue Storage.

- **Переваги:**

- a) **Інтеграція з іншими сервісами Azure:** підходить для компаній, які використовують інфраструктуру Microsoft Azure.
- b) **Простота налаштування:** проста в налаштуванні черга для передачі повідомлень, що підтримує сховище даних.
- c) **Масштабованість:** автоматично масштабується під навантаження.
- d) **Вартість:** має вигідну модель ціноутворення з оплатою за використання.

- **Недоліки:**

- a) Обмеження на максимальний розмір повідомлення — 64 КБ.
- b) Можливості для обробки поточкових даних не так розвинені, як у Google Pub/Sub або SQS.

Ключові критерії для вибору черги повідомлень для організації спільних пасажирських перевезень:

1. Масштабованість.

- **Amazon SQS** та **Google Cloud Pub/Sub** забезпечують високий рівень масштабованості, що є критично важливим для платформи спільних пасажирських перевезень. Вони дозволяють обробляти мільйони повідомлень без перевантажень системи.
- **Azure Queue Storage** також масштабується, але не має такої потужної підтримки потокових даних, як інші варіанти.

2. Надійність та доставка повідомлень.

- **Amazon SQS** підтримує гарантовану доставку повідомлень, забезпечуючи високу надійність, що важливо для передачі важливих подій, таких як зміни статусу запиту пасажирів чи івенти від водіїв.
- **Google Cloud Pub/Sub** має високу доступність завдяки внутрішній реплікації, що також є перевагою для забезпечення надійності.
- **Azure Queue Storage** забезпечує достатній рівень надійності, але не має таких функцій, як SQS.

3. Інтеграція з іншими сервісами.

- **Amazon SQS** найкраще інтегрується з іншими сервісами AWS, що дозволяє побудувати гнучку й потужну платформу для обробки подій.
- **Google Cloud Pub/Sub** має потужну інтеграцію з Google Cloud сервісами, що робить його ідеальним для організацій, вже використовують цю екосистему.
- **Azure Queue Storage** є гарним варіантом для компаній, що використовують Microsoft Azure.

4. Простота використання та налаштування.

- **Amazon SQS** є найбільш зручним варіантом завдяки своїй простоті та інтеграції з іншими сервісами AWS.

- **Google Cloud Pub/Sub** потребує дещо більше налаштувань для досягнення максимального потенціалу.
- **Azure Queue Storage** також має просте налаштування, але дещо обмежений у можливостях порівняно з іншими двома рішеннями.

Після ретельного аналізу та порівняння я дійшов до висновку що **Amazon SQS** є оптимальним вибором для системи отримання даних для організації спільних пасажирських перевезень завдяки своїй надійності, масштабованості, простоті інтеграції з іншими хмарними сервісами та здатності гарантувати доставку повідомлень навіть при високих навантаженнях. Він забезпечує найкращий баланс між функціональністю, простотою використання та вартістю, що робить його ідеальним варіантом для цієї задачі. Одним з відомих недоліків **SQS Queue** є обмеження повідомлення в 256 КБ але оскільки повідомлення від пристрою водія містить 100 – 200 байт це абсолютно ніяк не перешкодить процесу отримання даних. Тому дивлячись на переваги і недоліки саме в нашому випадку вибір пав на хмарну технологію **AWS SQS** сервіс.

2.2.2 Зберігання даних із можливістю масштабування з AWS Aurora DB Service

Після визначення брокера повідомлення наступним логічним кроком є вибір провайдера бази даних, який відіграє вирішальну роль у забезпеченні її стабільності та ефективності методу. З розвитком технологій і зростанням кількості користувачів такі системи стають дедалі складнішими, адже обробляють значні обсяги інформації в режимі реального часу. До цієї інформації належать дані про користувачів, водіїв, транспортні засоби, запити на поїздки, маршрути, а також численні події, що постійно виникають у системі.

Ефективна база даних у такому випадку повинна забезпечувати високу продуктивність, масштабованість, безпеку та надійність. Окрім цього, вона має бути здатною адаптуватися до змін у навантаженні та обсягів даних, адже активність користувачів може змінюватися залежно від часу доби чи інших чинників.

Бази даних діляться на реляційні та нереляційні, оскільки в розробляємій методиці складні зв'язки між сутностями то вибор очевидний, нам необхідно чітко відстежувати взаємозв'язки між користувачами, транспортними засобами, маршрутами та статусами. це реляційна база даних.

Крім цього вбачається що в організації спільних пасажирських перевезень можливе велике навантаження, багато записів в базу, і змінююча нагрузка на читання, наприклад вночі активність повідомлень від водіїв(записи) не повинні зменшуватись, в той самий час транзакції на читання даних вночі не такі активні порівнюючи день.

Таблиця 2.1

Порівняння сховищ даних

Назва	AWS Aurora	MySQL	PostgreSQL
Тип реплікації	Майже синхронна	Асинхронна / Синхронна	Асинхронна / Синхронна / Логічна
Кількість реплік	До 15	Обмежено продуктивністю	Залежить від налаштувань
Розподілення навантаження	Автоматичне (з підтримкою балансувальника)	Потребує ручного налаштування	Потребує ручного налаштування
Failover	Автоматичний	Вимагає додаткових рішень	Вимагає сторонніх інструментів

Розглянувши всі вищезазначені вимоги, вибір пав на AWS Aurora DB Service як ідеальне рішення для організації зберігання даних у системі спільних пасажирських перевезень. Aurora є сучасною хмарною базою даних, яка поєднує в собі переваги

традиційних реляційних баз даних (таких як MySQL і PostgreSQL) із високою продуктивністю та масштабованістю хмарних рішень.

Ця база даних дозволяє легко масштабуватися як у вертикальному (збільшення потужностей), так і в горизонтальному напрямках (додавання реплік для обробки читання). Окрім цього, Aurora підтримує високу ізоляцію транзакцій, що забезпечує коректну роботу системи навіть у випадках пікових навантажень.

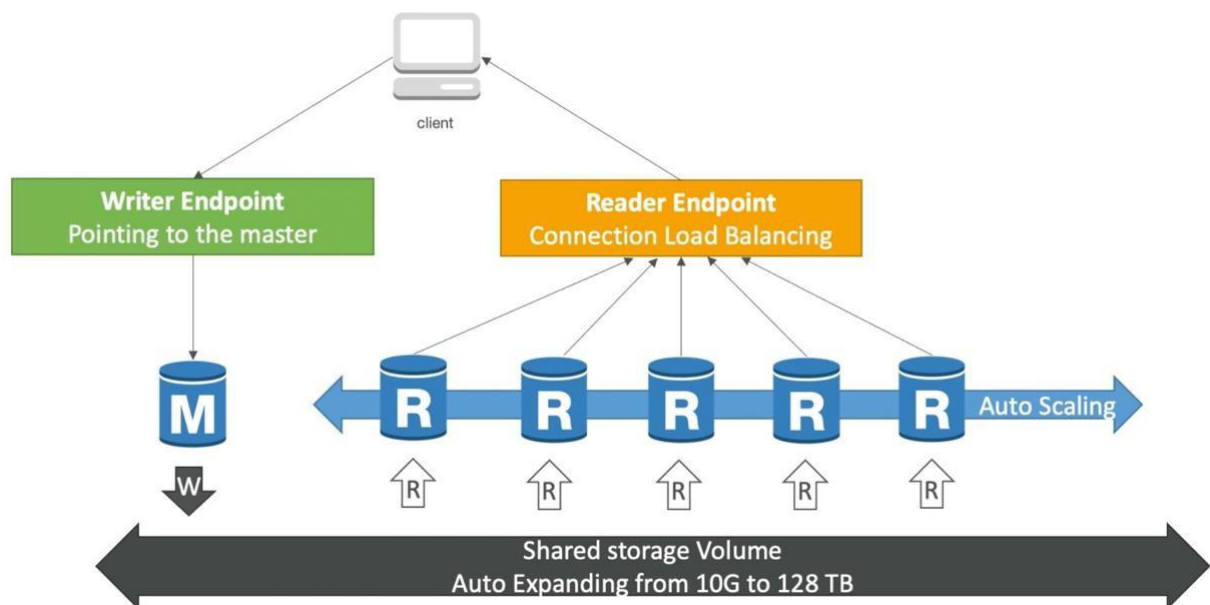


Рис. 2.2. Архітектура AWS Aurora DB Service

Також AWS Aurora автоматично резервує дані та має вбудовані механізми відновлення після збоїв, що робить її надзвичайно надійною. Ця функціональність ідеально підходить для організації пасажирських перевезень, де потрібна висока доступність даних і швидка обробка запитів.

У результаті, використання Aurora DB у даній системі дозволить забезпечити ефективне зберігання даних, високу продуктивність і надійність платформи, яка буде готова до масштабування разом із ростом кількості користувачів і даних.

2.2.3 Використання метеорологічних даних OpeanWeather

Під час розробки інноваційного методу організації спільних пасажирських

перевезень було приділено увагу не лише технічним аспектам і функціональним можливостям системи, але й використанню додаткових провайдерів хмарних технологій, які здатні суттєво підвищити ефективність і точність роботи. Одним із ключових факторів, що впливають на час виконання маршруту та його вартість, є погодні умови. У цьому контексті інтеграція з хмарними сервісами, які надають актуальні метеорологічні дані, виглядає як стратегічно важливе рішення.

Таке інтегрування дозволить системі не лише адаптувати маршрути до поточних умов, але й передбачати можливі затримки або складнощі на основі прогнозів погоди. Це, у свою чергу, підвищить рівень задоволеності пасажирів, зменшить ризики та оптимізує витрати на перевезення. Хмарні сервіси можуть надати доступ до детальних даних про температуру, опади, стан доріг та інші метеорологічні фактори, які критично важливі для планування безпечного та ефективного руху транспортних засобів.

Відповідно наступною задачею було дослідити доступні хмарні технології для отримання даних про метеорологічне становище задіяних ділянок маршруту та підсумувати зібрані дані.

Під час вибору сервісу для отримання метеорологічних даних я орієнтувався на кілька ключових критеріїв: доступність, точність даних, можливість масштабування та зручність інтеграції. OpenWeather API виявився ідеальним кандидатом завдяки наступним перевагам:

1. Широкий спектр метеорологічних даних: OpenWeather API надає доступ до поточних погодних умов, прогнозів на декілька днів вперед та історичних даних. Це дозволяє системі враховувати погодні зміни в реальному часі та планувати маршрути з урахуванням довгострокових прогнозів.
2. Глобальне покриття: Сервіс охоплює метеорологічні дані для всього світу, що важливо для масштабованої системи перевезень, яка може працювати у різних регіонах.
3. Висока точність прогнозів: OpenWeather API об'єднує дані з численних джерел, включаючи метеорологічні станції, супутники та математичні моделі, що забезпечує актуальність і точність даних.

4. Деталізована інформація: API надає не лише базові параметри, як-от температура та опади, а й специфічні дані:

- інтенсивність та тип опадів;
- швидкість і напрямок вітру;
- видимість і атмосферний тиск;
- індикатори дорожніх умов, наприклад, ризик ожеледиці;

5. Гнучкість інтеграції: OpenWeather API підтримує кілька мов програмування та надає зручну документацію, що полегшує його інтеграцію у систему.

6. Можливість налаштування: Сервіс дозволяє запитувати дані для конкретних географічних точок, а також групувати дані для кількох локацій одночасно, що особливо важливо для розрахунків маршрутів.

Для того щоб виконати запит треба значення широти і довготи, до прикладу показники мого місцеположення Київ, вул. Осокорська 2А є: широта: 50.45 , довгота: 30.52. Якщо я відправлю ці дані я отримаю у відповідь результат зображений на Рисунку 2.3.

```

{
  "lat": 50.4501,
  "lon": 30.5236,
  "timezone": "Europe/Kiev",
  "timezone_offset": 7200,
  "current": {
    "dt": 1684929490,
    "sunrise": 1684926645,
    "sunset": 1684977332,
    "temp": 292.55,
    "feels_like": 292.87,
    "pressure": 1014,
    "humidity": 89,
    "dew_point": 290.69,
    "uvi": 0.16,
    "clouds": 53,
    "visibility": 10000,
    "wind_speed": 3.13,
    "wind_deg": 93,
    "wind_gust": 6.71,
    "weather": [
      {
        "id": 803,
        "main": "Clouds",
        "description": "broken clouds",
        "icon": "04d"
      }
    ]
  }
}

```

Рис. 2.3. Приклад відповіді від OpenWeather

де:

lat: широта місцезнаходження, десятковий формат (від -90 до 90);

lon: довгота місцезнаходження, десятковий формат (від -180 до 180);

timezone: назва часового поясу для вказаного місцезнаходження (наприклад, "Europe/Kiev");

timezone_offset - зсув в секундах від UTC (7200 секунд = UTC+2);

current - об'єкт, що містить поточні метеорологічні дані;

current.dt - поточний час у вигляді Unix-мітки часу (UTC);

current.sunrise - час сходу сонця у вигляді Unix-мітки часу (UTC);

current.sunset - час заходу сонця у вигляді Unix-мітки часу (UTC);

current.temp - температура в Кельвінах (можна перевести в Цельсій або Фаренгейт);

current.feels_like - температура, яку сприймає людина (також в Кельвінах, можна перевести);

current.pressure - атмосферний тиск на рівні моря в гПа (гектопаскалях);

current.humidity - вологість в відсотках (%);

current.clouds - кількість хмарності в відсотках (%);

current.visibility - видимість в метрах (максимум 10 км);

current.wind_speed - швидкість вітру в метрах на секунду (можна перевести в милі на годину);

current.wind_deg - напрямок вітру в градусах (метеорологічний);

current.wind_gust - пориви вітру в метрах на секунду (можна перевести в милі на годину);

current.weather - умови погоди;

Таким чином, використання OpenWeather API не лише забезпечує систему актуальними метеорологічними даними, а й дозволяє враховувати вплив погодних умов на маршрути в режимі реального часу, оптимізуючи процес перевезень та підвищуючи задоволеність пасажирів. Приклад карти опадів спроектований OpenWeather зображений на Рисунку 2.5.

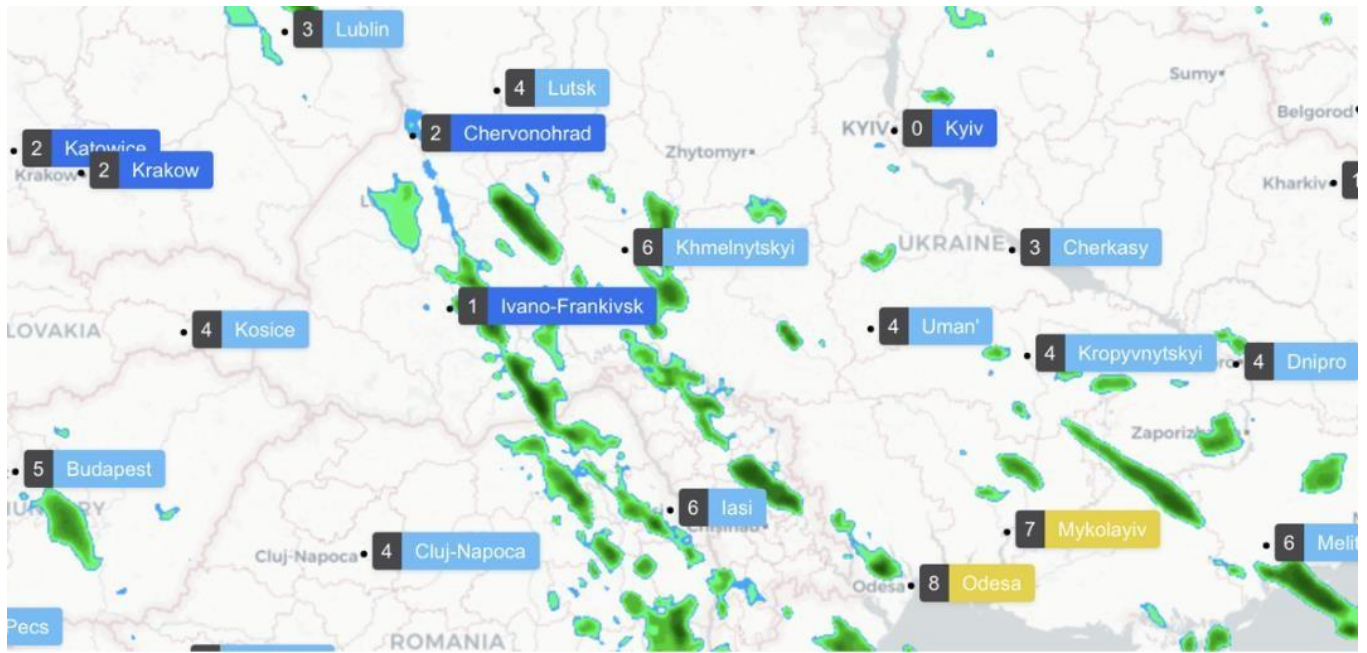


Рис. 2.4. Карта опадів OpenWeather

2.2.4 Побудова маршруту разом з Sygic

Після визначення погодних умов сучасні хмарні технології дозволяють враховувати різноманітні фактори, такі як дорожній рух у реальному часі покриття та завантаженість доріг. Задача провайдера – дати інформацію скільки часу займе поїздка від точки А до точки Б, є доволі багато ресурсів з такими даними але давайте роздивимось найпопулярніші серед них Google Maps, Sygic та TomTom.

Давайте побудуємо порівняльну таблицю

Порівняння постачальників інформації щодо маршруту

Назва	Sygis	Google Maps	TomTom
Покриття доріг	Всі основні маршрути, збереження карт для офлайн	Всі основні маршрути, оновлення в реальному часі	Всі основні маршрути, збереження карт для офлайн
Підтримка електричних автомобілів (EV)	Підтримка електричних автомобілів (EV)	Є маршрути для електричних авто	Є підтримка для електричних автомобілів
Офлайн режим	Підтримує повноцінну офлайн-навігацію	Немає (не можна зберігати карти офлайн)	Підтримує офлайн карти та навігацію
Метеорологічні умови	Вбудовані дані про погоду	Немає	Вбудовані дані про погоду

Недоліком Sygis є не врахування метеорологічних умов, але оскільки ми самі плануємо брати погоду від провайдера і мати контроль над розрахунками - цей недолік не вважається валідним для нашого методу. В одночас переваги продемонстровані в таблиці очевидні, тому приділивши багато уваги дослідженню наявних хмарних рішень, серед багатьох конкурентів був виділений Sygis.

1. Точність і якість карт.

- Використовує карти HERE, які оновлюються в реальному часі та забезпечують високу точність даних.
- Офлайн-доступ до карт, що особливо важливо у зонах зі слабким покриттям мобільного інтернету.

2. Функціональність для бізнесу.

- Підтримка маршрутизації для різних типів транспорту (легкові авто, вантажівки, автобуси).
- Можливість враховувати особливі параметри транспорту, як-от габарити, тоннаж, небезпечні вантажі.

3. Оптимізація маршрутів.

- Враховує трафік у реальному часі, допомагаючи уникнути заторів і скоротити час у дорозі.
- Пропонує кілька альтернативних маршрутів із зазначенням часу та витрат пального.

4. Інтеграція через API.

- Простий доступ до API для інтеграції в сторонні додатки або корпоративні системи.
- Надає розширені можливості для кастомізації: врахування платних доріг, зон із обмеженнями руху та інші параметри.

5. Робота в реальному часі.

- Дані про дорожній трафік, перекриття, ремонтні роботи та аварії доступні в реальному часі.
- Відображення точного прогнозу часу прибуття з урахуванням поточної ситуації на дорозі.

6. Безпека.

- Забезпечує безпечні маршрути для великих транспортних засобів, уникаючи небезпечних ділянок дороги, низьких мостів чи вузьких вулиць.
- Підтримка рекомендацій водіям щодо дотримання швидкості та інших правил дорожнього руху.

На Рисунку 2.7 зображено приклад заданого маршруту і його відображення

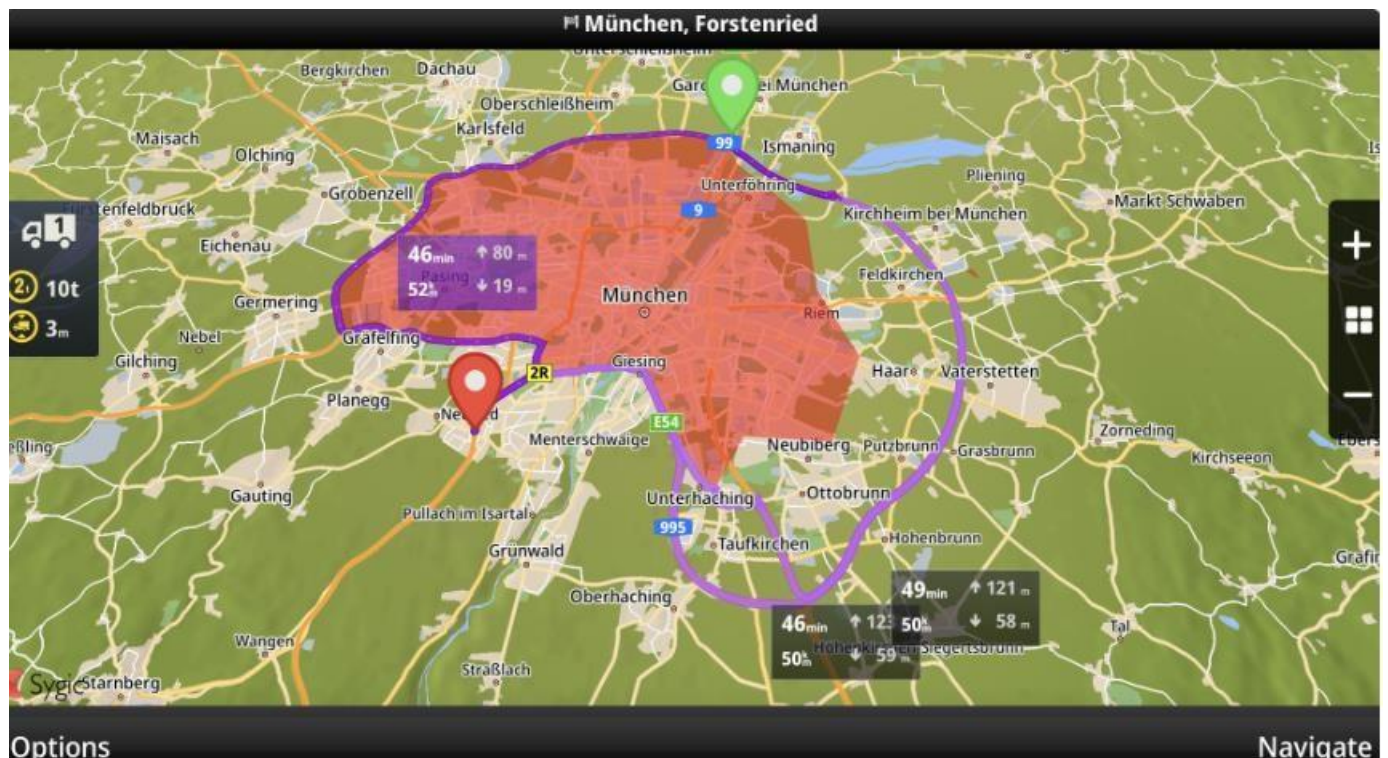


Рис. 2.5. Приклад прокладеного маршруту на карті

2.3 Дослідження та визначення формул розрахунку

2.3.1 Вага ребра графа

Для побудови маршруту було майже безальтернативно обрано використання графів, тому постає важливе завдання визначити метод побудови графа та спосіб обчислення ваги кожного ребра. Цей етап має об'єднати всі попередньо зібрані дані для точного аналізу складності маршруту між усіма вершинами графа. Вага ребра повинна відображати не лише відстань, а й інші фактори, які впливають на зручність і ефективність пересування.

Для визначення ваги кожного ребра необхідно врахувати наступні чинники:

- **Дистанція:** відстань між двома вершинами, що може бути отримана з картографічного провайдера.
- **Час у дорозі:** орієнтовна тривалість подорожі, яка враховує дорожні умови та швидкість руху.
- **Витрати:** передбачувані витрати на паливо чи енергію для подолання маршруту.
- **Кількість пасажирів:** впливає на завантаження транспортного засобу та загальний комфорт.
- **Наявність існуючого маршруту:** визначає, чи є вже прокладений шлях між точками, що може зменшити вагу ребра.
- **Завантаженість дороги:** інформація про поточний трафік, яка допомагає оцінити затори та затримки.
- **Погодні умови:** дані про погодні явища, які можуть впливати на безпеку та тривалість подорожі (дощ, сніг, вітер тощо).

Для побудови графа використовуються різні джерела, які забезпечують відповідні показники:

1. **Дистанція, час у дорозі, завантаження дороги** — ці дані можуть бути отримані від **Sygis**.
2. **Погодні умови** — дані надаються сервісом **OpenWeather**, який забезпечує актуальну інформацію про погоду в різних регіонах.
3. **Особисті дані** — включають витрати, кількість пасажирів, наявність існуючого маршруту. Ці показники є індивідуальними і не повинні отримуватись від зовнішніх провайдерів, а визначаються користувачем або системою управління маршрутом.

На основі цих даних можна розробити формулу, що враховуватиме вагомість кожного з параметрів

$$W = \sum_k C_k * i_k, \quad (1.1)$$

де W – вага ребра графа;

C_k – враховані значення(дистанція, час, витрата, кількість пасажирів, наявність вже існуючого маршруту, завантаження дороги та погодні умови);

i_k – індекси важливості кожного врахованого значення(може змінюватимь в залежнлсть від пріорітету відданого певним значенням).

2.3.2 Пошук найкоротшого шляху

Результуючим технічним завданням є пошук найкоротшого шляху у створеному графі. Це необхідно для визначення оптимального маршруту, який дозволить здійснити поїздку найвигіднішим способом. У цій сфері вже існує багато перевірених та ефективних алгоритмів, які використовуються для вирішення таких задач, тому вигадувати щось кардинально нове не має сенсу. Втім, є значний простір для вдосконалення самого процесу застосування цих алгоритмів, що в майбутньому може підвищити їх ефективність або адаптивність до конкретних умов.

На даному етапі наша увага зосереджена на аналізі найпоширеніших методів, які активно використовуються у подібних задачах, щоб визначити, який з них найкраще підходить для реалізації в цьому проєкті. У подальшому ми також розглянемо можливі способи оптимізації цих алгоритмів або їх інтеграції в існуючу систему.

Таблиця 2.2

Порівнянні алгоритмів пошуку найкоротших маршрутів

Назва	Дейкстри	A^*	Флойд-Уоршел
Призначення	Пошук найкоротшого шляху від однієї вершини до інших	Пошук найкоротшого шляху до конкретної цілі	Пошук найкоротших шляхів між всіма парами вершин
Часова складність	$O(n^2)$	$O((E + V) \log V)$	$O(n^3)$

Аналізуючи дані алгоритми для організації спільних пасажирських перевезень, я дійшов висновку, що унікальність підходу може полягати в наступному: оскільки мова йде про спільні перевезення, маршрути можуть динамічно змінюватися залежно від попиту. Постійний перерахунок складності можливих шляхів кожного разу створює значне навантаження на систему. Однак, якщо врахувати, що всі чинники, які впливають на розрахунки, є відносно стабільними, можна прийти до ідеї, що зберігання наперед розрахованих шляхів є цілком виправданим рішенням.

Наприклад, якщо користувач запитує маршрут із Києва до Білої Церкви, а в нашій системі вже побудовано граф для всієї Київської області, немає потреби кожного разу виконувати нові розрахунки. Ми просто використовуємо заздалегідь підготовлені дані, що дозволяє суттєво прискорити обробку запиту та знизити навантаження на систему. Такий підхід забезпечує ефективність та швидкість роботи, що є критично важливим у контексті динамічних пасажирських перевезень.

Тому кінцеве рішення це взяти всі переваги від формули Флойда-Уоршела та перекрити недоліки тим що підрахунки будуть вестись в незалежності від запитів на побудов. Запит від користувача буде використовувати вже розраховані значення.

$$d[i][j] = \min(d[i][j], d[i][k] + d[k][j]), \quad (1.2)$$

де $d[i][j]$ — мінімальна відстань між вершинами i та j ;

$d[i][k] + d[k][j]$ — шлях між вершинами i та j через проміжну вершину k .

Давайте побуємо умовний граф з вигаданими значеннями задля демонстрації обраної формули. На рисунку 2.4. зображений граф з 4 вершинами.

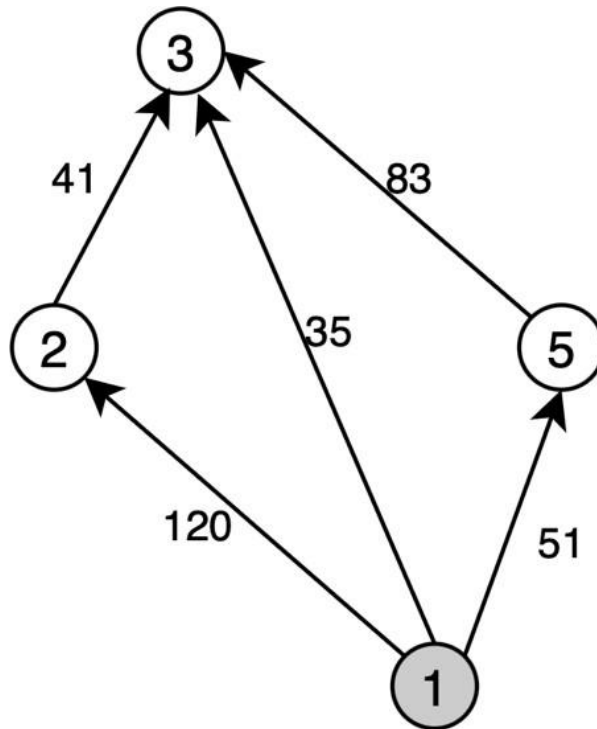


Рис. 2.6. Приклад графа з 4 вершинами

Результуюча матриця C4 зображена на Рисунку 2.7.

C4 =

	V1	V2	V3	V4
V1	0	120	35	51
V2	∞	0	41	124
V3	∞	∞	0	83
V4	∞	∞	∞	0

Рис. 2.7. Розрахована матриця з заданими вершинами і виділеним найкоротшим шляхом між V1 – V3

З якої виходить що найкоротший шлях – найменша вага графа, від V1 до V3 це значення 35.

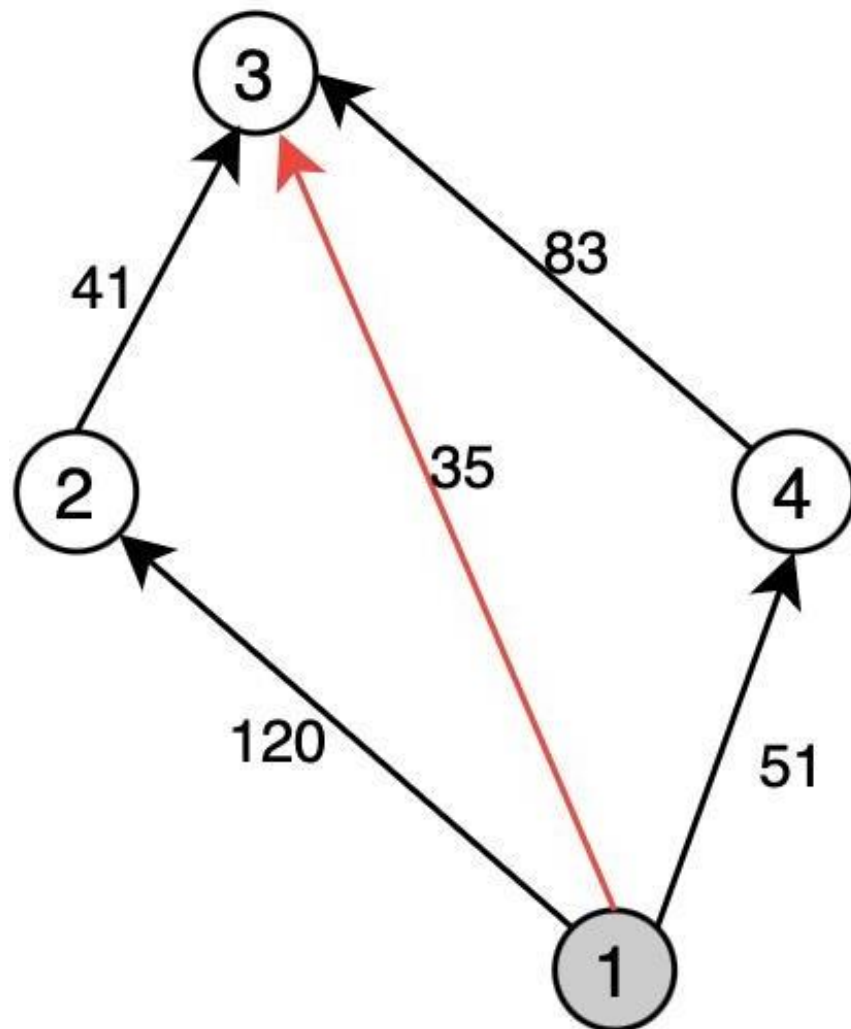


Рис. 2.8. Граф з 4 вершинами і найкоротшим шляхом між V1 – V3

2.4 Побудування алгоритмів

2.4.1 Алгоритм спільних пасажирських перевезень

Зважаючи на визначені технології та алгоритми, а також враховуючи переваги та недоліки кожного варіанту, можна розробити алгоритм організації спільних пасажирських перевезень, представлений на Рисунку 2.8.

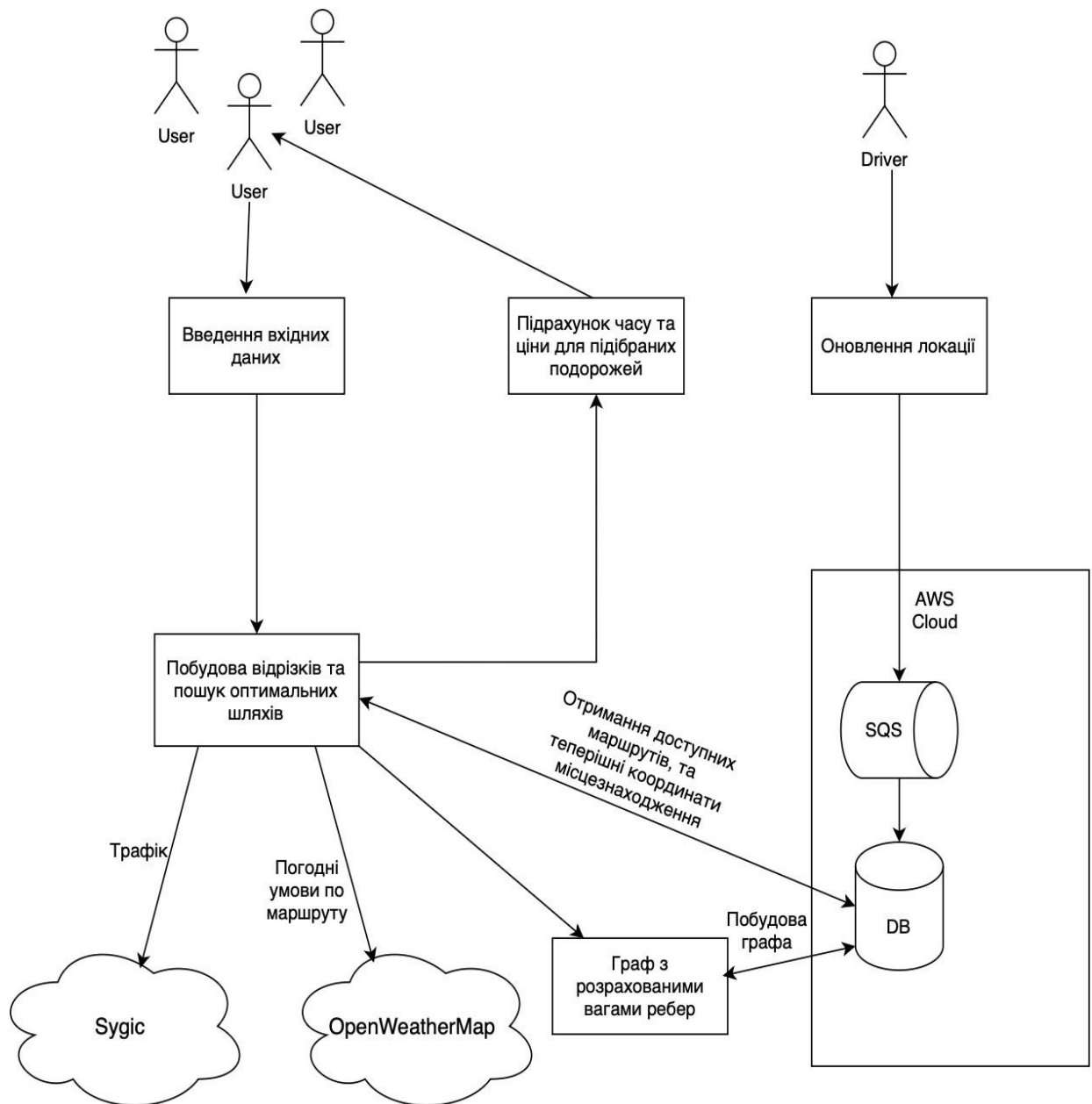


Рис. 2.6. Алгоритм спільних пасажирських перевезень

Де ключовими рушіями методу є

- Введення вхідних даних від користувача(точка старту, кінцева точка)
- Збір даних про стан доріг
- Збір даних про погодні умови
- Побудова відрізків та пошук оптимальних шляхів на основі попередньо розрахованих ребер, які в свою чергу не підраховуються кожен раз

- Підрахунок часу та ціни для підібраних подорожей
- Асинхронне оновлення локації водія і збереження цих даних в базу
- Підрахунок вагів ребер графа кожні n днів/годин

2.4.2 Алгоритм пошуку маршруту

Мета алгоритму пошуку маршруту полягає у визначенні оптимального набору шляхів між заданими точками на транспортній мережі. Це включає обробку великої кількості вхідних даних, таких як початкова і кінцева точки, обмеження щодо часу, витрат або відстані, а також інших параметрів, що впливають на вибір маршруту. Основним результатом роботи алгоритму є список можливих маршрутів, упорядкованих за заданими критеріями, які забезпечують зручність та ефективність подорожі для користувачів. Цей результат використовується для формування пропозицій, які система відправляє користувачу, допомагаючи йому швидко обрати найбільш прийнятний варіант. Алгоритм пошуку маршруту зображений на Рисунку 2.9

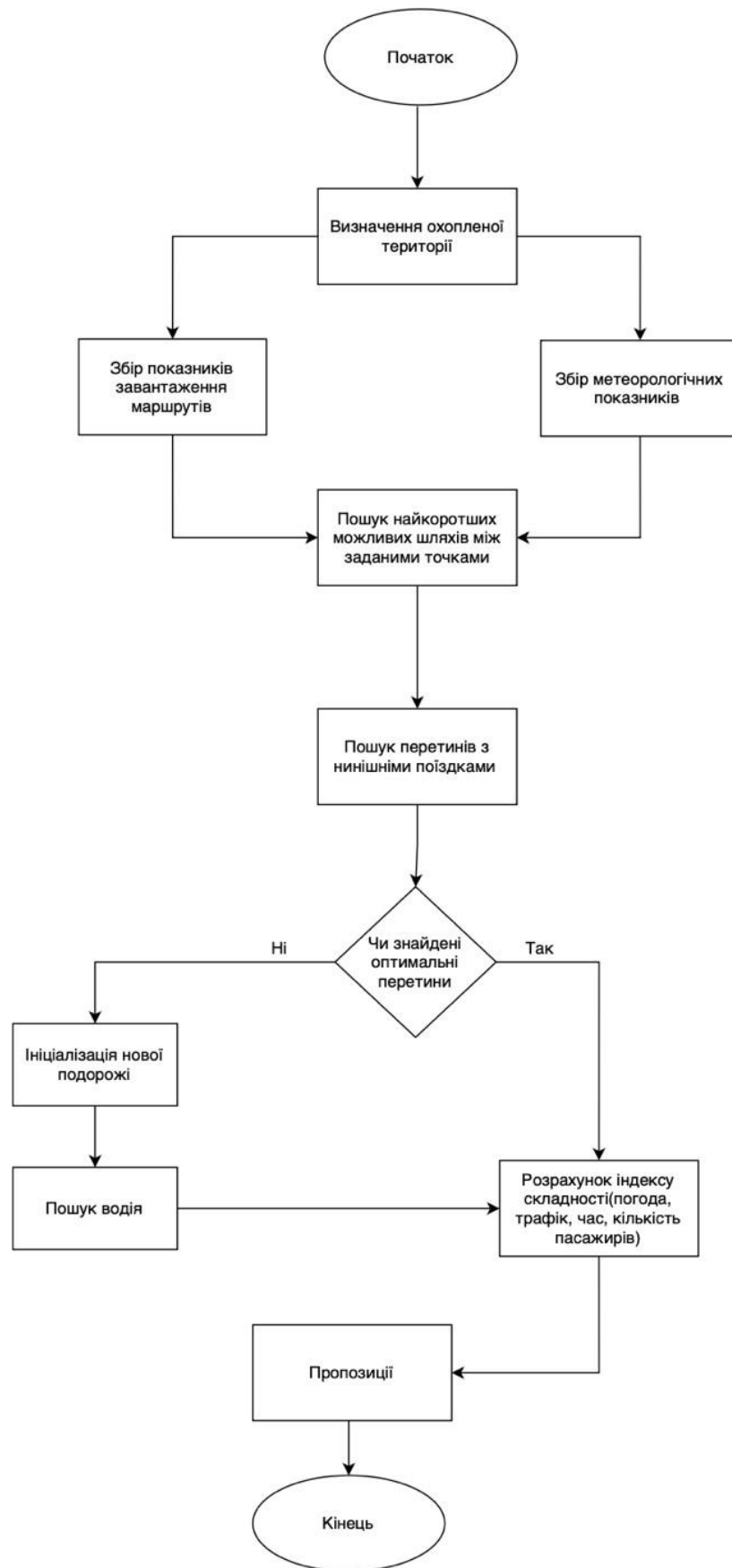


Рис. 2.7. Алгоритм пошуку маршруту

Алгоритм розпочинається з введення даних і завершується пропозицією потенційних маршрутів. У процесі може виникнути ситуація, коли оптимальні маршрути не знайдено. У такому разі необхідно створити новий маршрут, який, очевидно, буде дорожчим для кінцевого користувача через відсутність інших пасажирів на цей напрямок.

Додатково, після визначення охопленої території, слід зібрати дані про завантаженість маршруту та метеорологічні умови. Ці дані повинні збиратися паралельно, оскільки немає потреби виконувати ці дії послідовно. Однак до етапу пошуку найкоротших шляхів обидва ці показники мають бути готові.

Загалом, алгоритм демонструє високу ефективність і враховує всі можливі сценарії, що робить його ефективним для вирішення завдань організації спільних перевезень.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ МЕТОДУ ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ

3.1 Вимоги до вхідних та вихідних даних додатку організації спільних пасажирських перевезень

Процес управління ризиками у сфері спільних пасажирських перевезень, враховуючи його трудомісткість і багатофакторність, доцільно автоматизувати. Нині на українському ринку є декілька систем, спрямованих на забезпечення управління перевезеннями. Проте основними обмеженнями цих систем у контексті управління ризиками є:

- орієнтація переважно на логістичну оптимізацію, а не на управління ризиками;
- необхідність постійного підключення до Інтернету для управління процесами перевезень;
- недостатня інтеграція методик оцінки ризиків, що дає неузгоджені результати для прийняття рішень.

У зв'язку з цим було прийнято рішення про розробку власної програмної системи. Система включає багатофункціональний додаток, що реалізує основні алгоритми оцінки та управління ризиками в перевезеннях. Вона поєднує методичні та програмні засоби для вирішення наступних завдань:

- формування бази знань на основі даних перевізників, пасажирів і зовнішніх факторів (шляхом опитувань та збору статистичних даних);
- ідентифікація та оцінка ризиків у процесі організації перевезень за алгоритмами нечіткої логіки;
- підтримка прийняття рішень щодо оптимізації маршрутів, розподілу транспортних засобів та реагування на ризики (затори, затримки, поломки тощо);
- формування звітів про виявлені ризики, пропоновані заходи та їх реалізацію.

Вхідними даними програмної системи є:

- передумови ризиків (розклад перевезень, маршрутна мережа, погодні умови

тощо);

- експертні правила та рекомендації перевізників;
- технічні характеристики транспортних засобів;
- історичні дані про виконані перевезення.

Ця система дозволить підвищити ефективність управління ризиками та забезпечити безперебійне функціонування перевезень навіть у складних умовах.

3.2 Програмна реалізація та засоби реалізації

Для того щоб переконатися, що розроблена методика є дієвою та здатною суттєво покращити процес організації спільних пасажирських перевезень, було прийнято рішення створити програмний продукт, який отримав назву SmartRoute. Цей продукт є не просто технічним засобом, а втіленням інноваційного підходу до управління перевезеннями, який поєднує сучасні технології та глибоке розуміння специфіки галузі.

На етапі проєктування постало питання: яким саме має бути цей додаток? Розглядалися різні варіанти, серед яких були мобільні додатки для платформ iOS та Android, десктопні програми, що орієнтовані на стаціонарні пристрої, а також рішення для інших екосистем. Кожен із цих підходів має свої переваги й недоліки, тому вибір був нелегким.

Мобільні додатки, наприклад, могли б забезпечити доступність для користувачів смартфонів, які складають значну частину аудиторії. Десктопні додатки, у свою чергу, пропонували б розширені функціональні можливості та зручність для корпоративного сегмента. Проте всі ці варіанти мають один спільний недолік — залежність від конкретної платформи або типу пристрою.

З огляду на це, вибір було зроблено на користь створення веб-додатку. Цей формат дозволяє забезпечити максимально широкий доступ для користувачів з будь-яким пристроєм, що має браузер і підключення до Інтернету. Незалежно від того, чи це смартфон, планшет, ноутбук чи стаціонарний комп'ютер, веб-додаток гарантує, що користувач зможе скористатися усіма його функціями без потреби встановлювати

додаткове програмне забезпечення.

Крім того, веб-додаток має ще одну важливу перевагу — він є універсальним і не потребує адаптації під різні операційні системи. Це рішення значно скорочує витрати на розробку та підтримку, дозволяючи зосередитися на вдосконаленні функціональних можливостей, інтеграції з іншими системами та забезпеченні найвищого рівня зручності для користувачів.

Таким чином, обравши веб-додаток як основу для реалізації методики, ми забезпечуємо його доступність, гнучкість і зручність для всіх учасників процесу, незалежно від їхніх технічних умов чи географічного розташування. SmartRoute стане інструментом, який відкриє нові горизонти в організації пасажирських перевезень, зробить процеси більш ефективними та доступними для всіх.

Після того як було визначено тип додатка, наступним важливим етапом стала розробка його архітектури. Перед командою постало питання: обрати мікросервісну чи монолітну архітектуру? Обидва підходи мають свої переваги й недоліки, тому рішення потребувало ретельного аналізу.

Мікросервісна архітектура є сучасним підходом, який пропонує масштабованість, гнучкість у розробці та оновленні окремих компонентів. Однак її реалізація вимагає значних ресурсів: як фінансових, так і часових. Підтримка такої архітектури часто є складнішою, оскільки потребує належного управління взаємодією між сервісами, що може стати викликом для невеликого проєкту. Враховуючи, що концепція додатка поки не має активної бази користувачів, було вирішено обрати більш простий і швидкий у розробці підхід — монолітну архітектуру.

Монолітна архітектура дозволяє розробити продукт швидше й ефективніше, адже всі функції об'єднані в одному додатку. Це оптимальний вибір для прототипів і концептів, коли головним пріоритетом є швидкий запуск і тестування ідей. Такий підхід також спрощує управління та зменшує потребу в масштабній інфраструктурі.

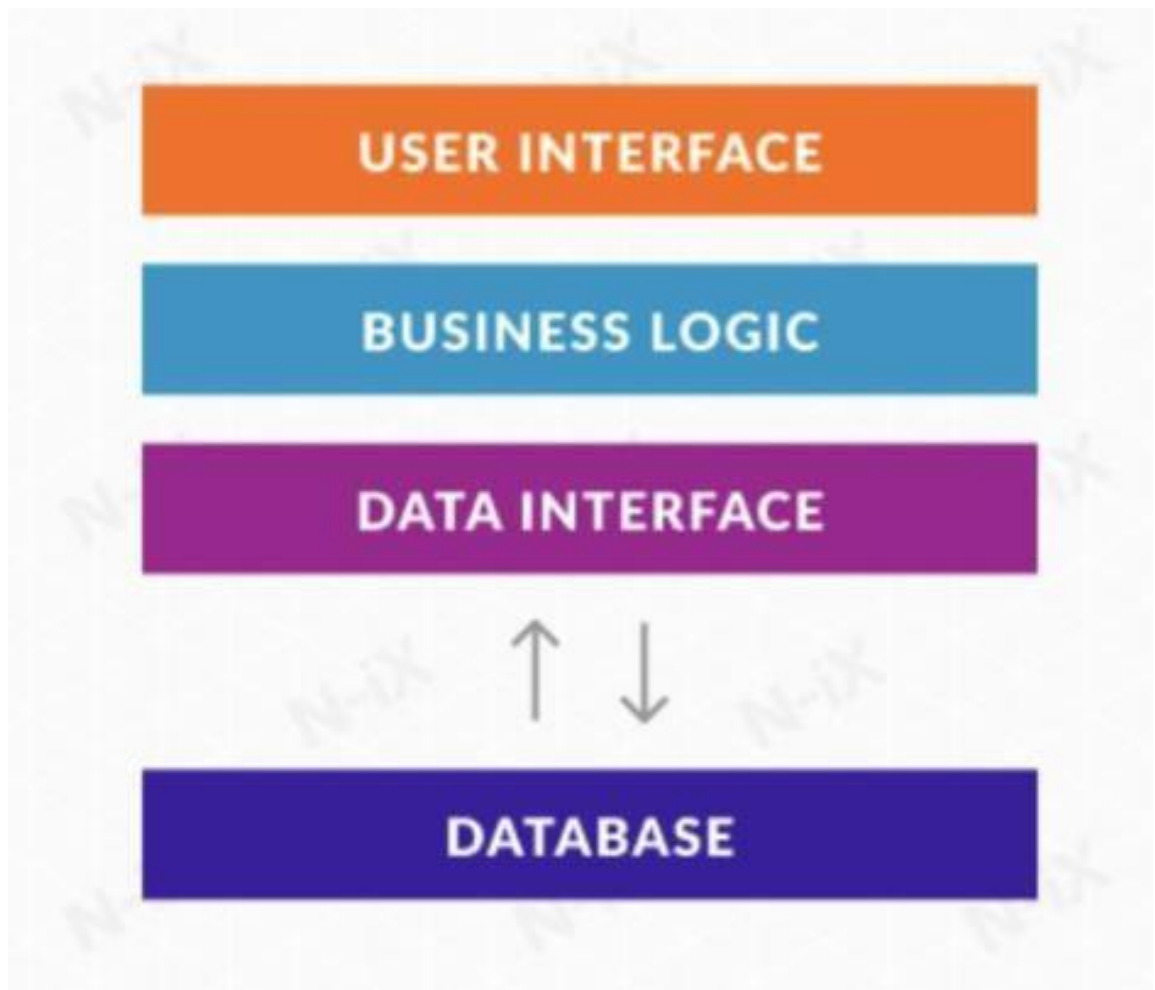


Рис. 3.1. Монолітна архітектура веб-додатка

Далі було визначено технології, які будуть використані для реалізації фронтенду та бекенду додатка. Для створення інтерфейсу користувача було обрано React. Ця бібліотека є одним із найпопулярніших інструментів для фронтенд-розробки, що має величезне співтовариство розробників і широкий спектр готових рішень. Крім того, React має відносно низький поріг входу, що дозволяє швидко залучати нових спеціалістів до команди. Завдяки своїй гнучкості й можливості створювати високопродуктивні користувацькі інтерфейси, React став ідеальним вибором для фронтенд-частини додатка.

Для бекенду було обрано мову програмування Java. Це один із найбільш перевірених часом і надійних інструментів у розробці корпоративних систем. Java має потужне співтовариство, яке пропонує численні бібліотеки та фреймворки для вирішення майже будь-яких завдань. Також значна частина підприємств, близько

80%, використовує Java у своїх проєктах, що робить її стабільним вибором для розробки. Висока надійність і передбачуваність Java дозволяють уникнути неприємних сюрпризів під час розробки та експлуатації системи.

Таким чином, архітектурне рішення проєкту базується на прагматичному підході: монолітна структура для швидкої реалізації, React для зручності й ефективності фронтенду, а Java для стабільного й надійного бекенду. Це поєднання забезпечує оптимальну базу для успішного запуску додатка SmartRoute



Рис. 3.2. Діаграма в взаємодії користувача з розробленою системою

Були створені два репозиторія, один під фронтент, другий під бекенд, намаплені сутності в базу даних, і впроваджена комунікація між всіма компонентами. Екрані форми роботи додатка представлені на наступних рисунках.

Деталі подорожі

Маршрут: Тернопіль - Львів

Відстань: 126.6 км

Бажаний дата відправки: -

Бажана дата прибуття: 26.11.25 10:00

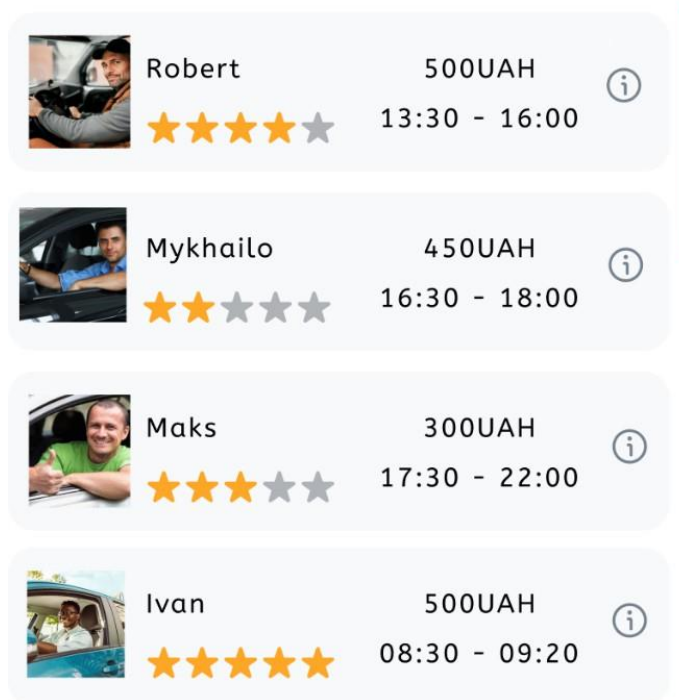


Рис. 3.3. Інтерфейс запропонованих маршрутів

На Рисунку 3.3 представлено результуючу форму пошуку для напрямку Тернопіль – Львів. У цій формі наочно відображені ключові параметри, які допомагають користувачам швидко і зручно знайти потрібну інформацію. Зокрема, зазначено відстань між містами, список доступних водіїв, дати відправлення та прибуття, а також додаткові деталі, такі як ціна та рейтинг кожного водія. Усі ці елементи оформлені у зрозумілому й зручному для користувача форматі, що сприяє простоті й ефективності взаємодії

Деталі подорожі

Маршрут: Тернопіль - Львів

Відстань: 126.6 км

Бажаний дата відправки: -

Бажана дата прибуття: 26.11.25 10:00

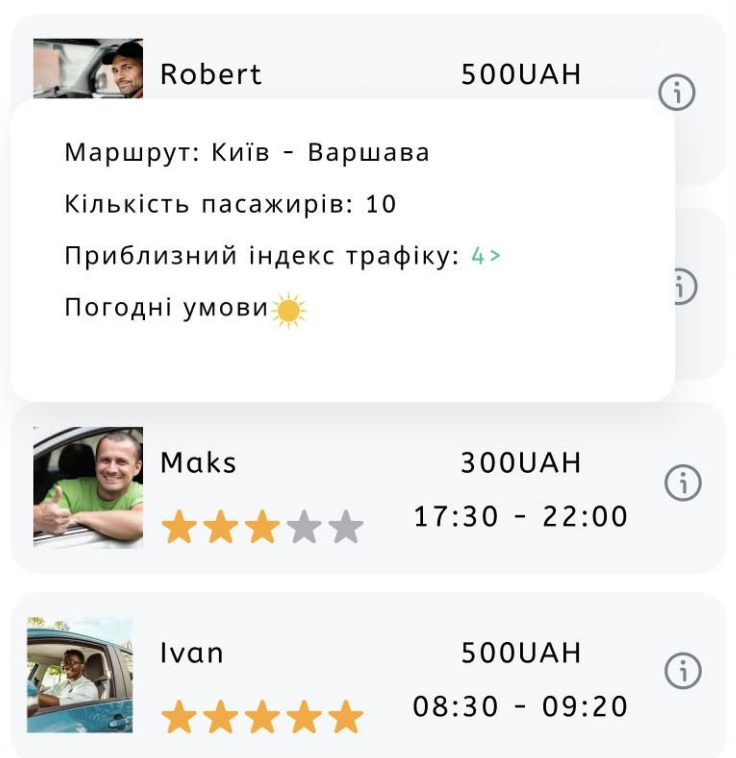


Рис. 3.4. Інтерфейс деталей поїздки

На Рисунку 3.4 зображено так званий Tooltip, який відображає додаткову інформацію на кожную запропоновану подорож. У цьому інструменті містяться такі корисні дані, як кількість доступних місць для пасажирів, приблизний індекс трафіку на маршруті, а також актуальні погодні умови. Усі ці показники подаються у зрозумілій формі, що дозволяє користувачам швидко отримати необхідну інформацію без зайвих пошуків, підвищуючи ефективність та комфорт використання додатка.

Деталі подорожі

Маршрут: Тернопіль - Львів

Відстань: 126.6 км

Бажаний дата відправки: -

Бажана дата прибуття: 26.11.25 10:00



Robert

500UAH

13:30 - 16:00

★★★★☆





Mykhailo

450UAH

16:30 - 18:00

★★★☆☆





Maks

300UAH

17:30 - 22:00

★★★☆☆





Ivan

500UAH

08:30 - 09:20

★★★★★



Вибрати

Рис. 3.5. Інтерфейс вибору водія

На Рисунку 3.5 показано стан інтерфейсу, коли користувач обрав конкретну поїздку. У цьому режимі доступна функція запису в маршрут, що дозволяє забронювати місце для подорожі

Таким чином, створений додаток для організації спільних пасажирських перевезень успішно реалізує всі основні функції та готовий до тестування в реальних умовах. Його архітектура, зручний інтерфейс та продумані можливості забезпечують ефективність і зручність використання, відкриваючи шлях до перевірки запропонованої методики на практиці.

3.3 Дослідження ефективності запропонованої методики

У сучасному світі, де мобільність і швидкість реагування стають ключовими факторами успіху, питання оптимізації пасажирських перевезень набуває особливого значення. Розробка нових підходів та інструментів для організації спільних поїздок є не лише актуальною, але й необхідною умовою для підвищення зручності, економічності та екологічної сталості транспортних рішень. Запропонована методика організації спільних пасажирських перевезень спрямована на вирішення цих завдань, поєднуючи сучасні технології та інноваційні підходи.

Однак будь-яка методика, навіть із найкращими теоретичними передумовами, потребує практичної перевірки, щоб підтвердити її ефективність і доцільність впровадження. Саме тому дослідження ефективності запропонованої методики стає важливим кроком на шляху.

Це дослідження має на меті оцінити, наскільки розроблений підхід відповідає вимогам сучасного ринку пасажирських перевезень, враховує потреби користувачів та вирішує актуальні проблеми, такі як перевантаженість маршрутів, недоступність інформації про поїздки та низька ефективність планування. Отримані результати дозволять не лише підтвердити доцільність використання методики, але й виявити можливості для її вдосконалення, адаптації до змінних умов і розширення функціоналу.

Таким чином, дослідження ефективності запропонованої методики є невід’ємною складовою її впровадження та сприяє досягненню головної мети — створення зручного, доступного та надійного інструменту для організації спільних пасажирських перевезень.

Почнемо з функціональних можливостей а до порівняння візьмем вищезазначені Uber та BlaBlaCar

Таблиця 3.1

Порівняння функціональних можливостей розробленого додатку

Розглянуті методики та їх можливості			
Урахування складності пересування при розрахунку ціни	+	-	+
Вибір водія	-	+	+
Урахування погодних умов	-	-	+
Прогнозування трафіку	-	-	+
Підтримка довготривалих маршрутів	-	+	+
Підтримка короткотривалих маршрутів	+	-	+

Як можна побачити, навіть за функціональними можливостями, наш метод має явні переваги перед конкурентами. Багато з них зосереджуються або на довгострокових, або на короткострокових маршрутах, тоді як наша методика готова обробляти і ті, і інші. Крім того, конкуренти не враховують аналіз трафіку, що є важливим елементом для оптимізації маршруту, в той час як ми активно використовуємо ці дані та включаємо їх у процес аналізу.

Один із значних переваг нашого підходу — це можливість вибору водія. Для користувачів дуже важливо мати змогу обирати водія самостійно, що не завжди доступно в таких сервісах, як Uber. Це дає змогу покращити досвід користувача, надаючи йому більше контролю над поїздкою та підвищуючи рівень задоволення від сервісу.

Після того, як ми розглянули функціональні можливості нашої методики організації спільних пасажирських перевезень, важливо зосередитись на кількісних показниках, які дозволять точно оцінити її ефективність у реальному використанні. Оскільки одна з головних цілей розробки нової методики — це оптимізація перевезень, важливо зібрати відповідні цифри та дані, які відображатимуть, наскільки успішно вона працює в умовах реальних поїздок.

Одним із перших аспектів, на який слід звернути увагу, є **час на обробку запиту**. Це важливий показник, який визначає, наскільки швидко система може обробити запит користувача та надати відповідь у вигляді доступних маршрутів, водіїв, та інших даних. Усі учасники повинні мати можливість швидко отримати необхідну інформацію, оскільки це безпосередньо впливає на досвід користувача та ефективність сервісу.

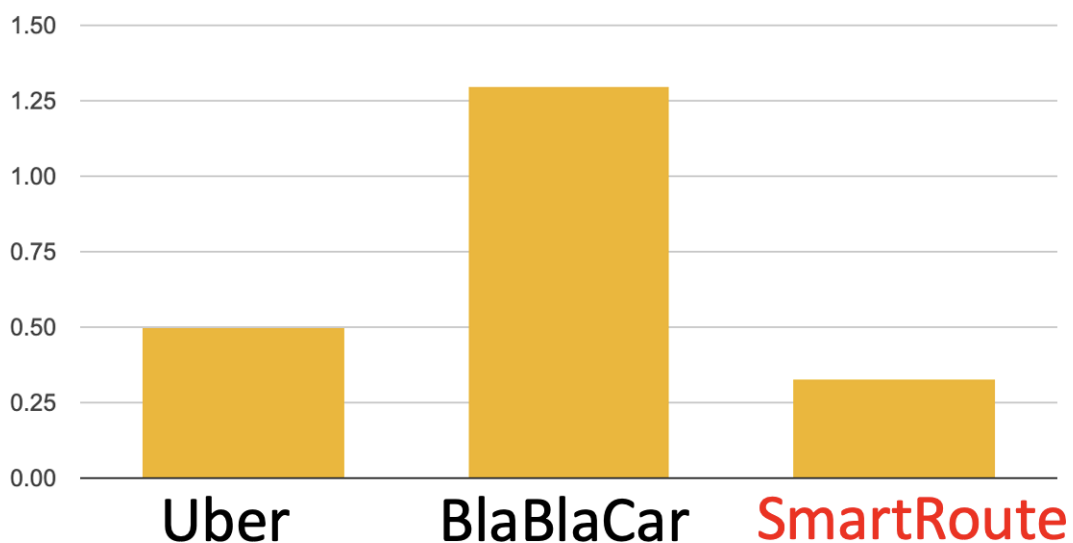


Рис. 3.6. Порівнянні середнього часу відповіді сервера

Рисунок 3.6 демонструє різницю у часі відповіді на 100 випадкових запитів з інтерфейсу трьох сервісів: Uber, BlaBlaCar та SmartRoute. Середній час відповіді для Uber складає 49 секунд, для BlaBlaCar — 1.3 секунди, а для SmartRoute — 0.29 секунди. Ці дані яскраво показують, що SmartRoute значно перевершує конкурентів у швидкості обробки запитів.

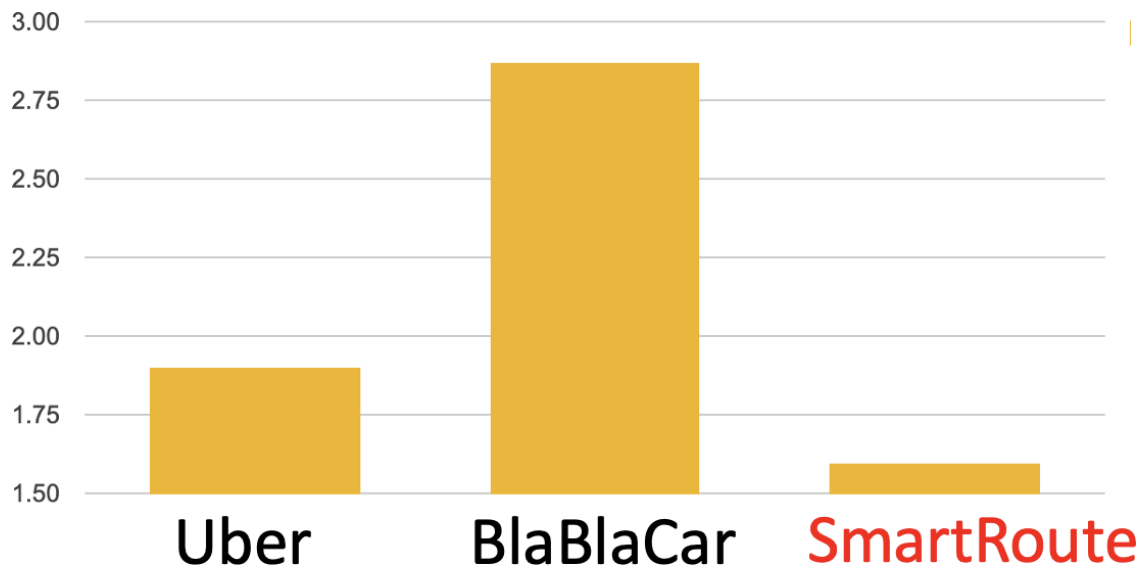


Рис. 3.7. Порівнянні середнього часу побудови маршрута

Рисунок 3.7 демонструє різницю у часі відповіді сервісів на запит побудови маршруту для відстані від 1 до 500 км. У результатах видно, що середній час відповіді для Uber становить 1.93 секунди, для BlaBlaCar — 2.8 секунди, а для SmartRoute — 1.6 секунди. Це свідчить про те, що SmartRoute має конкурентну перевагу щодо швидкості побудови маршруту порівняно з іншими сервісами, що також позитивно впливає на ефективність користування системою.

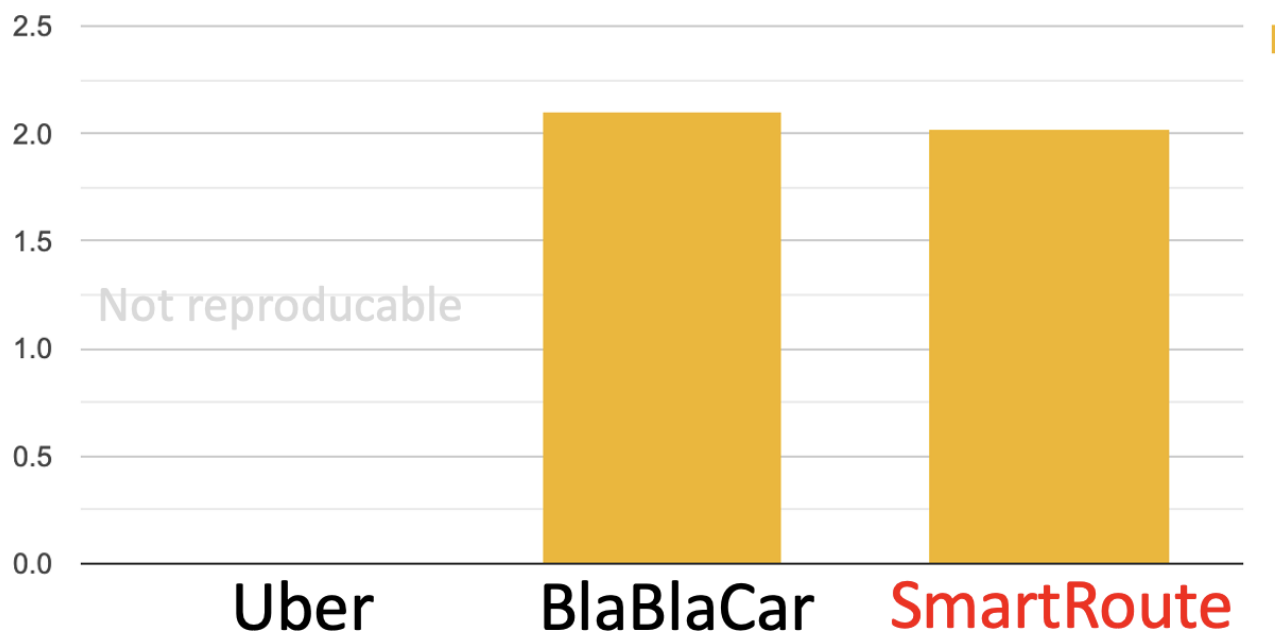


Рис. 3.8. Порівнянні середнього знаходження перетину маршрутів

Рисунок 3.8 демонструє швидкість знаходження перетину маршрутів та підбору водіїв для 100 запитів на відстані від 1 до 500 км. Для сервісу BlaBlaCar час становить 2.15 секунди, для SmartRoute — 2.01 секунди. Оскільки Uber не має можливості вибору водія, цей показник не вдалося відслідкувати. Різниця в часі між SmartRoute і BlaBlaCar свідчить про перевагу нашої системи у швидкості підбору відповідних маршрутів.

Розроблена методика організації спільних пасажирських перевезень має значно більше функціональних можливостей порівняно з існуючими конкурентами. Ключовими перевагами є можливість обробки як короткострокових, так і довгострокових маршрутів, аналіз трафіку для оптимізації поїздок, а також важлива функція вибору водія, яка не реалізована в таких сервісах, як Uber.

Що стосується показників ефективності, то результати тестування підтверджують високі досягнення в продуктивності: SmartRoute показує значно швидший час відповіді на запити порівняно з Uber та BlaBlaCar. Наприклад, середній час відповіді на 100 рандомних запитів становить 0.29 секунди для SmartRoute, тоді як для Uber цей показник складає 0.49 секунд, а для BlaBlaCar — 1.3 секунди.

Також у процесі побудови маршруту на відстані від 1 до 500 км SmartRoute демонструє кращу швидкість (1.6 секунди) порівняно з BlaBlaCar (2.8 секунди). Аналіз перетину маршрутів і підбору водіїв також показав, що SmartRoute швидше за BlaBlaCar (2.01 секунди проти 2.15 секунди).

Отже, розроблена методика не тільки забезпечує кращу швидкість та ефективність, але й пропонує важливі функціональні можливості, які роблять сервіс більш зручним і привабливим для користувачів.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи проведено аналіз особливостей організації спільних пасажирських перевезень, методів та алгоритмів, що застосовуються для організації спільних пасажирських перевезень. Завдяки цьому ми змогли визначити основні підходи до організації спільних пасажирських перевезень, а також сформулювати вимоги до методу.

Розроблена система організації спільних пасажирських перевезень включає в себе можливість користувачеві зробити запит на маршрут і отримати список потенційних водіїв з подальшим вибором будь-якого з запропонованого варіанту. Зі сторони водія доступна навігація з точки А в точку Б. Розроблена система включає в себе бекенд, фронтенд, базу даних та хмарні технології.

Розроблено алгоритм спільних пасажирських перевезень, який базується на аналізі обов'язкових дій, таких як оновлення локації, збирання метеорологічних даних та даних щодо стану доріг, побудова маршруту.

Розроблено математичну модель організації спільних пасажирських перевезень, яка включає формулу розрахунку ваги ребра графа та формулу пошуку найкоротшого шляху.

В рамках роботи було виконано порівняння функціональних можливостей конкурентів, і виявлено значні переваги, а саме:

- можливість побудови як короткострокових маршрутів так і довгострокових;
- можливість вибору водія
- урахування погодних умов
- прогнозування трафіку;
- середній час відповіді на 100 запитів становить 0.29 секунди для SmartRoute, тоді як для Uber цей показник складає 0.49 секунд, а для BlaBlaCar — 1.3 секунди;

Окрім цього, з технічної точки зору, розроблена методика гарантує автоматичну масштабованість та ізоляцію операції за допомогою AWS Aurora DB Service, відмовостікость та асинхроність за допомогою AWS SQS Queue Service.

Завдяки хмарним технологіям ми змогли не тільки розширити функціональні можливості, а і покращити продуктивність додатка в порівнянні з конкурентами.

В результаті магістерської роботи основні положення та результати дослідження доповідалися та обговорювались на II міжнародної науково-практичної конференції «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії». В результаті було подано 2 тези на тему «Застосування хмарних технологій для покращення методу спільних пасажирських перевезень» «Розробка методики організації спільних пасажирських перевезень на основі хмарних технологій»

ПЕРЕЛІК ПОСИЛАНЬ

1. Ізтелеуова М., Грицук І., Арімбекова П. Організація та логістика перевезень : підручник. Олді+, 2021. 264 с.
2. Кальченко А. Г. Логістика : Підручник. Київ : КНЕУ, 2003. 284 с.
3. Основні алгоритми в теорії графів. *DOU*. URL: <https://dou.ua/forums/topic/48433/>(дата звернення: 14.12.2024).
4. Пасажирські перевезення: логістично-інфраструктурні аспекти. *World science*. 2018. URL: https://doi.org/10.31435/rsglobal_ws/12072018/6034(дата звернення: 14.12.2024).
5. Amazon simple queue service documentation. *Amazon Web Services, Inc.* URL: <https://docs.aws.amazon.com/sqs/>(дата звернення: 14.12.2024).
6. Grant D. B., Trautrim A., Wong C. Y. Sustainable logistics and supply chain management. Kogan Page, 2016. 256 p.
7. Labadie N., Prins C., Prodhon C. Metaheuristics for vehicle routing problems. Hoboken, NJ, USA : John Wiley & Sons, Inc., 2016. URL: <https://doi.org/10.1002/9781119136767>(date of access: 14.12.2024).
8. Navigation API. *Sygic*. URL: <https://www.sygic.com/developers/professional-navigation-sdk/windows/api-examples/navigation-api>(date of access: 14.12.2024).
9. OLTP database, mysql and postgresql managed database - amazon aurora - AWS. *Amazon Web Services, Inc.* URL: <https://aws.amazon.com/rds/aurora/>(date of access: 14.12.2024).
10. One Call API 3.0 - OpenWeatherMap. *Current weather and forecast - OpenWeatherMap*. URL: <https://openweathermap.org/api/one-call-3>(date of access: 14.12.2024).
11. Relational database – amazon aurora mysql postgresql features – AWS. *Amazon Web Services, Inc.* URL: <https://aws.amazon.com/rds/aurora/features/>(date of access: 14.12.2024).

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«МЕТОДИКА ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ ЗА ДОПОМОГОЮ ХМАРНИХ ТЕХНОЛОГІЙ»

Виконав: студент групи ПДМ-62 Бацунов Дмитро Сергійович

Керівник роботи: доктор філософії, доцент кафедри ІІЗ Гребенюк Віктор Вікторович

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи – удосконалення процесу методики організації спільних пасажирських перевезень за допомогою хмарних технологій.

Об'єкт дослідження – процес організації спільних пасажирських перевезень.

Предмет дослідження – метод організації спільних пасажирських перевезень.

АНАЛІЗ АНАЛОГІВ

Розглянуті методики та їх можливості	Uber	 BlaBlaCar	 SmartRoute
Перерахунок маршруту в реальному часі	+	-	+
Урахування складності пересування при розрахунку ціни	+	-	+
Вибір водія	-	+	+
Урахування погодних умов	-	-	+
Прогнозування трафіку	-	-	+
Підтримка довготривалих маршрутів	-	+	+
Підтримка короткотривалих маршрутів	+	-	+

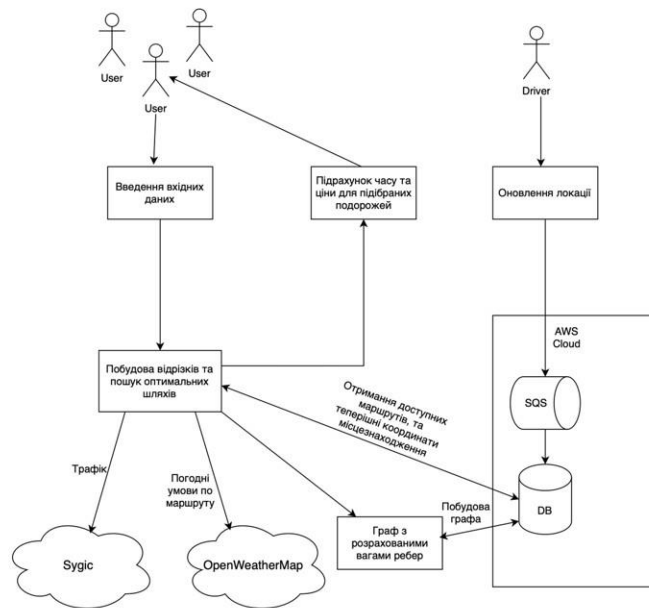
3

МЕТОДИКА ОРГАНІЗАЦІЇ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ

- Отримання адреси початку і кінця маршруту від користувача.
- Аналіз всіх наявних активних маршрутів в розрахованій межі на основі заданого початку і кінця.
- Збір інформації щодо навантаження маршруту та метеорологічних даних.
- Підрахунок найкоротшого шляху з урахуванням трафіку та погодних умов.
- Побудова таблиці можливих водіїв за допомогою Amazon Aurora Leaderboard з можливістю перерахунку при пересуванні в реальному часі.
- Пропозиція можливих водіїв та вибір бажаного варіанту користувачем з урахуванням ціни, часу та рейтингу водія.
- Супровід подорожі(моніторинг заданого маршруту та переробови у випадку будь-яких змін обставин).

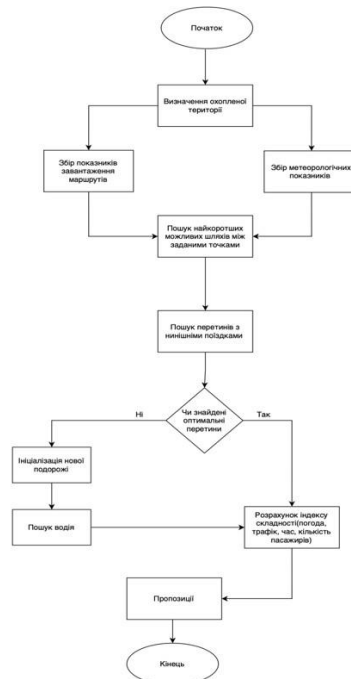
4

АЛОГРИТМ СПІЛЬНИХ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ



5

АЛГОРИТМ ПОШУКУ МАРШРУТУ



6

МАТЕМАТИЧНА МОДЕЛЬ

Розрахунок ваги ребра графа

$$W = \sum_k C_k \cdot i_k$$

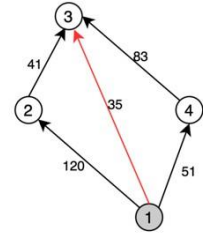

- C_k - враховані значення (дистанція, час, витрата, кількість пасажирів, наявність існуючого маршруту, завантаження дороги, погодні умови),
- i_k - індекси важливості кожного з врахованого значення (змінюються в залежності пріоритету бізнесу)

Пошук найкоротшого шляху в графі

$$d(i, j) = \min(d(i, j), d(i, k) + d(k, j))$$

C4 =

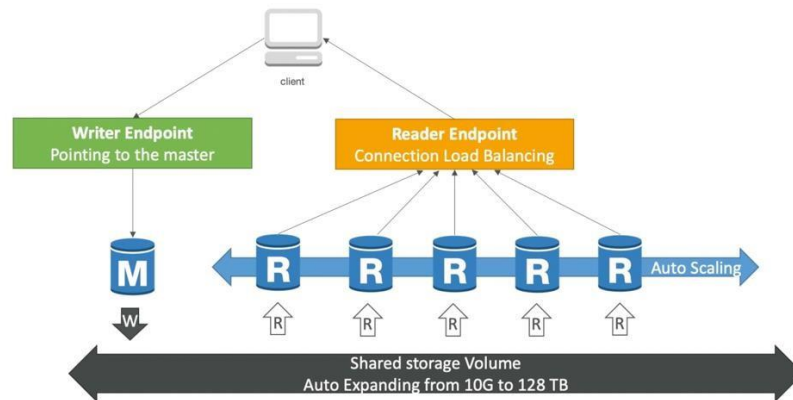
	V1	V2	V3	V4
V1	0	120	35	51
V2	∞	0	41	124
V3	∞	∞	0	83
V4	∞	∞	∞	0



- $d(i, j)$ — це поточне найкоротше відстань між вершинами ii та jj .
- $d(i, k)d(i, k)$ — це відстань від вершини ii до проміжної вершини kk .
- $d(k, j)d(k, j)$ — це відстань від вершини kk до вершини jj .

7

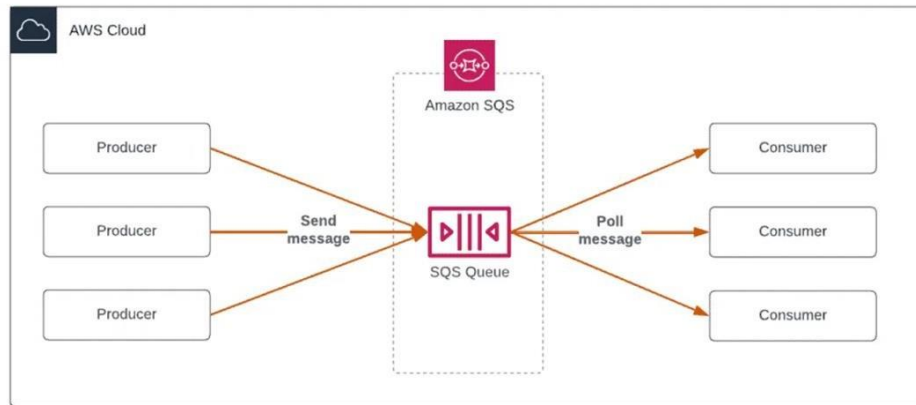
Amazon Aurora Cloud Storage



Ресурс: AWS Documentation

8

SQS Queue Cloud Service



Ресурс: AWS Documentation

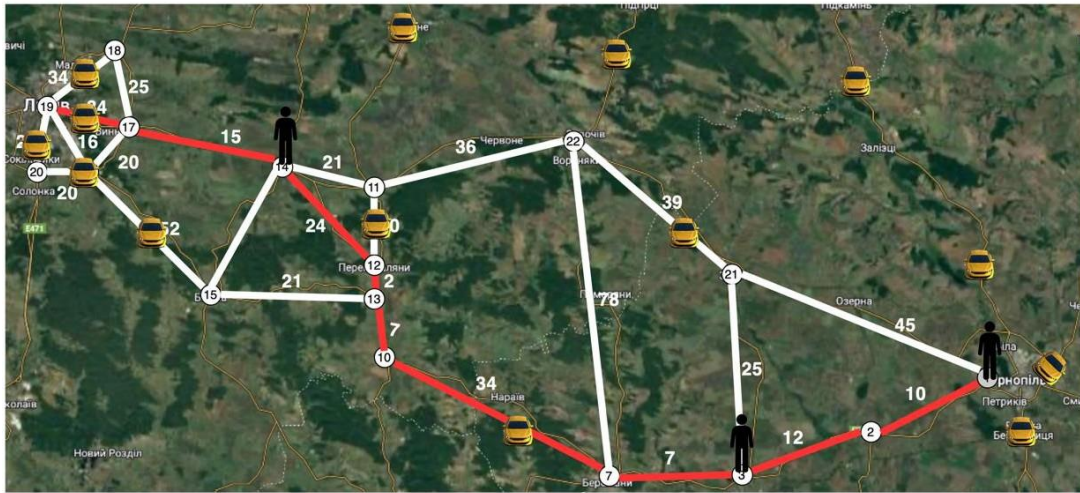
9

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



10

ПРАКТИЧНИЙ РЕЗУЛЬТАТ



11

ПРАКТИЧНИЙ РЕЗУЛЬТАТ

Деталі подорожі

Маршрут: Тернопіль - Львів
Відстань: 126.6 км
Бажаний дата відправки: -
Бажана дата прибуття: 26.11.25 10:00

	Robert	500UAH	13:30 - 16:00
	Mykhailo	450UAH	16:30 - 18:00
	Maks	300UAH	17:30 - 22:00
	Ivan	500UAH	08:30 - 09:20

Деталі подорожі

Маршрут: Тернопіль - Львів
Відстань: 126.6 км
Бажаний дата відправки: -
Бажана дата прибуття: 26.11.25 10:00

Маршрут: Київ - Варшава
Кількість пасажирів: 10
Приблизний індекс трафіку: 4
Погодні умови

	Robert	500UAH	13:30 - 16:00
	Maks	300UAH	17:30 - 22:00
	Ivan	500UAH	08:30 - 09:20

Деталі подорожі

Маршрут: Тернопіль - Львів
Відстань: 126.6 км
Бажаний дата відправки: -
Бажана дата прибуття: 26.11.25 10:00

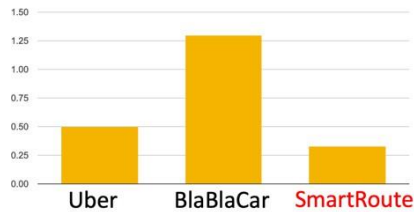
	Robert	500UAH	13:30 - 16:00
	Mykhailo	450UAH	16:30 - 18:00
	Maks	300UAH	17:30 - 22:00
	Ivan	500UAH	08:30 - 09:20

Вибрати

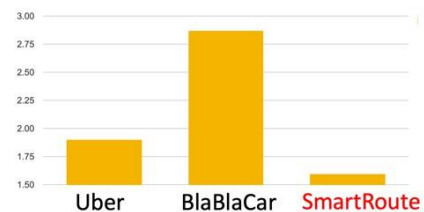
12

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

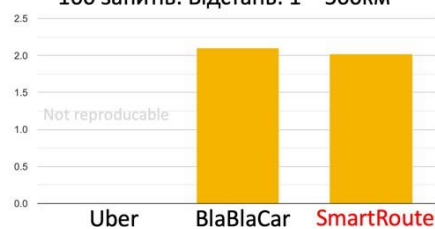
1. Середній час відповіді сервера
на 100 запитів



2. Середній час на побудову маршруту
100 запитів. Відстань: 1 – 500км



3. Швидкість знаходження перетину маршрутів та підбору водіїв
100 запитів. Відстань: 1 – 500км



13

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Тези доповідей

- Бацунов. Д.С. Застосування хмарних технологій для покращення методу спільних пасажирських перевезень / Д.С. Бацунов // Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії : Міжнародній науково-практичній конференції. Збірник тез, чекає на публікацію
- Бацунов. Д.С. Розробка методики організації спільних пасажирських перевезень на основі хмарних технологій / Д.С. Бацунов // Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії : Міжнародній науково-практичній конференції. Збірник тез, чекає на публікацію

14

ВИСНОВКИ

1. Проведено дослідження доцільності застосування хмарних технологій для організації спільних пасажирських перевезень яке продемонструвало переваги в адаптивності до змін, швидкості підбору маршрутів, незалежності до відстані маршруту та прогнозування інтенсивності руху на використовуваній ділянці дороги що забезпечило більшу прогнозованість.
2. Проаналізовано існуючі рішення щодо застосування методів організації спільних пасажирських перевезень та побудовано порівняльну характеристику.
3. Розроблено методику спільних пасажирських перевезень на основі хмарних технологій.
4. Проведено апробацію та тестування реалізованої методики та підтверджено доцільність використання створеної методики. Результатами моделювання є розширення функціональних можливостей у порівнянні з аналогами, а саме підтримка маршрутів різної тривалості, врахування погодних умов та завантаження доріг, вибір водія. Окрім цього було підтверджено покращення в продуктивності додатку в середньому на 30%