

DOI: <https://doi.org/10.15276/ict.02.2025.51>

УДК 004.9:656.3

Розробка інтерактивної платформи для пошуку автобусних квитків з використанням технологій розподіленої обробки великих даних

Тройніна Анастасія Сергіївна¹⁾

Канд. техніч. наук, доцент каф. Інженерії програмного забезпечення

ORCID: <https://orcid.org/0000-0001-6862-1266>; anastasiyatroynina@gmail.com. Scopus Author ID: 57193992712

Кулік Леонід Олегович¹⁾

Студент каф. Інженерії програмного забезпечення

ORCID: <https://orcid.org/0009-0005-4515-2635>; 9482038@stud.op.edu.ua

Назаров Іван Якович¹⁾

Студент каф. Інженерії програмного забезпечення

<https://orcid.org/0009-0005-3185-2390>; vet.nazarov7@gmail.com

¹⁾ Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

У сучасних умовах цифрової трансформації сфера пасажирських перевезень потребує інноваційних рішень, здатних забезпечити високу швидкість, точність та масштабованість у процесах планування маршрутів і тарифоутворення. Традиційні сервіси пошуку квитків часто характеризуються значними затримками обробки запитів і обмеженою гнучкістю у формуванні цінових пропозицій. Це зумовлює необхідність створення нових інформаційних систем, орієнтованих на інтеграцію технологій розподіленої обробки даних.

У роботі представлено розробку вебзастосунка для пошуку автобусних квитків у мультисегментних сполученнях із вбудованим модулем рекомендації тарифів. Система поєднує функціонал швидкого пошуку маршрутів, підтримку складних запитів із пересадками та автоматизоване ціноутворення. Як обчислювальне ядро використано Apache Spark, що дозволяє реалізувати розподілену обробку великих масивів даних. Серверна частина побудована з використанням Flask, а для зберігання інформації застосовано PostgreSQL. Користувачський інтерфейс реалізовано на основі HTML, CSS та JavaScript, що забезпечує доступність і зручність взаємодії.

Аналіз існуючих аналогів продемонстрував відсутність комплексних рішень, які б одночасно оптимізували пошук маршрутів і автоматизували тарифоутворення. Запропонована система долає ці обмеження завдяки інтеграції з API перевізників для збору актуальної інформації та впровадженню механізмів обробки великих даних. Експериментальне тестування підтвердило ефективність підходу: час відповіді системи скорочено до 300–500 мс, а точність рекомендацій досягла 92–95 %. У кластерному режимі система здатна обробляти понад 1 млн записів на годину, що свідчить про високу продуктивність і масштабованість рішення.

Практична значущість розробки полягає у можливості її застосування як окремого інструменту для перевізників або як складової комплексних транспортних платформ. Система сприяє оптимізації витрат, підвищенню прозорості цінової політики та покращенню користувацького досвіду, роблячи її перспективним рішенням для сучасної транспортної логістики.

Ключові слова: ціноутворення; розподілена обробка даних; Apache Spark; вебзастосунок; транспортна логістика; маршрути

Метою дослідження є проектування та розробка вебзастосунка з пошуку квитків з використанням технологій розподіленої обробки даних для підвищення його швидкодії та масштабованості.

Об'єктом дослідження є вебзастосунок для пошуку квитків із використанням технологій розподіленої обробки даних. Об'єкт включає архітектуру програмного забезпечення, базу даних, серверну логіку, користувацький інтерфейс та інтеграцію з обчислювальними платформами, що забезпечують підвищення швидкодії та масштабованості системи.

Дослідження об'єкта передбачає аналіз різних аспектів вебзастосунка для пошуку квитків, зокрема його ефективності у підвищенні швидкодії, масштабованості та зручності користувацької взаємодії. Особливу увагу зосереджено на архітектурі програмного забезпечення, структурі бази даних, серверній логіці та засобах інтеграції з зовнішніми сервісами. Для досягнення поставленої мети застосовуються сучасні інструменти проектування та візуалізації, методи інтеграції бази даних із вебінтерфейсом, а також технології розподіленої обробки даних (Apache Spark) для забезпечення високої продуктивності та надійності системи.

Архітектура розробленого вебзастосунка побудована відповідно до принципів масштабованості, гнучкості та ефективності обробки великих обсягів даних, що є ключовими

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/deed.uk>)

вимогами сучасних інформаційних систем у сфері пасажирських перевезень. Система реалізована як багаторівнева, де кожен логічний рівень відповідає за виконання власних функцій, взаємодіючи з іншими через чітко визначені інтерфейси. Такий підхід дозволяє розділити бізнес-логіку, користувацький інтерфейс та інструменти обробки даних, що робить систему зручною для подальшої підтримки й модернізації [1].

Загальна архітектурна модель ілюструється на рисунку 1, який відображає ключові компоненти та взаємозв'язки між ними. Клієнтська частина або frontend створена із використанням HTML, CSS та JavaScript. Вибір саме таких технологій пояснюється їх універсальністю, сумісністю з більшістю сучасних браузерів та простотою розгортання.

Клієнтська частина забезпечує інтерфейс для кінцевих користувачів, включаючи форми пошуку квитків, авторизації та реєстрації, а також візуалізацію результатів пошуку маршрутів. Вона виконує роль посередника між користувачем і сервером, збираючи параметри запитів і передаючи їх через HTTP-запити у форматі REST API до серверної частини. Завдяки цьому досягається уніфікована та стандартизована взаємодія між клієнтом і сервером, що робить можливим масштабування системи та її інтеграцію з іншими сервісами.

Серверна частина або backend реалізована за допомогою мови програмування Python із використанням вебфреймворку Flask. Основним її завданням є обробка API-запитів від клієнта, взаємодія з базою даних PostgreSQL. PostgreSQL використовується не лише як основне джерело даних, але й як платформа для інтеграції зі Spark, що забезпечується завдяки JDBC-конектору [2]. Такий підхід гарантує швидкий доступ до даних навіть при значному навантаженні та дозволяє підтримувати обробку понад одного мільйона записів на годину у кластерному режимі. Серверна частина генерує відповіді у форматі JSON, що дозволяє швидко передавати структуровані дані назад клієнту. Вибір Flask обумовлений його легкістю, гнучкістю та широкою підтримкою спільноти, що забезпечує стабільність та простоту інтеграції з іншими інструментами.

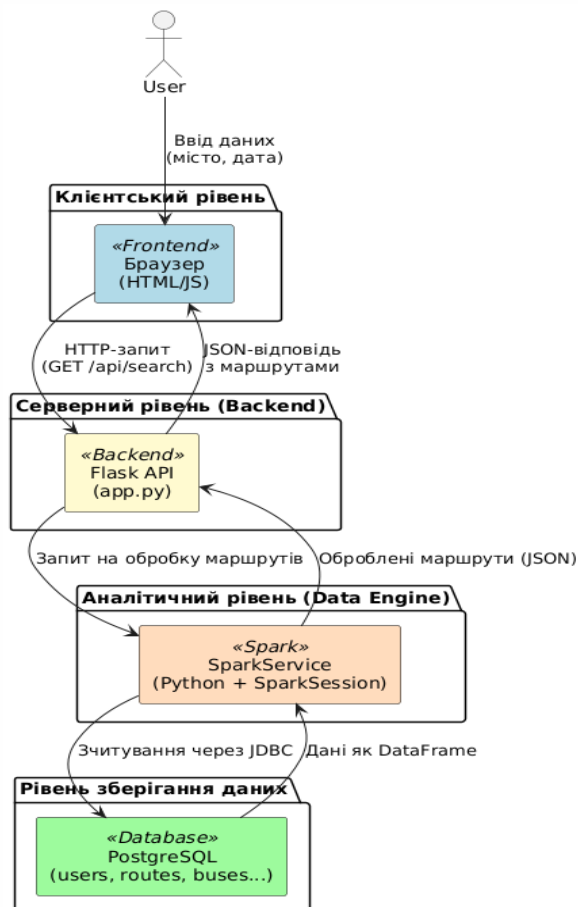


Рис. 1. Діаграма архітектури застосунка

Важливою складовою архітектури є модуль розподіленої обробки даних, реалізований на базі Apache Spark. Цей компонент інтегрований у систему через SparkSession, який запускається та керується з середовища Flask[4]. Основні завдання модуля полягають у зчитуванні даних із бази PostgreSQL через JDBC-конектор, виконанні агрегацій, фільтрації та обробки запитів, визначенні як прямих, так і комбінованих маршрутів, а також у розрахунку вартості квитків, тривалості поїздки та кількості пересадок. Однією з ключових переваг використання Apache Spark є можливість обробки даних у пам'яті (in-memory), що значно прискорює виконання запитів та забезпечує зменшення часу відгуку системи [3].

Після вибору архітектури наступним кроком є моделювання системи, що дозволяє визначити її внутрішню логіку та структуру. Для цього було створено діаграму класів, що представлена на рисунку 2. Вона демонструє структуру даних, методи та зв'язки між об'єктами й фактично формує внутрішнє ядро всієї програмної системи.

Ключовими класами є FlaskAPI, SparkService, DatabaseConnector, DictHelper, User, Driver, Bus, Route, City, а також допоміжні сутності Fuel і Distance. Кожен клас має свою чітко окреслену роль, що знижує рівень залежності між компонентами та підвищує модульність архітектури. FlaskAPI виступає фасадом серверної частини, реалізуючи REST API для пошуку маршрутів, керування користувачами, водіями, автобусами та маршрутами. Він забезпечує інтеграцію з іншими модулями, передаючи запити до SparkService для аналітичної обробки, DatabaseConnector для взаємодії з базою даних та DictHelper для перетворення результатів у формат JSON. SparkService реалізує розподілену обробку даних на основі Apache Spark, що дозволяє ефективно працювати з великими обсягами інформації у пам'яті, здійснювати пошук маршрутів, визначати пересадки та оптимальні поїздки. DatabaseConnector відповідає за єдиний спосіб підключення до PostgreSQL, забезпечуючи стабільний доступ до сховища даних, тоді як DictHelper виконує допоміжну роль у форматуванні результатів. Класи предметної області (User, Driver, Bus, City, Distance, Fuel, Route) описують логічні сутності системи. Вони відображають дані користувачів, транспортних засобів, міст і відстаней між ними, а також містять бізнес-логіку для опису маршруту як ключового елемента платформи.

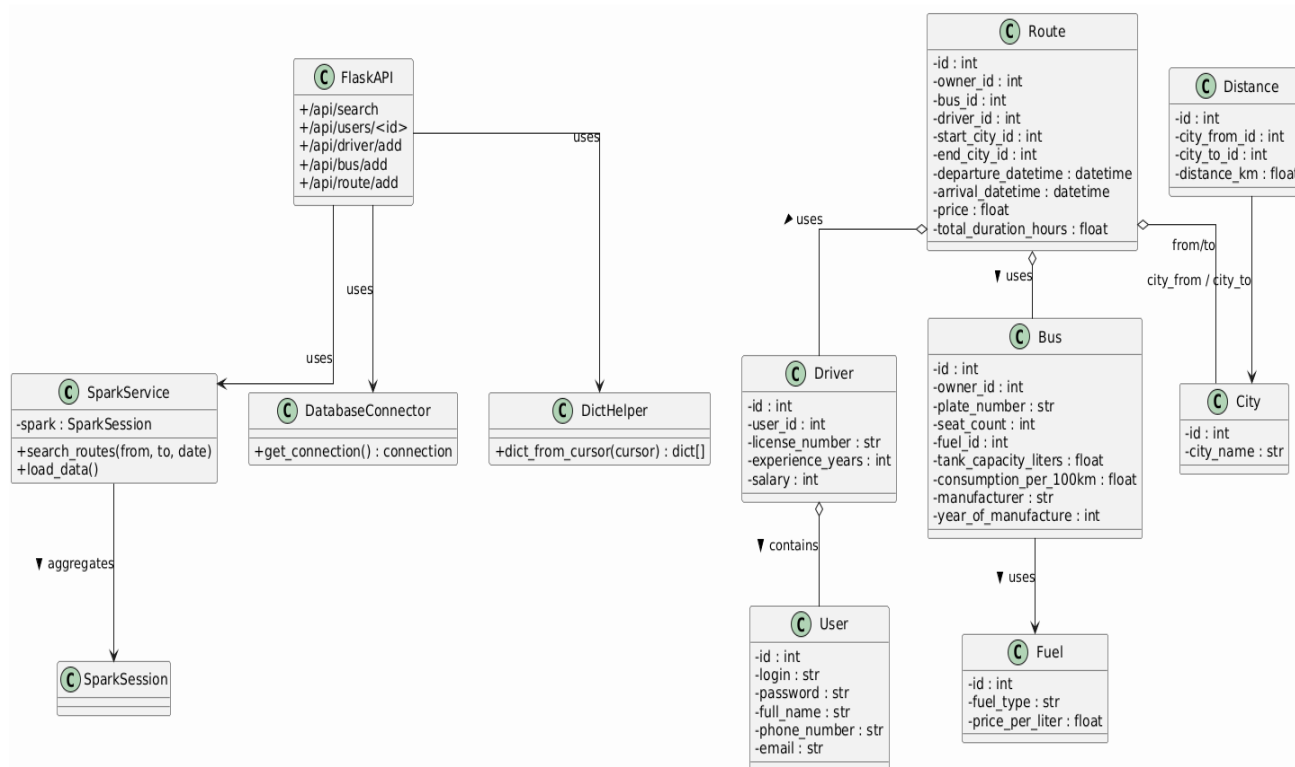


Рис. 2. Діаграма класів

Спроекована діаграма класів виступає основою для наступного етапу реалізації програмного продукту, оскільки вона визначає не лише статичну структуру системи, але й способи взаємодії між її складовими.

У процесі створення вебзастосунка наступним важливим етапом після побудови архітектури та моделі класів стало проектування бази даних, яка є ключовим елементом системи, оскільки забезпечує зберігання та доступ до структурованої інформації про користувачів, водіїв, автобуси, маршрути, відстані між містами та інші сутності [5]. Для реалізації було обрано реляційну систему керування базами даних PostgreSQL, яка завдяки своїй продуктивності, стабільності та підтримці складних зв'язків між таблицями дозволяє ефективно працювати з великими обсягами даних. Було створено схему бази даних, зображену на рисунку 3.

У структурі передбачено сім основних таблиць. Таблиця “users” містить дані про користувачів, включно з логіном, паролем, ПІБ, контактними даними та електронною поштою. Вона слугує базою для аутентифікації та авторизації у системі, а також для зв'язку з іншими сутностями, зокрема водіями та автобусами. Таблиця “drivers” відповідає за дані водіїв, зберігаючи посилання на користувачів, ліцензійні дані, досвід роботи та зарплату. Вона є логічним розширенням ролі користувача та використовується при формуванні рейсів. Таблиця “cities” містить унікальний список міст, що можуть виступати точками відправлення чи прибуття. Вона є базовим елементом для маршрутизації, тоді як таблиця “distances” відображає відстані між двома містами, що дозволяє виконувати розрахунок вартості та тривалості маршруту. Таблиця “fuels” зберігає типи пального та їхню актуальну вартість, що забезпечує можливість врахування витрат на паливе при формуванні цінової політики. Таблиця “buses” зберігає інформацію про транспортні засоби: власників, кількість місць, витрати палива, ємність баку, рік випуску та виробника. Вона пов'язана як із користувачами, так і з типами палива. Найважливішою є таблиця “routes”, що містить інформацію про рейси: власника, автобус, водія, початкове та кінцеве місто, час відправлення та прибуття, тривалість у годинах і ціну.

Створена база даних є основою для функціонування всієї системи, забезпечуючи централізоване зберігання, швидкий доступ і логічну узгодженість інформації. Це дозволяє не лише реалізувати пошук квитків у режимі реального часу, а й гнучко масштабувати систему для подальшої інтеграції з аналітичними модулями, зокрема Apache Spark, що забезпечує розподілену обробку запитів.

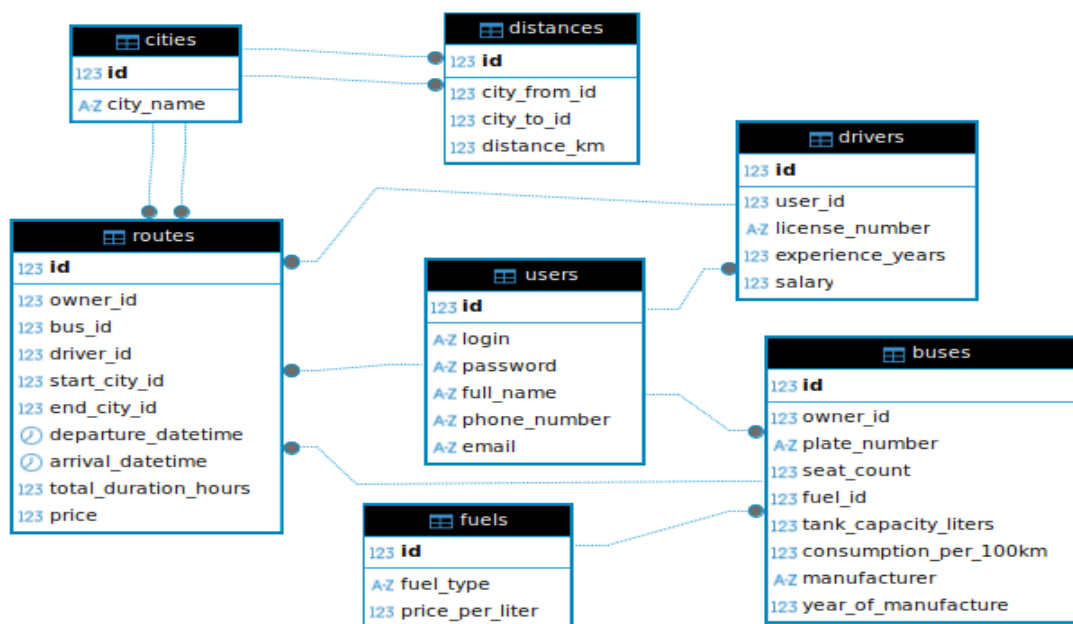


Рис. 3. Діаграма бази даних

Наступним кроком є розробка елементу інтерфейсу користувача. Розробка інтерфейсу користувача передбачає створення інтуїтивно зрозумілих форм та панелей для взаємодії кінцевого користувача із системою [6]. Основні елементи включають пошукові форми для введення параметрів маршруту (місто відправлення, місто прибуття, дата, кількість пасажирів), таблиці відображення результатів із деталями поїздки, варіанти сортування та фільтрації, а також модуль авторизації та реєстрації користувачів.

Створено низку функціональних елементів, що забезпечують зручність та інтуїтивність взаємодії з вебзастосунком. До основних елементів належать вікно входу, яке дозволяє користувачу здійснити авторизацію за допомогою логіну або електронної пошти та паролю, забезпечуючи безпечний доступ до персоналізованих функцій системи. Для зареєстрованих користувачів реалізовано меню, яке представляє собою вертикальний список доступних сторінок та функцій, що дозволяє швидко переходити між різними розділами застосунку. Крім того, створено вікно сортування, яке дає змогу впорядковувати маршрути за ціною або часом у дорозі, а також встановлювати фільтри для обмеження максимальної тривалості

Реалізовані сторінки з відображенням результатів пошуку маршрутів, які включають загальні поля для введення міст відправлення та прибуття, кнопки запуску пошуку та можливості вибору критерію сортування. Дані про маршрути відображаються у вигляді таблиці, що дозволяє користувачу швидко ознайомитися з усіма параметрами поїздки та здійснити вибір оптимального варіанту. Для нових користувачів створено сторінку реєстрації, що включає поля для введення логіну, електронної пошти та паролю, а для авторизованих користувачів передбачено сторінку налаштувань профілю з полями для редагування персональних даних, включаючи номер телефону та пароль.

Було розроблено сторінки профілю користувача, де відображаються персональні дані та інформація про виконані або заплановані поїздки, а також сторінки для реєстрації транспортних засобів, водіїв та маршрутів. Кожна із цих сторінок включає спеціальні поля для вводу необхідної інформації та кнопки підтвердження, що забезпечують інтеграцію введених даних із базою даних системи. Реалізація цих елементів дозволяє користувачу зручно управляти власними обліковими записами, здійснювати пошук та бронювання маршрутів, а адміністраторам ефективно додавати та редагувати дані про транспортні засоби, водіїв та маршрути, підвищуючи загальну функціональність та зручність користування вебзастосунком.

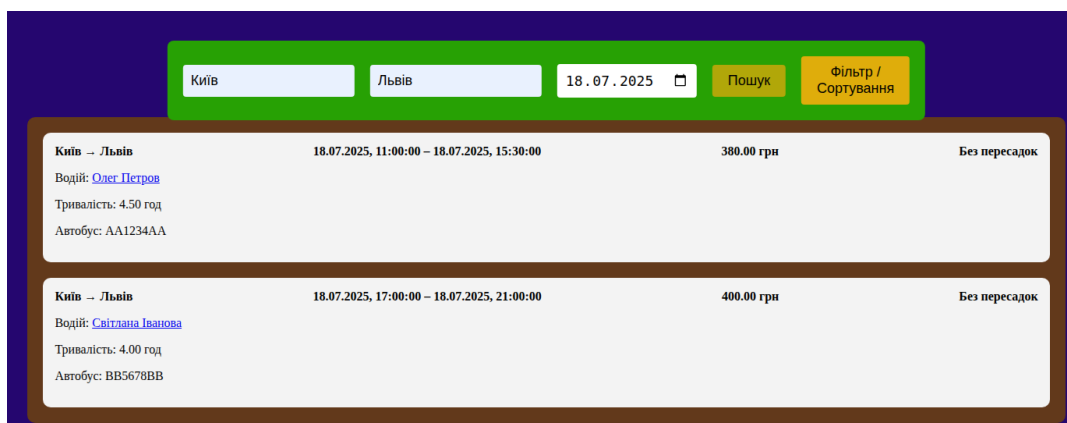


Рис. 4. Приклад елементу інтерфейсу користувача

В результаті проведення цього дослідження було створено програмну систему для пошуку автобусних квитків. За допомогою використання технологій розподіленої обробки даних час відповіді системи скорочено до 300–500 мс. Система здатна обробляти 1 млн. запитів на годину. Використання методів розподіленої обробки великих даних та впровадження оптимальних підходів до обробки маршрутної інформації сприятимуть підвищенню швидкодії системи, точності розрахунків та зручності користувацького

інтерфейсу, що дозволить більш ефективно підтримувати планування поїздок і покращувати досвід користувачів у цифровому середовищі.

СПИСОК ЛІТЕРАТУРИ

1. Matthew N., Stones R. “Beginning Databases With PostgreSQL: From Novice To Professional”. – URL: <https://theswissbay.ch/pdf/Gentoomen%20Library/Databases/postgresql/Beginning%20Databases%20With%20PostgreSQL%20-%20From%20Novice%20To%20Professional%2C%202nd%20Ed%202005.pdf>.
2. Zaharia M., Xin R. S., Wendell P., et al. “Apache Spark: a unified engine for big data processing”. *Communications of the ACM*. 2016; 59 (11): 56–65. DOI: <https://doi.org/10.1145/2934664>.
3. Chambers B., Zaharia M. “Spark: The Definitive Guide – Big Data Processing Made Simple”. – URL: https://analyticsdata24.wordpress.com/wp-content/uploads/2020/02/spark-the-definitive-guide40www.bigdatabugs.com_.pdf.
4. Dwyer G., Aggarwal S., Stouffer J. “Flask: Building Python Web Services”. *Packt Publishing*. 2017.
5. Stonebraker M. “SQL databases v. NoSQL databases”. *Communications of the ACM*. 2010; 53 (4): 10–11. DOI: <https://doi.org/10.1145/1721654.1721659>.
6. Galitz W. O. “The Essential Guide to User Interface Design”. *Wiley Publishing, Inc.* 2007. – URL: <https://content.e-bookshelf.de/media/reading/L-569772-a8c49f6b40.pdf>.

DOI: <https://doi.org/10.15276/ict.02.2025.51>

UDC 004.9:656.3

Development of an interactive platform for searching bus tickets using distributed big data processing technologies

Anastasiia S. Troinina¹⁾

PhD, Associate Professor of the Department of Software Engineering

ORCID: <https://orcid.org/0000-0001-6862-1266>; anastasiyatroinina@gmail.com. Scopus Author ID: 57193992712

Leonid O. Kulik¹⁾

Student of the Department of Software Engineering

ORCID: <https://orcid.org/0009-0005-4515-2635>; 9482038@stud.op.edu.ua

Ivan Y. Nazarov¹⁾

Student of the Department of Software Engineering

ORCID: <https://orcid.org/0009-0005-3185-2390>; vet.nazarov7@gmail.com

¹⁾ Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

In the modern context of digital transformation, the passenger transportation sector requires innovative solutions capable of ensuring high speed, accuracy, and scalability in route planning and fare calculation processes. Traditional ticket search services are often characterized by significant query processing delays and limited flexibility in forming pricing proposals. This highlights the need for new information systems oriented toward the integration of distributed data processing technologies.

This work presents the development of a web application for searching bus tickets across multi-segment connections with a built-in fare recommendation module. The system combines fast route search functionality, support for complex queries with transfers, and automated fare calculation. Apache Spark is used as the computational core, enabling distributed processing of large data sets. The server-side is built using Flask, while PostgreSQL is employed for data storage. The user interface is implemented using HTML, CSS, and JavaScript, ensuring accessibility and ease of interaction.

Analysis of existing analogs demonstrated the absence of comprehensive solutions that simultaneously optimize route search and automate fare calculation. The proposed system overcomes these limitations through integration with carrier APIs to collect up-to-date information and the implementation of big data processing mechanisms. Experimental testing confirmed the effectiveness of the approach: system response time was reduced to 300–500 ms, and recommendation accuracy reached 92–95%. In cluster mode, the system can process over 1 million records per hour, indicating high performance and scalability.

The practical significance of this development lies in its potential application as a standalone tool for carriers or as part of comprehensive transportation platforms. The system contributes to cost optimization, increased transparency of pricing policies, and improved user experience, making it a promising solution for modern transportation logistics.

Keywords: Fare calculation; distributed data processing; Apache Spark; web application; transportation logistics; routes