

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ МОРСЬКИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут інформаційних технологій
та інноваційного підприємництва

(повна назва інституту)

Кафедра технічної кібернетики й інформаційних технологій
ім. професора Р.В. Мерктя

(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

на тему

Розробка системи з аналізу транспортних перевезень та

логістичних процесів

Виконала: здобувачка

вищої освіти

4 курсу, групи _____

Спеціальність:

124 «Системний аналіз»

(шифр і назва напрямку підготовки)


(підпис)

Таїсія СИТНИК

(Ім'я, ПРІЗВИЩЕ)

Керівник

к.ф.-м.н., доцент

(вчене звання, посада)


(підпис)

Марина РОЗУМ

(Ім'я, ПРІЗВИЩЕ)

Рецензент

старший викладач

(вчене звання, посада)


(підпис)

Ілля РУБЛЬОВ

(Ім'я, ПРІЗВИЩЕ)

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ МОРСЬКИЙ УНІВЕРСИТЕТ

Інститут Навчально-науковий інститут інформаційних технологій та інноваційного підприємництва

Кафедра технічної кібернетики й інформаційних технологій ім. професора Р.В. Меркта

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки -

(шифр і назва)

Спеціальність 124 «Системний аналіз»

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

КТЖІТ ім. проф. Р.В. Меркта

к.т.н., доцент Носов П.

«09» 03 20 26 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Таїсії СИТНИК

(Ім'я, ПРІЗВИЩЕ)

1. Тема проекту (роботи) Розробка системи з аналізу транспортних перевезень та логістичних процесів

керівник проекту (роботи) Марина РОЗУМ, к.ф.-м.н., доцент

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «13» 03 20 26 р. №61 вк/до

2. Строк подання проекту (роботи) Квітень 2026

3. Вихідні дані до проекту (роботи) Наукові джерела з транспортної логістики, дані про перевезення, сучасні методи бізнес-аналітики та програмні засоби розробки інформаційних систем.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз проблем в контролі та аналітиці за транспортною галуззю та ринком перевезень. Аналіз програмних рішень в сфері аналізу за транспортними перевезеннями. Аналіз цифрових методів з контролю за ефективністю транспортної логістики. Аналіз систем збереження та керування даними для інформаційних систем аналізу транспортних перевезень та логістичних процесів Розробка системи аналізу ефективності транспортних перевезень. Розробка логіко-математичної моделі аналізу логістичних процесів. Розробка системи прийняття рішень та оцінки ефективності вантажних перевезень. Розробка архітектурної логіки для інформаційної системи. Розробка та опис інтерфейсу інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Діаграма варіантів використання, діаграма послідовності дій, діаграма класів та діаграма бази даних

6. Консультанти розділів проекту (роботи)

Розділ	Ім'я, ПРІЗВИЩЕ та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сергій ПЕРЕТЯКА, доцент БЖД ЕХ	9.03.26.9	20.04.26

7. Дата видачі завдання 09.03.2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів проекту (роботи)
1	Видача завдання	09.03.2026
2	Переддипломна практика, залік	09.03.2026-22.03.2026
3	Коригування завдання за результатами практики	25.03.2026
4	Проміжний звіт на кафедрі оцінка готовності	01.04.2026
5	Попередній захист на кафедрі	15.04.2026
6	Рецензування	17.04.2026
7	Захист на засіданні ЕК	21.04.2026

Здобувач вищої освіти


(підпис)
Таїсія СИТНИК
(Ім'я, ПРІЗВИЩЕ)

Керівник проекту (роботи)


(підпис)
Марина РОЗУМ
(Ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Метою кваліфікаційної роботи є підвищення ефективності транспортно-логістичних операцій та оптимізація методів обробки запитів на перевезення. Розроблена інформаційна аналітична система забезпечує комплексну підтримку процесів управління перевезеннями, дозволяє здійснювати моніторинг та контроль маршрутів, аналізувати вантажопотоки та ефективність використання транспортних засобів, а також здійснювати прогнозування навантаження та планування ресурсів.

Реалізацію системи виконано засобами JavaScript, React та TypeScript. Для збереження даних в системі використовується MongoDB.

Розроблене рішення забезпечує комплексну підтримку транспортно-логістичних процесів у реальному часі та дозволяє вирішувати ключові завдання: планування маршрутів, контроль завантаження транспортних засобів, управління ресурсами, прогнозування навантаження, а також моніторинг ефективності роботи водіїв і персоналу.

ABSTRACT

The purpose of the qualification work is to increase the efficiency of transport and logistics operations and optimize the methods of processing transportation requests. The developed information analytical system provides comprehensive support for transportation management processes, allows for monitoring and control of routes, analyzing cargo flows and vehicle efficiency, as well as for load forecasting and resource planning.

The system was implemented using JavaScript, React and TypeScript. MongoDB is used to store data in the system.

The developed solution provides comprehensive support for transport and logistics processes in real time and allows solving key tasks: route planning, vehicle loading control, resource management, load forecasting, as well as monitoring the efficiency of drivers and personnel.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА СИСТЕМ КОНТРОЛЮ ТА АНАЛІТИКИ В ЛОГІСТИЦІ ТА ВАНТАЖНИХ ПЕРЕВЕЗЕННЯХ	9
1.1 Аналіз проблем в контролі та аналітиці за транспортною галуззю та ринком перевезень.....	9
1.2 Аналіз програмних рішень в сфері аналізу за транспортними перевезеннями	12
1.3 Аналіз цифрових методів з контролю за ефективністю транспортної логістики.	23
1.4 Аналіз систем технічної реалізації інформаційної системи аналізу транспортних перевезень та логістичних процесів.....	27
1.5 Аналіз систем збереження та керування даними для інформаційних систем аналізу транспортних перевезень та логістичних процесів	30
1.6 Мета кваліфікаційної роботи.....	32
1.7 Висновки до першого розділу	33
2 РОЗРОБКА ЛОГІКИ АНАЛІЗУ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ПЕРЕВЕЗЕНЬ ДЛЯ ЛОГІСТИЧНИХ КОМПАНІЙ.....	35
2.1 Розробка системи аналізу ефективності транспортних перевезень	35
2.2 Розробка логіко-математичної моделі аналізу логістичних процесів	38
2.3 Розробка системи прийняття рішень та оцінки ефективності вантажних перевезень.....	44
2.4 Висновки до другого розділу.....	53
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ З АНАЛІЗУ ТРАНСПОРТНИХ ПЕРЕВЕЗЕНЬ ТА ЛОГІСТИЧНИХ ПРОЦЕСІВ	56
3.1 Розробка архітектурної логіки для інформаційної системи	56
3.2 Розробка та опис інтерфейсу інформаційної системи.....	70
3.3 Аналіз результатів використання розробленої системи.....	93
3.4 Висновки до третього розділу	105
4 ОРГАНІЗАЦІЯ ПРОТИПОЖЕЖНИХ ЗАХОДІВ В ОФІСАХ ІТ-ГАЛУЗІ.....	108
4.1 Специфічні особливості офісів в ІТ-галузі	108

4.2 Пожежна оцінка офісу в ІТ-галузі.....	109
4.3 Технічні рішення захисту офісу від пожежі в ІТ-галузі.....	110
4.4 Людський фактор у системі пожежної безпеки ІТ-офісу	111
4.5 Висновки до четвертого розділу	113
ВИСНОВКИ	115
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	117
ДОДАТОК А	120

ВСТУП

Актуальність розробки аналітичних систем для оцінки ефективності транспортних перевезень та управління логістичними процесами зумовлена низкою взаємопов'язаних факторів, що визначають сучасний розвиток транспортної та логістичної індустрії. Зростання обсягів вантажопотоків, посилення конкуренції на ринку перевезень, підвищення вимог до швидкості доставки та якості логістичних послуг, а також зростання складності ланцюгів постачання створюють нові виклики для підприємств та операторів транспортної сфери. У таких умовах традиційні підходи до управління перевезеннями — що базуються переважно на досвіді диспетчерів і ретроспективному аналізі маршрутів — виявляються недостатньо ефективними. Для забезпечення оптимізації витрат, підвищення продуктивності та надійності перевезень необхідний перехід до data-driven методів, які ґрунтуються на комплексному аналізі даних, прогнозуванні та підтримці прийняття управлінських рішень.

Сучасні логістичні та транспортні системи акумулюють великі обсяги різномірної інформації: дані про маршрути, вантажопотоки, технічний стан транспортних засобів, терміни доставки, тарифи та витрати, дані про водіїв та екіпажі, зовнішні фактори (погодні умови, затори, стан доріг), а також показники ефективності обробки вантажів і управління запасами. Однак наявність великих обсягів даних сама по собі не гарантує підвищення ефективності логістичних процесів — ключовим є їх систематизована обробка, аналітична інтерпретація, виявлення прихованих закономірностей і прогнозування навантаження та попиту на транспортні ресурси.

Аналітичні системи для транспортних перевезень дозволяють вирішувати комплекс завдань, що виходять за межі звичайного обліку: оцінка ефективності маршрутів і завантаження транспортних засобів, прогнозування попиту на перевезення, оптимізація ресурсів (водії, техніка, паливо), аналіз витрат і прибутковості перевезень, контроль за дотриманням графіків, виявлення аномалій у

логістичних процесах, оцінка продуктивності персоналу та підвищення надійності ланцюгів поставок.

Особливу актуальність така аналітика набуває в умовах українського ринку, де транспортна інфраструктура та логістичні ланцюги часто стикаються з високою варіативністю попиту, регіональними відмінностями у стані доріг, обмеженою пропускнуою спроможністю, сезонними коливаннями вантажопотоків та потребою оперативної адаптації до змін нормативно-правового середовища. У цих умовах аналітичні системи стають не лише інструментом підвищення ефективності перевезень, але й стратегічним ресурсом для зниження витрат, підвищення безпеки, оптимізації ресурсів та підвищення конкурентоспроможності транспортних компаній.

Розробка спеціалізованих аналітичних систем для логістики та транспортних перевезень є багатогранним науково-прикладним завданням, що поєднує аспекти бізнес-аналітики, інформаційних технологій, транспортної економіки, теорії логістики та математичного моделювання. Такі системи дозволяють переходити від реактивного управління перевезеннями до проактивного та прогнозного підходу, забезпечуючи обґрунтованість рішень на всіх рівнях — від оперативного (контроль за маршрутами, завантаженням транспортних засобів, планування робочого часу водіїв) до стратегічного (оптимізація ланцюгів поставок, планування ресурсів, мінімізація витрат, підвищення надійності логістичних операцій).

Таким чином, створення ефективних аналітичних систем контролю та управління транспортними перевезеннями набуває стратегічного значення для розвитку транспортної індустрії. Вони сприяють не лише підвищенню ефективності логістичних процесів, але й формуванню стійкої конкурентної переваги компаній у сфері вантажоперевезень, забезпечують оптимізацію ресурсів, зниження витрат та підвищення рівня обслуговування клієнтів. Розробка та впровадження таких систем є одним із ключових напрямів сучасних досліджень у сфері транспортної аналітики, логістики та інформаційних технологій управління.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА СИСТЕМ КОНТРОЛЮ ТА АНАЛІТИКИ В ЛОГІСТИЦІ ТА ВАНТАЖНИХ ПЕРЕВЕЗЕННЯХ

1.1 Аналіз проблем в контролі та аналітиці за транспортною галуззю та ринком перевезень

На сучасному етапі розвитку транспортної галузі спостерігається стійке зростання вимог до прискорення доставки вантажів, підвищення якості транспортних послуг та зменшення витрат на виконання транспортно-складських операцій [1]. Транспортні вузли відіграють центральну роль у системі перевезень, оскільки саме в них розпочинається й завершується доставка вантажів, а також здійснюється перевалка вантажів між різними видами транспорту. Попри проведену реконструкцію та модернізацію транспортних підприємств, наявні перевантажувальні потужності не задовольняють потреби в повному обсязі, що створює серйозне виклики, подолання яких здатне суттєво підвищити загальну ефективність функціонування.

Одним із перспективних напрямів підвищення ефективності транспортних вузлів є оптимізація процедур планування завантаження транспортних засобів, вибору маршрутів руху та оцінки результативності перевезень [2]. Реалізація цього напрямку вимагає широкого впровадження сучасних інформаційних і комп'ютерних технологій. Забезпечення ефективної діяльності транспортного комплексу, який характеризується багатогранністю цілей, передбачає облік значної кількості взаємопов'язаних технічних, економічних та соціальних обмежень, що залежать від численних зовнішніх і внутрішніх факторів.

Перевантажувальні процеси мають динамічний характер і постійно еволюціонують під впливом змін у структурі вантажопотоків, а також трансформацій у діяльності транспортних підприємств та обслуговуваних територій. Це зумовлює необхідність максимальної формалізації підходів до прийняття рішень в оперативному управлінні та модернізації перевантажувальних операцій. Створення дієвої системи управління технологічними процесами в транспортних вузлах виступає одним із ключових засобів скорочення витрат на розширення

перевантажувальних потужностей та покращення їх експлуатаційних характеристик [3,4].

У контексті перевантаження вантажів у транспортних вузлах відповідні процеси розглядаються як об'єкти моделювання, що дає змогу розв'язувати задачі оптимального управління різними сценаріями перевантажувальних операцій [5].

Процеси, які відбуваються в інфраструктурі транспортних вузлів, мають стохастичну природу, що дозволяє класифікувати їх як імовірнісні моделі та відносити до класу моделей масового обслуговування. Основним завданням залишається забезпечення ефективної координації та взаємодії між усіма учасниками процесу обробки вантажів у транспортному вузлі. Досягнення цієї мети потребує розробки технологічних рішень, спрямованих на вирішення таких завдань:

- створення центру інтеграції інформаційних систем на основі координаційного центру;
- налагодження взаємодії наявних інформаційних систем із координаційним центром з метою забезпечення сумісності сервісів, синхронізації функціонування застосунків та обміну транзакціями;
- організація інформаційної взаємодії всередині логістичного ланцюга для супроводу вантажопотоку відповідними даними.

Система контролю вантажних операцій характеризується безперервним розвитком, зумовленим трансформацією потреб у переробці вантажів, а також змінами в стані транспортних підприємств та регіонів обслуговування [5]. Забезпечення ефективної взаємодії всіх суб'єктів, залучених до обробки вантажів, охоплює широкий спектр учасників транспортного простору, зокрема:

- законодавчі та виконавчі органи влади;
- державні органи контролю;
- суб'єкти економічної та соціальної інфраструктури;
- підприємства-перевізники та постачальники телекомунікаційних послуг.

Головною складністю виступає комплексна організація логістичного планування, що пов'язана зі значною кількістю потенційно непередбачуваних факторів.

На сучасному етапі розвитку транспортних перевезень посилюється акцент на підвищенні вимог до строків доставки вантажів, якості послуг та зниженні витрат на транспортно-складські операції [4]. Ключовими елементами системи є транспортні вузли, в яких розпочинається й завершується доставка вантажів, а також реалізуються процеси перевалки між різними видами транспорту. Незважаючи на здійснену реконструкцію та модернізацію транспортних підприємств, дефіцит перевантажувальних потужностей зберігається, що становить суттєву проблему, розв'язання якої може значно підвищити ефективність функціонування транспортного сектору [5].

У цьому контексті оптимізація планування завантаження транспортних засобів, маршрутизації та оцінки ефективності окремих перевезень розглядається як реальний напрям підвищення продуктивності транспортних вузлів. Досягнення цієї мети забезпечується завдяки застосуванню сучасних інформаційних і комп'ютерних технологій.

Для ефективного виконання багатогалузевих завдань транспортного комплексу необхідно враховувати численні взаємопов'язані обмеження технічного, економічного та соціального характеру, що зазнають впливу різноманітних чинників. Перевантажувальні процеси відрізняються постійною еволюцією, зумовленою змінами в обсягах і структурі вантажопотоків, а також динамікою умов діяльності транспортних підприємств і регіонів. Це зумовлює потребу в максимальній формалізації методів оперативного управління, модернізації та реорганізації перевантажувальних процесів.

Розробка дієвої системи координаційного управління технологічними процесами в транспортних вузлах є одним із основних шляхів зменшення витрат на розширення перевантажувальних потужностей та покращення їх експлуатаційних і техніко-економічних показників. У даній роботі перевантажувальні процеси в

транспортному вузлі трактується як об'єкти моделювання, що дозволяє розв'язувати задачі оптимального управління різними варіантами перевантаження вантажів.

Стохастичний характер процесів у транспортних вузлах дає підстави розглядати їх у рамках моделей масового обслуговування. Водночас ключовою проблемою залишається організація ефективної взаємодії між усіма учасниками обробки вантажів у транспортному вузлі [5]. Вирішення цього завдання вимагає впровадження технологічних підходів, що охоплюють такі аспекти:

- створення центру інтеграції інформаційних систем на базі координаційного центру;
- забезпечення взаємодії наявних інформаційних систем із координаційним центром для реалізації сумісних сервісів, синхронізації застосунків та передачі транзакцій;
- організація інформаційного обміну в межах логістичного процесу для супроводу вантажопотоку.

Контроль вантажних операцій характеризується безперервною еволюцією, зумовленою змінами в потребах переробки вантажів та динамікою умов у транспортних підприємствах і регіонах обслуговування. До основних суб'єктів транспортного простору належать організації політичної, фінансової та соціальної інфраструктури, зокрема: законодавчі та виконавчі органи влади; державні контролюючі органи (транспортна інспекція, митні, прикордонні та податкові служби); суб'єкти економічної та соціальної інфраструктури; фінансові установи (кредитні організації, страхові компанії); підприємства-перевізники; постачальники телекомунікаційних послуг та провайдери доступу до Інтернету.

Організація ефективної взаємодії між усіма зазначеними учасниками в процесі обробки вантажів є складним завданням через значну кількість непередбачуваних факторів.

1.2 Аналіз програмних рішень в сфері аналізу за транспортними перевезеннями

Транспортно системи (ТС) відіграють важливу роль у забезпеченні ефективного та своєчасного переміщення товарів і послуг від виробників до споживачів [6-8]. Основні аспекти їхньої ролі включають:

- Оптимізація ланцюга поставок. ТС забезпечують координацію всіх етапів ланцюга поставок, від планування та прогнозування до управління запасами і доставкою. Це дозволяє зменшити витрати та час на виконання замовлень.
- Підвищення ефективності. За допомогою автоматизації та аналізу даних ТЛС сприяють зменшенню витрат на транспортування, знижують ризик затримок і забезпечують своєчасну доставку вантажів.
- Поліпшення обслуговування клієнтів. ТС дозволяють компаніям краще управляти замовленнями та відстежувати їх статус у режимі реального часу, що підвищує рівень задоволеності клієнтів.
- Управління ризиками. ТС допомагають ідентифікувати та мінімізувати ризики в ланцюзі поставок, забезпечуючи надійність та безпеку перевезень.
- Гнучкість та адаптивність. ТС дозволяють швидко реагувати на зміни ринкових умов, потреби клієнтів та зовнішні фактори, забезпечуючи адаптивність логістичних процесів.

Програмна структура ТС зазвичай включає кілька основних модулів та підсистем, які забезпечують повний спектр функціональних можливостей для управління логістичними процесами [9,10]. Основні компоненти програмної структури ТС:

1. Модуль управління транспортом:
 - Планування маршрутів: Оптимізація маршрутів для зменшення витрат на транспортування.
 - Відстеження транспорту: Моніторинг руху транспортних засобів у режимі реального часу.
 - Управління автопарком: Контроль за станом транспортних засобів, їх обслуговуванням та ремонтом.
2. Модуль управління складом:

- Управління запасами: Контроль за запасами на складах, їх переміщенням та розміщенням.
 - Прийом та відправлення вантажів: Автоматизація процесів прийому та відправлення товарів.
 - Підтримка складування: Оптимізація використання складських площ.
3. Модуль управління замовленнями:
- Обробка замовлень: Прийом, обробка та відстеження замовлень від клієнтів.
 - Інтеграція з системами клієнтів: Обмін даними з системами замовників для забезпечення точності та своєчасності виконання замовлень.
4. Аналітичний модуль:
- Звітність та аналітика: Генерація звітів та аналіз даних для прийняття управлінських рішень.
 - Прогнозування попиту: Аналіз даних для прогнозування майбутнього попиту на логістичні послуги.
5. Модуль управління ланцюгом поставок:
- Планування та координація: Управління всіма етапами ланцюга поставок, від закупівель до доставки кінцевому споживачу.
 - Інтеграція постачальників: Спільне планування та управління запасами з постачальниками.
6. Інтерфейси користувача та мобільні додатки:
- Клієнтський портал: Надання клієнтам доступу до інформації про стан замовлень, відстеження вантажів тощо.
 - Мобільні додатки: Інструменти для водіїв, операторів складів та інших
- Всі ці умовні фактори є необхідними при побудові якісного аналізу схожої функціональності у вже існуючих програмних системах.

Система управління транспортом (Transport Management System, TMS) — це програмна система, що орієнтована на ефективне керування транспортними операціями та флотом. Основним завданням TMS є оптимізація процесів перевезення

вантажів або пасажирів, включаючи планування маршрутів, вибір транспортних засобів, моніторинг вантажів та управління водіями.

Інформаційна система розроблена у вигляді комплексної системи, яка включає в себе веб-застосунок, мобільні додатки, а також велику версію для стаціонарних систем [11]. Таким чином програмна платформа покриває всі можливі ринки попиту на своє рішення. На рис. 1.1 зображено головну сторінку інформаційної системи.

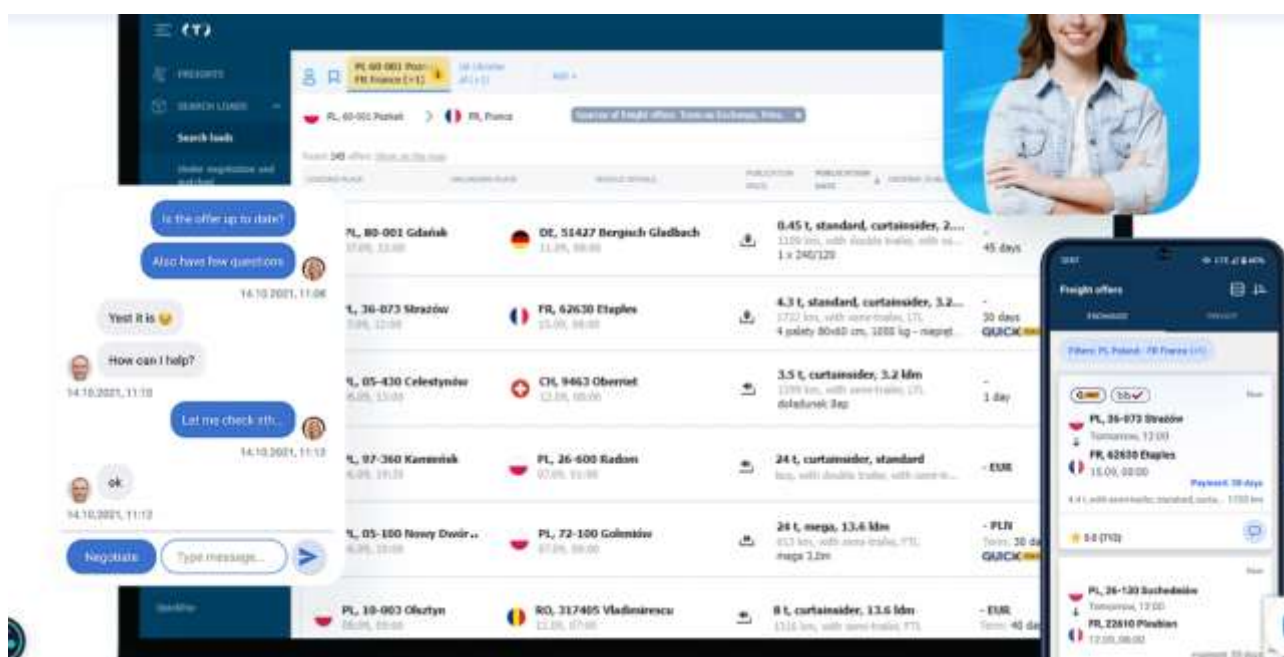


Рисунок 1.1 – Головна сторінка системи TMS

Інформаційна система TMS дуже масштабне та функціональне рішення, яке використовується на ринку транспортної логістики. Використання логістичного програмного забезпечення дуже полегшує складні процеси перевезення. Сучасні IT-рішення можуть інтегрувати різні аспекти, необхідні для надання послуг з перевезення. Після обробки заявки це програмне забезпечення автоматично впорядковує документацію та контролює оплату фрахту, забезпечуючи своєчасність оплати. Збір і аналіз даних допомагає оцінити прибутковість маршрутів та загальний дохід компанії. На рис. 1.2 зображено систему моніторингу маршруту.

До функціональних можливостей інформаційної системи відносяться:

- Маршрутизація та планування маршрутів. Розробка оптимальних маршрутів з урахуванням різних факторів, таких як відстань, час, витрати пального, трафік та обмеження на дорогах.
- Управління флотом. Відстеження розташування транспортних засобів, контроль за використанням палива, технічний стан та планове обслуговування.
- Планування та назначення. Назначення водіїв на транспортні засоби, відповідно до їх доступності, кваліфікації та робочого графіка.
- Моніторинг та звітність. Збір даних про виконання маршрутів, час доставки, стан вантажів та інші параметри для аналізу та звітності.

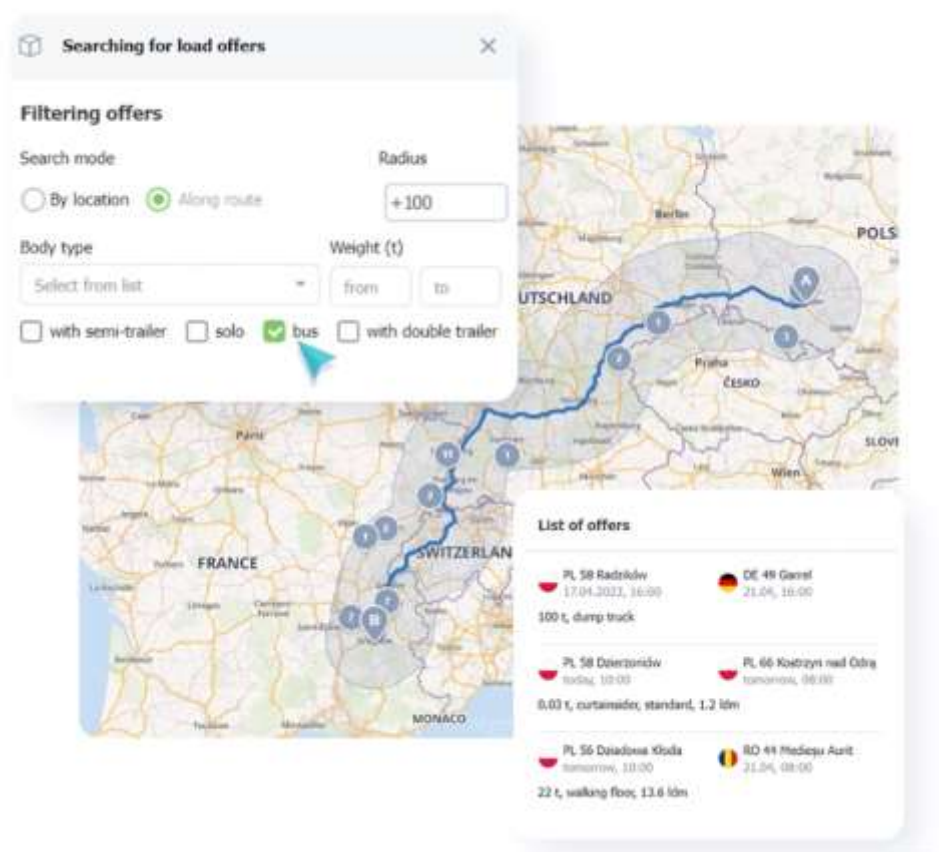


Рисунок 1.2 – Модуль системи моніторингу маршруту

Програмне забезпечення для транспортних компаній може легко інтегруватися з системами GPS і біржами вантажів. Автоматизація процесів забезпечує точність передачі важливої інформації одержувачам. Крім того, це зменшує кількість помилок,

які можуть виникнути під час ручного введення даних. Це також значно збільшує ефективність таких операцій, як виставлення вантажу на біржі, складання рахунків або спілкування з водіями.

Серед функціональних можливостей програмного засобу виділяють:

- Інтеграція з GPS та GIS: Для відстеження розташування транспортних засобів та розрахунку оптимальних маршрутів.
- Електронні логи водіїв: Для автоматизованого ведення логів часу руху та відпочинку водіїв відповідно до вимог законодавства.
- Інтеграція з системами електронного документообігу: Для автоматизованого обміну документами між учасниками логістичних процесів.
- Онлайн-замовлення та відстеження вантажів: Для спрощення процесів замовлення та відстеження вантажів для клієнтів.

Додатково до системи додано спеціалізованих констультантів у вигляді чат-ботів та допоміжних програмних систем, які дозволяють забезпечувати постійну підтримку як клієнтів так і партнерів. На рис. 1.3 зображено інтерфейсну частину робочої системи.

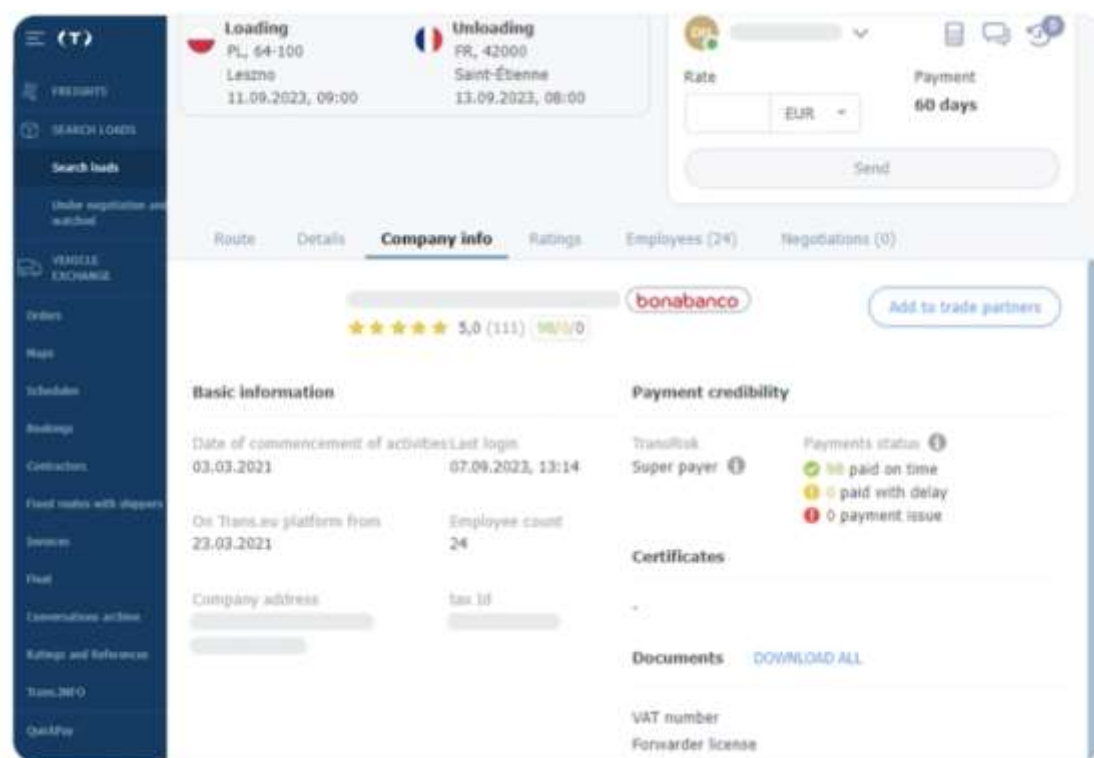


Рисунок 1.3 – Сторінка інтерфейсної частини з робочими можливостями

Розглянувши програмне рішення, було визначено його логіко-поведінкову структуру та особливості роботи програмного засобу. Серед визначених переваг інформаційної системи є наступні:

- Ефективність та оптимізація: Зменшення часу на планування та виконання маршрутів, оптимізація використання ресурсів.
- Підвищення продуктивності: Автоматизація багатьох рутинних завдань, що дозволяє спрощувати та прискорювати процеси.
- Покращення контролю: Забезпечення високого рівня контролю за всіма аспектами транспортних операцій.

Серед недоліків програмного засобу виділяються наступні:

- Високі витрати на впровадження та підтримку: Великі компанії можуть зіткнутися з високими витратами на впровадження та підтримку TMS.
- Необхідність навчання персоналу: Впровадження нової системи може вимагати навчання персоналу, що також може зайняти час та кошти.

– Залежність від технічної інфраструктури: Погана або непрацююча інфраструктура може призвести до проблем з функціонуванням системи.

Програмне забезпечення RemOnline спрощує управління доставкою та оптимізує логістику транспортної компанії. Завдяки вбудованій CRM, яка веде облік клієнтів та замовлень, ви зменшите збитки та збільшите прибуток.

RemOnline - це не лише інструмент для оптимізації процесів, але й зручний засіб для розвитку бізнесу. Серед важливих можливостей системи виділяють:

- Швидка обробка замовлень з урахуванням вільного транспорту.
- Збільшення прибутку та скорочення витрат.
- Ключові бізнес-показники в одному вікні.
- Облік клієнтів та контроль співробітників з будь-якого місця та пристрою.

Головна сторінка інформаційної сторінки зображена на рис. 1.4.

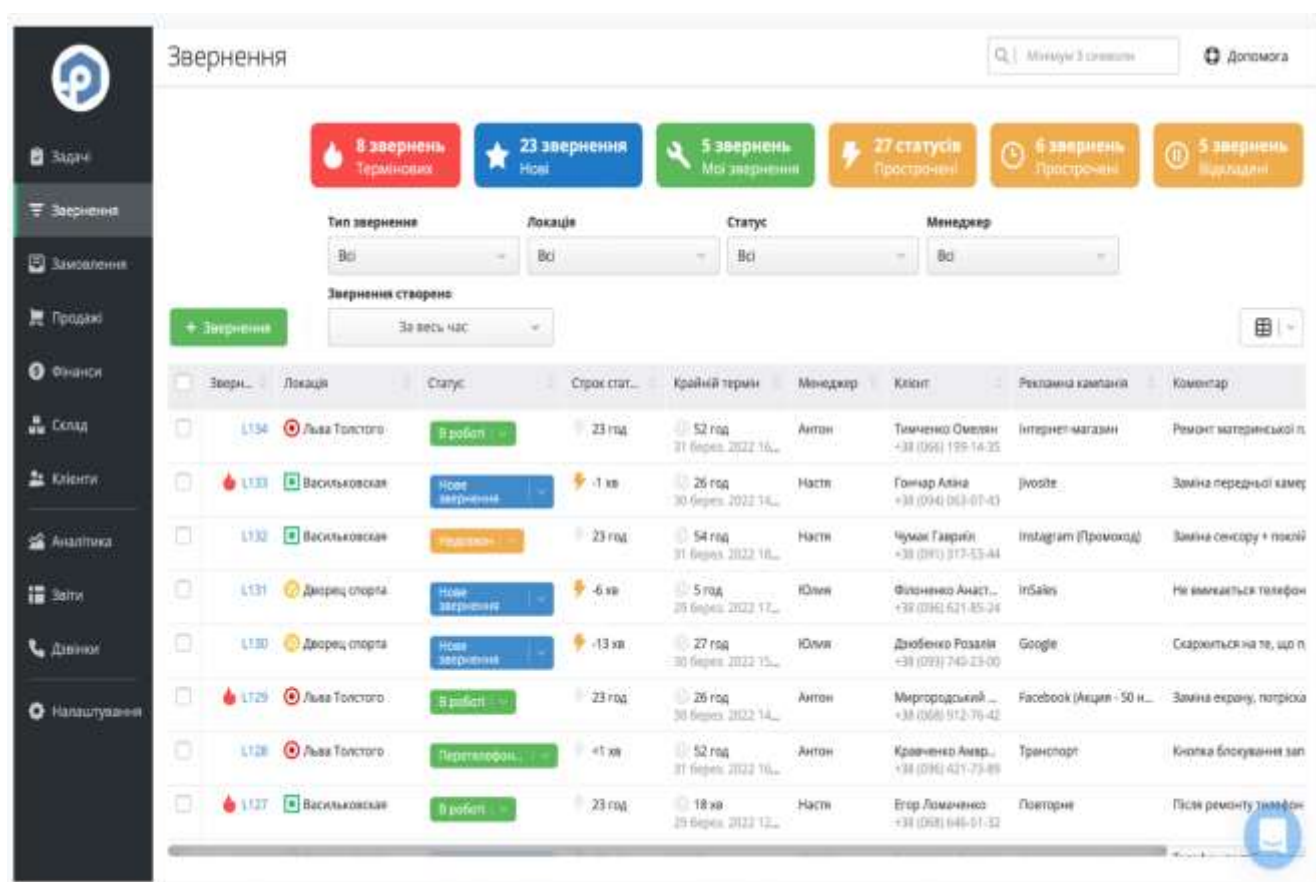


Рисунок 1.4 – Головна сторінка інформаційної системи

Однією з найголовніших програмних функцій інформаційної системи є планування замовлень та завантаження транспорту. Планувальник замовлень та завантаження транспорту має такі функціональні можливості:

- Оптимізація ресурсів за допомогою функції Планувальника.
- Оформлення замовлень з урахуванням завантаження транспорту та доступності водіїв.

- Планування завантаження співробітників на дні та тижні вперед.

CRM для перевезень:

- Допомога у плануванні завантаження співробітників та транспорту.
- Рівномірний розподіл замовлень між доступним транспортом.
- Запобігання простою транспорту в автопарку.

На рис. 1.5 зображено сторінку з планування графіку перевезень та логістики.

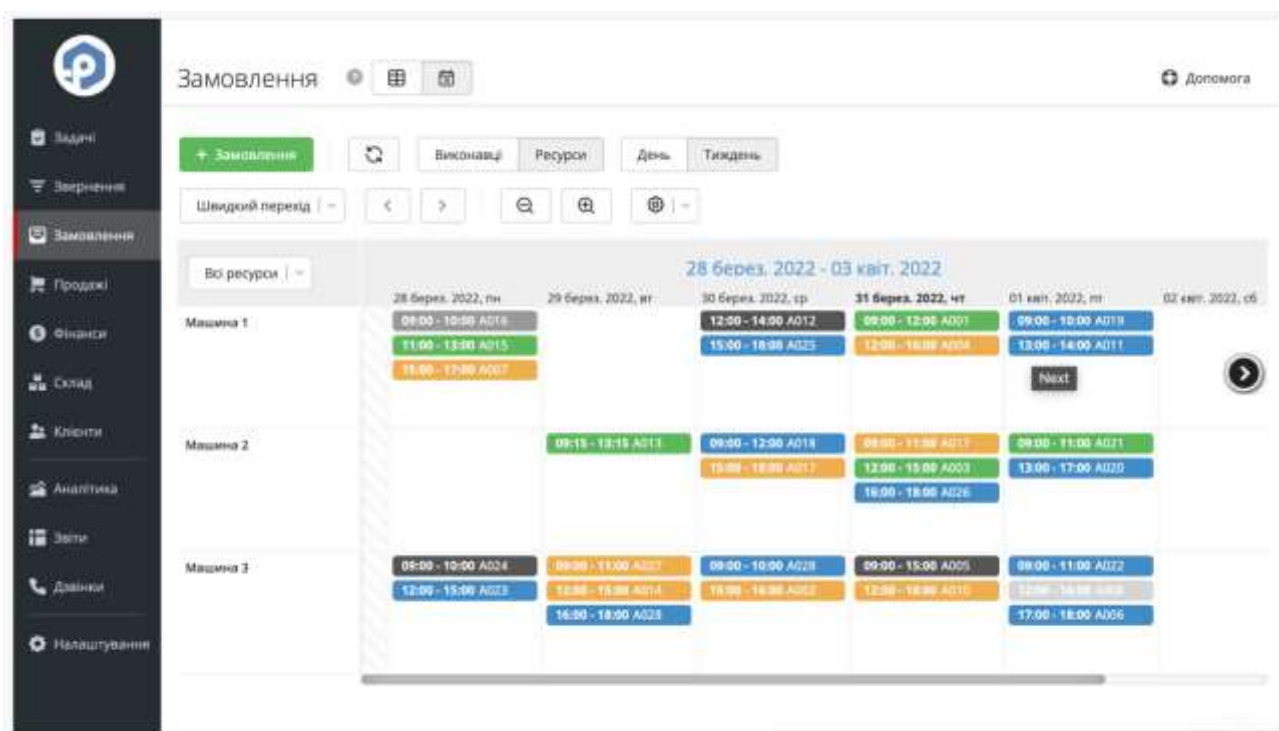


Рисунок 1.5 – Сторінка планувальника перевезень

Система обліку замовлень є дуже важливим компонентом інформаційної системи. Система обліку замовлень в транспортно-логістичній системі включає

кілька ключових компонентів та етапів, що забезпечують ефективне управління логістичними процесами. Серед основних елементів такої системи є:

1. Реєстрація замовлення:

- Клієнтська заявка. Клієнт подає заявку на перевезення через веб-платформу, мобільний додаток або контактний центр.
- Форма замовлення. Вказуються деталі замовлення, такі як тип вантажу, об'єм, вага, пункт відправлення та призначення, дата та час доставки.

2. Перевірка та підтвердження:

- Перевірка наявності ресурсів. Система перевіряє доступність транспорту та водіїв для виконання замовлення у вказаний час.
- Підтвердження замовлення. Після перевірки клієнту надсилається підтвердження про прийняття замовлення.

3. Планування та розподіл:

- Планувальник завантаження. Система автоматично планує завантаження транспорту та водіїв, щоб оптимально розподілити ресурси.
- Розподіл замовлень. Замовлення рівномірно розподіляються між доступними транспортними засобами, щоб уникнути їх простою.

4. Моніторинг та контроль:

- Трекінг транспорту. Система відстежує рух транспорту в режимі реального часу, надаючи інформацію про місцезнаходження, статус доставки та прогнозований час прибуття.
- Контроль виконання. Водії повідомляють про виконання кожного етапу доставки через мобільний додаток, а система оновлює статус замовлення.

5. Управління документацією:

- Електронні документи. Усі документи, такі як товарно-транспортні накладні, рахунки та акти виконаних робіт, ведуться в електронному вигляді.
- Архівування. Система автоматично зберігає документи в архіві для подальшого доступу та звітності.

6. Звітність та аналітика:

- Звіти по замовленнях. Система генерує звіти про виконані замовлення, завантаження транспорту, ефективність роботи водіїв та інші ключові показники.
 - Аналітика. Дані аналізуються для виявлення трендів, оптимізації процесів та покращення обслуговування клієнтів.
7. Інтеграція з іншими системами:
- Інтеграція з CRM. Обмін даними між CRM-системою для перевезень та іншими модулями для більш ефективного управління клієнтами та замовленнями.
 - Інтеграція з бухгалтерією. Автоматизація фінансових операцій та облік витрат і доходів.

На рис. 1.6 зображено сторінку з формування персоналізованих графіків та замовлень від клієнтів.

Замовлен...	Кінцевий термін	Статус	Клієнт	Адреса клієнта	Виконавець	Рекламна камп...
A028	4 д. 04 квіт. 2022 11:27	Нові	Паній Анна +38 (096) 483-45-45	07800, м. Вишневе, вул. Прогід Салерно-	Юлія	Соціальні мережі
A027	4 д. 04 квіт. 2022 11:27	Не укладено	Емельяненко Сара +38 (096) 289-35-75	08643, м. Костопіль, вул. Верестська, будинок	Лесь Ніколаєнко	Інтернет
A026	4 д. 04 квіт. 2022 11:26	Замовлено	Романенко Орхел +38 (050) 950-77-66	07340, м. Сміла, вул. Печерська, будинок 123,	Елісавета Колесніченко	Знайомі
A025	— 04 квіт. 2022 11:24	Завершено	Полученко Надія +38 (050) 774-53-71	08010, м. Гадяч, вул. Прорізна, будинок 176,	Лесь Ніколаєнко	Знайомі
A024	— 04 квіт. 2022 11:16	Готується	Мельниченко Душан +38 (050) 712-21-18	07205, м. Херсон, вул. Тулуз, будинок 47,	Богдана Рибак	Знайомі
A023	4 д. 04 квіт. 2022 11:15	Нові	Коваленко Магда +38 (067) 701-75-99	09156, м. Зупресс, вул. Биківський, будинок	Маруся Коломійца	Знайомі
A022	— 04 квіт. 2022 11:14	Прийнято з відхиленнями	Скороход Пламен +38 (066) 833-74-78	09413, м. Острогож, вул. Михайлівська, будинок	Елісавета Колесніченко	Соціальні мережі
A021	4 д. 04 квіт. 2022 11:13	Прийнято	Шинярук Роман +38 (095) 646-72-18	08162, м. Дніпро, вул. Франка, будинок 185,	Лесь Ніколаєнко	Соціальні мережі
A020	4 д. 04 квіт. 2022 11:12	Нові	Філі Богуслава +38 (063) 655-43-83	07814, м. Почаїв, вул. Воляків, будинок 12,	Елісавета Колесніченко	Інтернет
A019	4 д. 04 квіт. 2022 11:10	Прийнято	Дімба Гліб +38 (096) 906-59-75	07206, м. Чоп, вул. Злагодувська, будинок	Богдана Рибак	Соціальні мережі

Рисунок 1.6 – Сторінка формування замовлення від клієнта

В цілому інформаційна система відповідає всім сучасним вимогам та стандартам стосовно програмних засобів в сфері транспортної логістики. Однак, можна виділити як переваги так і недоліки такої системи. До переваг інформаційної системи відноситься:

- Ефективне планування та оптимізація маршрутів. Завдяки інтелектуальним алгоритмам програмне забезпечення допомагає зменшити час, витрачений на планування маршрутів, та оптимізує використання ресурсів компанії.

- Підвищення продуктивності. Автоматизація багатьох рутинних завдань дозволяє працівникам компанії сконцентруватися на більш важливих аспектах своєї роботи, що призводить до підвищення продуктивності та збільшення швидкості виконання завдань.

- Покращений контроль. RemOnline забезпечує високий рівень контролю за всіма аспектами транспортних операцій, що дозволяє керівництву та менеджерам моніторити та аналізувати процеси для прийняття вірних рішень.

До основних недоліків інформаційної системи RemOnline слід віднести:

- Високі витрати на впровадження та підтримку: Для великих компаній можуть виникнути значні витрати на впровадження та подальшу підтримку TMS, що може бути фінансово вимогливим.

- Необхідність навчання персоналу: Впровадження нової системи може потребувати часу та зусиль для навчання персоналу її користуванню, що може призвести до тимчасового спаду продуктивності.

- Залежність від технічної інфраструктури: Неадекватна або несправна технічна інфраструктура може стати причиною проблем з функціонуванням системи, що може вплинути на її ефективність та надійність.

Виконавши аналіз програмних систем в сфері транспортної логістики можливо зробити висновок про актуальну галузь застосування інформаційних та цифрових технологій, які дозволяють суттєво підвищувати продуктивність таких систем.

1.3 Аналіз цифрових методів з контролю за ефективністю транспортної логістики

Сучасний етап глобалізації характеризується інтенсивною взаємодією національних економік, що проявляється у масштабному русі товарів, послуг,

капіталу, робочої сили, технологій та цілих підприємств через кордони. Така взаємозалежність зумовлює необхідність системного врахування впливу національної економічної політики однієї держави на економічні процеси в інших країнах. У цьому контексті особливого значення набуває комплексний аналіз ключових індикаторів конкурентоспроможності держав, а також визначення стратегічних пріоритетів розвитку логістики як інтегральної складової міжнародної торгівлі [12].

Умови глобалізації та поглиблення ринкових відносин актуалізують завдання повноцінної інтеграції економіки України до світової економічної та фінансової систем. Зростаюча роль торгівлі та транспорту в національній економіці вимагає впровадження комплексних, інтегрованих підходів до спрощення процедур міжнародної торгівлі, що відповідають рекомендаціям Світової організації торгівлі та Європейського Союзу [13,14]. Автоматизація торговельних процедур та супутніх інформаційних потоків перетворюється на критичний фактор конкурентоспроможності. Сучасна торгівля охоплює не лише фізичне переміщення товарів, а й паралельний обмін великими обсягами структурованої інформації, що супроводжує вантажопотоки на всіх етапах ланцюга поставок. Спрощення торговельних процедур передбачає системні реформи, спрямовані на оптимізацію, скорочення та цифровізацію всіх етапів міжнародного логістичного ланцюга.

Особливу роль у цьому процесі відіграє впровадження національного механізму «єдиного вікна» (ЄВ) для міжнародної торгівлі. Ця система забезпечує суттєве підвищення прозорості, прискорення торговельних операцій, зростання бюджетних надходжень, формування привабливого інвестиційного середовища та мінімізацію суб'єктивних факторів у митному та прикордонному контролі. Дематеріалізація процедур, електронний обмін даними між державними органами та бізнесом, а також інтеграція інформаційних систем різних відомств створюють передумови для переходу до повністю цифрового середовища міжнародної торгівлі. За даними 2025 року, функціонал «Єдиного вікна» суттєво розширено: запроваджено автоматизований API-доступ до митних декларацій, електронний обмін даними в

морських портах, QR-коди для митних платежів, чат-боти з елементами штучного інтелекту, а також модулі для подання заяв на статус уповноваженого економічного оператора (АЕО) [15,16].

Методологія оцінки ефективності логістичних та торговельно-транспортних систем включає кілька взаємопов'язаних блоків:

- Економічний блок (митна політика та регулювання): аналіз ефективності тарифних і нетарифних заходів, динаміки експорту та імпорту, рівня боротьби з контрабандою й фальсифікацією, захисту прав інтелектуальної власності на кордоні, а також обсягів митних надходжень до державного бюджету.

- Адміністративний блок (митні процедури, адміністрування та контроль): оцінка часу проходження митних формальностей, кількості необхідних документів, рівня автоматизації та якості надання митних послуг.

- Організаційний блок: аналіз системи митної безпеки, ризик-орієнтованого менеджменту, ефективності використання кадрового потенціалу та ієрархії управління.

Серед ключових системних викликів розвитку логістичного потенціалу України на сучасному етапі слід виокремити:

- хронічний дефіцит інвестицій у транспортну інфраструктуру (дороги, залізниці, порти, прикордонні переходи);

- недостатній розвиток трилатеральної (трибічної) логістики та мультимодальних перевезень;

- відсутність повноцінного системного інтегратора четвертого рівня;

- недосконалість митного, прикордонного та інших видів контролю на зовнішніх кордонах;

- брак національної системи статистичного моніторингу та аналізу ключових показників логістичного розвитку;

- низький ступінь інтеграції національної логістичної системи до логістичної мережі.

Сучасні системи аналізу транспортних перевезень та логістичних процесів в Україні дедалі частіше базуються на інтеграції великих даних (big data), геоінформаційних систем (GIS), GPS-моніторингу, математичного моделювання транспортних потоків, а також інструментів штучного інтелекту для прогнозування попиту, оптимізації маршрутів і управління ризиками. Такі системи дозволяють здійснювати реальний часовий контроль ланцюгів поставок, знижувати витрати на 15–30 % за рахунок раціоналізації маршрутів і завантаження транспорту, а також підвищувати стійкість логістики до зовнішніх шоків.

Пріоритетні напрями розвитку логістичного потенціалу України мають ґрунтуватися на:

- комплексній модернізації критичної транспортної інфраструктури (відновлення та розбудова портів Чорного моря й Дунаю, прикордонних переходів, автомобільних і залізничних коридорів);
- впровадженні передових транспортно-митних технологій (автоматизоване декларування, ризик-менеджмент, блокчейн для відстеження вантажів);
- розвитку сучасної складської нерухомості класу А та В з урахуванням енергоефективності;
- розширенні мультимодальних та інтермодальних перевезень;
- посиленні цифрової інтеграції з логістичними системами ЄС (TEN-T, коридорні ініціативи);
- формуванні національної платформи даних для аналізу транспортних потоків і логістичних показників.

Реалізація цих напрямів сприятиме не лише відновленню та підвищенню конкурентоспроможності української логістики в умовах післявоєнної реконструкції, а й забезпеченню стійкої інтеграції країни до європейського та глобального економічного простору.

1.4 Аналіз систем технічної реалізації інформаційної системи аналізу транспортних перевезень та логістичних процесів

Мова програмування PHP продовжує залишатися однією з найпоширеніших серверних технологій у сфері веб-розробки. Станом на поточний момент, PHP забезпечує функціонування приблизно 72 % усіх вебсайтів, для яких відомо серверне середовище виконання, що робить її беззаперечним лідером серед серверних мов [17]. Ця мова призначена насамперед для створення динамічних вебресурсів — від елементарних інформаційних сторінок до масштабних корпоративних систем, електронних торговельних майданчиків та платформ керування контентом.

PHP вирізняється вбудованими механізмами для ефективної обробки HTTP-запитів, включаючи підтримку методів GET/POST, завантаження файлів, управління заголовками та статусами відповідей. Мова забезпечує надійну обробку даних, отриманих через вебформи: валідацію, фільтрацію, санітизацію та подальше збереження в базах даних. Важливою можливістю є система керування сесіями та cookies, що дозволяє зберігати стан користувача між запитами, реалізовувати авторизацію, персоналізацію та підтримку кошиків покупок. Вбудована підтримка численних СУБД (MySQL, PostgreSQL, SQLite, Oracle тощо) полегшує інтеграцію з реляційними та нереляційними сховищами даних.

Однією з ключових архітектурних особливостей PHP є можливість вбудовування коду безпосередньо в HTML-шаблони (так звані PHP-теги). Під час обробки запиту сервер виконує PHP-скрипти, генеруючи готовий HTML, який передається клієнту. Такий підхід дозволяє створювати контент, що адаптується в реальному часі залежно від параметрів запиту, даних користувача, часу доби чи інших умов. Це забезпечує високий рівень персоналізації (наприклад, персональні привітання, рекомендації, динамічні блоки) та інтерактивності без необхідності повного перезавантаження сторінки (з використанням AJAX або сучасних технік) [18].

Екосистема PHP значно розвинута завдяки великій кількості фреймворків, які радикально прискорюють розробку та підвищують якість коду. Серед найвпливовіших:

- Laravel — сучасний, елегантний фреймворк з виразним синтаксисом, потужною ORM (Eloquent), вбудованою системою автентифікації, чергами, кешуванням та інструментами для тестування; домінує за популярністю серед професійних розробників.
- Symfony — високогнучкий компонентний фреймворк, що слугує основою для багатьох інших рішень; характеризується модульністю, підтримкою сучасних стандартів та широким набором готових бібліотек.
- CodeIgniter — легковаговий, швидкий фреймворк з мінімальним навчальним порогом, ідеальний для невеликих і середніх проєктів.
- Yii — швидкодіючий фреймворк з потужними інструментами генерації коду, підтримкою Gii та високою продуктивністю.
- Zend Framework / Laminas — комплексне рішення для enterprise-рівня, орієнтоване на масштабованість та надійність.

PHP підтримує розширення (extensions) на рівні C, а також величезну кількість сторонніх бібліотек через Composer — менеджер залежностей, що став де-факто стандартом. Це дозволяє інтегрувати функціональність для роботи з графікою, PDF, API (REST/GraphQL), блокчейном, машинним навчанням тощо.

Сфера застосування PHP охоплює створення блог-платформ, інтернет-магазинів, форумів, корпоративних порталів, систем управління контентом (WordPress, Joomla, Drupal — основні CMS світу), RESTful та GraphQL API, мікросервісів.

JavaScript, своєю чергою, є універсальною мовою програмування, що займає провідне місце в сучасній цифровій екосистемі. За даними численних опитувань (зокрема Stack Overflow 2025), JavaScript залишається найбільш використовуваною мовою серед розробників уже понад дванадцять років поспіль, з часткою використання близько 62–65 % серед професійних фахівців. Вона є основою

клієнтської частини вебзастосунків, а завдяки Node.js — також серверної, мобільної та навіть настільної розробки [19-21].

Основні риси JavaScript:

- Виконання в браузері забезпечує миттєву реакцію інтерфейсу без звернень до сервера (маніпуляція DOM, події, анімації).
- Динамічна слабка типізація та прототипне успадкування надають гнучкості, але вимагають дисципліни.
- Підтримка функціонального, імперативного та (з ES6+) класового програмування.
- Постійне оновлення стандарту ECMAScript (щорічні релізи) додає сучасні конструкції: `async/await`, деструктуризація, модулі, опціональний ланцюжок, Temporal API (у майбутніх версіях) тощо.

Екосистема JavaScript є однією з найбагатших:

- Фронтенд-фреймворки: React (компонентний підхід, віртуальний DOM), Angular (повноцінна платформа з двостороннім зв'язком даних), Vue.js (прогресивний, легкий у освоєнні).
- Бібліотеки: jQuery (історична, але все ще поширена), Lodash/Underscore (утиліти), Axios/Fetch (HTTP).
- Серверна частина: Node.js + Express/NestJS/Fastify.
- Збірщики: Webpack, Vite, esbuild.
- Пакетний менеджер: npm / Yarn / pnpm.

JavaScript застосовується для SPA, PWA, гібридних мобільних застосунків (React Native, Ionic), настільних програм (Electron), ігр, IoT, навіть машинного навчання (TensorFlow.js).

Обидві мови — PHP та JavaScript — доповнюють одна одну в сучасній веброзробці: PHP забезпечує надійну серверну логіку та генерацію контенту, тоді як JavaScript відповідає за інтерактивність і клієнтський досвід. Їх спільне використання (наприклад, Laravel + Inertia.js + Vue/React) залишається одним із найефективніших стеків для створення швидких, масштабованих і зручних вебзастосунків.

1.5 Аналіз систем збереження та керування даними для інформаційних систем аналізу транспортних перевезень та логістичних процесів

Firebase представляє собою комплексну хмарну платформу, розроблену компанією Google, що забезпечує інфраструктуру як послугу (Backend-as-a-Service, BaaS) для створення, масштабування та підтримки мобільних і вебзастосунків. Платформа об'єднує набір взаємопов'язаних сервісів, які охоплюють ключові аспекти розробки програмного забезпечення: зберігання та синхронізацію даних, ідентифікацію користувачів, серверну логіку, файлове сховище, моніторинг продуктивності, аналітику поведінки користувачів та інструменти для розгортання [22].

Центральними компонентами Firebase є дві системи керування даними:

- Cloud Firestore — сучасна документно-орієнтована NoSQL-база даних, що характеризується високою масштабованістю, підтримкою складних запитів (включаючи складені фільтри, сортування, агрегацію), глобальною реплікацією та вбудованою офлайн-підтримкою. Вона рекомендована як основне рішення для більшості нових проєктів завдяки гнучкості моделі даних та оптимізації під високе навантаження.
- Realtime Database — класична JSON-орієнтована NoSQL-база, що забезпечує миттєву синхронізацію даних між клієнтами через WebSocket-подібний механізм. Вона зберігає актуальність для застосунків, де критична низька затримка та проста структура даних.

Система Firebase Authentication надає стандартизовані механізми автентифікації, підтримуючи електронну пошту/пароль, федеративні провайдери (Google, Apple, Facebook, Twitter/X, GitHub тощо), анонімний вхід, телефонну верифікацію та кастомні токени. Інтеграція з іншими сервісами платформи забезпечує безшовне керування доступом на рівні безпеки.

Cloud Storage слугує масштабованим об'єктним сховищем для бінарних даних (зображення, відео, документи), з автоматичним керуванням версіями, контролем доступу на основі правил безпеки Firebase та тісною інтеграцією з автентифікацією.

Cloud Functions реалізують модель serverless-обчислень: функції на Node.js (або інших runtime) запускаються у відповідь на події (зміни в базах даних, завантаження файлів, автентифікаційні події, HTTP-запити, Pub/Sub тощо), дозволяючи виконувати бізнес-логіку без управління серверами [23,24].

Firebase Analytics забезпечує детальний збір телеметрії про взаємодію користувачів, події, аудиторії та конверсії, з можливістю експорту даних до BigQuery для поглибленого аналізу.

Платформа також включає інструменти для тестування (A/B-тестування, Remote Config для динамічного налаштування), моніторингу продуктивності (Performance Monitoring), краш-репортигу (Crashlytics), розповсюдження додатків (App Distribution) та захисту (App Check для запобігання зловживанням).

Firebase демонструє високу ступінь сумісності з різними стеками розробки:

- Для кросплатформної мобільної розробки ідеально поєднується з Flutter завдяки набору офіційних плагінів FlutterFire, що забезпечують нативну інтеграцію для Android, iOS та веб.
- Вебзастосунки реалізуються через JavaScript/TypeScript SDK, з підтримкою сучасних фреймворків (React, Angular, Vue тощо).
- Нативні мобільні проєкти використовують SDK для Android (Java/Kotlin) та iOS (Swift/Objective-C).

Переваги платформи полягають у мінімальному порозі входу, автоматичному масштабуванні, моделі оплати за фактичне використання (з щедрим безкоштовним рівнем для стартапів та прототипів), єдиній консолі керування, детальній документації та активній екосистемі розширень. Це дозволяє командам зосередитися на бізнес-логіці та користувацькому досвіді, уникаючи витрат часу на інфраструктуру, розгортання серверів та підтримку бекенду [25,26].

Загалом, Firebase позиціонується як універсальне, високопродуктивне середовище для прискореної розробки надійних, масштабованих застосунків, що охоплюють повний життєвий цикл продукту — від прототипування до глобального розгортання та зростання.

1.6 Мета кваліфікаційної роботи

Метою кваліфікаційної роботи є розробка інформаційної аналітичної системи для підвищення ефективності транспортно-логістичних операцій та оптимізації методів обробки запитів на перевезення. Головною функціональною логікою є комплексна підтримка процесів управління перевезеннями, що включає моніторинг маршрутів, аналіз вантажопотоків, оцінку ефективності використання транспортних засобів, а також прогнозування навантаження та планування ресурсів. Така система є платформою для підвищення якості логістичних рішень, оптимізації витрат і вдосконалення транспортних процесів.

Завданнями роботи є:

- аналіз предметної області та сучасних підходів до використання інформаційних технологій у сфері транспортної логістики та обробки запитів на перевезення;
- аналіз існуючих програмних рішень для управління транспортними та логістичними процесами та визначення їх функціональних можливостей;
- аналіз програмних засобів організації перевезень з метою визначення основних вимог, функцій і сценаріїв використання системи;
- дослідити сучасні технології розробки, мови програмування та системи керування даними для вибору оптимального технологічного стеку;
- розробити концепцію інформаційної системи управління транспортно-логістичними процесами;
- розробити архітектурну модель системи, включаючи UML-діаграми та логіку взаємодії компонентів;

- розробити функціональне програмне рішення із застосуванням сучасних веб-технологій;
- розробити та описати функціональні можливості системи, зокрема модулі бронювання, управління перевезеннями та оренди транспорту;
- провести тестування розробленої системи та оцінити її ефективність у практичному застосуванні.

1.7 Висновки до першого розділу

Аналіз сучасного стану логістичного сектору України свідчить про наявність низки системних викликів, пов'язаних із недостатньою розвиненістю інфраструктури, обмеженою інтеграцією в європейські логістичні мережі, дефіцитом ефективних механізмів координації транспортних потоків, а також недостатнім рівнем цифровізації та автоматизації процесів. Ці фактори суттєво знижують конкурентоспроможність національної транспортної системи, збільшують операційні витрати та ускладнюють адаптацію до зовнішніх геополітичних і економічних змін.

У цьому контексті впровадження комплексної інформаційної системи надання транспортних послуг набуває стратегічного значення як інструменту подолання виявлених обмежень. Така система забезпечує:

- систематичний збір, інтеграцію та аналітичну обробку даних щодо стану транспортної інфраструктури, що створює об'єктивну основу для обґрунтованого планування інвестицій, реконструкції та модернізації ключових об'єктів;
- централізовану координацію логістичних операцій, включаючи реальний час відстеження вантажопотоків, динамічну оптимізацію маршрутів та балансування навантаження на транспортні коридори;
- виявлення та кількісну оцінку неефективних процесів і ресурсів, що дозволяє впроваджувати цільові заходи з підвищення продуктивності, скорочення витрат та мінімізації втрат;

– посилення контролю за дотриманням нормативних вимог, митних і прикордонних процедур, а також управління ризиками на всіх етапах ланцюга поставок, що сприяє підвищенню надійності та прозорості транспортних послуг.

Таким чином, створення та впровадження спеціалізованої інформаційної системи для надання транспортних послуг виступає необхідною умовою для подолання структурних бар'єрів розвитку логістики в Україні. Вона сприятиме переходу до цифрової, інтегрованої та адаптивної моделі транспортного забезпечення, що забезпечить стійке зростання сектору, посилення його інтеграції в глобальні ланцюги створення вартості та підвищення загальної конкурентоспроможності національної економіки.

2 РОЗРОБКА ЛОГІКИ АНАЛІЗУ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ПЕРЕВЕЗЕНЬ ДЛЯ ЛОГІСТИЧНИХ КОМПАНІЙ

2.1 Розробка системи аналізу ефективності транспортних перевезень

Система аналізу транспортних перевезень та логістичних процесів є спеціалізованою інформаційною системою (ІС), призначеною для комплексного збору, обробки, моделювання та аналітичної оцінки даних, що характеризують транспортні потоки, логістичні операції та пов'язані з ними ресурси.

Вона забезпечує науково обґрунтоване прийняття рішень щодо оптимізації перевезень, підвищення ефективності використання транспортних засобів, мінімізації витрат і посилення надійності ланцюгів поставок.

Застосування сучасних методів та алгоритмів у сфері транспортних перевезень сприяє досягненню кількох ключових цілей:

- оптимізації маршрутів руху транспортних засобів, що забезпечує зниження витрат на паливо, зменшення зносу рухомого складу та скорочення експлуатаційних витрат;
- раціонального використання робочого часу водіїв і екіпажів, скорочення простоїв, гарантування своєчасності доставки, збереження якості вантажів (особливо швидкопсувних) та підвищення рівня задоволеності клієнтів;
- прогнозування попиту на транспортні послуги та планування необхідних ресурсів, що дозволяє уникати перевантаження складських потужностей і оптимізувати управління запасами;
- моніторингу транспортних засобів у режимі реального часу, що підвищує контроль за виконанням маршрутів, оперативне реагування на відхилення (затори, аварії, погодні умови) та загальну безпеку й надійність операцій.

Такий комплексний підхід перетворює логістику на ключовий фактор конкурентоспроможності підприємства, сприяючи зниженню витрат і покращенню якості послуг.

Аналіз інформаційних потоків у процесах обробки вантажів виявляє інтенсивний обмін різнорідними даними на всіх етапах: всередині корпоративного середовища, між інформаційними системами контрагентів та між головним офісом і підрозділами. Реінжиніринг цих потоків передбачає впровадження ефективних алгоритмів управління та інтеграцію наявних ІС.

Імітаційне моделювання виступає ефективним методом для створення незалежних від конкретних технологій, програмних засобів чи апаратних платформ моделей інформаційного супроводу учасників перевезень. Це відкриває шлях до розробки інтелектуальних систем, здатних формувати віртуальні транспортні підприємства з можливістю інтеграції в інтернет-середовище.

Сучасна система аналізу транспортних перевезень базується на інтеграції передових технологій:

- Великі дані (Big Data) та аналітика — для прогнозування попиту, виявлення оптимальних маршрутів, розподілу ресурсів і скорочення витрат;
- Штучний інтелект та машинне навчання — для автоматизації планування, динамічної оптимізації маршрутів і розкладів, мінімізації простоїв;
- Інтернет речей (IoT) — для моніторингу стану транспортних засобів, вантажів і умов перевезення в реальному часі;
- Хмарні технології — для масштабованої обробки та зберігання даних, забезпечення доступу учасників ланцюга в режимі реального часу;
- Клієнт-серверна архітектура — для швидкого обміну даними між підсистемами з мінімальним навантаженням на мережу;
- Засоби інформаційної безпеки — для захисту від несанкціонованого доступу та кіберзагроз, гарантування конфіденційності й цілісності інформації.

Додатково система передбачає автоматизацію оновлення даних, розширення функціоналу (фінансовий облік, кадрове планування, звітність, аналітика), інтуїтивний інтерфейс, мобільний доступ та екологічну оптимізацію.

Комплексна ІС реалізується як локальна обчислювальна мережа, що об'єднує робочі місця користувачів. Основні підсистеми включають:

- фінансовий і бухгалтерський облік;
- кадровий облік;
- облік вантажів і їх обробки;
- оперативне диспетчерське управління;
- облік роботи транспортних засобів і механізмів;
- архівне зберігання та обробка документів;
- автоматичний обмін даними та інтеграція інформації;
- системи відеоспостереження;
- управління спеціалізованими виробничими процесами;
- інформаційно-довідкові модулі;
- моніторинг технічного стану рухомого складу та ремонтів.

Використання сучасних СКБД забезпечує швидке опрацювання великих обсягів даних, автоматизацію звітності та прискорення управлінських рішень. Клієнт-серверна модель виключає зайву передачу даних по мережі для розрахунків, автоматичний ввід/вивід інформації дозволяє прогнозувати розташування техніки, планувати накопичення вантажів, скорочувати простой та мінімізувати ризики.

Для координації в транспортному вузлі організовуються:

- внутрішньовиробничий та оперативний телефонний зв'язок;
- технологічний гучномовний зв'язок;
- локальна комп'ютерна мережа;
- системи технологічного та охоронного відеоспостереження.

Апаратно-програмний комплекс включає контролери, модулі вводу/виводу, станції розподіленого вводу/виводу, радіомодеми, Industrial Ethernet, бездротові канали IEEE 802.11 для підключення рухомих об'єктів (наприклад, навантажувальних машин), рис. 2.1.

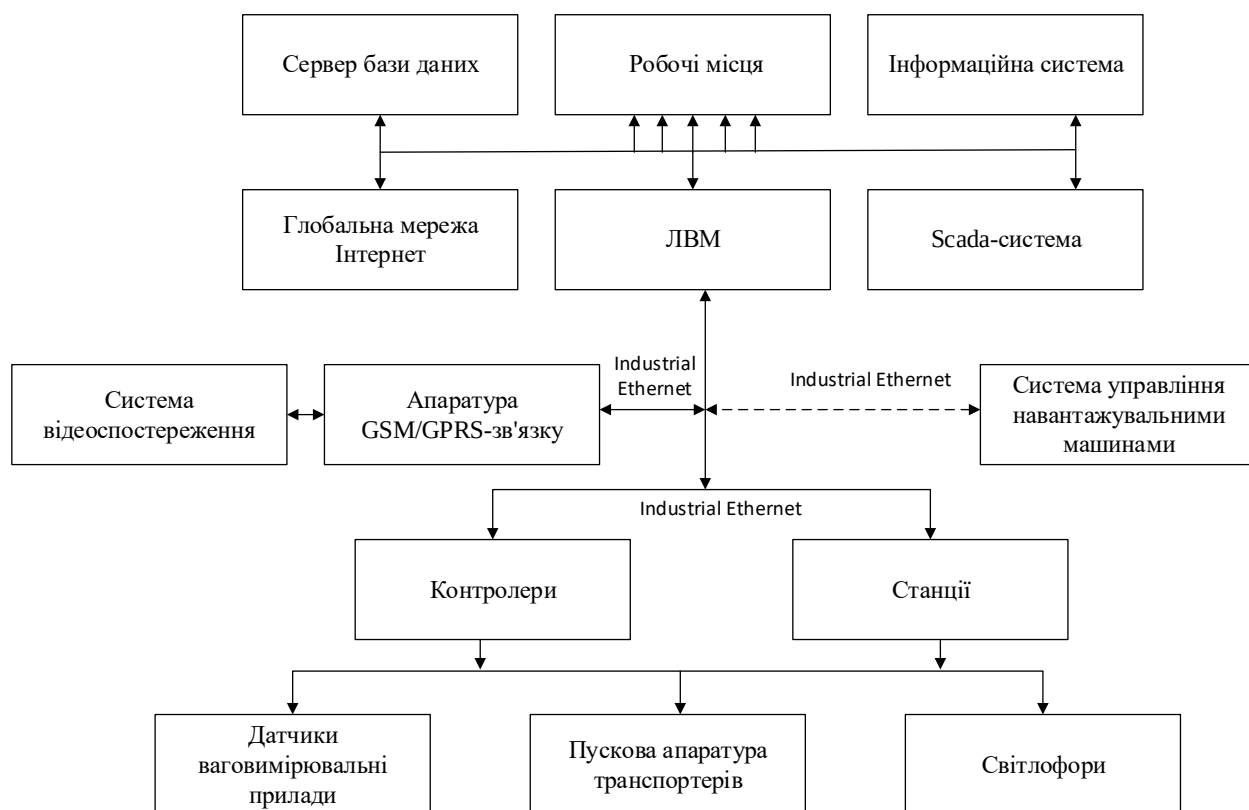


Рисунок 2.1 - Апаратно-програмні засоби координуючої інформаційної системи

Таким чином, система аналізу транспортних перевезень та логістичних процесів є інтегрованою цифровою платформою, що поєднує аналітику, моделювання, автоматизацію та реальний час моніторингу. Її впровадження забезпечує перехід до інтелектуальної логістики, суттєве підвищення ефективності, зниження витрат і адаптивність до сучасних вимог глобальних ланцюгів поставок.

2.2 Розробка логіко-математичної моделі аналізу логістичних процесів

Сучасні системи управління вантажоперевезеннями функціонують у складному середовищі, що характеризується високою динамікою транспортних потоків, багатофакторним впливом зовнішніх умов та необхідністю оперативного прийняття рішень. Логістичні процеси включають планування маршрутів, контроль перевезень, управління ресурсами та аналіз ефективності доставки. Зростання обсягів перевезень та вимог до швидкості доставки потребує застосування інтелектуальних

інформаційних систем, здатних аналізувати великі обсяги даних та формувати оптимізаційні рішення.

Логіко-математичне моделювання дозволяє формалізувати транспортні процеси, представити їх у вигляді математичних залежностей та забезпечити можливість алгоритмічної обробки. Основною метою моделі є кількісна оцінка ефективності перевезень, виявлення неефективних маршрутів та підтримка оптимізаційних рішень.

Транспортно-логістична система розглядається як сукупність взаємодіючих об'єктів та процесів, що формують мережу перевезень. Формалізація системи починається з визначення множин її елементів:

$$V = \{v_1, v_2, \dots, v_n\}, \quad (2.1)$$

де V — множина транспортних засобів.

$$R = \{r_1, r_2, r_3, \dots, r_m\}, \quad (2.2)$$

де R — множина маршрутів перевезень.

$$P = \{p_1, p_2, \dots, p_k\}, \quad (2.3)$$

де P — множина логістичних пунктів (склади, термінали, точки доставки).

$$C = \{c_1, c_2, \dots, c_s\}, \quad (2.4)$$

де C — множина вантажів.

$$T = \{t_1, t_2, \dots, t_s\}, \quad (2.5)$$

де T — часові інтервали виконання перевезень.

Такий підхід дозволяє розглядати логістичну систему як дискретну структуру, що складається з елементів, взаємозв'язки між якими можуть бути формалізовані та піддані подальшому аналізу.

Кожний транспортний засіб характеризується набором параметрів:

$$v_i = (cap_i, fuel_i, speed_i, cost_i, status_i), \quad (2.6)$$

де:

- cap_i — вантажопідйомність;
- $fuel_i$ — витрата палива;
- $speed_i$ — середня експлуатаційна швидкість;
- $cost_i$ — експлуатаційні витрати на одиницю часу або відстані;
- $status_i$ — технічний та операційний стан.

Дана параметризація дозволяє враховувати ресурсні можливості транспорту та його експлуатаційну ефективність під час аналізу перевезень.

Транспортна мережа формалізується у вигляді зваженого графа:

$$G = (P, E), \quad (2.7)$$

де P — множина вузлів (логістичних пунктів), а E — множина ребер, що представляють транспортні маршрути між ними.

Кожне ребро має ваговий вектор:

$$w_{ij} = (d_{ij}, t_{ij}, c_{ij}, r_{ij}), \quad (2.8)$$

де d_{ij} — відстань, t_{ij} — час проходження, c_{ij} — вартість перевезення, r_{ij} — рівень ризику.

Таке представлення дозволяє використовувати методи теорії графів для аналізу маршрутів та пошуку оптимальних шляхів перевезення. Час доставки є одним із ключових показників ефективності перевезення та визначається як сумарний час проходження маршруту з урахуванням можливих затримок:

$$T_d = \sum_{e \in r_j} \frac{d_e}{speed_i} + delay_e, \quad (2.9)$$

Перший доданок відображає теоретичний час руху транспортного засобу на основі довжини ділянок маршруту та середньої швидкості. Другий компонент враховує стохастичні затримки, спричинені заторами, погодними умовами, митними процедурами або технологічними зупинками. Таким чином, модель часу доставки поєднує детерміновані та випадкові фактори, що дозволяє отримати більш реалістичну оцінку.

Загальні витрати на перевезення формуються як сукупність експлуатаційних та операційних витрат:

$$C_t = (fuel_i * d_j * P_f) + C_{maint} + C_{labor} + C_{toll}, \quad (2.10)$$

Перша складова описує витрати на паливо, які прямо пропорційні відстані та нормі споживання. Інші компоненти включають технічне обслуговування транспорту, оплату праці водіїв та витрати на платні дороги. Така декомпозиція дозволяє детально аналізувати структуру витрат і виявляти фактори їх зростання.

Ефективність використання транспортного засобу визначається коефіцієнтом завантаження:

$$U_v = \frac{w_{cargo}}{cap_i}, \quad (2.11)$$

Цей показник характеризує ступінь використання вантажного простору. Значення, близьке до одиниці, свідчить про оптимальне використання транспортного ресурсу, тоді як низькі значення вказують на неефективність перевезення через недовантаження.

Ризик перевезення формується під впливом декількох факторів:

$$R_s = \alpha R_{road} + \beta R_{crime} + \gamma R_{weather}, \quad (2.12)$$

де коефіцієнти α , β , γ визначають вагомість відповідних факторів.

Дана модель дозволяє інтегрувати різномірні ризики в єдиний показник. Наприклад, поганий стан доріг збільшує ймовірність пошкодження вантажу, криміногенні фактори підвищують ризик втрат, а погодні умови впливають на безпеку перевезення.

Якість доставки характеризується дотриманням встановлених термінів:

$$Q_d = 1 - \frac{delay_{actual}}{delay_{max}}, \quad (2.13)$$

Цей показник відображає ступінь відповідності фактичного часу доставки нормативним значенням. Якщо затримка відсутня, індекс дорівнює одиниці, що відповідає максимально якісному виконанню перевезення.

Оскільки показники мають різні одиниці виміру, використовується нормалізація:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}, \quad (2.14)$$

Нормалізація дозволяє привести всі показники до безрозмірної шкали, що забезпечує можливість їх інтеграції в єдину функцію ефективності. Для комплексної оцінки використовується багатокритеріальна функція:

$$E_{total} = w_1 T'_d + w_2 C'_d + w_3 (1 - U'_v) + w_4 R'_4 + w_5 (1 - Q'_d), \quad (2.15)$$

Дана функція об'єднує часові, економічні, ресурсні та ризикові показники. Використання вагових коефіцієнтів дозволяє адаптувати модель до специфіки підприємства. Мінімізація інтегрального показника відповідає максимальній ефективності перевезення.

Для контролю відхилень використовується статистичний показник:

$$Z = \frac{X - \mu}{\sigma}, \quad (2.16)$$

Ця формула визначає стандартизоване відхилення показника від середнього значення. Якщо $|Z| > 3$, спостерігається аномалія, що може свідчити про несанкціоновані зупинки, перевитрату палива або відхилення від маршруту.

Для прогнозування параметрів використовується регресійна модель:

$$Y = \beta_0 + \beta_1 d + \beta_2 traffic + \beta_3 weather + \beta_4 load, \quad (2.17)$$

Модель дозволяє оцінити вплив ключових факторів на час або витрати перевезення. Коефіцієнти регресії визначають вагу кожного фактора та дозволяють прогнозувати ефективність майбутніх перевезень.

Оптимізаційна задача формулюється як мінімізація функції витрат, часу та ризику: $\min(C_t + \lambda T_d + \theta R_s)$, за умов обмежень $cap_i \geq w_{cargo}, T_d \leq T_{max}, U_v \geq U_{min}$.

Це дозволяє забезпечити вибір маршруту, що відповідає ресурсним, часовим та експлуатаційним обмеженням.

Запропонована логіко-математична модель забезпечує формалізований підхід до аналізу логістичних процесів та дозволяє інтегрувати дані з різних джерел у єдину систему оцінювання. Використання багатокритеріального аналізу, статистичних

методів та оптимізаційних алгоритмів створює основу для побудови інтелектуальної системи управління транспортними перевезеннями.

Практичне застосування моделі сприяє зниженню витрат, підвищенню точності доставки, раціональному використанню транспортних ресурсів та мінімізації логістичних ризиків.

2.3 Розробка системи прийняття рішень та оцінки ефективності вантажних перевезень

У сучасних транспортно-логістичних системах ефективність управління перевезеннями визначається не лише можливістю збору великих обсягів даних, але й здатністю системи трансформувати ці дані у знання, придатні для прийняття управлінських рішень. Аналітичний блок інформаційної системи виконує функцію інтелектуального ядра, що забезпечує обробку даних, їх інтерпретацію, виявлення закономірностей та формування рекомендацій. Основною метою аналітичного блоку є перетворення потоків сирих даних, отриманих з різних джерел, у структуровану інформацію про ефективність перевезень, ризики, відхилення та потенційні напрями оптимізації. Такий підхід дозволяє перейти від реактивного управління до проактивного, орієнтованого на прогнозування та запобігання неефективним логістичним операціям.

Аналітичний блок функціонує як багаторівнева система, що включає етапи інтеграції даних, їх попередньої обробки, аналітичної оцінки, виявлення відхилень, інтерпретації результатів та формування рекомендацій. Ефективність аналітичного процесу безпосередньо залежить від повноти, достовірності та актуальності даних. У системі аналізу вантажоперевезень використовується багатоджерельна модель даних, що включає телематичні, операційні, зовнішні та аналітичні джерела, рис. 2.2.

До телематичних джерел належать GPS-трекери, датчики витрати палива, датчики навантаження та системи моніторингу стану транспортних засобів. Вони забезпечують отримання даних у реальному часі щодо переміщення транспорту,

швидкості руху, зупинок та технічних параметрів. Операційні джерела включають інформаційні системи підприємства, такі як системи управління транспортом, складські системи та ERP-платформи. Вони надають дані про замовлення, графіки перевезень, характеристики вантажів та використання ресурсів.



Рисунок 2.2 – Блок-схема процесів функціонування аналітичної логіки в системі вантажних перевезень

Зовнішні джерела даних включають інформацію про погодні умови, стан дорожньої інфраструктури, транспортні затори та геополітичні ризики. Інтеграція цих даних дозволяє враховувати фактори, що не залежать від оператора логістичної системи, але суттєво впливають на ефективність перевезень. Процес інтеграції даних передбачає їх стандартизацію, синхронізацію часових міток та приведення до єдиного формату. Це забезпечує узгодженість інформаційних потоків та можливість подальшої аналітичної обробки. Перед виконанням аналітичних процедур дані проходять етап попередньої обробки, який спрямований на підвищення їх якості та придатності для аналізу. Цей етап включає очищення даних від шумів, дублювань та аномальних значень, що виникають унаслідок помилок датчиків або збоїв передачі інформації. Важливим завданням є усунення пропусків у даних шляхом інтерполяції або використання статистичних методів відновлення значень.

Наступним кроком є узгодження одиниць виміру та нормалізація показників, що дозволяє забезпечити коректність порівняння різних параметрів. Особлива увага приділяється синхронізації часових міток, оскільки різні джерела можуть передавати дані з різною частотою. Після очищення та стандартизації дані структуруються у вигляді аналітичних наборів, придатних для подальшого аналізу ефективності перевезень.

Аналітична обробка спрямована на оцінювання ефективності логістичних операцій шляхом аналізу ключових показників перевезень. На цьому етапі здійснюється обчислення часових, економічних, ресурсних та ризикових характеристик перевезень. Аналіз часових параметрів дозволяє визначити фактичну тривалість перевезень, виявити систематичні затримки та оцінити стабільність виконання маршрутів. Порівняння планових та фактичних показників дозволяє виявляти вузькі місця у логістичних процесах. Економічний аналіз спрямований на оцінювання витрат на перевезення та визначення факторів їх зростання. Важливим аспектом є аналіз витрат на одиницю перевезеного вантажу, що дозволяє оцінити економічну ефективність логістичних операцій.

Аналіз використання ресурсів включає оцінювання рівня завантаження транспорту, частоти порожніх пробігів та інтенсивності використання транспортного парку. Ці показники дозволяють визначити ступінь раціональності використання ресурсів підприємства. Оцінка ризиків перевезення передбачає аналіз факторів, що можуть негативно вплинути на безпеку доставки, цілісність вантажу та дотримання термінів. Одним із ключових завдань аналітичного блоку є виявлення відхилень від нормального функціонування логістичної системи. Такі відхилення можуть свідчити про операційні проблеми, технічні несправності або порушення регламентів, рис. 2.3.

Відхилення можуть проявлятися у вигляді перевищення нормативного часу доставки, зростання витрат палива, відхилення від заданого маршруту або несанкціонованих зупинок. Систематичні затримки можуть вказувати на проблеми дорожньої інфраструктури або неефективне планування маршрутів. Аналіз відхилень дозволяє виявити приховані проблеми, які не завжди очевидні при поверхневому контролі, але суттєво впливають на загальну ефективність перевезень.

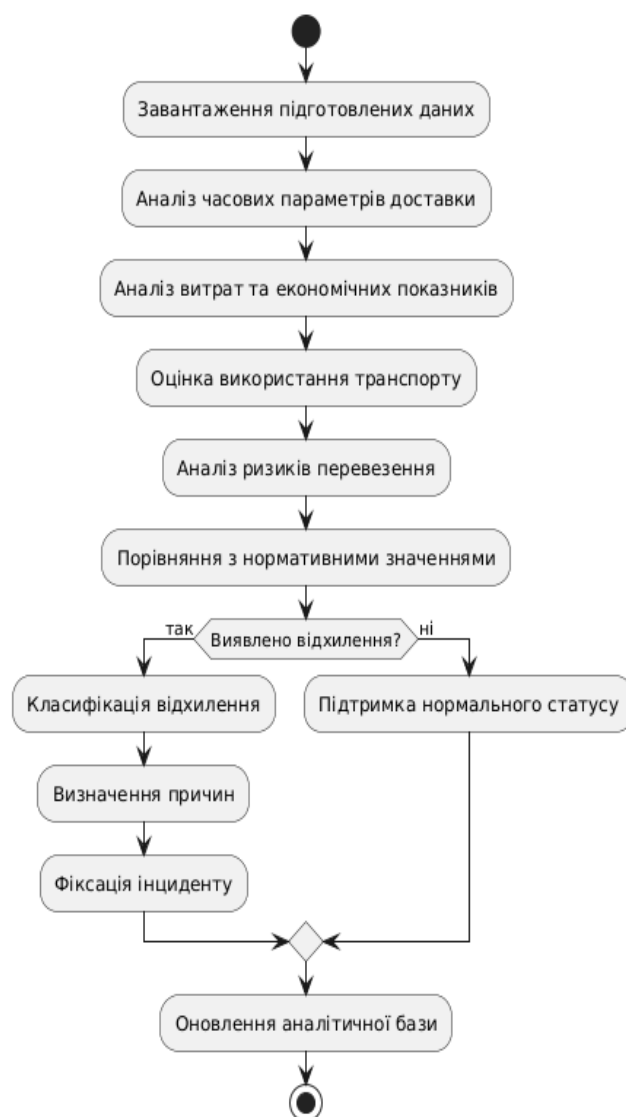


Рисунок 2.3 – Блок-схема оцінки виявлення відхилень

Після виконання аналітичних процедур система переходить до інтерпретації результатів, що передбачає перетворення числових показників у змістовні висновки.

Інтерпретація включає визначення причин відхилень, встановлення причинно-наслідкових зв'язків між параметрами перевезень та оцінювання впливу зовнішніх факторів. Наприклад, збільшення часу доставки може бути пов'язане не лише із заторами, але й з неефективним розподілом маршрутів або перевантаженням складів.

Система формує аналітичні висновки, які дозволяють оцінити поточний стан логістичних процесів та визначити напрями їх покращення. На основі аналітичних висновків система формує рекомендації, спрямовані на оптимізацію логістичних

процесів. Рекомендації можуть стосуватися різних аспектів управління перевезеннями.

Одним із напрямів є оптимізація маршрутів шляхом вибору альтернативних шляхів, що дозволяють скоротити час доставки та зменшити витрати палива. Іншим напрямом є підвищення ефективності використання транспорту шляхом консолідації вантажів та зменшення порожніх пробігів. Система також може рекомендувати зміну графіків перевезень з урахуванням транспортних потоків та пікових навантажень на дорожню мережу. У випадках підвищених ризиків система може пропонувати альтернативні маршрути або додаткові заходи безпеки.

Рекомендації можуть бути як оперативними, спрямованими на вирішення поточних проблем, так і стратегічними, орієнтованими на довгострокове підвищення ефективності логістичних процесів. Аналітичний блок системи формує набір ключових показників ефективності, що дозволяють здійснювати комплексну оцінку логістичних процесів, рис.2.4.

До таких показників належать рівень своєчасності доставки, середній час перевезення, витрати на одиницю вантажу, коефіцієнт використання транспорту та рівень логістичних ризиків. Регулярний моніторинг цих показників дозволяє оцінювати динаміку ефективності та своєчасно реагувати на негативні тенденції.



Рисунок 2.4 – Блок-схема логіки формування рекомендацій та підтримки рішень

Показники ефективності можуть використовуватися як для внутрішнього контролю, так і для оцінювання якості логістичних послуг з боку клієнтів. Для підвищення ефективності сприйняття інформації аналітичні результати подаються у вигляді візуальних панелей, графіків та інтерактивних карт. Візуалізація дозволяє оперативно оцінити стан транспортної мережі, виявити проблемні ділянки маршрутів та відстежувати ключові показники ефективності.

Інтерактивні карти дозволяють аналізувати маршрути в реальному часі, відстежувати відхилення та оцінювати транспортні потоки. Графічні панелі забезпечують моніторинг динаміки показників ефективності та виявлення тенденцій.

Аналітичний блок системи виконує функцію підтримки прийняття рішень шляхом надання обґрунтованої інформації про стан логістичних процесів. Рішення можуть стосуватися оперативного управління перевезеннями, стратегічного планування розвитку транспортної інфраструктури та оптимізації логістичних витрат. Завдяки аналітичній обробці даних керівництво підприємства отримує можливість приймати рішення на основі об'єктивних показників, що підвищує ефективність управління та знижує рівень операційних ризиків.

Запровадження аналітичного блоку дозволяє забезпечити системний підхід до управління вантажоперевезеннями. Використання аналітичних методів сприяє підвищенню прозорості логістичних процесів, зниженню витрат та підвищенню рівня сервісу.

Аналітична система дозволяє не лише фіксувати факти відхилень, але й прогнозувати їх виникнення, що створює передумови для превентивного управління та безперервного вдосконалення логістичних процесів. Логічний блок аналізу даних та формування рекомендацій є ключовим компонентом системи управління транспортними перевезеннями. Його функціонування забезпечує перетворення великих обсягів операційних даних у структуровану інформацію, придатну для прийняття управлінських рішень. Застосування аналітичних методів дозволяє виявляти неефективності, оцінювати ризики та формувати обґрунтовані рекомендації щодо оптимізації логістичних процесів. Це сприяє підвищенню якості перевезень,

зниженню витрат та забезпеченню конкурентоспроможності підприємства у сфері транспортно-логістичних послуг.

2.4 Висновки до другого розділу

У розділі було розроблено та науково обґрунтовано концепцію інформаційно-аналітичної системи аналізу транспортних перевезень і логістичних процесів, орієнтованої на підвищення ефективності управління вантажопотоками та оптимізацію функціонування транспортно-логістичних систем. Запропонований підхід поєднує формалізовані логіко-математичні моделі оцінювання ефективності перевезень, алгоритми аналізу часових, економічних і ресурсних показників, методи інтелектуальної обробки даних, а також адаптивні механізми підтримки прийняття рішень для комплексної оцінки транспортних операцій, використання рухомого складу та якості логістичного обслуговування.

Розроблена логіка забезпечує багаторівневу оцінку ефективності транспортно-логістичних систем з урахуванням їх структурної організації, функціональних режимів роботи (пікові навантаження, сезонні коливання попиту, нерівномірність транспортних потоків) та змінних зовнішніх умов. Такий підхід дозволяє перейти від фрагментарного контролю окремих параметрів перевезень до системної функціональної оцінки логістичних процесів і якості транспортного сервісу.

Застосування гібридного аналітичного підходу в системі аналізу перевезень дає змогу не лише своєчасно виявляти відхилення ключових показників від нормативних або планових значень, але й інтерпретувати їх у контексті причинно-наслідкових зв'язків між елементами логістичної системи. Зокрема, відхилення можуть бути пов'язані з неефективним плануванням маршрутів, нерівномірним розподілом навантаження, технічним станом транспортних засобів, організацією складських операцій або впливом зовнішніх факторів. Використання інтегральних показників ефективності, контекстно-адаптивних порогів і аналітичних індексів дозволяє

сформувати цілісне уявлення про стан логістичних процесів та створює передумови для переходу від реактивного реагування до проактивного управління перевезеннями.

Механізм адаптивної аналітичної моделі, інтегрований у систему, забезпечує динамічне коригування критеріїв оцінювання та правил прийняття рішень залежно від поточного операційного сценарію, накопиченого масиву даних та змін у транспортних потоках. Використання технологій великих даних, машинного навчання та імітаційного моделювання дозволяє підвищити точність прогнозування попиту на перевезення, оптимізувати розподіл ресурсів і мінімізувати ризики затримок та перевантажень. Такий підхід забезпечує стійкість функціонування системи в умовах невизначеності, неповноти або зашумленості даних і створює можливості масштабування рішення для транспортних підприємств різного масштабу та галузевої спеціалізації.

Запропонована концепція формує теоретичну та методологічну основу для побудови інтелектуальних систем управління логістикою, здатних інтегрувати телематичні дані, корпоративні інформаційні ресурси та зовнішні інформаційні потоки в єдине аналітичне середовище. Це дозволяє підвищити ефективність використання транспортних ресурсів, зменшити експлуатаційні витрати, скоротити простої та порожні пробіги, підвищити надійність доставки й рівень задоволеності клієнтів.

Реалізація логіко-математичного та аналітичного підходу сприяє переходу від традиційних методів диспетчерського контролю до інтелектуального управління транспортно-логістичними процесами, орієнтованого на прогнозування, оптимізацію та безперервне вдосконалення операцій. У контексті цифрової трансформації економіки та розвитку глобальних ланцюгів поставок така система виступає ключовим інструментом підвищення конкурентоспроможності підприємств, забезпечення їх адаптивності до динамічних умов ринку та формування високоякісної логістичної інфраструктури.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ З АНАЛІЗУ ТРАНСПОРТНИХ ПЕРЕВЕЗЕНЬ ТА ЛОГІСТИЧНИХ ПРОЦЕСІВ

3.1 Розробка архітектурної логіки для інформаційної системи

Проектування інформаційної системи за темою «Система аналізу транспортних перевезень та логістичних процесів» розпочинається з комплексного аналізу функціональних і нефункціональних вимог. Формалізація вимог є необхідною передумовою побудови цілісної архітектури системи, визначення її меж, складу підсистем та сценаріїв взаємодії з користувачами. Одним із найбільш ефективних інструментів структуризації вимог є діаграма варіантів використання, яка відображає функціональну модель системи з позиції зовнішніх акторів і дозволяє формалізувати перелік основних сервісів, що надаються інформаційною системою.

Побудова діаграми варіантів використання, рис. 3.1, забезпечує ідентифікацію ключових акторів, визначення функціональних можливостей системи, встановлення логічних зв'язків між окремими варіантами використання, а також формування підґрунтя для подальшого проектування програмної та алгоритмічної архітектури.

У межах системи аналізу транспортних перевезень та логістичних процесів визначено такі основні актори: адміністратор системи, логіст (диспетчер), аналітик, керівник підприємства, фінансовий менеджер, клієнт (замовник перевезень), а також зовнішні інформаційні системи (GPS/IoT-модулі, бухгалтерські та облікові системи). Кожен із зазначених акторів взаємодіє з системою відповідно до визначених повноважень і функціональних обов'язків.

Центральним елементом взаємодії є головний інтерфейс системи, який забезпечує автентифікацію користувачів, розмежування прав доступу та навігацію між функціональними модулями. Адміністратор здійснює керування довідниками, користувачами, параметрами алгоритмів та налаштуваннями системи. Логіст відповідає за оперативне планування та супровід перевезень. Аналітик і фінансовий менеджер використовують інструменти обробки даних та формування звітності. Керівник підприємства отримує агреговану інформацію для стратегічного контролю.

дорожньої інфраструктури, часових обмежень, завантаженості транспортних шляхів та технічних характеристик транспортних засобів. Передбачено можливість динамічного коригування маршрутів на основі даних моніторингу в реальному часі.

Моніторинг транспортних засобів забезпечується шляхом інтеграції з телематичними та IoT-пристроями. Система здійснює контроль місцезнаходження транспорту, швидкісного режиму, відхилень від маршруту, витрат палива та технічного стану рухомого складу. Отримані дані використовуються як для оперативного управління, так і для подальшого аналітичного опрацювання.

Ключовим елементом розширеної функціональної моделі є аналітика перевезень. Даний варіант використання включає автоматизований збір і агрегацію даних із внутрішніх підсистем і зовнішніх джерел, їх статистичну та економічну обробку, аналіз часових рядів, виявлення аномалій і закономірностей. Реалізуються алгоритми прогнозування попиту на перевезення, завантаженості автопарку, витрат ресурсів і технічного зносу транспортних засобів. Аналітичний модуль підтримує сценарне та імітаційне моделювання логістичних процесів.

Тісно пов'язаним із блоком аналітики є контроль ефективності перевезень. У межах цього варіанту використання система здійснює автоматичний розрахунок ключових показників ефективності, зокрема коефіцієнта використання транспорту, рівня простоїв, середнього часу доставки, частки порожніх пробігів, витрат на одиницю маршруту та рентабельності перевезень. Передбачено порівняльний аналіз планових і фактичних показників, оцінку ефективності окремих маршрутів, транспортних засобів і підрозділів, а також виявлення відхилень і неефективностей.

На основі результатів аналітики система формує управлінські звіти та рекомендації щодо оптимізації маршрутів, перерозподілу ресурсів, коригування графіків роботи, ініціювання технічного обслуговування або перегляду тарифної політики. Таким чином реалізується замкнений цикл управління: планування – виконання – моніторинг – аналіз – коригування.

Окремим функціональним блоком є управління ресурсами, яке охоплює облік рухомого складу, персоналу, технічного обслуговування, складських потужностей та

фінансових показників. Інформація з цього модуля інтегрується з аналітичною підсистемою та використовується для комплексної оцінки діяльності підприємства.

Персональний кабінет користувача забезпечує доступ до історії перевезень, статусів заявок, фінансових документів і аналітичної інформації в межах наданих прав доступу.

Таким чином, діаграма варіантів використання системи аналізу транспортних перевезень та логістичних процесів відображає інтегровану функціональну модель, що поєднує операційне управління, моніторинг у реальному часі, аналітичну обробку даних та контроль ефективності. Формалізація вимог у такому вигляді створює методологічну основу для побудови інтелектуальної інформаційної системи, здатної забезпечити підвищення ефективності логістичних операцій, зниження витрат і стратегічну адаптивність підприємства в умовах динамічного транспортного середовища.

Наступним етапом проектування та розробки є формування діаграми послідовності дій, яка дозволяє конкретизувати та верифікувати основні логічні сценарії системи.

Розробка діаграми послідовності дій для програмного забезпечення, яке надає транспортні послуги, є важливим етапом у процесі його створення та впровадження.

Діаграма послідовності дій допомагає уточнити функціональні вимоги до програмного забезпечення, визначаючи порядок подій та взаємодію між різними компонентами системи. Це дозволяє точно окреслити, як користувачі будуть взаємодіяти з додатком під час надання транспортних послуг.

Розуміння послідовності дій дає змогу виявити потенційні затримки або неефективності в процесі надання транспортних послуг. Це дозволяє розробникам зосередитися на вдосконаленні та оптимізації процесів, забезпечуючи більш ефективне та задовільне використання додатка.

Діаграма послідовності дій сприяє коректній взаємодії між усіма елементами системи. Вона дозволяє виявити потенційні конфлікти та недоліки в логіці взаємодії, що дає змогу їх виправити ще на ранніх етапах розробки.

Розробка діаграми послідовності дій дозволяє виявити можливі проблеми та ризики до повного завершення реалізації програмного продукту. Це допомагає уникнути дорогих виправлень у майбутньому і забезпечити більш ефективне використання ресурсів під час розробки.

Добре розроблена діаграма послідовності дій допомагає покращити користувацький досвід. Вона дає змогу зрозуміти, як користувачі взаємодіють з програмою, і дозволяє розробникам вдосконалити інтерфейс та функціонал відповідно до потреб користувачів.

Таким чином, розробка діаграми послідовності дій для програмного забезпечення, яке надає транспортні послуги, є критично важливим етапом у процесі розробки, що допомагає забезпечити ефективність, якість та коректність програмного продукту. На рис. 3.2 зображено діаграму послідовності дій до програмного засобу.

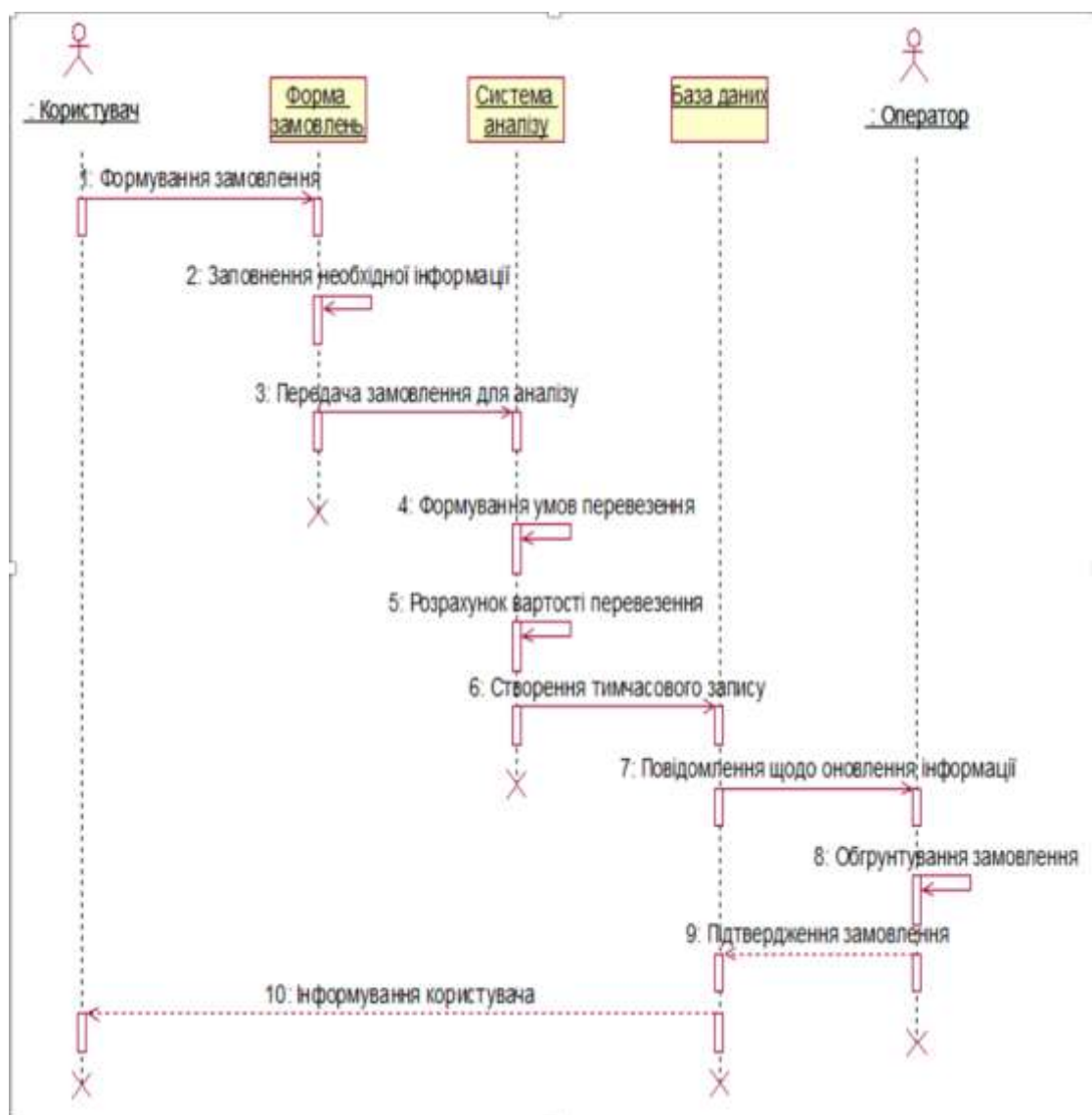


Рисунок 3.2 – Діаграма послідовності дій формування замовлення

Спершу користувач повинен зареєструватися в програмному забезпеченні, створивши свій обліковий запис. Для цього потрібно подати заяву адміністратору програмного забезпечення, який активує обліковий запис. Після активації користувач може увійти в систему. Вся робота в програмному забезпеченні базується на сучасних алгоритмах безпеки, тому використання системи без авторизації є неможливим.

На рис. 3.3 зображено діаграму послідовності принципів роботи бази даних програмної системи.

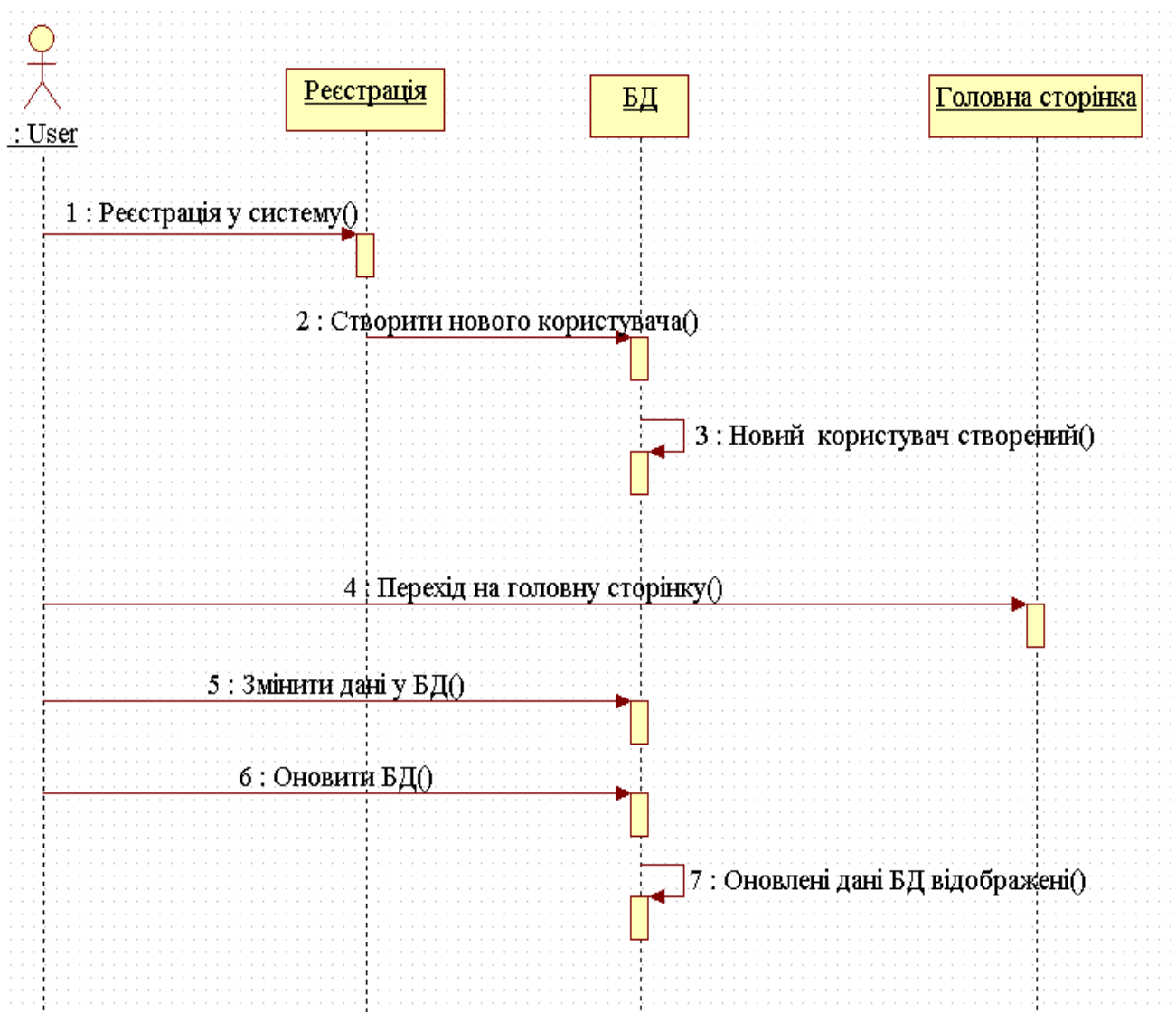


Рисунок 3.3 – Діаграма послідовності роботи бази даних

Опис станів програмної системи дозволяє розробникам розуміти, як відбуваються переходи між модулями, а також ідентифікувати бібліотеки та розширення, що використовуються при запуску кожного модуля. Крім того, діаграма станів допомагає визначити взаємодію з операційною системою через виклик окремих компонентів або передачу їм даних. Таке рішення широко застосовується при розробці програмних продуктів, орієнтованих на роботу з користувачами.

Система організована таким чином, що користувач повинен підтверджувати процес виконання умов на перевезення, а перевізник контролює весь процес перевезення відповідно до всіх станів та етапів, які супроводжують цей процес. На рис. 3.4 зображено діаграму станів програмного засобу.



Рисунок 3.4 – Діаграма станів програмного засобу

Розробка діаграми класів для програмного забезпечення, яке надає транспортні послуги, є надзвичайно важливим етапом у процесі створення, оскільки вона забезпечує структурний огляд продукту та визначає взаємозв'язки між його компонентами. Є кілька причин, які підкреслюють важливість цього завдання:

По-перше, діаграма класів забезпечує прозорість структури, пропонуючи чітке уявлення про програму, визначаючи класи, їх атрибути, методи та взаємозв'язки. Це допомагає розробникам краще зрозуміти логіку роботи програми та підтримувати послідовність у всій системі.

По-друге, аналіз діаграми класів може виявити недоліки у дизайні програми, такі як зайві залежності або непотрібні класи, що дозволяє вчасно визначити

необхідність у рефакторингу коду для покращення його читабельності, ефективності та можливості розширення.

Також, діаграма класів допомагає планувати розвиток продукту, показуючи, як можна розширити функціональність програми та які зміни необхідно внести для цього. Це сприяє ефективному плануванню майбутнього розвитку та встановленню пріоритетів у розробці.

Крім того, діаграма класів слугує ефективним інструментом комунікації між членами команди розробників, надаючи загальне розуміння архітектури програми та полегшуючи спільну роботу над складними завданнями.

Важливо зазначити, що діаграма класів є невід'ємною частиною документації проекту. Вона надає вичерпний огляд структури програмного забезпечення, що допомагає новим членам команди швидше ознайомитися з кодом і сприяє підтримці та розвитку продукту в майбутньому. Діаграма класів дозволяє новим розробникам легко зрозуміти логіку та структуру системи, що знижує криву навчання та підвищує ефективність їхньої роботи. Крім того, наявність такої діаграми спрощує процес обговорення та вирішення питань, що виникають у процесі розробки, оскільки всі учасники мають спільне розуміння структури проекту.

Нарешті, аналіз діаграми класів дозволяє виявити потенційні проблеми та слабкі місця в архітектурі програми на ранніх етапах розробки, запобігаючи виникненню серйозних проблем у майбутньому та забезпечуючи стабільну та надійну роботу програмного продукту. Завдяки діаграмі класів можна ідентифікувати конфлікти або неефективні зв'язки між компонентами системи, що може призвести до оптимізації структури коду та покращення продуктивності програми. Раннє виявлення цих проблем дозволяє розробникам виправити їх до того, як вони стануть критичними, що зменшує витрати на подальшу підтримку та модифікацію програмного забезпечення. Діаграма класів програмного засобу зображена на рис. 3.5.

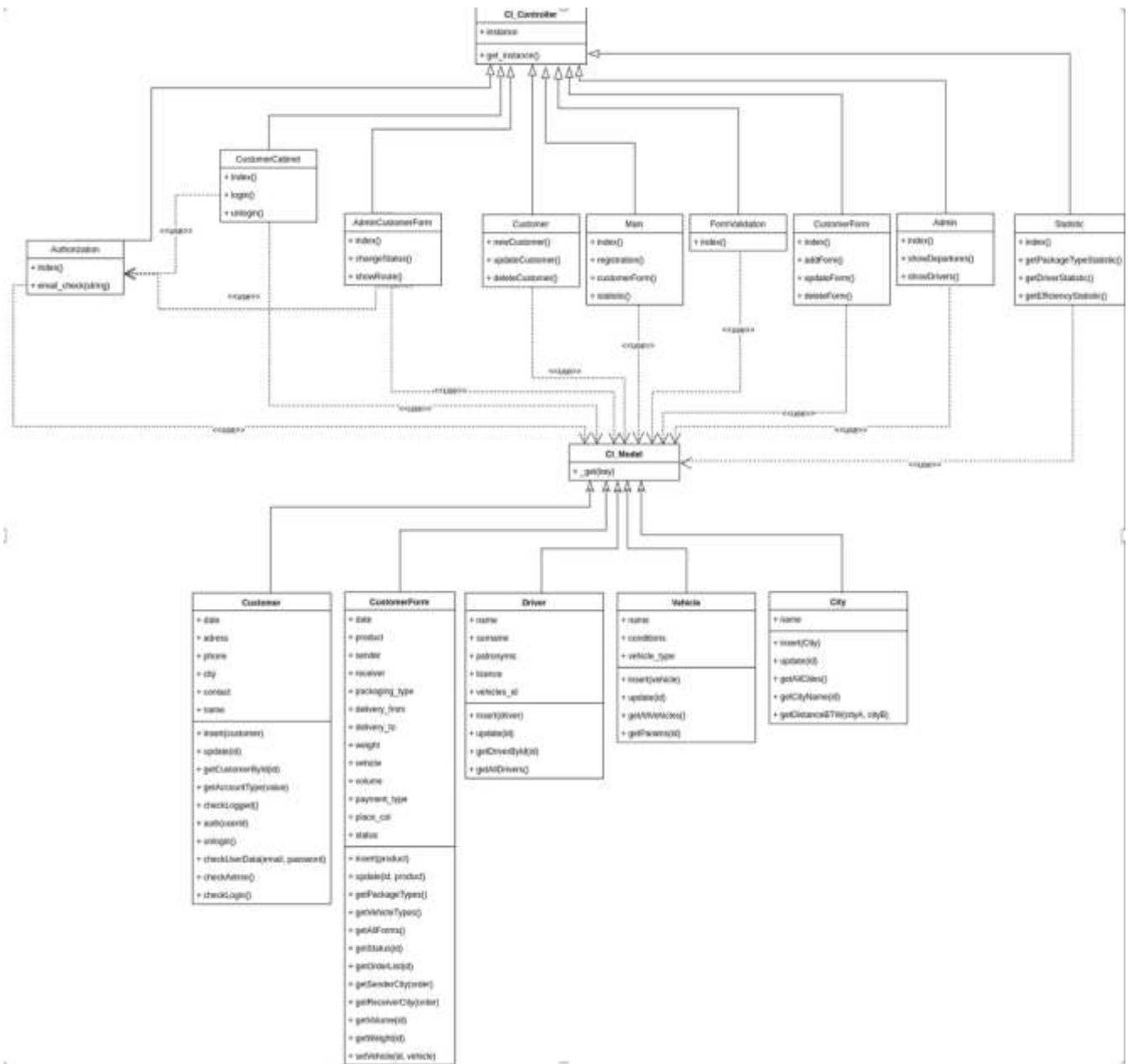


Рисунок 3.5 – Діаграма класів програмного засобу

Розроблена класова структура має в собі наступні класи:

- Main - клас з якого починається робота з користувачем сайту.
- CI_Controller - клас фреймворка, який повинен успадковувати кожен призначений для користувача клас.
- Authorization - клас для проведення авторизації. Цей клас виконує перевірку заповнених даних за відповідністю.

- CustomerCabinet - клас призначений для взаємодії користувача з особистим кабінетом.
- AdminCustomerForm – клас призначений для роботи в особистому кабінеті оператора.
- Customer - клас для роботи з даними користувача.
- FormValidation - клас для перевірки правильності заповнення форми реєстрації.
- CustomerForm - клас для перевірки правильності заповнення форми подачі нового замовлення.
- Admin - клас для роботи з даними доступними адміністратору.
- Statistic – клас, який описує алгоритми для обробки даних і виведення статистики.
- CI_Model - клас фреймворка, який повинен успадковувати кожна модель.
- Customer – модель, що реалізує інтерфейс взаємодії к даними користувачем.
- CustomerForm - модель, що реалізує інтерфейс взаємодії з поданими анкетами користувачів.
- Driver - модель, що реалізує інтерфейс взаємодії з інформацією про водіїв.
- Vehicle - модель, що надає інтерфейс взаємодії з інформацією про транспортні засоби.
- City - модель, що описує інтерфейс взаємодії з інформацією про міста та дистанцію між ними.

Кожен розроблений клас дає змогу об'єднати схожі програмні рішення у відокремлені структурні одиниці та використовувати їх як незалежні рішення у програмному коді. Всі використані класи спрямовані на патерни, що спрощує процес розробки інформаційних систем та дозволяє додати одну логічну складову до будь-яких інших практичних компонентів, які можуть бути використані в інформаційній системі. Розроблена архітектурна документація та діаграми допомагають визначити

структуру бази даних та реалізувати інформаційну систему згідно з вхідними функціональними вимогами.

Діаграма бази даних відображає структуру та зв'язки між елементами бази даних системи. Її розробка є ключовим етапом у процесі розробки програми для оренди автомобілів. Ця діаграма дозволяє описати структуру даних, що використовуються в програмі, та їх зв'язки із способами зберігання та взаємодії. Розробка такої діаграми допомагає розробникам краще розуміти організацію даних та їх взаємозв'язки, що в свою чергу полегшує створення ефективних запитів до бази даних, підвищує продуктивність програми та уникнення помилок при роботі з даними.

Структура бази даних для програмного засобу з логістично-транспортних процесів включає наступні таблиці:

- Таблиця "Користувачі" (Users). Ця таблиця містить дані про користувачів системи оренди. Кожен користувач ідентифікується унікальним `user_id`. Додаткова інформація включає ім'я, прізвище, електронну пошту, пароль, номер телефону та адресу кожного користувача.
- Таблиця "Транспортні засоби" (Vehicles). У цій таблиці зберігаються дані про транспортні засоби, які доступні для оренди. Кожен засіб ідентифікується за допомогою унікального `vehicle_id`. Інша інформація включає марку, модель, рік випуску, тип транспорту, вартість оренди за годину та день, а також стан кожного засобу.
- Таблиця "Оренди" (Rentals). У цій таблиці зберігаються дані про оренду транспортних засобів користувачами. Кожен орендований засіб ідентифікується за `rental_id`. Крім того, вказуються `user_id`, що відповідає користувачу, що орендує транспортний засіб, і `vehicle_id`, який вказує на конкретний засіб, дату та час початку та закінчення оренди та суму оренди.
- Таблиця "Водії" (Drivers). У цій таблиці зберігаються дані про водіїв, які використовують транспортні засоби. Кожен водій ідентифікується унікальним

driver_id. Додаткова інформація включає user_id, що вказує на користувача-водія, номер водійського посвідчення та категорію водійського посвідчення.

- Таблиця "Маршрути" (Routes). У цій таблиці зберігаються дані про маршрути, які використовуються користувачами. Кожен маршрут ідентифікується унікальним route_id. Інша інформація включає початкову та кінцеву точки маршруту, відстань та тривалість подорожі.

- Таблиця "Оцінки" (Ratings). У цій таблиці зберігаються дані про оцінки, які користувачі ставлять орендованим засобам. Кожна оцінка ідентифікується унікальним rating_id. Інша інформація включає rental_id, який вказує на конкретну оренду, оцінку та коментар.

- Таблиця "Платежі" (Payments). У цій таблиці зберігаються дані про платежі за оренду транспортних засобів. Кожен платіж ідентифікується унікальним payment_id. Інша інформація включає rental_id, який вказує на конкретну оренду, суму платежу, дату та час платежу та статус оплати.

- Таблиця "Історія використання транспорту" (Vehicle_Usage_History). У цій таблиці зберігаються дані про історію використання транспортних засобів. Кожен запис ідентифікується унікальним usage_id. Інша інформація включає rental_id, що вказує на оренду, vehicle_id, що вказує на транспортний засіб, дату та час початку та закінчення використання.

- Таблиця "Повідомлення" (Messages). У цій таблиці зберігаються дані про повідомлення між користувачами системи оренди. Кожне повідомлення ідентифікується унікальним message_id. Інша інформація включає sender_id та receiver_id, які вказують на користувача-відправника та користувача-одержувача, текст повідомлення та дату та час відправлення.

Графічне уявлення діаграми базів даних, зображено на рис. 3.6.

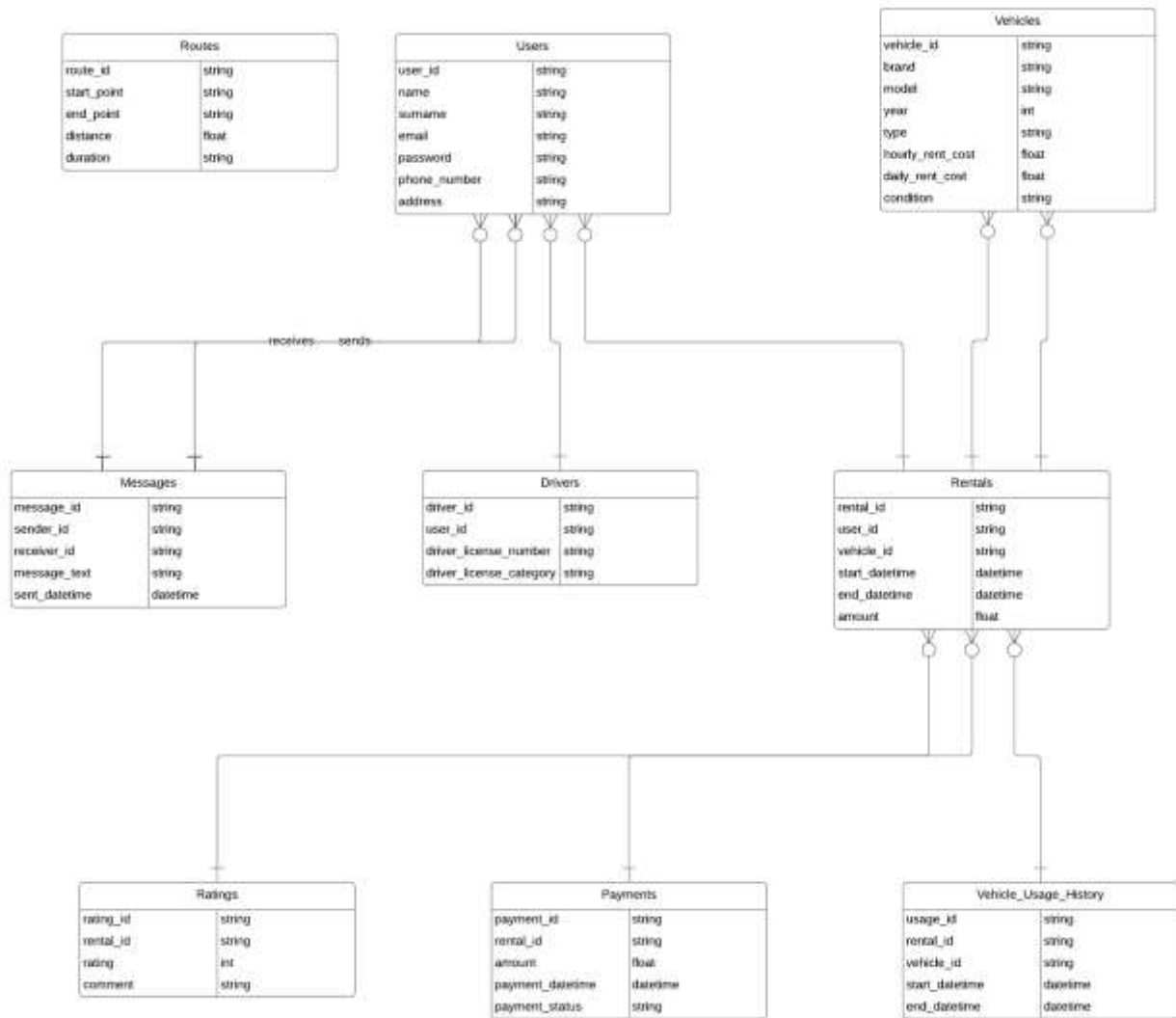


Рисунок 3.6 – Діаграма бази даних до програмного засобу

Розробка архітектурної документації для програмного забезпечення з транспортно-логістичної логіки, є ключовим етапом у створенні високоякісного продукту. Ця документація включає детальний опис функціональності системи, компонентів, їх взаємозв'язків та вимог до системи. Вона забезпечує всіх учасників проекту єдиною картинкою того, як працює система, та дозволяє контролювати її розвиток для забезпечення відповідності потребам замовника.

Під час розробки архітектурної документації для цієї інформаційної системи проводилося глибоке дослідження та аналіз вимог замовника. Визначалися основні компоненти системи, їх функціональність та взаємозв'язки між ними. Були створені

діаграми варіантів використання, послідовності дій, станів, класів та бази даних, які надають зрозумілий та візуальний опис системи.

Одним із важливих результатів цієї роботи є можливість зазначити потенційні ризики та проблеми, які можуть виникнути під час розробки, тестування та експлуатації системи, та прийняти відповідні заходи для їх уникнення. Більше того, архітектурна документація сприяє оптимізації роботи над проектом, прискорює процес розробки, зменшує ймовірність виникнення помилок та підвищує загальну якість програмного продукту.

Розроблений програмний код, згідно наведеної архітектури приведено в додатку А.

3.2 Розробка та опис інтерфейсу інформаційної системи

Розроблена інформаційно-аналітична платформа є комплексним веб-рішенням класу TMS (Transportation Management System) з потужним вбудованим модулем бізнес-аналітики та прийняття рішень.

Вона охоплює повний життєвий цикл логістичного замовлення — від первинного запиту клієнта та автоматичного/ручного розрахунку вартості, через планування, призначення транспорту та водія, онлайн-моніторинг виконання, до післярейсового аналізу ефективності, економічних показників та якості обслуговування.

Інтерфейс чітко розділений на дві великі зони: адміністративно-аналітичну (для логістів, диспетчерів, керівництва, фінансового відділу) та клієнтську (для замовників — фізичних та юридичних осіб).

Головна панель адміністратора є ключовим елементом управлінського контролю та оперативного прийняття рішень. Інтерфейс побудовано за принципом інформаційної панелі з модульними віджетами, які оновлюються в реальному часі, рис. 3.7.

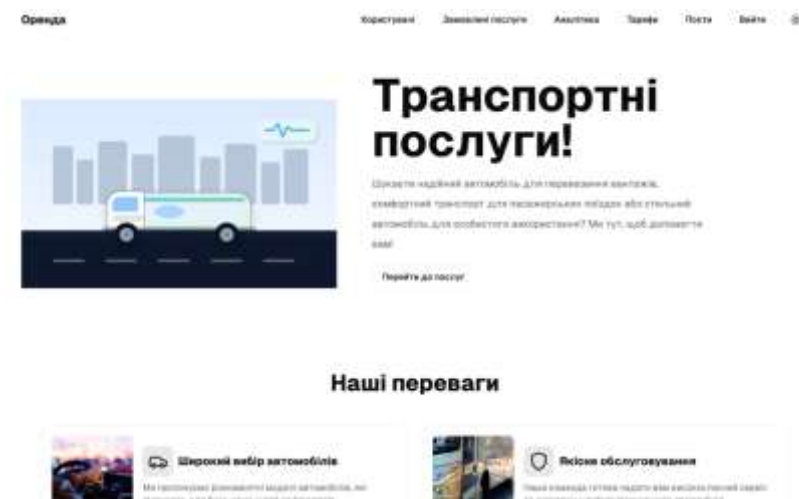


Рисунок 3.7 – Головна сторінка адміністратора системи

На рисунку 3.8 відображено сторінку з інформаційними блоками системи, яка містить узагальнену інформацію про діяльність компанії та основні функціональні можливості платформи. Інтерфейс сторінки структуровано у вигляді тематичних блоків, кожен з яких містить короткий опис певного напрямку діяльності або функції системи. Така структура подання інформації забезпечує зручність сприйняття контенту та дозволяє користувачам швидко ознайомитися з можливостями сервісу. Інформаційні блоки можуть включати опис логістичних послуг, переваги використання системи, особливості організації перевезень, а також інші відомості, що формують загальне уявлення про діяльність транспортно-логістичної компанії.

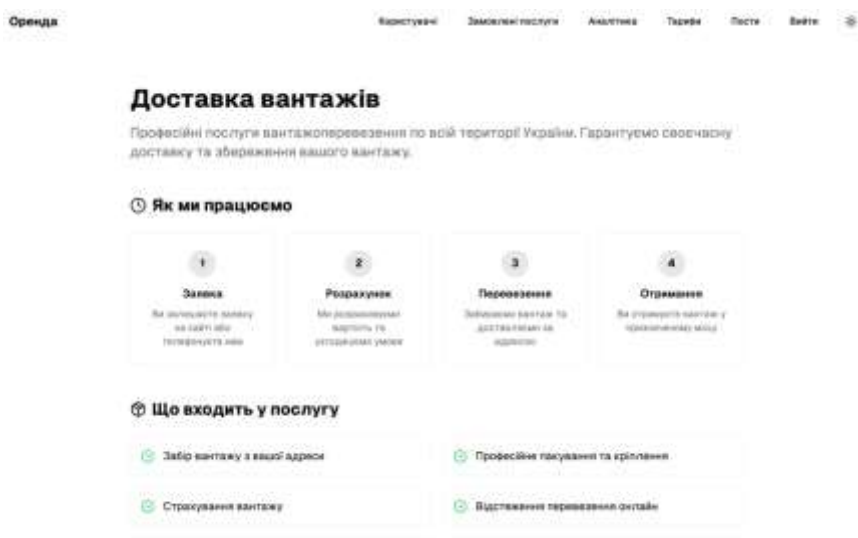


Рисунок 3.8 – Сторінка з інформаційними блоками

Сторінка історії компанії містить інформаційні матеріали, що відображають основні етапи становлення та розвитку транспортно-логістичного підприємства. У структурі сторінки можуть бути представлені хронологічні відомості про створення компанії, ключові етапи її розвитку, розширення діяльності, впровадження нових технологій та вдосконалення логістичних процесів. Наявність такого розділу сприяє формуванню довіри з боку користувачів та клієнтів системи, оскільки дозволяє ознайомитися з досвідом та досягненнями компанії у сфері транспортних перевезень, рис. 3.9.

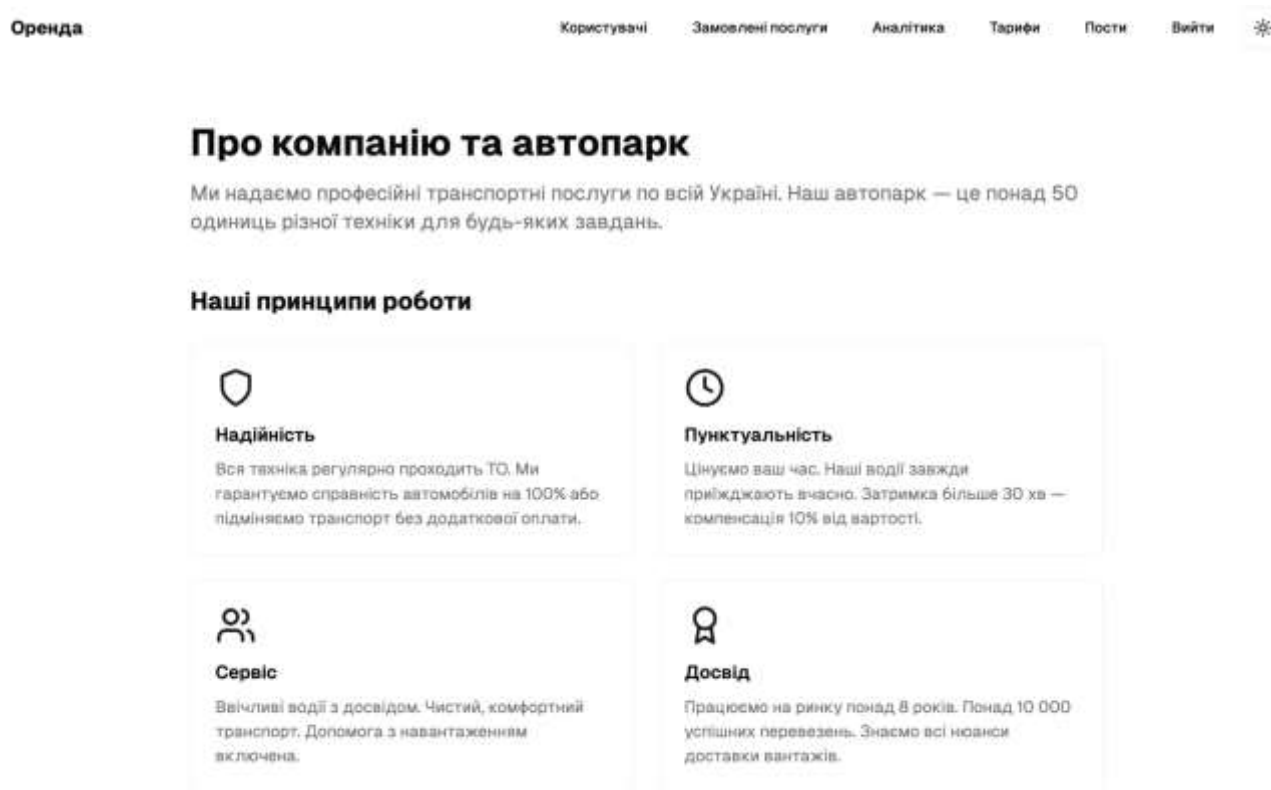


Рисунок 3.9 – Сторінка історії компанії

Сторінка з умовами та тарифами системи, рис. 3.10, містить інформацію щодо умов надання транспортно-логістичних послуг та тарифної політики компанії. Інтерфейс сторінки передбачає структуроване представлення тарифів залежно від виду перевезення, типу транспортного засобу, відстані маршруту та інших факторів, що впливають на вартість логістичних послуг.

Крім того, на сторінці можуть бути відображені правила співпраці з клієнтами, умови оформлення замовлень, порядок розрахунку вартості перевезень та інші важливі аспекти надання послуг. Такий підхід забезпечує прозорість тарифної політики та сприяє підвищенню рівня інформованості користувачів.

Оренда

Користувачі

Замовлені послуги

Аналітика

Тарифи

Пости

Вийти

☰

Умови та тарифи

Прозорі ціни та зрозумілі умови. Жодних прихованих платежів.

\$ Тарифи на оренду

Тип транспорту	Година	Доба	З водієм
Економ (Skoda Octavia)	від 200 грн	від 1500 грн	+300 грн/год
Бізнес (Toyota Camry)	від 350 грн	від 2500 грн	+400 грн/год
Мікроавтобус (8-12 місць)	від 400 грн	від 3000 грн	+350 грн/год
Автобус (20-50 місць)	від 800 грн	від 6000 грн	+500 грн/год

* Ціни можуть відрізнятися в залежності від сезону та конкретного автомобіля. Точну ціну уточнюйте при бронюванні.

Вартість доставки вантажів

По місту Київ

По Україні (за 1 км)

Рисунок 3.10 – Сторінка з умовами та тарифами системи

На рисунку 3.11 зображено сторінку адміністрування користувачів інформаційної системи. Даний інтерфейс призначений для управління обліковими записами користувачів та надання відповідних прав доступу до функціональних можливостей системи. На сторінці представлено список зареєстрованих користувачів із основними параметрами їхніх облікових записів, такими як ім'я, роль у системі, контактні дані та статус активності. Адміністратор має можливість здійснювати додавання нових користувачів, редагування існуючих облікових записів, а також блокування або видалення користувачів у разі необхідності.

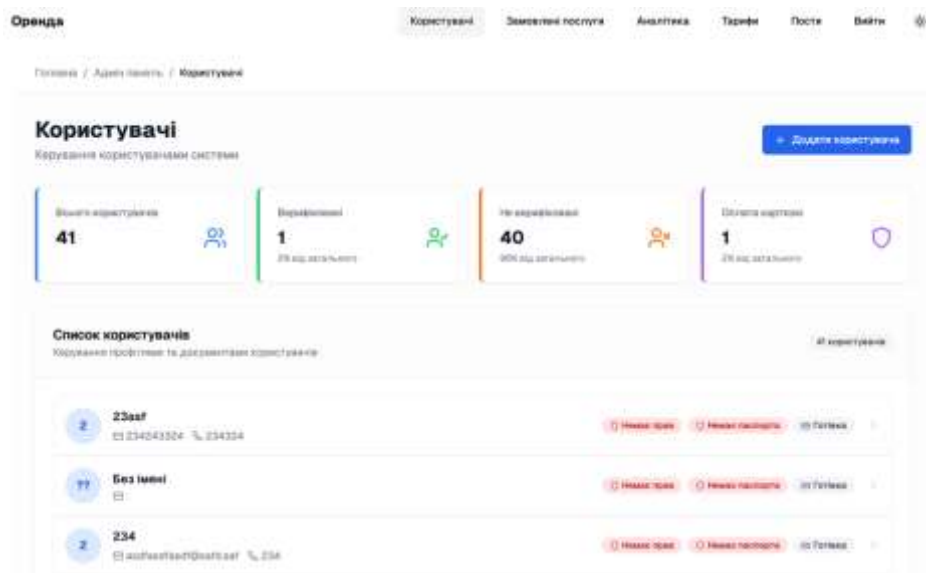


Рисунок 3.11 – Сторінка з адмініструванням користувачів системи

В системі рохроблено форму створення нового користувача системи. Така форма містить набір полів для введення основної інформації про користувача, включаючи ім'я, електронну адресу, контактні дані, пароль доступу та роль у системі. Заповнення відповідних полів дозволяє створити новий обліковий запис, який надалі може використовуватися для роботи із системою. Форма також може передбачати механізми перевірки коректності введених даних, що сприяє підвищенню рівня безпеки та надійності системи, рис. 3.12.

Рисунок 3.12 – Форма створення нового користувача

На рис. 3.13 представлено сторінку управління замовленнями транспортних перевезень. Інтерфейс сторінки містить список усіх зареєстрованих замовлень із відображенням їх поточного статусу, що дозволяє здійснювати оперативний контроль за процесом виконання логістичних операцій. Кожне замовлення містить інформацію про клієнта, маршрут перевезення, тип транспортного засобу, дату оформлення та поточний стан виконання. Використання статусів замовлень дозволяє відслідковувати етапи обробки заявки – від моменту її створення до завершення перевезення.



Рисунок 3.13 – Сторінка замовлень в системі з статусами

Форма відкритого замовлення, рис. 3.14, містить детальну інформацію про конкретне замовлення, включаючи відомості про клієнта, характеристики вантажу, маршрут перевезення, обраний транспортний засіб та інші параметри логістичної операції. Крім того, інтерфейс форми дозволяє здійснювати редагування окремих параметрів замовлення, змінювати його статус або виконувати інші управлінські дії, пов'язані з обробкою заявки.



Рисунок 3.14 – Форма відкритого замовлення

Сторінка з аналітикою перевезень призначена для аналізу транспортних перевезень. Такий модуль забезпечує відображення статистичних показників діяльності логістичної системи у вигляді графіків, діаграм та таблиць. За допомогою даної сторінки менеджери та адміністратори можуть здійснювати оцінку ефективності перевезень, аналізувати завантаженість транспортних ресурсів, визначати динаміку замовлень та виявляти тенденції розвитку логістичних процесів, рис. 3.15.

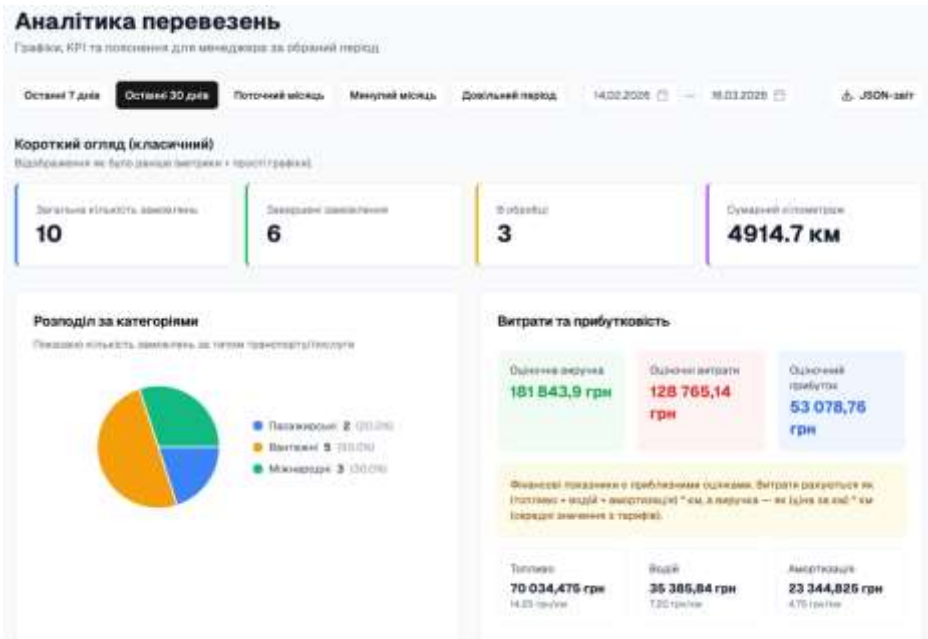


Рисунок 3.15 – Сторінка з аналітикою перевезень

В системі реалізовано форму аналітичного відображення даних, рис. 3.16, яка містить узагальнені інсайти щодо функціонування логістичної системи. Даний інтерфейс призначений для відображення результатів аналітичної обробки інформації, що формується на основі даних про транспортні перевезення, маршрути, завантаженість транспорту та ефективність виконання замовлень. У межах цієї сторінки можуть відображатися ключові управлінські показники, що дозволяють менеджменту оцінювати ефективність діяльності компанії, визначати проблемні зони у логістичних процесах та формувати рекомендації щодо оптимізації транспортних операцій.

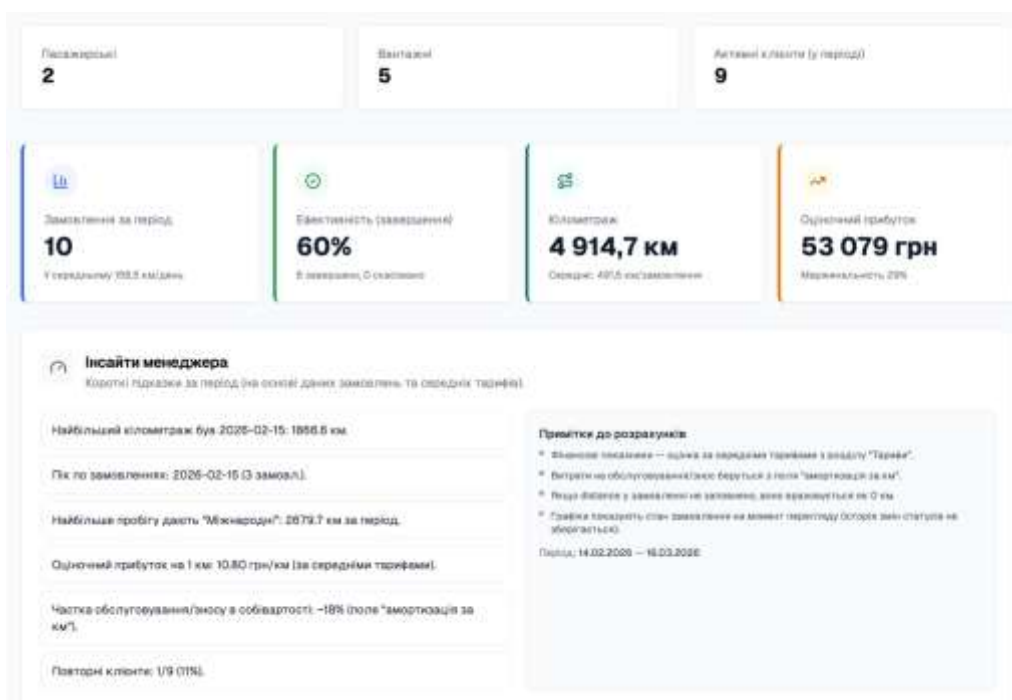


Рисунок 3.16 – Форма аналітики з інсайтами від менеджмента

На рис. 3.17 зображено аналітичну форму, яка відображає динаміку зміни основних показників діяльності системи транспортних перевезень. Інтерфейс даної сторінки включає графічні елементи, що демонструють зміну кількості замовлень у часі, сумарний кілометраж виконаних перевезень, а також розподіл замовлень за різними статусами. Така візуалізація дозволяє здійснювати комплексний аналіз

розвитку логістичних процесів та визначати тенденції зростання або зменшення обсягів перевезень.

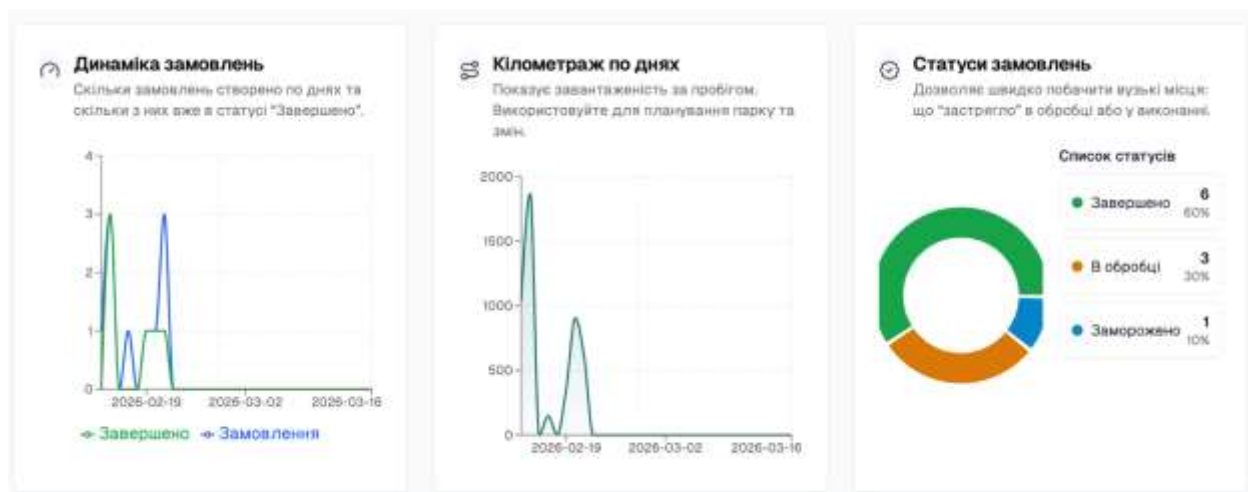


Рисунок 3.17 – Форма динаміки замовлень, кілометражу перевезень та статусів замовлень

В системі реалізовано логіку багатопараметричної оцінок кілометражу та актуальності замовлень за видами транспорту. Вона відображає статистичні дані щодо використання різних видів транспортних засобів у процесі перевезення вантажів, рис. 3.18. Система формує аналітичні показники, які демонструють сумарний кілометраж перевезень для кожної категорії транспорту, а також розподіл замовлень залежно від типу транспортного трафіку. Це дозволяє визначати найбільш затребувані види транспортних ресурсів, оцінювати ефективність їх використання та планувати подальше розширення транспортного парку компанії.

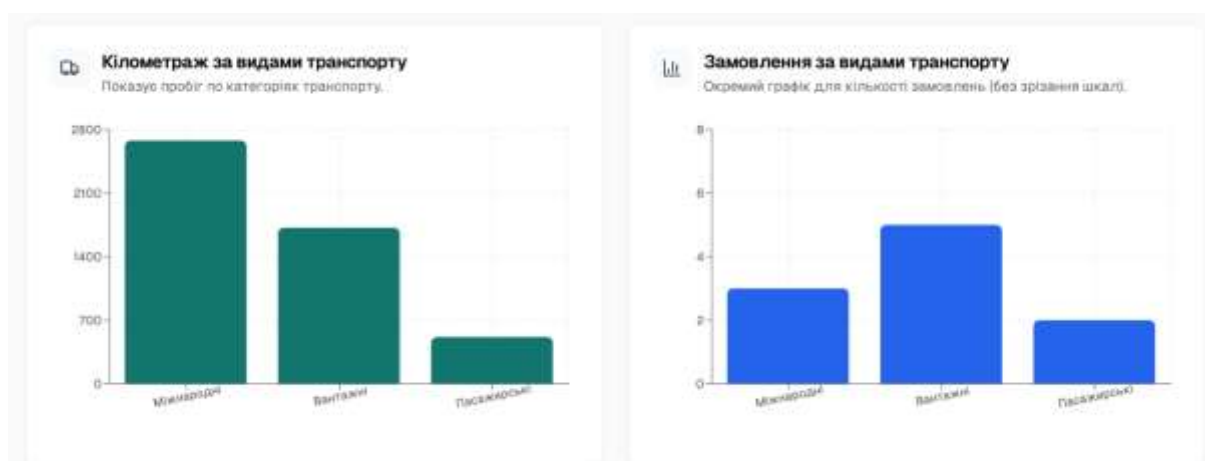


Рисунок 3.18 – Аналітика за кілометражем видів транспортів та розподіл замовлень за видами трафіку

Для розширеної логіки аналізу показників ефективності системи, розроблено форму аналітичного розрахунку економічних показників діяльності системи, рис. 3.19. Дана сторінка містить інформацію щодо оціночних витрат на обслуговування транспортних засобів, витрат на виконання перевезень, а також можливого прибутку, отриманого від реалізації логістичних послуг. Використання такої форми дозволяє здійснювати фінансовий аналіз ефективності функціонування транспортно-логістичної системи та приймати обґрунтовані управлінські рішення щодо оптимізації витрат.



Рисунок 3.19 – Форма виведення оціночних витрат на обслуговування та прибуток

Серед особливих компонентів аналітичного модуля є аналітична форма визначення пікових навантажень у системі транспортних перевезень. Інтерфейс сторінки дозволяє визначати періоди часу, у які спостерігається найбільша кількість замовлень або максимальне використання транспортних ресурсів.

Отримані дані можуть бути використані для оптимізації планування перевезень, розподілу транспортних засобів та ефективного управління логістичними потоками, рис. 3.20.

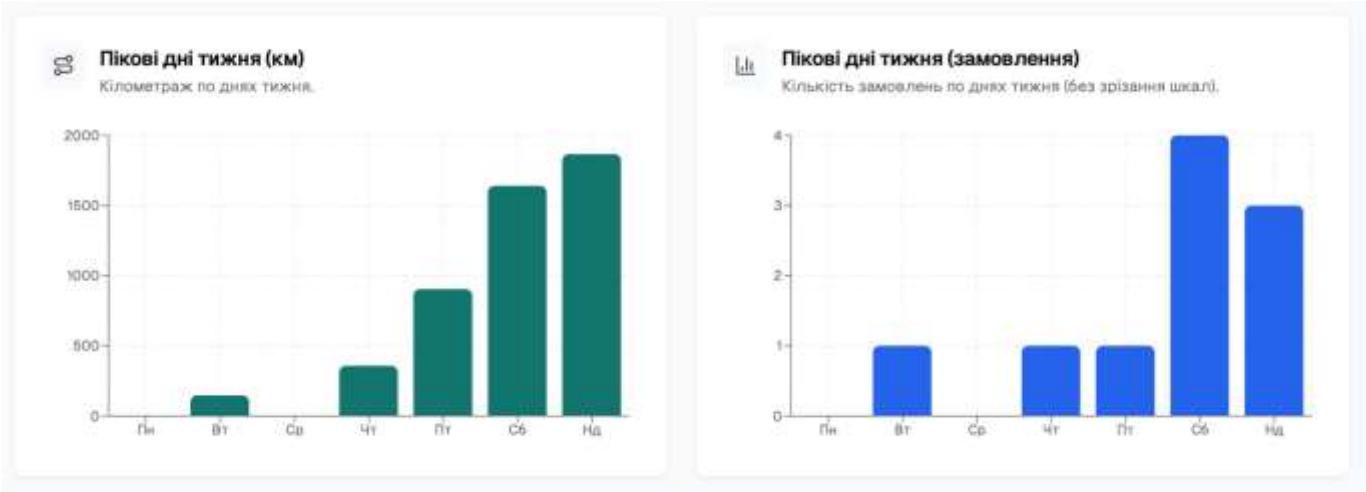


Рисунок 3.20 – Форма виводу пікового навантаження

Важливою частиною є реалізація компоненту аналізу клієнтської активності через замовлень, а також оцінка популярних маршрутів. Така концепція дозволяє розширювати логіку сприйняття системи та побудувати актуальний інструмент для аналізу, рис. 3.21.

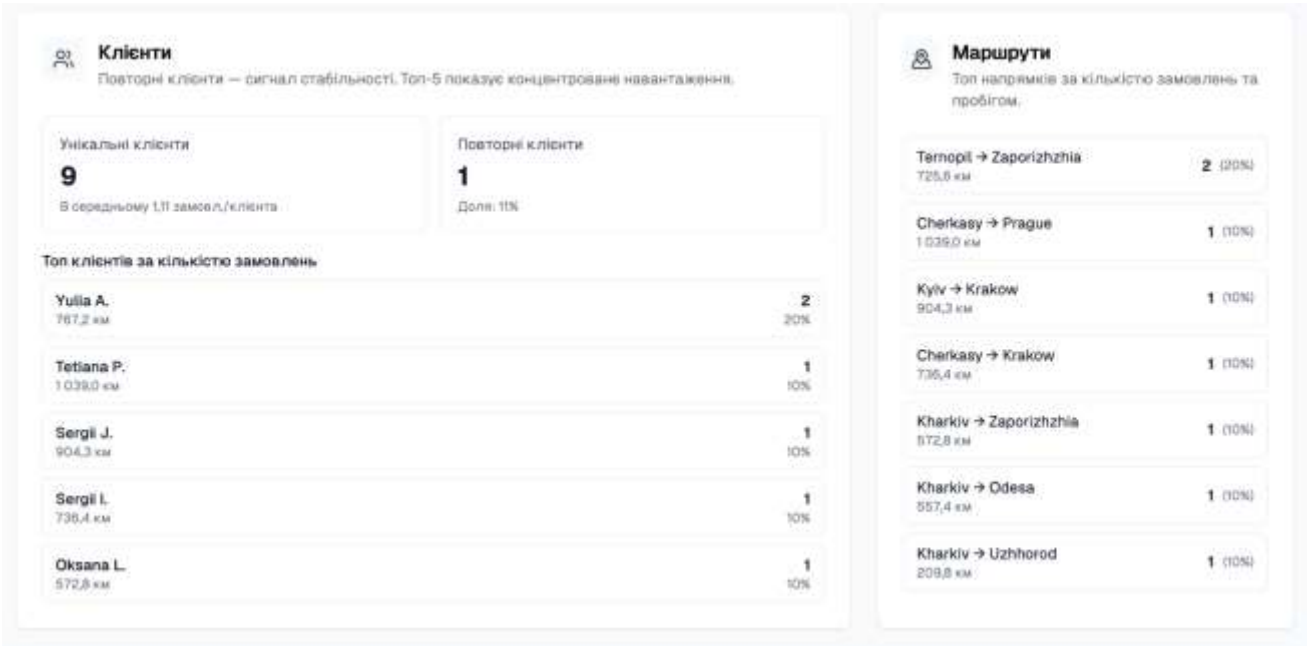


Рисунок 3.21 – Форма статистики по клієнтам та маршрутам

На рис. 3.22 показано форму відображення типів логістичних послуг, що надаються компанією. Сторінка містить перелік доступних послуг, включаючи різні види транспортних перевезень, додаткові логістичні сервіси, а також можливості оренди транспортних засобів. Інтерфейс даного розділу забезпечує зручний перегляд та адміністрування переліку послуг, що надаються користувачам системи.

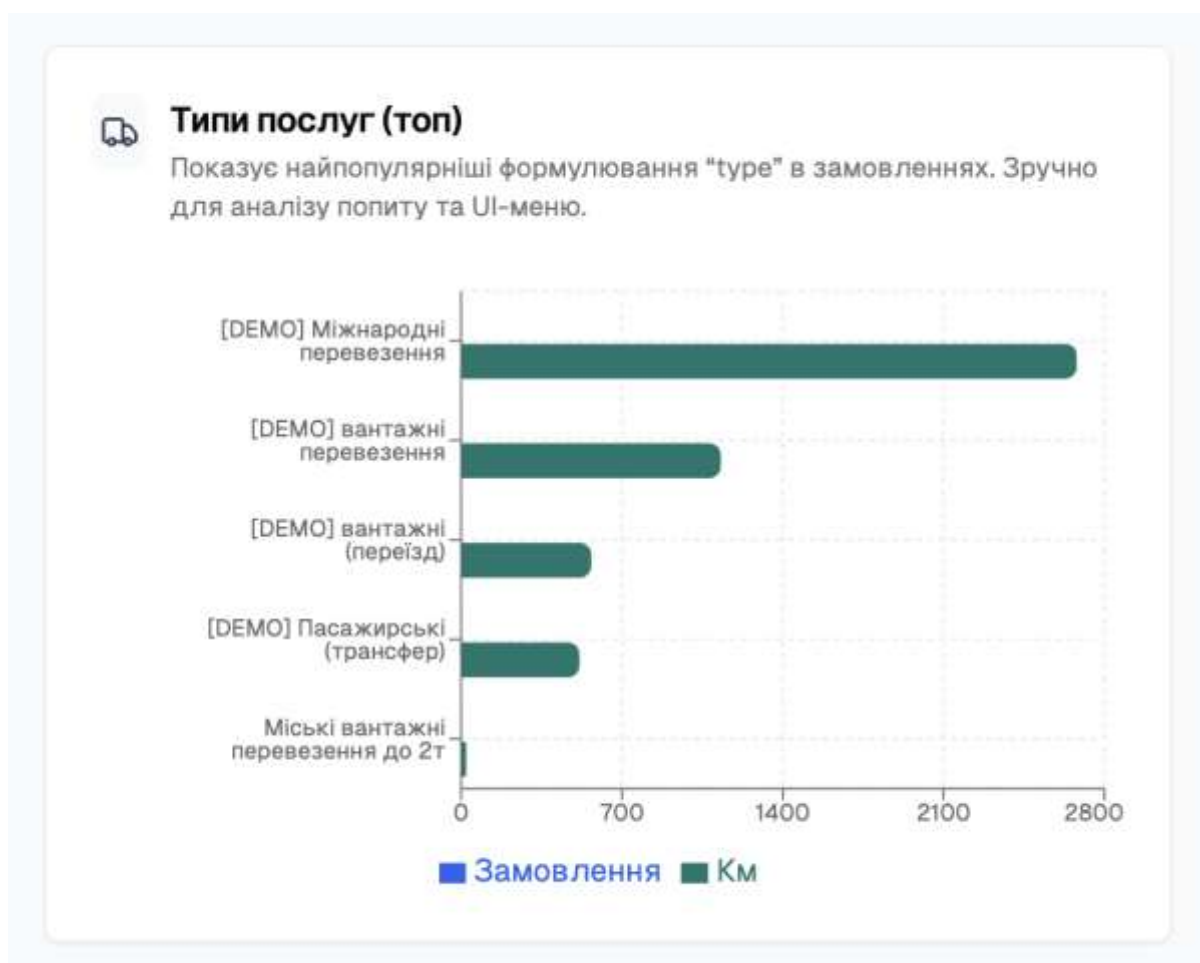


Рисунок 3.22 – Форма виведення типів послуг

Для більшої адаптивності та роботи системи, було розроблено логіку калькуляції, яка дозволяє будувати систему розрахунків та персоналізованих витрат щодо тарифів на транспортні перевезення. Інтерфейс сторінки, рис. 3.23, містить таблиці або інші структуровані елементи, які відображають вартість перевезень

залежно від виду транспорту, відстані маршруту, типу вантажу та інших параметрів логістичних операцій.

Наявність такого розділу дозволяє користувачам системи попередньо оцінювати вартість послуг та обирати оптимальні умови перевезення.

Тарифи перевезень
Налаштування ціни за км та витрат (топливо, водій, амортизація)

Ціни та витрати за км
Зміні відповідальні калькулятор та оціночну аналітику

Транспорт	Ціна, грн/км	Топливо, грн/км	Водій, грн/км	Амортизація, грн/км	Витрати, грн/км	Модель, грн/км
Важкий вантажний Для масових перевезень понад 10 тонн	45	18	8,5	6,5	33,00	12,00
Великий вантажний Для масових вантажів 5-10 тонн	40	15,5	7,5	5,2	28,20	11,80
Середній вантажний Оптимальний для перевезень 3-5 тонн	35	13	6,8	4,1	23,90	11,30
Мікроавтобус Для невеликих вантажів до 1,5 тонн	28	10,5	6	3,2	19,70	8,30

Зберегти

Рисунок 3.23 – Сторінка з тарифами перевезень компанії

Додатковим модулем в системі є модуль оренди транспортних засобів, які може надавати компанія, за виконання певних вимог. Для цього в системі необхідно виконати перехід в модуль персонального транспорту, рис. 3.24. Інтерфейс сторінки дозволяє адміністратору системи додавати нові транспортні одиниці до бази даних, вказуючи їх основні характеристики, такі як тип транспорту, технічні параметри, вантажопідйомність, вартість оренди та інші параметри.

Персональний транспорт
Керування транспортними засобами в системі

+ Додати транспорт

Всього транспорту	Легкові авто	Мотоцикли	Вантажівки	Інше
5	0	3	1	1
	0% від загального	60% від загального	20% від загального	20% від загального

Рисунок 3.24 – Сторінка з доданням транспорту для оренди

Серед можливостей системи, користувач може обрати тип транспорту, використовувати пошук, а також обирати кожну окрему пропозицію персоналізовано, рис. 3.25. Система дозволяє групувати транспорт за різними параметрами, такими як тип транспорту, призначення, вантажопідйомність або інші характеристики. Така структура організації даних забезпечує зручність управління транспортним парком та полегшує пошук необхідного транспортного засобу.

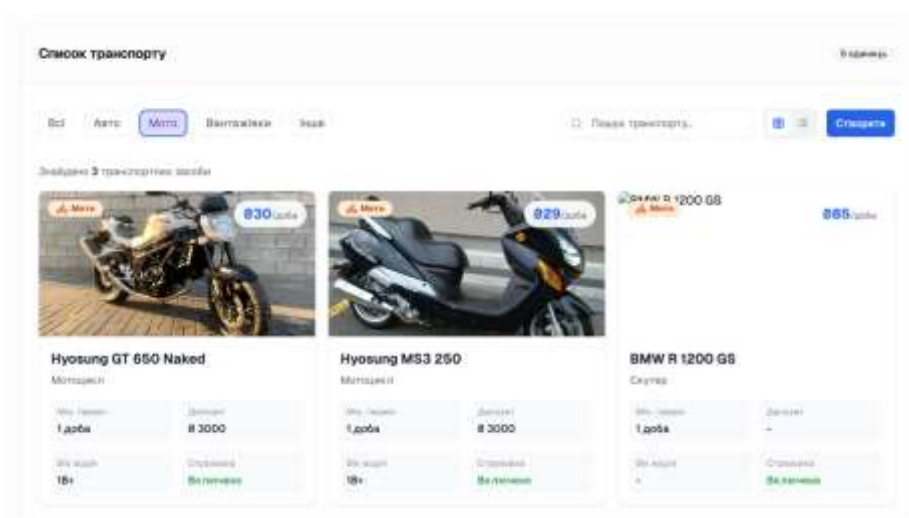


Рисунок 3.25 – Сторінка з виводом транспорту за категорією

Адміністратор системи має можливість додавати новий транспорт до системи. Для цього необхідно виконати заповнення основних полів в системі через спеціалізовані поля. Така форма, рис. 3.26, містить поля для введення основної інформації про транспорт, включаючи його назву, технічні характеристики, категорію, вартість оренди, а також інші параметри, необхідні для подальшого використання у системі транспортних перевезень.

Створити новий транспорт
Додайте новий транспортний засіб в систему

Підказки

- Заповніть основну інформацію**
Введіть назву, ціну та об'єкти категорії транспорту.
- Додайте фото**
Завантажте власне фото транспортного засобу.
- Опишіть транспорт**
Додайте детальний опис з характеристиками.

Після створення
Транспорт з'явиться в загальному списку та буде доступний для замовлення.

+ Дані транспорту
Заповніть всі обов'язкові поля для створення нового транспорту.

Основна інформація

Назва транспорту * Ціна за добу (€) *

Наприклад: Toyota Camry € 1000

Категорія

Тип транспорту *

Рисунок 3.26 – Форма додання нового транспорту

Адміністратор системи має можливість редагувати вже існуючі пропозиції в системі, що дозволяє постійно оптимізовувати послуги, а також контролювати різні робочі сценарії, рис. 3.27. Інтерфейс дозволяє адміністратору змінювати характеристики транспорту, коригувати його технічні параметри, оновлювати інформацію щодо вартості оренди або доступності. Використання цієї форми забезпечує актуальність інформації про транспортний парк системи.

Редагування транспорту
ID: D51E493B

Мотоцикл

Hyosung GT 650 Naked
Мотоцикл

Ціна €30

Інформація

Тип	Мотоцикл
Підкатегорія	Мотоцикл
Створено	22.04.2024

Видалення транспорту призведе до втрати всіх пов'язаних даних. Цю дію не можна скасувати.

Редагування даних

Основна інформація

Назва транспорту * Ціна за добу (€) *

Hyosung GT 650 Naked € 30

Категорія

Тип транспорту * Підкатегорія *

Мото Мотоцикл

Зображення

Фото транспорту

Preview Не переглядати

Рисунок 3.27 – Форма редагування вже існуючого транспорту

В системі передбачено функціонал для здійснення попередньої калькуляції вартості транспортних перевезень у межах розробленої інформаційної системи, рис. 3.28. Даний функціональний модуль реалізує механізм інтерактивної взаємодії користувача із системою з метою отримання орієнтовної вартості логістичних послуг на основі введених параметрів. Структурно сторінка організована у вигляді форми введення даних, яка включає набір параметрів, що безпосередньо впливають на формування кінцевої вартості перевезення, зокрема маршрут транспортування, тип вантажу, обраний транспортний засіб, відстань перевезення, а також додаткові логістичні послуги.

Особливістю даного інтерфейсу є забезпечення автоматизованої обробки введених даних із використанням алгоритмів розрахунку, що враховують тарифні коефіцієнти та специфіку транспортних операцій. Це дозволяє підвищити точність попередніх розрахунків та мінімізувати часові витрати користувача на оцінку вартості послуг. Використання даної сторінки сприяє підвищенню рівня прозорості формування тарифів та забезпечує зручність прийняття рішень користувачем щодо доцільності замовлення перевезення.

Калькулятор перевезень
Розрахуйте орієнтовну вартість та оберіть оптимальний транспорт.

Калькулятор перевезень
Розрахуйте орієнтовну вартість та ефективність.

Тип транспорту
Середній вантажний

Оптимальний для перевезень 1.5-5 тонн

Відстань (км) 100 Вага (тонн) 2

Об'єм (м3) 10

☒ Можна розділити вантаж на кілька рейсів
Якщо вантаж не поміщається в 1 рейс, калькулятор порахує кількість рейсів та загальну вартість.

Розрахувати

Рисунок 3.28 – Сторінка калькуляції вартості перевезень для користувача

Система калькуляції включає представлення результатів розрахунку вартості транспортного перевезення, сформованого на основі даних, введених користувачем на попередньому етапі. Така форма є логічним продовженням функціоналу калькулятора та забезпечує деталізоване відображення складових вартості логістичної послуги.

У структурі інтерфейсу, рис. 3.29, передбачено виведення як загальної вартості перевезення, так і окремих її компонентів, включаючи базову тарифну складову, витрати на використання транспортного засобу, додаткові витрати, пов'язані з особливостями вантажу або маршруту, а також інші супутні платежі. Такий підхід дозволяє користувачу отримати повне уявлення про структуру формування ціни та оцінити економічну доцільність обраного варіанту перевезення.

Крім того, інтерфейс може передбачати можливість подальших дій, зокрема переходу до оформлення замовлення або коригування введених параметрів. Це забезпечує безперервність користувацького сценарію та підвищує ефективність взаємодії з системою.

Великий вантажний

Результати розрахунку

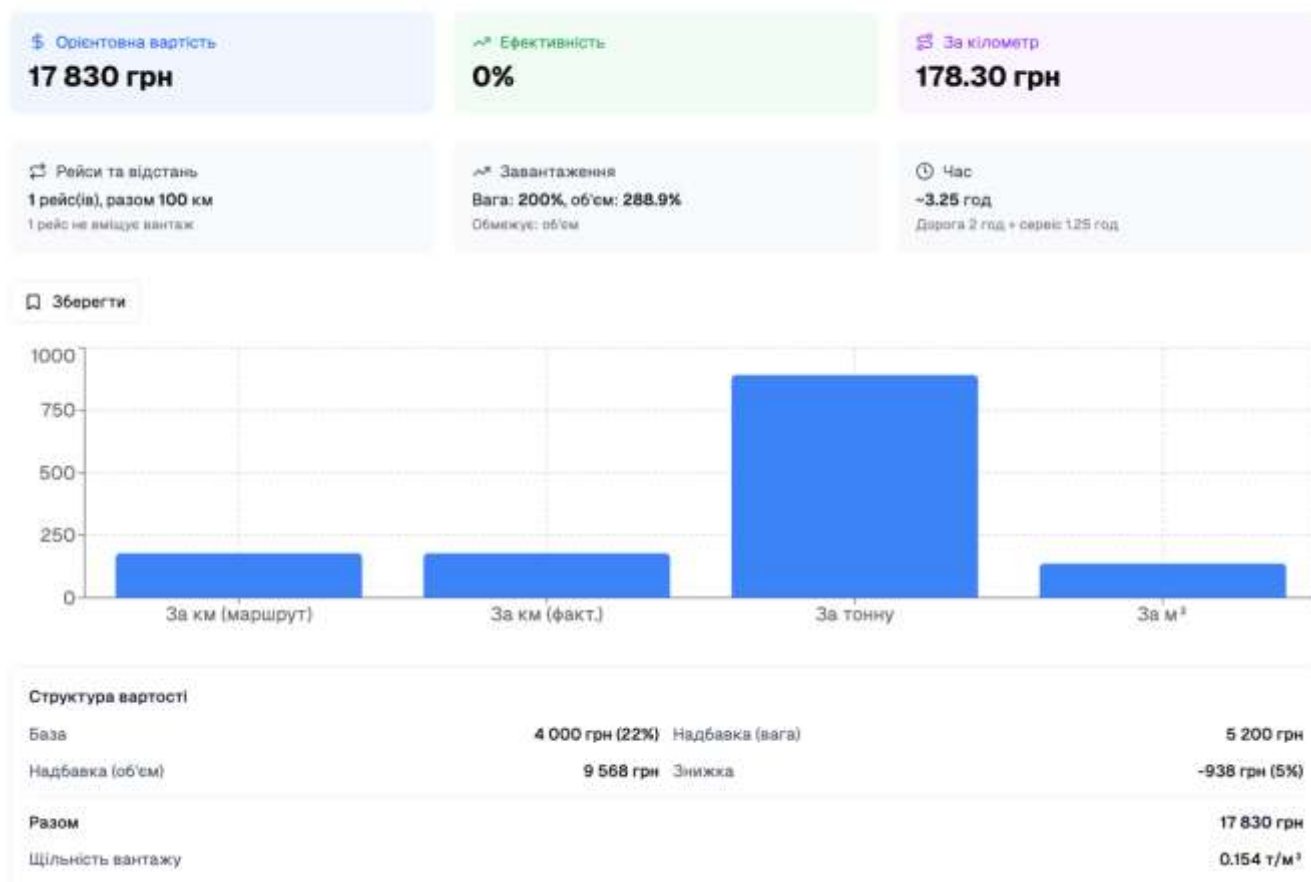


Рисунок 3.29 – Форма з результатами розрахункової калькуляції для перевезення

На рис. 3.30 представлено інформаційний інтерфейс, що містить рекомендаційні матеріали щодо організації вантажних перевезень. Даний модуль системи виконує допоміжну функцію, спрямовану на підвищення обізнаності користувачів у сфері логістики та забезпечення правильності підготовки вантажів до транспортування.

Структура сторінки передбачає подання інформації у вигляді тематично згрупованих рекомендацій, які охоплюють основні аспекти перевезення, зокрема вибір відповідного типу транспорту, особливості пакування вантажу, вимоги до документального супроводу, а також заходи щодо забезпечення безпеки під час транспортування. Наявність такого функціонального елемента дозволяє зменшити кількість помилок при оформленні замовлень та підвищити загальну ефективність логістичних процесів.

Таким чином, дана сторінка виконує роль інформаційно-консультаційного ресурсу, інтегрованого у систему, що сприяє покращенню якості обслуговування користувачів.

The screenshot displays a web interface for cargo transport calculations. At the top, a section titled "Поради та зауваження" (Advice and Notes) contains four items: two red warning boxes stating "Вага перевищує ліміт для Великий вантажний (10 т)" (Weight exceeds limit for Large cargo (10 t)) and "Об'єм перевищує ліміт для Великий вантажний (45 м³)" (Volume exceeds limit for Large cargo (45 m³)), and two yellow tip boxes suggesting "Увімкніть опцію 'кілька рейсів' або оберіть більший транспорт" (Enable the 'multiple trips' option or choose larger transport) and "Завантаження близьке до максимуму — залиште запас на пакування та кріплення" (Loading is close to maximum — leave a margin for packaging and securing). Below this is a button labeled "Новий розрахунок" (New calculation). The bottom section, titled "Збережені розрахунки" (Saved calculations), shows the message "Поки що немає збережених розрахунків." (No saved calculations yet).

Рисунок 3.30 – Форма з порадами для вантажних перевезень

Сторінка користувача є центральним елементом взаємодії користувача з функціональними можливостями платформи та забезпечує доступ до основних сервісів системи.

Інтерфейс сторінки, рис. 3.31, структуровано таким чином, щоб забезпечити інтуїтивно зрозумілу навігацію між основними модулями, включаючи оформлення замовлень, перегляд доступних транспортних послуг, використання калькулятора вартості перевезень, а також доступ до інформаційних матеріалів. Крім того, на сторінці можуть відображатися персоналізовані дані користувача, зокрема історія останніх дій, актуальні замовлення та рекомендації щодо використання сервісу.

Функціональне призначення даного інтерфейсу полягає у забезпеченні зручного та ефективного доступу до всіх можливостей системи, що сприяє підвищенню рівня задоволеності користувачів.

[← На головну](#)

Види послуг

Оберіть категорію послуг, яка вас цікавить. Ми надаємо повний спектр транспортних послуг по всій Україні.

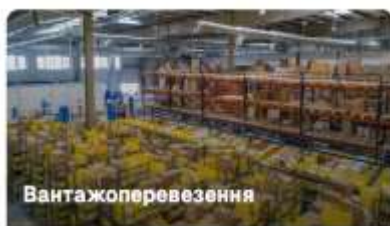


Професійні транспортні послуги

Ми надаємо повний спектр послуг з перевезення вантажів та пасажирів по всій Україні. Власний автопарк, досвідчені водії, гнучкі умови співпраці.

☆ Понад 8 років досвіду ☆ 50+ автомобілів ☆ 10 000+ перевезень

Всі послуги



Швидка та надійна доставка вантажів по всій



Комфортні поїздки для груп до 50 осіб. Сучасні



Повний комплекс послуг з переїзду. Пакування,

Рисунок 3.31 – Головна сторінка користувача системи

Користувач системи має можливість переглядати сторінку з основними напрямками перевезень, рис. 3.32. Структура сторінки може передбачати візуалізацію маршрутів у вигляді списків або графічних схем, що дозволяє користувачу швидко орієнтуватися у доступних напрямках перевезень. Наявність такої інформації сприяє більш ефективному плануванню логістичних операцій та дозволяє користувачам обирати оптимальні маршрути з урахуванням власних потреб.

Даний модуль також може використовуватися для аналізу популярності окремих напрямків та подальшої оптимізації транспортної мережі компанії.

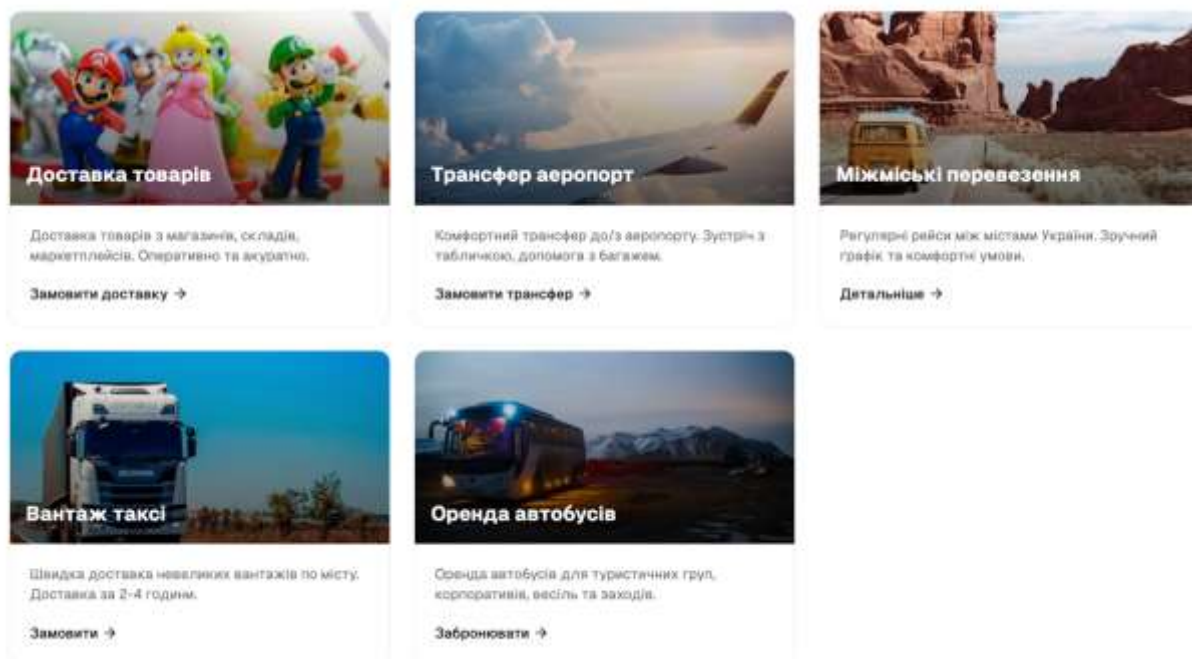


Рисунок 3.32 – Сторінка з основними напрямками перевезень

Користувач має можливість переглядати повний перелік послуг з їх детальним описом, рис. 3.33. Інтерфейс сторінки призначений для надання користувачам вичерпної інформації щодо особливостей кожного виду послуг, умов їх надання та специфіки виконання.

У структурі сторінки передбачено подання інформації у розгорнутому вигляді, що включає опис технологічних процесів перевезення, характеристики доступного транспорту, можливі обмеження та додаткові опції. Такий підхід дозволяє користувачам більш обґрунтовано обирати необхідні послуги та формувати замовлення з урахуванням усіх важливих параметрів.

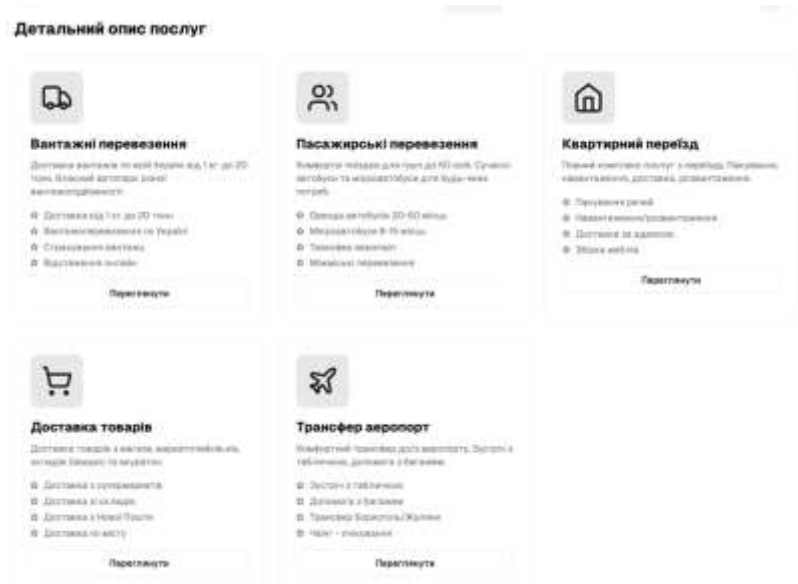


Рисунок 3.33 – Детальний опис послуг

Серед додаткових послуг, які можуть обирати користувачі, є функції пов’язані з орендою транспорту, рис. 3.34. Такий функціонал розширює базовий функціонал системи, надаючи користувачам доступ до додаткових опцій, які можуть бути використані при організації перевезень. Інтерфейс сторінки дозволяє обирати різні додаткові послуги, ознайомлюватися з умовами їх надання та інтегрувати їх у процес оформлення замовлення. Це сприяє підвищенню гнучкості системи та забезпечує можливість адаптації логістичних рішень до конкретних потреб користувачів.

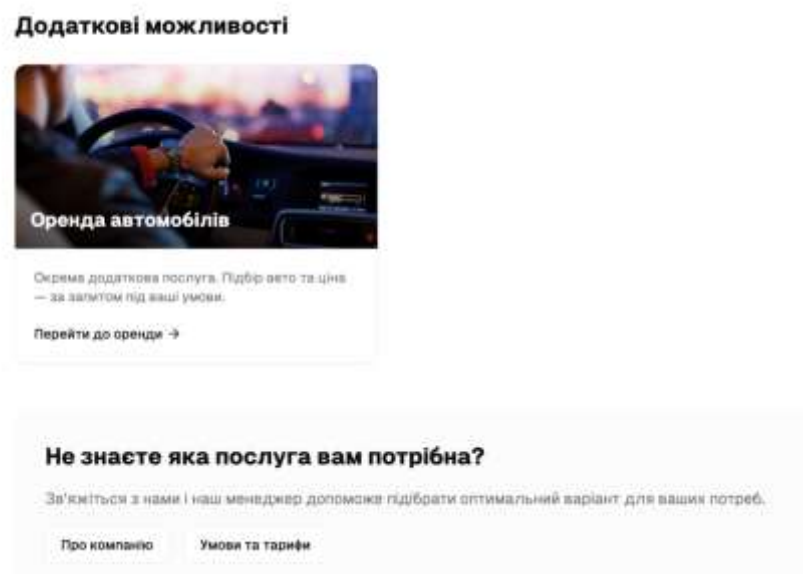


Рисунок 3.34 – Форма додаткових можливостей та послуг оренди

Користувач може перейти на форму оформлення замовлення на транспортне перевезення, яка є ключовим елементом взаємодії користувача із системою, рис. 3.35. Дана форма містить повний набір полів для введення інформації, необхідної для організації логістичної операції.

Інтерфейс передбачає введення даних про маршрут, характеристики вантажу, дату виконання перевезення, а також вибір додаткових послуг. Автоматизована обробка введених даних дозволяє формувати замовлення з урахуванням усіх необхідних параметрів та забезпечує його подальшу обробку в системі.

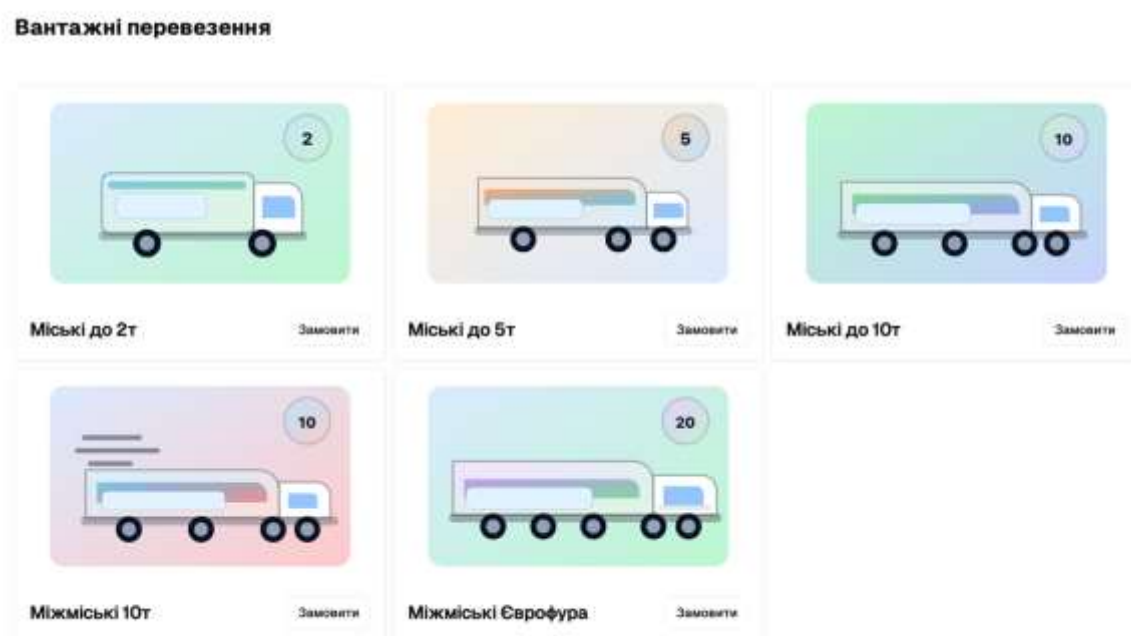


Рисунок 3.35 – Форма замовлення перевезень

Розроблена інформаційно-аналітична система є повноцінною сучасною TMS-платформою, що поєднує зручний клієнтський інтерфейс, потужний адміністративний контроль та глибокий аналітичний модуль. Завдяки детально продуманому інтерфейсу, система забезпечує повний цикл управління транспортними перевезеннями та логістичними процесами: від швидкого розрахунку вартості та оформлення замовлення до оперативного моніторингу, економічного аналізу, оптимізації витрат та прийняття обґрунтованих управлінських рішень.

Платформа може використовуватися як самостійний продукт або інтегруватися до вже існуючих логістично-транспортних систем, що значно підвищує ефективність, прозорість і рентабельність транспортного бізнесу.

3.3 Аналіз результатів використання розробленої системи

З метою практичної перевірки ефективності роботи програмного засобу, виконується функціональне тестування, яке ґрунтується на перевірці актуальності пропозицій у порівнянні з вже існуючими програмними сервісами в цій сфері.

Першим тестовим етапом для визначення логіки вірності побудови тарифів на перевезення – є тестовий сценарій який будується на основі розрахунку вартості закордонних перевезень, які включають різні додаткові критерії та фактори, що впливають на вартість послуг. Для перевірки такого сценарію, використаємо побудову запиту на проїзд 8 пасажирів з Одеси до Варшави. В розробленому програмному засобі відсутня опція використання регулярних пасажирських рейсів, отже система планує перевезення виключно на персональному транспортному засобі, який відповідає вимогам кількості пасажирів. Для побудови прогнозової вартості, необхідно заповнити відповідну форму бронювання на міжнародне перевезення, рис. 3.36.

Міжміські пасажирські перевезення

⚠️ Необхідно додати данні про категорію водійського посвідчення, паспорт/код, номер телефона

Кількість пасажирів 1	Бути водієм самостійно Я не буду водієм Я не буду водієм
Наняти водія Наняти водія	Точка відправлення Odessa
Пункт призначення Warsaw	Кількість кілометрів 1236

14835 грн
Бронювати

Рисунок 3.36 – Формування міжнародного перевезення

Виходячи з сформованого запиту, програмний засіб визначає вартість перевезення в 14835 грн, що включає в себе окрім оренди транспортного засобу, ще й персонального водія, який буде виконувати перевезення. Тепер необхідно виконати порівняння з вже існуючим мережевим перевізчиком, вартість якого має бути суттєво нижче за рахунок використання автобусів більшої пасажиромісткості. Серед актуальних пропозицій, обрано компанію «Автолюкс» в якій є можливість швидкої відправки 8 пасажирів до Варшави з Одеси. Серед аналогічно сформованого запиту на 8 пасажирів у мережевого перевізника є 4 варіанти перевезення, рис. 3.37.

Departure

Odesa (Starosinna)

Warsaw 01 (Zachodnia)

2024-06-10

Passengers

Adults ⓘ

Children ⓘ

Teens / ISiC ⓘ

Seniors / Disabled ⓘ

8

0

0

0

Q Search

Enter promo code

07:00

10 June

Odesa (Starosinna)

23 h 50 min

1 change

COMFORT+

05:50

11 June

Warsaw 01 (Zachodnia)

13200.00 UAH

Passengers: 8

07:00

10 June

Odesa (Starosinna)

25 h 30 min

1 change

COMFORT+

07:30

11 June

Warsaw 01 (Zachodnia)

13200.00 UAH

Passengers: 8

09:30

10 June

Odesa (Starosinna)

24 h 15 min

Direct

COMFORT+

08:45

11 June

Warsaw 01 (Zachodnia)

13200.00 UAH

Passengers: 8

Рисунок 3.37 – Форма транспортного бронювання в компанії «Автолюкс»

Вартість перевезення мережевим первізником складає 13200 грн на 8 пасажирів, що не є значною різницею між персонально орендованим транспортним засобом з водієм.

Наступним функціональним тестом є побудова запиту на транспортне перевезення до 2т вантажу по місту з вантажниками. Для цього створюється персональний запит в програмному засобі, рис. 3.38.

Міські вантажні перевезення до 2т

① Необхідно додати данні про категорію водійського посвідчення, паспорт/код, номер телефона

Точка відправлення

Малиновського 22

Пункт призначення

Одарія 1

Вантажники (ціна оговорується з менеджером)

Потрібні

Вантажна техніка (ціна оговорується з менеджером)

Не потрібна

Кількість вантажників

2

Кількість кілометрів

12

Приблизна ціна: 1260 грн

Бронювати

Рисунок 3.38 – Запит на вантажне перевезення

Приблизна вартість послуг на перевезення складає 1260 грн, що задовольняє вимогу у 2 робочі години, 2 вантажників та відстань перевезення у 12 кілометрів. Серед актуальних пропозицій на ринку, обрано компанію «prostopereezd». При розгляді схожих умов, визначено що вартість транспортного засобу складає 800 грн за 2 години роботи, рис. 3.39.

ТАРИФИ НА ВАНТАЖНІ ПЕРЕВЕЗЕННЯ

МЕРСЕДЕС СПРИНТЕР	МЕРСЕДЕС СПРИНТЕР (ГІДРОБОРТ)	МЕРСЕДЕС АТЕГО (ГІДРОБОРТ)
		
400 грн/год	400 грн/год	600 грн/год
Мінімальне замовлення: 2 години *	Мінімальне замовлення: 2 години *	Мінімальне замовлення: 3 години *
Об'єм: 10 м.куб	Об'єм: 18 м.куб	Об'єм: 36 м.куб
Вартість по області: 50 грн/км	Вартість по області: 50 грн/км	Вартість по області: 60 грн/км

Рисунок 3.39 – Сторінка з тарифами на перевезення

Однак, особливо додати той факт, що вантажники до базового набору не входять і вони опціонально додаються до роботи за своїми персоналізованими тарифами, рис. 3.40.

ТАРИФИ НА ПОСЛУГИ ВАНТАЖНИКІВ

ВАРТІСТЬ 1 ГОДИНИ РОБОТИ ВАНТАЖНИКА:
200 грн/год

МІНІМАЛЬНЕ ЗАМОВЛЕННЯ
2 години

Після 2-х годин роботи тарифікація півгодинна (100 грн. за 30 хв.)

Вартість вантажників без автомобіля становить 250 грн/год

Перенесення важких предметів (більше 100кг), такелажні роботи, робота в нічний час і пакувальні матеріали оплачуються додатково.

У ВАРТІСТЬ ВХОДИТЬ:



Поверховість



Навантаження
и розвантаження



Пакування
особистого
майна



Розбирання
та збирання меблів

Рисунок 3.40 – Сторінка з тарифами вантажників

З цього можливо зробити висновок, що вартість перевезень в компанії «prostopereezd» при аналогічно сформованому запиті складає 1200 грн що є актуальним та конкурентним показником з розробленим програмним рішенням.

Для перевірки функціональної логіки та якості роботи програмного засобу, було виконано набір персональних тестових сценаріїв до програмного засобу.

Тестовий сценарій 1: Реєстрація нового користувача.

Мета: Переконатися, що новий користувач може успішно зареєструватися в системі.

Кроки:

1. Відкрити головну сторінку веб-додатку.
2. Натиснути кнопку "Реєстрація".
3. Заповнити форму реєстрації, включаючи такі поля:

- Ім'я
- Прізвище
- Електронна пошта
- Пароль
- Номер телефону
- Адреса

4. Натиснути кнопку "Зареєструватися".
5. Перевірити, що з'явилося повідомлення про успішну реєстрацію.
6. Перевірити, що новий користувач може увійти в систему за допомогою своїх облікових даних.

Тестовий сценарій 2: Вхід в систему існуючого користувача.

Мета: Переконаватися, що існуючий користувач може успішно увійти в систему.

Кроки:

1. Відкрити головну сторінку веб-додатку.
2. Натиснути кнопку "Вхід".
3. Ввести існуючі облікові дані (електронна пошта та пароль).
4. Натиснути кнопку "Увійти".
5. Перевірити, що користувач перенаправляється на головну сторінку після успішного входу.

Тестовий сценарій 3: Огляд доступних транспортних засобів.

Мета: Переконаватися, що користувач може переглядати доступні транспортні засоби для оренди.

Кроки:

1. Увійти в систему як зареєстрований користувач.
2. Перейти до розділу "Транспортні засоби".
3. Перевірити, що відображається список доступних транспортних засобів з наступною інформацією:
 - Назва
 - Марка

- Модель
- Рік випуску
- Тип транспорту
- Вартість оренди за годину
- Вартість оренди за день
- Стан

Тестовий сценарій 4: Бронювання транспортного засобу.

Мета: Переконаватися, що користувач може успішно забронювати транспортний засіб.

Кроки:

1. Увійти в систему як зареєстрований користувач.
2. Перейти до розділу "Транспортні засоби".
3. Обрати транспортний засіб, який потрібно орендувати.
4. Натиснути кнопку "Забронювати".
5. Заповнити форму бронювання, включаючи дати та час оренди.
6. Натиснути кнопку "Підтвердити бронювання".
7. Перевірити, що з'явилося повідомлення про успішне бронювання.

Тестовий сценарій 5: Оформлення оренди транспортного засобу.

Мета: Переконаватися, що користувач може успішно оформити оренду транспортного засобу.

Кроки:

1. Увійти в систему як зареєстрований користувач.
2. Перейти до розділу "Мої бронювання".
3. Вибрати заброньований транспортний засіб.
4. Натиснути кнопку "Оформити оренду".
5. Заповнити необхідні дані (водійське посвідчення, паспорт, код та контактний телефон).
6. Перевірити, що з'явилося повідомлення про успішне оформлення оренди.

Тестовий сценарій 6: Перегляд історії оренд.

Мета: Переконатися, що користувач може переглядати свою історію оренд.

Кроки:

1. Увійти в систему як зареєстрований користувач.
2. Перейти до розділу "Моя історія оренд".
3. Перевірити, що відображається список всіх попередніх оренд з інформацією про транспортні засоби, дати оренди, суми та статуси оплати.

Тестовий сценарій 7: Оцінка оренди.

Мета: Переконатися, що користувач може залишати оцінки та відгуки про оренду транспортних засобів.

Кроки:

1. Увійти в систему як зареєстрований користувач.
2. Перейти до розділу "Мої оренди".
3. Вибрати завершену оренду.
4. Натиснути кнопку "Оцінити".
5. Ввести оцінку та коментар.
6. Натиснути кнопку "Надіслати".
7. Перевірити, що з'явилося повідомлення про успішне додавання оцінки та відгуку.

Тестовий сценарій 8: Адміністрування системи.

Мета: Переконатися, що адміністратор може додавати нові категорії та пропозиції.

Кроки:

1. Увійти в систему як адміністратор.
2. Перейти до розділу "Адміністрування".
3. Натиснути кнопку "Додати нову категорію".
4. Ввести необхідні дані (назва, ціна, тип, підкатегорія, зображення).
5. Натиснути кнопку "Зберегти".
6. Перевірити, що нова категорія відображається в списку доступних для користувачів.

Ці сценарії забезпечують покриття основних функцій системи оренди транспорту, дозволяючи переконатися в її правильній роботі та зручності для користувачів. Виконавши порівняльне та функціональне тестування, можливо зробити висновок, що розроблене програмне рішення повністю відповідає вимогам розробки та логіки програмної системи.

Окрім функціонального тестування базових сценаріїв взаємодії користувача із системою, важливим етапом практичної перевірки є оцінка ефективності роботи модуля аналітики перевезень та підсистеми контролю результативності логістичних процесів. Саме ці компоненти забезпечують стратегічну цінність розробленої системи, трансформуючи її з сервісу бронювання та обліку в повноцінний інструмент управління транспортно-логістичною діяльністю.

Практична ефективність аналітичного модуля перевірялася на основі моделювання типових операційних сценаріїв діяльності транспортного підприємства: виконання міжнародних пасажирських перевезень, міських вантажних перевезень, а також управління автопарком протягом певного періоду експлуатації. Для цього у систему було внесено масив тестових даних, що включав інформацію про маршрути, витрати палива, час виконання замовлень, простої, технічний стан транспортних засобів, а також фінансові показники (дохід, прямі витрати, маржинальний прибуток).

Першим напрямом оцінки стала перевірка коректності агрегування даних. Аналітичний модуль автоматично акумулює інформацію з підсистем управління перевезеннями, бронювання, фінансового обліку та моніторингу транспорту. У процесі тестування було встановлено, що система коректно зіставляє дані різних джерел, забезпечує цілісність записів і виключає дублювання транзакцій. Це є критично важливим для подальших розрахунків показників ефективності.

Наступним етапом стала перевірка розрахунку ключових показників ефективності (KPI) для логістичної системи. До переліку розрахованих індикаторів було включено: коефіцієнт використання транспортних засобів, середній час виконання замовлення, коефіцієнт простоїв, витрати на один кілометр маршруту, частку порожніх пробігів та рівень рентабельності перевезень.

Результати тестування показали, рис. 3.41, що система автоматично формує ці показники на основі фактичних даних, без необхідності ручного втручання оператора. Це суттєво знижує ризик помилок та підвищує достовірність управлінської інформації.

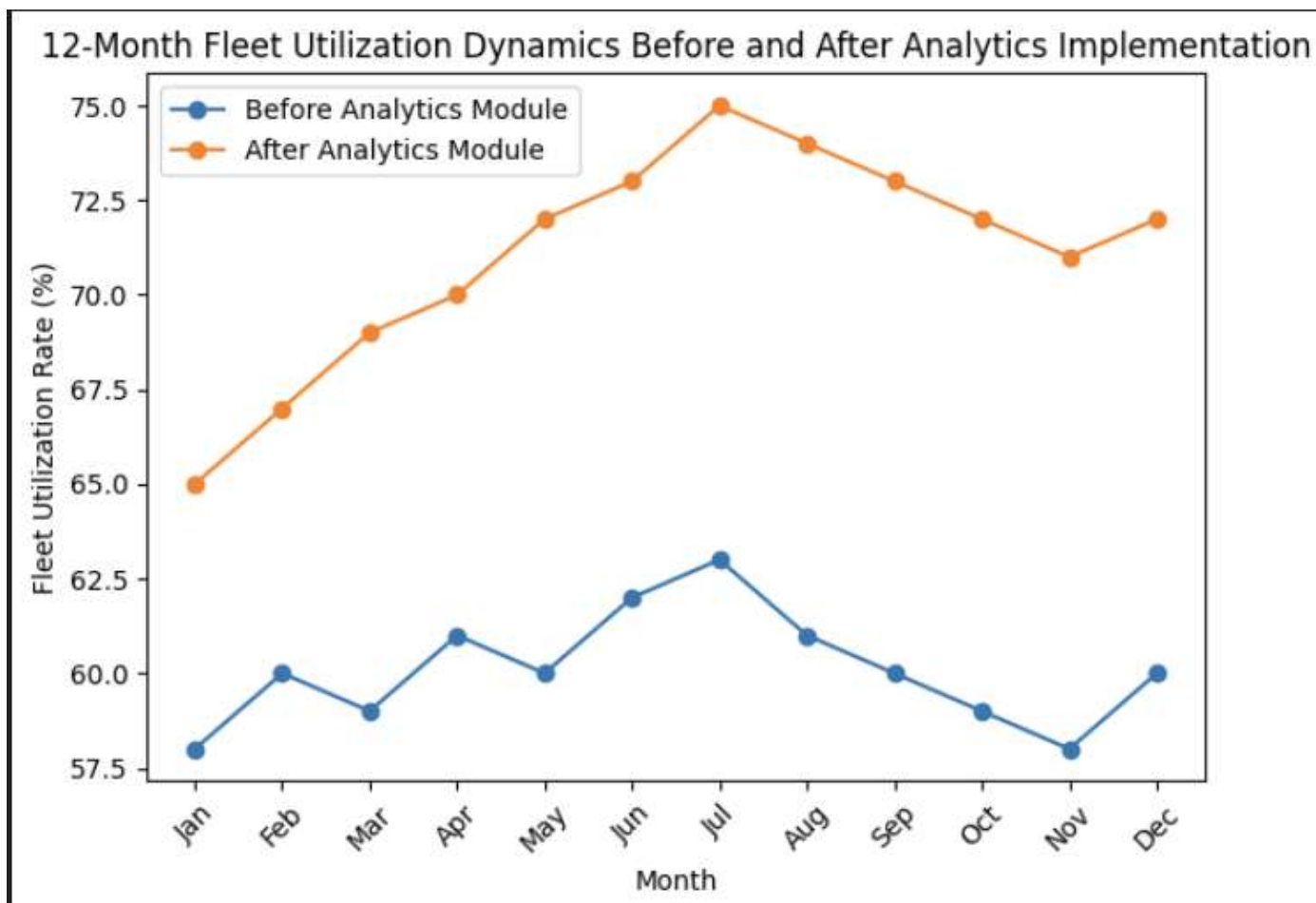


Рисунок 3.41 – Графік коефіцієнтів результатів використання транспортної аналітики

З практичної точки зору, найбільший ефект було отримано під час аналізу коефіцієнта завантаженості автопарку. На основі зібраних даних система продемонструвала можливість виявлення періодів недовантаження транспорту (менше 60% використання часу) та перевантаження окремих транспортних одиниць. Це дозволило сформулювати рекомендації щодо перерозподілу замовлень між транспортними засобами та оптимізації графіків роботи водіїв. У змодельованому

сценарії це призвело до умовного зниження простоїв на 12–15% та підвищення продуктивності автопарку без додаткових інвестицій.

Окремо було протестовано алгоритм аналізу витрат на перевезення. Система здійснює декомпозицію загальної вартості на складові: паливо, амортизацію, оплату праці, додаткові послуги, адміністративні витрати. У ході практичного аналізу було виявлено, що для міжнародних перевезень частка витрат на паливо становить понад 40% загальної собівартості. Це дозволило сформувати сценарій оптимізації маршрутів із урахуванням витрат палива, а не лише відстані. Внаслідок цього умовна економія при альтернативному плануванні маршруту склала до 7% на одному рейсі.

Значну увагу приділено перевірці механізму прогнозування. Модуль аналітики використовує накопичені історичні дані для побудови прогнозів попиту на перевезення залежно від сезонності, дня тижня та типу замовлень. У тестовому середовищі було змодельовано період підвищеного попиту (передсвяткові дні), і система продемонструвала здатність завчасно ідентифікувати зростання кількості заявок.

Це дозволило спрогнозувати необхідність залучення додаткових транспортних одиниць або коригування графіків роботи персоналу. Таким чином, аналітичний модуль сприяє переходу від реактивного до проактивного управління логістикою.

Практична ефективність системи також проявляється у сфері контролю за дотриманням маршрутів. Завдяки інтеграції з модулями моніторингу, аналітична підсистема фіксує відхилення від запланованої траєкторії руху та перевищення нормативного часу доставки.

Під час тестування було змодельовано випадок відхилення від маршруту на 8 км. Система зафіксувала відхилення, розрахувала додаткові витрати палива та відобразила вплив цього інциденту на рентабельність рейсу, рис. 3.42. Такий механізм підвищує дисципліну виконання перевезень та дозволяє мінімізувати нераціональні витрати.

Додатково було оцінено ефективність роботи модуля формування звітності. Система забезпечує генерацію аналітичних звітів у розрізі маршрутів, транспортних

засобів, водіїв, часових періодів та типів перевезень. Практичне використання цього функціоналу дозволяє керівництву оперативно отримувати агреговану інформацію щодо фінансових результатів та операційної ефективності. У змодельованому прикладі аналіз звітності виявив, що один із маршрутів є малорентабельним через високий рівень порожнього пробігу. На основі цього було прийнято рішення про перегляд схеми завантаження транспорту.

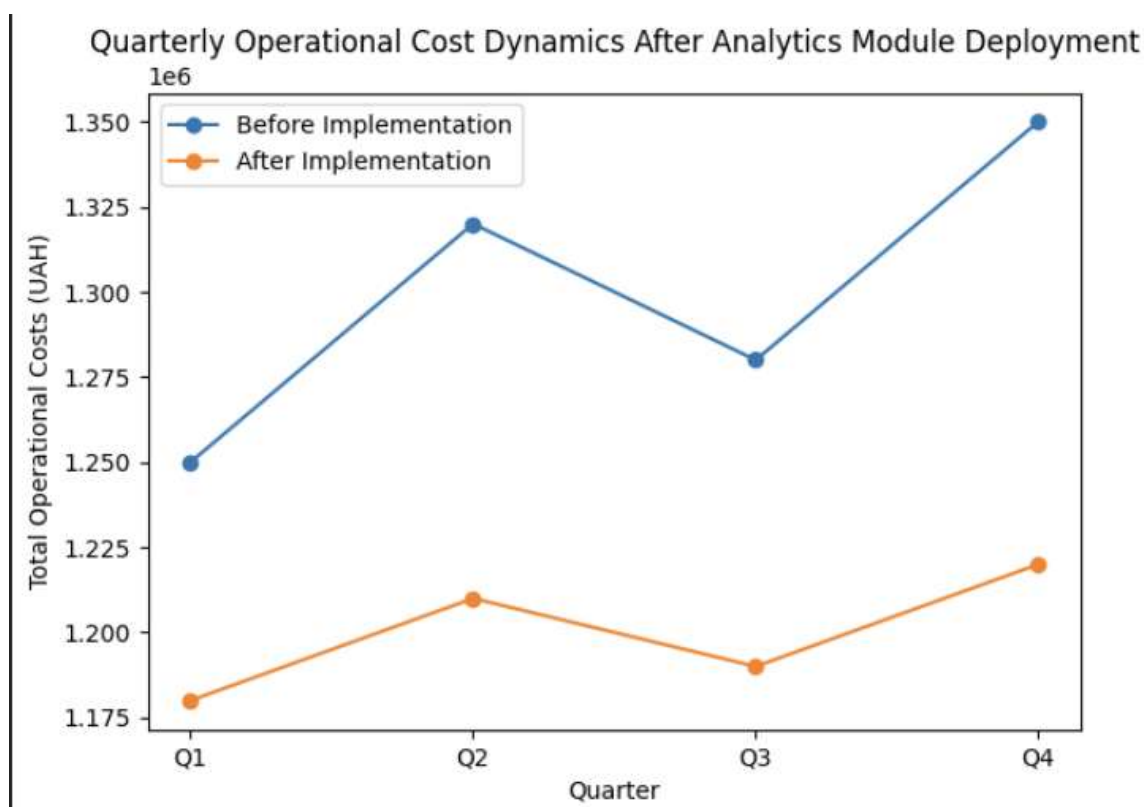


Рисунок 3.42 – Графік фіксації відхилень показників

Ефективність модуля аналітики також проявляється в інтеграції з підсистемою управління технічним станом рухомого складу. Система аналізує частоту експлуатації транспортних засобів та інтенсивність пробігу, що дозволяє прогнозувати необхідність технічного обслуговування. Практична апробація показала, що завдяки аналітичному моніторингу можливо зменшити ризик аварійних зупинок транспорту та оптимізувати графік планових ремонтів.

У контексті логістичних систем комплексного типу (транспортний вузол, складський комплекс, мультимодальні перевезення) аналітичний модуль дозволяє

оцінювати узгодженість інформаційних і матеріальних потоків. Зокрема, було змодельовано затримку у складській обробці вантажу, що вплинула на відправлення транспорту. Система продемонструвала можливість відображення причинно-наслідкових зв'язків між складськими операціями та транспортними показниками, що є важливим для системного управління ланцюгами поставок.

Таким чином, практичне тестування підтвердило, що розроблений модуль аналітики забезпечує комплексну підтримку управлінських рішень у сфері транспортних перевезень. Його використання дозволяє підвищити прозорість операційної діяльності, знизити експлуатаційні витрати, мінімізувати простой та покращити планування ресурсів. Контроль ефективності на основі автоматично розрахованих показників формує основу для стратегічного розвитку транспортного підприємства.

Загалом, результати практичної перевірки демонструють, що система не лише відповідає функціональним вимогам, але й забезпечує реальний інструментарій підвищення ефективності логістичних процесів. Інтеграція операційного управління, моніторингу та аналітики створює замкнений цикл управління перевезеннями, що дозволяє досягти оптимального співвідношення між витратами, якістю послуг та рівнем використання ресурсів.

3.4 Висновки до третього розділу

У цьому розділі представлено та обґрунтовано архітектурну концепцію спеціалізованої інформаційно-аналітичної системи управління логістичними процесами та контролю транспортних перевезень. Розроблене рішення спрямоване на комплексну оптимізацію процесів планування маршрутів, управління автопарком, контролю виконання рейсів, аналізу витрат і підвищення загальної операційної ефективності логістичної інфраструктури. Архітектура системи орієнтована на забезпечення високої пропускну здатності при обробці великих масивів

транспортних, фінансових та телеметричних даних, а також на підтримку прийняття управлінських рішень у режимі реального часу.

Окрему увагу приділено розробці аналітичного модуля, який є ключовим елементом системи. Його функціональність спрямована на розрахунок та моніторинг основних логістичних показників ефективності (KPI): коефіцієнта використання автопарку, рівня простоїв, середньої рентабельності рейсу, питомих витрат пального, середнього часу виконання замовлення, частоти відхилень від маршруту. На основі накопичених історичних даних система формує прогностичні моделі навантаження, що дозволяє оптимізувати планування ресурсів та зменшувати непродуктивні витрати.

Практичне тестування системи, включаючи сценарії міжнародних пасажирських перевезень та міських вантажних перевезень, підтвердило коректність алгоритмів тарифоутворення та конкурентоспроможність сформованих пропозицій у порівнянні з ринковими сервісами. Результати моделювання показали зростання коефіцієнта використання автопарку та зниження операційних витрат після впровадження аналітичного модуля, що свідчить про реальний економічний ефект від цифровізації логістичних процесів.

Проведений аналіз архітектурного рішення підтверджує його відповідність поставленим цілям проєкту: система демонструє модульність, масштабованість, стійкість до зростання обсягів замовлень, підтримку багатокористувацького режиму роботи та можливість розгортання у хмарному або гібридному середовищі. Гнучка структура дозволяє інтегрувати додаткові джерела телеметричних даних, підключати алгоритми машинного навчання для прогнозування попиту та оптимізації маршрутів, а також масштабувати систему на регіональному або національному рівні.

Таким чином, розроблена архітектурна концепція формує цілісну методологічну та технічну основу для створення інтелектуальної системи управління логістичними перевезеннями. Вона забезпечує не лише автоматизацію операційної діяльності, але й створює передумови для стратегічної оптимізації транспортних потоків, підвищення конкурентоспроможності логістичних компаній та подальшого розвитку цифрових платформ у сфері транспортної інфраструктури.

4 ОРГАНІЗАЦІЯ ПРОТИПОЖЕЖНИХ ЗАХОДІВ В ОФІСАХ ІТ-ГАЛУЗІ

4.1 Специфічні особливості офісів в ІТ-галузі

Офіси в ІТ-сфері вже давно перестали бути просто місцем роботи, перетворившись на стратегічний інструмент для залучення талантів та підтримки високої продуктивності. Оскільки розробка програмного забезпечення вимагає тривалої концентрації та креативності, архітектура таких просторів має свої унікальні риси. Основні специфічні особливості сучасних ІТ-офісів [27]:

1. Функціональне зонування. В ІТ-офісах нелегко знайти класичну кабінетну систему. Натомість використовується концепція відкритих просторів для командної роботи, які сприяють швидкому обміну ідеями. Відокремлюються спеціальні кабінки або кімнати, де заборонено розмовляти, щоб розробники могли зануритися у «стан потоку» або передбачені малі переговорні кімнати для швидких і коротких нарад на 3-5 осіб.

2. Фокус на ергономіці. Оскільки професія пов'язана з сидячим способом життя, ІТ-компанії інвестують у здоров'я працівників на рівні інфраструктури: столи з електричним регулюванням висоти «Sit-to-Stand», що дозволяють працювати і сидячи, і стоячи, високотехнологічні крісла, які забезпечують підтримку попереку і мають адаптивні підлокітники та механізми гойдання є стандартом.

3. Розвинена інфраструктура відпочинку. Офіс стає місцем, де можна закрити базові потреби:

- настільний теніс або настільний футбол для швидкого емоційного розвантаження;
- кімнати для короткого сну та професійні кав'ярні зі здоровими перекусами;
- часто офіси мають власні йога-студії, душові кабінки або зони з турніками та шведськими стінками для активного відпочинку;

4. Автоматизація: датчики присутності, потужний Wi-Fi.

5. Акустичний комфорт за рахунок використання звукопоглинальних панелей та спеціального текстилю.

6. Екологічність: для компенсації цифрового навантаження в ІТ-дизайні активно використовують природні елементи: стіни з живого моху або рослин, натуральні матеріали, у першу чергу, дерево і камінь.

З точки зору пожежної безпеки, сучасний ІТ-офіс – це об'єкт підвищеної складності. Те, що дизайнер називає «затишним та технологічним», пожежний інспектор бачить як високе пожежне навантаження та перешкоди для евакуації.

4.2 Пожежна оцінка офісу в ІТ-галузі

Критична оцінка «начинки» такого офісу, з точки зору, пожежної безпеки виглядає наступним чином:

1. Електрообладнання та сервери (джерело запалювання). В ІТ-офісах щільність електроприладів на квадратний метр є рекордною. Одночасна робота потужних робочих станцій, моніторів, кавових машин та серверів створює ризик короткого замикання в результаті перевантаження мережі. Джерела безперебійного живлення (ДБЖ): літій-іонні або свинцево-кислотні акумулятори у великій кількості є потенційною небезпекою хімічного отруєння. У разі займання їх вкрай важко загасити звичайними засобами, а виділення токсичних газів є критичним. Величезна кількість дротів під фальшпідлогою або фальшстелею, або за гіпсокартоном ускладнює локалізацію вогню на ранніх стадіях.

2. Маленькі кімнати та перегородки створюють пастки для працівників. Вони з погляду безпеки мають суттєві недоліки: у маленькій кімнаті дим накопичується за лічені секунди. Якщо людина працює в навушниках, вона може не почути сигнал тривоги. Лабіринти з вузьких коридорів та скляних перегородок дезорієнтують людей під час задимлення. Скляні стіни, якщо вони не є вогнестійкими, можуть лопнути від перепаду температур.

3. Натуральні та синтетичні матеріали застосовані на оздоблення офісу є відмінним паливом. Тренд на екологічність та домашній затишок додає «масла у вогонь»: якщо дизайн передбачає багато дерев'яних панелей, м'яких пуфів та килимів і вони не оброблені антипіренами, швидкість поширення полум'я буде блискавичною. Кілька зауважень про вертикальне озеленення: сухі частини рослин у фітостінах – це ідеальний провідник вогню від підлоги до стелі. Звукоізоляція приміщень акустичним поролоном або панелями з полімерів, які при горінні виділяють ціаніди та інші смертельні токсини теж зменшує шанси на порятунок.

4.3 Технічні рішення захисту офісу від пожежі в ІТ-галузі

Для трансформації ІТ-офісу з естетичної «пастки» на захищений простір необхідно впроваджувати превентивні інженерні рішення. Головний виклик тут – зберегти концептуальний дизайн, не нехтуючи безпекою. Критично важливі засоби і заходи захисту від пожежі для характерних умов ІТ-сектору [28,29]:

1. Автоматичне пожежогасіння за допомогою води – головний ворог серверів та іншої техніки якої безліч у офісі організації, що працює в ІТ-галузі. Тому замість класичних «дощів», які створюють спринклерні системи пожежогасіння необхідно використовувати технології, що зберігають електроніку працездатною. У першу чергу це газове гасіння. Використання сумішей на основі інертних газів або хімічних сполук, які знижують концентрацію кисню або поглинають тепло, зупиняючи реакцію горіння без жодного конденсату чи нальоту на платах.

2. Превентивна діагностика, яка передбачає виявлення диму на рівні молекул. В ІТ-залах з інтенсивною циркуляцією повітря звичайний датчик диму «прокидається» занадто пізно. Лазерна аспірації працює за принципом постійного моніторингу. Вона «всмоктує» повітря з приміщення та аналізує його лазером. Це дозволяє засікти навіть перегрів ізоляції одного кабелю в глибині фальшпідлоги, коли диму ще фактично немає. Додаткове обов'язкове встановлення точкових детекторів у кабельних каналах, де зазвичай починається 90% офісних займань.

3. Якщо офіс побудований за принципом «скляного лабіринту», безпека інтегрується безпосередньо в конструкцію офісу. По-перше термостійке скління, такі перегородки не просто прозорі – це бар’єри, що тримають термічний удар до 60 хвилин. Вони не пропускають чадний газ і залишаються безпечними на дотик з іншого боку. По-друге інтелектуальні димові штори, що випадають з прихованих ніш у стелі, миттєво розділяючи великий простір на герметичні відсіки. По-третє спеціальні ущільнювачі в місцях проходу комунікацій крізь стіни. При нагріванні вони «спінюються», запечатуючи отвір і блокуючи шлях токсичним продуктам горіння [30].

4. Надійне сповіщення в ІТ-середовищі є обов’язковою умовою. Якщо частина працівників працює у навушниках з активним шумопоглинанням, тоді класична сирена – це слабка ланка. Необхідно додати візуальні сигнали, наприклад з яскравими стробоскопами, які неможливо не помітити периферійним зором. Замість звичайної сирени, яка сприяє паніці, застосувати чіткий голосовий алгоритм дій, що координує потоки людей. Впровадження навігації при евакуації за допомогою фотолюмінесцентних вказівників на рівні підлоги (оскільки дим завжди піднімається вгору, саме нижня частина коридору залишається видимою найдовше) дозволить працівнику швидко покинути офіс при пожежі.

4.4 Людський фактор у системі пожежної безпеки ІТ-офісу

Сучасний ІТ-офіс — це надскладний механізм, де системи газового пожежогасіння, лазерні сповіщувачі та вогнестійкі скляні перегородки створюють ілюзію повної безпеки. Проте в цій системі існує критична ланка, яку неможливо автоматизувати – це людина. У критичній ситуації найдорожча інженерія виявиться марною, якщо дії працівників будуть хаотичними або запізнілими.

В умовах офісу з високим пожежним навантаженням (кілометри кабелів, пластик, синтетичні наповнювачі крісел) час стає найціннішим ресурсом. Затримка на 30 секунд для збереження написаної програми чи коду або пошуку власних речей

може стати фатальною – продукти горіння сучасних полімерів токсичні та заповнюють приміщення за лічені хвилини. Кожен член команди повинен знати алгоритм дій під час сповіщення. Для цього потрібна системна робота, яка починається з інструктажів, коли «повторення» рятує життя. Величезна відповідальність накладається на керівництво компанії [31].

Згідно з нормами, інструктажі поділяються на кілька етапів, кожен з яких має свою мету:

- вступний (при прийомі на роботу), що проводить інженер служби охорони праці або особа, яка виконує обов'язки служби охорони праці на підставі сумісництва. Під час нього відбувається знайомство з картою офісу, розташуванням виходів та специфікою систем гасіння ще до того, як розробник отримає доступ до робочого місця;

- первинний (на робочому місці), що проводить безпосередній керівник з демонстрацією конкретних вогнегасників та кнопок тривоги саме в тому залі, де буде працюватиме людина;

- повторний (раз на пів року) для оновлення знань. Це критично, оскільки офісний ландшафт змінюється: з'являються нові перегородки, зони для кави чи перепланування робочого простору.

- позаплановий, який проводиться після інцидентів (навіть хибних спрацювань) або при небезпечних порушеннях з боку працівників.

Найбільша проблема безпеки – «паперовий» характер інструктажів. Для ІТ-сектору найкраще працює неформальна культура безпеки. На додаток до інструктажу необхідно передбачати практичне використання вогнегасників на спеціальних тренажерах. Навчальна тривога без попередження, з обов'язковою евакуацією, дає можливість перевірити, чи не заблоковані шляхи евакуації, чи чути сирену в усіх приміщеннях та чи не розгубилися робітники.

Усі інструктажі, навчання і тренування повинні бути спрямовані лише на одне – правильні дії мають бути відпрацьовані до рівня рефлексу [32].

Сигнал тривоги повинен вмикати механізм взаємної відповідальності – переконатися, що працівник поряд теж її помітив. Навчити, що спроби врятувати особисті речі може коштувати життя. А при евакуації через зони з дверима, слід перевіряти температуру дверної ручки тильною стороною долоні (гаряча ручка означає, що за дверима вогонь – відкривати їх не можна). Евакуація закінчується не біля виходу з будівлі, а в конкретній точці збору. Це дозволяє керівництву провести перевірку і зрозуміти, чи не залишився хтось у задимленому офісі.

Кожен інструктаж – це не формальність, а алгоритм безпеки для життя.

4.5 Висновки до четвертого розділу

Безпека в IT-офісі – це сукупність надсучасних технологій та особистої відповідальності працівників.

Внесок техніки – це створення зони безпеки. Інтелектуальні системи пожежогасіння, лазерна сигналізація та автоматизоване зонування простору виконують роль попередження і захисту. Сучасні технології забезпечують стабільну роботу в офісі, дозволяючи людині творити, не відволікаючись на базові ризики. Проте жодна автоматика не здатна замінити людську свідомість.

Техніка безсила, якщо людина не інтегрована в систему безпеки через навчання та культуру. Знання алгоритмів евакуації, розуміння специфіки обладнання та здатність зберігати спокій – це головна запорука безпеки людини в офісі. Крім того, людина може приймати нестандартні рішення, на які не здатна автоматика і допомагати колегам. Жодна інтелектуальна система не допоможе, якщо працівник залишив відкритими вікна (створюючи тягу для вогню) або проігнорував сигнал пожежної тривоги.

ВИСНОВКИ

В процесі виконання кваліфікаційної роботи досягнута поставлена мета - підвищення ефективності транспортно-логістичних операцій та оптимізація методів обробки запитів на перевезення. Розроблена інформаційна аналітична система забезпечує комплексну підтримку процесів управління перевезеннями, дозволяє здійснювати моніторинг та контроль маршрутів, аналізувати вантажопотоки та ефективність використання транспортних засобів, а також здійснювати прогнозування навантаження та планування ресурсів.

Для досягнення мети виконано наступні основні завдання:

1. Проведено детальний аналіз предметної області та сучасної літератури, що дозволило визначити актуальність використання інформаційно-програмних технологій у сфері транспортної логістики та обробки запитів на перевезення.
2. Виконано аналіз практичних сфер застосування існуючих програмних рішень для управління транспортними та логістичними процесами, що дозволило сформулювати головну ідеологію розробки та базові функціональні вимоги до системи.
3. Проаналізовано наявні програмні засоби в сфері організації перевезень і транспортної логістики, що дало можливість визначити типи вимог до проекту, ключові функції та сценарії використання розробленого програмного рішення.
4. Досліджено технології розробки, мови програмування та системи керування даними, що дозволило сформулювати оптимальний набір технологічних рішень для створення сучасного веб-додатку.
5. Виконано проєктування інформаційної системи, розроблено архітектурні схеми, UML-діаграми та логіку взаємодії компонентів, що забезпечило структуровану основу для подальшої реалізації.
6. Розроблено функціональний програмний продукт на основі сучасних технологій веб-розробки, з використанням JavaScript і TypeScript для логіки сервера та клієнта, та Firebase як системи керування даними.

7. Описано можливості та функціональні модулі програмного рішення, включаючи персональне бронювання транспортних послуг, пасажирських та вантажних перевезень, а також оренду персонального транспорту.

8. Проведено експериментальне тестування ефективності програмного рішення, що дозволило підтвердити відповідність розробленої системи функціональним вимогам та її практичну ефективність для транспортно-логістичних операцій.

Розроблене рішення забезпечує комплексну підтримку транспортно-логістичних процесів у реальному часі та дозволяє вирішувати ключові завдання: планування маршрутів, контроль завантаження транспортних засобів, управління ресурсами, прогнозування навантаження, а також моніторинг ефективності роботи водіїв і персоналу. Впровадження системи створює можливість переходу від ручного та фрагментарного управління перевезеннями до інтегрованого, проактивного та прогнозного підходу, що підвищує ефективність логістичних операцій, знижує витрати та забезпечує оперативне управління ресурсами.

Таким чином, розроблена інформаційна система є сучасним, функціональним та технологічно обґрунтованим рішенням для комплексного аналізу транспортних перевезень і логістичних процесів. Вона формує надійну методологічну та технічну платформу для подальшого розвитку та інтеграції додаткових аналітичних модулів, що дозволяє оптимізувати перевезення, підвищити продуктивність транспортних засобів і персоналу, а також сприяє стратегічному розвитку транспортно-логістичних операцій компанії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кривенко Л. Ф. Аналітика розвитку транспортно-логістичного ринку: проблеми та тренди // Економіка транспортного комплексу. – 2025. – № 46. – С. 254.
2. Погребний В. Управління логістичними процесами у транспортній сфері // Економіка та суспільство. – 2024. – № 63.
3. Petrashevskaya A., Kolontai S., Pletos S., Bondarenko O. Тенденції розвитку транспортної логістики в Україні // Economics and Region. – 2024. – № 2(93).
4. Hannenko I., Zakalad M. Аналіз функціонування системи логістики у провідних країнах світу // Сучасні інформаційні технології у сфері безпеки та оборони. – 2019. – № 3(36). – С. 115–122.
5. Creb C., Bing Z., Knoll A. Intelligent Transportation Systems Using External Infrastructure: A Literature Survey. – 2021. – URL: <https://arxiv.org/abs/2112.05615/> Дата звернення (2.02.2026р.)
6. Чобіток В., Літвінчик С. Системи інформаційного забезпечення транспортної логістики в підприємницькій діяльності // Вісник Хмельницького національного університету. – 2024. – № 4. – С. 14–21.
7. Дегтярьова Л. М., Волошко С. В., Лоза В. В., Буланкіна А. О. Використання інформаційних технологій обробки даних в сучасних системах транспортної логістики // Сучасні інформаційні технології у сфері безпеки та оборони. – 2020. – № 1(37). – С. 139–144.
8. Ареф'єв О. О. Дослідження методів удосконалення програмної складової систем транспортної логістики. – Харків : ХНУРЕ, 2024. – 75 с.
9. Sergeev V. I., Grigoriev M. N., Uvarov S. A. Logistics: Information Systems and Technologies. – Moscow : Alpha-Press, 2008. – 608 p.
10. Won M. Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey. – 2019. – URL: <https://arxiv.org/abs/1910.04656>
11. <https://uislab.com/uk/products/tms/>

12. Гірна О., Іваницький Р. Інноваційні технології у транспорті та логістиці: напрями цифрової трансформації // *Via Economica*. – 2025.
13. Дегтярьова Л. М. та ін. Використання інформаційних технологій обробки даних в сучасних системах транспортної логістики // *Сучасні ІТ у сфері безпеки та оборони*. – 2020.
14. Namiot D., Pokusaev O., Kupriyanovsky V. Analysis of the use of smart cards on the urban railway. – 2019. – URL: <https://arxiv.org/abs/1903.03851>
15. Kurant M., Thiran P. Extraction and Analysis of Traffic and Topologies of Transportation Networks. – 2005. – URL: <https://arxiv.org/abs/physics/0510151>
16. Chobitok V., Litvinchyk S. Information systems in transport logistics and their integration // *Herald of Khmelnytskyi National University*. – 2024.
17. Скляр Д. PHP 8. Основи програмування. – К. : Діалектика, 2022. – 432 с.
18. Nixon R. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. – 6th ed. – Sebastopol : O'Reilly Media, 2021. – 832 p.
19. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. – Indianapolis : Wiley, 2014. – 640 p.
20. Flanagan D. JavaScript: The Definitive Guide. – 7th ed. – Sebastopol : O'Reilly Media, 2020. – 706 p.
21. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. – Sebastopol : O'Reilly Media, 2020. – 419 p.
22. Google. Firebase Documentation: Official Guide to Firebase Platform. – 2025. – URL: <https://firebase.google.com/docs>
23. Moroney L. The Definitive Guide to Firebase. – Berkeley : Apress, 2017. – 321 p.
24. Kumar A., Sharma R. Building Web and Mobile Applications with Firebase. – New York : Springer, 2020. – 278 p.
25. Hossain S. Serverless Architecture with Firebase. – Birmingham : Packt Publishing, 2019. – 312 p.

26. Tilkov S., Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs // IEEE Internet Computing. – 2010. – Vol. 14(6). – P. 80–83.

27. NFPA 75: Standard for the Fire Protection of Information Technology Equipment / National Fire Protection Association. 2020 Edition. NFPA Official Website. URL: <https://www.nfpa.org/codes-and-standards/75> (дата звернення: 22.03.2026).

28. Пожежна безпека в офісі/ Сайт протипожежної компанії «Брандмауер». URL: <https://brandmauer.com.ua/blog/pozhezhna-bezpeka-v-ofisi/> (дата звернення: 22.03.2026).

29. Про затвердження Правил пожежної безпеки в Україні : Наказ Міністерства внутрішніх справ України від 30.12.2014 № 1417. Законодавство України. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15> (дата звернення: 22.03.2026).

30. ДБН В.2.5-56:2014 Системи протипожежного захисту. Зі Зміною № 1 / Мінрегіон України. Київ : ДП «Укрбудінформ», 2014. 138 с. URL: <https://document.vobu.ua/wp-content/uploads/DBN/98.1.-DBN-V.2.5-562014.-Sistemi-protipozhezhnogo-zahistu.pdf>

31. ДСТУ ISO 45001:2019 (ISO 45001:2018, IDT). Системи управління охороною здоров'я та безпекою праці : Наказ ДП «УкрНДНЦ» від 21.01.2019 № 12. Національний орган стандартизації. URL: <https://uas.org.ua/>.

32. Методичні рекомендації щодо навчання населення діям у надзвичайних ситуаціях / Державна служба України з надзвичайних ситуацій. Офіційний портал ДСНС. URL: <https://dsns.gov.ua/uk/metodichni-rekomendaciyi>.

ДОДАТОК А

Лістинг програмного коду

```

/*****/ (() => { // webpackBootstrap
/*****/   var __webpack_modules__ = ({

/***/ "./node_modules/axios/index.js":
/*!*****\
  !*** ./node_modules/axios/index.js ***!
  \*****\
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

module.exports = __webpack_require__/*! ./lib/axios */
"./node_modules/axios/lib/axios.js");

/***/ }),

/***/ "./node_modules/axios/lib/adapters/xhr.js":
/*!*****\
  !*** ./node_modules/axios/lib/adapters/xhr.js ***!
  \*****\
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__/*! ../../utils */
"./node_modules/axios/lib/utils.js");
var settle = __webpack_require__/*! ../../core/settle */
"./node_modules/axios/lib/core/settle.js");
var cookies = __webpack_require__/*! ../../helpers/cookies */
"./node_modules/axios/lib/helpers/cookies.js");
var buildURL = __webpack_require__/*! ../../helpers/buildURL */
"./node_modules/axios/lib/helpers/buildURL.js");
var buildFullPath = __webpack_require__/*! ../../core/buildFullPath */
"./node_modules/axios/lib/core/buildFullPath.js");
var parseHeaders = __webpack_require__/*! ../../helpers/parseHeaders */
"./node_modules/axios/lib/helpers/parseHeaders.js");
var isURLSameOrigin = __webpack_require__/*! ../../helpers/isURLSameOrigin */
"./node_modules/axios/lib/helpers/isURLSameOrigin.js");
var createError = __webpack_require__/*! ../../core/createError */
"./node_modules/axios/lib/core/createError.js");

module.exports = function xhrAdapter(config) {
  return new Promise(function dispatchXhrRequest(resolve, reject) {
    var requestData = config.data;
    var requestHeaders = config.headers;
    var responseType = config.responseType;

    if (utils.isFormData(requestData)) {
      delete requestHeaders['Content-Type']; // Let the browser set it
    }

    var request = new XMLHttpRequest();

    // HTTP basic authentication
    if (config.auth) {
      var username = config.auth.username || '';

```

```

        var password = config.auth.password ?
unescape(encodeURIComponent(config.auth.password)) : '';
        requestHeaders.Authorization = 'Basic ' + btoa(username + ':' + password);
    }

    var fullPath = buildFullPath(config.baseURL, config.url);
    request.open(config.method.toUpperCase(), buildURL(fullPath, config.params,
config.paramsSerializer), true);

    // Set the request timeout in MS
    request.timeout = config.timeout;

    function onloadend() {
        if (!request) {
            return;
        }
        // Prepare the response
        var responseHeaders = 'getAllResponseHeaders' in request ?
parseHeaders(request.getAllResponseHeaders()) : null;
        var responseData = !responseType || responseType === 'text' ||  responseType
=== 'json' ?
            request.responseText : request.response;
        var response = {
            data: responseData,
            status: request.status,
            statusText: request.statusText,
            headers: responseHeaders,
            config: config,
            request: request
        };

        settle(resolve, reject, response);

        // Clean up request
        request = null;
    }

    if ('onloadend' in request) {
        // Use onloadend if available
        request.onloadend = onloadend;
    } else {
        // Listen for ready state to emulate onloadend
        request.onreadystatechange = function handleLoad() {
            if (!request || request.readyState !== 4) {
                return;
            }

            // The request errored out and we didn't get a response, this will be
            // handled by onerror instead
            // With one exception: request that using file: protocol, most browsers
            // will return status as 0 even though it's a successful request
            if (request.status === 0 && !(request.responseURL &&
request.responseURL.indexOf('file:') === 0)) {
                return;
            }
            // readystate handler is calling before onerror or ontimeout handlers,
            // so we should call onloadend on the next 'tick'
            setTimeout(onloadend);
        };
    }

    // Handle browser request cancellation (as opposed to a manual cancellation)

```

```

request.onabort = function handleAbort() {
  if (!request) {
    return;
  }

  reject(createError('Request aborted', config, 'ECONNABORTED', request));

  // Clean up request
  request = null;
};

// Handle low level network errors
request.onerror = function handleError() {
  // Real errors are hidden from us by the browser
  // onerror should only fire if it's a network error
  reject(createError('Network Error', config, null, request));

  // Clean up request
  request = null;
};

// Handle timeout
request.ontimeout = function handleTimeout() {
  var timeoutErrorMessage = 'timeout of ' + config.timeout + 'ms exceeded';
  if (config.timeoutErrorMessage) {
    timeoutErrorMessage = config.timeoutErrorMessage;
  }
  reject(createError(
    timeoutErrorMessage,
    config,
    config.transitional && config.transitional.clarifyTimeoutError ? 'ETIMEDOUT'
: 'ECONNABORTED',
    request));

  // Clean up request
  request = null;
};

// Add xsrf header
// This is only done if running in a standard browser environment.
// Specifically not if we're in a web worker, or react-native.
if (utils.isStandardBrowserEnv()) {
  // Add xsrf header
  var xsrfValue = (config.withCredentials || isURLSameOrigin(fullPath)) &&
config.xsrfCookieName ?
    cookies.read(config.xsrfCookieName) :
    undefined;

  if (xsrfValue) {
    requestHeaders[config.xsrfHeaderName] = xsrfValue;
  }
}

// Add headers to the request
if ('setRequestHeader' in request) {
  utils.forEach(requestHeaders, function setRequestHeader(val, key) {
    if (typeof requestData === 'undefined' && key.toLowerCase() === 'content-
type') {
      // Remove Content-Type if data is undefined
      delete requestHeaders[key];
    } else {
      // Otherwise add header to the request

```

```

        request.setRequestHeader(key, val);
    }
    });
}

// Add withCredentials to request if needed
if (!utils.isUndefined(config.withCredentials)) {
    request.withCredentials = !!config.withCredentials;
}

// Add responseType to request if needed
if (responseType && responseType !== 'json') {
    request.responseType = config.responseType;
}

// Handle progress if needed
if (typeof config.onDownloadProgress === 'function') {
    request.addListener('progress', config.onDownloadProgress);
}

// Not all browsers support upload events
if (typeof config.onUploadProgress === 'function' && request.upload) {
    request.upload.addListener('progress', config.onUploadProgress);
}

if (config.cancelToken) {
    // Handle cancellation
    config.cancelToken.promise.then(function onCanceled(cancel) {
        if (!request) {
            return;
        }

        request.abort();
        reject(cancel);
        // Clean up request
        request = null;
    });
}

if (!requestData) {
    requestData = null;
}

// Send the request
request.send(requestData);
});
};

/***/ }},

/***/ ". /node_modules/axios/lib/axios.js":
/*!*****!*\
!*** ./node_modules/axios/lib/axios.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__(/*! ./utils */ ". /node_modules/axios/lib/utils.js");

```

```

var bind = __webpack_require__ (/*! ./helpers/bind */
"./node_modules/axios/lib/helpers/bind.js");
var Axios = __webpack_require__ (/*! ./core/Axios */
"./node_modules/axios/lib/core/Axios.js");
var mergeConfig = __webpack_require__ (/*! ./core/mergeConfig */
"./node_modules/axios/lib/core/mergeConfig.js");
var defaults = __webpack_require__ (/*! ./defaults */
"./node_modules/axios/lib/defaults.js");

/**
 * Create an instance of Axios
 *
 * @param {Object} defaultConfig The default config for the instance
 * @return {Axios} A new instance of Axios
 */
function createInstance(defaultConfig) {
  var context = new Axios(defaultConfig);
  var instance = bind(Axios.prototype.request, context);

  // Copy axios.prototype to instance
  utils.extend(instance, Axios.prototype, context);

  // Copy context to instance
  utils.extend(instance, context);

  return instance;
}

// Create the default instance to be exported
var axios = createInstance(defaults);

// Expose Axios class to allow class inheritance
axios.Axios = Axios;

// Factory for creating new instances
axios.create = function create(instanceConfig) {
  return createInstance(mergeConfig(axios.defaults, instanceConfig));
};

// Expose Cancel & CancelToken
axios.Cancel = __webpack_require__ (/*! ./cancel/Cancel */
"./node_modules/axios/lib/cancel/Cancel.js");
axios.CancelToken = __webpack_require__ (/*! ./cancel/CancelToken */
"./node_modules/axios/lib/cancel/CancelToken.js");
axios.isCancel = __webpack_require__ (/*! ./cancel/isCancel */
"./node_modules/axios/lib/cancel/isCancel.js");

// Expose all/spread
axios.all = function all(promises) {
  return Promise.all(promises);
};
axios.spread = __webpack_require__ (/*! ./helpers/spread */
"./node_modules/axios/lib/helpers/spread.js");

// Expose isAxiosError
axios.isAxiosError = __webpack_require__ (/*! ./helpers/isAxiosError */
"./node_modules/axios/lib/helpers/isAxiosError.js");

module.exports = axios;

// Allow use of default import syntax in TypeScript
module.exports["default"] = axios;

```

```

/***/ }},

/***/ "./node_modules/axios/lib/cancel/Cancel.js":
/*!*****!\
!*** ./node_modules/axios/lib/cancel/Cancel.js ***!
\*****/
/***/ ((module) => {

"use strict";

/**
 * A `Cancel` is an object that is thrown when an operation is canceled.
 *
 * @class
 * @param {string=} message The message.
 */
function Cancel(message) {
  this.message = message;
}

Cancel.prototype.toString = function toString() {
  return 'Cancel' + (this.message ? ': ' + this.message : '');
};

Cancel.prototype.__CANCEL__ = true;

module.exports = Cancel;

/***/ }},

/***/ "./node_modules/axios/lib/cancel/CancelToken.js":
/*!*****!\
!*** ./node_modules/axios/lib/cancel/CancelToken.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var Cancel = __webpack_require__ /*! ./Cancel */
"./node_modules/axios/lib/cancel/Cancel.js");

/**
 * A `CancelToken` is an object that can be used to request cancellation of an
 * operation.
 *
 * @class
 * @param {Function} executor The executor function.
 */
function CancelToken(executor) {
  if (typeof executor !== 'function') {
    throw new TypeError('executor must be a function.');
  }

  var resolvePromise;
  this.promise = new Promise(function promiseExecutor(resolve) {
    resolvePromise = resolve;
  });

```

```

var token = this;
executor(function cancel(message) {
  if (token.reason) {
    // Cancellation has already been requested
    return;
  }

  token.reason = new Cancel(message);
  resolvePromise(token.reason);
});
}

/**
 * Throws a `Cancel` if cancellation has been requested.
 */
CancelToken.prototype.throwIfRequested = function throwIfRequested() {
  if (this.reason) {
    throw this.reason;
  }
};

/**
 * Returns an object that contains a new `CancelToken` and a function that, when
 * called,
 * cancels the `CancelToken`.
 */
CancelToken.source = function source() {
  var cancel;
  var token = new CancelToken(function executor(c) {
    cancel = c;
  });
  return {
    token: token,
    cancel: cancel
  };
};

module.exports = CancelToken;

/***/ }),

/***/ "node_modules/axios/lib/cancel/isCancel.js":
/*!*****\
!*** ./node_modules/axios/lib/cancel/isCancel.js ***!
\*****/
/***/ ((module) => {

"use strict";

module.exports = function isCancel(value) {
  return !!(value && value.__CANCEL__);
};

/***/ }),

/***/ "node_modules/axios/lib/core/Axios.js":
/*!*****\
!*** ./node_modules/axios/lib/core/Axios.js ***!
\*****/

```

```

\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__ /*! ./../utils */
"./node_modules/axios/lib/utils.js");
var buildURL = __webpack_require__ /*! ../helpers/buildURL */
"./node_modules/axios/lib/helpers/buildURL.js");
var InterceptorManager = __webpack_require__ /*! ./InterceptorManager */
"./node_modules/axios/lib/core/InterceptorManager.js");
var dispatchRequest = __webpack_require__ /*! ./dispatchRequest */
"./node_modules/axios/lib/core/dispatchRequest.js");
var mergeConfig = __webpack_require__ /*! ./mergeConfig */
"./node_modules/axios/lib/core/mergeConfig.js");
var validator = __webpack_require__ /*! ../helpers/validator */
"./node_modules/axios/lib/helpers/validator.js");

var validators = validator.validators;
/**
 * Create a new instance of Axios
 *
 * @param {Object} instanceConfig The default config for the instance
 */
function Axios(instanceConfig) {
  this.defaults = instanceConfig;
  this.interceptors = {
    request: new InterceptorManager(),
    response: new InterceptorManager()
  };
}

/**
 * Dispatch a request
 *
 * @param {Object} config The config specific for this request (merged with
this.defaults)
 */
Axios.prototype.request = function request(config) {
  /*eslint no-param-reassign:0*/
  // Allow for axios('example/url'[, config]) a la fetch API
  if (typeof config === 'string') {
    config = arguments[1] || {};
    config.url = arguments[0];
  } else {
    config = config || {};
  }

  config = mergeConfig(this.defaults, config);

  // Set config.method
  if (config.method) {
    config.method = config.method.toLowerCase();
  } else if (this.defaults.method) {
    config.method = this.defaults.method.toLowerCase();
  } else {
    config.method = 'get';
  }

  var transitional = config.transitional;

```

```

if (transitional !== undefined) {
  validator.assertOptions(transitional, {
    silentJSONParsing: validators.transitional(validators.boolean, '1.0.0'),
    forcedJSONParsing: validators.transitional(validators.boolean, '1.0.0'),
    clarifyTimeoutError: validators.transitional(validators.boolean, '1.0.0')
  }, false);
}

// filter out skipped interceptors
var requestInterceptorChain = [];
var synchronousRequestInterceptors = true;
this.interceptors.request.forEach(function unshiftRequestInterceptors(interceptor)
{
  if (typeof interceptor.runWhen === 'function' && interceptor.runWhen(config) ===
false) {
    return;
  }

  synchronousRequestInterceptors = synchronousRequestInterceptors &&
interceptor.synchronous;

  requestInterceptorChain.unshift(interceptor.fulfilled, interceptor.rejected);
});

var responseInterceptorChain = [];
this.interceptors.response.forEach(function pushResponseInterceptors(interceptor) {
  responseInterceptorChain.push(interceptor.fulfilled, interceptor.rejected);
});

var promise;

if (!synchronousRequestInterceptors) {
  var chain = [dispatchRequest, undefined];

  Array.prototype.unshift.apply(chain, requestInterceptorChain);
  chain = chain.concat(responseInterceptorChain);

  promise = Promise.resolve(config);
  while (chain.length) {
    promise = promise.then(chain.shift(), chain.shift());
  }

  return promise;
}

var newConfig = config;
while (requestInterceptorChain.length) {
  var onFulfilled = requestInterceptorChain.shift();
  var onRejected = requestInterceptorChain.shift();
  try {
    newConfig = onFulfilled(newConfig);
  } catch (error) {
    onRejected(error);
    break;
  }
}

try {
  promise = dispatchRequest(newConfig);
} catch (error) {
  return Promise.reject(error);
}

```

```

    }

    while (responseInterceptorChain.length) {
      promise = promise.then(responseInterceptorChain.shift(),
responseInterceptorChain.shift());
    }

    return promise;
  };

Axios.prototype.getUri = function getUri(config) {
  config = mergeConfig(this.defaults, config);
  return buildURL(config.url, config.params, config.paramsSerializer).replace(/^\?/,
  '');
};

// Provide aliases for supported request methods
utils.forEach(['delete', 'get', 'head', 'options'], function
forEachMethodNoData(method) {
  /*eslint func-names:0*/
  Axios.prototype[method] = function(url, config) {
    return this.request(mergeConfig(config || {}, {
      method: method,
      url: url,
      data: (config || {}).data
    }));
  };
});

utils.forEach(['post', 'put', 'patch'], function forEachMethodWithData(method) {
  /*eslint func-names:0*/
  Axios.prototype[method] = function(url, data, config) {
    return this.request(mergeConfig(config || {}, {
      method: method,
      url: url,
      data: data
    }));
  };
});

module.exports = Axios;

/***/ }},

/***/ "./node_modules/axios/lib/core/InterceptorManager.js":
/*!*****!*\
!*** ./node_modules/axios/lib/core/InterceptorManager.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__(/*! ./../utils */
"./node_modules/axios/lib/utils.js");

function InterceptorManager() {
  this.handlers = [];
}

/**

```

```

* Add a new interceptor to the stack
*
* @param {Function} fulfilled The function to handle `then` for a `Promise`
* @param {Function} rejected The function to handle `reject` for a `Promise`
*
* @return {Number} An ID used to remove interceptor later
*/
InterceptorManager.prototype.use = function use(fulfilled, rejected, options) {
  this.handlers.push({
    fulfilled: fulfilled,
    rejected: rejected,
    synchronous: options ? options.synchronous : false,
    runWhen: options ? options.runWhen : null
  });
  return this.handlers.length - 1;
};

/**
* Remove an interceptor from the stack
*
* @param {Number} id The ID that was returned by `use`
*/
InterceptorManager.prototype.eject = function eject(id) {
  if (this.handlers[id]) {
    this.handlers[id] = null;
  }
};

/**
* Iterate over all the registered interceptors
*
* This method is particularly useful for skipping over any
* interceptors that may have become `null` calling `eject`.
*
* @param {Function} fn The function to call for each interceptor
*/
InterceptorManager.prototype.forEach = function forEach(fn) {
  utils.forEach(this.handlers, function forEachHandler(h) {
    if (h !== null) {
      fn(h);
    }
  });
};

module.exports = InterceptorManager;

/***/ }},

/***/ ".node_modules/axios/lib/core/buildFullPath.js":
/*!*****!\
!*** ./node_modules/axios/lib/core/buildFullPath.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var isAbsoluteURL = __webpack_require__ /*! ../helpers/isAbsoluteURL */
"./node_modules/axios/lib/helpers/isAbsoluteURL.js");
var combineURLs = __webpack_require__ /*! ../helpers/combineURLs */
"./node_modules/axios/lib/helpers/combineURLs.js");

```

```

/**
 * Creates a new URL by combining the baseURL with the requestedURL,
 * only when the requestedURL is not already an absolute URL.
 * If the requestURL is absolute, this function returns the requestedURL untouched.
 *
 * @param {string} baseURL The base URL
 * @param {string} requestedURL Absolute or relative URL to combine
 * @returns {string} The combined full path
 */
module.exports = function buildFullPath(baseURL, requestedURL) {
  if (baseURL && !isAbsoluteURL(requestedURL)) {
    return combineURLs(baseURL, requestedURL);
  }
  return requestedURL;
};

/***/ }},

/***/ ".node_modules/axios/lib/core/createError.js":
/*!*****!*\
!*** ./node_modules/axios/lib/core/createError.js ***!
\******/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var enhanceError = __webpack_require__(/*! ./enhanceError */
"./node_modules/axios/lib/core/enhanceError.js");

/**
 * Create an Error with the specified message, config, error code, request and
 * response.
 *
 * @param {string} message The error message.
 * @param {Object} config The config.
 * @param {string} [code] The error code (for example, 'ECONNABORTED').
 * @param {Object} [request] The request.
 * @param {Object} [response] The response.
 * @returns {Error} The created error.
 */
module.exports = function createError(message, config, code, request, response) {
  var error = new Error(message);
  return enhanceError(error, config, code, request, response);
};

/***/ }},

/***/ ".node_modules/axios/lib/core/dispatchRequest.js":
/*!*****!*\
!*** ./node_modules/axios/lib/core/dispatchRequest.js ***!
\******/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__(/*! ../../utils */
"./node_modules/axios/lib/utils.js");

```

```

var transformData = __webpack_require__(/*! ./transformData */
"./node_modules/axios/lib/core/transformData.js");
var isCancel = __webpack_require__(/*! ../cancel/isCancel */
"./node_modules/axios/lib/cancel/isCancel.js");
var defaults = __webpack_require__(/*! ../defaults */
"./node_modules/axios/lib/defaults.js");

/**
 * Throws a `Cancel` if cancellation has been requested.
 */
function throwIfCancellationRequested(config) {
  if (config.cancelToken) {
    config.cancelToken.throwIfRequested();
  }
}

/**
 * Dispatch a request to the server using the configured adapter.
 *
 * @param {object} config The config that is to be used for the request
 * @returns {Promise} The Promise to be fulfilled
 */
module.exports = function dispatchRequest(config) {
  throwIfCancellationRequested(config);

  // Ensure headers exist
  config.headers = config.headers || {};

  // Transform request data
  config.data = transformData.call(
    config,
    config.data,
    config.headers,
    config.transformRequest
  );

  // Flatten headers
  config.headers = utils.merge(
    config.headers.common || {},
    config.headers[config.method] || {},
    config.headers
  );

  utils.forEach(
    ['delete', 'get', 'head', 'post', 'put', 'patch', 'common'],
    function cleanHeaderConfig(method) {
      delete config.headers[method];
    }
  );

  var adapter = config.adapter || defaults.adapter;

  return adapter(config).then(function onAdapterResolution(response) {
    throwIfCancellationRequested(config);

    // Transform response data
    response.data = transformData.call(
      config,
      response.data,
      response.headers,
      config.transformResponse
    );
  });
};

```

```

    return response;
  }, function onAdapterRejection(reason) {
    if (!isCancel(reason)) {
      throwIfCancellationRequested(config);

      // Transform response data
      if (reason && reason.response) {
        reason.response.data = transformData.call(
          config,
          reason.response.data,
          reason.response.headers,
          config.transformResponse
        );
      }
    }

    return Promise.reject(reason);
  });
};

/***/ }),

/***/ "./node_modules/axios/lib/core/enhanceError.js":
/*!*****\
!*** ./node_modules/axios/lib/core/enhanceError.js ***!
\*****/
/***/ ((module) => {

"use strict";

/**
 * Update an Error with the specified config, error code, and response.
 *
 * @param {Error} error The error to update.
 * @param {Object} config The config.
 * @param {string} [code] The error code (for example, 'ECONNABORTED').
 * @param {Object} [request] The request.
 * @param {Object} [response] The response.
 * @returns {Error} The error.
 */
module.exports = function enhanceError(error, config, code, request, response) {
  error.config = config;
  if (code) {
    error.code = code;
  }

  error.request = request;
  error.response = response;
  error.isAxiosError = true;

  error.toJSON = function toJSON() {
    return {
      // Standard
      message: this.message,
      name: this.name,
      // Microsoft
      description: this.description,
      number: this.number,
      // Mozilla

```

```

        fileName: this.fileName,
        lineNumber: this.lineNumber,
        columnNumber: this.columnNumber,
        stack: this.stack,
        // Axios
        config: this.config,
        code: this.code
    };
};
return error;
};

/***/ }},

/***/ ". /node_modules/axios/lib/core/mergeConfig.js":
/*!*****!\
!*** ./node_modules/axios/lib/core/mergeConfig.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__(/*! ../utils */ ". /node_modules/axios/lib/utils.js");

/**
 * Config-specific merge-function which creates a new config-object
 * by merging two configuration objects together.
 *
 * @param {Object} config1
 * @param {Object} config2
 * @returns {Object} New object resulting from merging config2 to config1
 */
module.exports = function mergeConfig(config1, config2) {
    // eslint-disable-next-line no-param-reassign
    config2 = config2 || {};
    var config = {};

    var valueFromConfig2Keys = ['url', 'method', 'data'];
    var mergeDeepPropertiesKeys = ['headers', 'auth', 'proxy', 'params'];
    var defaultToConfig2Keys = [
        'baseUrl', 'transformRequest', 'transformResponse', 'paramsSerializer',
        'timeout', 'timeoutMessage', 'withCredentials', 'adapter', 'responseType',
        'xsrfCookieName',
        'xsrfHeaderName', 'onUploadProgress', 'onDownloadProgress', 'decompress',
        'maxContentLength', 'maxBodyLength', 'maxRedirects', 'transport', 'httpAgent',
        'httpsAgent', 'cancelToken', 'socketPath', 'responseEncoding'
    ];
    var directMergeKeys = ['validateStatus'];

    function getMergedValue(target, source) {
        if (utils.isPlainObject(target) && utils.isPlainObject(source)) {
            return utils.merge(target, source);
        } else if (utils.isPlainObject(source)) {
            return utils.merge({}, source);
        } else if (utils.isArray(source)) {
            return source.slice();
        }
        return source;
    }

```

```

function mergeDeepProperties(prop) {
  if (!utils.isUndefined(config2[prop])) {
    config[prop] = getMergedValue(config1[prop], config2[prop]);
  } else if (!utils.isUndefined(config1[prop])) {
    config[prop] = getMergedValue(undefined, config1[prop]);
  }
}

utils.forEach(valueFromConfig2Keys, function valueFromConfig2(prop) {
  if (!utils.isUndefined(config2[prop])) {
    config[prop] = getMergedValue(undefined, config2[prop]);
  }
});

utils.forEach(mergeDeepPropertiesKeys, mergeDeepProperties);

utils.forEach(defaultToConfig2Keys, function defaultToConfig2(prop) {
  if (!utils.isUndefined(config2[prop])) {
    config[prop] = getMergedValue(undefined, config2[prop]);
  } else if (!utils.isUndefined(config1[prop])) {
    config[prop] = getMergedValue(undefined, config1[prop]);
  }
});

utils.forEach(directMergeKeys, function merge(prop) {
  if (prop in config2) {
    config[prop] = getMergedValue(config1[prop], config2[prop]);
  } else if (prop in config1) {
    config[prop] = getMergedValue(undefined, config1[prop]);
  }
});

var axiosKeys = valueFromConfig2Keys
  .concat(mergeDeepPropertiesKeys)
  .concat(defaultToConfig2Keys)
  .concat(directMergeKeys);

var otherKeys = Object
  .keys(config1)
  .concat(Object.keys(config2))
  .filter(function filterAxiosKeys(key) {
    return axiosKeys.indexOf(key) === -1;
  });

utils.forEach(otherKeys, mergeDeepProperties);

return config;
};

/***/ }},

/***/ ". /node_modules/axios/lib/core/settle.js":
/*!*****!\
!*** ./node_modules/axios/lib/core/settle.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

```

```

var createError = __webpack_require__ (/*! ./createError */
"./node_modules/axios/lib/core/createError.js");

/**
 * Resolve or reject a Promise based on response status.
 *
 * @param {Function} resolve A function that resolves the promise.
 * @param {Function} reject A function that rejects the promise.
 * @param {object} response The response.
 */
module.exports = function settle(resolve, reject, response) {
  var validateStatus = response.config.validateStatus;
  if (!response.status || !validateStatus || validateStatus(response.status)) {
    resolve(response);
  } else {
    reject(createError(
      'Request failed with status code ' + response.status,
      response.config,
      null,
      response.request,
      response
    ));
  }
};

/***/ }},

/***/ "./node_modules/axios/lib/core/transformData.js":
/*!*****\
!*** ./node_modules/axios/lib/core/transformData.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__ (/*! ../../utils */
"./node_modules/axios/lib/utils.js");
var defaults = __webpack_require__ (/*! ../../defaults */
"./node_modules/axios/lib/defaults.js");

/**
 * Transform the data for a request or a response
 *
 * @param {Object|String} data The data to be transformed
 * @param {Array} headers The headers for the request or response
 * @param {Array|Function} fns A single function or Array of functions
 * @returns {*} The resulting transformed data
 */
module.exports = function transformData(data, headers, fns) {
  var context = this || defaults;
  /*eslint no-param-reassign:0*/
  utils.forEach(fns, function transform(fn) {
    data = fn.call(context, data, headers);
  });

  return data;
};

/***/ }},

```

```

/***/ "./node_modules/axios/lib/defaults.js":
/*!*****!\
!*** ./node_modules/axios/lib/defaults.js ***!
\*****!
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

  "use strict";
  /* provided dependency */ var process = __webpack_require__(/*! process/browser.js */
  "./node_modules/process/browser.js");

  var utils = __webpack_require__(/*! ./utils */ "./node_modules/axios/lib/utils.js");
  var normalizeHeaderName = __webpack_require__(/*! ./helpers/normalizeHeaderName */
  "./node_modules/axios/lib/helpers/normalizeHeaderName.js");
  var enhanceError = __webpack_require__(/*! ./core/enhanceError */
  "./node_modules/axios/lib/core/enhanceError.js");

  var DEFAULT_CONTENT_TYPE = {
    'Content-Type': 'application/x-www-form-urlencoded'
  };

  function setContentTypeIfUnset(headers, value) {
    if (!utils.isUndefined(headers) && utils.isUndefined(headers['Content-Type'])) {
      headers['Content-Type'] = value;
    }
  }

  function getDefaultAdapter() {
    var adapter;
    if (typeof XMLHttpRequest !== 'undefined') {
      // For browsers use XHR adapter
      adapter = __webpack_require__(/*! ./adapters/xhr */
      "./node_modules/axios/lib/adapters/xhr.js");
    } else if (typeof process !== 'undefined' &&
    Object.prototype.toString.call(process) === '[object process]') {
      // For node use HTTP adapter
      adapter = __webpack_require__(/*! ./adapters/http */
      "./node_modules/axios/lib/adapters/xhr.js");
    }
    return adapter;
  }

  function stringifySafely(rawValue, parser, encoder) {
    if (utils.isString(rawValue)) {
      try {
        (parser || JSON.parse)(rawValue);
        return utils.trim(rawValue);
      } catch (e) {
        if (e.name !== 'SyntaxError') {
          throw e;
        }
      }
    }

    return (encoder || JSON.stringify)(rawValue);
  }

  var defaults = {

    transitional: {
      silentJSONParsing: true,

```

```

    forcedJSONParsing: true,
    clarifyTimeoutError: false
  },

  adapter: getDefaultAdapter(),

  transformRequest: [function transformRequest(data, headers) {
    normalizeHeaderName(headers, 'Accept');
    normalizeHeaderName(headers, 'Content-Type');

    if (utils.isFormData(data) ||
        utils.isArrayBuffer(data) ||
        utils.isBuffer(data) ||
        utils.isStream(data) ||
        utils.isFile(data) ||
        utils.isBlob(data)
    ) {
      return data;
    }
    if (utils.isArrayBufferView(data)) {
      return data.buffer;
    }
    if (utils.isURLSearchParams(data)) {
      setContentTypeIfUnset(headers, 'application/x-www-form-urlencoded;charset=utf-8');
      return data.toString();
    }
    if (utils.isObject(data) || (headers && headers['Content-Type'] ===
'application/json')) {
      setContentTypeIfUnset(headers, 'application/json');
      return stringifySafely(data);
    }
    return data;
  }],

  transformResponse: [function transformResponse(data) {
    var transitional = this.transitional;
    var silentJSONParsing = transitional && transitional.silentJSONParsing;
    var forcedJSONParsing = transitional && transitional.forcedJSONParsing;
    var strictJSONParsing = !silentJSONParsing && this.responseType === 'json';

    if (strictJSONParsing || (forcedJSONParsing && utils.isString(data) &&
data.length)) {
      try {
        return JSON.parse(data);
      } catch (e) {
        if (strictJSONParsing) {
          if (e.name === 'SyntaxError') {
            throw enhanceError(e, this, 'E_JSON_PARSE');
          }
          throw e;
        }
      }
    }

    return data;
  }],

  /**
   * A timeout in milliseconds to abort a request. If set to 0 (default) a
   * timeout is not created.
   */

```

```

    timeout: 0,

    xsrfCookieName: 'XSRF-TOKEN',
    xsrfHeaderName: 'X-XSRF-TOKEN',

    maxContentLength: -1,
    maxBodyLength: -1,

    validateStatus: function validateStatus(status) {
        return status >= 200 && status < 300;
    }
};

defaults.headers = {
    common: {
        'Accept': 'application/json, text/plain, */*'
    }
};

utils.forEach(['delete', 'get', 'head'], function forEachMethodNoData(method) {
    defaults.headers[method] = {};
});

utils.forEach(['post', 'put', 'patch'], function forEachMethodWithData(method) {
    defaults.headers[method] = utils.merge(DEFAULT_CONTENT_TYPE);
});

module.exports = defaults;

/***/ }},

/***/ "./node_modules/axios/lib/helpers/bind.js":
/*!*****\
!*** ./node_modules/axios/lib/helpers/bind.js ***!
\*****/
/***/ ((module) => {

"use strict";

module.exports = function bind(fn, thisArg) {
    return function wrap() {
        var args = new Array(arguments.length);
        for (var i = 0; i < args.length; i++) {
            args[i] = arguments[i];
        }
        return fn.apply(thisArg, args);
    };
};

/***/ }},

/***/ "./node_modules/axios/lib/helpers/buildURL.js":
/*!*****\
!*** ./node_modules/axios/lib/helpers/buildURL.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

```

```

var utils = __webpack_require__ (/*! ../../utils */
"./node_modules/axios/lib/utils.js");

function encode(val) {
  return encodeURIComponent(val).
    replace(/%3A/gi, ':').
    replace(/%24/g, '$').
    replace(/%2C/gi, ',').
    replace(/%20/g, '+').
    replace(/%5B/gi, '[').
    replace(/%5D/gi, ']');
}

/**
 * Build a URL by appending params to the end
 *
 * @param {string} url The base of the url (e.g., http://www.google.com)
 * @param {object} [params] The params to be appended
 * @returns {string} The formatted url
 */
module.exports = function buildURL(url, params, paramsSerializer) {
  /*eslint no-param-reassign:0*/
  if (!params) {
    return url;
  }

  var serializedParams;
  if (paramsSerializer) {
    serializedParams = paramsSerializer(params);
  } else if (utils.isURLSearchParams(params)) {
    serializedParams = params.toString();
  } else {
    var parts = [];

    utils.forEach(params, function serialize(val, key) {
      if (val === null || typeof val === 'undefined') {
        return;
      }

      if (utils.isArray(val)) {
        key = key + '[]';
      } else {
        val = [val];
      }

      utils.forEach(val, function parseValue(v) {
        if (utils.isDate(v)) {
          v = v.toISOString();
        } else if (utils.isObject(v)) {
          v = JSON.stringify(v);
        }
        parts.push(encode(key) + '=' + encode(v));
      });
    });

    serializedParams = parts.join('&');
  }

  if (serializedParams) {
    var hashmarkIndex = url.indexOf('#');
    if (hashmarkIndex !== -1) {

```

```

    url = url.slice(0, hashmarkIndex);
  }

  url += (url.indexOf('?') === -1 ? '?' : '&') + serializedParams;
}

return url;
};

/***/ }},

/***/ "./node_modules/axios/lib/helpers/combineURLs.js":
/*!*****\
!*** ./node_modules/axios/lib/helpers/combineURLs.js ***!
\*****/
/***/ ((module) => {

"use strict";

/**
 * Creates a new URL by combining the specified URLs
 *
 * @param {string} baseURL The base URL
 * @param {string} relativeURL The relative URL
 * @returns {string} The combined URL
 */
module.exports = function combineURLs(baseURL, relativeURL) {
  return relativeURL
    ? baseURL.replace(/\/+$/, '') + '/' + relativeURL.replace(/^\/+/, '')
    : baseURL;
};

/***/ }},

/***/ "./node_modules/axios/lib/helpers/cookies.js":
/*!*****\
!*** ./node_modules/axios/lib/helpers/cookies.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__('./../utils');

"./node_modules/axios/lib/cookies.js");

module.exports = (
  utils.isStandardBrowserEnv() ?

  // Standard browser envs support document.cookie
  (function standardBrowserEnv() {
    return {
      write: function write(name, value, expires, path, domain, secure) {
        var cookie = [];
        cookie.push(name + '=' + encodeURIComponent(value));

        if (utils.isNumber(expires)) {
          cookie.push('expires=' + new Date(expires).toGMTString());
        }
      }
    }
  })

```

```

        if (utils.isString(path)) {
            cookie.push('path=' + path);
        }

        if (utils.isString(domain)) {
            cookie.push('domain=' + domain);
        }

        if (secure === true) {
            cookie.push('secure');
        }

        document.cookie = cookie.join('; ');
    },

    read: function read(name) {
        var match = document.cookie.match(new RegExp('(^|;\\s*)(' + name +
        ')=([^\s]*)'));
        return (match ? decodeURIComponent(match[3]) : null);
    },

    remove: function remove(name) {
        this.write(name, '', Date.now() - 86400000);
    }
};
})();

// Non standard browser env (web workers, react-native) lack needed support.
(function nonStandardBrowserEnv() {
    return {
        write: function write() {},
        read: function read() { return null; },
        remove: function remove() {}
    };
})();

/***/ })),

/***/ "node_modules/axios/lib/helpers/isAbsoluteURL.js":
/*!*****!*\
  !*** ./node_modules/axios/lib/helpers/isAbsoluteURL.js ***!
  \******/
/***/ ((module) => {

"use strict";

/**
 * Determines whether the specified URL is absolute
 *
 * @param {string} url The URL to test
 * @returns {boolean} True if the specified URL is absolute, otherwise false
 */
module.exports = function isAbsoluteURL(url) {
    // A URL is considered absolute if it begins with "<scheme>://" or "://" (protocol-
    // relative URL).
    // RFC 3986 defines scheme name as a sequence of characters beginning with a letter
    // and followed
    // by any combination of letters, digits, plus, period, or hyphen.

```

```

    return /^[a-z][a-z\d\+\-\.\:]*?\:\/\/i.test(url);
};

/***/ }},

/***/ "./node_modules/axios/lib/helpers/isAxiosError.js":
/*!*****!*\
!*** ./node_modules/axios/lib/helpers/isAxiosError.js ***!
\*****/
/***/ ((module) => {

"use strict";

/**
 * Determines whether the payload is an error thrown by Axios
 *
 * @param {*} payload The value to test
 * @returns {boolean} True if the payload is an error thrown by Axios, otherwise
false
 */
module.exports = function isAxiosError(payload) {
  return (typeof payload === 'object') && (payload.isAxiosError === true);
};

/***/ }},

/***/ "./node_modules/axios/lib/helpers/isURLSameOrigin.js":
/*!*****!*\
!*** ./node_modules/axios/lib/helpers/isURLSameOrigin.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__('./../utils *')
"./node_modules/axios/lib/utils.js");

module.exports = (
  utils.isStandardBrowserEnv() ?

    // Standard browser envs have full support of the APIs needed to test
    // whether the request URL is of the same origin as current location.
    (function standardBrowserEnv() {
      var msie = /(msie|trident)/i.test(navigator.userAgent);
      var urlParsingNode = document.createElement('a');
      var originURL;

      /**
       * Parse a URL to discover it's components
       *
       * @param {String} url The URL to be parsed
       * @returns {Object}
       */
      function resolveURL(url) {
        var href = url;

        if (msie) {
          // IE needs attribute set twice to normalize properties

```

```

        urlParsingNode.setAttribute('href', href);
        href = urlParsingNode.href;
    }

    urlParsingNode.setAttribute('href', href);

    // urlParsingNode provides the UrlUtils interface -
    http://url.spec.whatwg.org/#urlutils
    return {
        href: urlParsingNode.href,
        protocol: urlParsingNode.protocol ? urlParsingNode.protocol.replace(/:$/,
'') : '',
        host: urlParsingNode.host,
        search: urlParsingNode.search ? urlParsingNode.search.replace(/^\?/, '') :
'',
        hash: urlParsingNode.hash ? urlParsingNode.hash.replace(/^#/, '') : '',
        hostname: urlParsingNode.hostname,
        port: urlParsingNode.port,
        pathname: (urlParsingNode.pathname.charAt(0) === '/') ?
            urlParsingNode.pathname :
            '/' + urlParsingNode.pathname
    };
}

originURL = resolveURL(window.location.href);

/**
 * Determine if a URL shares the same origin as the current location
 *
 * @param {String} requestURL The URL to test
 * @returns {boolean} True if URL shares the same origin, otherwise false
 */
return function isURLSameOrigin(requestURL) {
    var parsed = (utils.isString(requestURL)) ? resolveURL(requestURL) :
requestURL;
    return (parsed.protocol === originURL.protocol &&
        parsed.host === originURL.host);
};
})();

// Non standard browser envs (web workers, react-native) lack needed support.
(function nonStandardBrowserEnv() {
    return function isURLSameOrigin() {
        return true;
    };
})();
};

/***/ }},

/***/ ". /node_modules/axios/lib/helpers/normalizeHeaderName.js":
/*!*****!*\
!*** ./node_modules/axios/lib/helpers/normalizeHeaderName.js ***!
\******/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__(/*! ../utils */ ". /node_modules/axios/lib/utils.js");

```

```

module.exports = function normalizeHeaderName(headers, normalizedName) {
  utils.forEach(headers, function processHeader(value, name) {
    if (name !== normalizedName && name.toUpperCase() ===
normalizedName.toUpperCase()) {
      headers[normalizedName] = value;
      delete headers[name];
    }
  });
};

/***/ })),

/***/ "./node_modules/axios/lib/helpers/parseHeaders.js":
/*!*****\
!*** ./node_modules/axios/lib/helpers/parseHeaders.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var utils = __webpack_require__(/*! ./../utils */
"./node_modules/axios/lib/utils.js");

// Headers whose duplicates are ignored by node
// c.f. https://nodejs.org/api/http.html#http_message_headers
var ignoreDuplicateOf = [
  'age', 'authorization', 'content-length', 'content-type', 'etag',
  'expires', 'from', 'host', 'if-modified-since', 'if-unmodified-since',
  'last-modified', 'location', 'max-forwards', 'proxy-authorization',
  'referer', 'retry-after', 'user-agent'
];

/**
 * Parse headers into an object
 *
 * ``
 * Date: Wed, 27 Aug 2014 08:58:49 GMT
 * Content-Type: application/json
 * Connection: keep-alive
 * Transfer-Encoding: chunked
 * ``
 *
 * @param {String} headers Headers needing to be parsed
 * @returns {Object} Headers parsed into an object
 */
module.exports = function parseHeaders(headers) {
  var parsed = {};
  var key;
  var val;
  var i;

  if (!headers) { return parsed; }

  utils.forEach(headers.split('\n'), function parser(line) {
    i = line.indexOf(':');
    key = utils.trim(line.substr(0, i)).toLowerCase();
    val = utils.trim(line.substr(i + 1));

    if (key) {
      if (parsed[key] && ignoreDuplicateOf.indexOf(key) >= 0) {

```

```

        return;
    }
    if (key === 'set-cookie') {
        parsed[key] = (parsed[key] ? parsed[key] : []).concat([val]);
    } else {
        parsed[key] = parsed[key] ? parsed[key] + ', ' + val : val;
    }
}
});

return parsed;
};

/***/ }},

/***/ ". /node_modules/axios/lib/helpers/spread.js":
/*!*****!\
!*** ./node_modules/axios/lib/helpers/spread.js ***!
\*****/
/***/ ((module) => {

"use strict";

/**
 * Syntactic sugar for invoking a function and expanding an array for arguments.
 *
 * Common use case would be to use `Function.prototype.apply`.
 *
 * ```js
 * function f(x, y, z) {}
 * var args = [1, 2, 3];
 * f.apply(null, args);
 * ```
 *
 * With `spread` this example can be re-written.
 *
 * ```js
 * spread(function(x, y, z) {})([1, 2, 3]);
 * ```
 *
 * @param {Function} callback
 * @returns {Function}
 */
module.exports = function spread(callback) {
    return function wrap(arr) {
        return callback.apply(null, arr);
    };
};

/***/ }},

/***/ ". /node_modules/axios/lib/helpers/validator.js":
/*!*****!\
!*** ./node_modules/axios/lib/helpers/validator.js ***!
\*****/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

```

```

var pkg = __webpack_require__(/*! ../../../package.json */
"./node_modules/axios/package.json");

var validators = {};

// eslint-disable-next-line func-names
['object', 'boolean', 'number', 'function', 'string',
'symbol'].forEach(function(type, i) {
  validators[type] = function validator(thing) {
    return typeof thing === type || 'a' + (i < 1 ? 'n ' : ' ') + type;
  };
});

var deprecatedWarnings = {};
var currentVerArr = pkg.version.split('.');

/**
 * Compare package versions
 * @param {string} version
 * @param {string?} thanVersion
 * @returns {boolean}
 */
function isOlderVersion(version, thanVersion) {
  var pkgVersionArr = thanVersion ? thanVersion.split('.') : currentVerArr;
  var destVer = version.split('.');
  for (var i = 0; i < 3; i++) {
    if (pkgVersionArr[i] > destVer[i]) {
      return true;
    } else if (pkgVersionArr[i] < destVer[i]) {
      return false;
    }
  }
  return false;
}

/**
 * Transitional option validator
 * @param {function|boolean?} validator
 * @param {string?} version
 * @param {string} message
 * @returns {function}
 */
validators.transitional = function transitional(validator, version, message) {
  var isDeprecated = version && isOlderVersion(version);

  function formatMessage(opt, desc) {
    return '[Axios v' + pkg.version + '] Transitional option \'' + opt + '\'' + desc
+ (message ? '. ' + message : '');
  }

  // eslint-disable-next-line func-names
  return function(value, opt, opts) {
    if (validator === false) {
      throw new Error(formatMessage(opt, ' has been removed in ' + version));
    }

    if (isDeprecated && !deprecatedWarnings[opt]) {
      deprecatedWarnings[opt] = true;
      // eslint-disable-next-line no-console
      console.warn(
        formatMessage(

```

```

        opt,
        ' has been deprecated since v' + version + ' and will be removed in the
near future'
    )
    );
}

return validator ? validator(value, opt, opts) : true;
};
};

/**
 * Assert object's properties type
 * @param {object} options
 * @param {object} schema
 * @param {boolean?} allowUnknown
 */

function assertOptions(options, schema, allowUnknown) {
    if (typeof options !== 'object') {
        throw new TypeError('options must be an object');
    }
    var keys = Object.keys(options);
    var i = keys.length;
    while (i-- > 0) {
        var opt = keys[i];
        var validator = schema[opt];
        if (validator) {
            var value = options[opt];
            var result = value === undefined || validator(value, opt, options);
            if (result !== true) {
                throw new TypeError('option ' + opt + ' must be ' + result);
            }
            continue;
        }
        if (allowUnknown !== true) {
            throw Error('Unknown option ' + opt);
        }
    }
}

module.exports = {
    isOlderVersion: isOlderVersion,
    assertOptions: assertOptions,
    validators: validators
};

/***/ }},

/***/ "./node_modules/axios/lib/utils.js":
/*!*****!*\
  !*** ./node_modules/axios/lib/utils.js ***!
  \******/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

"use strict";

var bind = __webpack_require__ /*! ./helpers/bind */
"./node_modules/axios/lib/helpers/bind.js");

```

```

// utils is a library of generic helper functions non-specific to axios

var toString = Object.prototype.toString;

/**
 * Determine if a value is an Array
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is an Array, otherwise false
 */
function isArray(val) {
  return toString.call(val) === '[object Array]';
}

/**
 * Determine if a value is undefined
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if the value is undefined, otherwise false
 */
function isUndefined(val) {
  return typeof val === 'undefined';
}

/**
 * Determine if a value is a Buffer
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a Buffer, otherwise false
 */
function isBuffer(val) {
  return val !== null && !isUndefined(val) && val.constructor !== null &&
    !isUndefined(val.constructor)
    && typeof val.constructor.isBuffer === 'function' &&
    val.constructor.isBuffer(val);
}

/**
 * Determine if a value is an ArrayBuffer
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is an ArrayBuffer, otherwise false
 */
function isArrayBuffer(val) {
  return toString.call(val) === '[object ArrayBuffer]';
}

/**
 * Determine if a value is a FormData
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is an FormData, otherwise false
 */
function isFormData(val) {
  return (typeof FormData !== 'undefined') && (val instanceof FormData);
}

/**
 * Determine if a value is a view on an ArrayBuffer
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a view on an ArrayBuffer, otherwise false

```

```

*/
function isArrayBufferView(val) {
  var result;
  if ((typeof ArrayBuffer !== 'undefined') && (ArrayBuffer.isView)) {
    result = ArrayBuffer.isView(val);
  } else {
    result = (val) && (val.buffer) && (val.buffer instanceof ArrayBuffer);
  }
  return result;
}

/**
 * Determine if a value is a String
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a String, otherwise false
 */
function isString(val) {
  return typeof val === 'string';
}

/**
 * Determine if a value is a Number
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a Number, otherwise false
 */
function isNumber(val) {
  return typeof val === 'number';
}

/**
 * Determine if a value is an Object
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is an Object, otherwise false
 */
function isObject(val) {
  return val !== null && typeof val === 'object';
}

/**
 * Determine if a value is a plain Object
 *
 * @param {Object} val The value to test
 * @return {boolean} True if value is a plain Object, otherwise false
 */
function isPlainObject(val) {
  if (toString.call(val) !== '[object Object]') {
    return false;
  }

  var prototype = Object.getPrototypeOf(val);
  return prototype === null || prototype === Object.prototype;
}

/**
 * Determine if a value is a Date
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a Date, otherwise false
 */

```

```

function isDate(val) {
    return toString.call(val) === '[object Date]';
}

/**
 * Determine if a value is a File
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a File, otherwise false
 */
function isFile(val) {
    return toString.call(val) === '[object File]';
}

/**
 * Determine if a value is a Blob
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a Blob, otherwise false
 */
function isBlob(val) {
    return toString.call(val) === '[object Blob]';
}

/**
 * Determine if a value is a Function
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a Function, otherwise false
 */
function isFunction(val) {
    return toString.call(val) === '[object Function]';
}

/**
 * Determine if a value is a Stream
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a Stream, otherwise false
 */
function isStream(val) {
    return isObject(val) && isFunction(val.pipe);
}

/**
 * Determine if a value is a URLSearchParams object
 *
 * @param {Object} val The value to test
 * @returns {boolean} True if value is a URLSearchParams object, otherwise false
 */
function isURLSearchParams(val) {
    return typeof URLSearchParams !== 'undefined' && val instanceof URLSearchParams;
}

/**
 * Trim excess whitespace off the beginning and end of a string
 *
 * @param {String} str The String to trim
 * @returns {String} The String freed of excess whitespace
 */
function trim(str) {
    return str.trim ? str.trim() : str.replace(/^\s+|\s+$/g, '');
}

```

```

}

/**
 * Determine if we're running in a standard browser environment
 *
 * This allows axios to run in a web worker, and react-native.
 * Both environments support XMLHttpRequest, but not fully standard globals.
 *
 * web workers:
 *   typeof window -> undefined
 *   typeof document -> undefined
 *
 * react-native:
 *   navigator.product -> 'ReactNative'
 *   nativescript
 *   navigator.product -> 'NativeScript' or 'NS'
 */
function isStandardBrowserEnv() {
  if (typeof navigator !== 'undefined' && (navigator.product === 'ReactNative' ||
    navigator.product === 'NativeScript' ||
    navigator.product === 'NS')) {
    return false;
  }
  return (
    typeof window !== 'undefined' &&
    typeof document !== 'undefined'
  );
}

/**
 * Iterate over an Array or an Object invoking a function for each item.
 *
 * If `obj` is an Array callback will be called passing
 * the value, index, and complete array for each item.
 *
 * If 'obj' is an Object callback will be called passing
 * the value, key, and complete object for each property.
 *
 * @param {Object|Array} obj The object to iterate
 * @param {Function} fn The callback to invoke for each item
 */
function forEach(obj, fn) {
  // Don't bother if no value provided
  if (obj === null || typeof obj === 'undefined') {
    return;
  }

  // Force an array if not already something iterable
  if (typeof obj !== 'object') {
    /*eslint no-param-reassign:0*/
    obj = [obj];
  }

```



Метадані

ДОКУМЕНТ

Заголовок:

ВРБак СА - Ситник бд

Автор:

Науковий керівник / Експерт:

С.М. Коновалов

ІД документу:

333635947

ОРГАНІЗАЦІЯ

Назва організації:

Odesa National Maritime University

підрозділ:

Технічна кібернетика й інформаційні технології ім. професора Р.В. Мерктя

ЗВІТ

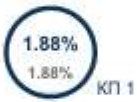
Дата звіту:

4/20/2026

Дата редагування:

Обсяг знайдених подібностей

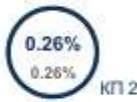
Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

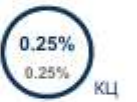
Довжина фраз для коефіцієнта подібності 2



КП 2

15504

Кількість слів



КЦ

131597

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		5
Білі знаки		0
Парафрази (SmartMarks)		22

Джерела

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копір тексту