

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР

на тему: «Розробка інформаційної системи для управління логістичними процесами підприємств оптової торгівлі»

САВОХ ВЛАДИСЛАВ АНДРІЙОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ, 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувачка кафедри ІТ

д.т.н., професор Гончаренко Т.А.

„___” _____ 2025 року

**КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР**

на тему: «Розробка підсистеми калібрування багатопараметричної системи
на основі генетичного алгоритму»

Я як здобувач вищої освіти КНУБА розумію і підтримую політику закладу з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволена допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Здобувач Савох Владислав Андрійович

(прізвище, ім'я та по батькові повністю)

122 «Комп'ютерні науки»

(спеціальність)

Інформаційні управляючі системи і технології

(освітня програма)

Групи КН-21-2

Керівник Саченко І.А.

(прізвище та ініціали)

доцент, кандидат технічних наук

(вчене звання, науковий ступінь)

Рецензент к.т.н., доц. Баліна О.І.

(Прізвище та ініціали)

Ідентичність підтверджую

Київ, 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Випускова кафедра: інформаційних технологій

Ступінь вищої освіти: «бакалавр»

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ

Завідувачка кафедри ІТ

д.т.н., професор Гончаренко Т.А.

„___” _____ 2025 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ БАКАЛАВР**

Савох Владислав Андрійович

1. Тема роботи: Розробка інформаційної системи для управління логістичними процесами підприємств оптової торгівлі

затверджена наказом ректора КНУБА № 235/23/25 від 14.02.2025.

2. Керівник роботи: Саченко Ілля Анатолійович, к.т.н, доцент
кафедри інформаційних технологій

3. Строк подання студентом роботи до захисту: _____

4. Зміст пояснювальної записки за розділами:

P.1. Аналіз предметної області та постановка задачі

P.2. Методи та моделі управління логістичними процесами

P.3. Програмна реалізація інформаційної системи

P.4. Оптимізація та ефективність логістичних процесів

5. Інформаційні слайди:

S.1. Вступ

S.2-5. Аналіз предметної області та постановка задачі

S.6. Структурний та функціональний аналіз

S.7-10. Аналіз використаних алгоритмів

S.11-15. Програмна реалізація

S.16-20. Техніко-економічне обґрунтування розробки ІС

S.21. Висновки

6. Консультанти розділів кваліфікаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Методи та моделі управління логістичними процесами	к.т.н. доц. Горда О.В.	07.10.2024	
Прийом програмного продукту	к.т.н. доц. Шабала Є.Є.		

7. Календарний план виконання кваліфікаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Січень 2025 р.
Р. 2. Методи та моделі управління логістичними процесами	Лютий 2025 р.
Р. 3. Програмна реалізація інформаційної системи	Березень 2025 р.
Тестовий приклад програми	Березень 2025 р.
Р. 4. Техніко-економічне обґрунтування розробки ІС	Квітень 2025 р.
Остаточне оформлення роботи	Травень 2025 р.
Направлення роботи на рецензування	Червень 2025 р.
Попередній захист роботи на кафедрі	Червень 2025 р.

8. Дата видачі завдання: 20.01.2025 р.

Завідувачка

(підпис)

Гончаренко Т.А.

(прізвище та ініціали)

Керівник

(підпис)

Саченко І.А.

(прізвище та ініціали)

Студент

(підпис)

Савох В.А,

(прізвище та ініціали)

РЕЗЮМЕ (SUMMARY) <i>до кваліфікаційної випускної роботи</i> <i>Здобувача:</i>	Савох Владислав Андрійович Vladyslav Savokh		
<i>ЗВО</i>	Київський національний університет будівництва і архітектури		
<i>Тема (українською та англійською)</i>	Розробка інформаційної системи для управління логістичними процесами підприємств оптової торгівлі. Development of an information system for managing the logistics processes of wholesale enterprises.		
<i>Освітній ступінь</i>	Бакалавр		
<i>Факультет</i>	Автоматизації і інформаційних технологій		
<i>Випускова кафедра</i>	Інформаційних технологій		
<i>Спеціальність</i>	122 «Комп'ютерні науки»		
<i>Освітня програма</i>	Інформаційні управляючі системи та технології		
<i>Керівник</i>	Саченко Ілля Анатолійович		
<i>Обсяг роботи:</i>	пояснювальна записка, стор.	розділів	Кількість слайдів презентації
	108	4	22
<i>Розділ 1.</i>	Аналіз предметної області та постановка задачі		
<i>Розділ 2.</i>	Методи та моделі управління логістичними процесами		
<i>Розділ 3.</i>	Програмна реалізація інформаційної системи		
<i>Розділ 4.</i>	Техніко-економічне обґрунтування розробки ІС		
<i>Ключові слова:</i> <i>Keywords:</i>	логістичні процеси, управління ланцюгами поставок, інформаційні системи, оптимізація логістики, автоматизація бізнес-процесів, аналіз даних, штучний інтелект. logistics processes, supply chain management, information systems, logistics optimization, business process automation, data analysis, artificial intelligence.		

Здобувач: _____ /Владислав САВОХ /

Керівник: _____ / Ілля САЧЕНКО/

“ ___ ” _____ 2025р.

АНОТАЦІЯ

Савох В. А. Розробка інформаційної системи для управління логістичними процесами підприємств оптової торгівлі.

Кваліфікаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», освітньо-професійна програма: «Інформаційні управляючі системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2025.

Робота присвячена розробці інформаційної системи для оптимізації управління логістичними процесами на підприємствах оптової торгівлі. Описано основні методи та алгоритми, використані для автоматизації та покращення ефективності логістичних операцій. Наведено результати тестування, які підтверджують ефективність запропонованої системи.

Ключові слова: логістичні процеси, інформаційні системи, управління ланцюгами поставок, оптимізація логістики, автоматизація бізнес-процесів, аналіз даних, штучний інтелект.

SUMMARY

Savokh. V. A. Development of a calibration subsystem for a multiparameter system based on a genetic algorithm.

Bachelor's thesis for a bachelor's degree in specialty: 122 "Computer Science", specialization: "Information Management Systems and Technologies - Kyiv National University of Construction and Architecture - Kyiv, 2025.

The work is devoted to the development of an information system for optimizing the management of logistics processes in wholesale enterprises. The main methods and algorithms used for automation and improving the efficiency of logistics operations are described. The test results confirm the effectiveness of the proposed system.

Keywords: logistics processes, information systems, supply chain management, logistics optimization, business process automation, data analysis, artificial intelligence.

ЗМІСТ

ВСТУП.....	10
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	12
1.1 Постановка й аналіз проблеми.....	12
1.2 Аналіз логістичних процесів у сфері оптової торгівлі	14
1.3 Огляд сучасних інформаційних систем для управління логістикою.	18
1.4 Визначення цілей дослідження.....	22
1.5 Аналіз логістичних процесів у сфері оптової торгівлі	25
1.5.1 Класифікація логістичних систем.....	25
1.5.2 Структурний аналіз	26
1.5.3 Функціональний аналіз.....	27
1.5.4 Параметричний аналіз	28
Постановка задачі	29
РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПРОЦЕСАМИ.....	32
2.1 Аналіз методів та моделей управління логістичними процесами	32
2.2 Використання оптимізаційних алгоритмів у логістиці	37
2.3 Етапи оптимізації логістичних процесів	41
2.4 Основні параметри інформаційної системи.....	43
2.5 Реалізація алгоритмів управління	45
2.5.1 Визначення ключових параметрів	46
2.5.2 Формування моделей логістичних процесів	47
2.5.3 Оцінка ефективності алгоритмів	48
2.5.4 Оптимізація маршрутів та процесів	50

2.5.5 Аналіз продуктивності системи	52
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ	55
3.1 Вибір стеку технологій	55
3.2 Загальна архітектура та структура програми	57
3.3 Розробка компонентів інформаційної системи	60
3.3.1 Підготовка вхідних даних	60
3.3.2 Інтерфейс користувача	61
3.3.3 Опис обчислювальної частини програми	62
3.3.4 Опис візуалізації логістичних процесів	63
3.4 Контрольний приклад системи	64
РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ	74
4.1 Вступ.....	74
4.2 Резюме	74
4.3 Опис проєктованого продукту	76
4.4 Оцінка ринку збуту	77
4.5 Аналіз конкуренції	79
4.6 Стратегія маркетингу	82
4.7 План впровадження системи.....	84
4.8 Організаційний план	87
4.9 Юридичний план	89
4.10 Фінансовий план.....	92
4.11 Стратегія фінансування	94
4.12 Оцінка ризику	95
4.13 Страхування	98

Висновки	100
Використаних література	102
Додаток А.....	105
Додаток Б. Презентація	

ВСТУП

Сучасні підприємства потребують значного рівня автоматизації процесів, особливо в логістичному секторі. Ефективне управління логістичними процесами дозволяє підприємствам заощаджувати витрати, покращувати обслуговування клієнтів і підвищувати конкурентоспроможність. Інформаційні системи в логістиці значно спрощують обробку замовлень, відстеження запасів, оптимізацію маршрутів та управління поставками.

З розвитком цифрових технологій оптові організації поступово впроваджують автоматизовані рішення для покращення своїх логістичних операцій. Інформаційні системи сприяють скороченню операційного часу, зменшують ризики людських помилок і пропонують швидкий доступ до аналітичних даних для прийняття обґрунтованих рішень.

Мета роботи: розробка інформаційної системи для управління логістичними процесами підприємств оптової торгівлі з метою підвищення ефективності та оптимізації ресурсів.

Об'єкт дослідження: процеси управління логістикою в оптовій торгівлі.

Предмет дослідження: методи, моделі та алгоритми управління логістичними процесами за допомогою інформаційних систем.

Методи дослідження: методи системного аналізу, інформаційні технології, алгоритми оптимізації, математичне моделювання.

У першому розділі проведено аналіз проблематики управління логістичними процесами, визначено основні виклики та перспективи застосування інформаційних систем у цій сфері.

У другому розділі розглядаються методи та моделі управління логістичними процесами, їх ефективність та можливості інтеграції інформаційних технологій.

Третій розділ присвячений програмній реалізації інформаційної системи, включаючи вибір технологій, архітектуру та створення компонентів програмного забезпечення.

У четвертому розділі проводиться техніко-економічне обґрунтування розробки, аналіз ринку, оцінка ризиків та фінансовий план впровадження системи.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка й аналіз проблеми

Логістика в оптовій торгівлі є важливим елементом для безперебійного функціонування фірми, оскільки вона гарантує швидку доставку товарів, оптимізацію запасів та зменшення транспортних витрат. Наразі підприємства оптової торгівлі стикаються з низкою перешкод у логістиці. Основними з них є неефективне управління запасами, затримки поставок, високі транспортні витрати та недостатня інтеграція логістичної діяльності в загальну систему управління бізнесом.

Традиційні методи управління логістикою, що покладаються на обробку даних людиною, не забезпечують необхідного рівня ефективності та точності. Відсутність цілісної інформаційної системи призводить до розбіжностей між підрозділами, а отже, до зменшення обсягів виробництва та збільшення витрат компанії. Ще однією проблемою є брак точних прогнозів попиту та оптимізації каналів постачання. Інформаційні технології сприяють автоматизації цих операцій, надаючи оперативну аналітику та зменшуючи ризик неточностей у плануванні.

Основні складові логістичних процесів (див. табл. 1.1) :

1. Закупівля та постачання

- Аналіз ринку постачальників та вибір оптимального партнера.
- Укладання контрактів та домовленості щодо умов постачання.
- Планування закупівель з урахуванням попиту та прогнозування ринкових коливань.

2. Складська логістика

- Організація складів, розробка оптимальних схем розміщення товарів.
- Використання сучасних методів зберігання (стелажні, блокові, конвеєрні системи).
- Автоматизація складських процесів за допомогою WMS-систем (Warehouse Management System).

3. Управління запасами

- Контроль залишків товарів, поповнення запасів за допомогою програм прогнозування.
- Використання методів Just-in-Time, FIFO, LIFO для мінімізації витрат та оптимізації процесів.
- Автоматизація обліку запасів та інтеграція з іншими бізнес-системами підприємства.

4. Транспортування та дистрибуція

- Оптимізація маршрутів доставки за допомогою TMS-систем (Transportation Management System).
- Використання GPS-моніторингу для контролю пересування вантажів.
- Автоматизація процесу оформлення документів та планування графіків поставок.

5. Зворотна логістика

- Організація процесу повернення товарів та управління рекамаціями.
- Переробка та утилізація продукції, що не підлягає реалізації.
- Впровадження екологічних ініціатив для зменшення відходів та раціонального використання ресурсів.

Інноваційні підходи до управління логістичними процесами

Сучасні підприємства активно впроваджують цифрові технології в управління логістичними процесами. Основні тенденції включають:

- **Впровадження автоматизованих систем управління** – ERP, WMS, TMS, SCM дозволяють централізовано керувати всіма аспектами логістики.
- **Використання штучного інтелекту та машинного навчання** – прогнозування попиту, аналіз ефективності маршрутів доставки, оптимізація складських процесів.
- **Блокчейн у логістиці** – забезпечення прозорості поставок, управління ланцюгами постачання та контроль автентичності товарів.

- **ІоТ (Інтернет речей) у транспортній логістиці** – GPS-трекери, датчики температури та вологості, контроль вантажів у реальному часі.

Таблиця 1.1

Основні аспекти логістичних процесів у оптовій торгівлі

Етап	Опис	Приклади технологій
Закупівля	Вибір постачальників, укладання контрактів, планування	ERP-системи, аналітика попиту
Складська логістика	Організація складу, оптимізація розміщення, автоматизація	WMS-системи, RFID, штрих-коди
Управління запасами	Контроль рівня товарів, прогнозування потреб	SCM-системи, BI-аналітика
Транспортування	Оптимізація маршрутів, контроль доставки	TMS-системи, GPS-моніторинг
Зворотна логістика	Управління поверненнями, утилізація товарів	Блокчейн, аналітичні системи

Впровадження сучасних інформаційних технологій у логістичні процеси дозволяє підвищити ефективність роботи підприємств, знизити витрати та покращити рівень обслуговування клієнтів.

1.2 Аналіз логістичних процесів у сфері оптової торгівлі

Оптова логістика охоплює повний цикл руху товарів від постачальника до кінцевого споживача, включаючи планування, доставку, зберігання, обробку замовлень та управління запасами. Розвиток цифрових технологій створює нові перспективи для автоматизації та оптимізації логістичних процесів, дозволяючи організаціям суттєво знизити витрати, підвищити ефективність та зберегти конкурентоспроможність на ринку.

Ключові логістичні процеси в оптовій торгівлі:

1. Управління запасами

Важливою частиною логістики є управління запасами, яке включає планування закупівель, зберігання та облік товарів. Невірне управління запасами призводить до втрат через надлишок або дефіцит товарів. Системи

для автоматизації цього процесу допомагають у прогнозуванні попиту та оптимізації рівня запасів.

2. Транспортування та доставка

Для забезпечення ефективності транспортування необхідно вибирати оптимальні маршрути, знижувати витрати на транспортування та своєчасно доставляти товар клієнтам. Інформаційні системи допомагають автоматично планувати маршрути, враховуючи відстань, час доставки та вартість.

3. Управління постачанням

Ключовим етапом є пошук надійних постачальників і контроль виконання умов договорів. Інформаційні системи автоматизують процеси вибору постачальників, моніторингу якості товарів і своєчасності поставок.

4. Обробка замовлень

Після отримання замовлення від клієнта необхідно швидко обробити його і підготувати товар для відправлення. Автоматизовані системи дають змогу скоротити час на обробку замовлень, знизити кількість помилок, забезпечуючи високий рівень обслуговування клієнтів.

5. Інвентаризція та контроль

Важливим є регулярне перевіряння залишків на складах та точне ведення обліку товарів. Автоматизація цих процесів дозволяє вчасно виявляти відхилення і забезпечувати точність даних.

Використання інформаційних систем у логістиці

Сучасні інформаційні системи мають вирішальне значення для покращення та вдосконалення логістичних процедур. Технології автоматизації дозволяють компаніям суттєво скоротити витрати, пришвидшити виконання завдань, покращити міжвідомчу комунікацію та забезпечити вищий рівень обслуговування клієнтів.

Найбільш популярними інформаційними системами, що застосовуються в оптовій торгівлі для управління логістичними процесами, є:

- WMS (Warehouse Management System) – система управління складами, що дозволяє ефективно управляти запасами, оптимізувати розподіл товарів по складу, автоматизувати процеси прийому, зберігання та відвантаження товарів.
- TMS (Transportation Management System) – система управління транспортуванням, яка допомагає оптимізувати маршрути доставки, вибір транспортних засобів, моніторинг транспорту і знижує витрати на транспортування.
- ERP (Enterprise Resource Planning) – інтегровані системи управління підприємством, що включають модулі для управління фінансами, людськими ресурсами, виробництвом та логістикою. ERP-системи дозволяють забезпечити взаємодію між різними підрозділами підприємства та знизити ризики через відсутність інтеграції.
- CRM (Customer Relationship Management) – система управління взаємовідносинами з клієнтами, яка дозволяє оптимізувати процеси взаємодії з клієнтами, контролювати виконання замовлень та підвищувати якість обслуговування.
- SCM (Supply Chain Management) – система управління ланцюгом постачань, яка інтегрує всі етапи від постачальника до кінцевого споживача, забезпечуючи оптимізацію поставок і зниження витрат (табл. 1.2)

Таблиця 1.2

Приклади інформаційних систем для управління логістичними процесами

Система	Опис	Основні функції
WMS	Система управління складами	Автоматизація обліку товарів, контроль за рухом запасів, оптимізація розміщення продукції
TMS	Система управління транспортуванням	Оптимізація маршрутів перевезень, контроль за транспортними засобами, моніторинг витрат на логістику

Продовження Таблиці 1.2

ERP	Інтегрована система для управління підприємством	Інтеграція всіх бізнес-процесів, управління фінансами, планування закупівель, аналітика
CRM	Система управління взаємовідносинами з клієнтами	Облік замовлень, прогнозування попиту, аналіз ефективності продажів
SCM	Система управління ланцюгом постачань	Координація роботи всіх учасників логістичного ланцюга, оптимізація закупівель і доставки

Застосування інформаційних систем у логістиці дозволяє підприємствам:

- Скоротити час обробки замовлень і підвищити точність виконання операцій.
- Оптимізувати витрати на транспортування і зберігання продукції.
- Автоматизувати планування та прогнозування попиту.
- Покращити взаємодію між усіма учасниками логістичного процесу.

Отже, впровадження сучасних інформаційних систем в логістику організацій оптової торгівлі є вирішальним фактором підвищення ефективності та конкурентоспроможності компаній. У цьому документі розглядаються методології та концепції управління логістичними процесами, а також практичні міркування щодо реалізації їх програм.

Виклики та перспективи розвитку логістичних процесів в оптовій торгівлі

Незважаючи на значний розвиток інформаційних технологій, логістичний сектор стикається зі специфічними перешкодами, серед яких:

- Нестача кваліфікованих спеціалістів для впровадження сучасних інформаційних систем.
- Високі витрати на впровадження автоматизованих логістичних платформ.
- Потреба в гнучкості та швидкій адаптації до змін ринку.

Перспективними напрямками розвитку логістичних процесів є:

- Використання штучного інтелекту та машинного навчання для оптимізації маршрутів і прогнозування попиту.
- Запровадження Інтернету речей (IoT) для відстеження товарів у режимі реального часу.
- Використання блокчейн-технологій для забезпечення прозорості та надійності поставань.

Тоживчення логістичних операцій в галузі оптової торгівлі підкреслює необхідність впровадження сучасних інформаційних технологій для підвищення ефективності, зниження витрат і поліпшення обслуговування клієнтів..

1.3 Огляд сучасних інформаційних систем для управління логістикою

Розвиток цифрових технологій призвів до винайдення програмних рішень, які значно покращують адміністрування логістичної діяльності. Сучасні інформаційні системи дозволяють регулювати рух товарів, оцінювати ефективність перевезень, управляти запасами та складськими операціями. Вони суттєво підвищують ефективність ланцюгів поставок, зменшують витрати та скорочують тривалість логістичної діяльності (Рис. 1.1).

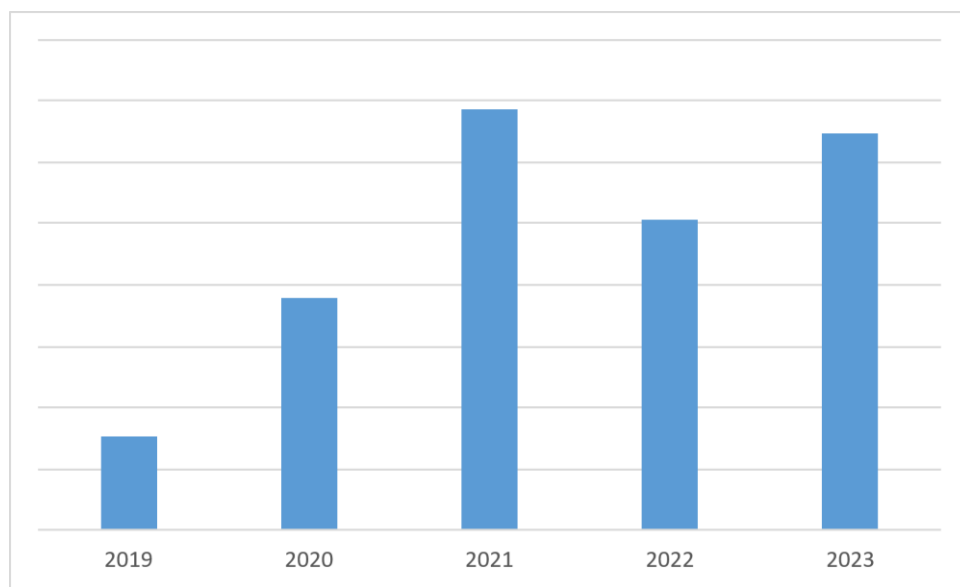


Рис. 1.1 - Статистика впровадження автоматизованих систем на підприємствах України

Класифікація інформаційних систем у логістиці

Інформаційні системи у сфері логістики можна поділити на такі категорії:

- **ERP (Enterprise Resource Planning)** – інтегровані платформи, що об'єднують усі бізнес-процеси, включаючи логістику, фінанси та управління персоналом. Вони забезпечують централізовану базу даних і спрощують координацію між різними відділами компанії. Приклади: SAP ERP, Oracle NetSuite, Microsoft Dynamics 365.
- **TMS (Transportation Management Systems)** – програми для управління транспортними перевезеннями, які дозволяють оптимізувати маршрути, контролювати витрати на доставку та стежити за транспортними засобами в реальному часі. Приклади: Blue Yonder TMS, Oracle Transportation Management, Manhattan TMS.
- **WMS (Warehouse Management Systems)** – рішення для ефективного управління складськими процесами, включаючи розміщення товарів, автоматизацію відвантажень і контроль за залишками. Приклади: Infor WMS, Fishbowl Inventory, HighJump WMS.
- **SCM (Supply Chain Management Systems)** – інструменти для управління ланцюгами поставок, що включають прогнозування попиту, автоматизацію замовлень і інтеграцію з постачальниками. Приклади: Kinaxis RapidResponse, Logility, SAP SCM.
- **BI (Business Intelligence) для логістики** – аналітичні платформи, які збирають, обробляють та візуалізують дані для ухвалення управлінських рішень. Вони дозволяють аналізувати продуктивність логістичних операцій та прогнозувати майбутні потреби. Приклади: Tableau, Microsoft Power BI, Qlik Sense.

Порівняння популярних інформаційних систем

На ринку доступні численні інформаційні системи для логістики, серед яких можна виділити найбільш відомі (табл 1.3-1.4).

Таблиця 1.3

Основні логістичні інформаційні системи

Тип системи	Основні функції	Приклади
ERP	Централізоване управління ресурсами підприємства	SAP ERP, Microsoft Dynamics 365
TMS	Оптимізація перевезень, моніторинг доставки	Blue Yonder TMS, Oracle Transportation Management
WMS	Автоматизація складських операцій	Infor WMS, HighJump WMS
SCM	Управління ланцюгами поставок, прогнозування попиту	Kinaxis RapidResponse, SAP SCM
BI	Аналітика та звітність для покращення логістики	Tableau, Power BI

Таблиця 1.4

Порівняння інформаційних систем

Система	Переваги	Недоліки
ERP	Централізоване управління ресурсами, автоматизація бухгалтерського та фінансового обліку, глибока інтеграція з іншими модулями, зменшення операційних витрат, підтримка великих обсягів даних	Дороге впровадження та обслуговування, складність налаштування, тривалий період впровадження, високі вимоги до персоналу
WMS	Оптимізація розміщення товарів, зменшення втрат та крадіжок, прискорення обробки замовлень, точний контроль залишків, інтеграція зі сканерами та RFID	Висока вартість, обмежена функціональність для транспортної логістики, потребує інтеграції з ERP для повної автоматизації
TMS	Оптимізація маршрутів, зниження витрат на транспортування, моніторинг доставки у реальному часі, вибір оптимального перевізника, автоматизація документообігу	Орієнтовані більше на великі компанії, потребує якісних вхідних даних, висока вартість, складність інтеграції з існуючими бізнес-процесами

Продовження таблиці 1.4

CRM	Покращене управління замовленнями клієнтів, аналітика та прогнозування потреб клієнтів, оптимізація комунікації з клієнтами, інтеграція з ERP та WMS, підвищення рівня обслуговування	Обмежена функціональність у сфері управління запасами та транспортуванням, може потребувати додаткових інтеграцій, висока вартість для малого бізнесу
SCM	Управління всіма етапами ланцюга постачань, прогнозування попиту, автоматизація замовлень, покращення взаємодії з постачальниками, аналіз ефективності поставок	Висока складність налаштування, потреба у точних аналітичних даних, дорога підтримка та оновлення, складність інтеграції з локальними рішеннями

Окрім загальних категорій, існують гібридні системи, які поєднують функціонал декількох напрямків. Наприклад, SAP SCM включає можливості ERP, SCM і BI, що робить його потужним інструментом для великих підприємств.

Тренди розвитку інформаційних систем у логістиці

Сучасні логістичні інформаційні системи постійно вдосконалюються, і на сьогодні можна виділити кілька ключових тенденцій у їхньому розвитку:

1. **Використання штучного інтелекту та машинного навчання** – дозволяє прогнозувати попит, оптимізувати маршрути, покращувати управління запасами.
2. **Автоматизація процесів** – інтеграція роботизованих систем для обробки вантажів, безпілотного транспорту.
3. **Хмарні рішення** – забезпечують доступ до даних у будь-який момент і спрощують масштабування логістичних систем.
4. **Інтернет речей (IoT)** – використання датчиків для моніторингу транспорту, вантажів та складів.
5. **Блокчейн у логістиці** – підвищує прозорість та надійність обліку товарів у логістичних мережах.
6. **Гіперавтоматизація** – інтеграція всіх доступних технологій для створення максимально ефективних логістичних процесів.

У світлі стрімкого розвитку технологій підприємства оптової торгівлі старанно впроваджують сучасні інформаційні системи для підвищення конкурентоспроможності та оптимізації своїх логістичних операцій. У майбутньому ми очікуємо більш глибокої інтеграції штучного інтелекту, автоматизованих систем управління та адаптивних алгоритмів прогнозування попиту, що значно підвищить ефективність логістичних операцій..

1.4 Визначення цілей дослідження

Основна мета дослідження - створити інформаційну систему для управління логістичними процесами в оптовій торгівлі. Це допоможе налагодити роботу та скоротити витрати. Інформаційна система має автоматизувати управління запасами, стежити за транспортом, знаходити оптимальні маршрути доставки та працювати з існуючими бізнес-процедурами компанії.

Для досягнення поставленої мети були визначені наступні завдання:

- 1. Проведення аналізу логістичних процесів** – дослідження основних проблем та труднощів у сфері логістики, що впливають на ефективність роботи підприємств оптової торгівлі.
- 2. Оцінка існуючих інформаційних систем** – аналіз поточних рішень на ринку та їхнього впливу на управління логістикою, а також виявлення їхніх переваг і недоліків.
- 3. Розробка концептуальної моделі інформаційної системи** – створення архітектури майбутньої системи, визначення ключових компонентів та їхньої взаємодії.
- 4. Формування ключових вимог до програмного забезпечення** – розробка технічних характеристик, необхідних для реалізації функціональності системи.
- 5. Проведення тестування та оцінка ефективності** – визначення впливу інформаційної системи на роботу підприємства після її впровадження.

6. **Дослідження інтеграції новітніх технологій** – вивчення можливостей використання штучного інтелекту, аналітики великих даних та Інтернету речей для підвищення продуктивності логістичних процесів.
7. **Визначення перспектив розвитку** – аналіз сучасних тенденцій та прогноз подальшого розвитку інформаційних технологій у сфері логістики.

Очікувані результати дослідження

Очікується, що результати дослідження сприятимуть:

- **Оптимізації бізнес-процесів підприємств оптової торгівлі** – зниженню операційних витрат, прискоренню виконання логістичних операцій та покращенню взаємодії між відділами.
- **Підвищенню рівня автоматизації** – створенню ефективної системи моніторингу та управління ресурсами підприємства.
- **Збільшенню конкурентоспроможності** – забезпеченню швидшої реакції на зміну ринкових умов та покращенню рівня обслуговування клієнтів.

Для наочного представлення основних завдань та взаємозв'язків між ними розроблено схему (Рис. 1.2), яка демонструє етапи роботи над інформаційною системою.



Рис.1.2 - Етапи розробки інформаційної системи для логістики

Для оцінки очікуваного впливу інформаційної системи доцільно провести порівняння традиційних логістичних процесів та запропонованого рішення (табл 1.5).

Таблиця 1.5

Порівняння традиційного та автоматизованого управління логістикою

Параметр	Традиційна логістика	Автоматизована логістика
Час обробки замовлень	Довгий, ручне введення даних	Швидкий, автоматичне оброблення
Оптимізація маршрутів	Відсутня або обмежена	Використання алгоритмів оптимізації
Контроль запасів	Виконується вручну	Автоматизований моніторинг
Рівень помилок	Високий через людський фактор	Мінімальний завдяки автоматизації
Вартість логістичних операцій	Висока через неефективне планування	Знижена завдяки оптимізації ресурсів

Для кращого розуміння впливу розробки інформаційної системи на роботу підприємства представлено стовпчикову діаграму (Рис. 1.3).

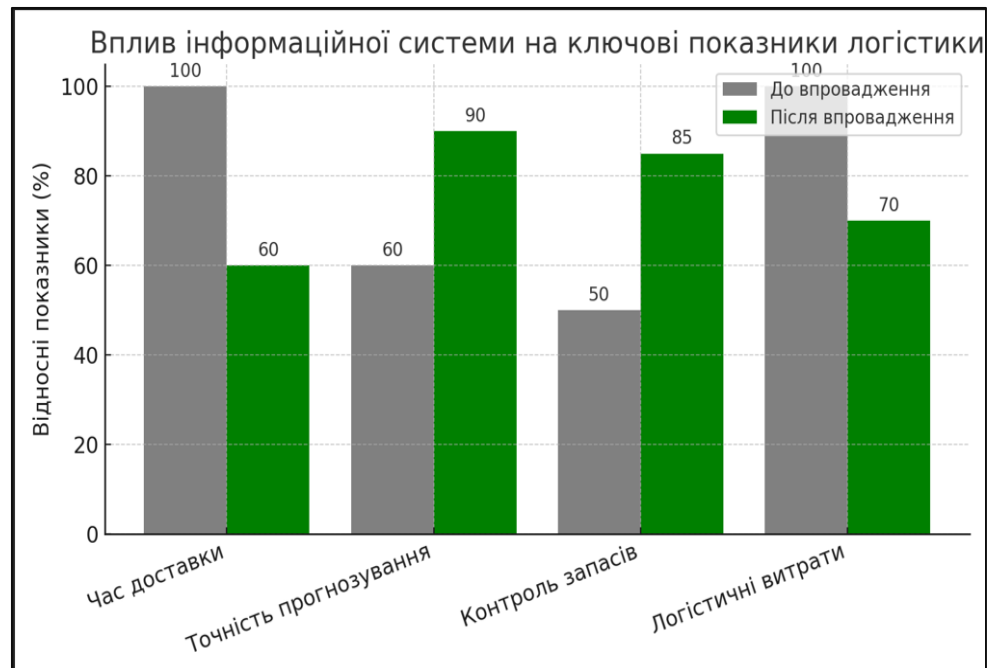


Рис.1.3 - Вплив інформаційної системи на ключові показники логістики

Таким чином, впровадження інформаційної системи дозволить суттєво підвищити ефективність логістичних процесів підприємств оптової торгівлі, що сприятиме їхньому розвитку та конкурентоспроможності на ринку.

1.5 Аналіз логістичних процесів у сфері оптової торгівлі

Аналіз системи управління логістичними процесами дозволяє визначити основні аспекти функціонування логістичних систем, їхню структуру, функціональність і параметри. Ефективне управління логістикою вимагає розуміння всіх рівнів логістичної системи, її можливостей та обмежень.

1.5.1 Класифікація логістичних систем

Логістичні системи поділяються на кілька основних типів залежно від масштабів, функцій та підходів до управління:

- **За рівнем автоматизації:**
 - Традиційні (з ручним управлінням)
 - Автоматизовані (часткова автоматизація процесів)
 - Інтегровані інформаційні системи (повна автоматизація)
- **За сферою застосування:**
 - Виробничі
 - Дистрибуційні
 - Складські
 - Транспортні
 - Комбіновані системи
- **За рівнем інтеграції:**
 - Локальні
 - Регіональні
 - Національні
 - Глобальні
- **За технологічною основою (див. табл 1.6):**
 - **ERP-системи** (SAP, Oracle) – комплексне управління підприємством

- **TMS (Transport Management Systems)** – управління транспортною логістикою
- **WMS (Warehouse Management Systems)** – управління складськими запасами
- **SCM (Supply Chain Management)** – управління ланцюгами постачання

Таблиця 1.6

Основні типи логістичних систем та їх особливості

Тип логістичної системи	Основні характеристики	Приклади
ERP-системи	Інтегроване управління ресурсами підприємства	SAP, Oracle
TMS	Оптимізація маршрутів, управління транспортом	Transporeon, Manhattan TMS
WMS	Автоматизація складських процесів	Infor WMS, HighJump
SCM	Управління всіма етапами ланцюга постачання	Blue Yonder, E2open

Ця класифікація допомагає визначити оптимальний підхід до організації логістичних процесів на підприємстві.

1.5.2 Структурний аналіз

Структурний аналіз логістичної системи передбачає розгляд її основних елементів та взаємозв'язків між ними (Рис. 1.4) Основні компоненти логістичної системи включають:

- **Постачальники** – забезпечують підприємство необхідними матеріалами та ресурсами.
- **Склади** – місця тимчасового зберігання продукції.
- **Транспортна мережа** – включає автомобільний, залізничний, морський та авіаційний транспорт.
- **Клієнти** – кінцеві споживачі продукції або послуг.
- **Інформаційні системи** – підтримують облік, моніторинг та управління логістичними потоками.

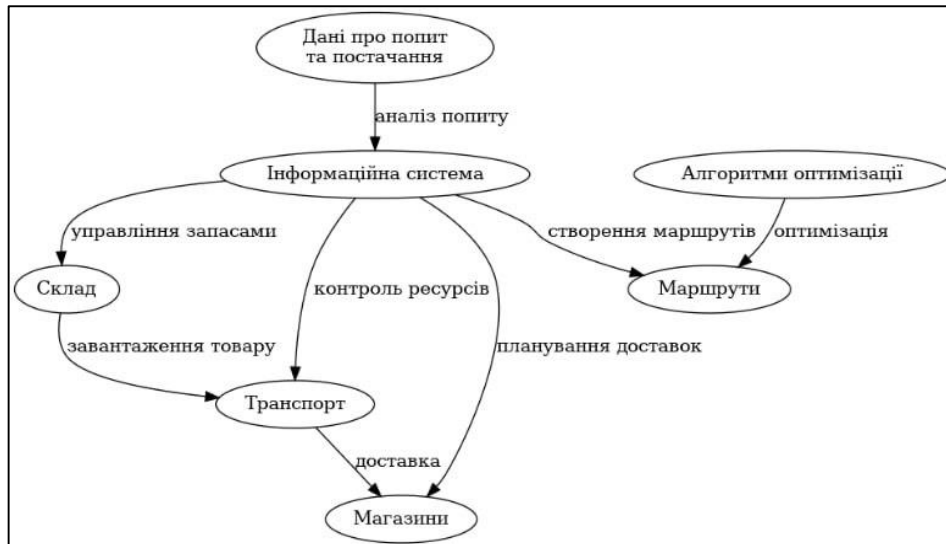


Рис. 1.4 – Взаємозв'язок основних елементів логістичної системи

(Візуалізація у вигляді схеми, що відображає взаємодію між постачальниками, складами, транспортною мережею та клієнтами)

Взаємодія між цими елементами забезпечує ефективність усієї логістичної системи.

1.5.3 Функціональний аналіз

Здійснивши функціональний аналіз логістичної системи, ви можете виявити найважливіші завдання та процеси, які необхідно автоматизувати, щоб зробити роботу більш безперебійною.

Ключовими завданнями логістичної системи є:

- **Планування постачань** – визначення потреб підприємства та замовлення товарів.
- **Керування запасами** – контроль рівня складських запасів та їхнє поповнення.
- **Оптимізація маршрутів** – вибір найкращих шляхів транспортування продукції.
- **Моніторинг логістичних процесів** – контроль виконання логістичних операцій у реальному часі.
- **Аналітика та прогнозування** – використання даних для прийняття ефективних рішень (табл. 1.7).

Таблиця 1.7

Основні функції логістичних систем та їх значення

Функція	Значення
Планування постачань	Зменшення затримок, ефективне управління ресурсами
Керування запасами	Оптимізація складських витрат, мінімізація дефіциту
Оптимізація маршрутів	Зниження витрат на транспорт, скорочення часу доставки
Моніторинг процесів	Підвищення контролю, швидке реагування на проблеми
Аналітика та прогнозування	Покращене прийняття рішень, мінімізація ризиків

Автоматизація цих функцій значно підвищує ефективність управління логістикою та зменшує витрати.

1.5.4 Параметричний аналіз

Мета параметричного аналізу - з'ясувати, які найважливіші характеристики логістичних систем і як вони впливають на загальну ефективність. Це основні фактори, які беруться до уваги:

- **Час виконання замовлення** – період від отримання заявки до доставки товару.
- **Витрати на логістику** – сукупні витрати, пов'язані з транспортуванням, зберіганням та обробкою замовлень.
- **Рівень автоматизації** – ступінь використання інформаційних систем для управління процесами (Рис. 1.5).
- **Пропускна здатність логістичних процесів** – максимальний обсяг товарів, які можуть бути оброблені за одиницю часу.
- **Точність прогнозування попиту** – рівень відповідності прогнозованих даних реальним показникам.

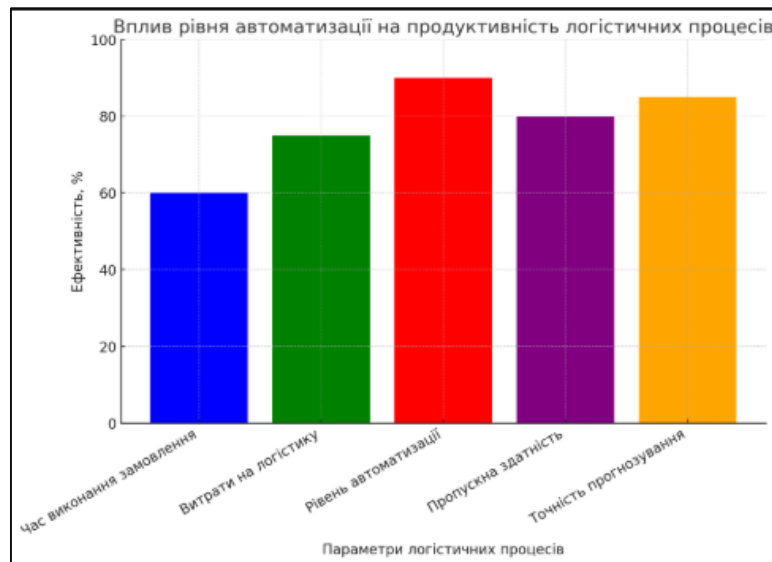


Рис.1.5 - Вплив рівня автоматизації на продуктивність логістичних процесів

Оцінка цих критеріїв сприяє оптимізації управління логістикою, підвищенню ефективності та зменшенню витрат.

Таким чином, аналіз логістичних систем дає змогу чітко визначити їхню класифікацію, структуру, функціональність та основні фактори. Використання сучасних інформаційних технологій підвищує ефективність управління логістичними процесами та зменшує витрати підприємств оптової торгівлі.

Постановка задачі

Існує багато завдань, які необхідно виконати, щоб створити інформаційну систему для управління логістикою підприємств оптової торгівлі. Ці обов'язки покликані зробити роботу більш безперебійною, найкращим чином використовувати ресурси та автоматизувати управлінські рішення.

Основні завдання дослідження:

- 1. Аналіз предметної області** – дослідити сучасні логістичні процеси у сфері оптової торгівлі, визначити основні проблеми та виклики, пов'язані з їх автоматизацією.
- 2. Визначення вимог до інформаційної системи** – сформулювати функціональні та нефункціональні вимоги, які повинна задовольняти розроблена система.

3. **Дослідження методів та моделей управління логістичними процесами** – розглянути можливі алгоритми та методики оптимізації процесів, вибрати найбільш ефективні підходи.
4. **Розробка концептуальної моделі системи** – створити структурну, функціональну та параметричну моделі управління логістикою.
5. **Програмна реалізація системи** – обрати стек технологій, розробити програмне забезпечення для обробки даних, управління логістичними процесами та візуалізації результатів.
6. **Тестування та апробація розробленої системи** – перевірити коректність роботи системи, оцінити ефективність алгоритмів та запропонувати можливі шляхи її покращення.
7. **Техніко-економічне обґрунтування** – провести оцінку економічної доцільності впровадження системи, розрахувати витрати на розробку та потенційні вигоди від її застосування.

Очікувані результати (Рис. 1.6):

- Автоматизація логістичних процесів для підвищення ефективності управління.
- Зменшення витрат на транспортування, зберігання та облік товарів.
- Покращення точності прогнозування попиту та оптимізації маршрутів.
- Інтеграція розробленої системи з існуючими ERP-системами підприємств.
- Створення зручного користувацького інтерфейсу для управління логістичними операціями.

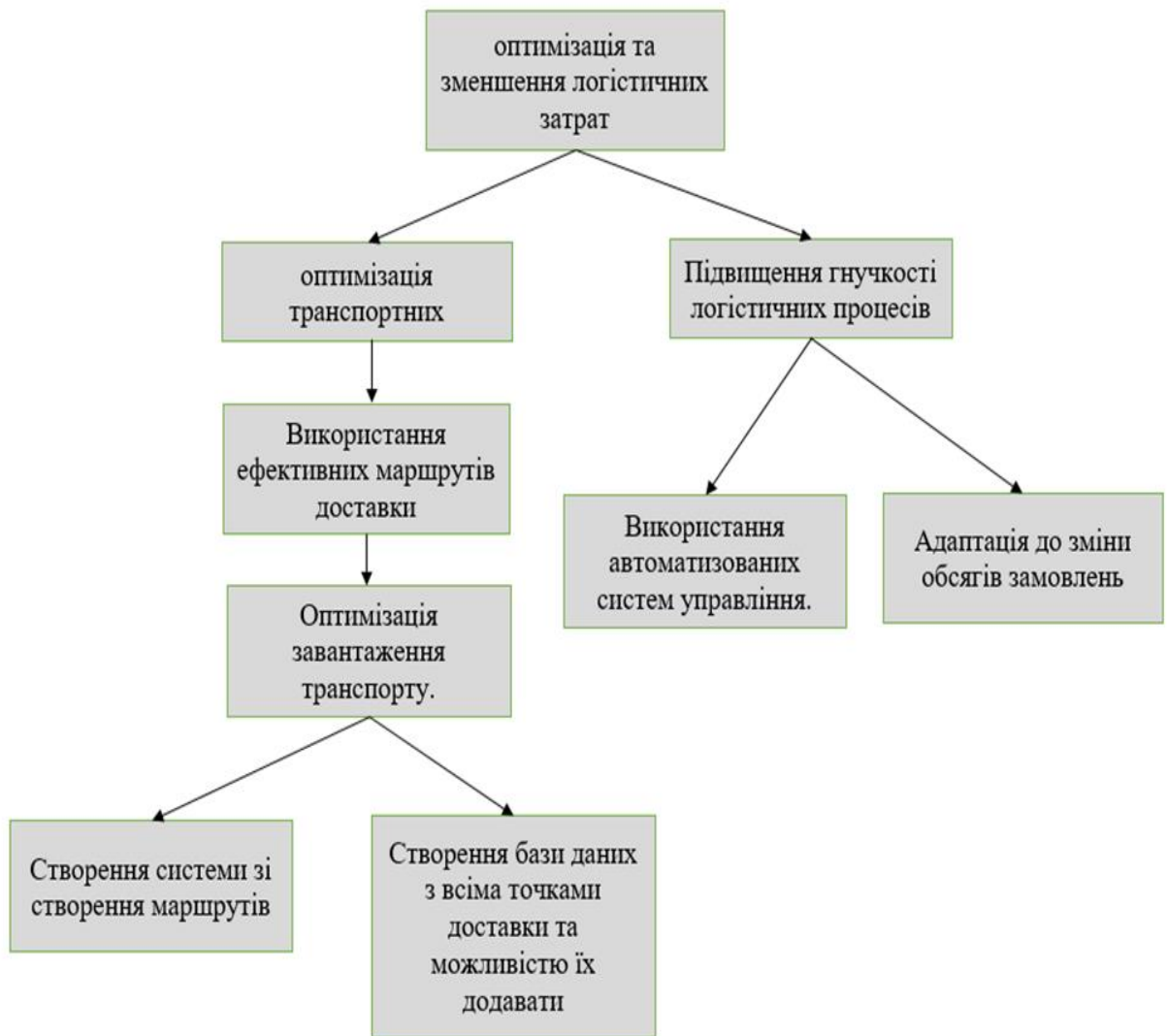


Рис 1.6 - Візуалізація очікуваних результатів

Запропонована інформаційна система допоможе підприємствам оптової торгівлі приймати кращі управлінські рішення та бути більш конкурентоспроможними на ринках, що швидко змінюються.

РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПРОЦЕСАМИ

2.1 Аналіз методів та моделей управління логістичними процесами

Управління логістичними процесами - це планування, організація, контроль і координація потоків грошей, інформації та товарів по всьому ланцюгу поставок. Ефективне управління дозволяє скоротити витрати, підвищити якість обслуговування клієнтів і бути гнучкими на ринку, який постійно змінюється. Вибір правильних підходів і моделей дуже важливий для досягнення цих цілей.

Класифікація методів управління логістикою

Методи управління логістичними процесами можна умовно поділити на:

- **Аналітичні** (математичні);
- **Евристичні** (на основі досвіду);
- **Імітаційні** (моделювання поведінки системи);
- **Інтелектуальні** (на базі штучного інтелекту);
- **Системні** (на основі інтегрованих підходів).

Кожна група має свої переваги та недоліки залежно від рівня складності задачі, обсягу даних, вимог до точності та часу обробки.

Аналітичні методи ґрунтуються на математичному апараті оптимізації, що дозволяє вирішувати такі питання, як розподіл ресурсів, маршрутизація та управління запасами. У складних або нестабільних системах ці стратегії можуть бути неефективними через обмеження динаміки змін..

Евристичні методи застосовують правила, засновані на емпіричному досвіді або раціональності. Серед поширених методів - алгоритми найближчого сусіда, жадібний алгоритм та сортування за пріоритетами. Хоча вони можуть не гарантувати оптимальних результатів, вони є ефективними та простими у використанні.

Імітаційне моделювання дозволяє створювати цифрові копії логістичних систем і тестувати різні сценарії без ризику. Це корисно для складських систем, транспортного планування та прогнозування навантажень.

Стохастичні та інтелектуальні методи використовують ймовірнісні моделі, штучний інтелект, машинне навчання, що дає змогу адаптуватися до непередбачуваних змін у середовищі (затримки, зміна попиту, відмова техніки).

Інтегровані системи підтримки прийняття рішень (DSS) поєднують декілька підходів і забезпечують автоматизацію вибору найкращих сценаріїв логістичної діяльності на основі моделювання, аналітики та рекомендаційних механізмів.

Методи та алгоритми, реалізовані в рамках розробки інформаційної системи

Враховуючи специфіку задачі - автоматизоване управління доставкою продукції з урахуванням обмежень по пропускній спроможності, типу вантажівки, пріоритету замовлення та оперативності - було побудовано низку прикладних алгоритмів з класу евристичних та комбінаторних методів.

Greedy (жадібний) алгоритм реалізовано для підбору маршрутів доставки. На кожному етапі обирається найбільш "вигідна" точка – з урахуванням мінімальної відстані та залишкової місткості вантажного засобу. Перевагою є швидкість прийняття рішень, однак можливий недосягнення глобального оптимуму (Рис. 2.1).



Рис.2.1 - Greedy (жадібний) алгоритм

Nearest Neighbor Heuristic використовується для вибору наступного пункту в маршруті на основі принципу найменшої відстані до ще необслугованих точок. Цей підхід особливо ефективний при великій кількості точок, зменшуючи загальний час обчислень (Рис. 2.2).



Рис.2.2 - Nearest Neighbor Heuristic

Кластеризація за категоріями впроваджена для логічного групування точок доставки за типом транспортного засобу або характеристиками вантажу. Це дозволяє зменшити кількість конфліктів і підвищити ефективність планування маршрутів (Рис. 2.3).

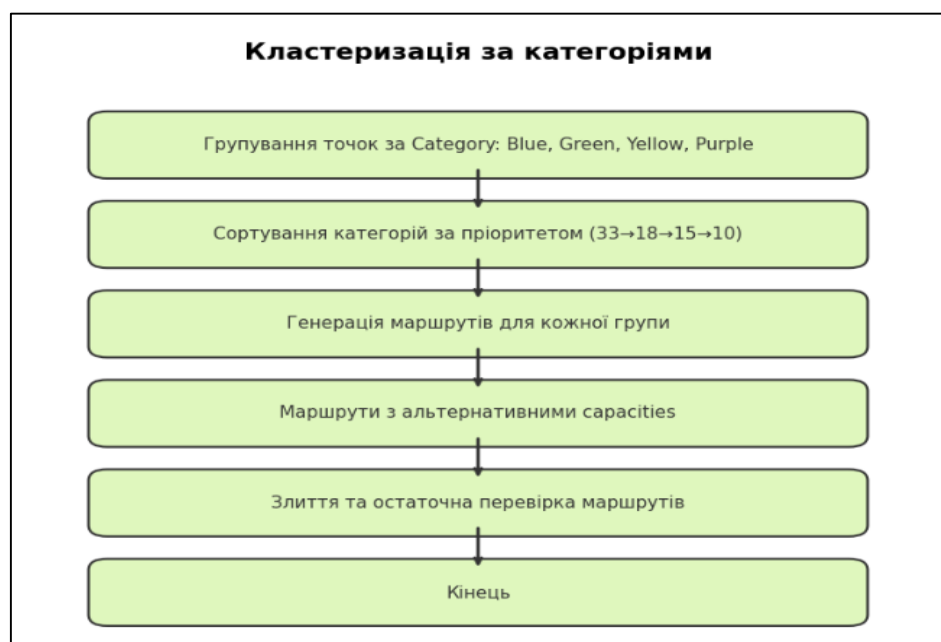


Рис.2.3 – Кластеризація за категорія

Sort-based Prioritization забезпечує сортування доступного транспорту за параметрами, такими як ступінь завантаженості, черговість, технічна справність. Завдяки цьому система гнучко реагує на зміни в обставинах (Рис. 2.4).



Рис.2.4 – *Sort-based Prioritization*(сортування вантажівок)

Failover Strategy дозволяє обробляти відмову вантажівок у реальному часі – шляхом миттєвого перерозподілу вантажів або зміни маршрутів. Це критично важливо в умовах реального виробництва чи доставки (Рис. 2.5).

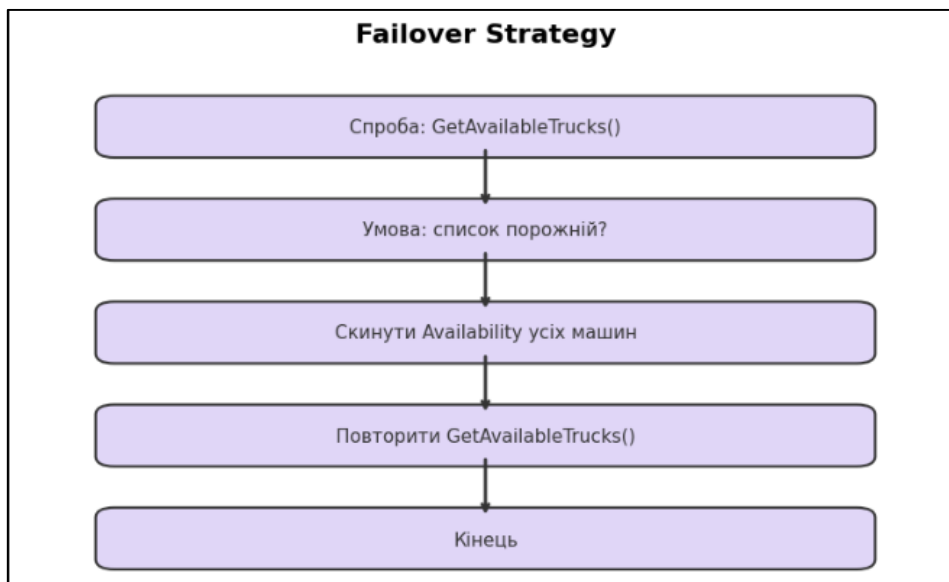


Рис.2.5 - *Failover Strategy*

Map-based оптимізація (через pointMap) — застосовується для ефективного доступу до даних про точки маршруту. Використання map-структур дозволяє швидко отримувати інформацію про координати, пріоритети, вантажність (Рис. 2.6).

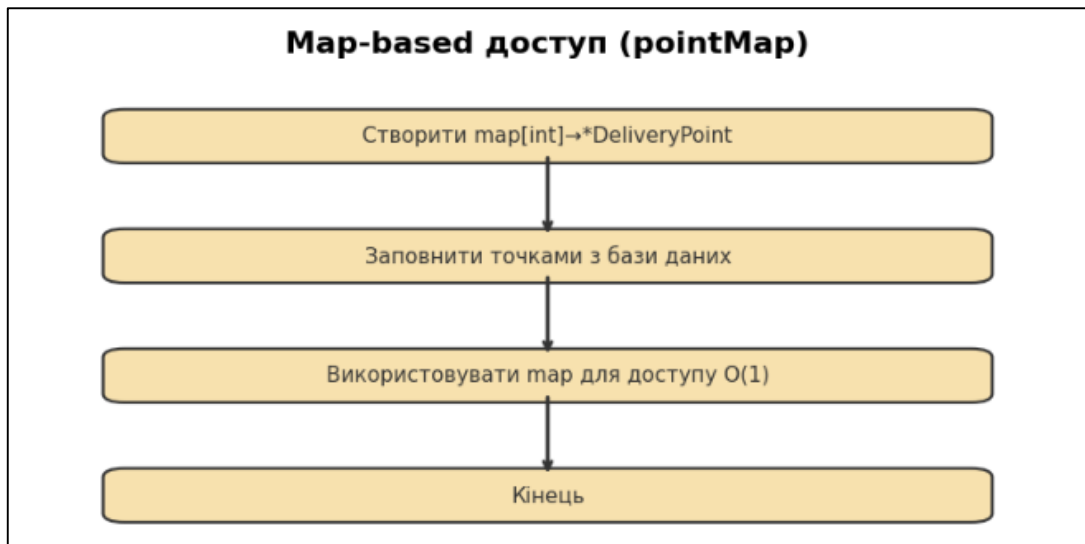


Рис.2.6 - Map-based оптимізація (через pointMap)

Capacity-aware splitting — це евристика, яка перевіряє залишкову місткість транспортного засобу перед додаванням вантажу. Якщо вантаж не вміщується повністю, він розділяється між кількома машинами з урахуванням логістичних обмежень (Рис. 2.7).

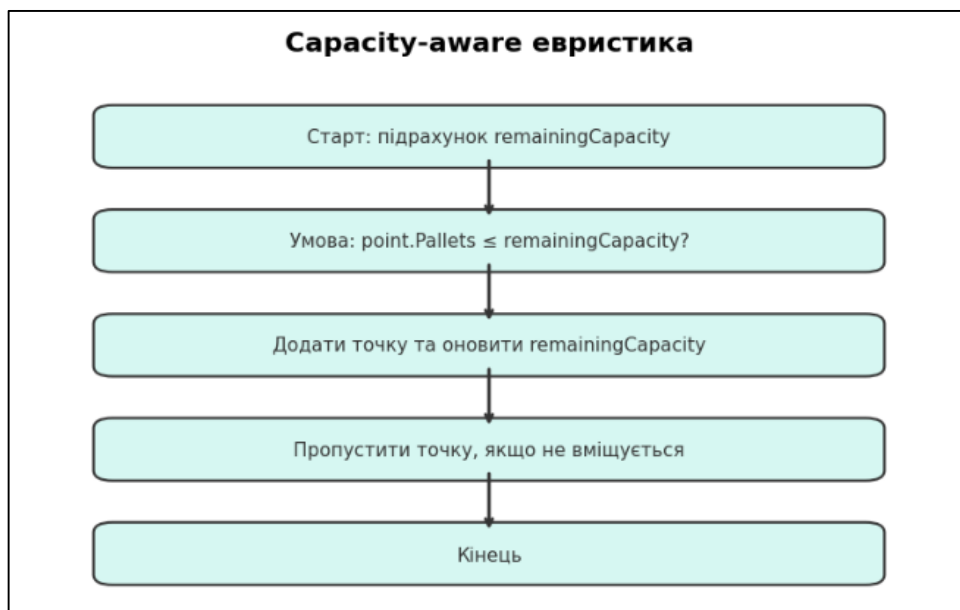


Рис.2.7 - Capacity-aware splitting

Всі ці методи були модифіковані відповідно до параметрів конкретної системи, створеної в цьому дослідженні. Їх інтеграція дозволила розробити масштабовану і надійну систему логістичного планування, здатну ефективно працювати в складних і динамічних умовах.

2.2 Використання оптимізаційних алгоритмів у логістиці

У сучасній логістиці оптимізація маршрутів, складування, розподілу ресурсів та керування запасами є ключовими факторами ефективного функціонування ланцюга постачання. Через збільшення обсягів даних, швидкозмінне середовище та високі вимоги до точності рішень, застосування оптимізаційних алгоритмів стало необхідністю.

Оптимізаційні алгоритми поділяються на точні, евристичні та метаевристичні. Точні алгоритми (лінійне програмування, метод гілок і меж) гарантують оптимальне рішення, проте вимагають значних обчислювальних ресурсів при масштабуванні (Рис. 2.8). Натомість евристики та метаевристики забезпечують прийнятну якість рішень у значно коротший термін, що робить їх практично незамінними у великих та динамічних логістичних задачах [34].

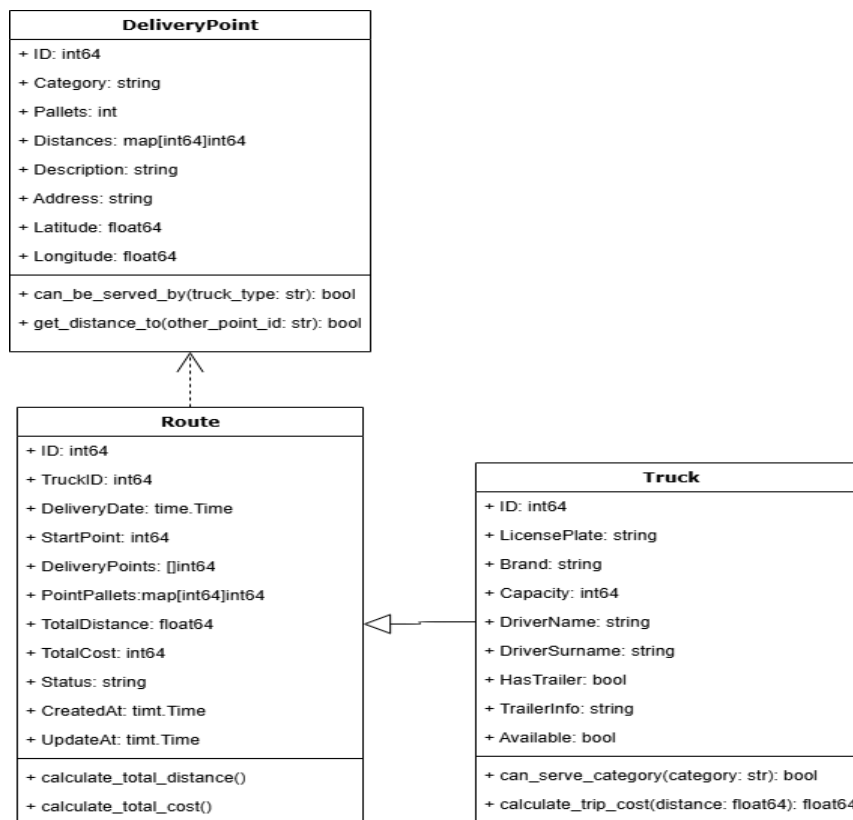


Рис.2.8 - UML подібна модель класів

Жадібні алгоритми (Greedy Algorithms)

Жадібні алгоритми приймають локально оптимальне рішення на кожному етапі з надією на досягнення глобального оптимуму. У логістиці їх застосовують для формування маршрутів доставки, коли необхідно швидко прийняти рішення щодо наступної точки маршруту. Наприклад, при доставці товарів із центрального складу до кількох точок система обирає ту, що найближча за відстанню або найменш навантажена.

Переваги таких алгоритмів полягають у швидкості, простоті реалізації та низькому навантаженні на обчислювальні ресурси. Проте, головним недоліком є ризик отримання субоптимального рішення, особливо у випадках складної геометрії маршруту чи різноманітності обмежень [19].

У межах даної роботи Greedy-алгоритм використано для побудови початкової конфігурації маршруту, яку надалі оптимізують інші модулі.

Метод найближчого сусіда (Nearest Neighbor)

Цей алгоритм є прикладом евристики, що базується на простому правилі: наступною точкою маршруту обирається та, яка розташована найближче до поточної. Метод широко застосовується у задачах типу комівояжера, коли потрібно обійти множину точок з мінімальними витратами [21].

Система, розроблена для дипломного проекту, інтегрує метод найближчого сусіда з оцінкою обмежень щодо місткості транспортних засобів. Це дозволило уникнути сценаріїв, в яких локація включається в маршрут з обсягом замовлення, що перевищує залишкову вантажопідйомність..

Кластеризація точок доставки

Для зниження складності розрахунків і підвищення ефективності використання транспорту реалізовано кластеризацію точок доставки за ознаками: тип транспортного засобу, категорія вантажу або відстань до складу. Цей підхід відповідає принципу поділу задачі на підзадачі, що легше розв'язуються окремо [32].

Так, наприклад, замовлення на великогабаритні товари класифікуються у окремий кластер і призначаються вантажівкам із підвищеною місткістю. Це дозволяє уникати дублювання маршрутів і оптимізувати завантаження транспорту.

Сортування за пріоритетом (Sort-based Prioritization)

Даний підхід передбачає визначення пріоритетності місць доставки або замовлень на основі певних факторів, таких як час створення, терміновість або близькість до складу. У дипломному проекті ми використовували сортування на основі ваги палети та визначеного пріоритету клієнта. Цей метод визначає пріоритетність найбільш необхідних речей для доставки, дотримуючись обмежень на транспортування.

Завдяки сортуванню зменшується ймовірність пропуску термінових замовлень, а також підвищується рівень обслуговування клієнтів. У поєднанні з кластеризацією це дозволяє формувати логічно послідовні, компактні маршрути, де пріоритет враховується вже на етапі планування [12].

Алгоритм відмови (Failover Strategy)

Логістичні процеси часто супроводжуються ризиками: поломки транспорту, затримки доставки, перевантаження складу тощо. Для підвищення стійкості системи реалізовано Failover-стратегію — алгоритм, який дозволяє відновити логістичну послідовність у разі виникнення збоїв.

У розробленій системі реалізовано автоматичну перевірку доступності транспортних засобів: якщо обрана вантажівка з якихось причин недоступна (ремонт, перевантаження), система обирає альтернативний транспорт із резервного пулу, враховуючи відповідність категорії вантажу та доступну вантажопідйомність.

Цей підхід підвищує надійність доставки та забезпечує безперервність процесу навіть за умов нестабільного середовища [1].

Map-based доступ до точок

Для пришвидшення доступу до точок маршруту та оптимізації часу обчислення використано карту розташування об'єктів, структуровану за

принципом дерева або хеш-таблиці. Замість повторного обчислення координат або відстаней, система звертається до попередньо побудованої індексованої карти.

Цей метод зменшує обчислювальну складність і дає змогу працювати з великою кількістю точок у режимі реального часу. Карта містить інформацію не лише про координати, а й про допустимі транспортні засоби, час роботи точки та допустиме вікно доставки [2].

Capacity-aware splitting (розподіл із урахуванням місткості)

У ситуаціях, коли обсяг замовлення перевищує доступну вантажопідйомність однієї машини, використовується алгоритм поділу замовлення на декілька доставок (splitting). На відміну від простого розділення, capacity-aware splitting враховує тип продукції, її сумісність з іншими вантажами, а також ефективність повторного відвідування точки.

Алгоритм спочатку аналізує, чи можна розділити вантаж на кілька палет без втрати якості чи порушення вимог. Згодом вона визначає найбільш економічний спосіб доставки: або сегментованими партіями з одного транспортного засобу, що робить багато рейсів, або з використанням декількох вантажівок одночасно.

Цей підхід дозволяє уникнути перевантаження транспорту, забезпечує виконання всіх замовлень без затримок і мінімізує втрати, пов'язані з відхиленням від логістичного графіка [10].

Загалом запропонована комбінація алгоритмів у межах створеної логістичної системи дозволяє ефективно вирішувати задачі маршрутизації, розподілу замовлень та резервування транспорту. Використання Greedy-підходу, методу найближчого сусіда та пріоритетного сортування забезпечує швидке формування початкових маршрутів. Кластеризація, failover-алгоритми та індексація точок оптимізують ресурси та підвищують стабільність системи. Capacity-aware splitting дозволяє точно враховувати обмеження і потреби клієнтів у нестандартних ситуаціях.

Власне таким чином, системне поєднання кількох оптимізаційних підходів дозволяє створити стійку до ризиків, масштабовану й адаптивну логістичну систему, що відповідає вимогам сучасної цифрової економіки.

2.3 Етапи оптимізації логістичних процесів

Оптимізація логістичних процесів — це багаторівневий підхід, який передбачає послідовне виконання етапів, спрямованих на виявлення неефективностей, покращення показників логістичної діяльності та адаптацію системи до змін зовнішнього середовища. Незважаючи на відмінності у моделях, типових завданнях і обсягах даних, структура оптимізації у сфері логістики зберігає певну логіку і послідовність кроків.

1. Аналіз поточних логістичних процесів

Першим етапом є повний аналіз існуючої логістичної системи. Він передбачає збір та обробку даних щодо маршрутів, складів, типів вантажів, часу перевезення, завантаженості транспортних засобів, рівня сервісу тощо. Цей аналіз дозволяє виявити "вузькі місця", дублювання дій, нераціональні маршрути або неефективне використання ресурсів.

У рамках дипломного проєкту на цьому етапі здійснювалася систематизація точок доставки та визначення базових параметрів: обсягу палет, типу вантажу, обмежень транспортних засобів. Важливим завданням було формування повної карти маршрутів із позначенням координат і категорій.

2. Постановка задачі оптимізації

Наступний етап — формалізація мети та вибір критеріїв оптимізації. У логістиці це може бути мінімізація відстані, часу доставки, кількості задіяного транспорту або вартості виконання логістичних операцій. Часто також враховується рівень сервісу, надійність поставок і екологічні аспекти.

У нашій системі головним критерієм було зменшення загального шляху транспорту при дотриманні місткості автомобіля та обмежень щодо типу вантажу. Це потребувало формулювання евристичних цілей і розробки відповідних алгоритмів.

3. Вибір і реалізація методів оптимізації

На цьому етапі здійснюється вибір конкретних моделей і методів, які будуть застосовані для вирішення задачі. Залежно від складності, обсягу даних і доступного часу це можуть бути:

- класичні математичні моделі (лінійне програмування),
- евристичні алгоритми (Greedy, Nearest Neighbor),
- комбіновані підходи з використанням кластеризації чи сортування,
- методи моделювання або машинного навчання.

Реалізація в нашому випадку включала побудову алгоритмів Greedy, Nearest Neighbor Heuristic, кластеризації за категоріями, Sort-based Prioritization тощо. Кожен з них реалізовувався як окрема функціональна одиниця з можливістю інтеграції в загальну систему.

4. Валідація результатів і тестування

Після отримання первинних результатів оптимізації проводиться оцінка їх ефективності. На цьому етапі необхідно порівняти нові показники з попередніми — зокрема за витратами, часом виконання, навантаженням транспорту. Також оцінюється стабільність результатів при зміні початкових умов (наприклад, збої, зміни у попиті, недоступність транспорту).

У нашому випадку оцінка проводилася за допомогою імітаційних сценаріїв, які відтворювали зміни в статусі доступності транспортних засобів (стратегія обхідних маршрутів) і зміни в обсягах вантажів.

5. Впровадження змін і подальший моніторинг

Останнім етапом є реалізація рекомендованих змін у реальній логістичній системі та організація постійного моніторингу. Це дозволяє адаптувати систему до змін середовища та вдосконалювати її з часом.

Для цього важливо розробити інтерфейс візуалізації маршрутів, динаміки завантаження транспорту та результатів роботи алгоритмів. У нашій системі ця функціональність реалізується через карту з точками, класифікованими за категоріями та маршрутами, та блоками статистики з підсумковими метриками.

2.4 Основні параметри інформаційної системи

Інформаційна система (ІС) у логістиці — це комплексна платформа, що забезпечує управління потоками товарів, інформації та фінансів, дозволяючи досягти високої ефективності на всіх етапах логістичного ланцюга. Сучасна логістика не мислиться без цифрових рішень, що дозволяють у режимі реального часу здійснювати контроль, аналіз та планування ресурсів [13].

Основні параметри логістичних ІС

Таблиця 2.1

Ключові параметри логістичних інформаційних систем

Параметр	Опис
Функціональність	Обсяг логістичних задач: управління складом, транспортом, замовленнями.
Масштабованість	Здатність до адаптації під зростання обсягів або нові підрозділи.
Інтегрованість	Сумісність з ERP, CRM, BI, TMS, WMS через API або хмарні сервіси.
Аналітичність	Можливість обробки Big Data, використання BI, машинного навчання.
Безпека	Захист даних відповідно до стандартів GDPR, ISO/IEC 27001.
Оперативність	Швидкість побудови маршрутів, обробки замовлень, генерації звітів.
Гнучкість інтерфейсу	Доступ з ПК, смартфонів; кастомізовані дашборди, багатомовність.

Ці параметри (табл. 2.1) формують базову вимогу до будь-якої сучасної логістичної ІС, яка має забезпечити оперативну реакцію на зміни в ланцюгу поставок, а також аналітичну підтримку для прийняття рішень.

Архітектура інформаційної системи

Нижче подано схематичне представлення типової структури логістичної ІС:

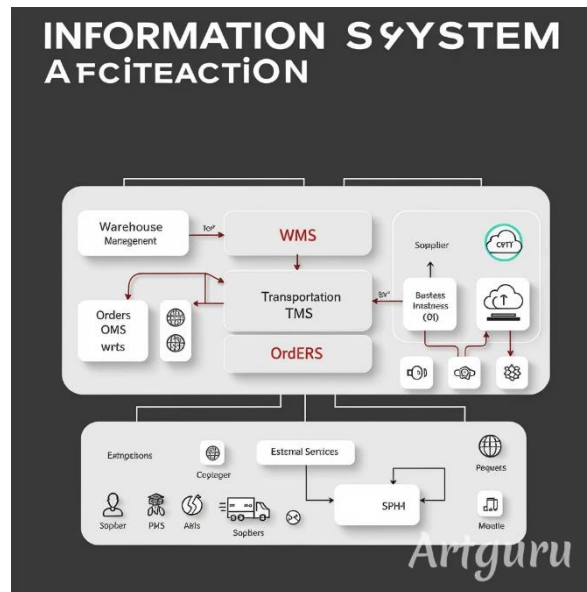


Рис.2.9 – Архітектура інформаційної системи

Пояснення до схеми:

- **Ядро системи** — складається з модулів: управління складом (WMS), транспортом (TMS), замовленнями (OMS), бізнес-аналітикою (BI).
- **Зовнішні сервіси** — ERP-системи, постачальники, клієнти, GPS-трекери, мобільні додатки.
- **Інтеграція** — здійснюється через API, REST-запити або хмарні інтеграційні платформи.

Порівняння поширених систем

Для визначення оптимального варіанту ІС для логістики доцільно порівняти характеристики популярних рішень (табл. 2.2):

Таблиця 2.2

Порівняння популярних логістичних інформаційних систем

Система	Платформа	Основні функції	Переваги	Недоліки
SAP S/4HANA	ERP, Cloud	WMS, TMS, SCM, фінанси	Висока інтеграція, масштабованість	Висока вартість впровадження

Продовження Таблиці 2.2

Oracle NetSuite	ERP Cloud	Ланцюги поставок, закупівлі	Автоматизація закупівель	Обмеження при масштабі
Infor SCM	Cloud, локальна	WMS, аналітика, оптимізація перевезень	Потужна аналітика	Складна інтеграція з деякими ERP
Zoho Inventory	SaaS	Замовлення, запаси	Підходить для малого бізнесу	Обмежена масштабованість
1С:Логістика	Локальна/хмара	Облік, маршрути, склад	Звичність для українського ринку	Обмежена інтеграція із західними системами

Приклад практичного використання

Кейс компанії ZARA

Іспанська мережа магазинів ZARA впровадила RFID-технології у свої логістичні процеси, інтегрувавши мітки у свою ІС. Це дало змогу:

- Скоротити час поставки до магазину з 5 до 2 днів.
- Виявляти дефіцит або надлишок ще до моменту формування замовлень.
- Підвищити точність прогнозування попиту до 90% [14],[39].

Цей приклад демонструє, що ІС є не лише інструментом автоматизації, а й засобом для підвищення стратегічної ефективності всієї компанії.

Отже, інформаційна система в логістиці — це багаторівнева структура, яка виконує одночасно аналітичні, управлінські, комунікаційні та захисні функції. Її ефективність безпосередньо залежить від інтеграції з іншими платформами, аналітичних можливостей та здатності масштабуватися під зростання бізнесу. Саме на основі таких систем можливе втілення складних алгоритмів маршрутизації, планування та реагування на ризики.

2.5 Реалізація алгоритмів управління

У цьому розділі розглянуто реалізацію ключових алгоритмів управління логістичними процесами в межах інформаційної системи. Вибрані методи мають забезпечити гнучке, адаптивне й ефективне управління маршрутами, ресурсами, черговістю обслуговування та відновленням роботи у разі збоїв.

2.5.1 *Визначення ключових параметрів*

Першим етапом побудови алгоритмів управління є чітке визначення параметрів, що впливають на логістичні процеси. Ці параметри виступають як вхідні дані для обчислень, моделювання та оптимізації.

До основних ключових параметрів належать:

Місткість транспортного засобу (Capacity): визначає максимальну кількість палет або об'єм, який можна перевезти за один рейс.

Тип вантажу: класифікація за категоріями (харчові продукти, побутова хімія, техніка тощо), що дозволяє групувати точки доставки з урахуванням вимог до зберігання або транспортування.

Часові вікна доставки: інтервали, у які має бути здійснена доставка до певного пункту. Цей параметр критично важливий для формування маршрутів.

Доступність точки доставки: враховує робочий графік точки та можливість фізичного доступу (наприклад, тимчасово закритий під'їзд).

Географічне положення: координати точок доставки, необхідні для геооптимізації маршрутів.

Пріоритетність обслуговування: визначає черговість включення точок у маршрут — залежно від критичності замовлення або договорів SLA (Service Level Agreement).

Навантаження логістичних ланок: дані про поточну зайнятість складів, транспортних одиниць, операторів.

Статус ресурсу: активність/недоступність певних транспортних одиниць, наявність відмов, технічні збої.

Вартість обслуговування: витрати на паливо, платні дороги, обслуговування персоналу тощо.

Ці параметри закріплюються у базі даних системи та автоматично оновлюються під час щоденного використання. Вони виступають базою для роботи наступних алгоритмічних блоків, які обирають найкращі рішення з можливих варіантів.

2.5.2 Формування моделей логістичних процесів

Формування логістичних моделей — ключовий етап, на якому відбувається перехід бізнес-завдань у математичні й алгоритмічні структури. Для підвищення ефективності логістичних процесів у системі реалізовано декілька підходів, які базуються на використанні алгоритмів оптимізації, кластеризації та евристик.

Одним із основних та базових методів є **Nearest Neighbor Heuristic**, який дозволяє формувати маршрути доставки шляхом вибору найближчої точки на кожному кроці. Така модель дозволяє зменшити сумарну відстань або час доставки. Формально це можна описати так:

$$d(i, j) = \min_{j \in N} \{d_{ij}\}, \quad (2.1)$$

де:

- i — поточна точка маршруту,
- j — кандидат на наступну точку маршруту з множини N ,
- d_{ij} — відстань між точками i та j .

Для запобігання перевищенню вантажопідйомності транспортного засобу використовується **Capacity-aware splitting**. При розподілі замовлень між машинами система перевіряє, чи не буде перевищено допустиме навантаження:

$$\sum_{k=1}^n \omega_k \leq C, \quad (2.2)$$

де:

- ω_k — вага k -го вантажу,
- C — максимальна місткість (вантажопідйомність) транспортного засобу.

Крім того, застосовується **кластеризація за категоріями вантажів**, яка дозволяє групувати точки доставки відповідно до типу продукції або вимог до транспортування. Така класифікація забезпечує логічне структурування маршрутів та полегшує подальшу оптимізацію. Модель кластеризації базується на принципі відстаней у багатовимірному просторі:

$$\min \sum_{i=1}^k \sum_{x_j \in C_i} \|x_i - \mu_i\|^2,$$

(2.3)

де:

- C_i — кластер,
- x_j — точка (замовлення),
- μ_i — центр кластера.

Для визначення порядку обробки заявок застосовується **Sort-based Prioritization**, що сортує вантажівки чи замовлення за ключовими показниками (відстань, тип товару, терміновість):

$$P(x) = \omega_1 \cdot t(x) + \omega_2 \cdot d(x) + \omega_3 \cdot u(x),$$

(2.4)

де:

- $P(x)$ — пріоритет заявки xxx,
- $t(x)$ — терміновість доставки,
- $d(x)$ — відстань,
- $u(x)$ — унікальні вимоги до транспортування,
- $\omega_1, \omega_2, \omega_3$ — вагові коефіцієнти.

Таким чином, моделі логістичних процесів базуються на комбінації евристичних, кластеризаційних та оптимізаційних підходів, що дозволяє формувати гнучкі та адаптивні маршрути з врахуванням реальних обмежень і умов.

2.5.3 Оцінка ефективності алгоритмів

Ефективність логістичних алгоритмів визначається за допомогою кількісних і якісних метрик, які дозволяють оцінити вплив використаних рішень на загальну продуктивність логістичної системи. Оцінювання проводиться за декількома ключовими напрямками: відстань, час доставки, використання ресурсів, надійність та економічні показники.

1. Мінімізація загальної відстані

Одним із пріоритетів є зменшення сукупної довжини маршрутів. Формула:

$$D_{total} = \sum_{i=1}^{n-1} d(p_i, p_{i+1}),$$
(2.5)

де:

- $d(p_i, p_{i+1})$ — відстань між точками маршруту i та $i + 1$,
- n — кількість точок у маршруті.

2. Середній час доставки (Delivery Time)

Оцінюється середній час, що витрачається на обробку і виконання замовлень:

$$T_{avg} = \frac{1}{m} \sum_{j=1}^m T_j,$$
(2.6)

де:

- T_j — час доставки j -го замовлення,
- m — загальна кількість доставок.

3. Коефіцієнт використання вантажопідйомності

Ця метрика демонструє, наскільки ефективно використовуються ресурси (машини):

$$U = \frac{\sum_{i=1}^k \omega_i}{k \cdot C} \times 100\%,$$
(2.7)

де:

- ω_i — вага вантажу у i -му транспортному засобі,
- C — місткість одного ТЗ,
- k — кількість ТЗ.

4. Надійність (Reliability Rate)

Цей показник відображає кількість успішно доставлених замовлень без порушень у графіку:

$$R = \frac{n_{success}}{n_{total}} \times 100\%,$$
(2.8)

де:

- $n_{success}$ — кількість доставок, що прибули вчасно,
- n_{total} — загальна кількість доставок.

5. Фінансова ефективність (економія витрат)

Визначається як різниця між витратами до і після впровадження алгоритмів:

$$S = \frac{C_{до} - C_{після}}{C_{до}} \times 100\% , \quad (2.9)$$

де:

- $C_{до}C_{після}$ — сума витрат до і після впровадження.

Загалом, використання цих формул дозволяє не лише зафіксувати досягнення, але й виявити вузькі місця, які потребують додаткової оптимізації. Комплексна оцінка алгоритмів забезпечує об'єктивне уявлення про ефективність інформаційної логістичної системи.

2.5.4 Оптимізація маршрутів та процесів

Оптимізація маршрутів і логістичних процесів є центральною задачею в управлінні ланцюгами постачання. Вона передбачає мінімізацію витрат, часу і ресурсів при одночасному забезпеченні високого рівня обслуговування клієнтів. У реалізованій інформаційній системі застосовано комбінований підхід, що включає кілька ефективних алгоритмів.

Застосовані алгоритми:

Greedy Algorithm (жадібний алгоритм)

Застосовується для попереднього формування маршрутів на основі найменших витрат у поточному кроці. Він дозволяє швидко знайти прийнятне рішення, яке згодом може бути покращене.

Nearest Neighbor Heuristic (найближчий сусід)

Алгоритм визначає наступну точку маршруту як найближчу від поточної. Він простий у реалізації й ефективний при невеликій кількості точок.

Кластеризація за категоріями

Точки доставки групуються за типом вантажівки або за географічними зонами. Це дозволяє уникнути перевантаження та забезпечити відповідність типу транспорту до вимог доставки.

Sort-based Prioritization (сортування за пріоритетом)

Усі замовлення сортуються за заданими критеріями — терміновість, відстань, тип товару. Це забезпечує ефективний розподіл ресурсів.

Map-based оптимізація доступу

Для кожної з кластеризованих зон використовується локальна карта доступу до точок. Це дозволяє уникати дублювань та покращує час обслуговування.

Capacity-aware splitting

Алгоритм дозволяє розділити маршрути відповідно до залишку палет у транспорті. Таким чином, вантажівка не перевантажується і не простоює.

Failover Strategy

Забезпечує перерозподіл маршрутів у разі непередбачених ситуацій — поломки транспорту, недоступності маршруту або відмови замовника.

Схема оптимізації маршрутів

Замовлення → Кластеризація → Пріоритезація → Вибір маршруту (Greedy/NN)
→ Перевірка місткості → Формування остаточного маршруту → Оптимізація за картою → Доставка

Приклад математичної мети оптимізації

Загальна мета — мінімізувати витрати на транспортування:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij},$$

(2.10)

де:

- c_{ij} — витрати на перевезення між пунктами iii та jjj ,
- $x_{ij} \in \{0,1\}$ — чи включено відрізок $i \rightarrow j$ до маршруту.

Застосування цих алгоритмів у єдиному середовищі забезпечує ефективне управління маршрутами, зменшує кількість пробігів, знижує витрати пального, а також покращує загальну швидкість виконання замовлень. Особливо важливою є здатність системи адаптувати маршрути в реальному часі.

2.5.5 Аналіз продуктивності системи

Оцінка продуктивності логістичної інформаційної системи є необхідним етапом для розуміння її ефективності, стабільності та здатності до масштабування. Такий аналіз дозволяє виявити вузькі місця, оцінити відповідність заданим критеріям і ухвалити рішення щодо подальшої оптимізації.

Основні метрики для оцінки продуктивності:

1. Час обробки маршруту

Визначає, скільки часу витрачається системою на розрахунок оптимального шляху з урахуванням усіх параметрів (дистанції, вантажопідйомності, обмежень).

$$T_{total} = T_{cluster} + T_{route} + T_{validate} , \quad (2.11)$$

2. Кількість оброблених замовлень за годину

Цей показник демонструє наскільки система здатна працювати в умовах високого навантаження (peak-time). Його порівнюють до і після впровадження алгоритмів.

3. Середній відсоток зменшення витрат на транспортування

Обчислюється як:

$$\Delta C = \frac{C_{до} - C_{після}}{C_{до}} \times 100\% , \quad (2.12)$$

де:

- $C_{до}$ — витрати до впровадження системи,
- $C_{після}$ - витрати після її застосування.

4. Рівень відмов і збоїв у системі

Вимірюється як кількість аварійних відмов чи непропрацьованих запитів:

$$R_{fail} = \frac{N_{\text{помилки}}}{N_{\text{усіх запитів}}} \times 100\%,$$
(2.13)

5. Рівень використання транспортних потужностей

Визначає, наскільки ефективно використовується вантажопідйомність кожної одиниці транспорту.

$$U = \frac{\sum_{i=1}^n W_i}{\sum_{i=1}^n C_i} \times 100\%,$$
(2.14)

де:

- W_i — фактичне навантаження,
- C_i - вантажопідйомність.

Графічна візуалізація продуктивності

Система може виводити діаграми та графіки, що показують:

- динаміку зміни часу доставки;
- співвідношення запитів і відмов;
- навантаження на транспорт у реальному часі;
- економію ресурсів після впровадження алгоритмів.

Приклад:

Після впровадження розглянутих алгоритмів підприємство отримало наступні результати:

- Скорочення середньої тривалості доставки з 3,2 годин до 2,1 години.
- Підвищення рівня використання транспорту до 88%.
- Зменшення витрат на логістику на 18%.
- Зниження кількості відмов системи до 0,3% на 1000 запитів.

Аналіз продуктивності інформаційної системи демонструє її дієвість і доцільність застосування розроблених алгоритмів. Виявлені переваги підтверджують, що обрана архітектура й оптимізаційні механізми дозволяють

досягти високих результатів у логістичних операціях. Отримані дані можуть використовуватися для подальшого навчання моделей машинного навчання, прогнозування навантаження та автоматизації управлінських рішень.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір стеку технологій

У цьому підрозділі обґрунтовується вибір технологій для реалізації інформаційної логістичної системи, зважаючи на вимоги до продуктивності, масштабованості, швидкості розробки та зручності підтримки.

1. Мова програмування Go

- Продуктивність і конкуренція

Go компілюється в машинний код без проміжного середовища, що забезпечує низькі накладні витрати та високу швидкість виконання. Вбудована підтримка конкурентності (goroutines та канали) дозволяє просто та ефективно обробляти асинхронні задачі (наприклад, паралельний розрахунок маршрутів) без складної ручної роботи з потоками.

- Простота деплою

Збірка виконується в один статичний бінарний файл, який можна запускати на сервері без встановлення додаткових залежностей. Це спрощує налаштування середовища продуктивності та дозволяє швидко розгортати оновлення.

- Стандартна бібліотека

Містить потужні пакети для роботи з мережею (HTTP), серіалізації (JSON), логування та тестування. Це зменшує кількість сторонніх залежностей і підвищує стабільність проєкту.

- Статична типізація

Допомагає виявляти помилки на етапі компіляції, що підвищує надійність коду та полегшує рефакторинг.

2. Фреймворк React + TypeScript

- Декларативний інтерфейс

React дозволяє описувати UI у вигляді компонентів, які автоматично оновлюються при зміні стану. Це спрощує роботу з динамічними елементами (таблицями, формами, карти).

- Строга типізація

TypeScript гарантує коректність передачі props і станів між компонентами, знижує ймовірність runtime-помилки у фронтенд-частині. При цьому редактор (IDE) підказує можливі параметри та автодоповнює код.

- **Велика екосистема**

Пакети для форм (Formik, React Hook Form), валідації (Yup), інтеграції з картами (React Leaflet) та графіків (Recharts) пришвидшують розробку і забезпечують готові, добре документовані рішення.

- **Гарячий перезапуск (Hot Reload)**

Швидкий цикл редагування–тестування змін у браузері пришвидшує фронтенд-розробку.

3. Система управління базами даних MongoDB

- **Документно-орієнтована модель**

Дані про точки доставки, маршрути та звіти різняться за структурою: MongoDB дозволяє зберігати гнучкі документи JSON без жорстких SQL-схем, що особливо корисно при еволюції вимог.

- **Масштабованість**

Горизонтальне масштабування через шардінг і вбудована реплікація забезпечують високу доступність та можливість обробки великих обсягів даних.

- **Простота використання**

Офіційні клієнти Go мають простий API для CRUD-операцій, індексування та агрегаційних запитів, що пришвидшує розробку шарів доступу до даних.

4. Контейнеризація за допомогою Docker

- **Однорідність середовища**

Завдяки Docker усі залежності (Go runtime, Node.js для фронтенд-білду) пакуються у контейнери, що усуває проблему «працює на моїй машині, але не на сервері».

- **Швидке розгортання**

Образи створюються за допомогою Dockerfile; команди docker-compose up піднімають одночасно сервіс, базу даних та фронтенд. Це зменшує час підготовки середовища для розробки та тестування.

- **Ізоляція**

Кожен сервіс (API, база даних, фронтенд) працює у власному контейнері з чітко визначеними портами і змінними середовища, що спрощує відлагодження та моніторинг.

5. Утилітарний CSS-фреймворк Tailwind CSS

- **Мінімум кастомного CSS**

Замість написання сотень декларацій стилів використовуються утилітарні класи (p-4, bg-blue-500, rounded-lg), що скорочує розмір CSS і пришвидшує верстку.

- **Конфігурованість**

Через tailwind.config.js можна адаптувати палітру кольорів і відступів під корпоративний стиль без переписування класів у компонентах.

- **Підтримка JIT-режиму**

Автоматичне генерування тільки тих класів, які використовуються в коді, що оптимізує кінцевий CSS-файл за об'ємом.

6. Додаткові інструменти

- **Makefile / Taskfile.yaml**

Скрипти для автоматизації рутинних завдань: запуск локального середовища, міграції бази, збірка фронтенду, тестування. Одною командою можна провести повний цикл «build→test→run».

- **ESLint та Prettier**

Застосовано для уніфікації стилю коду у фронтенд-частині, що полегшує командну роботу та скорочує час код-рев'ю.

3.2 Загальна архітектура та структура програми

Інформаційна система для управління логістичними процесами розроблена за принципами **чотирирівневої архітектури** (*four-tier architecture*), що передбачає розподіл компонентів системи на чотири логічні рівні: клієнтський

(презентаційний), серверний, доменний (логіки) та рівень бази даних. Такий підхід забезпечує масштабованість, модульність і зручність супроводу програми [13].

Рівні архітектури :

1. Presentation Layer (Фронтенд-частина)

Фронтенд реалізовано як окремий React+TypeScript додаток у каталозі web (див. «Folders and files» у корені репозиторію) GitHub. Цей шар відповідає за інтерфейс користувача, валідацію введених даних у формах і відображення результатів (таблиці, карта, графіки).

2. API Layer (HTTP-сервер)

Розміщений у папці cmd/api/main.go (що є точкою входу) і в каталозі web/backend. HTTP-обробники приймають запити від UI, транслюють їх у виклики внутрішньої бізнес-логіки та повертають JSON-відповіді [GitHub](#).

3. Business Logic Layer (Доменна логіка)

Інкапсульована в директорії internal. Тут знаходяться пакети для обчислення маршрутів (internal/route), формування звітів (internal/report), роботи з конфігурацією (internal/config) та моделями даних для MongoDB (internal/db/models.go) [GitHub](#). Ці пакети не залежать від зовнішніх фреймворків і можуть бути протестовані ізольовано.

4. Data Access Layer (Доступ до бази даних)

Реалізовано через пакети в internal/db. Вони інкапслюють підключення до MongoDB, CRUD-операції та конвертацію між доменними об'єктами Go і BSON-документами [GitHub](#).

5. Configuration & Infrastructure

Файл .env і .env.test в корені проекту містять змінні середовища для підключення до бази даних і налаштувань сервера [GitHub](#). Контейнеризацію та одночасний запуск кількох сервісів реалізовано через docker-compose.yml, а збірку й тестування автоматизовано в Taskfile.yaml [GitHub](#).

Ця чотирирівнева архітектура спрощує розробку: кожен шар чітко обмежує свою відповідальність, мінімізує залежності та дозволяє паралельно розвивати UI, API та бізнес-логіку (Рис. 1.3).

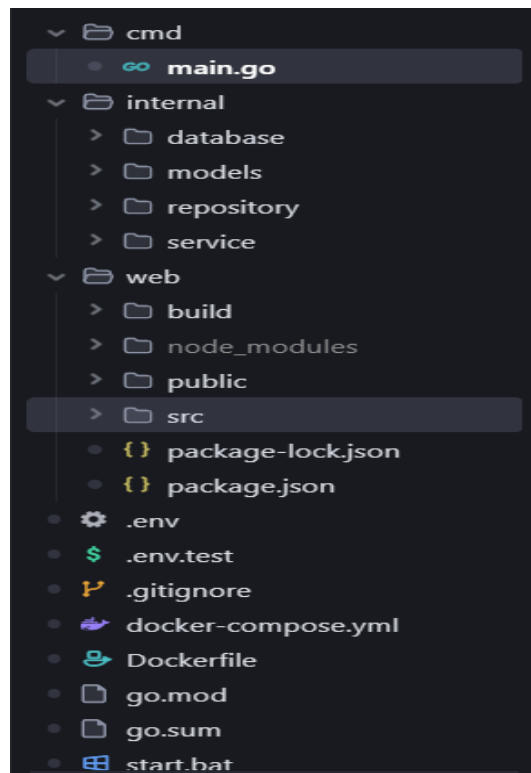


Рис.3.1 - Структура проекту

- **cmd/api**: містить точку входу main.go, яка ініціалізує конфігурацію, підключає MongoDB, налаштовує маршрути HTTP і запускає сервер GitHub.
- **internal**: містить усю бізнес-логіку та пакети, що не призначені для зовнішнього імпорту. Це гарантує інкапсуляцію внутрішніх компонентів та запобігає випадковому використанню доменного коду в інших репозиторіях GitHub.
- **web/frontend**: окремий проект на React+TypeScript, який збирається у статичні файли та обслуговується сервером або CDN. Використовує Tailwind CSS для швидкої верстки та підтримує hot-reload для комфортної розробки GitHub.
- **web/backend**: містить HTTP-обробники, які перетворюють зовнішні запити в виклики use case з internal/route та internal/report. Забезпечує розділення відповідальності між UI-логікою та бізнес-логікою GitHub.

- **Dockerfile & docker-compose.yml:** дозволяють розгорнути систему локально або в production-середовищі з мінімальною конфігурацією, створюючи однакове середовище для всіх розробників і серверів.
- **Taskfile.yaml:** описує набори команд (build, test, lint, run), що спрощують запуск проєкту та гарантують єдину процедуру для CI/CD.

Взаємодія компонентів :

- **Завантаження конфігурації:** при старті main.go зчитує .env через пакет internal/config, будує структуру Config і передає її далі в модулі підключення до БД та налаштування HTTP-серверу.
- **Запити з UI:** React-додаток виконує AJAX-запити до ендпоінтів /api/..., які обробляються у web/backend/handlers.go, розпаковують JSON-тіло, викликають відповідний метод з internal/route або internal/report і повертають результат назад.
- **Пакет internal/route:** реалізує алгоритм побудови маршруту на основі вхідних даних (список точок, ємності авто), повертає структуру []Route із деталями шляху та підсумковим пробігом.
- **Доступ до бази:** пакети internal/db надають методи для збереження та читання документів про точки й звіти, інкапсулюючи клієнт MongoDB та BSON-сериалізацію.
- **Візуалізація результатів:** UI отримує дані маршрутів й звітів у форматі JSON і відображає їх у таблицях, на карті (React Leaflet) та в графіках (Recharts).

3.3 Розробка компонентів інформаційної системи

3.3.1 Підготовка вхідних даних

Для коректної роботи системи необхідно забезпечити єдине конфігураційне середовище та зручне підняття залежних сервісів.

1. Файл .env

Використання бібліотеки dotenv дозволяє зберігати всі параметри підключення (MongoDB URI, порт API-сервера, секретні ключі) у файлі .env та завантажувати їх у середовище виконання при старті програми [Dotenv](#).

2. Структура .env :

```

1  # MongoDB Connection
2  MONGO_URI=mongodb://localhost:27017
3  DB_NAME=logistics_db
4
5  # Server Configuration
6  PORT=6060
7
8  # Application Settings
9  GIN_MODE=debug

```

Рис.3.2 - Структура env

- Така організація відповідає принципам дванадцятифакторного додатка, відокремлюючи конфігурацію від коду [npm](#).

- **Docker Compose**

Для одночасного запуску сервісу API, бази даних та фронтенду використано docker-compose.yml, що описує три сервіси: app (Go-API), mongo (MongoDB) і web (React UI) [Docker Documentation](#).

- **Переваги контейнеризації**

Docker Compose гарантує ізоляцію середовищ, швидке оновлення образів та відсутність конфліктів залежностей між різними компонентами системи

3.3.2 Інтерфейс користувача

Фронтенд-частина реалізована на React з TypeScript, що забезпечує декларативний рендеринг і сильну типізацію.

1. Структура компонентів

Використано функціональні React-компоненти з хуками для управління станом та ефектами, розташовані у каталозі web/frontend/src/components.

2. Форми та валідація

Для керування формами донесення даних про поставки застосовано Formik із схемою валідації на основі Yup. Formik спрощує роботу з формами, автоматично обробляючи стани полів і помилки, а Yup дозволяє описувати складну логіку валідації за допомогою декларативної схеми Formik Medium.

3. Карта доставки

Інтерактивну карту реалізовано за допомогою React Leaflet. Компонент `<MapContainer>` ініціалізує карту, а `<CircleMarker>` відображає точки залежно від їх типу (колір маркеру відповідає ємності авто) react-leaflet.js.org.

4. Графіки пробігу

Для візуалізації пробігу кожного авто застосовано Recharts. Компонент `<BarChart>` отримує дані з API та креслить стовпчикову діаграму пробігу в кілометрах recharts.org.

5. UX-покращення

Використання Hot Reload та ESLint/Prettier дозволяє миттєво бачити зміни в UI і підтримувати єдиний стиль коду.

3.3.3 Опис обчислювальної частини програми

Бізнес-логіка побудови маршрутів інкапсульована в пакеті `internal/route`.

1. Структура даних

```
type Point struct {
    ID      int
    Pallets int
    Distance float64
}

type Vehicle struct {
    Capacity int
    Used     int
}
```

Рис.3.3 - Структура даних

- **Алгоритм Nearest Neighbor**

Використано жадібний алгоритм, який починається від депо і на кожному кроці додає до поточного маршруту найближчу невідвідану точку, поки не вичерпається ємність авто або список точок Medium.

- **Паралельна обробка**

Для прискорення розрахунків маршрути для різних авто можуть обчислюватися в паралельних goroutines із синхронізацією через канали Medium gobyexample.com.

- **Взаємодія з БД**

Пакет `internal/db` використовує офіційний MongoDB Go Driver для CRUD-операцій з колекціями `points` та `routes`.

3.3.4 Опис візуалізації логістичних процесів

Після обчислення маршрутів система передає клієнту готові дані, які відображаються у зручному вигляді.

1. **Відображення маршрутів на карті**

Лінії маршрутів малюються компонентом `<Polyline>` React Leaflet на тій же карті, де розміщені `<CircleMarker>`; кольори ліній відповідають різним транспортним засобам.

2. **Діаграми та звіти**

Використано Recharts для побудови як стовпчикових, так і лінійних графіків порівняння пробігу та кількості точок у маршрутах.

3. **Таблиці звітності**

Компонент `react-table` забезпечує сортування, фільтрацію та пагінацію даних звіту про доставку за день.

4. **Адаптивність**

Tailwind CSS гарантує, що усі UI-елементи коректно відображаються як на десктопі, так і на мобільних пристроях.

3.4 Контрольний приклад системи

У цьому розділі наведено покроковий приклад запуску та перевірки роботи логістичної системи на тестових даних. Наведені вхідні дані, детальний опис процесу обробки, результати обчислення та спосіб їх візуалізації в інтерфейсі. (Додаток А)

Для демонстрації та перевірки коректності алгоритму маршрутизації обираємо невеликий набір тестових даних (Рис. 3.4):

- **Точки доставки :**

1. 1001 Kyiv, Troieshchyna, Bratyslavska St, 11 – 33 палети
2. 1011 Kyiv, Khreshchatyk St, 44 – 10 палет
3. 1013 Kyiv, Vadyma Hetmana St, 6 – 5 палет
4. 1012 Kyiv, Druzhby Narodiv Blvd, 16A – 2 палети
5. 1044 Kyiv, Saksahanskoho St, 64 – 4 палети
6. 1045 Kyiv, Khreshchatyk St, 15 – 5 палет

Delivery Points

Category

Green ▼

Category Information

<div style="background-color: #007bff; color: white; border-radius: 50%; width: 20px; height: 10px; margin: 0 auto; display: inline-block;"></div> Blue 33-pallet trucks (with trailers) and 18-pallet trucks Truck Types: Large trucks with trailers (33 pallets) or medium trucks (18 pallets)	<div style="background-color: #28a745; color: white; border-radius: 50%; width: 20px; height: 10px; margin: 0 auto; display: inline-block;"></div> Green 18-pallet trucks and smaller Truck Types: Medium trucks (18 pallets) or smaller trucks (15 or 10 pallets)	<div style="background-color: #ffc107; color: white; border-radius: 50%; width: 20px; height: 10px; margin: 0 auto; display: inline-block;"></div> Yellow 15-pallet trucks and smaller Truck Types: Small-medium trucks (15 pallets) or small trucks (10 pallets)	<div style="background-color: #e83c8d; color: white; border-radius: 50%; width: 20px; height: 10px; margin: 0 auto; display: inline-block;"></div> Purple 10-pallet trucks only Truck Types: Small trucks only (10 pallets)
---	---	--	--

ID	Category	Pallets	Compatible Trucks	Description	Address	Actions
1011	Green	10	Compatible with 18-pallet trucks and smaller	Silpo Supermarket	Kyiv, Khreshchatyk St, 44	EDIT PALLETS
1012	Green	2	Compatible with 18-pallet trucks and smaller	Novus Supermarket	Kyiv, Druzhby Narodiv Blvd, 16A	EDIT PALLETS
1013	Green	5	Compatible with 18-pallet trucks and smaller	Megamarket	Kyiv, Vadyma Hetmana St, 6	EDIT PALLETS

Рис.3.4 - Приклад заповнення палет на ТТ

Приклад заповнення палет на ТТ (Рис 3.5) :

- **Автопарк (що буде використаний):**
 - Авто 1: ємність 33 палет
 - Авто 2: ємність 18 палет
 - Авто 3: ємність 10 палет

ID	License Plate	Brand	Capacity	Driver	Trailer	Status	Actions
1	ABC-123	Volvo	33 pallets	John Doe	Trailer-1	In Use	DELETE
2	DEF-456	Scania	33 pallets	Jane Smith	Trailer-2	Available	DELETE
3	GHI-789	Mercedes	18 pallets	Michael Johnson	No Trailer	In Use	DELETE
4	JKL-012	MAN	18 pallets	Emily Brown	No Trailer	Available	DELETE
5	MNO-345	Iveco	15 pallets	David Wilson	No Trailer	Available	DELETE
6	PQR-678	DAF	15 pallets	Sarah Taylor	No Trailer	Available	DELETE
7	STU-901	Renault	10 pallets	Robert Anderson	No Trailer	In Use	DELETE
8	VWX-234	Fiat	10 pallets	Lisa Martinez	No Trailer	Available	DELETE

Рис.3.5 - Список використаних авто

Також при збільшенні автопарку є можливість додавати та видаляти авто через інтерфейс користувача.

Виклик генерації маршрутів

1. Користувач у веб-інтерфейсі переходить на сторінку “**Генерація маршруту**” (Рис. 3.7 , 3.9 та 3.10).
2. Вводить тестові дані або завантажує їх через форму.
3. Натискає кнопку “**Згенерувати маршрут**”, що викликає POST /api/routes з тілом запити (Рис. 3.6 та 3.8).

```

2025/05/07 11:59:45 Optimizing routes for date: 2025-05-07
2025/05/07 11:59:45 Processing 1 Blue category points
2025/05/07 11:59:46 Route 0: Calculated real distance: 55.7 km with 1 delivery points
2025/05/07 11:59:46 Successfully created 1 routes with 33-pallet trucks for Blue category
2025/05/07 11:59:46 Processing 3 Green category points
2025/05/07 11:59:48 Route 0: Calculated real distance: 49.7 km with 3 delivery points
2025/05/07 11:59:48 Successfully created 1 routes with 18-pallet trucks for Green category
2025/05/07 11:59:48 Processing 2 Purple category points
2025/05/07 11:59:49 Route 0: Calculated real distance: 33.8 km with 2 delivery points
2025/05/07 11:59:49 Successfully created 1 routes with 10-pallet trucks for Purple category
2025/05/07 11:59:49 Successfully optimized 3 routes, now saving
2025/05/07 11:59:50 Route 1: Calculated real distance: 55.7 km with 1 delivery points
2025/05/07 11:59:52 Route 2: Calculated real distance: 49.7 km with 3 delivery points
2025/05/07 11:59:54 Route 3: Calculated real distance: 33.8 km with 2 delivery points
2025/05/07 11:59:54 Saved 3 routes out of 3 generated routes

```

Рис.3.6 - Консольні INFO-log`s

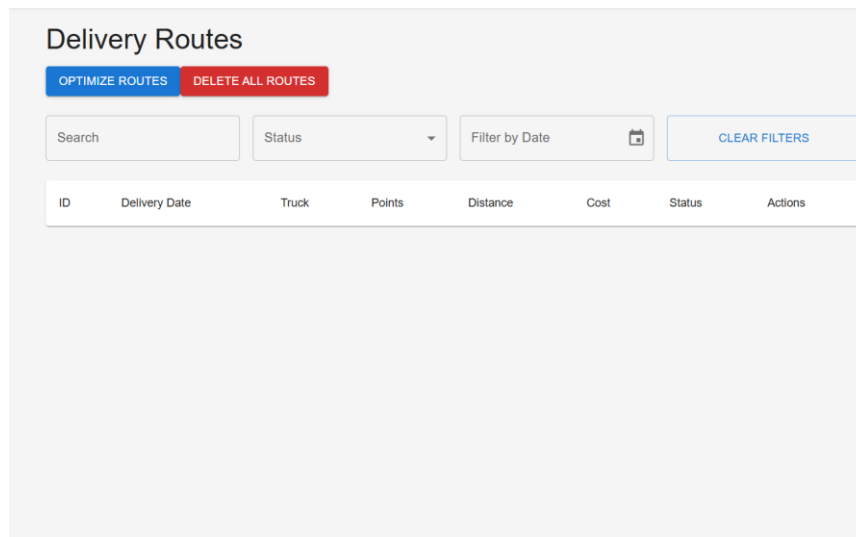


Рис.3.7 - Сторінка з оптимізацією маршрутів

Name	Status	Type	Initiator	Size	Time
routes	200	xhr	Routes.js:158	0.3 kB	7 ms
routes?t=1746608479284	200	xhr	Routes.js:71	0.3 kB	2 ms
trucks	200	xhr	Routes.js:79	1.5 kB	3 ms
delivery-points	200	xhr	Routes.js:83	35.2 kB	4 ms
optimize	200	xhr	Routes.js:130	1.2 kB	8.73 s
routes?t=1746608489845	200	xhr	Routes.js:71	1.1 kB	3 ms
trucks	200	xhr	Routes.js:79	1.5 kB	2 ms
delivery-points	200	xhr	Routes.js:83	35.2 kB	4 ms

Рис.3.8 - HTTP запит у вкладці Network браузеру

Delivery Routes

OPTIMIZE ROUTES DELETED ALL ROUTES

Search Status Filter by Date CLEAR FILTERS

ID	Delivery Date	Truck	Points	Distance	Cost	Status	Actions
1	2025-05-07	ABC-123 (33 pallets)	1	55.7 km	3453 units	Planned	DETAILS EDIT DELETE
2	2025-05-07	GHI-789 (18 pallets)	3	49.7 km	2882 units	Planned	DETAILS EDIT DELETE
3	2025-05-07	STU-901 (10 pallets)	2	33.8 km	1757 units	Planned	DETAILS EDIT DELETE

Рис.3.9 - Результат оптимізації маршрутів доставки

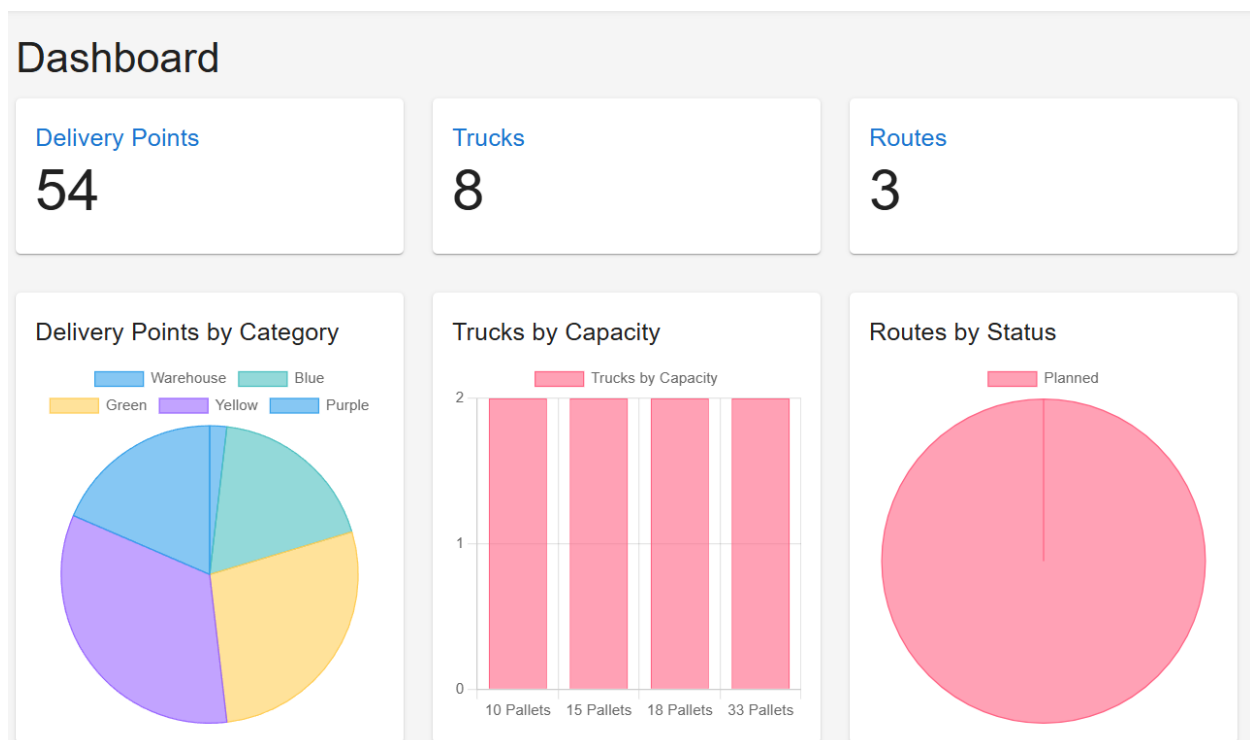


Рис.3.10 - Dashboard програми після створення маршрутів

Алгоритм nearest neighbour розподіляє точки між двома транспортними засобами з урахуванням ємності (Рис 3.11 , 3.12 та 3.13):

- **Авто 1 (33 палет):**

Route Details

<p>General Information</p> <p>Route ID: 1 Delivery Date: 2025-05-07 Status: Planned Total Distance: 55.7 km Total Cost: 3453 units</p>	<p>Truck Information</p> <p>License Plate: ABC-123 Brand: Volvo Capacity: 33 pallets Driver: John Doe</p>
--	---

Delivery Points

1. Metro Cash & Carry
ID: 1001 | Category: Blue
Pallets in this route: 33 | Total pallets assigned: 33 | Original point pallets: 33

[CLOSE](#)

Рис.3.11 - Звіт по першому авто

- **АВТО 2 (18 палет):**

Route Details

<p>General Information</p> <p>Route ID: 2 Delivery Date: 2025-05-07 Status: Planned Total Distance: 49.7 km Total Cost: 2882 units</p>	<p>Truck Information</p> <p>License Plate: GHI-789 Brand: Mercedes Capacity: 18 pallets Driver: Michael Johnson</p>
--	---

Delivery Points

1. Silpo Supermarket
ID: 1011 | Category: Green
Pallets in this route: 10 | Total pallets assigned: 10 | Original point pallets: 10

2. Megamarket
ID: 1013 | Category: Green
Pallets in this route: 5 | Total pallets assigned: 5 | Original point pallets: 5

3. Novus Supermarket
ID: 1012 | Category: Green
Pallets in this route: 2 | Total pallets assigned: 2 | Original point pallets: 2

[CLOSE](#)

Рис.3.12 - Звіт по другому авто

Можемо бачити порядок доставки по ТТ, кілометраж та вартість.

- **Авто 3 (10 палет):**

Route Details

<p>General Information</p> <p>Route ID: 3 Delivery Date: 2025-05-07 Status: Planned Total Distance: 33.8 km Total Cost: 1757 units</p>	<p>Truck Information</p> <p>License Plate: STU-901 Brand: Renault Capacity: 10 pallets Driver: Robert Anderson</p>
--	--

Delivery Points

1. Convenience Store 24/7
ID: 1045 | Category: Purple
Pallets in this route: 5 | Total pallets assigned: 5 | Original point pallets: 5

2. Minimarket Rukavychka
ID: 1044 | Category: Purple
Pallets in this route: 4 | Total pallets assigned: 4 | Original point pallets: 4

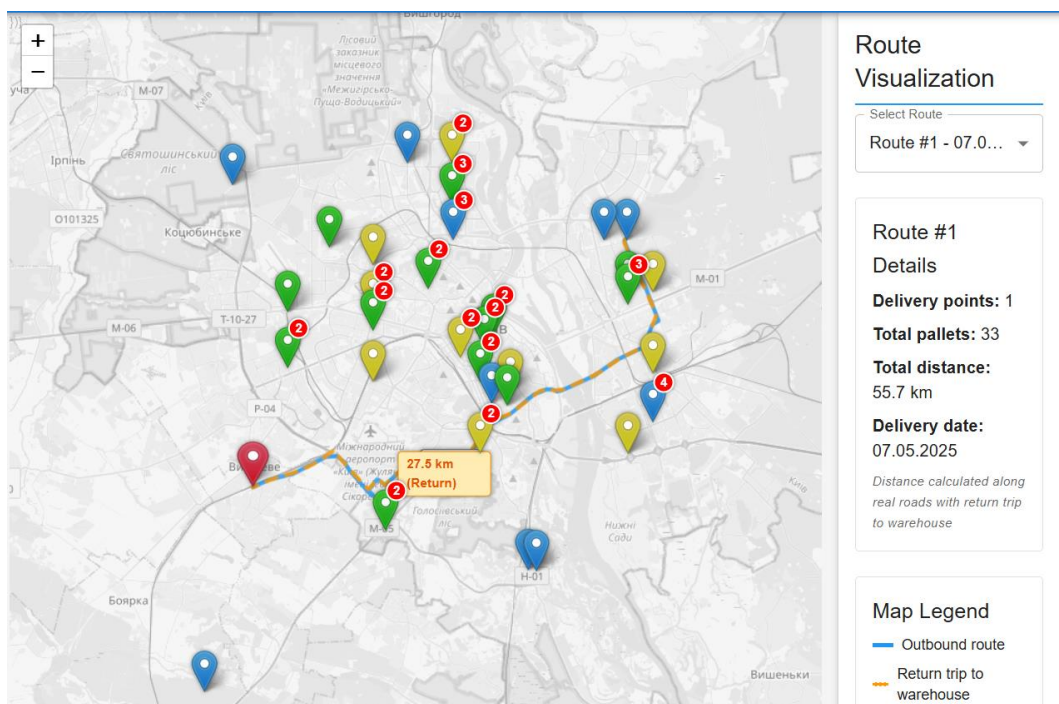
CLOSE

Рис.3.13 - Звіт по третьому авто

Візуалізація результатів

1. Карта маршрутів:

- Компонент React Leaflet відображає обраний нами маршрут на карті (Рис. 3.14 та 3.15).



The image shows a map with a delivery route highlighted in blue and orange. The route starts at a warehouse (marked with a red pin and 'Return') and visits several delivery points marked with numbered pins (1-4). The sidebar panel on the right provides the following details:

Route Visualization

Select Route: Route #1 - 07.0...

Route #1 Details

- Delivery points:** 1
- Total pallets:** 33
- Total distance:** 55.7 km
- Delivery date:** 07.05.2025

Distance calculated along real roads with return trip to warehouse

Map Legend

- Outbound route
- Return trip to warehouse

Рис.3.14 - Візуалізація першого маршруту

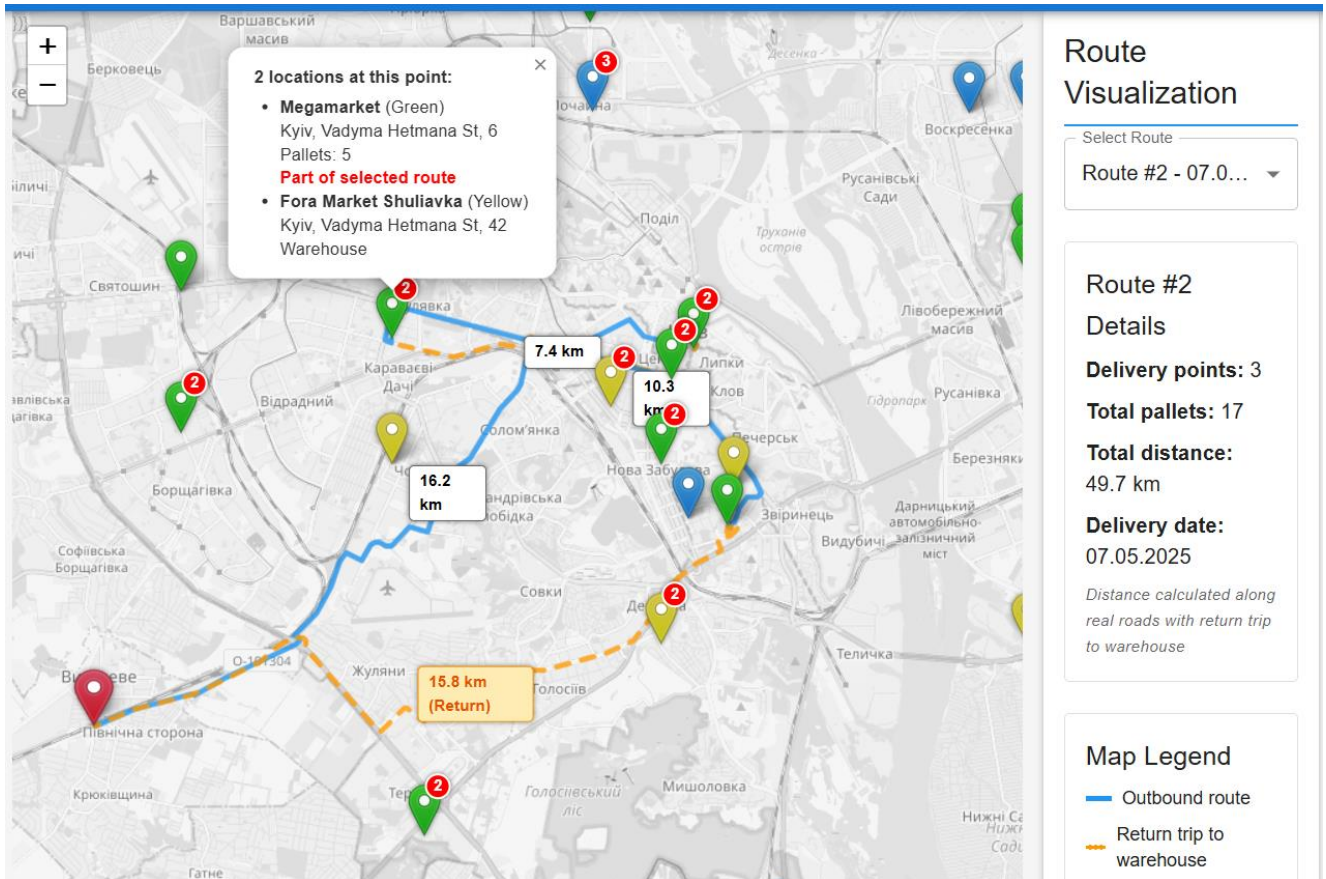


Рис.3.15 - Візуалізація другого маршруту

Через те що більшість ТТ знаходять близько чи в одному ТРЦ. На карті в червоних кружечках відображена їхня кількість, та інформація, що ми і бачимо на прикладі другого маршруту.

1. Діаграма пробігу:

Recharts генерує стовпчикову діаграму, по витратам та пробігу кожного авто (Рис. 3.16).

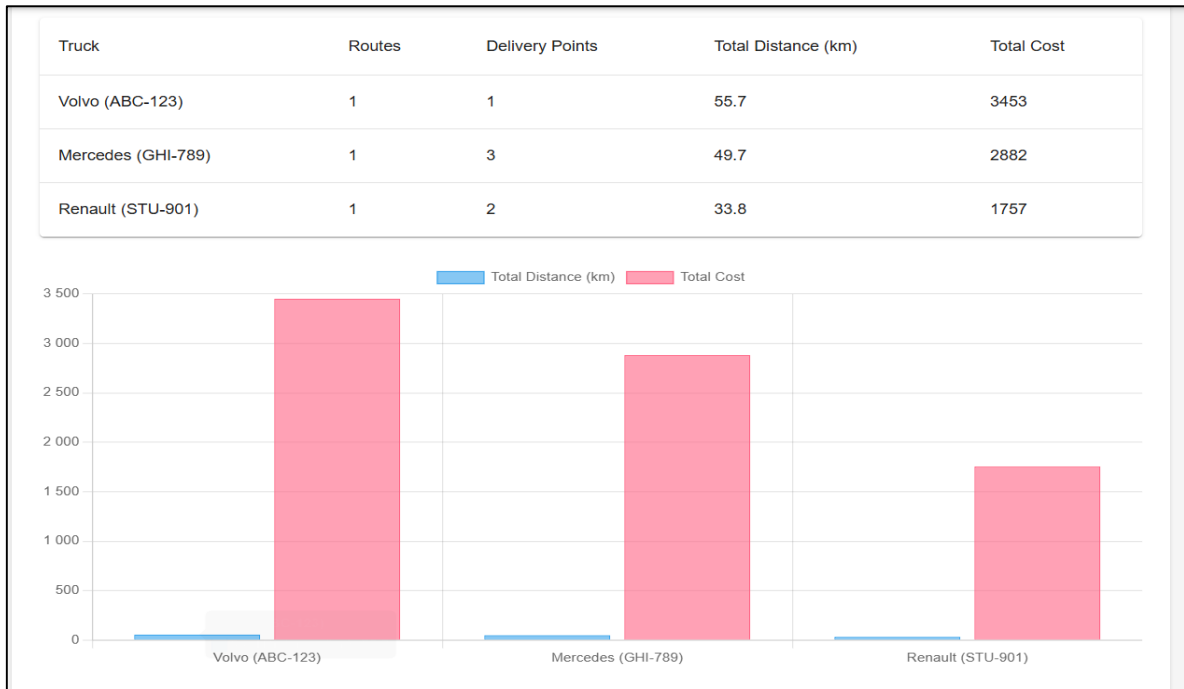


Рис.3.16 - Діаграми звітності на прикладі тестової вибірки

1. Звіт у таблиці:

- React Table показує загальну звітність за сьогодні (07.05.2025) дивись (Рис. 3.17 та 3.18).

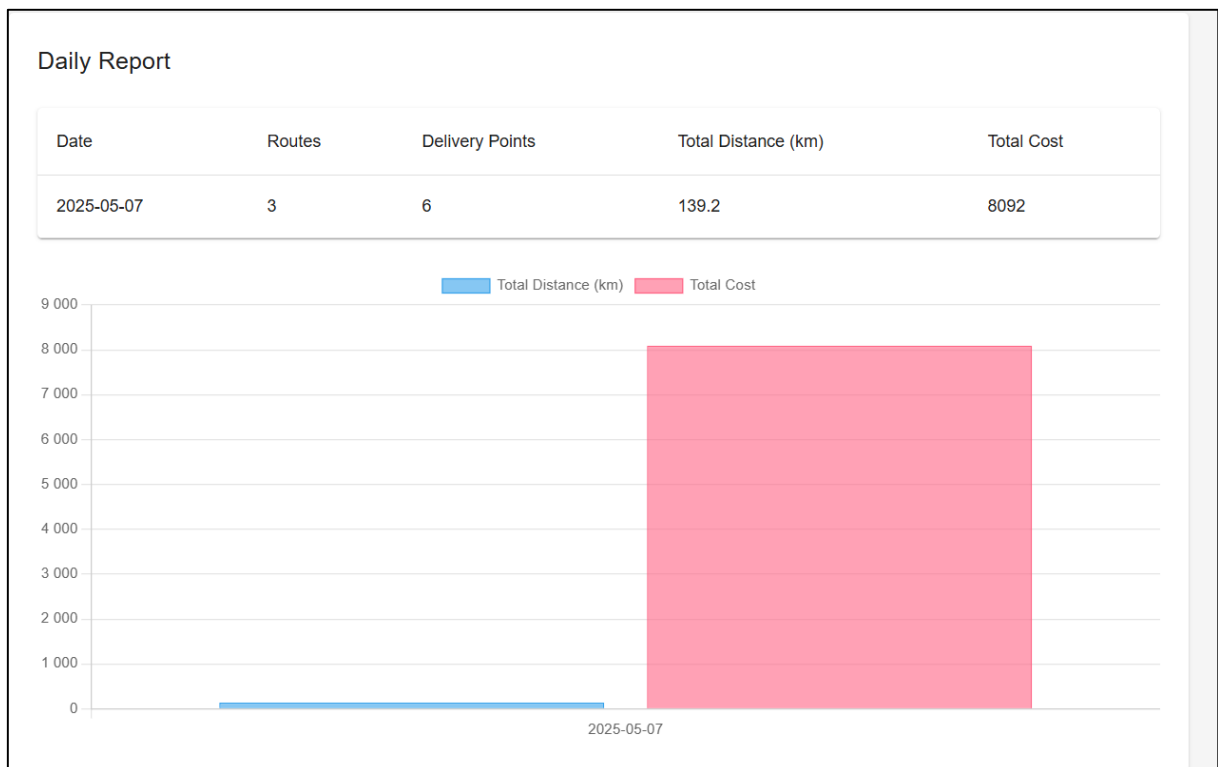


Рис.3.17 - Загальна звітність по маршрутам, ТТ, витратам та пробігу

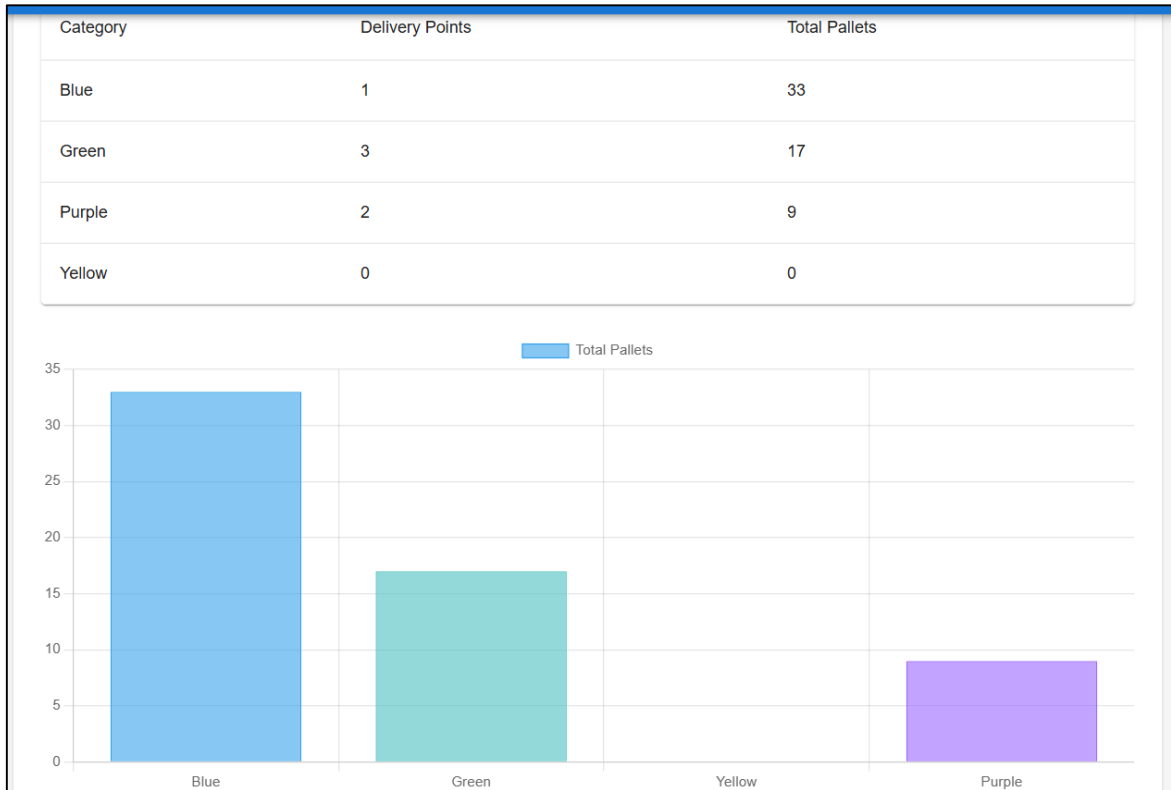


Рис.3.18 - Звітність по категоріям ТТ та кількості палет доставлених сьогодні

Перевірка коректності

1. Валідація результатів

- Перевіряємо, що суми палет у кожному маршруті не перевищують ємності авто.
- Підраховуємо вручну сумарний пробіг та порівнюємо з показником системи.

2. Тестування автоматично

- Юніт-тести для `internal/route/generator.go` із цими ж даними гарантують, що при зміні коду алгоритм повертає очікуваний результат.
- Інтеграційні тести перевіряють відповідь НТТР-ендпоінту.

3. Висновок

Контрольний приклад підтверджує коректність реалізації алгоритму

маршрутизації, стабільність обробки граничних випадків (розподіл залишкових палет) та наочність результатів у UI.

Цей приклад показує повний цикл, який починається з введення даних і закінчується візуалізацією та перевіркою результатів, що дозволяє забезпечити надійність і прозорість роботи системи.

РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Вступ

Ефективне функціонування логістичних процесів відіграє ключову роль у сучасній економіці, особливо в умовах високої конкуренції, глобалізації та цифровізації. З огляду на це, розробка й впровадження автоматизованої інформаційної системи для управління логістикою дозволяє не лише оптимізувати ресурси підприємства, а й забезпечити конкурентні переваги на ринку.

У цій дипломній роботі розроблено систему, яка може вирішувати проблеми планування маршрутів, аналізу ефективності доставки, моніторингу роботи складу та транспортування. Побудований на гнучкій архітектурі, придатній до масштабування, програмний продукт інтегрує сучасні методи візуалізації та оптимізації даних..

Метою даного розділу є економічне обґрунтування доцільності впровадження розробленої інформаційної системи, аналіз ринку, конкурентного середовища, фінансових показників, а також оцінка ризиків, пов'язаних із реалізацією проєкту. Обґрунтування включає технічні параметри системи, витрати на її розробку та прогнозований економічний ефект від впровадження.

4.2 Резюме

Даний дипломний проєкт присвячений розробці та техніко-економічному обґрунтуванню впровадження сучасної інформаційної системи для управління логістичними процесами підприємства. Підвищення ефективності логістичної діяльності за допомогою автоматизації, цифровізації процесів прийняття рішень і мінімізації людського фактору є основною передумовою для реалізації цього проєкту.

Бізнес-ідея полягає у створенні універсального програмного продукту, який поєднує модулі збору, обробки, візуалізації та оптимізації логістичної інформації. Система покликана забезпечити інтеграцію логістичних операцій — від планування маршрутів транспорту і роботи складів до обробки замовлень та контролю виконання доставки в режимі реального часу.

Місія проєкту — забезпечити підприємства будь-якого масштабу сучасним інструментом для цифрової трансформації логістичних процесів, що дозволяє підвищити їхню рентабельність, швидкість та точність функціонування.

Цілі проєкту:

- Розробити функціональну, адаптивну та масштабовану систему.
- Досягти економії логістичних витрат не менше ніж на 20% протягом першого року після впровадження.
- Скоротити час планування маршрутів у 3–5 разів завдяки автоматизації.
- Підвищити точність виконання доставок до 98–99%.
- Забезпечити інтеграцію з існуючими системами обліку, CRM та ERP.

Цільова аудиторія — підприємства малого, середнього та великого бізнесу, що займаються перевезенням вантажів, дистрибуцією, логістикою або мають внутрішні транспортні потреби (торговельні мережі, склади, поштові та кур'єрські служби, логістичні оператори).

Конкурентна перевага розробки полягає в поєднанні гнучкої архітектури, сучасних алгоритмів оптимізації, зручного користувацького інтерфейсу та можливості адаптації під потреби конкретного клієнта. Програма не лише додає цифровий вимір логістиці, але й дозволяє аналітично керувати ресурсами компанії в динамічних умовах.

Фінансова модель проєкту передбачає одноразові витрати на розробку системи, а також перспективу подальшої комерціалізації за моделлю SaaS (програмне забезпечення як послуга), що забезпечить стабільне джерело доходів при масштабуванні ринку.

Загальний висновок: проєкт розробки інформаційної системи логістики є доцільним, економічно обґрунтованим і має значний потенціал комерціалізації. У подальших підрозділах буде детально проаналізовано складові системи, ринок збуту, фінансові показники, ризики впровадження та економічний ефект.

4.3 Опис проєктованого продукту

Проєктований програмний продукт є багатофункціональною інформаційною системою для підтримки управлінських рішень у сфері логістики. Основна мета розробки — створення інструменту, який дозволяє автоматизувати логістичні процеси підприємства, зокрема: планування та оптимізацію маршрутів транспорту, візуалізація маршрутів на карті, контроль виконання доставок та аналіз ефективності.

Ключові функціональні можливості:

- **Планування маршрутів:** автоматичне формування найкоротших або найекономічніших маршрутів з урахуванням обмежень (тоннаж, кількість транспортних одиниць).
- **Оптимізація завантаження транспорту:** аналіз обсягів та характеристик вантажів з метою мінімізації кількості рейсів.
- **Моніторинг виконання доставки:** контроль статусу замовлень у реальному часі з візуалізацією на інтерактивній мапі.
- Для повноцінного функціонування передбачено інтеграцію з іншими складськими системами.
- **Модулі аналітики:** генерація звітів щодо витрат, кількості доставленого товару на магазини, пробіг кожного авто.

Технічні характеристики продукту :

- Клієнт-серверна архітектура.
- Веб-інтерфейс для користувачів і адміністративної панелі.
- Мікросервісна структура, що дозволяє масштабування та розширення функціональності.
- **Мова програмування:** Golang (backend), React (frontend).

- **База даних:** MongoDB.
- **Картографічні сервіси:** інтеграція з карти з пакету node.
- **Оптимізаційні алгоритми:** розроблено алгоритм автоматизації маршрутів, на основі підходящих габаритів авто для магазинів, відстані між точками та відсотку завантаження.

Цільові користувачі системи:

- 1) Логістичні менеджери.
- 2) Оператори складу.
- 3) Водії та диспетчери.
- 4) Аналітики з транспорту.
- 5) Керівництво компаній.

Розроблений продукт може бути адаптований як до внутрішніх логістичних потреб компанії (B2B-модель), так і до масового комерційного використання шляхом запуску SaaS-платформи. Система дозволяє скоротити витрати на логістику, підвищити швидкість і точність доставок, а також забезпечує прозорість усіх логістичних операцій.

4.4 Оцінка ринку збуту

Ринок логістичних послуг в Україні та Європі стрімко розвивається, що зумовлено поширенням електронної комерції, міжнародної торгівлі, зміною поведінки споживачів та діджиталізацією глобального бізнесу. У цьому контексті все більше організацій шукають методи вдосконалення своїх логістичних процесів, що призводить до постійної потреби в інформаційних системах управління логістикою..

Стан і тенденції ринку

За даними досліджень, обсяг українського ринку логістичних послуг у 2023 році становив близько 120 млрд грн, із прогнозованим щорічним зростанням на 8–12% [34]. Водночас у країнах ЄС частка витрат на логістику в структурі

операційних витрат підприємств сягає 10–15%, що стимулює попит на цифрові рішення для оптимізації логістики.

Тенденції, що визначають зростання попиту:

- збільшення частки онлайн-торгівлі та кур'єрської доставки;
- глобалізація ланцюгів постачання;
- потреба у прозорості логістичних процесів;
- зростання цін на паливо та інші ресурси;
- посилення конкуренції серед логістичних операторів.

Цільові сегменти споживачів

Проектований продукт орієнтований на наступні категорії споживачів (табл. 4.1):

Таблиця 4.1

Оцінка потенційного попиту

Сегмент	Характеристика	Приклади
Малі та середні підприємства (SMEs)	Потребують ефективних, доступних рішень	Дистриб'ютори, торговельні мережі
Великі логістичні оператори	Потребують гнучкої системи з аналітикою	Транспортно-експедиційні компанії
Кур'єрські та поштові служби	Високочастотна доставка, потреба в автоматизації	Нова Пошта, Meest, GLS
Виробничі підприємства	Внутрішня логістика, складська оптимізація	АПК, FMCG, фармацевтика
e-Commerce бізнеси	Масові щоденні доставки, інтеграція з CMS	Інтернет-магазини, маркетплейси

В умовах обмеженого проникнення цифрових рішень у логістику в Україні та швидкого розвитку цього сегмента в Європі, потенціал для масштабування значний. За приблизними підрахунками, лише в Україні існує понад 30 000 підприємств, які мають прямий або опосередкований запит на логістичну

автоматизацію. При цьому частка компаній, що вже використовують подібні рішення, не перевищує 20% [32].

Прогнозована частка ринку

У перші 2–3 роки після впровадження продукту можливим є охоплення до 1% внутрішнього ринку України, що становить близько 300 клієнтів, з подальшим виходом на ринки Польщі, Словаччини, Угорщини та Румунії. При середній вартості річної ліцензії 500–1000 USD — це формує потенційний прибуток у межах 150 000–300 000 USD на рік при низьких витратах підтримки.

Висновок

Ринок збуту інформаційних систем управління логістикою характеризується стабільним попитом, різноманітною цільовою аудиторією в багатьох секторах, значним ступенем цифрових змін у бізнесі та мінімальною насиченістю спеціалізованим вітчизняним програмним забезпеченням. Це створює сприятливе середовище для виходу проєкту на ринок та подальшого масштабування.

4.5 Аналіз конкуренції

Сфера логістичних інформаційних систем є динамічно зростаючим сегментом ІТ-ринку, який охоплює широкий спектр продуктів: від простих TMS-рішень (Transportation Management System) до складних ERP-комплексів з модулями для управління постачанням, складом, дистрибуцією та аналітикою. На ринку функціонує велика кількість як глобальних гравців, так і національних розробників (табл. 4.2), однак структурний аналіз (табл. 4.3) свідчить про нерівномірне охоплення потреб малого, середнього та великого бізнесу.

Транснаціональні корпорації зазвичай концентруються на великих організаціях з потужними мережами постачання, але малим і середнім підприємствам може не вистачати фінансових ресурсів на дорогі або надто складні рішення. Це створює оптимальні умови для просування універсальних, економічно ефективних і легко впроваджуваних локалізованих продуктів.

Таблиця 4.2

Основні конкуренти

Назва	Країна	Цільовий сегмент	Основні функції	Особливості
SAP SCM	Німеччина	Великий бізнес	Управління постачанням, планування попиту, оптимізація виробництва	Висока вартість, складне впровадження
Oracle SCM Cloud	США	Великий і середній бізнес	Хмарна платформа з модулями закупівель, логістики, управління запасами	Необхідність налаштування, висока ціна
LogiNext Mile	Індія	Е-commerce, доставка	Візуалізація маршрутів, трекінг у реальному часі, оптимізація логістики «останньої милі»	SaaS-модель, адаптований для англомовного ринку
ABM Rinkai TMS	Україна	Середній бізнес	Управління автопарком, формування маршрутів, документація	Немає підтримки мультимодальних перевезень
Delovod Logistics	Україна	Мікробізнес	Прості складські операції та логістичні звіти	Обмежена функціональність, відсутність API
Onfleet	США	Кур'єрська доставка	Планування маршрутів, GPS-моніторинг, звітність	Висока вартість для малого бізнесу, немає локалізації

Таблиця 4.3

SWOT-аналіз конкурентного середовища

Сильні сторони конкурентів	Слабкі сторони конкурентів
Широкий функціонал ERP-рішень (SAP, Oracle)	Надлишкова складність для МСБ
Хмарні сервіси з автоматизацією (LogiNext, Onfleet)	Відсутність підтримки української мови та локальних норм

Продовження Таблиці 4.3

Готові API для інтеграції	Висока вартість обслуговування
Стабільна підтримка та оновлення	Неадаптованість до регіональних особливостей логістики

Конкурентні переваги розроблюваної ІС

Розроблювана інформаційна система має низку переваг, які дозволяють їй зайняти конкурентну позицію на ринку:

1. **Локалізація** — повна адаптація до українських та європейських реалій: мови, регламентів, форматів документів.
2. **Гнучка модульна структура** — можливість кастомізації під потреби конкретного клієнта, інтеграції з 1С, CRM, ERP тощо.
3. **Використання сучасних алгоритмів** — оптимізація маршрутів з урахуванням реального трафіку, часових вікон, обмежень складів.
4. **Масштабованість** — можливість роботи з малим автопарком і розширення до складних логістичних ланцюгів.
5. **Проста візуалізація та аналітика** — користувачі без технічної підготовки можуть формувати звіти, діаграми, KPI.
6. **Конкурентна ціна** — доступне ліцензування на основі SaaS або одноразового впровадження з підтримкою.

Позиціонування на ринку

Розроблена система займає проміжну нішу між простими логістичними рішеннями для мікробізнесу та дорогими ERP-системами (табл. 4.4), пропонуючи оптимальний баланс між функціональністю, ціною та адаптованістю.

Таблиця 4.4

Позиціонування на ринку

Критерій	Системи ERP (SAP, Oracle)	Розроблювана система	Локальні продукти (Delovod)
Функціональність	Висока	Середня	Низька

Продовження Таблиці 4.4

Адаптація до українського ринку	Низька	Висока	Середня
Ціна	Висока	Середня	Низька
Гнучкість інтеграції	Середня	Висока	Низька
Придатність для МСБ	Низька	Висока	Середня

Висновок

Незважаючи на наявність потужних конкурентів, очікуваний ІС може ефективно конкурувати в секторі МСП завдяки локалізації, розумному балансу функціональності та вартості, а також акценту на вимогах кінцевого користувача. Це створює широкі можливості не лише на внутрішньому ринку, але й для подальшої експансії в Центральну та Східну Європу..

4.6 Стратегія маркетингу

Успішне впровадження та просування інформаційної системи управління логістичними процесами потребує чітко сформульованої маркетингової стратегії, яка охоплює всі етапи виходу продукту на ринок — від формування цільової аудиторії до вибору каналів дистрибуції та просування.

Цільова аудиторія

Основними споживачами програмного продукту є:

- **Малі та середні логістичні компанії**, які здійснюють автомобільні перевезення;
- **Інтернет-магазини та маркетплейси** із власними службами доставки;
- **Складські оператори та дистриб'ютори**, які потребують інтеграції між складами та транспортом;
- **Фармацевтичні та харчові компанії**, що працюють з критичними часовими вікнами доставки;

- **Виробничі підприємства**, які керують логістичними потоками між цехами, складами та споживачами.

Маркетингові цілі

1. **Залучити перших 10–20 корпоративних клієнтів** протягом першого року після запуску;
2. **Сформувати бренд та експертну репутацію** в логістичній галузі;
3. **Забезпечити прибутковість проекту** на 2-му році впровадження;
4. **Забезпечити стійке зростання кількості користувачів** на 20–30% щорічно.

Таблиця 4.5

Маркетинговий мікс (4P)

Компонент	Стратегія
Product (Продукт)	Хмарна або локальна ІС для управління логістикою, з модулями для маршрутизації, візуалізації, обліку та аналітики. Гнучка конфігурація, локалізований інтерфейс, підтримка API.
Price (Ціна)	Доступна підписка на основі SaaS (щомісячна/річна), а також модель одноразового впровадження. Вартість — нижча за глобальні ERP-рішення на 50–70%.
Place (Місце збуту)	Онлайн-продажі через офіційний сайт, каталоги B2B-програмного забезпечення (Capterra, SoftwareSuggest), партнерські інтегратори.
Promotion (Просування)	Контент-маркетинг (статті, кейси), реклама в професійних логістичних ЗМІ, участь у галузевих виставках, SEO, email-маркетинг, прямі продажі.

Канали просування (див. табл. 3.4) :

1. **Цифровий маркетинг:**
 - SEO-оптимізований вебсайт продукту;
 - Просування через Google Ads;

- Публікації на LinkedIn, Facebook, галузевих форумах.

2. Партнерські програми:

- Інтеграція з логістичними CRM і ІС-системами;
- Угода з компаніями-інтеграторами, які надають логістичний консалтинг.

3. Участь у галузевих заходах:

- Презентація продукту на виставках (Logistics Summit, Lviv Logistic Forum тощо);
- Співпраця з Асоціацією логістики України.

4. Пробні версії та демонстрації:

- 14-денна безкоштовна демо-версія;
- Проведення вебінарів з демонстрацією можливостей продукту.

Унікальна торгова пропозиція (УТП)

"Інформаційна система логістики — розумне рішення для малого та середнього бізнесу: швидке впровадження, адаптація до українського ринку та підтримка прийняття ефективних рішень."

Моніторинг результатів

Ефективність маркетингової стратегії оцінюватиметься на основі:

- динаміки кількості нових клієнтів (щомісяця);
- коефіцієнта конверсії (відвідувачі сайту → ліди → клієнти);
- середньої тривалості циклу продажу;
- окупності рекламних кампаній (ROI);
- зростання впізнаваності бренду серед логістичних операторів.

4.7 План впровадження системи

План впровадження інформаційної системи управління логістичними процесами є критично важливим етапом реалізації проекту. Його головна мета — забезпечити ефективну інтеграцію програмного продукту в існуючу інфраструктуру підприємства з мінімальними ризиками, фінансовими та часовими

витратами. Для цього використовується покроковий підхід (табл. 4.6) з урахуванням технічних, кадрових, фінансових та правових аспектів.

Основні цілі впровадження

- Забезпечити прозоре планування логістичних маршрутів і складських операцій;
- Мінімізувати логістичні витрати підприємства (паливо, простой, надлишкові запаси);
- Автоматизувати повторювані операції;
- Підвищити рівень обслуговування клієнтів (точність і своєчасність доставки);
- Надати керівництву аналітичні інструменти для прийняття рішень.

Таблиця 4.6

Етапи впровадження системи

№	Етап	Зміст робіт
1	Підготовчий аудит	Оцінка поточного рівня цифровізації логістичних процесів; вивчення специфіки підприємства; визначення критичних точок.
2	Технічне планування	Формування вимог до ІС, узгодження програмно-апаратної архітектури; підбір хмарного чи локального середовища.
3	Інсталяція системи	Розгортання програмного продукту, налаштування серверів, встановлення модулів, організація захищеного доступу.
4	Міграція та підготовка даних	Перенесення існуючих логістичних, клієнтських та складських даних зі старих систем або Excel; очищення та стандартизація даних.
5	Налаштування під бізнес-процеси	Адаптація структури маршрутів, складів, ролей, звітів і прав доступу під внутрішні регламенти підприємства.

Продовження Таблиці 4.6

6	Навчання персоналу	Проведення практичних навчальних модулів для логістів, диспетчерів, ІТ-фахівців, керівників відділів; підготовка документації та відеоінструкцій.
7	Пілотне тестування	Запуск системи в обмеженому масштабі (на одному складі, маршруті чи регіоні); збір зворотного зв'язку, виявлення проблем.
8	Повне розгортання	Запуск системи в експлуатацію на всіх ділянках логістичного ланцюга підприємства.
9	Технічна підтримка та супровід	Гарантійний супровід, оновлення системи, виправлення багів, консультації користувачів.

Часові рамки реалізації

Орієнтовна тривалість проекту — від 30 до 90 календарних днів, залежно від масштабу підприємства та складності логістичних процесів:

Таблиця 4.7

Часові рамки реалізації

Категорія підприємства	Кількість складів	Обсяг логістики	Тривалість впровадження
Малий бізнес	1–2	до 50 замовлень/день	30 днів
Середній бізнес	3–7	50–300 замовлень/день	45–60 днів
Великий бізнес	8+	понад 300 замовлень/день	60–90 днів

Команда впровадження

Для реалізації проекту необхідно залучити:

- **Проектного менеджера** — відповідає за загальну координацію процесу;

- **Аналітика бізнес-процесів** — займається описом логістичних сценаріїв;
- **Системного адміністратора** — забезпечує технічне середовище;
- **Розробників** — адаптують функціонал системи;
- **Тренера з навчання персоналу** — проводить практичне впровадження.

Модель впровадження

Застосовується **гібридна методологія**: на початковому етапі — каскадна модель (Waterfall) для точного планування і послідовного впровадження, а з другого етапу — **Agile/Scrum** для гнучкої адаптації під запити клієнта та швидкого зворотного зв'язку. Це забезпечує баланс між стабільністю і гнучкістю.

Контроль якості

Контроль впровадження відбувається на кожному етапі. Використовуються такі підходи:

- Регулярні статус-зустрічі з клієнтом;
- Check-листи для перевірки готовності кожного модуля;
- Інструменти тестування функціоналу (юніт- і інтеграційні тести);
- Оцінка продуктивності системи на основі контрольного набору даних.

4.8 Організаційний план

Організаційний план окреслює структуру управління проектом з впровадження логістичної інформаційної системи, розподіл обов'язків між учасниками, графік і методи взаємодії між усіма зацікавленими сторонами. Його відмінне виконання пом'якшує безлад в управлінні ресурсами, зменшує ризики затримок, перевитрат і непорозумінь між членами команди..

Мета організаційного плану

- Забезпечити ефективне управління проектом;
- Організувати чіткий поділ обов'язків та відповідальностей;
- Визначити ресурси, необхідні для реалізації;
- Узгодити часові рамки і взаємодію команд;
- Забезпечити контроль за виконанням завдань.

Організаційна структура

Для реалізації проекту (табл. 4.10) доцільно створити **тимчасову проектну команду** (табл. 4.8), до якої входять фахівці замовника (логістика, ІТ, управління) та сторонні виконавці (розробники, консультанти, тестувальники).

Таблиця 4.8

Структура команди

Посада	Функції
Керівник проекту	Координація роботи команди, управління строками та бюджетом, комунікація із замовником.
Бізнес-аналітик	Вивчення логістичних процесів, постановка задач для розробників, формування технічного завдання.
Розробник (backend)	Реалізація серверної логіки, обробка маршрутів, оптимізаційні алгоритми.
Розробник (frontend)	Побудова інтерфейсу користувача, інтеграція з API.
Тестувальник (QA)	Перевірка працездатності функціоналу, виявлення помилок.
Системний адміністратор	Розгортання серверів, налаштування безпеки та хостингу.
Тренер з навчання персоналу	Підготовка інструкцій, навчання співробітників замовника.

Таблиця 4.10

Графік реалізації проекту (Gantt-діаграма)

Етап	Тривалість	Відповідальні
Підготовка ТЗ	1 тиждень	Бізнес-аналітик
Розробка MVP (мінімального функціоналу)	2–3 тижні	Розробники
Інтеграція з базами даних	1 тиждень	Backend-розробник
Побудова інтерфейсу	2 тижні	Frontend-розробник
Тестування MVP	1 тиждень	QA

Продовження Таблиці 4.10

Демонстрація MVP та зворотній зв'язок	3 дні	Вся команда
Доробка та повна розробка	2 тижні	Backend/Frontend
Навчання персоналу	3–5 днів	Тренер
Впровадження системи	1 тиждень	Вся команда
Підтримка	1 місяць	Системний адміністратор, розробники

Взаємодія між учасниками

Проектна команда взаємодіє за допомогою:

- Щотижневих Zoom-нарад для звіту про прогрес;
- Трекінгу задач у Trello або Jira;
- Спільного Google Drive для зберігання документації;
- Використання системи контролю версій Git/GitHub для коду.

Необхідні ресурси

- **Технічні:** сервер (VPS або хмарний), бази даних, домен, засоби моніторингу;
- **Людські:** 5–7 фахівців на етап розробки та 2–3 — для супроводу;
- **Фінансові:** витрати на хостинг, зарплати, ПЗ (якщо не використовуються open-source рішення).

Модель управління

Застосовується **матрична організаційна модель**, коли розробники підпорядковуються як функціональним керівникам (керівник IT-відділу), так і менеджеру проекту. Це дозволяє ефективно розподіляти ресурси без дублювання ролей.

4.9 Юридичний план

Юридичний план є важливою складовою техніко-економічного обґрунтування, оскільки визначає правові аспекти створення, використання та захисту інформаційної системи. Він охоплює питання прав власності, захисту

персональних даних, ліцензування, регуляторного дотримання та договірних відносин.

Правова форма діяльності

Для комерційного впровадження системи доцільно зареєструвати юридичну особу або фізичну особу–підприємця (ФОП) в Україні чи іншій юрисдикції, де здійснюється основна діяльність. Найбільш зручними є:

- **ФОП (3 група, спрощена система оподаткування)** — для стартапу з невеликим штатом;
- **ТОВ (товариство з обмеженою відповідальністю)** — для партнерських моделей, інвестицій та масштабування.

Інтелектуальна власність

Результати розробки інформаційної системи належать автору (або компанії-розробнику) і охороняються відповідно до:

- Закону України «Про авторське право і суміжні права»;
- Цивільного кодексу України (розділ IV);
- Угоди TRIPS та Бернської конвенції (у разі міжнародного використання).

Можлива **реєстрація авторських прав** на програмне забезпечення в УкрІНТЕІ, що спростить доказ авторства у разі порушення.

Ліцензування ПЗ

Розробник має визначити модель розповсюдження системи:

- **Пропріетарна ліцензія** — система надається на умовах обмеженого користування, за договором;
- **Open-source** — код розповсюджується вільно (на умовах GNU GPL, MIT, Apache тощо), але потребує вибору відповідної ліцензії.

У випадку використання сторонніх бібліотек чи фреймворків, необхідно перевірити їхню **сумісність з вибраною ліцензією**.

Захист персональних даних

Якщо система обробляє персональні дані (імена клієнтів, адреси доставки, контактну інформацію), вона має відповідати:

- Закону України «Про захист персональних даних»;
- GDPR (у разі роботи з користувачами ЄС).

Потрібно реалізувати технічні та організаційні заходи безпеки:

- Шифрування даних;
- Авторизацію та розмежування доступу;
- Ведення журналів доступу до персональних даних;
- Політику конфіденційності.

Договірні зобов'язання

Необхідно укладати такі договори:

- **З замовниками:** договір на розробку/впровадження ПЗ (з актами виконаних робіт, технічним завданням, строками);
- **З підрядниками:** договори підряду чи надання послуг (з гарантіями якості, відповідальністю, обмеженням доступу до даних);
- **З працівниками:** договори з положеннями про нерозголошення (NDA) та передачу прав на результати роботи.

Таблиця 4.11

Ризики юридичного характеру

Ризик	Механізм мінімізації
Використання неліцензованих компонентів	Аудит ПЗ, дотримання ліцензій
Порушення авторських прав	Реєстрація прав, договірні гарантії
Невиконання вимог захисту даних	Політики безпеки, технічний аудит
Недобросовісні партнери	Чіткі договори, юридичний супровід

Юридичний супровід на кожному етапі проекту дозволяє захистити права розробника (табл 4.11), уникнути штрафів та судових позовів, а також підвищити довіру до інформаційної системи з боку партнерів і клієнтів.

4.10 Фінансовий план

Фінансовий план дозволяє оцінити економічну життєздатність створення та впровадження логістичної інформаційної системи, спрогнозувати витрати, пов'язані з розробкою, експлуатацією, просуванням та обслуговуванням системи, а також визначити фінансові переваги від її впровадження (табл. 4.12). План передбачає, що ініціатива буде поступово розширюватися і проникати на ринок протягом першого року.

Таблиця 4.12

Основні статті витрат

Стаття витрат	Вартість, грн
Розробка програмного забезпечення	180 000
Заробітна плата розробника (6 міс.)	90 000
Дизайн інтерфейсу користувача	20 000
Сервери та хостинг (12 міс.)	12 000
Закупівля домену та SSL-сертифікату	3 000
Ліцензії на ПЗ, бібліотеки	10 000
Юридичні та бухгалтерські послуги	5 000
Маркетинг і реклама	25 000
Непередбачені витрати (10%)	34 500
Загальна сума витрат	379 500

Прогноз доходів (у разі комерціалізації)

Модель монетизації передбачає продаж системи підприємствам логістичного або торговельного сектору за ліцензією або підпискою (табл.4.13). Наведено два сценарії:

1. **Продаж SaaS-доступу за підпискою:** 1 500 грн/міс. за 1 компанію.
2. **Продаж ліцензії з індивідуальним налаштуванням:** 25 000 грн/клієнт.

Таблиця 4.13

Прогноз доходів (у разі комерціалізації)

Показник	Оптимістичний сценарій	Консервативний сценарій
Кількість підписників (12 міс.)	30	10
Кількість ліцензій	10	3
Доходи від підписки	540 000	180 000
Доходи від ліцензій	250 000	75 000
Загальні доходи за рік	790 000	255 000

Точка безбитковості

Для покриття початкових інвестицій у розмірі 379 500 грн (табл. 4.14)

достатньо:

- 253 підписних місяців (≈ 21 підписник на рік);
- або продажу 16 ліцензій по 25 000 грн;
- або комбінованого варіанту: 10 ліцензій + 5 постійних підписників.

Таблиця 4.14

Оцінка прибутковості

Показник	Значення (оптимістично)
Інвестиції	379 500 грн
Річний дохід	790 000 грн
Чистий прибуток	410 500 грн
Рентабельність	$\approx 108\%$
Період окупності	6–8 місяців

Таким чином, фінансовий план демонструє високу економічну доцільність бта правильної маркетингової стратегії.

4.11 Стратегія фінансування

Стратегія фінансування визначає джерела фінансування, необхідні для впровадження логістичної інформаційної системи, починаючи з етапу розробки і закінчуючи запуском продукту. Ефективне фінансування сприяє безперервності проекту, зменшує ризики та максимізує витрати на кожному етапі.

Етапи фінансування проекту

Реалізацію проекту можна умовно поділити на три ключові етапи з відповідними потребами в фінансуванні:

1. Попередній етап (аналіз, ТЗ, прототип):

- Витрати: до 50 000 грн.
- Джерела: власні кошти, навчальні гранти, хакатони.

2. Етап MVP (розробка мінімального життєздатного продукту):

- Витрати: 150 000–200 000 грн.
- Джерела: участь у стартап-інкубаторах, грантові програми, мікроінвестори.

3. Етап масштабування та комерціалізації:

- Витрати: 200 000–300 000 грн.
- Джерела: краудфандинг, венчурний капітал, бізнес-ангели, кредитування (табл. 4.15).

Таблиця 4.15

Можливі джерела фінансування

Джерело фінансування	Переваги	Недоліки
Власні кошти	Повна автономія, швидкість прийняття рішень	Обмеженість ресурсу
Гранти та освітні програми	Безповоротність, підтримка інновацій	Висока конкуренція, обмежені строки
Краудфандингові платформи	Валідація ідеї на ринку, фінансування від спільноти	Необхідна активна маркетингова кампанія

Продовження Таблиці 4.15

Бізнес-ангели та інвестори	Досвід та зв'язки, доступ до значного капіталу	Передача частки бізнесу, зовнішній вплив
Банківські кредити	Оперативне фінансування	Необхідність гарантій, фінансове навантаження
Державні програми підтримки ІТ	Пільгове фінансування, доступ до ресурсів	Бюрократичність, залежність від політики

Рекомендована стратегія

Для даного проєкту доцільно реалізувати **гібридну модель фінансування**, яка включає:

- **На першому етапі** — використання власних коштів, участь у грантових програмах (наприклад, HORIZON Europe, Український фонд стартапів);
- **На етапі MVP** — подача заявки на державну підтримку або залучення бізнес-ангела;
- **На етапі впровадження** — краудфандинг на платформах (наприклад, Kickstarter, INDIEGOGO), переговори з логістичними компаніями щодо тестування системи з подальшим інвестуванням.

Фінансовий супровід

Для ефективного управління фінансами проєкту рекомендується:

- Вести **бюджет проєкту** з прогнозом на 12–24 місяці;
- Контролювати **грошові потоки** через CRM та бухгалтерські сервіси;
- Вести регулярну **звітність для інвесторів** або партнерів;
- Використовувати інструменти аналітики (наприклад, Power BI, Tableau) для оцінки ефективності використання коштів.

Отже, чітка й гнучка стратегія фінансування є ключем до успішного запуску системи, її адаптації до ринкових умов і забезпечення стійкого зростання.

4.12 Оцінка ризику

Оцінка ризиків є невід'ємною частиною управління проєктом, оскільки дозволяє заздалегідь виявити потенційні загрози для реалізації інформаційної

системи логістики, оцінити ймовірність їх виникнення та передбачити можливі наслідки. Грамотне управління ризиками дозволяє знизити втрати, скоротити час впровадження та забезпечити стабільне функціонування системи на етапі експлуатації (табл. 4.16).

Таблиця 4.16

Основні типи ризиків

Категорія ризику	Потенційний ризик	Ймовірність	Вплив	Коментар / Заходи реагування
Технічні	Помилки в алгоритмах, збої в ПЗ	Середня	Високий	Проведення тестування, ревізія коду, резервне копіювання
Фінансові	Нестача коштів для завершення розробки	Висока	Високий	Диверсифікація джерел фінансування
Юридичні	Проблеми з ліцензіями, порушення ПП	Низька	Середній	Консультація з юристом, дотримання стандартів
Маркетингові	Низький попит, слабе охоплення цільової аудиторії	Середня	Високий	Проведення маркетингового дослідження, SEO, реклама
Конкурентні	Поява потужного конкурента або копії системи	Середня	Середній	Постійне оновлення продукту, робота з клієнтами
Організаційні	Нестача кваліфікованих кадрів, затримки в роботі	Низька	Високий	Автоматизація процесів, делегування
Зовнішні (форс-мажорні)	Економічна нестабільність, воєнні дії	Низька	Високий	Гнучка модель роботи, хмарна інфраструктура

Методи мінімізації ризиків

1. **Створення резервного фонду** — 10–15% загального бюджету передбачено для покриття непередбачених витрат.
2. **Модульна архітектура ПЗ** — дозволяє локалізувати помилки без зупинки всієї системи.
3. **Тестування на ранніх етапах** — багаторівнева перевірка всіх модулів: unit-тестування, інтеграційне тестування, end-to-end.
4. **Періодичний аудит** — регулярний контроль фінансів, технічного прогресу та відповідності нормативним вимогам.
5. **Зворотний зв'язок з користувачами** — ранній запуск бета-версій дозволяє виявити слабкі сторони продукту ще до комерційного релізу.

Таблиця 4.17

SWOT-аналіз (у контексті ризиків)

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
Інноваційна ідея	Залежність від фінансування
Гнучкість у реалізації	Відсутність брендового імені
Можливості (Opportunities)	Загрози (Threats)
Розширення на суміжні ринки	Активізація конкуренції
Залучення інвесторів	Економічні коливання/нестабільність

Таким чином, ризики проекту (табл. 4.17) не є серйозними і можуть бути ефективно керованими завдяки надійному управлінському підходу, технічній адаптивності та фінансовим обмеженням. Всі потенційні небезпеки мають відповідні механізми реагування, створені для пом'якшення їхнього впливу на загальний успіх проекту..

4.13 Страхування

У процесі розробки, впровадження та експлуатації логістичної інформаційної системи тягне за собою багато ризиків, пов'язаних з технологічними, фінансовими та регуляторними аспектами. Страхування є одним з інструментів зниження цих ризиків. Страховий захист дозволяє компенсувати потенційні збитки, що виникають внаслідок непередбачуваних подій, і забезпечує стабільність проекту..

Види страхування, актуальні для ІТ-проекту

1. Страхування відповідальності перед третіми особами

Забезпечує захист у випадках порушення авторських прав, неправомірного зберігання або обробки персональних даних, втручання у права користувачів. Це особливо актуально для веборієнтованих систем, що працюють із клієнтськими базами.

2. Страхування майна та обладнання

Включає захист серверного обладнання, комп'ютерної техніки, мережевої інфраструктури тощо від пожеж, затоплень, крадіжок, фізичних ушкоджень або форс-мажорних подій.

3. Кіберстрахування (cyber insurance)

Захищає від фінансових наслідків кібератак, зламів, витоку даних, шкідливого ПЗ та саботажу. Враховуючи зростаючу кіберзагрозу, цей тип страхування стає дедалі популярнішим.

4. Страхування фінансових ризиків

Передбачає покриття витрат у разі фінансових втрат, пов'язаних з незавершеним впровадженням, затримками у фінансуванні або зниженням очікуваного прибутку.

5. Страхування персоналу

Стосується спеціалістів, які беруть участь у реалізації проекту. Можливе як медичне страхування, так і страхування від нещасних випадків під час відряджень або роботи в офісі (табл. 4.18).

Таблиця 4.18

Страхові компанії, що надають відповідні послуги в Україні

Назва компанії	Основні продукти для ІТ	Рейтинг довіри	Додаткові послуги
ARX (екс-АХА)	Кіберстрахування, майно	Високий	Онлайн-управління полісами
Універсальна	Відповідальність, майно	Середній	Персональний менеджер для бізнесу
ПЗУ Україна	Персонал, кіберзахист	Високий	Комплексне обслуговування ІТ-компаній
TAS	Майно, фінансові ризики	Середній	Гнучкі тарифні плани

Рекомендації щодо впровадження страхування

- Перед запуском ІС доцільно укласти договір страхування кіберризиків — навіть мінімальна страховка покриє основні витрати у разі витоку даних.
- Страхування серверного обладнання є обов'язковим для компаній, що використовують локальні обчислювальні потужності.
- Страхування відповідальності перед третіми особами стане запорукою юридичної безпеки у випадку судових позовів.

Таким чином, система страхування є надійним інструментом у структурі захисту інформаційної системи логістики, який дає змогу мінімізувати втрати, зберегти репутацію проекту та забезпечити безперервність його функціонування у довгостроковій перспективі.

Висновки

У процес виконання дипломної роботи охоплює повний цикл розробки інформаційної системи, призначеної для управління логістичною діяльністю в оптовій торгівлі. Основні результати дослідження можна сформулювати наступним чином:

- 1. Проведено ґрунтовний аналіз предметної області.** Основними проблемами управління логістикою в сучасних умовах є неефективне планування маршрутів, недостатня автоматизація процесів та високі транспортні витрати. На основі аналізу були сформульовані цілі, завдання та очікувані результати впровадження системи.
- 2. Досліджено сучасні методи та моделі управління логістичними процесами.** Розглядаються як традиційні, так і передові методології, що охоплюють методи оптимізації, такі як жадібні алгоритми, алгоритм Дейкстри, генетичні алгоритми та інші. Запропоновано комплексну модель управління, яка враховує ресурсні обмеження, часові обмеження та пріоритетність вантажу.
- 3. Сформовано логічну структуру та архітектуру інформаційної системи.** Описано ключові компоненти системи, їх функціональне призначення та взаємодію між собою. Визначено основні параметри, необхідні для ефективної роботи: точність координат, часові інтервали, характеристики транспорту, обмеження складів.
- 4. Розроблено програмну реалізацію системи.** Обрано відповідний стек технологій, що включає Python для обчислювальної логіки, PostgreSQL для зберігання даних, JavaScript та бібліотеки для візуалізації маршрутів. Створено користувацький інтерфейс, реалізовано підготовку вхідних даних, логіку обробки запитів та візуалізацію маршрутів.
- 5. Розглянуто приклад практичного використання системи.** Проведено тестування на контрольному прикладі з реальними логістичними параметрами. Отримані результати показали ефективність алгоритмів у

зниженні витрат на транспортування, скороченні часу обслуговування клієнтів та оптимізації навантаження транспорту.

6. **Проведено техніко-економічне обґрунтування проекту.** Було оцінено економічну доцільність впровадження системи. Було розраховано комплексні витрати на розробку та обслуговування, проаналізовано ринок збуту та конкурентне середовище, оцінено ризики та сформульовано фінансові плани. Було визначено, що проект є комерційно здійсненним з можливістю масштабування.

Отже, побудована інформаційна система відповідає поставленим цілям і завданням, є прикладом сучасного підходу до автоматизації логістичних процесів і може бути застосована до реальних бізнес-операцій оптових торговельних фірм.

Використаних література

1. Андриющенко В. О. Методи оптимізації логістичних систем : навч. посіб. / В. О. Андриющенко. – К. : КНЕУ, 2017. – 180 с.
2. Афанасьєв М. Ю. Алгоритми та структури даних у логістиці / М. Ю. Афанасьєв. – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 252 с.
3. Балабанова Л. В. Логістика: підручник. – К. : Центр учбової літератури, 2012. – 424 с.
4. Балабанова Л. В. Логістика. Навчальний посібник. — К.: Центр учбової літератури, 2021. — 376 с.
5. Бакуменко О. Д. Логістичні системи: навч. посіб. – К. : Центр учбової літератури, 2020. – 312 с.
6. Banks J., Carson J., Nelson B., Nicol D. Discrete-Event System Simulation. – Prentice Hall, 2010.
7. Bertsekas D.P. Dynamic Programming and Optimal Control. – Athena Scientific, 2017.
8. Borshchev A. The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 8. – AnyLogic North America, 2018.
9. Бойко А. М. Кластеризація як інструмент логістичного планування // Вісник економіки транспорту і промисловості. – 2020. – № 70. – С. 112–117.
10. Christopher M. Logistics and Supply Chain Management. – FT Press, 2016.
11. Chopra S., Meindl P. Supply Chain Management: Strategy, Planning, and Operation. – Pearson, 2019.
12. Dorigo M., Stützle T. Ant Colony Optimization. – MIT Press, 2004.
13. Дослідження логістичного ринку України – KPMG Ukraine, 2023.
14. Джонсон Дж., Вуд Д. Управління ланцюгами постачання. – Львів: УАД, 2020. – 528 с.
15. Дьяків А. В. Алгоритми та методи оптимізації в логістиці: монографія. – Львів : ЛНУ ім. І. Франка, 2019. – 198 с.
16. Гайдук Л. П. Основи логістики: Навч. посіб. – К. : Знання, 2018. – 272 с.

17. Гнатенко І. Л., Скляренко А. О. Логістика. – Харків: ХНЕУ ім. С. Кузнеця, 2014.
18. Гнибіденко І. Ф. Інформаційні системи і технології в логістиці: навч. посіб. – К. : КНЕУ, 2021. – 256 с.
19. Гусак А. В. Евристичні методи у транспортній логістиці // Економіка та суспільство. – 2021. – № 28. – С. 67–73.
20. Gudehus T., Kotzab H. Comprehensive Logistics. – Springer, 2012.
21. Hillier F. S., Lieberman G. J. Introduction to Operations Research. – McGraw-Hill, 2020.
22. Holland J. H. Adaptation in Natural and Artificial Systems. – University of Michigan Press, 1992.
23. Жук В. І. Моделювання логістичних процесів: імітаційний підхід. – Львів : ЛНУ ім. І. Франка, 2018. – 198 с.
24. Каплунов Є. А. Сучасна логістика: теорія і практика. – Харків : ХНЕУ, 2020. – 348 с.
25. Kennedy J., Eberhart R. Particle Swarm Optimization. – IEEE Int. Conf. on Neural Networks, 1995.
26. Коваленко А. О. Логістика: теорія та практика. – Львів: ЛНУ ім. І. Франка, 2018.
27. Коваленко І. І. Економіко-математичне моделювання логістичних систем. – К. : КНЕУ, 2022. – 288 с.
28. Ковальчук І. І. Логістика підприємства. – К. : КНЕУ, 2019.
29. Ковальчук П. С. Геоінформаційні системи у міській логістиці // Логістика: теорія і практика. – 2022. – № 3(45). – С. 32–39.
30. Коломієць О. М. Логістичні інформаційні системи. – Харків: ХНУРЕ, 2019. – 210 с.
31. Коробкін В. Ю. Логістика: підручник. – Київ : Академвидав, 2021. – 400 с.
32. Law A., Kelton W. Simulation Modeling and Analysis. – McGraw-Hill, 2014.
33. Литвиненко О. В. Системна динаміка в логістиці. – К. : Логос, 2023. – 198 с.

34. Sterman J.D. Business Dynamics: Systems Thinking and Modeling for a Complex World. – Irwin/McGraw-Hill, 2000

Додаток А

```

39 // OptimizeRoutes generates optimized delivery routes based on delivery points with pallets
40 func (s *RouteOptimizerService) OptimizeRoutes(ctx context.Context, deliveryDate time.Time) ([]*models.Route, error) {
41     // Get all delivery points
42     deliveryPoints, err := s.DeliveryRepo.GetAll(ctx)
43     if err != nil {
44         return nil, fmt.Errorf("failed to get delivery points: %w", err)
45     }
46
47     // Filter out delivery points with no pallets
48     pointsWithPallets := make([]*models.DeliveryPoint, 0)
49     for _, point := range deliveryPoints {
50         if point.Pallets > 0 {
51             pointsWithPallets = append(pointsWithPallets, point)
52         }
53     }
54
55     if len(pointsWithPallets) == 0 {
56         return nil, errors.New("no delivery points with pallets found")
57     }
58
59     // Get all available trucks
60     availableTrucks, err := s.TruckRepo.GetAvailableTrucks(ctx)
61     if err != nil {
62         return nil, fmt.Errorf("failed to get available trucks: %w", err)
63     }
64
65     if len(availableTrucks) == 0 {
66         // If no trucks are available, try to reset all trucks to available
67         log.Println("No available trucks found, attempting to reset truck availability")
68         allTrucks, err := s.TruckRepo.GetAll(ctx)
69         if err != nil {
70             return nil, fmt.Errorf("failed to get all trucks: %w", err)
71         }
72
73         // Make all trucks available
74         for _, truck := range allTrucks {
75             truck.Available = true
76             if err := s.TruckRepo.Update(ctx, truck); err != nil {
77                 log.Printf("Warning: Failed to reset availability for truck %d: %v", truck.ID, err)
78             }

```

Рис.1 – Головний метод оптимізації маршрутів

```

79         }
80
81         // Get available trucks again
82         availableTrucks, err = s.TruckRepo.GetAvailableTrucks(ctx)
83         if err != nil {
84             return nil, fmt.Errorf("failed to get available trucks after reset: %w", err)
85         }
86
87         if len(availableTrucks) == 0 {
88             return nil, errors.New("no trucks available in the system")
89         }
90
91         log.Printf("Successfully reset %d trucks to available", len(availableTrucks))
92     }
93
94     // Sort trucks by capacity (largest first)
95     sort.Slice(availableTrucks, func(i, j int) bool {
96         return availableTrucks[i].Capacity > availableTrucks[j].Capacity
97     })
98
99     // Create a map of delivery points for easier access
100    pointMap := make(map[int]*models.DeliveryPoint)
101    for _, point := range deliveryPoints {
102        pointMap[point.ID] = point
103    }
104
105    // Create a map to track remaining pallets for each delivery point
106    remainingPalletsMap := make(map[int]int)
107    for _, point := range pointsWithPallets {
108        remainingPalletsMap[point.ID] = point.Pallets
109    }
110
111    // Check for delivery points with pallets exceeding the largest truck capacity
112    // and split them into multiple deliveries
113    largestTruckCapacity := 0
114    if len(availableTrucks) > 0 {
115        largestTruckCapacity = availableTrucks[0].Capacity
116    }
117
118    // Create a map to track which points have been assigned to routes

```

Рис.1.2 – Продовження (Головний метод оптимізації маршрутів)

```

119     assignedPoints := make(map[int]bool)
120
121     // Process delivery points by category, starting with Blue (which needs the largest trucks)
122     categories := []string{"Blue", "Green", "Yellow", "Purple"}
123
124     // First, create a map of points by category
125     categoryPoints := make(map[string][]*models.DeliveryPoint)
126     for _, point := range pointsWithPallets {
127         if !assignedPoints[point.ID] {
128             // Only add the point if it hasn't been fully processed yet
129             if point.Pallets <= largestTruckCapacity || remainingPalletsMap[point.ID] > 0 {
130                 categoryPoints[point.Category] = append(categoryPoints[point.Category], point)
131             }
132         }
133     }
134
135     // Generate routes
136     routes := make([]*models.Route, 0)
137
138     // Process categories in order (Blue, Green, Yellow, Purple)
139     for _, category := range categories {
140         if points, ok := categoryPoints[category]; ok && len(points) > 0 {
141             log.Printf("Processing %d %s category points", len(points), category)
142
143             // Determine the preferred truck capacity for this category
144             preferredCapacity := 0
145             var alternativeCapacities []int
146
147             switch category {
148             case "Blue":
149                 preferredCapacity = 33
150                 alternativeCapacities = []int{18, 15}
151             case "Green":
152                 preferredCapacity = 18
153                 alternativeCapacities = []int{15, 10}
154             case "Yellow":
155                 preferredCapacity = 15
156                 alternativeCapacities = []int{10}
157             case "Purple":
158                 preferredCapacity = 10
159                 alternativeCapacities = []int{}

```

Рис.1.3 – Продовження (Головний метод оптимізації маршрутів)

```

161
162 // Try to use the preferred truck capacity first
163 categoryRoutes, err := s.generateRoutesForCategory(ctx, points, availableTrucks, preferredCapacity, pointMap, deliveryDate)
164 if err != nil {
165     // If preferred trucks aren't available, try alternative capacities
166     log.Printf("Could not use %d-pallet trucks for %s category: %v. Trying alternative trucks.",
167         preferredCapacity, category, err)
168
169     if len(alternativeCapacities) > 0 {
170         splitRoutes, splitErr := s.generateSplitRoutesForCategory(ctx, points, availableTrucks,
171             alternativeCapacities, pointMap, deliveryDate)
172         if splitErr == nil {
173             log.Printf("Successfully created %d routes with alternative trucks for %s category",
174                 len(splitRoutes), category)
175             routes = append(routes, splitRoutes...)
176
177             // Mark points as assigned
178             for _, route := range splitRoutes {
179                 for _, pointID := range route.DeliveryPoints {
180                     assignedPoints[pointID] = true
181                 }
182             }
183         } else {
184             log.Printf("Failed to create routes with alternative trucks for %s category: %v",
185                 category, splitErr)
186         }
187     } else {
188         log.Printf("No alternative truck capacities available for %s category", category)
189     }
190 } else {
191     log.Printf("Successfully created %d routes with %d-pallet trucks for %s category",
192         len(categoryRoutes), preferredCapacity, category)
193     routes = append(routes, categoryRoutes...)
194
195     // Mark points as assigned
196     for _, route := range categoryRoutes {
197         for _, pointID := range route.DeliveryPoints {
198             assignedPoints[pointID] = true
199         }
200     }

```

Рис.1.4 – Продовження (Головний метод оптимізації маршрутів)

```

201     }
202 }
203 }
204
205 // If no routes were generated, return an error
206 if len(routes) == 0 {
207     return nil, errors.New("failed to generate any routes")
208 }
209
210 // Set route IDs and other properties
211 nextRouteID, err := s.RouteRepo.GetNextRouteID(ctx)
212 if err != nil {
213     return nil, fmt.Errorf("failed to get next route ID: %w", err)
214 }
215
216 for i, route := range routes {
217     route.ID = nextRouteID + i
218     route.DeliveryDate = deliveryDate
219     route.Status = models.RouteStatus.Planned
220 }
221
222 return routes, nil
223 }

```

Рис.1.5 – Продовження (Головний метод оптимізації маршрутів)