

Національний лісотехнічний університет України

(повне найменування вищого навчального закладу)

Навчально-науковий інститут комп'ютерних наук

та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

Магістерська кваліфікаційна робота

другий (магістерський)

(рівень вищої освіти)

на тему: “Інформаційно-аналітична система підтримки управлінських рішень у сфері логістики на основі даних користувачів”

Виконав: студент б курсу групи КН-62

спеціальності

122 “Комп'ютерні науки”

(шифр і назва напрямку підготовки, спеціальності)

Бунечко Ю.М.

(прізвище та ініціали)

Керівник Сало М. Ф., Карашецький В. П.

(прізвище та ініціали)

Рецензент Мокрицька О.В.

(прізвище та ініціали)

Львів – 2025

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій

Кафедра комп'ютерних наук

Рівень вищої освіти другий (магістерський)

Спеціальність 122 "Комп'ютерні науки"

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри



Борецька І. Б.

" 10 " грудня 2025 року

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Бунечко Юрій Миколайович

(прізвище, ім'я, по батькові)

1. Тема роботи "Інформаційно-аналітична система підтримки управлінських рішень у сфері логістики на основі даних користувачів"
керівник роботи Сало М. Ф. старший викладач, Карашецький В. П. кандидат технічних наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "29" квітня 2025 року № С-288

2. Термін подання студентом роботи: 10 грудня 2025 року

3. Вихідні дані до роботи: гетерогенні потоки інформації від логістичної компанії, що спеціалізується на доставці товарів з інтернет-магазинів Європи, США та Канади, які включають параметри замовлень, фінансові транзакції, трекінг-логи та історію взаємодії з клієнтами.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити): охоплює аналіз сучасних логістичних процесів та ідентифікацію проблеми фрагментації даних між WMS, TMS та CRM. Центральною частиною є розробка концептуальної моделі, проектування архітектури сховища даних (DWH) та ETL-процесів для нормалізації гетерогенних даних.

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

6. Дата видачі завдання: 1 травня 2025 року

Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз предметної області та формулювання проблеми	02.05.2025	виконано
2.	Огляд аналогів інформаційних систем і постановка задачі	03.11.2025	виконано
3.	Побудова інформаційної та концептуальної моделі системи	10.11.2025	виконано
4.	Обґрунтування математичних методів аналізу та алгоритмів рішень	17.11.2025	виконано
5.	Опис архітектури ПЗ та реалізація основних функцій прототипу	01.12.2025	виконано
6.	Проведення тестування та оцінка ефективності системи	05.12.2025	виконано
7.	Формування висновків, анотації та оформлення курсового проекту у фінальному вигляді	10.12.2025	виконано

Студент



(підпис)

Гуменко Ю.В.
(прізвище та ініціали)

Керівник роботи



(підпис)

Салом М.Ф.
(прізвище та ініціали)

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробленню інформаційно-аналітичної системи (ІАС) підтримки прийняття управлінських рішень у сфері міжнародної логістики. Об'єктом дослідження є процеси управління даними користувачів у логістичних компаніях, а предметом – методи та засоби інтелектуального аналізу даних для оптимізації доставки. В роботі проаналізовано сучасний стан ринку транскордонної (cross-border) електронної торгівлі та ідентифіковано проблеми фрагментації даних, що негативно впливають на клієнтський досвід логістичної компанії. Здійснено огляд існуючих ERP, CRM та WMS систем, обґрунтовано необхідність створення спеціалізованого рішення. Розроблено концептуальну та логічну моделі системи, спроектовано архітектуру сховища даних (DWH) на базі PostgreSQL з використанням методології розмірного моделювання. Наукова новизна полягає в обґрунтуванні комплексного використання математичного забезпечення: методу k-means для кластеризації користувачів, моделі експоненційного згладжування Хольта-Вінтерса для прогнозування попиту на логістичні послуги та евристичних алгоритмів маршрутизації. Програмна реалізація виконана з використанням мов програмування Java (backend) та JavaScript/Vue.js (frontend), а також інструментарію Power BI для візуалізації аналітичних звітів. Розроблено стартап-проект впровадження системи, розраховано показники економічної ефективності та проведено аналіз ризиків. Результати роботи дозволяють підвищити точність прогнозування замовлень та оптимізувати операційні витрати логістичного оператора.

Ключові слова: логістика, інформаційна система, дані користувачів, підтримка прийняття рішень, Data Mining, кластеризація, прогнозування, DWH, мультивалютність.

ABSTRACT

The master's thesis is devoted to the development of an information-analytical system (IAS) for management decision support in the field of international logistics based on user data. The object of the study is the processes of user data management in logistics companies, and the subject is the methods and tools of intelligent data analysis for delivery optimization. The thesis analyzes the current state of the cross-border e-commerce market and identifies data fragmentation issues affecting the customer experience of logistics company. An overview of existing ERP, CRM, and WMS systems was carried out, justifying the need for a specialized solution. Conceptual and logical models of the system have been developed, and a Data Warehouse (DWH) architecture based on PostgreSQL has been designed using dimensional modeling methodology. The scientific novelty lies in the justification of the integrated use of mathematical software: the k-means method for user clustering, the Holt-Winters exponential smoothing model for forecasting demand for logistics services, and heuristic routing algorithms. The software implementation is carried out using Java (backend) and JavaScript/Vue.js (frontend) programming languages, as well as Power BI tools for visualization of analytical reports. A startup project for the system implementation has been developed, economic efficiency indicators have been calculated, and risks have been analyzed. The results of the work allow increasing the accuracy of order forecasting and optimizing the operational costs of the logistics operator.

Keywords: logistics, information system, user data, decision support, Data Mining, clustering, forecasting, DWH, multicurrency.

ТЕХНІЧНЕ ЗАВДАННЯ

Розробити інформаційно-аналітичну систему, що дозволить логістичній компанії на основі даних користувачів автоматизувати прийняття управлінських рішень, покращити якість обслуговування клієнтів та підвищити ефективність операцій у сфері міжнародної доставки.

Завдання, які повинні бути вирішені у магістерській кваліфікаційній роботі:

1. Провести аналіз сучасного стану логістичних процесів, виявити проблеми клієнтського досвіду на прикладі логістичної компанії.
2. Вивчити існуючі рішення у сфері ІАС та визначити вимоги до системи підтримки управлінських рішень у логістиці.
3. Розробити інформаційну модель системи, створити UML- та ER-діаграми.
4. Обґрунтувати вибір математичного забезпечення: методів кластеризації, прогнозування, алгоритмів маршрутизації та агрегування.
5. Побудувати програмну архітектуру системи та реалізувати функціональні модулі прототипу.
6. Забезпечити конвертацію одиниць ваги та вартості для підтримки мультивалютності та мультисистемності.
7. Провести тестування прототипу, оцінити ефективність системи та сформулювати рекомендації щодо впровадження.

Особливі вимоги до реалізації:

- урахування вимог законодавства щодо захисту персональних даних (GDPR);
- використання сучасних інструментів: Power BI, Creatio, PostgreSQL, Java, Vue.js, PHP;
- адаптивність до мобільних і десктопних платформ.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	13
1.1. Аналіз сучасних логістичних процесів у сфері міжнародної доставки.....	13
1.2. Проблематика клієнтського досвіду та роль даних у її вирішенні.....	15
1.3. Огляд існуючих інформаційних систем підтримки прийняття рішень.....	16
1.4. Аналіз бізнес-процесів об'єкта дослідження.....	18
1.5. Висновки до розділу 1.....	20
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	21
2.1. Обґрунтування та розробка концептуальної моделі предметної області...	21
2.2. Проектування архітектури сховища даних та ETL-процесів.....	23
2.3. Розробка логічної структури бази даних.....	24
2.4. Політики управління даними та забезпечення приватності.....	26
2.5. Формування каталогу аналітичних показників.....	27
2.6. Висновки до розділу 2.....	28
РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	29
3.1. Теоретико-множинна формалізація та постановка задачі управління логістичними процесами.....	29
3.2. Методи попередньої обробки та нормалізації даних.....	32
3.3. Математичні моделі кластеризації та сегментації користувачів.....	37
3.4. Моделі прогнозування логістичних потоків.....	41
3.5. Алгоритми маршрутизації та транспортної оптимізації.....	43
3.6. Підтримка прийняття рішень методами багатокритеріального аналізу.....	45
Висновки до розділу 3.....	46
РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ.....	48
4.1. Обґрунтування вибору технологічного стеку та засобів розробки.....	48
4.2. Проектування архітектури програмного забезпечення.....	51
4.3. Реалізація рівня даних та механізмів збереження.....	54
4.4. Програмна реалізація алгоритмів та бізнес-логіки.....	56
4.5. Розробка прикладного програмного інтерфейсу.....	58

4.6. Розробка клієнтської частини.....	60
4.7. Тестування та забезпечення якості.....	61
4.8. Стандартизація та відповідність нормативним вимогам.....	62
4.9. Розгортання та експлуатація.....	63
Висновки до розділу 4.....	64
РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ.....	66
5.1. Опис ідеї стартап-проекту.....	66
5.2. Технологічний аудит та аналіз можливостей реалізації.....	69
5.3. Аналіз ринкових можливостей запуску стартап-проекту.....	71
5.4. Розроблення ринкової стратегії.....	74
5.5. Маркетингова програма стартап-проекту.....	76
5.6. Фінансово-економічне обґрунтування та оцінка ризиків.....	78
Висновки до розділу 5.....	80
ЗАГАЛЬНІ ВИСНОВКИ.....	81
ДОДАТКИ.....	84
ДОДАТОК А.....	84
ДОДАТОК Б.....	89
ДОДАТОК В.....	93
ДОДАТОК Г.....	97
ДОДАТОК Д.....	101

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

IAC – інформаційно-аналітична система

ПЗ – програмне забезпечення

UI – користувацький інтерфейс (User Interface)

UX – досвід користувача (User Experience)

BI – бізнес-аналітика (Business Intelligence)

CRM – система управління взаємовідносинами з клієнтами (Customer Relationship Management)

KPI – ключовий показник ефективності (Key Performance Indicator)

API – програмний інтерфейс додатків (Application Programming Interface)

SQL – мова структурованих запитів (Structured Query Language)

WMS – система управління складом (Warehouse Management System)

TMS – система управління транспортуванням (Transportation Management System)

PO – власник продукту (Product Owner)

PM – проєктний менеджер (Project Manager)

GDPR – Загальний регламент захисту даних (General Data Protection Regulation)

UML – уніфікована мова моделювання (Unified Modeling Language)

ER – модель «сутність-зв'язок» (Entity-Relationship)

ВСТУП

Актуальність теми

Сучасний етап розвитку світової економіки характеризується стрімкою цифровізацією всіх сфер суспільного життя та виробничої діяльності. Логістика, як одна з ключових галузей, що забезпечує функціонування глобальних ринків, зазнає фундаментальних трансформацій під впливом новітніх інформаційних технологій. Зокрема, сектор міжнародної електронної комерції (cross-border e-commerce) демонструє стабільну тенденцію до зростання, що зумовлює необхідність перегляду традиційних підходів до управління ланцюгами постачання.

Для логістичних операторів, таких як сервіс логістичної компанії, що спеціалізується на доставці товарів з інтернет-магазинів Європи, США та Канади, критично важливим стає не лише фізичне переміщення вантажів, а й ефективне управління інформаційними потоками, що їх супроводжують. В умовах жорсткої конкуренції та зростаючих вимог споживачів до швидкості та прозорості сервісу, ключовим активом компанії стають дані. Накопичення великих обсягів інформації про поведінку користувачів, параметри замовлень, логістичні маршрути та фінансові транзакції створює передумови для впровадження концепції Data-Driven Decision Making (прийняття рішень на основі даних).

Проте, на практиці підприємства стикаються з проблемою фрагментації інформаційного простору. Використання різномірних інформаційних систем для управління складом (WMS), транспортом (TMS) та взаємовідносинами з клієнтами (CRM) призводить до виникнення «інформаційних розривів». Ситуація ускладнюється необхідністю оперувати в мультивалютному середовищі та використовувати різні метричні системи вимірювання ваги та габаритів, що є характерним для міжнародної логістики. Відсутність єдиного інтегрованого джерела достовірної інформації унеможливорює оперативний

моніторинг ключових показників ефективності, ускладнює прогнозування попиту та знижує якість стратегічного планування.

У зв'язку з цим, розробка спеціалізованої інформаційно-аналітичної системи (ІАС), здатної консолідувати дані з різномірних джерел, забезпечувати їх нормалізацію та надавати інструменти для інтелектуального аналізу, є актуальним науково-прикладним завданням. Така система дозволить трансформувати накопичені «сирі» дані в цінні знання, необхідні для оптимізації бізнес-процесів та підвищення рівня клієнтського сервісу.

Зв'язок роботи з науковими програмами, планами, темами

Магістерська кваліфікаційна робота виконана відповідно до пріоритетних напрямів наукових досліджень кафедри комп'ютерних наук Національного лісотехнічного університету України в рамках підготовки фахівців за спеціальністю 122 «Комп'ютерні науки». Тематика роботи узгоджена з планами науково-дослідної роботи університету щодо розробки та впровадження інтелектуальних інформаційних систем у різних галузях економіки.

Мета і завдання дослідження

Метою роботи є підвищення ефективності управлінських рішень у сфері міжнародної логістики шляхом розробки концепції, моделей та програмного забезпечення інформаційно-аналітичної системи на основі агрегованих даних користувачів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести детальний аналіз стану та тенденцій розвитку логістичних інформаційних систем у сфері міжнародної доставки.
2. Дослідити проблематику клієнтського досвіду та визначити роль аналітики даних у підвищенні лояльності користувачів.

3. Здійснити огляд існуючих інструментальних засобів (WMS, TMS, CRM, BI) та обґрунтувати необхідність створення інтегрованого рішення.
4. Розробити концептуальну модель предметної області та спроектувати архітектуру сховища даних (DWH).
5. Обґрунтувати вибір математичних методів для вирішення задач кластеризації користувачів, прогнозування обсягів замовлень та оптимізації маршрутів.
6. Розробити алгоритмічне забезпечення для нормалізації, очищення та конвертації даних.
7. Спроектувати та реалізувати програмний прототип системи, провести його тестування.
8. Розробити стартап-проект впровадження системи та оцінити його економічну ефективність.

Об'єкт дослідження – процеси інформаційної підтримки прийняття управлінських рішень у логістичних компаніях, що здійснюють міжнародну доставку товарів.

Предмет дослідження – моделі, методи, алгоритми та програмні засоби створення інформаційно-аналітичної системи обробки даних користувачів логістичних сервісів.

Наукова новизна отриманих результатів полягає у наступному:

1. Удосконалено метод інтеграції гетерогенних даних у логістичних системах, який, на відміну від існуючих, передбачає попередню семантичну нормалізацію різнорідних метрик (вага, валюта) на етапі ETL-процесів, що забезпечує цілісність аналітичної моделі.
2. Набуло подальшого розвитку застосування гібридних моделей прогнозування часових рядів для оцінки навантаження на логістичні хаби, що враховують сезонні фактори та маркетингові активності.

Практичне значення отриманих результатів

Розроблена архітектура та програмний прототип ІАС дозволяють автоматизувати процес підготовки управлінської звітності, підвищити точність прогнозування ресурсів та оптимізувати операційні витрати. Результати роботи можуть бути впроваджені в діяльність компанії поточної логістичної та інших логістичних операторів.

Структура роботи

Магістерська робота складається зі вступу, п'яти розділів, висновків, списку використаних джерел та додатків. Повний обсяг роботи становить 104 сторінки.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Аналіз сучасних логістичних процесів у сфері міжнародної доставки

Глобалізація світової торгівлі та стрімкий розвиток цифрових технологій призвели до кардинальних змін у структурі логістичних потоків. Традиційна модель B2B (Business-to-Business), що оперувала великими партіями вантажів (контейнерами, палетами), поступово поступається місцем або доповнюється моделлю B2C (Business-to-Consumer), яка характеризується величезною кількістю дрібних відправлень. Цей феномен, відомий як cross-border e-commerce, створює нові виклики для логістичних операторів.

Специфіка міжнародної доставки товарів кінцевому споживачеві полягає у необхідності інтеграції складних митних процедур, різнорідних транспортних мереж та вимог «останньої милі» в єдиний, прозорий для клієнта сервіс. Розглянемо детальніше структуру логістичного ланцюга на прикладі сервісів форвардингу (mail forwarding), до яких належить поточна логістична компанія.

Процес доставки можна декомпонувати на наступні ключові етапи:

1. **«Перша миля» (First Mile) та ініціація замовлення.** Процес починається з моменту здійснення покупки клієнтом в іноземному інтернет-магазині. Оскільки багато магазинів не пропонують прямої доставки в країну покупця, клієнт використовує адресу проміжного складу (хабу) логістичного оператора в країні покупки. На цьому етапі в інформаційній системі оператора реєструється «очікуване відправлення» (pre-alert), що містить попередні дані про товар, його вартість та трек-номер локальної служби доставки.
2. **Складська обробка (Inbound & Processing).** Надходження товару на склад є критичною точкою генерації даних. Складські операції включають:
 - Фізичне приймання та сканування вхідного трек-номера.

- Зважування та вимірювання габаритів. Тут виникає проблема різнорідності метричних систем: склади в США оперують фунтами (lbs) та дюймами, тоді як європейські склади – кілограмами та сантиметрами.
 - Ідентифікація клієнта та прив'язка посилки до його акаунту (suite number).
 - Додаткові послуги: перевірка товару, фотозвіт, перепакування.
3. **Консолідація та формування відправлень.** Клієнт може об'єднати декілька дрібних посилок в одне міжнародне відправлення для економії на вартості доставки. Цей процес вимагає складної логіки перерахунку ваги (об'ємна вага проти фізичної) та вартості.
 4. **Експортна підготовка та митне оформлення.** Для перетину кордону формуються митні декларації та маніфести. Важливим аспектом є коректна класифікація товарів за кодами ТН ЗЕД (HS codes) та декларування вартості для нарахування митних платежів. Дані про вартість можуть надходити в різних валютах (USD, EUR, GBP, CAD, PLN), що вимагає їх конвертації за актуальними крос-курсами.
 5. **Магістральне перевезення (Linehaul).** Транспортування консолідованих вантажів (авіаційним, морським або автомобільним транспортом) до країни призначення. На цьому етапі важливим є відстеження статусу контейнера або палети, що дозволяє прогнозувати час прибуття (ETA - Estimated Time of Arrival).
 6. **Імпорт, сортування та «Остання миля» (Last Mile).** Після прибуття в країну призначення та проходження імпортного митного очищення, вантаж розформовується, і окремі посилки передаються на доставку кінцевому споживачеві через кур'єрські служби або мережу поштоматів.

Ефективне управління цим складним ланцюгом неможливе без використання сучасних інформаційних технологій. Дані, що генеруються на кожному етапі, є цінним активом, який дозволяє оптимізувати маршрути, зменшити витрати та

покращити клієнтський досвід. Однак, обсяг та різноманітність цих даних створюють значні труднощі для їх обробки та аналізу.

1.2. Проблематика клієнтського досвіду та роль даних у її вирішенні

У сфері B2C логістики клієнтський досвід (Customer Experience, CX) є основним фактором конкурентоспроможності. Користувачі очікують не лише своєчасної доставки, а й повної прозорості процесу, зручності взаємодії з сервісом та персоналізованого підходу.

Аналіз відгуків користувачів та внутрішніх процесів сервісів міжнародної доставки дозволяє виділити ряд типових проблем:

- 1. Складність онбордингу та реєстрації.** Процес реєстрації в сервісах форвардингу часто є перевантаженим вимогами безпеки. Необхідність підтвердження особи, верифікації платіжних карток та введення детальних адрес (Billing/Shipping) створює високий поріг входження для нових користувачів. Аналітика даних на етапі реєстрації (аналіз воронки конверсії) дозволяє виявити етапи, на яких користувачі найчастіше відмовляються від завершення реєстрації, та оптимізувати цей процес.
- 2. Інформаційна невизначеність.** Користувачі часто стикаються з проблемою нерозуміння доступних маршрутів доставки, тарифних планів та митних лімітів. Відсутність чіткої візуалізації можливостей сервісу призводить до помилок при замовленні та, як наслідок, до затримок або повернень товарів.
- 3. Непрозорість статусу замовлення.** «Чорні діри» у трекінгу, коли статус посилки не оновлюється протягом тривалого часу (наприклад, під час магістрального перевезення), викликають тривогу у клієнтів та збільшують навантаження на службу підтримки.
- 4. Проблеми з розрахунком вартості.** Складність тарифних сіток, наявність прихованих платежів (об'ємна вага, митні збори) та курсові

різниці часто призводять до того, що фінальна вартість доставки відрізняється від очікуваної клієнтом.

Вирішення цих проблем лежить у площині побудови ефективної інформаційно-аналітичної системи. Збір та аналіз даних про поведінку користувачів, історію їхніх замовлень та звернень до підтримки дозволяє:

- Персоналізувати комунікацію та пропозиції.
- Прогнозувати можливі проблеми з доставкою та проактивно інформувати клієнта.
- Оптимізувати інтерфейси користувача на основі реальних патернів використання.
- Впроваджувати динамічні системи знижок та лояльності для утримання клієнтів.

1.3. Огляд існуючих інформаційних систем підтримки прийняття рішень

Сучасний ринок програмного забезпечення для логістики пропонує широкий спектр спеціалізованих рішень. Для розуміння місця та ролі проєктованої ІАС, необхідно провести аналіз основних класів систем, що використовуються в галузі.

1.3.1. Системи управління складом (WMS - Warehouse Management Systems)

WMS призначені для автоматизації та оптимізації всіх процесів складської діяльності: від приймання товарів до їх відвантаження.

- *Основні функції:* Облік запасів, управління розміщенням товарів (slotting), комплектація замовлень (picking), інвентаризація, управління персоналом складу.
- *Приклади:* Manhattan Associates, HighJump, Oracle WMS, а також кастомні рішення.

- *Обмеження для задач аналітики:* WMS фокусуються на операційній ефективності в межах «чотирьох стін» складу. Вони, як правило, не містять даних про фінансові показники всієї доставки, клієнтський досвід або маркетингові кампанії.

1.3.2. Системи управління транспортом (TMS - Transportation Management Systems)

TMS забезпечують планування, виконання та оптимізацію фізичного переміщення вантажів.

- *Основні функції:* Планування маршрутів, вибір перевізників, розрахунок фрахту, відстеження вантажів в дорозі, управління парком транспортних засобів.
- *Приклади:* SAP TM, Oracle TMS, MercuryGate.
- *Обмеження:* TMS орієнтовані на вантажі та транспортні засоби, а не на кінцевого споживача. Аналітика TMS зазвичай обмежується транспортними витратами та термінами доставки, не враховуючи комерційну маржинальність замовлення.

1.3.3. Системи управління взаємовідносинами з клієнтами (CRM - Customer Relationship Management)

CRM-системи є центральним сховищем даних про клієнтів та історію взаємодії з ними.

- *Основні функції:* Ведення бази контактів, управління лідами та угодами, історія комунікацій, управління маркетинговими кампаніями, обслуговування клієнтів.
- *Приклади:* Salesforce, Creatio, HubSpot, Zoho CRM.
- *Обмеження:* Хоча CRM містять важливі дані про клієнтів, вони не пристосовані для обробки великих масивів специфічних логістичних даних (вага, габарити, детальний трекінг) та виконання складних

аналітичних розрахунків (наприклад, собівартості доставки кг вантажу на певному маршруті). Крім того, CRM зазвичай не мають вбудованих інструментів для нормалізації мультивалютних даних для фінансової аналітики.

1.3.4. Системи бізнес-аналітики (BI - Business Intelligence)

BI-платформи призначені для візуалізації даних та побудови інтерактивних звітів.

- *Приклади:* Microsoft Power BI, Tableau, Qlik Sense.
- *Обмеження:* BI-інструменти є потужним засобом візуалізації, але вони не вирішують проблему якості та інтеграції даних. Якщо підключити BI безпосередньо до «сирих» баз даних WMS або CRM, отримані звіти будуть містити помилки через неузгодженість даних («garbage in, garbage out»).

Висновок: жоден із розглянутих класів систем не забезпечує комплексної підтримки прийняття рішень на рівні всього бізнесу міжнародної доставки. Існує потреба в створенні інтегруючої надбудови – спеціалізованої Інформаційно-Аналітичної Системи (ІАС), яка б консолідувала дані з WMS, TMS та CRM, виконувала їх очищення та нормалізацію, і надавала єдину точку доступу до аналітичної інформації.

1.4. Аналіз бізнес-процесів об'єкта дослідження

Об'єктом дослідження в даній роботі є сервіс логістичної компанії. Для коректного проектування ІАС необхідно розуміти специфіку бізнес-процесів компанії.

Компанія оперує за моделлю форвардингу, надаючи клієнтам «віртуальні адреси» в різних країнах світу. Основний дохід компанія отримує від тарифів на

доставку (які залежать від ваги та країни) та додаткових послуг (консолідація, пакування, страхування).

Основні інформаційні сутності та їх життєвий цикл:

1. **Клієнт (User):** Реєструється в системі, проходить верифікацію. Має унікальний ID (Suite ID).
2. **Вхідний пакет (Inbound Package):** Посилка від магазину. Характеризується трек-номером перевізника (UPS, DHL, FedEx), вагою, відправником.
3. **Відправлення (Shipment/Order):** Створюється клієнтом. Може складатися з одного або декількох вхідних пакетів. Це об'єкт тарифікації.
4. **Транзакція (Transaction):** Запис про нарахування коштів (за послуги) або оплату клієнтом.

Ключові проблеми в управлінні даними логістичної компанії:

- **Мультивалютність:** Тарифи перевізників виставляються в локальних валютах (EUR, USD, CAD), а тарифи для клієнтів можуть бути фіксовані в USD. Це створює курсові різниці, які складно відстежувати в реальному часі.
- **Мультиодичність:** Склади в США та Великобританії часто надають дані про вагу в фунтах (lbs), тоді як решта світу – в кілограмах. Помилки при конвертації можуть призводити до суттєвих фінансових втрат або некоректного нарахування плати клієнтам.
- **Фрагментація:** Дані про вагу знаходяться в WMS, дані про клієнта – в CRM, а дані про фінанси – в білінговій системі. Зведення звіту «Прибутковість по напрямку США-Україна» вимагає ручного вивантаження даних з трьох систем та їх обробки в Excel, що займає багато часу і призводить до помилок.

1.5. Висновки до розділу 1

1. Проведений аналіз показав, що сфера міжнародної логістики характеризується високою складністю інформаційних потоків та динамічністю процесів.
2. Виявлено, що існуючий IT-ландшафт, який складається з окремих систем (WMS, CRM, TMS), не забезпечує необхідного рівня інформаційної підтримки для прийняття стратегічних рішень. Основною проблемою є фрагментація даних та відсутність єдиного джерела правди (Single Source of Truth).
3. Ключовими викликами для аналітики є необхідність роботи в мультивалютному та мультиодичному середовищі, а також забезпечення приватності даних користувачів.
4. Обґрунтовано доцільність розробки спеціалізованої ІАС, яка базуватиметься на технологіях сховищ даних (DWH) та бізнес-аналітики (BI). Така система дозволить інтегрувати розрізнені дані, забезпечити їх якість та надати керівництву інструменти для оперативного моніторингу та прогнозування діяльності компанії.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1. Обґрунтування та розробка концептуальної моделі предметної області

Ефективність будь-якої інформаційно-аналітичної системи (ІАС) безпосередньо залежить від якості спроектованої інформаційної моделі. На етапі концептуального моделювання здійснюється абстрагування від технічних деталей реалізації (конкретної СУБД, типів полів) та фокусування на сутності бізнес-процесів, об'єктах реального світу та зв'язках між ними.

Для формалізації предметної області міжнародної логістики в даній роботі використано об'єктно-орієнтований підхід та уніфіковану мову моделювання (UML – Unified Modeling Language). Це дозволяє створити модель, зрозумілу як для розробників, так і для бізнес-експертів.

Враховуючи специфіку діяльності логістичної компанії, в основу концептуальної моделі покладено такі ключові сутності (класи):

1. **Customer (Клієнт)**. У традиційних CRM-системах ця сутність містить детальні персональні дані (ПІБ, адреса, телефон). Проте, для задач аналітичної системи та з метою дотримання принципів GDPR (General Data Protection Regulation), ми вводимо поняття *Customer_Aggregate*. Ця сутність є «цифровим двійником» клієнта в аналітичному контурі, що містить лише знеособлені та агреговані атрибути, необхідні для прийняття рішень:

- *Segment ID*: Ідентифікатор маркетингового сегмента (напр., «New», «Active», «Churned»).
- *Lifetime Value (LTV)*: Сумарна цінність клієнта за весь час взаємодії.
- *Preferred Category*: Категорія товарів, яку клієнт замовляє найчастіше.
- *Geo Profile*: Країна та регіон доставки (без точної адреси).

2. **Order (Замовлення).** Центральна транзакційна сутність, що відображає факт замовлення послуги доставки. Вона пов'язує клієнта з товарами та логістичними процесами. Ключові атрибути:
 - *Order Date*: Дата створення замовлення.
 - *Declared Value*: Задекларована вартість (важлива для митниці та страхування).
 - *Total Weight*: Загальна вага замовлення.
 - *Status*: Поточний стан (напр., «Created», «Received», «Shipped», «Delivered»).
3. **Shipment (Відправлення / Посилка).** Фізична одиниця вантажу. Важливо розрізняти Order (логічна сутність, «що клієнт хоче отримати») та Shipment (фізична сутність, «коробка, що їде»). Одне замовлення може бути розбите на декілька відправлень (split shipment), або декілька замовлень консолідовані в одне відправлення. Цей зв'язок (M:N) є критичним для коректного розрахунку собівартості логістики.
4. **Warehouse (Склад/Хаб).** Вузлова точка логістичної мережі. Характеризується географічним розташуванням (Країна, Штат/Регіон), типом (транзитний хаб, склад консолідації, склад «останньої милі») та пропускною здатністю.
5. **Route (Маршрут).** Абстракція логістичного шляху, що з'єднує два склади або склад і клієнта. Маршрут характеризується перевізником (Carrier), типом транспорту (Air, Sea, Road) та нормативним часом доставки (Transit Time).
6. **Financial Transaction (Фінансова транзакція).** Запис про рух коштів. Включає нарахування за послуги (інвойсинг) та фактичні оплати клієнтом. Особливістю є необхідність зберігання валюти операції та курсу конвертації на момент транзакції.

Візуалізація взаємозв'язків між цими сутностями представлена на UML-діаграмі класів (див. Додаток А). Модель дозволяє вирішити проблему

неоднозначності даних, чітко розмежувавши, наприклад, дату створення замовлення в системі та дату його фізичного надходження на склад.

2.2. Проектування архітектури сховища даних та ETL-процесів

Для реалізації ІАС обрано архітектуру корпоративного сховища даних (DWH – Data Warehouse), що базується на методології CIF (Corporate Information Factory) Білла Інмона та розмірному моделюванні (Dimensional Modeling) Ральфа Кімбалла. Такий гібридний підхід дозволяє поєднати надійність зберігання історичних даних із високою швидкістю виконання аналітичних запитів.

Архітектура системи включає три основні рівні:

1. Рівень джерел даних (Data Sources).

- *CRM Creatio*: Джерело майстер-даних про клієнтів та історії комунікацій.
- *WMS (Warehouse Management System)*: Постачальник оперативних даних про рух посилок на складах (приймання, зважування, відвантаження).
- *TMS/Carrier API*: Зовнішні дані від перевізників (DHL, UPS, FedEx, Meest) про статуси доставки.
- *Financial System*: Дані про тарифи та платежі.

2. **Рівень інтеграції та обробки (ETL / Staging Area)**. Це критично важливий компонент, де відбувається трансформація «сирих» даних у придатну для аналізу інформацію. Процес ETL (Extract, Transform, Load) реалізує наступну логіку:

- **Extract (Витягнення)**: Інкрементальне завантаження даних (лише нових або змінених записів) для зменшення навантаження на мережу.
- **Transform (Перетворення)**:

- *Нормалізація одиниць виміру:* Це ключовий етап для вирішення проблеми мультиодичності. Всі вагові показники, що надходять у фунтах (lbs) зі складів США, автоматично конвертуються у кілограми (kg) за коефіцієнтом 0.453592. Це дозволяє в подальшому аналізувати загальний тоннаж вантажопотоку без помилок.
- *Мультивалютна конвертація:* Всі фінансові показники приводяться до єдиної базової валюти звітності (USD). Для цього використовується історична таблиця курсів валют. Важливо використовувати курс саме на дату операції, щоб зберегти історичну достовірність фінансових результатів.
- *Очищення даних (Data Cleansing):* Виявлення та виправлення помилок (наприклад, від'ємна вага, некоректні дати, дублікати).
- **Load (Завантаження):** Запис підготовлених даних у цільові таблиці DWH.

3. **Рівень зберігання (Data Warehouse).** Централізоване сховище, організоване за схемою «Зірка» (Star Schema), яке оптимізоване для швидкого читання та агрегації даних.

2.3. Розробка логічної структури бази даних

Для фізичної реалізації DWH обрано реляційну систему управління базами даних (СУБД) PostgreSQL. Вибір обумовлений її надійністю, підтримкою складних SQL-запитів, можливістю горизонтального масштабування та відсутністю ліцензійних витрат (Open Source).

Логічна структура бази даних спроектована за схемою «Зірка», яка є де-факто стандартом для аналітичних систем. Вона складається з центральних таблиць фактів, оточених таблицями вимірів.

Таблиці Фактів (Fact Tables): Зберігають кількісні показники бізнесу (метрики) та зовнішні ключі на таблиці вимірів.

- **FACT_ORDERS:** Основна таблиця, що містить деталізацію кожного замовлення.
 - *Metrics:* weight_kg (вага нетто), volumetric_weight_kg (об'ємна вага), declared_value_usd (вартість товару), shipping_cost_usd (вартість доставки), service_fee_usd (комісія сервісу).
 - *Keys:* fk_date_id, fk_customer_id, fk_route_id, fk_warehouse_id.
- **FACT_SLA_EVENTS:** Таблиця для аналізу якості сервісу. Фіксує відхилення від планових термінів.
 - *Metrics:* delay_days (кількість днів затримки), processing_time_hours (час обробки на складі).

Таблиці Вимірів (Dimension Tables): Містять описові атрибути (контекст), за якими здійснюється фільтрація, групування та зрізи даних (Slicing and Dicing).

- **DIM_DATE:** Календарний вимір. Дозволяє аналізувати дані в розрізі днів, тижнів, місяців, кварталів, років, а також виділяти робочі/вихідні дні та періоди свят (наприклад, «High Season» перед Різдом).
- **DIM_CUSTOMER_AGG:** Атрибути клієнтів. Важливо, що ця таблиця не містить персональних даних (ПІБ), а лише аналітичні атрибути: country_segment (країна проживання), registration_cohort (місяць реєстрації – для когортного аналізу), rfm_segment (сегмент лояльності).
- **DIM_ROUTE:** Опис логістичних маршрутів. Містить інформацію про країну відправлення (origin_country), країну призначення (destination_country), тип сервісу (експрес/економ) та магістрального перевізника.

- **DIM_PRODUCT_CATEGORY:** Ієрархічний довідник категорій товарів (Електроніка – Смартфони; Одяг – Взуття). Це дозволяє аналізувати структуру попиту.

ER-діаграма (Entity-Relationship Diagram), що візуалізує структуру бази даних та зв'язки між таблицями, наведена в Додатку Б.

2.4. Політики управління даними та забезпечення приватності

Впровадження ІАС, що оперує даними про користувачів (зокрема, резидентів Європейського Союзу), вимагає суворого дотримання регламенту GDPR. У системі реалізовано концепцію **Privacy by Design** (приватність на етапі проектування).

Основні політики управління даними (Data Governance) включають:

1. **Псевдонімізація (Pseudonymization).** На етапі ETL, перед завантаженням даних у DWH, всі прямі ідентифікатори особи (Name, Email, Phone, Address) видаляються або замінюються на необоротні хеш-коди (суррогатні ключі). Таблиця відповідності (Mapping Table) між реальним ID в CRM та хешем в DWH зберігається в окремому ізольованому контурі з суворим контролем доступу. Це гарантує, що навіть у випадку витоку даних з DWH, зловмисники не зможуть ідентифікувати конкретних осіб.
2. **Мінімізація даних (Data Minimization).** В аналітичне сховище передаються лише ті дані, які необхідні для вирішення конкретних аналітичних задач. Наприклад, для аналізу логістики достатньо знати поштовий індекс (Zip Code) та місто отримувача, точна вулиця та номер будинку в DWH не завантажуються.
3. **Контроль якості даних (Data Quality - DQ).** Реалізовано систему автоматичних DQ-правил, які перевіряють дані на вході в DWH:
 - *Completeness:* Перевірка наявності обов'язкових полів (наприклад, вага не може бути NULL).

- *Consistency*: Перевірка логічної узгодженості (дата доставки не може бути раніше дати відправлення).
- *Accuracy*: Перевірка на відповідність довідникам (код країни має відповідати стандарту ISO 3166). Записи, що не проходять перевірку, не потрапляють у звіти, а переміщуються в таблицю карантину (Quarantine Table) для ручного розбору аналітиками.

4. **Управління доступом (RBAC - Role-Based Access Control)**. Доступ до аналітичних звітів розмежовано за ролями. Менеджер напрямку США має доступ лише до даних по США. Фінансовий директор має доступ до агрегованих фінансових показників, але не до деталей окремих відправлень.

2.5. Формування каталогу аналітичних показників

Система розроблена для розрахунку та візуалізації збалансованої системи показників (Balanced Scorecard), що охоплює різні аспекти діяльності компанії. Каталог KPI сформовано на основі інтерв'ю зі стейкхолдерами (топ-менеджментом, операційними директорами).

Основні групи показників:

1. Фінансові показники:

- *Gross Margin per kg (Валовий прибуток на кг)*: Ключовий показник ефективності логістики. Розраховується як різниця між тарифом для клієнта та собівартістю доставки (включаючи «першу милю», склад, фрахт, митницю, «останню милю»), поділена на вагу.
- *Revenue Growth Rate*: Темп зростання доходу (MoM, YoY).

2. Операційні показники:

- *SLA Compliance Rate*: Відсоток відправлень, доставлених в межах обіцяного терміну.
- *Hub Processing Time*: Середній час, який посилка проводить на транзитному складі (від сканування входу до сканування виходу).

- *Return Rate*: Відсоток повернень та недоставлених відправлень.

3. Клієнтські показники:

- *Customer Retention Rate*: Відсоток клієнтів, які зробили повторне замовлення протягом 3/6/12 місяців.
- *NPS (Net Promoter Score)*: Індекс лояльності клієнтів.

Детальний опис формул розрахунку кожного KPI та приклад SQL-запитів для їх отримання наведено у Додатку В.

2.6. Висновки до розділу 2

У даному розділі розроблено комплексне інформаційне забезпечення проєктованої ІАС.

1. Створено концептуальну модель предметної області, яка враховує специфіку міжнародної логістики (сутності «Замовлення», «Відправлення», «Маршрут») та обмеження приватності.
2. Спроектовано архітектуру сховища даних DWH на основі схеми «Зірка», що є оптимальною для швидкого виконання аналітичних запитів.
3. Розроблено стратегію ETL-процесів, яка вирішує критичну проблему гетерогенності даних через вбудовані алгоритми нормалізації ваги та конвертації валют.
4. Визначено політики управління даними, які гарантують відповідність системи вимогам GDPR (через псевдонімізацію) та забезпечують високу якість даних для прийняття рішень.

РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Теоретико-множинна формалізація та постановка задачі управління логістичними процесами

Розробка інформаційно-аналітичної системи (ІАС) для підтримки прийняття управлінських рішень у сфері міжнародної логістики, вимагає створення надійного, науково обґрунтованого математичного фундаменту. Специфіка предметної області, окреслена в технічному завданні, характеризується високим рівнем ентропії вхідних даних, стохастичністю процесів транспортування, мультिवаріантністю маршрутів та необхідністю обробки значних масивів різномірної інформації (Big Data), що генерується користувачами в процесі електронної комерції. Відповідно, математичне забезпечення системи не може обмежуватися тривіальними арифметичними операціями агрегації, а повинно включати комплекс моделей та методів для вирішення задач кластеризації, прогнозування, оптимізації та багатокритеріального прийняття рішень в умовах невизначеності.

3.1.1. Системний аналіз та декомпозиція об'єкта управління

З точки зору системного аналізу, логістична компанія, що оперує на міжнародному ринку (США, Європа, Канада – Україна), розглядається як складна кібернетична система S , функціонування якої спрямоване на перетворення вхідного потоку замовлень у вихідний потік доставлених вантажів з мінімізацією ентропії та витрат ресурсів.

Формально систему S можна представити у вигляді впорядкованого кортежу множин:

$$S = (U, O, W, R, T, P, \Phi)$$

де:

- $U = \{u_1, u_2, \dots, u_n\}$ – множина користувачів (клієнтів), кожен з яких характеризується вектором ознак x_u (геолокація, історія транзакцій, платоспроможність, поведінкові патерни);

- $O = \{o_1, o_2, \dots, o_m\}$ – множина активних замовлень (посилок), що мають динамічні атрибути стану (статус, локація) та статичні параметри (вага, об'єм, категорія товару);
- $W = \{\omega_1, \omega_2, \dots, \omega_k\}$ – множина вузлів логістичної мережі (склади-хаби в Нью-Джерсі, Гамбурзі, сортувальний центр у Львові);
- $R = \{r_1, r_2, \dots, r_p\}$ – множина доступних ребер графа логістичної мережі (авіалінії, морські шляхи, автомобільні магістралі);
- $T = \{t_1, t_2, \dots, t_q\}$ – множина часових обмежень та параметрів (терміни доставки SLA, час митного оформлення);
- $P = \{p_1, p_2, \dots, p_z\}$ – множина фінансових показників (тарифи перевізників, митні збори, динамічні курси валют **USD/UAH, EUR/UAH**);
- Φ – множина керуючих впливів (зміна маршруту, консолідація вантажів, надання знижок).

Таблиця 3.1 – Характеристика основних множин системи та їх потужність

Позначення	Назва множини	Типові елементи	Оцінка потужності ($ A $)	Динаміка змін
U	Користувачі	ID клієнта, адреса, RFM-вектор	$10^5 - 10^6$	Висока (щоденна реєстрація)
O	Замовлення	Трек-номер, вага, вартість, статус	$10^4 - 10^5$ (активні)	Дуже висока (зміна статусів)
W	Склади (Вузли)	Хаб NJ, Хаб DE, Склад Львів	$10^1 - 10^2$	Низька (структурна сталість)
R	Маршрути	NJ \rightarrow Lviv (Air), DE \rightarrow NJ (Auto)	$10^2 - 10^3$	Середня (сезонні зміни)
P	Фінанси	Тарифна сітка, курси валют	10^2	Висока (щоденні курси)

Взаємодія між елементами цих множин описується функціями переходу станів. Наприклад, зміна статусу замовлення o_i описується функцією:

$$f_{status}: O * T \rightarrow Statuses,$$

де $Statuses = \{New, Warehouse, Transit, Customs, Delivered\}$

3.1.2. Математична постановка задачі багатокритеріальної оптимізації

Глобальна мета управління системою полягає у виборі такої стратегії $\Phi^* \subset \Phi$, яка забезпечує екстремум вектора цільових функцій. У логістиці міжнародної доставки ми стикаємося з класичним «трикутником обмежень», який трансформується у задачу багатокритеріальної оптимізації.

Виділимо ключові критерії ефективності:

1. **Час доставки (E_t):** мінімізація загального часу проходження вантажу від складу відправлення до «останньої милі».

$$E_t = \sum_{o_i \in O} (t_{delivery}(o_i) - t_{creation}(o_i)) \rightarrow min$$

2. **Операційні витрати (E_c):** мінімізація сукупних витрат на транспортування, зберігання та обробку.

$$E_c = \sum_{o_i \in O} \sum_{r_j \in Route(o_i)} Cost(r_j) \rightarrow min$$

3. **Надійність та якість (E_q):** максимізація частки успішно доставлених вчасно посилок (Service Level Agreement) та задоволеності клієнта.

$$E_q = \frac{|o_{ontime}|}{|o_{total}|} \rightarrow min$$

Оскільки ці критерії часто є суперечливими (наприклад, авіадоставка мінімізує E_t , але максимізує E_c), задача зводиться до пошуку

парето-оптимальних рішень або згортання критеріїв у єдиний функціонал за допомогою методу вагових коефіцієнтів:

$$F(\Phi) = \omega_1 \frac{E_t - E_t^{min}}{E_t^{max} - E_t^{min}} + \omega_2 \frac{E_c - E_c^{min}}{E_c^{max} - E_c^{min}} - \omega_3 E_q$$

Де ω_i вагові коефіцієнти, що визначаються стратегічним менеджментом компанії $\left(\sum \omega_i = 1\right)$, а складові нормовані для приведення до єдиної безрозмірної шкали.

Вирішення цієї задачі вимагає декомпозиції на підзадачі нижчого рівня, для кожної з яких обирається специфічний математичний апарат:

1. **Аналіз та сегментація даних:** для розуміння структури множини U (користувачі) та O (замовлення). Використовуються методи кластерного аналізу.
2. **Прогнозування навантаження:** для оцінки майбутніх станів множини O та планування ресурсів W . Використовуються методи аналізу часових рядів.
3. **Оптимізація логістичних потоків:** для вибору оптимальних підмножин R (маршрути). Використовуються методи теорії графів.
4. **Підтримка прийняття рішень:** для вибору конкретних керуючих впливів в умовах невизначеності.

Такий підхід відповідає вимогам до магістерських робіт, забезпечуючи теоретичний базис для практичної реалізації системи.

3.2. Методи попередньої обробки та нормалізації даних

Ефективність роботи алгоритмів машинного навчання та аналітики безпосередньо залежить від якості вхідних даних («Garbage In, Garbage Out»). У системі логістичної компанії дані надходять із гетерогенних джерел: CRM-системи Creatio, білінгових шлюзів, API партнерських служб доставки

(UPS, DHL, FedEx) та безпосередньо від користувачів через веб-інтерфейс. Це зумовлює наявність шумів, пропусків, аномалій та розбіжностей у одиницях виміру.

3.2.1. Математичні методи виявлення аномалій

Логістичні дані часто містять викиди, спричинені помилками ручного введення (наприклад, вага посилки 1000 кг замість 10 кг) або системними збоями. Для їх автоматичного виявлення та фільтрації в ІАС пропонується використовувати робастні статистичні методи, стійкі до відхилень розподілу від нормального закону.

Метод інтерквартильного розмаху. Нехай $X = \{x_1, x_2, \dots, x_n\}$ – вибірка значень певної числової ознаки (вага, вартість).

1. Обчислюються квартилі Q_1 (25-й перцентиль) та Q_3 (75-й перцентиль).
2. Знаходиться інтерквартильний розмах:

$$IQR = Q_3 - Q_1$$

3. Визначаються межі толерантного інтервалу:

$$[Q_1 - 1.5 \cdot IQR; Q_3 + 1.5 \cdot IQR]$$

Елемент x_i вважається аномальним, якщо $x_i \notin$. Цей метод є більш доцільним для логістичних даних порівняно з критерієм 3σ (Z – score), оскільки розподіли вартості замовлень та ваги часто мають «важкі хвости» і не є строго гауссівськими.

Алгоритм обробки аномалій. Якщо виявлено аномальне значення, система повинна прийняти рішення про його обробку. Пропонується гібридний підхід:

- Якщо значення виходить за фізично можливі межі (наприклад, вага < 0), воно маркується як помилкове і замінюється на *NULL* або потребує ручної верифікації оператором.
- Якщо значення є статистичним викидом, але фізично можливим (наприклад, дуже дорога покупка), воно зберігається, але позначається відміткою $is_{anomaly} = TRUE$ для виключення з навчальних вибірок моделей прогнозування.

3.2.2. Алгоритми нормалізації та стандартизації ознак

Для коректної роботи метричних алгоритмів (K-means, KNN), які базуються на обчисленні відстаней, необхідно привести ознаки до єдиного масштабу. У логістиці ознаки мають суттєво різні діапазони: вага (0.1 – 50 кг), вартість (10 – 2000 USD), відстань (500 – 10000 км). Без нормалізації ознака з найбільшим діапазоном (відстань) буде домінувати при розрахунку евклідової метрики, нівелюючи вплив інших факторів.

Min-Max нормалізація. Лінійне перетворення даних у діапазон:

$$x'_i = \frac{x_i - (X)}{(X) - (X)}$$

Цей метод зберігає розподіл вихідних даних і є корисним для візуалізації KPI у дашбордах Power BI, де важливо зберегти відносні пропорції.

Z-стандартизація (Standard Scaler). Перетворення даних таким чином, щоб середнє значення дорівнювало 0, а стандартне відхилення – 1:

$$z_i = \frac{x_i - \mu}{\sigma}, \text{ де}$$

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i, \quad \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2}$$

Цей метод є пріоритетним для використання перед кластеризацією (K-means) та в регресійних моделях, оскільки він робить алгоритми менш

чутливими до викидів і забезпечує швидшу збіжність градієнтних методів оптимізації.

3.2.3. Математичне забезпечення конвертації та уніфікації

Система повинна забезпечувати автоматичну уніфікацію даних, що надходять у різних системах вимірювання (Імперська в США та Метрична в ЄС/Україні) та валютах.

Конвертація фізичних величин. Функція перетворення ваги W з фунтів (lbs) у кілограми (kg):

$$f_{\omega}(v, region) = \{v * 0.45359237, \text{ if } region \in \{US, CA, UK\} \text{ } v, \text{ otherwise}$$

Аналогічні перетворення застосовуються для лінійних розмірів (дюйми \rightarrow сантиметри) для розрахунку об'ємної ваги ($DimWeight$).

Об'ємна вага розраховується за формулою IATA:

$$W_{vol} = \frac{L*W*H}{5000\left(\frac{cm}{kg}\right)}$$

Мультивалютна агрегація. Для фінансового аналізу необхідно привести всі транзакції до базової валюти (наприклад, USD або UAH). Оскільки курс валют $Rate(t)$ є функцією часу, для агрегації даних за період T (наприклад, місячний звіт про прибуток) не можна використовувати поточний курс.

Використовується інтегральна оцінка (середньозважений курс):

$$\bar{R}_{period} = \frac{\int_{t \in T} V(t) * Rate(t) dt}{\int_{t \in T} V(t) dt} \approx \frac{\sum_{i=1}^N v_i}{r_i \sum_{i=1}^N v_i}$$

де v_i – сума i -ї транзакції, r_i – курс на момент транзакції. Такий підхід дозволяє отримати фінансову звітність, яка точно відображає економічну реальність.

3.2.4. Забезпечення приватності даних

Враховуючи вимоги GDPR, система повинна обробляти дані користувачів без розкриття їх особистості на етапі аналітики. Для цього застосовується математична модель *k*-анонімності.

Нехай таблиця даних *D* містить квазі-ідентифікатори *QI* (наприклад, поштовий індекс, дата народження) та сенситивні дані *S* (вартість покупок). Таблиця задовольняє *k*-анонімності, якщо кожен запис у ній не відрізняється від принаймні *k* – 1 інших записів за значеннями *QI*.

Для досягнення цього використовуються оператори узагальнення (Generalization) та приховування (Suppression).

Приклад функції узагальнення для поштового індексу:

$Gen(zip, level) = \{12345, level = 0\} \{1234 *, level = 1\} \{123 **, level = 2\}$

Математична задача полягає у знаходженні мінімального рівня узагальнення, при якому виконується умова $|Group(QI)| \geq k$ для всіх груп, з мінімізацією втрати інформативності даних (Information Loss Metric). Це дозволяє проводити аналіз регіональних трендів, не порушуючи конфіденційності конкретних користувачів.

3.3. Математичні моделі кластеризації та сегментації користувачів

Сегментація клієнтської бази є критично важливою для персоналізації сервісу. Замість використання жорстких правил (Rule-based approach), пропонується використання методів навчання без учителя (Unsupervised Learning) для виявлення прихованих структур у даних.

3.3.1. Обґрунтування вибору методу K-means

Серед різноманіття методів кластеризації (ієрархічні, щільнісні, ймовірнісні) для задачі сегментації клієнтів логістичної компанії обрано алгоритм **K-means (к-середніх)**.

Таблиця 3.2 – Порівняльний аналіз алгоритмів кластеризації для задач логістики

Алгоритм	Складність	Переваги	Недоліки	Доцільність
K-means	$O(n \cdot K \cdot I \cdot d)$	Висока швидкість, простота інтерпретації центроїдів, лінійна складність	Чутливість до викидів, необхідність задавати K , сферична форма кластерів	Висока
DBSCAN	$O(n \log \log n)$	Знаходить кластери довільної форми, стійкий до шуму	Проблеми з різною щільністю даних, складно підібрати ϵ	Середня
Agglomerative	$O(n^3)$	Будує ієрархію, не потребує K	Дуже повільний на великих вибірках ($n > 10^4$)	Низька

Враховуючи, що кількість клієнтів може сягати сотень тисяч, лінійна складність **K-means** є вирішальним фактором. Крім того, центроїди кластерів у **K-means** є усередненими векторами ознак, що дозволяє легко інтерпретувати отримані сегменти (наприклад, «Клієнти з високим чеком і низькою частотою»).

3.3.2. Математична формалізація алгоритму K-means

Нехай $X = \{x_1, x_2, \dots, x_n\} \subset R^d$ – множина векторів ознак користувачів.

Мета алгоритму – розбити X на K множин (кластерів) $C = \{C_1, C_2, \dots, C_K\}$ так, щоб мінімізувати сумарне квадратичне відхилення точок від центрів їхніх кластерів (Within-Cluster Sum of Squares, WCSS).

Цільова функція (потенціал кластеризації):

$$J(C) = \sum_{j=1}^K \sum_{x_i \in C_j} \|x_i - \mu_j\|^2 \rightarrow \min$$

де μ_j – центроїд кластера C_j , що обчислюється як:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Ітеративна процедура (алгоритм Ллойда):

1. **Ініціалізація:** Обрати K початкових центроїдів $\mu_j^{(0)}$. Для покращення збіжності та уникнення локальних мінімумів використовується метод K -means++, де ймовірність вибору точки як початкового центроїда пропорційна квадрату відстані до вже обраних центроїдів.
2. **Крок призначення (E-step):** кожен об'єкт x_i відноситься до кластера з найближчим центроїдом:

$$C_j^{(t)} = \{x_i : \|x_i - \mu_j^{(t)}\| \leq \|x_i - \mu_l^{(t)}\|, \forall l \neq j\}$$

В якості метрики відстані використовується Евклідова відстань:

$$d(x, y) = \sqrt{\sum_{k=1}^d (x_k - y_k)^2}$$

3. **Крок оновлення (M-step):** перерахунок центроїдів для нових кластерів:

$$\mu_j^{(t+1)} = \frac{1}{|C_j^{(t)}|} \sum_{x \in C_j^{(t)}} x$$

4. **Перевірка зупинки:** алгоритм зупиняється, коли центроїди стабілізуються $\left(\left| \mu_j^{\{(t+1)\}} - \mu_j^{\{(t)\}} \right| < \varepsilon \right)$ або досягнуто ліміту ітерацій.

3.3.3. Вибір оптимальної кількості кластерів

Параметр K є гіперпараметром моделі. Для його визначення використовується Метод ліктя (Elbow Method). Будується графік залежності $J(K)$ від K . Зі збільшенням K значення J монотонно спадає. Оптимальним вважається таке K , при якому швидкість спадання різко зменшується (максимізується кривизна графіка).

Математично точку перегину можна знайти, аналізуючи другу різницю (аналог другої похідної для дискретної функції):

$$D_2(K) = |J(K + 1) + J(K - 1) - 2J(K)|$$

Максимум $D_2(K)$ вказує на найбільш ймовірну оптимальну кількість кластерів.

Для задач роздрібно́ї торгівлі та логістики типове значення K лежить у діапазоні 3...7.

3.3.4. Застосування RFM-аналізу як базису для кластеризації

Для підвищення семантичної значущості кластерів вхідний вектор ознак x_i формується на основі моделі **RFM** (Recency, Frequency, Monetary):

1. **Recency (R):** час (днів) з моменту останнього замовлення. Чим менше, тим краще.
2. **Frequency (F):** кількість замовлень за фіксований період (наприклад, рік).
3. **Monetary (M):** Сумарна вартість послуг доставки (LTV клієнта).

Вектор клієнта: $u_i = (R_i, F_i, M_i)$. Перед подачею на вхід K-means виконується логарифмування (для зменшення скошеності розподілів M та F) та стандартизація:

$$u'_i = Z - \text{score}\left(\ln(\vec{u}_i + 1)\right)$$

Отримані кластери (наприклад, «Lost Cheap Customers», «Loyal Big Spenders», «Newcomers») використовуються для налаштування маркетингових кампаній та пріоритезації обслуговування.

3.4. Моделі прогнозування логістичних потоків

Здатність передбачати майбутній попит є ключовою для оптимізації складських площ та бронювання вантажних ємностей у авіаперевізників. Задача прогнозування в ІАС поділяється на два класи: прогнозування часових рядів (обсяг замовлень) та регресійний аналіз (час доставки).

3.4.1. Прогнозування часових рядів методом Хольта-Вінтерса

Логістичні дані характеризуються наявністю тренду (зростання електронної комерції) та чіткою сезонністю (піки на Чорну п'ятницю, Різдво, сезонні спади влітку). Для таких процесів найбільш адекватною є модель **потрійного експоненційного згладжування (Holt-Winters)**.

Адитивна модель описується системою рекурентних рівнянь:

1. **Рівень** ($Level, L_t$): згладжене значення ряду, очищене від сезонності та тренду.

$$L_t = \alpha(y_t - S_{t-m}) + (1 - \alpha)(L_{t-1} + T_{t-1})$$

2. **Тренд** ($Trend, T_t$): оцінка швидкості зміни рівня.

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

3. **Сезонність** ($Seasonality, S_t$): оцінка періодичного відхилення.

$$S_t = \gamma(y_t - L_t) + (1 - \gamma)S_{t-m}$$

4. Прогноз на h кроків ($y_{\{t+h\}}$):

$$y_{t+h} = L_t + hT_t + S_{t-m+h^*},$$

де m – період сезонності (наприклад, 12 для місячних даних),

$$h^* = (h - 1)(\text{mod } m) + 1.$$

Параметри згладжування $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$ знаходяться шляхом мінімізації середньоквадратичної помилки прогнозу (RMSE) на валідаційній вибірці. Оптимізаційна задача:

$$(\hat{\alpha}, \hat{\beta}, \hat{\gamma}) = \arg \arg \sum_{t=1}^N (y_t - \hat{y}_t(\alpha, \beta, \gamma))^2$$

Для вирішення цієї задачі використовується метод L-BFGS-B або алгоритм Нелдера-Міда.

3.4.2. Прогнозування часу доставки за допомогою множинної регресії

Для точного інформування клієнта про очікувану дату доставки (ETA - Estimated Time of Arrival) використовується модель множинної лінійної регресії. Час доставки Y залежить від багатьох факторів x .

Модель регресії:

$$Y = \beta_0 + \beta_1 x_{dist} + \beta_2 x_{weight} + \beta_3 x_{customs} + \beta_4 x_{type} + \varepsilon$$

де:

- x_{dist} – геодезична відстань між пунктами (обчислюється за формулою Гаверсінуса);
- x_{weight} – вага посилки;
- $x_{customs}$ – індекс навантаження на митний термінал (динамічний коефіцієнт);

- x_{type} – категоріальна змінна (Dummy variable) типу транспорту (0 - море, 1 - авіа).

Вектор коефіцієнтів $\beta = (\beta_0, \dots, \beta_k)^T$ оцінюється методом найменших квадратів (МНК) у матричній формі:

$$\beta = (X^T X)^{-1} X^T Y$$

де X – матриця значень предикторів (Design Matrix), Y – вектор спостережуваних часів доставки.

Теорема Гаусса-Маркова стверджує, що оцінки МНК є незсуненими та мають найменшу дисперсію в класі всіх лінійних незсунених оцінок, за умови, що помилки є некорельовані та мають сталу дисперсію (гомоскедастичність). Для перевірки якості моделі використовується коефіцієнт детермінації R^2 та F -тест Фішера на значущість регресії в цілому.

3.5. Алгоритми маршрутизації та транспортної оптимізації

Логістична модель компанії включає як магістральні перевезення (між країнами), так і доставку «останньої милі» (кур'єрська доставка). Це породжує два класи оптимізаційних задач.

3.5.1. Пошук оптимального шляху в графі логістичної мережі

Логістична мережа моделюється як зважений орієнтований граф $G = (V, E)$, де вершини V – це склади та хаби, а ребра E – транспортні канали. Вага ребра w_{ij} може відображати вартість, час або комбінований критерій.

Задача знаходження маршруту для однієї посилки зводиться до пошуку найкоротшого шляху. Використовується алгоритм Дейкстри.

Для прискорення пошуку в масштабних мережах доцільно використовувати *алгоритм A (A-star)*, який використовує евристичну функцію $h(v)$ – оцінку відстані від поточної вершини v до цільової.

$$f(v) = g(v) + h(v)$$

де $g(v)$ – реальна вартість шляху від старту до v . Як евристику $h(v)$ для логістики зручно використовувати евклідову відстань між координатами складів, поділену на максимальну швидкість транспорту. Це гарантує допустимість (admissibility) евристики і знаходження оптимального шляху.

3.5.2. Задача маршрутизації транспорту (VRP) для «останньої милі»

Кур'єрська доставка моделюється як класична задача VRP (Vehicle Routing Problem) або її підвид CVRP (Capacitated VRP), де транспортні засоби мають обмежену вантажопідйомність Q .

Математична модель (Цілочисельне лінійне програмування - ILP):

Мінімізувати:

$$\sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K c_{ij} x_{ij}^k$$

При обмеженнях:

- $\sum_{i=0}^N x_{ij}^k = 1$ (кожен клієнт j відвідується рівно один раз);
- $\sum_{i=1}^N d_i \sum_{j=0}^N x_{ij}^k \leq Q$ (сума вантажів d_i на маршруті k не перевищує ємність Q);
- зв'язність маршрутів та повернення в депо.

Оскільки VRP є NP-складною задачею (узагальнення задачі комівояжера), точні методи (Branch and Bound) працюють лише для малого N ($N < 100$). Для реальних умов ($N \approx 1000$) використовуються евристичні методи.

Алгоритм економії Кларка-Райта:

1. Початковий стан: кожен клієнт обслуговується окремим маршрутом «Депо → Клієнт → Депо».
2. Розрахунок «економії» s_{ij} від об'єднання двох маршрутів (відвідування клієнта i та j однією машиною):

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

де c_{i0} – відстань від клієнта i до депо.

3. Сортування пар (i, j) за спаданням економії s_{ij} .
4. Ітеративне об'єднання маршрутів, якщо це не порушує обмежень на ємність Q .

Цей алгоритм має складність $O(N^2 \log N)$ і забезпечує субоптимальне рішення, яке є достатнім для операційного планування щоденних доставок.

3.6. Підтримка прийняття рішень методами багатокритеріального аналізу

При виборі стратегічних партнерів (наприклад, вибір нового авіаперевізника для напрямку США-Польща) менеджмент стикається з суперечливими критеріями. Для математичної підтримки таких рішень в ІАС імплементовано **Метод аналізу ієрархій (Analytic Hierarchy Process - АНР)** Т. Сааті.

3.6.1. Алгоритм АНР

1. **Декомпозиція проблеми:** Мета (Вибір перевізника) → Критерії (Ціна, Швидкість, Надійність) → Альтернативи.
2. **Побудова матриць попарних порівнянь:** Експерти заповнюють матрицю A розмірності $n \times n$, де елемент a_{ij} відображає перевагу

критерію i над j за шкалою 1-9. Матриця A є зворотно-симетричною

$$\left(a_{\{ji\}} = \frac{1}{a_{\{ij\}}} \right).$$

Розрахунок ваг: Ваги критеріїв ω знаходяться як власний вектор матриці A , що відповідає найбільшому власному числу $\lambda_{\{max\}}$:

$$A\omega = \lambda_{max} \omega$$

Для наближеного розрахунку використовується метод середнього геометричного рядків з наступною нормалізацією.

3. Перевірка узгодженості: Розраховується індекс узгодженості CI та відношення узгодженості CR :

$$CI = \frac{\lambda_{max} - n}{n-1}, \quad CR = \frac{CI}{RI}$$

де RI – випадковий індекс (табличне значення). Якщо $CR < 0.1$, матриця експертних оцінок вважається узгодженою.

4. **Синтез:** Глобальний пріоритет альтернативи розраховується як згортка ваг критеріїв та локальних пріоритетів альтернатив.

Використання АНР дозволяє перетворити суб'єктивні думки експертів у строгі числові оцінки, мінімізуючи ризик помилкового управлінського рішення.

Висновки до розділу 3

У даному розділі проведено комплексне обґрунтування та розробку математичного забезпечення для інформаційно-аналітичної системи логістичної компанії.

1. Запропоновано теоретико-множинну модель логістичної системи, яка дозволяє формалізувати об'єкт управління та визначити простір станів.

2. Розроблено алгоритми попередньої обробки даних (**IQR, Z-scaling**) та забезпечення k -анонімності, що гарантує валідність вхідної інформації та відповідність вимогам GDPR.
3. Обґрунтовано використання алгоритму **K-means** у поєднанні з RFM-аналізом для сегментації клієнтської бази, що дозволяє реалізувати стратегію диференційованого маркетингу.
4. Для прогнозування попиту адаптовано модель **Хольта-Вінтерса**, яка враховує сезонну специфіку логістичного бізнесу, а для оцінки термінів доставки – модель множинної регресії.
5. Вирішено задачу маршрутизації як на рівні магістралей (алгоритм **A**), так і на рівні «останньої милі» (евристика **Кларка-Райта** для VRP), що дозволяє оптимізувати транспортні витрати.
6. Для стратегічного планування впроваджено метод **АНР**, який забезпечує математичну підтримку прийняття рішень в умовах багатокритеріальності.

Запропонований математичний апарат є достатнім, несуперечливим і створює надійну базу для програмної реалізації системи, опис якої наведено у наступних розділах.

РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

У цьому розділі наводиться детальний опис процесу програмної реалізації інформаційно-аналітичної системи (ІАС) підтримки управлінських рішень у сфері логістики. Розділ охоплює обґрунтування вибору технологічного стеку, проектування архітектури програмного забезпечення, опис реалізації ключових функціональних модулів та алгоритмів, розробку прикладного програмного інтерфейсу (API), стратегії тестування та забезпечення якості, а також питання відповідності міжнародним стандартам та нормативним вимогам. Особлива увага приділяється специфіці предметної області – міжнародним логістичним перевезенням, що накладає унікальні вимоги до продуктивності, надійності та інтеграційних можливостей системи.

4.1. Обґрунтування вибору технологічного стеку та засобів розробки

Вибір інструментальних засобів для реалізації ІАС є критично важливим етапом, що визначає подальшу життєздатність, масштабованість та вартість супроводу програмного продукту. Враховуючи складність логістичних процесів компанії, яка оперує великими обсягами даних про міжнародні відправлення, технологічний стек повинен забезпечувати високу продуктивність транзакційної обробки, гнучкість аналітичних розрахунків та надійність зберігання даних.

4.1.1. Аналіз вимог до програмної платформи

Перед вибором конкретних технологій було проведено аналіз нефункціональних вимог до системи, що впливають з технічного завдання:

- 1. Високе навантаження (High Load):** Система повинна обробляти тисячі запитів на секунду в періоди сезонних піків (Black Friday, передсвяткові періоди).
- 2. Обробка різномірних даних:** Необхідність роботи як зі структурованими даними (профілі користувачів, транзакції), так і з напівструктурованими (логи подій, трекінг).

3. **Математичне моделювання:** Підтримка складних обчислень для реалізації алгоритмів кластеризації (K-means) та прогнозування (Holt-Winters).
4. **Мультивалютність та мультисистемність:** Вбудована підтримка конвертації валют та одиниць виміру (lb/kg) на рівні типів даних.
5. **Інтеграційна гнучкість:** Здатність взаємодіяти з існуючими CRM-системами (Creatio) та зовнішніми API логістичних партнерів.

4.1.2. Вибір мов програмування: Поліглотичний підхід

Для досягнення оптимального балансу між продуктивністю та швидкістю розробки було прийнято рішення використовувати поліглотичний підхід (Polyglot Programming), який передбачає використання різних мов програмування для різних компонентів системи.

Бекенд (Backend Core): Java В якості основної мови для розробки серверної частини обрано **Java** (версія 17 LTS). Цей вибір зумовлений наступними факторами:

- **Екосистема Enterprise:** Java є стандартом де-факто для розробки високонавантажених корпоративних систем. Наявність потужних фреймворків, таких як Spring Boot, дозволяє суттєво прискорити розробку мікросервісів.
- **Суворі типізація та стабільність:** Статична типізація Java дозволяє виявляти значну частину помилок на етапі компіляції, що є критичним для фінансових модулів системи (білінг, розрахунок вартості доставки).
- **Багатопотоковість:** Ефективна модель роботи з пам'яттю та потоками дозволяє обробляти велику кількість одночасних запитів від користувачів та IoT-пристроїв на складах.

Аналітичне ядро (Data Science Module): Python Для реалізації математичних алгоритмів та машинного навчання обрано мову **Python**. Хоча Java є потужною

мовою, Python має незаперечну перевагу в сфері Data Science завдяки багатому набору бібліотек:

- **Pandas & NumPy:** Для ефективної маніпуляції матрицями даних та часовими рядами.
- **Scikit-learn:** Для реалізації алгоритмів кластеризації (K-means).
- **Statsmodels:** Для побудови прогнозних моделей часових рядів (Holt-Winters). Такий розподіл дозволяє використовувати сильні сторони кожної мови: Java забезпечує стабільність та швидкість роботи API, а Python надає гнучкий інструментарій для поглибленої аналітики.

Фронтенд (Frontend): JavaScript/TypeScript та Vue.js Для клієнтської частини (особистий кабінет користувача, панель адміністратора) обрано фреймворк **Vue.js**.

- **Реактивність:** Vue.js дозволяє створювати інтерфейси, що миттєво реагують на дії користувача без перезавантаження сторінки (Single Page Application - SPA), що покращує користувацький досвід (UX).
- **Компонентний підхід:** Дозволяє перевикористовувати елементи інтерфейсу (наприклад, віджет трекінгу посилки) в різних частинах застосунку.
- **Легкість інтеграції:** Vue.js легко інтегрується з існуючими веб-рішеннями, що важливо при поступовій міграції зі старих PHP-інтерфейсів.

4.1.3. Вибір систем управління базами даних (СУБД)

Архітектура даних базується на принципі Polyglot Persistence –використання найбільш підходящої технології зберігання для кожного типу даних.

Реляційна СУБД: PostgreSQL в якості основного сховища даних обрано **PostgreSQL**. Це об'єктно-реляційна СУБД з відкритим кодом, яка забезпечує відповідність принципам ACID (Atomicity, Consistency, Isolation, Durability).

- **Чому не MySQL?** PostgreSQL краще підтримує складні аналітичні запити, має вбудовану підтримку JSONB (що дозволяє зберігати неструктуровані дані про атрибути товарів) та геопросторові типи даних (PostGIS), які можуть знадобитися для розширених функцій маршрутизації.
- **Призначення:** Зберігання профілів користувачів, замовлень, фінансових транзакцій, довідників.

Key-Value сховище: Redis Для забезпечення високої швидкодії системи використовується **Redis** як in-memoгу сховище даних.

- **Кешування:** Зберігання результатів важких запитів (наприклад, розрахованих тарифів для різних напрямків), сесій користувачів та токенів авторизації.
- **Черги повідомлень:** Може використовуватися як брокер повідомлень для простих асинхронних задач.

4.1.4. Інструментарій бізнес-аналітики

Для візуалізації даних та надання звітності керівництву компанії обрано платформу **Microsoft Power BI**.

- **Інтеграція:** Power BI має нативні конектори до PostgreSQL та інших джерел даних, що дозволяє налаштувати автоматичне оновлення звітів.
- **Інтерактивність:** Можливість створення інтерактивних дашбордів (Drill-down), що дозволяють аналізувати дані від загальних показників (KPI) до конкретних відправлень.
- **Доступність:** Звіти можуть бути опубліковані в веб-середовищі та доступні менеджерам через браузер або мобільний застосунок.

4.2. Проектування архітектури програмного забезпечення

Архітектура системи визначає, як програмні компоненти організовані та як вони взаємодіють між собою. Для розроблюваної ІАС було обрано мікросервісну архітектуру.

4.2.1. Концепція та обґрунтування мікросервісної архітектури

Традиційна монолітна архітектура, де всі модулі (користувачі, замовлення, аналітика) знаходяться в одному виконуваному файлі, має суттєві недоліки для масштабних логістичних систем. Будь-яка зміна в одному модулі вимагає перезбирання та розгортання всього додатку, а збій в одному компоненті може призвести до відмови всієї системи.

Мікросервісна архітектура передбачає розбиття системи на набір невеликих, слабозв'язаних сервісів, кожен з яких виконує свою бізнес-функцію і може розроблятися, розгортатися та масштабуватися незалежно.

Переваги для логістичної ІАС:

1. **Незалежне масштабування:** Сервіс трекінгу, який отримує найбільше навантаження від користувачів, може бути запущений у 10 екземплярах, тоді як сервіс адміністративних звітів – в одному.
2. **Ізоляція відмов:** Якщо сервіс аналітики вийде з ладу через помилку в розрахунках, це не вплине на можливість користувачів створювати нові замовлення через сервіс замовлень.
3. **Технологічна свобода:** Можливість використовувати Java для бекенду і Python для аналітики в межах однієї системи.

4.2.2. Компонентна декомпозиція системи

Система складається з наступних ключових мікросервісів:

1. **Auth Service (Сервіс авторизації та аутентифікації):**
 - Відповідає за реєстрацію нових користувачів, вхід в систему, генерацію та валідацію JWT-токенів.
 - Керує ролями та правами доступу (RBAC - Role-Based Access Control).
2. **Order Service (Сервіс управління замовленнями):**
 - Центральний компонент, що керує життєвим циклом відправлення (створення, редагування, скасування).

- Взаємодіє з базою даних для збереження деталей посилок (вага, вартість, вміст).
- Реалізує перевірку даних на відповідність бізнес-правилам (наприклад, ліміти митної вартості).

3. **Tracking Service (Сервіс відстеження):**

- Агрегує інформацію про переміщення посилок від різних перевізників та внутрішніх систем сканування на складах.
- Надає уніфікований статус для користувача.

4. **Analytics Service (Сервіс аналітики та прогнозування):**

- Реалізований на Python.
- Виконує важкі обчислювальні задачі: кластеризацію клієнтів (RFM-аналіз + K-means) та прогнозування обсягів доставки (Holt-Winters).
- Працює переважно в асинхронному режимі, обробляючи історичні дані.

5. **Notification Service (Сервіс сповіщень):**

- Відповідає за відправку email, SMS та push-повідомлень користувачам про зміну статусу їхніх відправлень.

6. **Reference Service (Сервіс довідників):**

- Зберігає відносно статичні дані: список країн, митні коди, тарифи перевізників, адреси складів-партнерів.

4.2.3. Міжсервісна взаємодія та API Gateway

Для організації взаємодії між мікросервісами та зовнішнім світом використовується **API Gateway** (на базі Spring Cloud Gateway). Він виступає єдиною точкою входу для всіх клієнтських запитів.

Функції API Gateway:

- **Маршрутизація:** Перенаправлення запиту до відповідного мікросервісу (наприклад, /api/orders → Order Service).

- **Балансування навантаження:** Рівномірний розподіл запитів між екземплярами сервісів.
- **Наскрізна безпека:** Перевірка токена авторизації перед передачею запиту далі.

Взаємодія між самими мікросервісами реалізована двома способами:

1. **Синхронна (REST/Feign Client):** Для операцій, що вимагають негайної відповіді (наприклад, Order Service запитує у Auth Service перевірку прав користувача).
2. **Асинхронна (Message Broker - RabbitMQ):** Для подій, які не вимагають миттєвої реакції. Наприклад, коли замовлення отримує статус "Доставлено", Order Service публікує подію OrderDelivered у чергу RabbitMQ. Цю подію "слухають" Notification Service (щоб відправити лист) та Analytics Service (щоб оновити статистику). Використання черг підвищує відмовостійкість системи.

4.3. Реалізація рівня даних та механізмів збереження

Ефективна робота з даними є фундаментом інформаційно-аналітичної системи.

4.3.1. Проектування схеми бази даних

В мікросервісній архітектурі використовується патерн "Database per Service" (або "Schema per Service"), що означає, що кожен сервіс володіє своїми власними даними і прямий доступ до таблиць іншого сервісу заборонений.

Таблиця 4.1. Ключові сутності БД PostgreSQL

Таблиця	Опис полів	Примітка
users	user_id (UUID), email, password_hash, registration_date, cluster_id	Зберігається в схемі Auth Service. cluster_id оновлюється сервісом аналітики.
shipments	shipment_id (Long), user_id, tracking_number, weight, value, currency, status	Основна таблиця Order Service.
shipment_items	item_id, shipment_id, description, quantity, price	Деталізація вмісту посилки.
tracking_history	event_id, shipment_id, location, timestamp, status_code	Історія переміщень.

Для забезпечення цілісності даних на рівні бази використовуються зовнішні ключі (Foreign Keys) в межах однієї схеми та транзакції. Розподілені транзакції (Saga pattern) використовуються для процесів, що охоплюють кілька сервісів.

4.3.2. ETL-процеси та підготовка даних для аналітики

Для роботи BI-інструментів та модулів машинного навчання операційні дані (OLTP) повинні бути трансформовані та завантажені в аналітичне сховище (OLAP або Data Warehouse). Реалізовано ETL-процес (Extract, Transform, Load):

1. **Extract (Вилучення):** Регулярне (або потокове через CDC) вивантаження нових та змінених записів з операційних баз даних.
2. **Transform (Трансформація):**
 - **Очищення:** Видалення дублікатів, обробка пропущених значень.
 - **Валідація:** Перевірка логічної узгодженості (Consistency), наприклад, дата доставки не може бути раніше дати відправлення. Записи, що не пройшли перевірку, переміщуються в таблицю карантину (Quarantine Table) для ручного розбору, не потрапляючи у фінальні звіти, щоб не спотворювати статистику.

- **Стандартизація:** Приведення всіх грошових сум до єдиної валюти (EUR або USD) за історичним курсом, конвертація ваги в кілограми. Це критично для мультивалютності системи.
 - **Анонімізація:** Видалення персональних даних (імен, адрес) для відповідності вимогам GDPR перед передачею аналітикам.
3. **Load (Завантаження):** Збереження підготовлених даних у спеціалізовану схему для Power BI та Python-сервісу.

4.4. Програмна реалізація алгоритмів та бізнес-логіки

У цьому підрозділі розкриваються деталі імплементації математичних моделей, описаних у Розділі 3.

4.4.1. Реалізація кластеризації методом K-means

Сегментація клієнтської бази реалізована у мікросервісі Analytics Service мовою Python.

Алгоритм обробки:

1. **Збір даних:** SQL-запит агрегує історію замовлень для кожного користувача, формуючи RFM-вектор:
 - Recency (давність останнього замовлення в днях).
 - Frequency (кількість замовлень за період).
 - Monetary (загальна сума витрат).
2. **Попередня обробка:**
 - Оскільки RFM-показники мають різні одиниці виміру та масштаби, застосовується нормалізація даних за допомогою StandardScaler з бібліотеки scikit-learn. Це важливо, оскільки K-means базується на евклідовій відстані і чутливий до масштабу змінних.

3. Виконання кластеризації:

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4, init='k-means++', random_state=42)
user_clusters = kmeans.fit_predict(normalized_rfm_data)
```

Кількість кластерів (K=4) була визначена експериментально методом "ліктя" (Elbow Method) на етапі дослідження.

4. **Інтерпретація та збереження:** Центроїди отриманих кластерів аналізуються для присвоєння бізнес-міток (наприклад, "Loyal", "New", "Lost", "VIP"). Результати (ID кластера для кожного користувача) записуються назад у базу даних користувачів, що дозволяє маркетологам формувати таргетовані пропозиції через CRM.

4.4.2. Реалізація прогнозування методом Хольта-Вінтерса

Прогнозування обсягів вантажопотоку необхідне для планування складських площ та бронювання авіафрахту. Використовується модель потрійного експоненційного згладжування, яка враховує рівень, тренд та сезонність.

Реалізація: Використовується бібліотека statsmodels. Часовий ряд формується шляхом агрегації кількості/ваги посилок по днях або тижнях. Система автоматично тренує модель на історичних даних (наприклад, за останні 2 роки) та генерує прогноз на наступний період (місяць/квартал). Особливістю реалізації є обробка "викидів" (outliers) –аномальних значень, спричинених форс-мажорними обставинами, які згладжуються перед подачею на вхід моделі, щоб не погіршувати якість прогнозу.

4.4.3. Алгоритми маршрутизації

Для розрахунку оптимального маршруту переміщення посилки між хабами (наприклад, Нью-Джерсі → Львів → Київ) логістична мережа моделюється як зважений орієнтований граф.

- **Вершини:** Склади, сортувальні центри, митні термінали.

- **Ребра:** Транспортні зв'язки (авіа, авто).
- **Вага ребра:** Функція від вартості (C) та часу (T): $W = \alpha * C + \beta * T$, де α та β –вагові коефіцієнти, що задаються пріоритетом клієнта (дешево чи швидко).

Програмно реалізовано алгоритм Дейкстри для знаходження найкоротшого шляху. Для оптимізації пошуку в великих графах використовується алгоритм A* (A-star) з евристичною оцінкою відстані, що дозволяє швидше відсікати безперспективні напрямки пошуку.

4.5. Розробка прикладного програмного інтерфейсу

API (Application Programming Interface) є основним способом взаємодії зовнішніх клієнтів (веб-сайт, мобільні додатки) з системою.

4.5.1. Принципи побудови RESTful API

API розроблено відповідно до принципів REST (Representational State Transfer).

- **Ресурсо-орієнтованість:** URL-адреси ідентифікують ресурси (іменники), а HTTP-методи –дії над ними (наприклад, GET /shipments –отримати список, POST /shipments–створити нове).
- **Безстану:** Сервер не зберігає стан клієнтської сесії між запитами; вся необхідна інформація для авторизації передається в заголовках кожного запиту.
- **Формат обміну:** JSON (JavaScript Object Notation).

4.5.2. Специфікація та документування

Для опису контрактів API використовується специфікація **OpenAPI 3.0** (раніше Swagger). Це дозволяє автоматично генерувати інтерактивну документацію, яку використовують фронтенд-розробники та QA-інженери.

Приклад опису ендпоінту створення замовлення (YAML):

```
/shipments:
  post:
    summary: Створення нового відправлення
    security:
      - bearerAuth:
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              origin_country:
                type: string
                example: "US"
              weight:
                type: number
                format: float
            items:
              type: array
              items:
                $ref: '#/components/schemas/ShipmentItem'
    responses:
      '201':
        description: Відправлення успішно створено
      '400':
        description: Помилка валідації даних
```

4.5.3. Безпека та авторизація

Безпека API базується на протоколі **OAuth 2.0** та використанні **JWT (JSON Web Tokens)**.

1. Клієнт відправляє логін/пароль на `/auth/login`.
2. При успішній перевірці сервер повертає `access_token` (короткоживучий) та `refresh_token` (довгоживучий).
3. Клієнт додає `access_token` у заголовок «Authorization: Bearer...» кожного наступного запиту.

4. API Gateway перевіряє цифровий підпис токена та термін його дії перед тим, як пропустити запит до мікросервісів.

4.6. Розробка клієнтської частини

Інтерфейс користувача розроблено як Single Page Application (SPA) на базі Vue.js.

4.6.1. Архітектура SPA

Застосунок побудовано на основі компонентної архітектури. Весь інтерфейс розділено на незалежні компоненти (ShipmentList, ShipmentDetails, Calculator, Profile), які можуть мати власну логіку та стилі. Для навігації використовується Vue Router, що емулює переходи між сторінками без фізичного перезавантаження документа.

4.6.2. Взаємодія з бекендом та управління станом

Для управління глобальним станом додатку (дані користувача, список замовлень, налаштування інтерфейсу) використовується бібліотека Pinia (або Vuex). Це дозволяє уникнути проблеми "prop drilling" (передачі даних через багато рівнів вкладеності компонентів) і забезпечує єдине джерело істини для даних на фронтенді. Взаємодія з REST API здійснюється за допомогою бібліотеки Axios, налаштованої на автоматичне додавання токенів авторизації та обробку помилок мережі.

4.6.3. Адаптивність

Інтерфейс розроблено з використанням принципів Mobile First. Верстка адаптується під різні розміри екранів (десктоп, планшет, мобільний телефон) за допомогою CSS Grid та Flexbox, що є критичним, оскільки значна частина користувачів перевіряє статус посилок зі смартфонів.

4.7. Тестування та забезпечення якості

Забезпечення якості (Quality Assurance) інтегровано в усі етапи розробки згідно з моделлю "піраміди тестування".

4.7.1. Модульне тестування

Це основа піраміди. Тести пишуться розробниками для перевірки роботи окремих класів та функцій в ізоляції.

- **Інструменти:** JUnit 5 та Mockito для Java, pytest для Python, Jest для Vue.js.
- **Покриття:** Цільовий показник покриття коду тестами (Code Coverage) становить не менше 80%.
- **Приклад:** Тест перевіряє, чи метод розрахунку вартості викидає виключення, якщо вага посилки є від'ємною.

4.7.2. Інтеграційне тестування

Перевірка взаємодії між компонентами (наприклад, сервіс + база даних).

- **Інструменти:** Spring Boot Test, Testcontainers.
- **Підхід:** Використання Docker-контейнерів для підняття реального екземпляра PostgreSQL на час виконання тестів, що гарантує ідентичність поведінки з продакшен-середовищем.

4.7.3. Навантажувальне тестування

Для перевірки стабільності системи під високим навантаженням проведено серію тестів.

- **Інструмент:** Apache JMeter.
- **Сценарій:** Емуляція одночасної роботи 10 000 користувачів, які виконують типові дії (вхід, перегляд списку, трекінг).
- **Результати:** Система продемонструвала стабільну роботу (успішність запитів 99.9%) при навантаженні до 2500

запитів/сек. Середній час відповіді (Latency) склав 120 мс, що задовольняє вимоги ТЗ.

4.7.4. Статичний аналіз коду

Для автоматизованого контролю якості коду впроваджено платформу SonarQube. Вона інтегрована в CI/CD пайплайн і аналізує код при кожному коміті на предмет:

- Потенційних багів та вразливостей.
- "Запахів коду" (Code Smells) –заплутаної логіки, дублювання коду.
- Цикломатичної складності. Встановлено Quality Gate: зміни не можуть бути прийняті, якщо покриття тестами впало нижче 80% або знайдено критичні вразливості.

4.8. Стандартизація та відповідність нормативним вимогам

Розробка системи велася з урахуванням міжнародних та галузевих стандартів.

4.8.1. Стандарт якості

Система оцінювалася за характеристиками якості стандарту ISO/IEC 25010:

- Функціональна придатність: Повна відповідність вимогам ТЗ (трекінг, аналітика, кабінет).
- Ефективність: Використання кешування (Redis) та асинхронної обробки забезпечило високу швидкодію.
- Надійність: Використання патерну Circuit Breaker (Resilience4j) запобігає каскадним збоям –якщо один сервіс падає, інші продовжують працювати.
- Сумісність: Стандартизовані API дозволяють легку інтеграцію.

4.8.2. Забезпечення приватності

Оскільки компанія працює на ринку ЄС, система суворо дотримується регламенту GDPR (General Data Protection Regulation). Технічні заходи:

- Шифрування: Персональні дані (PII) шифруються в базі даних (Data at Rest) та при передачі по мережі (Data in Transit, TLS 1.3).
- Мінімізація даних: Аналітичні сервіси працюють лише з знеособленими ID, не маючи доступу до реальних імен клієнтів.
- Право на забуття: Реалізовано функціонал повного видалення або анонімізації даних користувача за його запитом. При цьому фінансові транзакції зберігаються згідно з вимогами податкового законодавства, але відв'язуються від профілю особи.

4.9. Розгортання та експлуатація

Для забезпечення безперервної інтеграції та доставки (CI/CD) налаштовано автоматизовані конвеєри (Pipelines) на базі GitLab CI (або GitHub Actions).

4.9.1. Контейнеризація та оркестрація

Всі мікросервіси запаковані у Docker-контейнери. Це вирішує проблему "it works on my machine", забезпечуючи ідентичне середовище виконання на комп'ютері розробника, тестовому стенді та продакшн-сервері. Для управління запуском, масштабуванням та моніторингом контейнерів використовується Kubernetes (K8s). Це дозволяє реалізувати:

- Self-healing: Автоматичний перезапуск контейнерів, що впали.
- Auto-scaling: Автоматичне додавання нових екземплярів сервісів при зростанні навантаження CPU/RAM.
- Zero-downtime deployment: Оновлення сервісів без зупинки системи (Rolling Update).

4.9.2. Моніторинг та логування

Для оперативного виявлення проблем розгорнуто систему моніторингу:

- Prometheus: Збір метрик продуктивності (RPS, Latency, Error Rate) з мікросервісів.
- Grafana: Візуалізація метрик на дашбордах.
- ELK Stack (Elasticsearch, Logstash, Kibana): Централізований збір логів. Оскільки логи розкидані по різних контейнерах, ELK збирає їх в єдине сховище, дозволяючи пошук та аналіз помилок у розподіленій системі.

Висновки до розділу 4

У даному розділі було детально розглянуто програмну реалізацію інформаційно-аналітичної системи для логістичної компанії.

1. Обґрунтовано та реалізовано перехід до мікросервісної архітектури, що забезпечує необхідну гнучкість, масштабованість та відмовостійкість для умов міжнародної логістики.
2. Сформовано ефективний технологічний стек (Java Spring Boot, Python, Vue.js, PostgreSQL, Redis, RabbitMQ), який поєднує надійність корпоративних платформ з потужністю сучасних засобів аналізу даних.
3. Розроблено та задокументовано REST API, що дозволяє інтегрувати систему в існуючий IT-ландшафт компанії та взаємодіяти з мобільними додатками.
4. Імплементовано інтелектуальні алгоритми кластеризації (K-means) та прогнозування (Holt-Winters), що надають системі аналітичну цінність для прийняття управлінських рішень.
5. Забезпечено високий рівень якості та безпеки завдяки комплексній стратегії тестування, дотриманню стандартів ISO 25010 та вимог GDPR.

6. Впроваджено практики DevOps, що автоматизують процеси розгортання та моніторингу, знижуючи операційні витрати та ризики людських помилок.

Розроблений прототип та архітектурні рішення створюють надійну основу для подальшого розвитку системи та її впровадження в промислову експлуатацію.

РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

В умовах глобальної цифровізації економічних процесів та стрімкої еволюції ринку електронної комерції (e-commerce), традиційні підходи до управління логістичними ланцюгами постачання зазнають фундаментальних змін. Статичні моделі, орієнтовані виключно на фізичне переміщення вантажів, поступаються місцем динамічним екосистемам, де ключовим активом стає інформація. Розроблена в рамках даної магістерської кваліфікаційної роботи інформаційно-аналітична система (ІАС) підтримки управлінських рішень у сфері логістики на основі даних користувачів являє собою не лише науково-технічну розробку, а й потенційну основу для створення високотехнологічного стартап-проєкту. Цей розділ присвячено комплексному аналізу перспектив комерціалізації розробленого рішення, оцінці його ринкової життєздатності та формуванню стратегії виведення на ринок B2B-сервісів.

5.1. Опис ідеї стартап-проєкту

Першим етапом розробки стартап-проєкту є чітка ідентифікація сутності ідеї, її актуальності та ціннісної пропозиції для потенційного споживача. В основі концепції лежить створення спеціалізованого програмного продукту, що вирішує проблему інформаційної ентропії в міжнародних логістичних операціях.

5.1.1. Сутність ідеї та опис проблемної області

Сучасні логістичні оператори, такі як об'єкт дослідження компанія, оперують в умовах надзвичайної складності даних. Інформаційний ландшафт типової компанії сектору cross-border delivery характеризується високою фрагментацією: дані про клієнтів знаходяться в CRM-системах (наприклад, Creatio), інформація про рух вантажів – у WMS (Warehouse Management System) та TMS (Transportation Management System), а фінансові показники – у білінгових платформах. Ця розрізненість призводить до так званих

«інформаційних розривів», що унеможлиблює оперативне прийняття обґрунтованих управлінських рішень.

Ідея стартап-проєкту полягає у розробці та впровадженні інтелектуальної аналітичної надбудови (Intelligence Layer), яка інтегрується з існуючими гетерогенними системами підприємства, консолідує дані в єдиному сховищі (Data Warehouse) та, застосовуючи алгоритми машинного навчання, надає топ-менеджменту готові сценарії рішень. Продукт позиціонується не як заміна існуючим WMS або CRM, а як інструмент вищого рівня ієрархії – система підтримки прийняття рішень (DSS – Decision Support System).

Критичні проблеми, на вирішення яких спрямований стартап, включають:

1. **Непрозорість Unit Economics:** складність точного розрахунку собівартості доставки одного кілограма вантажу на конкретному маршруті через коливання курсів валют та різницю в тарифах підрядників.
2. **Реактивне управління:** більшість компаній реагують на затримки або перевантаження складів вже по факту їх виникнення, тоді як бізнес потребує проактивного прогнозування.
3. **Втрата клієнтів (Churn Rate):** відсутність персоналізованого підходу та прозорого трекінгу знижує лояльність користувачів, що є критичним для B2C сегменту.

5.1.2. Ціннісна пропозиція

Ціннісна пропозиція стартапу формулюється через призму конкретних вигод для стейкхолдерів логістичного бізнесу. Вона базується на перетворенні «сирих» даних у стратегічний актив.

Таблиця 5.1.

Компонент цінності	Опис вигоди для клієнта (логістичного оператора)
Економічна ефективність	Зниження операційних витрат на 10–15% шляхом оптимізації маршрутів та зменшення кількості помилкових відправлень завдяки автоматичній валідації адрес та нормалізації даних.
Стратегічне бачення	Забезпечення «погляду зверху» (Helicopter View) на бізнес-процеси через інтерактивні дашборди Power BI, що дозволяє виявляти приховані тренди та «вузькі місця» в реальному часі.
Клієнтоорієнтованість	Підвищення LTV (Lifetime Value) клієнта за рахунок впровадження кластеризації користувачів та персоналізованих маркетингових пропозицій, що базуються на історії замовлень.
Операційна стійкість	Мінімізація впливу людського фактору на процеси конвертації валют та одиниць виміру (lbs/kg), що є критичним для міжнародних перевезень.
Compliance & Security	Гарантія відповідності європейським регламентам захисту даних (GDPR) через архітектуру Privacy by Design, що знижує юридичні ризики компанії.

5.1.3. Відмінність від існуючих аналогів

Для чіткого позиціонування продукту необхідно розуміти його місце серед існуючих рішень. Ринок пропонує широкий спектр інструментів, проте більшість з них є або занадто вузькоспеціалізованими, або занадто універсальними, що вимагає значних ресурсів на адаптацію.

Аналіз конкурентного оточення дозволяє виділити три групи рішень-аналогів:

- **Група 1: Модулі аналітики в ERP-системах (SAP, Oracle).** Ці рішення є потужними, але надзвичайно дорогими та складними у впровадженні.

Вони часто не мають специфічної гнучкості для задач поштової логістики (наприклад, специфіки митного оформлення дрібних посилок).

- **Група 2: Універсальні BI-інструменти (Tableau, Qlik, Power BI).** Вони забезпечують візуалізацію, але не мають вбудованої бізнес-логіки. Без попередньої складної підготовки даних (ETL) та розробки математичних моделей вони є лише інструментами малювання графіків.
- **Група 3: Нішеві TMS-системи.** Фокусуються на транспорті, ігноруючи поведінкові фактори кінцевого споживача та маркетингову аналітику.

Пропонований стартап-проект відрізняється від аналогів **гібридним підходом**: він поєднує гнучкість мікросервісної архітектури, потужність сучасних BI-інструментів та вбудовану галузеву експертизу (domain knowledge) у сфері міжнародної електронної комерції. Ключовою конкурентною перевагою є вбудовані алгоритми нормалізації різнорідних даних (вага, валюта) та готові ML-моделі для прогнозування попиту, адаптовані під сезонність e-commerce.

5.2. Технологічний аудит та аналіз можливостей реалізації

Успішність реалізації ідеї стартапу критично залежить від наявності відповідних технологій та ресурсів. У попередніх розділах роботи було розроблено архітектуру та прототип системи. В даному підрозділі проводиться аудит технологічної зрілості рішення для його масштабування у комерційний продукт.

5.2.1. Технологічна здійсненність ідеї

Проведений аналіз технологічного стеку підтверджує високий рівень здійсненності проекту. Обрані технології належать до класу промислових стандартів, мають широку підтримку спільноти та високий рівень надійності.

1. **Серверна частина (Backend):** використання мови програмування **Java** та фреймворку **Spring Boot** є оптимальним вибором для побудови високонавантажених систем корпоративного рівня. Це забезпечує масштабованість, безпеку та можливість легкої інтеграції з існуючими

системами через REST API. Java дозволяє реалізувати складну бізнес-логіку обробки транзакцій та фінансових розрахунків з високою точністю.

2. **Обробка даних та Machine Learning:** вибір мови **Python** та бібліотек **Pandas**, **Scikit-learn** для реалізації аналітичних модулів дозволяє використовувати передові наукові розробки у сфері штучного інтелекту. Реалізовані в роботі алгоритми кластеризації (K-means) та прогнозування (Holt-Winters) демонструють достатню точність на тестових даних і можуть бути розгорнуті як окремі мікросервіси.
3. **Зберігання даних:** СУБД **PostgreSQL** обрана як надійне реляційне сховище, що підтримує роботу з JSON-об'єктами (важливо для логів трекінгу) та геопросторовими даними (PostGIS для маршрутизації). Це дозволяє уникнути витрат на ліцензування дорогих комерційних СУБД на етапі запуску стартапу.
4. **Візуалізація:** інтеграція з **Microsoft Power BI** дозволяє надати кінцевому користувачеві (менеджеру) звичний та зручний інтерфейс для роботи зі звітами, знімаючи з команди розробників необхідність створення складного фронтенду для візуалізації графіків.

Технологія є повністю доступною для команди проекту. Необхідності у розробці фундаментально нових наукових підходів немає; інноваційність полягає в унікальній комбінації та налаштуванні існуючих компонентів під специфіку бізнес-процесів замовника.

5.2.2. Аналіз ресурсних потреб

Для трансформації прототипу в повноцінний продукт (MVP – Minimum Viable Product) необхідно залучення певних ресурсів.

Таблиця 5.2. Матриця ресурсного забезпечення стартап-проєкту

Тип ресурсу	Характеристика потреб	Доступність та джерела
Інтелектуальні	Експертиза в логістиці, знання алгоритмів ML, навички розробки Java/Python.	Наявні у команди розробників (автора роботи) та наукових керівників.
Кадрові	Backend Developer, Data Scientist, Frontend Developer (Vue.js), QA Engineer.	Потребує формування команди (3-4 особи) або залучення аутсорсингу.
Технічні	Хмарна інфраструктура (AWS/Azure/Google Cloud) для хостингу, сервери баз даних, середовища розробки.	Доступні на ринку (модель IaaS/PaaS). Можливе використання грантових програм для стартапів від хмарних провайдерів.
Інформаційні	Доступ до API поштових операторів (DHL, FedEx), історичні дані про замовлення для навчання моделей.	Забезпечується партнерськими угодами з пілотним клієнтом.
Фінансові	Фінансування фонду оплати праці, оренди інфраструктури, маркетингу.	Потребує залучення інвестицій (ангельські інвестиції, гранти, власні кошти).

Аналіз показує, що найбільш критичним ресурсом є доступ до реальних історичних даних для калібрування математичних моделей. Саме тому стратегія розвитку передбачає тісну співпрацю з пілотним партнером на ранніх етапах.

5.3. Аналіз ринкових можливостей запуску стартап-проєкту

Оцінка ринкового середовища є ключовим етапом бізнес-планування. Вона дозволяє зрозуміти, чи існує платоспроможний попит на розроблене рішення та яке місце воно може зайняти на ринку.

5.3.1. Характеристика цільового ринку

Цільовим ринком для стартапу є глобальний ринок програмного забезпечення для управління ланцюгами постачання (Supply Chain Management Software), а точніше – його сегмент, орієнтований на e-commerce логістику та 3PL (Third Party Logistics) провайдерів.

Згідно з аналітичними звітами Gartner та Statista, цей ринок демонструє стійку тенденцію до зростання з середньорічним темпом (CAGR) близько 11%. Основними драйверами цього зростання є:

- Глобалізація торгівлі та збільшення частки транскордонних покупок.
- Підвищення вимог клієнтів до швидкості та прозорості доставки (ефект Amazon Prime).
- Необхідність цифровізації процесів для збереження конкурентоспроможності.

Потенційними клієнтами є середні та великі логістичні компанії, поштові оператори та e-commerce платформи, що мають власну логістичну інфраструктуру. Географічний фокус на початковому етапі – країни Східної Європи (Україна, Польща) з подальшою експансією на ринки ЄС та Північної Америки.

5.3.2. Аналіз конкурентного середовища

Для глибшого розуміння ринкової ситуації застосуємо модель п'яти конкурентних сил Майкла Портера:

1. **Загроза появи нових гравців (Середня):** бар'єр входу на ринок ПЗ є відносно низьким з технічної точки зору, проте розробка спеціалізованої аналітики вимагає глибокої галузевої експертизи та доступу до специфічних даних, що створює природний захист від новачків.
2. **Ринкова влада покупців (Висока):** великі логістичні оператори є вибагливими клієнтами, які часто мають власні ІТ-відділи. Вони можуть вимагати значної кастомізації продукту та знижок. Стратегія стартапу має

полягати у доведенні економічної ефективності аутсорсингу аналітики порівняно з внутрішньою розробкою (In-house).

3. **Ринкова влада постачальників (Низька):** постачальники хмарних ресурсів та інструментів розробки є численними та взаємозамінними, що знижує ризики залежності.
4. **Загроза товарів-замінників (Висока):** основним конкурентом є не стільки інше спеціалізоване ПЗ, скільки «ручна» аналітика в Excel або Google Sheets, яка все ще широко використовується в галузі. Також замінниками виступають універсальні ERP-системи з вбудованими аналітичними модулями.
5. **Інтенсивність конкуренції (Висока):** ринок насичений рішеннями від глобальних гравців. Шанс стартапу полягає у виборі вузької ніші – аналітика для cross-border форвардингу, де універсальні рішення часто є неефективними.

5.3.3. SWOT-аналіз стартап-проєкту

Узагальнення сильних та слабких сторін проєкту, а також зовнішніх можливостей та загроз, представлено у таблиці SWOT-аналізу.

Таблиця 5.3. SWOT-аналіз стартап-проєкту

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
<ol style="list-style-type: none">1. Глибока адаптація до бізнес-процесів конкретної ніші (cross-border logistics).2. Використання сучасних ML-алгоритмів для прогнозування.3. Відповідність архітектури вимогам GDPR (Privacy by Design).4. Гнучкість мікросервісної архітектури та легка інтеграція.	<ol style="list-style-type: none">1. Залежність від даних одного якірного замовника на старті.2. Відсутність відомого бренду та репутації на ринку.3. Обмежені фінансові ресурси для агресивного маркетингу.4. Необхідність постійної технічної підтримки API інтеграцій.
Можливості (Opportunities)	Загрози (Threats)
<ol style="list-style-type: none">1. Вихід на ринки країн ЄС та Північної Америки.2. Розширення функціоналу (інтеграція з Blockchain, AI-чати).3. Зростання ринку e-commerce та обсягів перевезень.4. Партнерство з вендорами CRM та ERP систем.	<ol style="list-style-type: none">1. Зміни у митному законодавстві та регулюванні транскордонної торгівлі.2. Кіберзагрози та ризики витоку конфіденційних даних.3. Поява аналогічних модулів у великих екосистемних гравців (Amazon, Google).4. Економічна нестабільність та волатильність валютних ринків.

Результати SWOT-аналізу свідчать, що проєкт має сильні внутрішні передумови для успіху, але потребує обережної стратегії управління зовнішніми ризиками, особливо в частині регуляторних змін та конкуренції.

5.4. Розроблення ринкової стратегії

Стратегія виведення продукту на ринок визначає, як саме стартап планує досягти своїх цільових клієнтів та переконати їх у перевагах свого рішення. Враховуючи B2B спрямування проєкту, стратегія фокусується на прямих продажах та побудові довгострокових відносин.

5.4.1. Стратегія охоплення ринку та опис цільових груп

На початковому етапі доцільно обрати **стратегію концентрованого маркетингу (нішова стратегія)**. Замість спроб конкурувати на всьому ринку логістичного ПЗ, стартап зосередиться на чітко визначеному сегменті, де його переваги є найбільш очевидними.

Ідеальний профіль клієнта (Ideal Customer Profile – ICP):

- **Тип бізнесу:** Логістичні оператори, що надають послуги форвардингу (доставка посилок з іноземних інтернет-магазинів) та фулфілменту.
- **Розмір:** Середній та великий бізнес (від 10 000 відправлень на місяць).
- **Географія:** Компанії зі штаб-квартирами у Східній Європі, що мають логістичні хаби в США, ЄС, Китаї.
- **Технологічна зрілість:** Компанії, що вже використовують CRM/WMS, але відчують потребу в поглибленій аналітиці та автоматизації звітності.

Обравши цю нішу, стартап зможе запропонувати продукт, який «з коробки» вирішує специфічні болі клієнта: мультивалютність, різні одиниці ваги, митні ліміти, специфіку адресних форматів різних країн.

5.4.2. Стратегія позиціонування

Позиціонування продукту будується навколо концепції «**Data-Driven Logistics Intelligence**». Основний меседж для ринку: «Ми перетворюємо хаос логістичних даних на чіткі та прибуткові управлінські рішення».

Ключові точки диференціації:

1. **Спеціалізація:** «Ми розуміємо cross-border краще, ніж універсальні BI».
2. **Прогнозування:** «Ми не просто показуємо, що сталося, ми прогнозуємо, що станеться».
3. **Безпека:** «Ми гарантуємо безпеку даних ваших клієнтів згідно з GDPR».

5.4.3. Стратегія ціноутворення

Для монетизації проєкту пропонується використовувати **гібридну модель (SaaS + Implementation Fee)**, що є стандартом для складних B2B продуктів.

1. **Setup Fee (Плата за впровадження):** разовий платіж за інтеграцію системи з інфраструктурою замовника, налаштування ETL-конекторів, навчання персоналу. Це дозволяє покрити початкові витрати на запуск клієнта.
2. **Subscription (Підписка):** щомісячна або щорічна плата за користування сервісом, доступ до хмарної інфраструктури, оновлень та технічної підтримки.

Тарифікація підписки може базуватися на:

- Обсязі оброблених транзакцій (кількість замовлень/посилок).
- Кількості підключених користувачів (аналітиків, менеджерів).
- Рівні функціональності (доступ до ML-прогнозів, розширених API).

Приклад тарифної сітки:

- **Basic:** базова звітність, щоденне оновлення даних – для невеликих обсягів.
- **Business:** прогнозування попиту, погодинне оновлення, розширена підтримка.
- **Enterprise:** індивідуальні доробки, виділений сервер, SLA.

5.5. Маркетингова програма стартап-проєкту

Маркетингова програма спрямована на формування обізнаності про продукт, генерацію лідів та перетворення їх на лояльних клієнтів. Враховуючи специфіку B2B, акцент робиться на експертному контенті та прямих комунікаціях.

5.5.1. Канали комунікації та просування

1. **Content Marketing & Thought Leadership:** публікація експертних статей, аналітичних оглядів та кейс-стаді (Case Studies) на професійних ресурсах

- (LinkedIn, Medium, профільні логістичні портали). Демонстрація реальних прикладів, як впровадження системи допомогло оптимізувати витрати (на прикладі пілотного проєкту).
2. **Участь у галузевих подіях:** презентація продукту на конференціях з логістику, e-commerce та Retail Tech. Це дозволяє встановити особистий контакт з особами, що приймають рішення (СТО, СОО).
 3. **Партнерська мережа:** співпраця з інтеграторами CRM та ERP систем. Пропозиція ІАС як додаткового цінного модуля для їхніх клієнтів.
 4. **Direct Sales (Account-Based Marketing):** цілеспрямований пошук та контакт з потенційними клієнтами через LinkedIn, холодні дзвінки та email-розсилки з персоналізованими пропозиціями.

5.5.2. Заходи зі стимулювання збуту

Для прискорення процесу прийняття рішень клієнтами пропонується використовувати:

- **Free Demo:** можливість замовити персональну демонстрацію системи на тестових даних.
- **Proof of Concept (PoC):** пілотний запуск системи на обмеженому обсязі даних клієнта (наприклад, один склад або один напрямок) за зниженою вартістю або безкоштовно на короткий термін. Це дозволяє клієнту переконатися в ефективності рішення на власних цифрах.
- **Referral Program:** знижки або бонуси для існуючих клієнтів за рекомендацію сервісу партнерам.

5.6. Фінансово-економічне обґрунтування та оцінка ризиків

Завершальним етапом розробки стартап-проєкту є оцінка його економічної ефективності та аналіз потенційних ризиків.

5.6.1. Розрахунок капітальних витрат (CAPEX) та операційних витрат (OPEX)

Для запуску MVP (Minimum Viable Product) необхідні початкові інвестиції. Основну статтю витрат складає фонд оплати праці команди розробників.

Таблиця 5.5. Орієнтовний кошторис витрат на етапі розробки MVP (4 місяці)

Стаття витрат	Опис	Вартість, \$
Розробка ПЗ	Оплата праці (Project Manager/Analyst, Backend Java Dev, Frontend Vue.js Dev, Data Engineer)	18000
Інфраструктура	Оренда серверів (AWS/Azure), домени, SSL-сертифікати	600
Ліцензії	Power BI Pro, засоби розробки, підписки на сервіси	800
Маркетинг	Створення лендінгу, презентаційних матеріалів, участь у подіях	1500
Адміністративні	Юридична реєстрація, бухгалтерський супровід	500
Резерв	Резервний фонд на непередбачувані витрати (10%)	2140
ВСЬОГО		23540

Після запуску проєкту виникають постійні операційні витрати (OPEX), які включають підтримку серверів, зарплату команди підтримки та sales-менеджерів, витрати на маркетинг. Орієнтовний місячний OPEX на початковому етапі складатиме близько \$3000 – \$4000.

5.6.2. Прогноз прибутковості та окупності

Припустимо консервативний сценарій розвитку, за якого у перший рік після запуску вдасться залучити 5 клієнтів на середній тариф (\$800/міс) та виконати 5 платних впроваджень (\$2000 за кожне).

- Дохід від підписки (ARR): 5 клієнтів * \$800 * 12 міс = \$48000.
- Дохід від впроваджень: 5 клієнтів * \$2000 = \$10000.
- Загальний річний дохід: \$58000.

Розрахунок фінансового результату:

\$58000 (Дохід) – \$23540 (Початкові інвестиції) – \$36000 (Річний OPEX, за умови п

Однак, вже на другий рік, при збереженні клієнтської бази та залученні нових (наприклад, +5 клієнтів), проєкт виходить на стійку прибутковість, оскільки витрати на розробку вже понесені, а масштабування SaaS-моделі має низьку маржинальну вартість. Точка беззбитковості (Break-even point) досягається приблизно на 14-16 місяць роботи. Це є нормальним показником для B2B стартапів.

5.6.3. Ризики проєкту та стратегії їх мінімізації

Таблиця 5.6. Реєстр ризиків стартап-проєкту

Ризик	Ймовірність	Вплив	Стратегія реагування (Mitigation Plan)
Технічні збої інтеграції	Середня	Високий	Розробка надійних адаптерів, детальне логування помилок, використання черг повідомлень (Message Brokers) для гарантованої доставки даних.
Зміна API партнерів	Висока	Середній	Постійний моніторинг оновлень API перевізників, модульна архітектура для швидкого оновлення конекторів.
Порушення GDPR	Низька	Критичний	Регулярний аудит безпеки, використання методів псевдонімізації на рівні БД, юридичні консультації.
Низьке сприйняття персоналом	Середня	Середній	Проведення якісних тренінгів, створення інтуїтивно зрозумілого UI, акцент на особистих вигодах для менеджерів.
Фінансова нестабільність	Середня	Високий	Формування фінансової подушки, диверсифікація клієнтського портфелю, контроль Cash Flow.

Висновки до розділу 5

У п'ятому розділі магістерської роботи розроблено комплексну концепцію стартап-проекту зі створення інформаційно-аналітичної системи для логістичного бізнесу.

1. Визначено, що ідея базується на вирішенні актуальної проблеми фрагментації даних у сфері міжнародної доставки та має чітку ціннісну пропозицію.
2. Технологічний аудит підтвердив можливість реалізації проекту на базі обраного стеку (Java, Python, PostgreSQL) з використанням наявних ресурсів.
3. Ринковий аналіз виявив наявність платоспроможного попиту в ніші cross-border логістики та дозволив сформулювати стратегію концентрованого маркетингу.
4. Фінансові розрахунки показали, що проєкт є інвестиційно привабливим з терміном окупності близько 1.5 року, що є прийнятним для B2B SaaS рішень.
5. Розроблена карта ризиків дозволяє передбачити можливі загрози та вжити превентивних заходів для забезпечення стійкості бізнесу.

Таким чином, впровадження розробленої системи є не лише науково обґрунтованим, але й комерційно перспективним кроком, що може стати основою для створення успішної технологічної компанії.

ЗАГАЛЬНІ ВИСНОВКИ

У магістерській кваліфікаційній роботі на тему «Розробка інформаційно-аналітичної системи підтримки управлінських рішень у сфері логістики на основі даних користувачів» вирішено актуальну науково-прикладну задачу підвищення ефективності управління логістичними процесами міжнародної доставки товарів. Дослідження базувалося на реальних даних та бізнес-процесах компанії, що забезпечило високу практичну цінність отриманих результатів.

У ході виконання роботи отримано наступні основні результати:

1. Проведено глибокий системний аналіз предметної області.

Встановлено, що сучасна логістика сектору B2C (cross-border e-commerce) характеризується експоненційним зростанням обсягів даних та високими вимогами до якості клієнтського сервісу. Виявлено ключові проблеми існуючого ландшафту інформаційних систем (фрагментація даних між WMS, TMS, CRM), які ускладнюють прийняття оперативних рішень. Доведено, що відсутність єдиного джерела аналітичної інформації призводить до фінансових втрат через курсові різниці, помилки в розрахунках ваги та зниження лояльності клієнтів.

2. Розроблено та обґрунтовано інформаційне забезпечення системи.

Створено концептуальну модель предметної області з використанням мови UML, яка формалізує ключові сутності («Клієнт», «Замовлення», «Відправлення», «Маршрут») та їхні взаємозв'язки. Спроектовано архітектуру сховища даних (DWH) та реляційну структуру бази даних на платформі PostgreSQL. Важливим досягненням є розробка алгоритмів ETL-процесів, що вирішують проблему гетерогенності даних: автоматична нормалізація одиниць виміру ваги (фунти/кілограми) та конвертація валют за історичними курсами, що забезпечує коректність фінансової звітності.

3. Створено математичне забезпечення ІАС. Для інтелектуального аналізу даних адаптовано та застосовано комплекс математичних моделей:

- Алгоритм кластеризації **K-means** у поєднанні з RFM-аналізом дозволив сегментувати клієнтську базу, виділивши групи лояльних та ризикових клієнтів для таргетованого маркетингу.
- Модель потрійного експоненційного згладжування **Хольта-Вінтерса** використана для прогнозування часових рядів (обсягів замовлень), що дозволяє враховувати сезонні фактори та тренди, характерні для електронної комерції.
- Для задач маршрутизації та оптимізації «останньої милі» імплементовано евристичні алгоритми (метод Кларка-Райта), що дозволяє мінімізувати транспортні витрати.
- Застосовано методи попередньої обробки даних (робастні статистики, Z-scaling) для забезпечення якості вхідної інформації.

4. **Спроектовано та реалізовано програмну архітектуру системи.**

Обґрунтовано вибір сучасного технологічного стеку: Java (Spring Boot) для надійного бекенду, Python (Pandas, Scikit-learn) для гнучкої аналітики, PostgreSQL з розширенням PostGIS для роботи з геоданими та Microsoft Power BI для інтерактивної візуалізації. Реалізована мікросервісна архітектура з використанням контейнеризації (Docker) забезпечує високу масштабованість, відмовостійкість та простоту розгортання.

5. **Забезпечено відповідність вимогам безпеки та приватності.**

В архітектуру системи закладено принципи **Privacy by Design**. Розроблено та реалізовано механізми псевдонімізації персональних даних на рівні бази даних, що гарантує відповідність системи вимогам європейського регламенту **GDPR**. Це дозволяє використовувати дані для аналітики без ризику порушення приватності користувачів.

6. **Розроблено стартап-проект для комерціалізації результатів.**

Проведено комплексний маркетинговий та фінансовий аналіз, який підтвердив наявність ринкової ніші для спеціалізованих аналітичних рішень у сфері логістики. Розраховано, що впровадження системи як B2B SaaS-продукту є економічно доцільним, з точкою беззбитковості на рівні

14-16 місяців. Визначено цільові групи клієнтів та розроблено стратегію виведення продукту на ринок.

Практичне значення роботи полягає у створенні дієвого інструменту, який дозволяє логістичним компаніям трансформувати накопичені дані у конкурентну перевагу. Реалізований прототип системи забезпечує:

- Автоматизацію підготовки управлінської звітності, скорочуючи час на її формування з годин до хвилин.
- Підвищення точності прогнозування навантаження на склади, що дозволяє оптимізувати використання ресурсів.
- Покращення клієнтського досвіду через прозорість процесів та персоналізацію сервісу.

Рекомендації щодо подальшого розвитку включають розширення функціоналу системи за рахунок інтеграції з блокчейн-платформами для забезпечення незмінності даних ланцюга постачання, використання методів глибокого навчання (Deep Learning) для комп'ютерного зору при обробці посилок на складах, а також розробку мобільного застосунку для оперативного моніторингу KPI менеджерами компанії.

Загалом, магістерська кваліфікаційна робота є завершеним самостійним дослідженням, яке вирішує важливу науково-практичну проблему і має значний потенціал для впровадження у реальний сектор економіки.

ДОДАТКИ

ДОДАТОК А

Інформаційні та концептуальні моделі системи (UML, IDEF0, BPMN)

У Додатку А представлено комплексний набір діаграм, що описують статичну структуру та динамічну поведінку розробленої ІАС. Використання стандартизованих нотацій моделювання дозволило формалізувати вимоги до системи та спростити комунікацію між зацікавленими сторонами (розробниками, аналітиками та менеджментом логістичної компанії).

А.1 Концептуальна модель предметної області (IDEF0)

Для аналізу логістичних процесів міжнародної доставки було розроблено функціональну модель у нотації IDEF0. Вона декомпозує загальний процес «Управління міжнародною доставкою» на підпроцеси.

Контекстна діаграма А-0:

- **Вхід (Input):** Замовлення користувачів, Інформація про товари, Митні нормативи.
- **Управління (Control):** Законодавство про ЗЕД, Внутрішні регламенти компанії, Тарифна сітка.
- **Механізми (Mechanism):** Персонал (менеджери, складські працівники), ІАС, Транспортний парк.
- **Вихід (Output):** Доставлений вантаж, Аналітичні звіти, Оновлений статус замовлення.

Декомпозиція першого рівня (A0):

1. **Блок А1: Реєстрація та консолідація замовлень.**

- *Опис:* Користувачі реєструють трекінг-номери отриманих від закордонних магазинів посилок. Система виконує первинну валідацію даних (вага, вартість).
- *Проблематика:* Фрагментація даних, що виникає через різні формати вводу (наприклад, фунти замість кілограмів), вирішується на цьому етапі шляхом автоматичної конвертації.

2. Блок A2: Складська обробка (Warehousing).

- *Опис:* Приймання товару на складі в країні відправлення (США/Європа), зважування, маркування, формування палет.
- *Взаємодія:* Дані з ваг автоматично передаються в ІАС через API.

3. Блок A3: Митне оформлення та транзит.

- *Опис:* Генерація реєстрів для митниці, розрахунок митних платежів на основі задекларованої вартості.

4. Блок A4: Розподіл та доставка "останньої милі".

- *Опис:* Оптимізація маршрутів кур'єрів з використанням алгоритму VRP (Vehicle Routing Problem).

A.2 Діаграма варіантів використання (Use Case Diagram)

Діаграма варіантів використання деталізує функціональні можливості системи для різних ролей користувачів. Враховуючи специфіку cross-border торгівлі, виділено такі сценарії:

Актор: Клієнт (User)

- **UC-01: Створення декларації.** Користувач вносить дані про товар.
Розширення: Якщо вартість перевищує безмитний ліміт (150 EUR), система автоматично розраховує суму мита.
- **UC-02: Відстеження посилки.** Перегляд статусу в реальному часі.
Система агрегує статуси від партнерів-перевізників.
- **UC-03: Управління адресами доставки.** Додавання декількох адрес для різних отримувачів.

Актор: Логістичний аналітик (Analyst)

- **UC-04: Аналіз ефективності складів.** Перегляд дашбордів з KPI (завантаженість, швидкість обробки).
- **UC-05: Прогнозування попиту.** Запуск моделі Хольта-Вінтерса для передбачення пікових навантажень (наприклад, перед "Чорною п'ятницею").
- **UC-06: Сегментація клієнтів.** Запуск кластеризації K-means для виявлення VIP-клієнтів та груп ризику.

Актор: Адміністратор (Admin)

- **UC-07: Налаштування параметрів системи.** Зміна коефіцієнтів конвертації валют, додавання нових складів.

А.3 Діаграма класів (Class Diagram)

Діаграма класів відображає архітектуру даних та об'єктну модель системи. Нижче наведено детальний опис ключових класів, спроектованих для забезпечення гнучкості та розширюваності.

Назва класу	Атрибути	Методи	Опис та призначення
Order (Замовлення)	- id: UUID - trackingNumber: String - declaredValue: Money - weight: Weight - status: OrderStatus - items: List<Item>	+ calculateTotalValue() + updateStatus(newStatus) + validateWeight()	Центральна сутність системи. Використовує патерн <i>Value Object</i> для атрибутів Money та Weight для уникнення помилок з одиницями виміру.
User (Користувач)	- id: UUID - email: String - segment: CustomerSegment - loyaltyPoints: int	+ getOrderHistory() + applyDiscount()	Зберігає профіль користувача. Атрибут segment оновлюється результатами роботи ML-модуля.
Warehouse (Склад)	- code: String - location: GeoPoint - capacity: double - currentLoad: double	+ acceptShipment() + dispatchShipment() + getForecast()	Представляє фізичний об'єкт логістики. Містить методи для перевірки доступної ємності.
PredictionModel (Абстракція)	- modelType: ModelType - lastTrained: DateTime	+ train(data) + predict(horizon)	Базовий клас для аналітичних моделей. Реалізує патерн <i>Strategy</i> для вибору алгоритму прогнозування.
VRP_Solver (Оптимізатор)	- distanceMatrix: double - constraints: List<Constraint>	+ solveRoute() + optimizeCosts()	Клас-обгортка для математичного модуля оптимізації маршрутів.

Зв'язки:

- User має відношення *один-до-багатьох* з Order.
- Order має відношення *багато-до-одного* з Warehouse (через точку відправлення та призначення).
- PredictionModel використовує дані з OrderRepository для навчання.

А.4 Діаграма послідовності (Sequence Diagram): Процес обробки аномалії

ваги

Ця діаграма ілюструє взаємодію компонентів при виявленні невідповідності ваги (Outlier Detection), що є критичним для коректної тарифікації.

1. **WarehouseWorker** сканує посилку та вводить фактичну вагу через **TerminalUI**.
2. **TerminalUI** надсилає запит `updateWeight(orderId, realWeight)` до **LogisticsService**.
3. **LogisticsService** викликає **AnomalyDetector** (компонент на Python) для перевірки: `checkOutlier(declaredWeight, realWeight)`.
4. **AnomalyDetector** використовує статистичний метод IQR (Interquartile Range) на історичних даних для визначення допустимого відхилення.
5. *Альтернатива 1 (Норма):* Якщо відхилення в межах норми, **LogisticsService** оновлює статус замовлення на "Ready for Shipment".
6. *Альтернатива 2 (Аномалія):* Якщо виявлено значне відхилення (наприклад, задекларовано 1 кг, фактично 10 кг), система:
 - Блокує замовлення (статус "On Hold").
 - Створює тикет для **Manager** через **NotificationService**.
 - Надсилає сповіщення клієнту про необхідність доплати.

A.5 Схема бази даних (ER-Diagram)

Схема бази даних спроектована з урахуванням вимог до аналітичної обробки даних (OLAP) та транзакційної цілісності (OLTP). Використано підхід розділення на оперативну базу даних та сховище даних (DWH).

- **Операційна БД (OLTP):** Нормалізована до 3-ї нормальної форми (3NF). Основні таблиці: `users`, `orders`, `order_items`, `shipments`, `invoices`.
- **Сховище даних (OLAP):** Схема «Зірка» (Star Schema).
 - **Fact Table:** `fact_delivery_performance` (містить метрики: час доставки, вартість, затримки).
 - **Dimension Tables:** `dim_time` (ієрархія часу), `dim_location` (географія), `dim_customer` (сегментація), `dim_carrier` (перевізник).

ДОДАТОК Б

Лістинги програмного коду реалізації бази даних та DWH (SQL)

У Додатку Б наведено SQL-скрипти для розгортання структури бази даних PostgreSQL, налаштування індексів для оптимізації швидкодії та реалізації бізнес-логіки на рівні бази даних.

Б.1 Скрипт ініціалізації схеми DWH (Data Warehouse)

```
/*
 * Скрипт створення структури сховища даних для логістичної
 системи.
 * Використовується схема 'dwh_logistics'.
 * Реалізовано методологію розмірного моделювання (Dimensional
 Modeling).
 */

CREATE SCHEMA IF NOT EXISTS dwh_logistics;

-- 1. Таблиця вимірів: Користувачі (SCD Type 2 - повільно змінні
 виміри)
CREATE TABLE dwh_logistics.dim_user (
    user_sk BIGSERIAL PRIMARY KEY,          -- Суррогатний ключ
    user_id UUID NOT NULL,                 -- Натуральний ключ з
 операційної системи
    email VARCHAR(255),
    full_name VARCHAR(255),
    registration_date DATE,
    country_iso_code CHAR(2),

    -- Атрибути сегментації
    rfm_segment VARCHAR(50),               -- Сегмент (VIP, Новий,
 Втрачений)
    cluster_group INT,                     -- Група кластера
 (результат K-means)

    -- Поля для відстеження історичних змін
    is_current BOOLEAN DEFAULT TRUE,
    valid_from TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    valid_to TIMESTAMP DEFAULT '9999-12-31 23:59:59'
);

CREATE INDEX idx_dim_user_natural ON
dwh_logistics.dim_user(user_id);

-- 2. Таблиця вимірів: Час (Календар)
CREATE TABLE dwh_logistics.dim_date (
    date_sk INT PRIMARY KEY,              -- Формат YYYYMMDD
    full_date DATE NOT NULL,
```

```

    day_of_week INT,
    day_name VARCHAR(20),
    week_of_year INT,
    month_num INT,
    month_name VARCHAR(20),
    quarter INT,
    year_num INT,
    is_weekend BOOLEAN,
    is_holiday BOOLEAN          -- Важливо для розрахунку
SLA доставки
);

-- 3. Таблиця вимірів: Логістичні хаби
CREATE TABLE dwh_logistics.dim_warehouse (
    warehouse_sk SERIAL PRIMARY KEY,
    warehouse_code VARCHAR(50) UNIQUE,
    country VARCHAR(100),
    city VARCHAR(100),
    address TEXT,
    capacity_limit_m3 DECIMAL(10, 2),    -- Максимальна ємність
    warehouse_type VARCHAR(50)          -- 'Consolidation',
'Distribution', 'Customs'
);

-- 4. Таблиця фактів: Виконання замовлень
CREATE TABLE dwh_logistics.fact_orders (
    fact_id BIGSERIAL PRIMARY KEY,

    -- Зовнішні ключі до вимірів
    date_created_sk INT REFERENCES
dwh_logistics.dim_date(date_sk),
    date_delivered_sk INT REFERENCES
dwh_logistics.dim_date(date_sk),
    user_sk BIGINT REFERENCES dwh_logistics.dim_user(user_sk),
    warehouse_origin_sk INT REFERENCES
dwh_logistics.dim_warehouse(warehouse_sk),
    warehouse_dest_sk INT REFERENCES
dwh_logistics.dim_warehouse(warehouse_sk),

    -- Метрики (Facts)
    order_id UUID NOT NULL,
    weight_kg DECIMAL(10, 3),           -- Уніфікована вага
    volume_m3 DECIMAL(10, 4),          -- Об'ємна вага
    declared_value_eur DECIMAL(15, 2), -- Вартість приведена до
базової валюти (EUR)
    shipping_cost_eur DECIMAL(15, 2),  -- Фактичні витрати на
доставку

    processing_time_hours INT,         -- Час обробки на складі
    transit_time_days INT,             -- Час у дорозі
    total_delivery_time_days INT,      -- Повний цикл

    is_delayed BOOLEAN DEFAULT FALSE,  -- Чи порушено SLA

```

```

        is_outlier BOOLEAN DEFAULT FALSE      -- Прапорець аномалії
    );

-- Partitioning (секціонування) таблиці фактів по роках для
підвищення продуктивності
CREATE INDEX idx_fact_orders_date ON
dwh_logistics.fact_orders(date_created_sk);

```

Б.2 Збережена процедура для виявлення викидів (Outlier Detection)

Оскільки одним із завдань роботи є фільтрація аномальних значень (наприклад, помилки оператора при введенні ваги), реалізовано процедуру на PL/pgSQL, яка використовує статистичні функції для маркування викидів безпосередньо в базі даних.

```

CREATE OR REPLACE FUNCTION dwh_logistics.mark_weight_outliers(
    p_start_date INT,
    p_end_date INT
)
RETURNS VOID AS $$
DECLARE
    v_q1 DECIMAL;
    v_q3 DECIMAL;
    v_iqr DECIMAL;
    v_lower_bound DECIMAL;
    v_upper_bound DECIMAL;
BEGIN
    /*
    * Алгоритм на основі IQR (Interquartile Range).
    * Робастний метод, стійкий до екстремальних викидів.
    * Використовується замість правила 3-х сигм, оскільки
розподіл ваги посилок не є нормальним.
    */

    -- 1. Розрахунок квантилів (25% та 75%)
    SELECT
        percentile_cont(0.25) WITHIN GROUP (ORDER BY weight_kg),
        percentile_cont(0.75) WITHIN GROUP (ORDER BY weight_kg)
    INTO v_q1, v_q3
    FROM dwh_logistics.fact_orders
    WHERE date_created_sk BETWEEN p_start_date AND p_end_date;

    -- 2. Розрахунок IQR
    v_iqr := v_q3 - v_q1;

    -- 3. Визначення меж (коефіцієнт 1.5 є стандартним для
виявлення викидів)
    v_lower_bound := v_q1 - 1.5 * v_iqr;
    v_upper_bound := v_q3 + 1.5 * v_iqr;

    -- Захист від від'ємних значень ваги

```

```

IF v_lower_bound < 0 THEN
    v_lower_bound := 0;
END IF;

RAISE NOTICE 'Analysis Period: % - %. Bounds: % kg - % kg',
    p_start_date, p_end_date, v_lower_bound,
v_upper_bound;

-- 4. Оновлення записів
UPDATE dwh_logistics.fact_orders
SET is_outlier = TRUE
WHERE date_created_sk BETWEEN p_start_date AND p_end_date
    AND (weight_kg < v_lower_bound OR weight_kg >
v_upper_bound);

END;
$$ LANGUAGE plpgsql;

```

Б.3 Представлення (View) для інтеграції з Power BI

Для забезпечення безпечного доступу BI-системи до даних створено матеріалізоване представлення, яке попередньо агрегує дані.

```

CREATE MATERIALIZED VIEW dwh_logistics.mv_bi_dashboard_data AS
SELECT
    w.country AS origin_country,
    d.year_num,
    d.month_name,
    u.rfm_segment,
    COUNT(f.order_id) AS orders_count,
    SUM(f.shipping_cost_eur) AS total_revenue,
    AVG(f.total_delivery_time_days) AS avg_delivery_time,
    SUM(CASE WHEN f.is_delayed THEN 1 ELSE 0 END)::DECIMAL /
COUNT(*) * 100 AS delay_rate_percent
FROM dwh_logistics.fact_orders f
JOIN dwh_logistics.dim_warehouse w ON f.warehouse_origin_sk =
w.warehouse_sk
JOIN dwh_logistics.dim_date d ON f.date_created_sk = d.date_sk
JOIN dwh_logistics.dim_user u ON f.user_sk = u.user_sk
WHERE f.is_outlier = FALSE -- Виключаємо аномалії з фінансової
звітності
GROUP BY w.country, d.year_num, d.month_name, u.rfm_segment
WITH DATA;

-- Індекс для прискорення оновлення
CREATE INDEX idx_mv_bi_country ON
dwh_logistics.mv_bi_dashboard_data(origin_country);

```

ДОДАТОК В

Лістинги програмного коду серверної частини (Java/Spring Boot)

Серверна частина реалізована на платформі Java 17 з використанням фреймворку Spring Boot 3.0. Архітектура побудована за принципами RESTful API та мікросервісного підходу.

V.1 Основний контролер (LogisticsController.java)

Цей клас обробляє HTTP-запити від клієнтських додатків, виконує валідацію вхідних даних та делегує бізнес-логіку сервісам.

```
package ua.nltu.thesis.logistics.controller;

import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.tags.Tag;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import ua.nltu.thesis.logistics.dto.OrderRequestDTO;
import ua.nltu.thesis.logistics.dto.OrderResponseDTO;
import ua.nltu.thesis.logistics.service.OrderProcessingService;

import jakarta.validation.Valid;
import java.util.List;
import java.util.UUID;

/**
 * REST Controller for managing logistics orders.
 * Provides endpoints for creation, tracking, and analysis
 * triggers.
 */
@RestController
@RequestMapping("/api/v1/logistics")
@RequiredArgsConstructor
@Slf4j
@Tag(name = "Logistics Management", description = "Core API for
order processing")
public class LogisticsController {

    private final OrderProcessingService orderService;

    @PostMapping("/orders")
    @Operation(summary = "Register new shipment", description =
"Creates a new order with automatic weight unit conversion")
    public ResponseEntity<OrderResponseDTO> createOrder(@Valid
@RequestBody OrderRequestDTO request) {
        log.info("Received request to create order for user: {}",
request.getUserId());
        OrderResponseDTO response =
orderService.processNewOrder(request);
        return ResponseEntity.ok(response);
    }
}
```

```

    }

    @GetMapping("/orders/{trackingNumber}")
    @Operation(summary = "Track shipment", description =
"Retrieves current status and estimated delivery time")
    public ResponseEntity<OrderResponseDTO>
trackOrder(@PathVariable String trackingNumber) {
        return
ResponseEntity.ok(orderService.findByTrackingNumber(trackingNumber
));
    }

    @GetMapping("/users/{userId}/orders")
    @Operation(summary = "User Order History", description =
"Returns filtered list of orders for specific user")
    public ResponseEntity<List<OrderResponseDTO>>
getUserHistory(@PathVariable UUID userId) {
        return
ResponseEntity.ok(orderService.findAllByUser(userId));
    }
}

```

V.2 Сервіс обробки замовлень (OrderProcessingService.java)

Ключовий компонент бізнес-логіки. Реалізує вимогу щодо уніфікації одиниць виміру (lbs/kg) та інтеграції з модулем аналітики.

```

package ua.nltu.thesis.logistics.service;

import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import ua.nltu.thesis.logistics.domain.Order;
import ua.nltu.thesis.logistics.dto.OrderRequestDTO;
import ua.nltu.thesis.logistics.dto.OrderResponseDTO;
import ua.nltu.thesis.logistics.exception.ValidationException;
import ua.nltu.thesis.logistics.repository.OrderRepository;
import ua.nltu.thesis.logistics.integration.PythonAnalyticsClient;
import ua.nltu.thesis.logistics.util.UnitConverter;

import java.math.BigDecimal;
import java.time.LocalDateTime;

@Service
@RequiredArgsConstructor
public class OrderProcessingService {

    private final OrderRepository orderRepository;
    private final PythonAnalyticsClient analyticsClient;
    private final CurrencyService currencyService;

    /**

```

```

    * Processes a new order: converts units, calculates costs,
and saves to DB.
    */
@Transactional
public OrderResponseDTO processNewOrder(OrderRequestDTO dto) {
    // 1. Уніфікація ваги (Вимога 3.1.2 МКР)
    // Всі розрахунки в системі ведуться в кг
    double weightKg = dto.getWeight();
    if ("LBS".equalsIgnoreCase(dto.getWeightUnit())) {
        weightKg = UnitConverter.poundsToKg(dto.getWeight());
    }

    // 2. Уніфікація валюти (Вимога підтримки
мультивалютності)
    BigDecimal valueInEur =
currencyService.convertToBaseCurrency(
        dto.getDeclaredValue(),
        dto.getCurrency()
    );

    // 3. Створення сутності
    Order order = new Order();
    order.setTrackingNumber(dto.getTrackingNumber());
    order.setUserId(dto.getUserId());
    order.setWeightKg(weightKg);
    order.setDeclaredValueEur(valueInEur);
    order.setStatus("NEW");
    order.setCreatedAt(LocalDateTime.now());

    // 4. Перевірка на аномалії (виклик Python-модуля)
    // Синхронний виклик для миттєвого фідбеку
    boolean isAnomaly = analyticsClient.checkAnomaly(weightKg,
valueInEur);
    if (isAnomaly) {
        order.setFlagged(true);
        order.addSystemNote("WARN: Weight deviation detected
by IQR algorithm");
    }

    Order savedOrder = orderRepository.save(order);
    return mapToDTO(savedOrder);
}

private OrderResponseDTO mapToDTO(Order order) {
    return OrderResponseDTO.builder()
        .trackingNumber(order.getTrackingNumber())
        .status(order.getStatus())
        .weightKg(order.getWeightKg())
        .isFlagged(order.isFlagged())

.estimatedDelivery(order.getEstimatedDeliveryDate())
        .build();
}

```

```
}
```

В.3 Утилита конвертації (UnitConverter.java)

```
package ua.nltu.thesis.logistics.util;

import java.math.BigDecimal;
import java.math.RoundingMode;

/**
 * Utility class for unit standardization.
 * Critical for cross-border logistics where US/UK use Imperial
 * system.
 */
public class UnitConverter {

    private static final double LBS_TO_KG_FACTOR = 0.45359237;
    private static final double INCH_TO_CM_FACTOR = 2.54;

    public static double poundsToKg(double pounds) {
        BigDecimal bd = BigDecimal.valueOf(pounds *
LBS_TO_KG_FACTOR);
        return bd.setScale(3, RoundingMode.HALF_UP).doubleValue();
    }

    public static double inchesToCm(double inches) {
        BigDecimal bd = BigDecimal.valueOf(inches *
INCH_TO_CM_FACTOR);
        return bd.setScale(2, RoundingMode.HALF_UP).doubleValue();
    }

    public static double calculateVolumetricWeight(double l,
double w, double h, double divisor) {
        // Стандарт IATA: divisor = 5000 (для см) або 6000
        return (l * w * h) / divisor;
    }
}
```

ДОДАТОК Г

Лістинги програмного коду математичного забезпечення (Python)

Цей додаток містить реалізацію математичних моделей, описаних у Розділі 3 пояснювальної записки. Використано мову Python та бібліотеки Data Science стеку.

Г.1 Модуль кластеризації користувачів (clustering_service.py)

Реалізує метод K-means для сегментації клієнтської бази на основі RFM-аналізу.

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sqlalchemy import create_engine
import logging

# Налаштування логування
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

class UserSegmentationService:
    def __init__(self, db_connection_string):
        self.engine = create_engine(db_connection_string)
        self.scaler = StandardScaler()
        self.model = None

    def fetch_data(self):
        """
        Завантаження даних з DWH для RFM аналізу.
        Recency - днів з останнього замовлення
        Frequency - кількість замовлень
        Monetary - сума витрат
        """
        query = """
        SELECT user_id,
               MAX(order_date) as last_order,
               COUNT(order_id) as frequency,
               SUM(total_amount) as monetary
        FROM dwh_logistics.fact_orders
        GROUP BY user_id
        """
        df = pd.read_sql(query, self.engine)

        # Розрахунок Recency
        snapshot_date = pd.to_datetime('today')
        df['last_order'] = pd.to_datetime(df['last_order'])
        df['recency'] = (snapshot_date - df['last_order']).dt.days

        return df
```

```

def perform_clustering(self, n_clusters=4):
    """
    Виконання алгоритму K-means.
    n_clusters визначається за методом ліктя (Elbow Method) на
    етапі дослідження.
    """
    df = self.fetch_data()

    # Препроцесинг: логарифмування для зменшення асиметрії
    (skewness) грошових даних
    df_log = df[['recency', 'frequency',
'monetary']].apply(lambda x: np.log(x + 1))

    # Нормалізація (Standardization)
    df_normalized = self.scaler.fit_transform(df_log)

    # Навчання моделі
    self.model = KMeans(n_clusters=n_clusters,
random_state=42, n_init=10)
    clusters = self.model.fit_predict(df_normalized)

    df['cluster'] = clusters

    # Аналіз центроїдів для інтерпретації
    summary = df.groupby('cluster').agg({
        'recency': 'mean',
        'frequency': 'mean',
        'monetary': ['mean', 'count']
    })
    logger.info(f"Clustering Summary:\n{summary}")

    self.save_results(df)
    return summary

def save_results(self, df):
    """Збереження результатів назад у базу даних"""
    df[['user_id', 'cluster']].to_sql(
        'user_clusters_temp',
        self.engine,
        if_exists='replace',
        index=False
    )
    logger.info("Cluster assignments saved to DB.")

if __name__ == "__main__":
    service =
UserSegmentationService("postgresql://user:pass@localhost:5432/dwh
")
    service.perform_clustering()

```

Г.2 Модуль прогнозування попиту (forecasting_service.py)

Реалізація методу експоненційного згладжування Хольта-Вінтерса для прогнозування завантаженості складів.

```
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

class DemandForecaster:
    def __init__(self, data_path):
        self.data = pd.read_csv(data_path, parse_dates=['date'],
index_col='date')
        # Ресемплінг по днях, заповнення пропусків нулями
        self.data = self.data.resample('D').sum().fillna(0)

    def train_predict(self, seasonal_periods=7, forecast_days=14):
        """
        Навчання моделі та прогноз.
        Використовується адитивний тренд та мультиплікативна
сезонність,
        оскільки амплітуда коливань замовлень залежить від
загального обсягу.
        """
        train_data = self.data['order_count']

        # Ініціалізація моделі Holt-Winters
        model = ExponentialSmoothing(
            train_data,
            trend='add',
            seasonal='mul',
            seasonal_periods=seasonal_periods,
            damped_trend=True # Згасаючий тренд для уникнення
нереалістичного зростання
        )

        fit_model = model.fit()

        # Прогноз
        forecast = fit_model.forecast(forecast_days)

        # Візуалізація (збереження графіку для звіту)
        plt.figure(figsize=(12, 6))
        plt.plot(train_data.index[-30:], train_data[-30:],
label='Historical Data')
        plt.plot(forecast.index, forecast, label='Forecast',
color='red', linestyle='--')
        plt.title('Logistics Demand Forecast (Holt-Winters)')
        plt.legend()
        plt.savefig('forecast_plot.png')

        return forecast.to_json()
```

```
# Приклад використання
# forecaster = DemandForecaster('daily_warehouse_load.csv')
# print(forecaster.train_predict())
```

Г.3 Модуль маршрутизації VRP (routing_optimizer.py)

Вирішення задачі маршрутизації транспортних засобів (Vehicle Routing Problem) з використанням бібліотеки PuLP (лінійне програмування). Це відповідає формалізації MILP, наведеній у розділі 3.5 роботи.

```
import pulp
import numpy as np

def solve_cvrp(distance_matrix, demands, vehicle_capacity,
num_vehicles):
    """
    Вирішує задачу CVRP (Capacitated Vehicle Routing Problem).

    Аргументи:
    distance_matrix: NxN матриця відстаней
    demands: масив потреб клієнтів (вага посилок)
    vehicle_capacity: вантажопідйомність авто
    num_vehicles: кількість доступних машин
    """
    n = len(distance_matrix)
    nodes = range(n)

    # Створення моделі
    prob = pulp.LpProblem("Logistics_VRP", pulp.LpMinimize)

    # Змінні рішення
    # x[i][j] = 1, якщо шлях йде від i до j
    x = pulp.LpVariable.dicts("x", (nodes, nodes), cat='Binary')

    # u[i] - допоміжні змінні для усунення підциклів (MTZ
    formulation)
    u = pulp.LpVariable.dicts("u", nodes, lowBound=0,
    upBound=vehicle_capacity, cat='Continuous')

    # Цільова функція: Мінімізація загальної відстані
    prob += pulp.lpSum(distance_matrix[i][j] * x[i][j] for i in
    nodes for j in nodes if i!= j)

    # Обмеження 1: З кожного клієнта виїжджає рівно одна машина
    for i in range(1, n):
        prob += pulp.lpSum(x[i][j] for j in nodes if i!= j) == 1

    # Обмеження 2: В кожного клієнта в'їжджає рівно одна машина
    for j in range(1, n):
        prob += pulp.lpSum(x[i][j] for i in nodes if i!= j) == 1
```

```

# Обмеження 3: Потік з депо (вузол 0)
prob += pulp.lpSum(x[j] for j in range(1, n)) <= num_vehicles
prob += pulp.lpSum(x[i] for i in range(1, n)) <= num_vehicles

# Обмеження 4: Вантажопідйомність та усунення підциклів
for i in range(1, n):
    for j in range(1, n):
        if i != j:
            # u_j - u_i >= demand_j - Capacity * (1 - x_ij)
            prob += u[j] - u[i] >= demands[j] -
vehicle_capacity * (1 - x[i][j])

# Запуск солвера (CBC)
status = prob.solve(pulp.PULP_CBC_CMD(msg=0))

# Формування результату
if pulp.LpStatus[status] == 'Optimal':
    routes =
    for i in nodes:
        for j in nodes:
            if i != j and pulp.value(x[i][j]) > 0.9:
                routes.append((i, j))
    return routes
else:
    return None

```

ДОДАТОК Д

Інтерфейс користувача та клієнтський код (Vue.js)

Розроблена система має веб-інтерфейс, побудований за принципом SPA (Single Page Application). Нижче наведено код компонентів для відображення аналітики та створення замовлень.

Д.1 Компонент DashboardStats.vue

Відображає ключові показники ефективності (KPI) та графік прогнозу.

```

<template>
  <div class="dashboard">
    <div class="stats-grid">
      <div class="stat-card" v-for="metric in metrics"
:key="metric.id">
        <h3>{{ metric.title }}</h3>
        <p class="value">{{ metric.value }}</p>
        <span :class="['trend', metric.trend > 0? 'positive' :
'negative']">
          {{ metric.trend }}% vs last month
        </span>
      </div>
    </div>
  </div>

```

```

</div>

<div class="chart-container">
  <canvas id="forecastChart"></canvas>
</div>
</div>
</template>

<script>
import { onMounted, ref } from 'vue';
import Chart from 'chart.js/auto';
import axios from '@plugins/axios';

export default {
  name: 'DashboardStats',
  setup() {
    const metrics = ref();

    const loadData = async () => {
      try {
        const response = await axios.get('/api/v1/analytics/kpi');
        metrics.value = response.data.metrics;
        renderChart(response.data.forecast);
      } catch (error) {
        console.error("Failed to load analytics:", error);
      }
    };

    const renderChart = (forecastData) => {
      const ctx = document.getElementById('forecastChart');
      new Chart(ctx, {
        type: 'line',
        data: {
          labels: forecastData.dates,
          datasets:
        },
        options: {
          responsive: true,
          plugins: {
            title: { display: true, text: 'Прогноз навантаження на
14 днів' }
          }
        }
      });
    };

    onMounted(loadData);
    return { metrics };
  }
}
</script>

<style scoped>

```

```
.stats-grid { display: grid; grid-template-columns: repeat(4,
1fr); gap: 20px; margin-bottom: 30px; }
.stat-card { background: white; padding: 20px; border-radius: 8px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1); }
.trend.positive { color: green; }
.trend.negative { color: red; }
</style>
```