

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

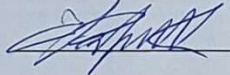
Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління
(повна назва кафедри)

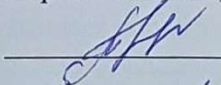
Магістерська кваліфікаційна робота
на тему:

«Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями»

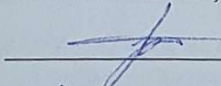
Виконав: студент 2-го курсу,
групи ЗАКІТР-23м
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка
(шифр і назва спеціальності)

 Дмитро ФАРИНЯК
(ім'я та прізвище)

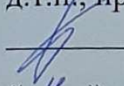
Керівник: к.т.н., доц. каф. КСУ

 Тетяна ГРИЦУК
(ім'я та прізвище)
« 9 » 12 2024 р.

Опонент: к.т.н., доцент каф. АІТ

 Володимир ГАРМАШ
(ім'я та прізвище)
« 4 » 12 2024 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.

 В'ячеслав КОВТУН
(ім'я та прізвище)
« 4 » 12 2024 р.

Вінниця ВНТУ - 2024 рік

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра комп'ютерних систем управління

Рівень вищої освіти другий (магістерський)

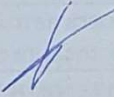
Галузь знань – 17 Електроніка, автоматизація та електронні комунікації

Спеціальність – 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітньо-професійна програма – Інформаційні системи і Інтернет речей

ЗАТВЕРДЖУЮ

Зав. кафедри КСУ

 В'ячеслав Ковтун

«14» жовтня 2024 року

З А В Д А Н Н Я
НА МАГІСТЕРСКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Фариняку Дмитру Леонідовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями»

керівник магістерської кваліфікаційної роботи к.т.н. доцент кафедри КСУ
Грищук Тетяна Вікторівна,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ВНТУ від «17» вересня 2024 р. № 310

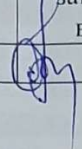
2. Термін подання студентом роботи «1» грудня 2024 р.

3. Вихідні дані до магістерської кваліфікаційної роботи включають підтримку веб-платформ; максимальний час відгуку системи – 3 секунди; авторизація користувачів – так; максимальна кількість одночасних користувачів – до 5000 осіб; мови інтерфейсу – українська; використання шифрування даних під час передачі – так; інтеграція з зовнішніми картографічними сервісами – так; забезпечення роботи системи в режимі 24/7.

4. Зміст текстової частини: вступ, дослідження об'єкта автоматизації, аналіз сучасних тенденцій та викликів у сфері управління збірними вантажоперевезеннями; взаємодія автоматизованої системи управління збірними вантажоперевезеннями з іншими логістичними системами; порівняльний аналіз існуючих автоматизованих систем управління збірними вантажоперевезеннями; вимоги та специфікації для автоматизованої системи управління збірними вантажоперевезеннями; спроектовано архітектуру та реалізовано автоматизовану систему управління збірними вантажоперевезеннями; тестування та оцінка ефективності розробленої системи

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) UML-діаграма компонентів, UML-діаграми розгортання, концептуальна схема бази даних, ER-діаграма структури бази даних, Use Case UML-діаграма, Class UML-діаграма, UML-діаграма послідовності, вигляд екрану розробленого додатку – 11 шт.

1. Консультанти розділів роботи

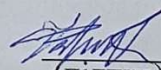
Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1-5	Бісікало О.В., проф. каф. АІТ		

2. Дата видачі завдання “14” жовтня 2024 року

КАЛЕНДАРНИЙ ПЛАН

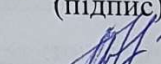
№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Дослідження об'єкта автоматизації та постановка задач дослідження	03.09.24	12.09.24	
2	Аналіз і порівняння існуючих систем управління збірними вантажоперевезеннями	14.09.24	19.09.24	
3	Аналіз функцій систем управління школами та проектування архітектури автоматизованої системи	19.09.24	01.10.24	
4	Проектування автоматизованої системи управління збірними вантажоперевезеннями	01.10.24	15.11.24	
5	Апробація результатів дослідження	11.11.24	12.11.24	
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	15.11.24	29.11.24	
7	Графічні матеріали: - UML-діаграми - ER-діаграма структури бази даних - вигляд екранів розробленого веб додатку	04.10.24 18.10.24 18.11.24	18.10.24 18.10.24 18.11.24	
8	Захист МКР	12.12.24	12.12.24	

Студент


(підпис)

Дмитро ФАРИНЯК

Керівник роботи


(підпис)

Тетяна ГРИЦУК

АНОТАЦІЯ

УДК 004.043

Фариняк Д.Л. Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями. Магістерська кваліфікаційна робота зі спеціальності 174 – автоматизація та комп'ютерно-інтегровані технології та робототехніка, освітня програма – інформаційні системи і інтернет речей. Вінниця: ВНТУ, 2024. 116 с.

На укр. мові. Бібліогр.: 50 назв; рис.:18; табл. 2.

Мета роботи – розробка та впровадження автоматизованої системи управління збірними вантажоперевезеннями, спрямованої на підвищення ефективності логістичних процесів та оптимізацію взаємодії з іншими логістичними платформами.

В теоретичній частині роботи розглянуто основні аспекти автоматизації систем управління збірними вантажоперевезеннями, аналіз поточних методів управління та особливості інтеграції з іншими логістичними платформами. Також проведено дослідження можливостей автоматизації процесів відстеження та контролю, аналіз алгоритмів оптимізації маршрутів та системи управління ризиками.

В практичній частині виконано аналіз та порівняння існуючих систем управління збірними вантажоперевезеннями, проектування автоматизованої системи, включаючи визначення вимог, розробку архітектури системи та бази даних, а також розробку UML-діаграм. Реалізовано функціональність системи та проведено її тестування.

Ключові слова: автоматизована система, управління вантажоперевезеннями, оптимізація маршрутів, UML-діаграми.

ANNOTATION

Farinyak D.L. Development of an automated system for effective management of groupage cargo transportation. Master's qualification work in specialty 174 - automation and computer-integrated technologies and robotics, educational program - information systems and the Internet of Things. Vinnytsia: VNTU, 2024. 116 p.

In Ukrainian. Bibliography: 50 titles; fig.: 18; table. 2.

The aim of the thesis is to develop and implement an automated system for managing consolidated freight transportation, aimed at enhancing the efficiency of logistical processes and optimizing interactions with other logistical platforms.

In the theoretical part of the thesis, the main aspects of automating systems for managing consolidated freight transportation are examined, including an analysis of current management methods and the specifics of integration with other logistical platforms. Additionally, the possibilities for automating tracking and control processes, analyzing route optimization algorithms, and managing risk systems are investigated.

In the practical part, an analysis and comparison of existing systems for managing consolidated freight transportation are conducted, along with the design of the automated system, including defining requirements, developing the system architecture, and designing the database, as well as developing UML diagrams. The functionality of the system has been implemented and tested.

Keywords: automated system, freight transportation management, route optimization, UML diagrams.

ЗМІСТ

ВСТУП	4
1 ДОСЛІДЖЕННЯ ОБ'ЄКТА АВТОМАТИЗАЦІЇ	8
2 ДОСЛІДЖЕННЯ ОСНОВНИХ АСПЕКТІВ АВТОМАТИЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ ЗБІРНИМИ ВАНТАЖОПЕРЕВЕЗЕННЯМИ	12
2.1 Аналіз поточних методів управління збірними вантажоперевезеннями	12
2.2 Особливості інтеграції системи з іншими логістичними платформами	17
2.3 Дослідження можливостей автоматизації процесів відстеження та контролю	22
2.4 Аналіз алгоритмів оптимізації маршрутів	24
2.5 Впровадження системи управління ризиками	27
2.6 Аналіз вимог до інформаційної безпеки системи	28
2.7 Висновки	30
3 АНАЛІЗ І ПОРІВНЯННЯ ІСНУЮЧИХ СИСТЕМ УПРАВЛІННЯ ЗБІРНИМИ ВАНТАЖОПЕРЕВЕЗЕННЯМИ	32
3.1 Загальний огляд систем управління збірними вантажоперевезеннями	32
3.1.1 SAP Logistics Execution System (LES)	32
3.1.2 Oracle Transportation Management (OTM)	36
3.1.3 JDA Transportation Management	40
3.1.4 Manhattan Associates Transportation Management	44
3.2 Порівняльний аналіз ефективності та обмежень кожної системи	47
3.3 Висновки	50

4 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ЗБІРНИМИ ВАНТАЖОПЕРЕВЕЗЕННЯМИ	51
4.1 Визначення вимог до системи	51
4.2 Розробка архітектури системи	52
4.3 Проектування бази даних	54
4.4 Розробка UML-діаграм	59
4.4.1 Use Case UML-діаграма	59
4.4.2 Class UML-діаграма	61
4.4.3 Sequence UML-діаграма	64
4.5 Висновки	65
5 РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ	66
5.1 Вибір технологій розробки	66
5.2 Реалізація функціональності системи	68
5.3 Тестування системи	72
5.4 Висновки	78
ВИСНОВКИ	79
ПЕРЕЛІК ПОСИЛАНЬ	80
ДОДАТКИ	84
Додаток А (Обов'язковий) Протокол перевірки навчальної (кваліфікаційної) роботи.....	85
Додаток Б (Обов'язковий) Технічне завдання.....	86
Додаток В (Вибірковий) Лістинг програмного забезпечення.....	89
Додаток Г (Обов'язковий) Ілюстративна частина	95

ВСТУП

Актуальність. У сучасному світі, де швидкість і ефективність є ключовими факторами успіху в бізнесі, автоматизація процесів набуває особливої важливості. Це стосується і логістики, зокрема управління збірними вантажоперевезеннями, де точність, швидкість обробки інформації та здатність швидко реагувати на зміни є критично важливими. Автоматизовані системи управління вантажоперевезеннями дозволяють значно підвищити ефективність логістичних операцій, оптимізувати маршрутизацію, зменшити витрати та підвищити загальну якість обслуговування клієнтів.

З огляду на зростаючу конкуренцію та вимоги до швидкості доставки, системи управління збірними вантажоперевезеннями стають не просто зручним інструментом, а необхідністю для бізнесів, які прагнуть залишатися конкурентоспроможними на ринку. Ці системи дозволяють автоматизувати багато аспектів логістики, від трекінгу вантажів до управління складськими запасами та оптимізації маршрутів, що в свою чергу забезпечує більш ефективне використання ресурсів.

Важливість автоматизованих систем управління збірними вантажоперевезеннями також зростає у контексті глобалізації та збільшення обсягів міжнародної торгівлі. Вони допомагають компаніям адаптуватися до змінних умов ринку, забезпечуючи можливість швидкої реорганізації логістичних ланцюгів, що є критично важливим у випадках логістичних перебоїв або непередбачених змін у попиті.

Особливу увагу варто звернути на виклики, які постають перед компаніями у зв'язку зі зростанням вимог до екологічності та сталості вантажоперевезень. Сучасні тенденції свідчать про необхідність інтеграції зелених технологій та практик, які можуть зменшити вуглецевий слід та оптимізувати використання ресурсів. Автоматизовані системи управління збірними вантажоперевезеннями мають потенціал не тільки для підвищення ефективності, але й для впровадження більш сталих методів роботи, що є важливим кроком у напрямку екологічної відповідальності [1, 2].

Таким чином, розробка та впровадження ефективних автоматизованих систем управління збірними вантажоперевезеннями є актуальним завданням, що вимагає глибокого аналізу існуючих рішень та розробки нових підходів, здатних задовольнити потреби сучасного ринку. Це дозволить не тільки оптимізувати внутрішні процеси, але й підвищити рівень задоволеності клієнтів, забезпечивши їм більш якісні та швидкі послуги.

Мета і завдання дослідження. Метою магістерської кваліфікаційної роботи є розробка та впровадження автоматизованої системи управління збірними вантажоперевезеннями, спрямованої на підвищення ефективності логістичних процесів, зниження витрат та покращення якості обслуговування клієнтів. Робота має на меті створення інноваційного інструменту, який дозволить компаніям оптимізувати управління вантажопотоками, забезпечити високий рівень інтеграції з іншими системами та підвищити загальну конкурентоспроможність на ринку.

Для досягнення поставленої мети були визначені наступні завдання:

- 1) Провести аналіз сучасних тенденцій та викликів у сфері управління збірними вантажоперевезеннями;
- 2) Дослідити взаємодію автоматизованої системи управління збірними вантажоперевезеннями з іншими логістичними системами;
- 3) Виконати порівняльний аналіз існуючих автоматизованих систем управління збірними вантажоперевезеннями;
- 4) Розробити вимоги та специфікації для автоматизованої системи управління збірними вантажоперевезеннями;
- 5) Спроекувати архітектуру та реалізувати автоматизовану систему управління збірними вантажоперевезеннями;
- 6) Провести тестування та оцінку ефективності розробленої системи.

Об'єктом дослідження є процеси автоматизованого управління збірними вантажоперевезеннями.

Предметом дослідження є методи та засоби програмування, а також алгоритми оптимізації логістичних процесів, які використовуються для розробки

та впровадження автоматизованої системи управління збірними вантажоперевезеннями.

Методи дослідження. У процесі дослідження застосовувалися: теорія алгоритмів, Agile підхід до розробки програмного забезпечення, методика Scrum, методи управління базами даних, методи розробки веб-сайту, засоби безпеки та захисту даних.

Іноваційність.

Спроектовано та розроблено автоматизовану систему управління збірними вантажоперевезеннями, в межах якої:

1. Застосовано підхід АОР (аспектно-орієнтованого програмування), що дозволяє модульно управляти такими перехресними аспектами, як безпека, логування та моніторинг. Це значно спрощує підтримку та розвиток системи, в також сприяє підвищенню якості коду за рахунок зменшення дублювання та покращення структурованості.

2. Досягнуто інтеграції системи з Google Maps, що забезпечує глобальне покриття, актуальність даних та доступ до таких функцій, як побудова маршрутів, оцінка часу в дорозі, аналіз трафіку в реальному часі тощо. Це дозволяє точніше оптимізувати логістичні процеси, знижуючи витрати на транспортування, підвищуючи ефективність використання ресурсів і зменшуючи екологічний вплив.

3. Реалізовано RESTful API, що сприяє простоті масштабування, забезпечило сумісність між різними платформами та полегшило інтеграцію нових функцій завдяки стандартизованим методам взаємодії.

Практичне значення одержаних результатів полягає в тому, що на основі отриманих теоретичних положень виконано розробку та впровадження автоматизованої системи управління збірними вантажоперевезеннями. Це дозволяє компаніям значно підвищити ефективність логістичних процесів, знизити витрати на транспортування та покращити якість обслуговування клієнтів.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням

усіх методів та засобів функціонування інформаційних систем, використанням Agile методики при проведенні розробки та впровадження автоматизованої системи управління збірними вантажоперевезеннями.

Особистий внесок здобувача. Усі результати отримано автором самостійно.

Апробація результатів дослідження: Представлені в роботі результати апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)». За результатами виконаних досліджень підготовлено та опубліковано тези доповіді «Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями» [3].

1 ДОСЛІДЖЕННЯ ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Згідно з темою магістерської кваліфікаційної роботи об'єктом автоматизації є процеси управління збірними вантажоперевезеннями. В рамках дослідження актуальності теми та планування етапів виконання роботи необхідно дослідити особливості логістичних систем управління збірними вантажами, варіанти інтеграції з іншими логістичними та інформаційними системами, методи оптимізації маршрутів доставки, особливості розробки інтерфейсу для систем управління логістикою, типи автоматизованого моніторингу та відслідковування вантажів, аналіз даних про вантажопотоки, реалізацію технічної підтримки та забезпечення безпеки даних у автоматизованій системі.

Аналіз кожного з цих аспектів допоможе створити повне уявлення про кінцевий програмний продукт та його функціональні можливості, розробити на основі цього необхідні UML та ER діаграми, а також визначитися з необхідним програмним забезпеченням та алгоритмами для ефективного управління збірними вантажоперевезеннями [4].

Місце задачі, що розв'язується в роботі, у загальному контексті автоматизації систем управління збірними вантажоперевезеннями є критично важливим для підвищення ефективності логістичних процесів у сучасних умовах глобалізації та зростання обсягів вантажоперевезень. Автоматизована система управління збірними вантажоперевезеннями, яка буде спроектована та розроблена в даній роботі, вирішує ряд ключових задач, зокрема оптимізацію маршрутів доставки, зниження витрат на транспортування та підвищення якості обслуговування клієнтів.

Однією з основних задач є автоматизація процесів планування та управління вантажопотоками, що дозволяє зменшити час доставки та забезпечити більш точне слідування за статусом вантажів. Це особливо важливо для компаній, які прагнуть підвищити свою конкурентоспроможність на ринку.

Крім того, автоматизована система дозволяє клієнтам в режимі онлайн відслідковувати місцезнаходження своїх вантажів, вибирати оптимальні маршрути та часи доставки, а також отримувати повідомлення про будь-які зміни

у статусі вантажу. Це робить процес логістики більш прозорим та зручним для кінцевих споживачів.

Додатково, система сприяє більш ефективному плануванню робочого часу та ресурсів логістичних компаній, зменшуючи простої та непродуктивні перевезення, що в свою чергу веде до зниження витрат та підвищення загальної продуктивності.

Усі ці функції та можливості роблять автоматизовану систему управління збірними вантажоперевезеннями невід'ємною складовою сучасної логістичної інфраструктури, забезпечуючи ефективність, зручність та надійність у процесах доставки товарів.

Область застосування розробки

Розроблена автоматизована система управління збірними вантажоперевезеннями має широкий спектр можливих застосувань. Перш за все, вона стане невід'ємною складовою для будь-якої логістичної компанії, незалежно від її масштабу та спеціалізації. Така система може бути успішно впроваджена в транспортних компаніях, складських комплексах, дистриб'юторських центрах, а також у виробничих підприємствах, які потребують ефективного управління логістикою.

Крім того, автоматизована система може знайти застосування в інших галузях, де важливо ефективно керувати потоками товарів та забезпечувати їх своєчасну доставку. Наприклад, в роздрібній торгівлі, електронній комерції, автомобільній промисловості та будь-яких інших секторах, де критично важливою є логістика та управління запасами.

Ця універсальність та гнучкість системи дозволяють застосовувати її в різних галузях і сприяють вдосконаленню організації робочих процесів та покращенню обслуговування клієнтів у всіх сферах діяльності, де важлива ефективна управління логістичними процесами та оптимізація вантажоперевезень.

Вимоги до програмно-технічного забезпечення, що розробляється в магістерській кваліфікаційній роботі, можна описати наступним чином:

- Локальна база даних: Для зберігання та управління інформацією про вантажі, маршрути та логістичні процеси використовується реляційна база даних PostgreSQL. Ця система вибрана через її високу надійність, консистентність даних та підтримку складних транзакцій, що є критично важливим для логістичних операцій [4].
- Веб-розробка: Розробка веб-сайту з автоматизованою системою управління збірними вантажоперевезеннями використовує сучасні технології. Фронтенд створюється з використанням HTML, CSS та JavaScript, а також фреймворку React для розробки інтуїтивно зрозумілих інтерфейсів. Бекенд розробляється на Java з використанням Spring Framework, що забезпечує високу продуктивність та ефективне управління бізнес-логікою.
- Серверні рішення: Для розгортання та управління веб-сервером, базою даних та контейнеризації використовується Docker. Це дозволяє ефективно створювати, тестувати та розгортати контейнери, забезпечуючи консистентність середовища в усіх етапах розробки та впровадження.
- Засоби безпеки: Забезпечення безпеки даних та системи є пріоритетом. Використовуються сучасні методи шифрування та аутентифікації, включаючи OAuth та SSL/TLS, для захисту інформації на рівні веб-застосунку та бази даних. Docker додатково допомагає ізолювати та захищати компоненти системи, підвищуючи загальний рівень безпеки.

Функції розробки:

- забезпечення доступу користувачів до системи та персональних акаунтів;
- управління маршрутами та вантажоперевезеннями;
- моніторинг та відслідковування статусу вантажів;
- аналіз ефективності логістичних процесів.

Для виконання роботи необхідні наступні вхідні дані:

- локальна база даних, яку описує ER-діаграма;

- відповідно налаштовані програмні та технічні засоби, що необхідні для розробки та розгортання автоматизованої системи управління збірними вантажоперевезеннями.

Для перевірки розробки та розгортання автоматизованої системи управління збірними вантажоперевезеннями необхідними будуть результати роботи готового продукту, правильна взаємодія його процесів та команд, відображення вихідної інформації.

2 ДОСЛІДЖЕННЯ ОСНОВНИХ АСПЕКТІВ АВТОМАТИЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ ЗБІРНИМИ ВАНТАЖОПЕРЕВЕЗЕННЯМИ

2.1 Аналіз поточних методів управління збірними вантажоперевезеннями

Збірні вантажоперевезення – це специфічний вид логістичних послуг, який передбачає транспортування невеликих партій товарів від різних відправників у спільному транспортному засобі до різних отримувачів. Цей метод є вигідним рішенням для компаній, які не мають достатньої кількості товару для заповнення цілого вантажного контейнера або автомобіля, оскільки дозволяє зменшити витрати на доставку, оптимізувавши використання вантажного простору.

Основна ідея збірних вантажоперевезень полягає в ефективному об'єднанні декількох малих вантажів у один великий для досягнення максимальної ефективності транспортних засобів. Це вимагає від логістичних компаній високого рівня координації, планування та управління, щоб забезпечити, що всі вантажі будуть доставлені вчасно та в непошкодженому стані.

Збірні вантажоперевезення також включають в себе складні процеси збору, сортування, зберігання та подальшої відправки товарів. Ці процеси вимагають від логістичних операторів використання спеціалізованих складських терміналів та транспортних хабів, де товари можуть бути агреговані та розподілені за маршрутами доставки.

Завдяки своїй економічності та гнучкості, збірні вантажоперевезення є популярним вибором серед бізнесів різного масштабу, особливо в умовах глобалізації та зростаючої взаємозалежності ринків [6, 7].

Традиційні методи управління логістикою збірних вантажоперевезень базуються на ручному плануванні та контролі. Логісти використовують паперові документи, телефонні дзвінки та факси для координації відправлень, що може бути часомістким і схильним до помилок. Цей підхід часто вимагає значних зусиль для управління та координації між різними відділами та партнерами, що може призводити до затримок у доставці та підвищених витрат.

Перевагами таких методів можна назвати:

- простота використання без необхідності складних технологічних рішень;
- можливість гнучкого реагування на непередбачені обставини завдяки безпосередньому людському контакту.

В той час як існують і певні недоліки традиційних методів:

- висока ймовірність помилок через людський фактор;
- низька ефективність управління великими обсягами даних;
- труднощі у координації та відстеженні вантажів у реальному часі.

З розвитком технологій, багато компаній перейшли до використання інформаційних систем для управління логістикою. Ці системи, як-от ERP (Enterprise Resource Planning), TMS (Transportation Management System) та WMS (Warehouse Management System), дозволяють автоматизувати багато аспектів логістичних операцій. Вони забезпечують централізоване управління даними, що покращує видимість всього ланцюга постачання, оптимізує маршрутизацію, планування вантажів та управління складськими запасами.

Переваги використання ІТ у логістиці:

- підвищення точності та прозорості управління ланцюгом постачання;
- автоматизація рутинних процесів знижує витрати та покращує ефективність;
- здатність аналізувати великі обсяги даних для оптимізації логістичних операцій.

Недоліки використання ІТ у логістиці:

- висока вартість впровадження та підтримки систем;
- потреба у навчанні персоналу для ефективного використання нових систем;
- ризик залежності від технологічних рішень та потенційних збоїв у їх роботі.

Загалом, вибір між традиційними методами та сучасними ІТ-рішеннями залежить від специфіки компанії, її потреб, обсягу операцій та готовності інвестувати в технологічний розвиток [8].

Аналіз інструментів та технологій, що використовуються у галузі:

– Системи планування ресурсів підприємства (ERP)

ERP системи інтегрують різноманітні бізнес-процеси, включаючи інвентаризацію, замовлення, бухгалтерський облік, людські ресурси, клієнтське обслуговування та більше. У контексті збірних вантажоперевезень, ERP допомагає управляти всіма аспектами логістики, забезпечуючи централізоване управління даними, що сприяє підвищенню ефективності та зниженню витрат. Ці системи дозволяють автоматизувати планування, відслідковування та аналіз вантажоперевезень, що забезпечує оптимізацію маршрутів та ресурсів.

– Системи управління складом (WMS)

WMS спеціалізуються на оптимізації процесів, пов'язаних із складським господарством. Вони допомагають управляти запасами, автоматизувати підбір та упаковку товарів, а також забезпечувати точність складських операцій. У контексті збірних вантажоперевезень, WMS можуть значно підвищити швидкість та точність обробки замовлень, знижуючи час на складські операції та мінімізуючи помилки, що виникають при ручному введенні даних.

– Транспортні управлінські системи (TMS)

TMS дозволяють ефективно управляти всіма аспектами транспортування вантажів. Вони включають функції для планування маршрутів, відслідковування вантажів у реальному часі, управління тарифами та витратами на доставку. Такі системи сприяють зниженню витрат на транспортування, оптимізації використання транспортних засобів та покращенню сервісу завдяки своєчасній доставці товарів.

– Системи управління відносинами з клієнтами (CRM)

CRM системи зосереджені на управлінні взаємодіями з клієнтами, від маркетингу до продажу та сервісного обслуговування. У логістиці, CRM може використовуватися для збору та аналізу даних про клієнтів, що дозволяє покращувати якість обслуговування та розробляти персоналізовані пропозиції. Це сприяє підвищенню лояльності клієнтів та оптимізації взаємодій з ними, що є критично важливим у конкурентному середовищі збірних вантажоперевезень.

Кожен з цих інструментів та технологій відіграє ключову роль у сучасній логістиці, дозволяючи компаніям досягати вищої ефективності, знижувати витрати та покращувати якість обслуговування клієнтів [9, 10]. Їх взаємодію в сфері управління вантажними перевезеннями зображено на рисунку 2.1.

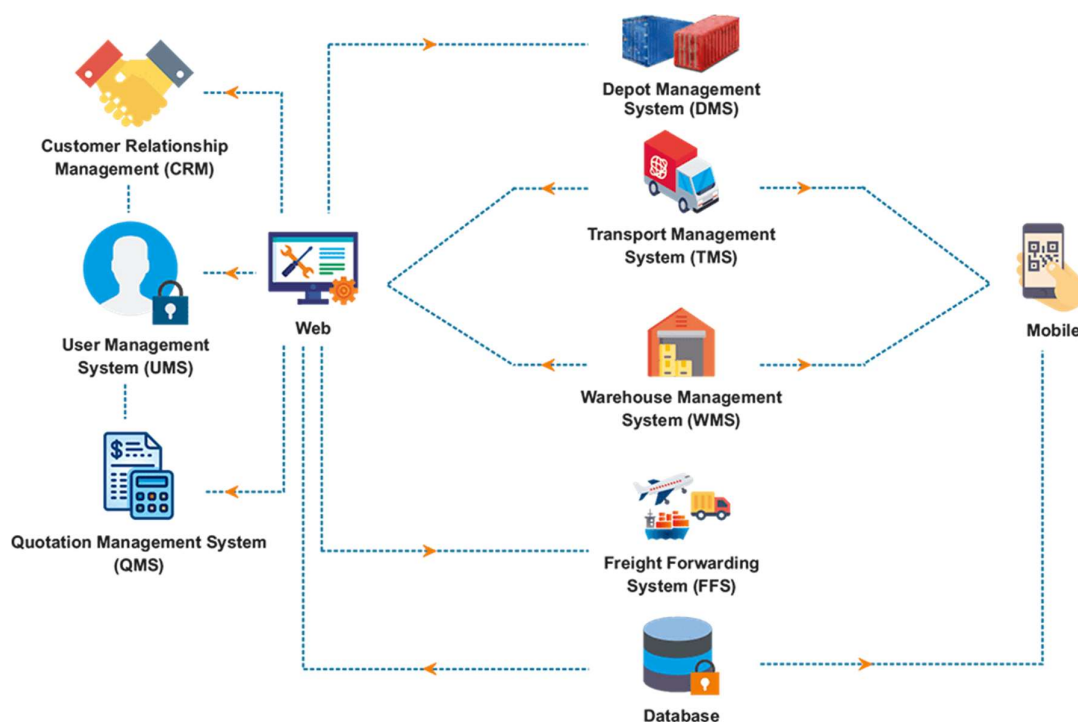


Рисунок 2.1 – Взаємодія описаних компонентів в сфері управління вантажними перевезеннями[11]

Оцінку ефективності поточних методів можна провести за такими показниками як аналіз часу та витрат, вплив на задоволеність клієнтів та вплив на екологічність та стійкість логістичних операцій.

– Аналіз часу та витрат, пов'язаних з різними методами

Для оцінки ефективності поточних методів управління збірними вантажоперевезеннями важливо аналізувати час та витрати, які вони займають. Традиційні методи, які включають багато ручної роботи та паперової документації, часто виявляються більш часомісткими та дорогими у порівнянні з автоматизованими системами. Впровадження ІТ-рішень, таких як ERP або TMS,

може значно скоротити час обробки замовлень та оптимізувати витрати на логістику за рахунок ефективнішого планування та управління ресурсами.

– Вплив на задоволеність клієнтів

Задоволеність клієнтів є ключовим показником успішності будь-якої логістичної операції. Сучасні методи управління, що включають використання CRM систем, дозволяють підвищити рівень обслуговування клієнтів шляхом персоналізації та оперативного реагування на їхні запити. Це, в свою чергу, може покращити відносини з клієнтами та збільшити їх лояльність.

– Вплив на екологічність та стійкість логістичних операцій

Екологічність та стійкість стають все більш важливими у сучасному світі. Використання автоматизованих систем може допомогти зменшити викиди вуглецю за рахунок оптимізації маршрутів доставки та зменшення кількості порожніх поїздок. Також, ефективне управління складом може знизити кількість відходів та покращити використання ресурсів.

Методи управління збірними вантажоперевезеннями можуть значно відрізнятися в залежності від країни, що обумовлено різними економічними, регуляторними та культурними факторами. Наприклад, у США та Європі великий акцент робиться на автоматизацію та використання передових технологій, таких як TMS та інтегровані ERP системи, для оптимізації логістичних процесів. В той час як у країнах, що розвиваються, може спостерігатися більша залежність від традиційних методів через обмежені технологічні ресурси.

У країнах Азії, таких як Китай та Японія, велика увага приділяється інноваціям у логістиці, включаючи використання штучного інтелекту та машинного навчання для прогнозування попиту та оптимізації доставки. Це дозволяє компаніям ефективно масштабувати свої операції та забезпечувати високий рівень задоволеності клієнтів [12].

Міжнародні норми та стандарти відіграють важливу роль у формуванні логістичних процесів в Україні, особливо в контексті інтеграції з європейськими та глобальними ринками. Стандарти ISO, наприклад, впливають на управління

якістю та безпекою в логістиці, забезпечуючи, що українські компанії відповідають міжнародним вимогам.

Крім того, регуляції ЄС щодо транспортування та екологічні стандарти вимагають від українських компаній впровадження більш стійких практик. Це включає використання екологічно чистих транспортних засобів, оптимізацію маршрутів для зниження викидів CO₂, та впровадження систем управління відходами.

Вплив міжнародних стандартів також спонукає українські компанії до впровадження сучасних IT-рішень для вдосконалення своїх логістичних операцій. Це стосується не тільки технологічних інновацій, але й удосконалення процесів управління клієнтськими відносинами, складського обліку та транспортного менеджменту [13].

Таким чином, міжнародний досвід та стандарти відіграють ключову роль у формуванні логістичних стратегій в Україні, сприяючи її інтеграції в глобальні ланцюги постачання та підвищенню конкурентоспроможності українських компаній на міжнародному рівні.

2.2 Особливості інтеграції системи з іншими логістичними платформами

Інтеграція різних логістичних систем вимагає використання стандартизованих технологічних протоколів, які забезпечують безперешкодне та ефективне спілкування між системами. Нижче розглянемо ключові технологічні стандарти, які використовуються в індустрії для досягнення цієї мети.

– API (Application Programming Interface)

API є одним з найбільш популярних методів для інтеграції різноманітних програмних рішень. API дозволяє двом або більше програмам "розмовляти" одна з одною, обмінюючись даними та функціональними можливостями в реальному часі. В логістиці, API може бути використаний для інтеграції систем управління складом (WMS) з транспортними управлінськими системами (TMS), CRM

системами, а також з іншими зовнішніми сервісами, такими як системи слідкування за вантажами, погодні сервіси та ін.

- EDI (Electronic Data Interchange)

EDI – це стандарт обміну даними, який дозволяє передавати документи між різними компаніями в стандартизованому електронному форматі. Це особливо корисно в логістиці для обміну інформацією, такою як замовлення на покупку, інвойси, відомості про вантаж та інші логістичні документи. EDI знижує ризик помилок, пов'язаних з ручним введенням даних, та підвищує швидкість обробки транзакцій [14, 15].

Інші протоколи

- SOAP (Simple Object Access Protocol): Протокол, який дозволяє програмам обмінюватися інформацією через мережу. SOAP може використовуватися для виконання операцій на серверах, що розташовані в інших місцях.
- REST (Representational State Transfer): Архітектурний стиль, який використовується для створення веб-сервісів. REST використовує стандартні HTTP методи, такі як GET, POST, PUT та DELETE, що робить його простим у використанні та інтеграції.
- JSON (JavaScript Object Notation) та XML (eXtensible Markup Language): Формати даних, які використовуються для структурування інформації, що передається між системами через API, SOAP, або REST.

Використання цих технологічних стандартів дозволяє компаніям легко інтегрувати різні логістичні системи, забезпечуючи плавність та ефективність логістичних операцій. Це, в свою чергу, сприяє зниженню витрат, підвищенню продуктивності та покращенню обслуговування клієнтів.

Інтеграція різних логістичних платформ може принести компаніям значні переваги, які впливають на всі аспекти їхньої діяльності, насамперед це: зниження витрат, підвищення точності даних, покращення обслуговування клієнтів та збільшення загальної ефективності операцій.

- Зниження витрат

Інтеграція систем може значно знизити операційні витрати. Автоматизація процесів, таких як введення даних, обробка замовлень та управління запасами, зменшує потребу в ручній праці та знижує ймовірність помилок, що може призвести до додаткових витрат. Крім того, оптимізація маршрутів доставки та покращення управління запасами можуть знизити витрати на транспортування та зберігання.

- Підвищення точності даних

Інтеграція систем забезпечує централізацію та стандартизацію даних, що підвищує їх точність та надійність. Це важливо для прийняття обґрунтованих управлінських рішень та планування. Точні дані допомагають краще розуміти потреби клієнтів, прогнозувати попит та оптимізувати запаси, що в кінцевому підсумку сприяє збільшенню продажів та прибутку.

- Покращення обслуговування клієнтів

Інтегровані системи дозволяють компаніям надавати вищий рівень обслуговування клієнтів. Завдяки централізованій інформації, співробітники можуть швидше реагувати на запити клієнтів, точніше відстежувати статус замовлень та ефективніше вирішувати проблеми. Також, здатність надавати клієнтам доступ до інформації про статус їхніх замовлень в реальному часі підвищує їхнє задоволення та лояльність.

- Збільшення загальної ефективності операцій

Інтеграція логістичних платформ сприяє збільшенню загальної ефективності операцій. Це досягається через автоматизацію процесів, покращення координації між різними відділами та зменшення часу на виконання задач. Оптимізація логістичних процесів дозволяє швидше обробляти замовлення, знижувати час доставки та підвищувати загальну продуктивність.

Завдяки цим перевагам, інтеграція логістичних систем є важливим стратегічним кроком для компаній, які прагнуть оптимізувати свої операції, знизити витрати та підвищити конкурентоспроможність на ринку [16].

Проте поруч із перевагами завжди необхідно розглядати можливі виклики та бар'єри, а саме:

- Несумісність технологій

Одним з найбільших викликів при інтеграції логістичних систем є несумісність технологій. Різні системи можуть використовувати різні формати даних, протоколи комунікації або архітектурні рішення, що ускладнює їхню інтеграцію. Це може вимагати додаткових зусиль та ресурсів для розробки адаптерів або мідлварів, які дозволять системам "розуміти" одна одну.

- Високі витрати на інтеграцію

Інтеграція систем може бути дорогою, особливо якщо вона вимагає значних технічних змін або найму зовнішніх консультантів. Витрати можуть включати не тільки безпосередньо технологічні аспекти, але й навчання персоналу, тестування системи та вирішення проблем після запуску. Для деяких компаній ці витрати можуть бути занадто високими, що змушує їх відкладати або відмовлятися від інтеграції.

- Проблеми з безпекою даних

Інтеграція систем збільшує ризики, пов'язані з безпекою даних. Обмін даними між різними системами може створювати потенційні вразливості, через які злоумисники можуть спробувати отримати доступ до конфіденційної інформації. Компанії повинні забезпечити, що всі системи належним чином захищені, а дані шифруються під час передачі та зберігання.

- Опір змінам

Інтеграція нових систем може зустрічати опір з боку співробітників, які звикли до існуючих процесів і систем. Зміна може викликати занепокоєння або невпевненість, особливо якщо співробітники не відчують, що вони достатньо підготовлені до використання нових систем. Ефективне навчання та комунікація є ключовими для забезпечення гладкого переходу.

- Технічні проблеми під час інтеграції

Навіть з добре спланованим процесом інтеграції, технічні проблеми можуть виникати несподівано. Це може включати помилки у програмному забезпеченні, проблеми з сумісністю апаратного забезпечення, або непередбачені збої в системах. Швидке вирішення цих проблем є важливим для мінімізації збоїв у роботі [17-19].

Враховуючи ці виклики, компанії повинні ретельно планувати процес інтеграції, враховувати потенційні ризики та розробляти стратегії їх подолання. Залучення досвідчених ІТ-спеціалістів та проведення ретельного тестування можуть допомогти забезпечити успішну інтеграцію систем.

Інтеграція логістичних систем продовжує еволюціонувати, а майбутні тенденції та інновації обіцяють змінити спосіб, яким компанії управляють своїми логістичними операціями. Однією з ключових тенденцій є використання штучного інтелекту (ШІ) та машинного навчання, які відіграють все більш важливу роль у автоматизації та оптимізації інтеграційних процесів.

Штучний інтелект та машинне навчання дозволяють системам не тільки автоматично обробляти великі обсяги даних, але й аналізувати їх для виявлення тенденцій, прогнозування попиту та оптимізації логістичних маршрутів. Це може включати автоматичне планування доставок на основі прогнозованих патернів попиту, а також миттєве реагування на зміни в логістичному ланцюгу, такі як затримки в доставці або зміни в запасах.

Крім того, інтеграція ШІ може допомогти підвищити точність та ефективність управління запасами. Системи, здатні аналізувати історичні дані та враховувати зовнішні чинники, такі як сезонні коливання або економічні умови, можуть точніше прогнозувати потреби в запасах та автоматично регулювати замовлення, знижуючи ризик надлишків або дефіциту товарів.

Іншою важливою тенденцією є зростання інтеграції Інтернету речей (IoT) у логістичні системи. IoT-пристрої, такі як датчики та трекери, можуть надсилати реальні дані про стан вантажів, їхнє місцезнаходження та умови зберігання. Ці дані можуть бути інтегровані в логістичні системи для забезпечення більш точного та своєчасного моніторингу вантажів, що підвищує якість обслуговування клієнтів та знижує ризики, пов'язані з втратою або пошкодженням товарів.

Також важливо відзначити роль блокчейн технологій, які можуть революціонізувати інтеграцію логістичних систем, забезпечуючи високий рівень прозорості та безпеки в обміні даними. Блокчейн може допомогти створити надійну та незмінну базу даних транзакцій, доступну всім учасникам ланцюга

поставок, що значно спрощує процеси верифікації та зменшує можливість шахрайства.

Ці інновації та тенденції відкривають нові можливості для компаній у сфері логістики, дозволяючи їм підвищити ефективність, знизити витрати та покращити якість обслуговування клієнтів. Впровадження цих технологій вимагає значних інвестицій та стратегічного планування, але потенційні вигоди роблять це вкрай важливим для майбутнього логістики.

2.3 Дослідження можливостей автоматизації процесів відстеження та контролю

Сучасний світ логістики не може існувати без ефективних технологій відстеження, які дозволяють компаніям моніторити та управляти своїми вантажами в реальному часі. Розвиток технологій значно трансформував цей процес, зробивши його більш автоматизованим та точним.

GPS-трекінг є одним з найбільш відомих та широко використовуваних методів відстеження. GPS-пристрої встановлюються на транспортні засоби або контейнери, що дозволяє логістичним компаніям отримувати точні дані про місцезнаходження вантажу в будь-який час. Це не тільки сприяє кращому плануванню маршрутів, але й допомагає уникнути затримок та інших проблем під час транспортування [20].

RFID-технології відіграють ключову роль у відстеженні продуктів на складах та в роздрібній торгівлі. RFID-мітки вмонтовані в товари або упаковки, що дозволяє автоматично ідентифікувати та відстежувати кожен елемент через радіочастотну ідентифікацію. Це значно підвищує ефективність складських операцій та знижує можливість помилок або втрати товарів.

IoT-датчики стають все більш популярними у логістиці, оскільки вони можуть надавати детальну інформацію не тільки про місцезнаходження, але й про стан вантажу. Ці датчики можуть вимірювати різноманітні параметри, такі як температура, вологість, вібрація тощо, що є критично важливим для

транспортування чутливих товарів, наприклад, продуктів харчування або фармацевтичних препаратів.

Крім того, розвиток мобільних та хмарних технологій також впливає на процеси відстеження. Мобільні додатки дозволяють логістичним менеджерам та клієнтам відстежувати статус вантажу в реальному часі, отримувати повідомлення про зміни статусу, а також управляти логістичними задачами на ходу. Хмарні рішення, в свою чергу, забезпечують централізоване зберігання даних та легкий доступ до інформації з будь-якого пристрою, що підключений до інтернету.

Ці технології не тільки оптимізують процеси відстеження, але й сприяють автоматизації процесів відстеження, що є критично важливим для підвищення ефективності логістичних операцій. Автоматизація процесів відстеження включає в себе декілька ключових аспектів, які разом формують більш інтегровану та ефективну логістичну систему.

Перш за все, інтеграція систем відстеження з іншими логістичними системами є фундаментальним кроком. Це дозволяє створити єдину інформаційну платформу, де дані з різних джерел збираються та аналізуються в єдиному інтерфейсі. Наприклад, інтеграція системи GPS-трекінгу з системою управління складом (WMS) може дозволити автоматично оновлювати статус запасів на основі інформації про місцезнаходження вантажів. Така інтеграція сприяє кращому плануванню та оптимізації запасів, знижуючи ризики пов'язані з перевантаженням або нестачею товарів.

Далі, автоматизація збору даних значно знижує людський фактор та підвищує точність інформації. Сучасні технології, такі як RFID та IoT-датчики, дозволяють збирати дані автоматично, без необхідності ручного введення. Це не тільки підвищує ефективність процесів, але й забезпечує миттєве оновлення даних у системі, що є важливим для оперативного реагування на зміни у логістичних ланцюгах.

Обробка та аналіз інформації є наступним кроком у процесі автоматизації. Завдяки застосуванню алгоритмів машинного навчання та штучного інтелекту, системи можуть не тільки збирати дані, але й аналізувати їх для виявлення

тенденцій, прогнозування потенційних проблем та автоматичного прийняття рішень. Наприклад, система може автоматично розраховувати оптимальні маршрути доставки на основі аналізу трафіку та погодних умов, а також враховувати історичні дані про затримки або проблеми на певних ділянках маршруту [21].

Таким чином, автоматизація процесів відстеження не тільки спрощує управління логістичними операціями, але й відкриває нові можливості для підвищення ефективності, зниження витрат та покращення обслуговування клієнтів. Це стає можливим завдяки інтеграції різноманітних технологій та їх взаємодії в рамках єдиної логістичної системи.

2.4 Аналіз алгоритмів оптимізації маршрутів

Оптимізація маршрутів є ключовою задачею в логістиці, яка впливає на ефективність доставки та загальні витрати компанії. Основні виклики, з якими стикаються логістичні компанії при плануванні маршрутів, включають:

- Зниження витрат на паливо: Оптимізація маршрутів може значно знизити кілометраж та час у дорозі, що прямо впливає на витрати палива.
- Зменшення часу доставки: Ефективне планування маршрутів дозволяє скоротити час доставки, що підвищує задоволеність клієнтів та дозволяє швидше обертати запаси.
- Балансування навантаження між транспортними засобами: Рівномірне розподілення завантаження між транспортними засобами допомагає уникнути перевантаження окремих одиниць та збільшує їхній термін служби.
- Врахування часових вікон доставки: Багато клієнтів вимагають доставки у строго визначений час, що вимагає точного планування маршрутів.

- Дотримання інших обмежень: Це може включати географічні обмеження, правила дорожнього руху, обмеження на вагу та розмір вантажів тощо.

Огляд існуючих алгоритмів

Для вирішення цих викликів існує багато алгоритмів оптимізації маршрутів. Ось декілька з найбільш відомих:

- Алгоритм найближчого сусіда: Це простий жадібний алгоритм, який в кожному кроці вибирає найближчу невіддану точку до поточної. Хоча цей алгоритм швидкий, він не завжди забезпечує оптимальне рішення.
- Алгоритми генетичного програмування: Ці алгоритми використовують принципи природного відбору, такі як мутація та кросовер, для генерації оптимальних маршрутів. Вони добре підходять для складних логістичних задач, де потрібно враховувати багато різних факторів і обмежень.
- Алгоритми колонії мурах (Ant Colony Algorithms): Ці алгоритми імітують поведінку мурах, які шукають шляхи до їжі. Вони ефективні для знаходження оптимальних маршрутів у складних умовах і можуть адаптуватися до змін у даних або умовах завдання.
- Інші евристичні та метаевристичні методи: Це можуть бути симульоване відпалення, табу-пошук, ітеративний пошук найкращого рішення та інші методи, які допомагають знаходити близькі до оптимальних рішення в складних логістичних задачах [22].

Кожен з цих алгоритмів має свої переваги та недоліки, і вибір конкретного алгоритму залежить від специфіки задачі, доступних ресурсів та вимог до точності та швидкості обробки.

Моделювання та тестування алгоритмів оптимізації маршрутів дозволяють ідентифікувати потенційні проблеми, оцінити ефективність алгоритмів та внести необхідні корективи перед їх реальним впровадженням. Ось декілька методик та інструментів, які використовуються для моделювання та тестування:

Методики моделювання:

- Сценарійне моделювання: Створення різних сценаріїв з різними умовами (наприклад, пікові години, різні погодні умови), щоб оцінити, як алгоритм впорається з різними викликами.
- Моделювання на основі історичних даних: Використання даних про попередні маршрути та їх ефективність для тестування алгоритмів на реальних прикладах.
- Моделювання з використанням штучно створених даних: Генерація даних для тестування алгоритмів у контрольованих умовах, що дозволяє оцінити їхню роботу в ідеальних або екстремальних умовах.

Інструменти для моделювання та тестування:

- MATLAB: Потужне середовище для числових обчислень, яке має вбудовані інструменти для оптимізації та моделювання. MATLAB дозволяє легко візуалізувати маршрути та аналізувати результати оптимізації.
- Python: Завдяки бібліотекам, таким як NumPy, Pandas для обробки даних, і SciPy, NetworkX для оптимізації та аналізу графів, Python є відмінним вибором для розробки та тестування алгоритмів оптимізації маршрутів.
- Спеціалізоване ПЗ для логістичних задач: Програми, такі як Route4Me, OR-Tools від Google, які спеціально розроблені для рішення задач оптимізації маршрутів і можуть використовуватися для тестування та впровадження різних логістичних стратегій.

В якості прикладів використання цих інструментів можна назвати наступні:

- MATLAB: Може бути використаний для симуляції доставки в міських умовах, моделюючи різні маршрути та аналізуючи їх на предмет мінімізації часу та витрат.
- Python: Розробка кастомізованого алгоритму з використанням бібліотеки NetworkX для аналізу мережі доріг та визначення оптимальних маршрутів на основі реальних даних про трафік.

- Google OR-Tools: Використання для розробки мультимодальних маршрутів доставки, що включають автомобільний, залізничний та морський транспорт.

Моделювання та тестування алгоритмів оптимізації маршрутів дозволяють не тільки перевірити їх ефективність, але й адаптувати під конкретні умови та вимоги, що є ключовим для успішної логістичної діяльності [23, 24].

2.5 Впровадження системи управління ризиками

Впровадження системи управління ризиками є критично важливим для будь-якої компанії, що прагне мінімізувати потенційні втрати та оптимізувати свої операційні процеси. Система управління ризиками дозволяє компанії ідентифікувати, аналізувати, моніторити та контролювати потенційні ризики, забезпечуючи тим самим стабільність та надійність бізнесу. Ось основні кроки для впровадження ефективної системи управління ризиками:

1. Ідентифікація ризиків

Перший крок полягає у визначенні потенційних ризиків, які можуть вплинути на бізнес. Це можуть бути фінансові ризики, ризики безпеки, технологічні ризики, ризики невиконання контрактів, екологічні ризики та інші. Використання SWOT-аналізу (сильні та слабкі сторони, можливості та загрози) може допомогти у цьому процесі.

2. Оцінка ризиків

Після ідентифікації ризиків наступним кроком є їх оцінка. Це включає аналіз ймовірності виникнення ризиків та потенційного впливу на компанію. Ризики можуть бути класифіковані за рівнем важливості, що допоможе пріоритизувати зусилля щодо їх мінімізації.

3. Розробка стратегій управління ризиками

На основі оцінки ризиків розробляються стратегії їх мінімізації або усунення. Стратегії можуть включати уникнення ризику, зменшення ризику через контрольні заходи, передачу ризику (наприклад, через страхування) та прийняття ризику, якщо він виправдовує потенційні вигоди.

4. Впровадження стратегій

Цей крок включає реалізацію обраних стратегій управління ризиками. Це може включати введення нових політик, процедур, технологій або навчання персоналу. Важливо забезпечити, щоб всі зацікавлені сторони були належним чином інформовані та залучені в процес управління ризиками.

5. Моніторинг та перегляд

Управління ризиками є неперервним процесом. Регулярний моніторинг та перегляд системи управління ризиками дозволяють виявляти нові ризики, оцінювати ефективність впроваджених заходів та вносити корективи за необхідності. Це може включати щорічні перегляди, аудити та оновлення стратегій управління ризиками.

Приклади інструментів для управління ризиками:

- Програмне забезпечення для управління ризиками, таке як Riskalyze, LogicManager або RiskLens, яке допомагає автоматизувати процеси ідентифікації, оцінки та моніторингу ризиків.
- Інструменти для аналізу даних, такі як Tableau або Microsoft Power BI, які можуть використовуватися для візуалізації ризиків та їх впливу на бізнес [25, 26].

Впровадження ефективної системи управління ризиками дозволяє компанії не тільки захистити себе від потенційних загроз, але й оптимізувати свої ресурси, підвищити стабільність та покращити загальну стратегічну позицію на ринку.

2.6 Аналіз вимог до інформаційної безпеки системи

Аналіз вимог до інформаційної безпеки системи є фундаментальним етапом у забезпеченні захисту даних та інформаційних активів компанії. Цей процес включає ідентифікацію потенційних загроз, визначення вразливостей та розробку стратегій для їхнього усунення або мінімізації.

Ключові кроки, які слід врахувати при аналізі вимог до інформаційної безпеки зображено на рисунку 2.2.

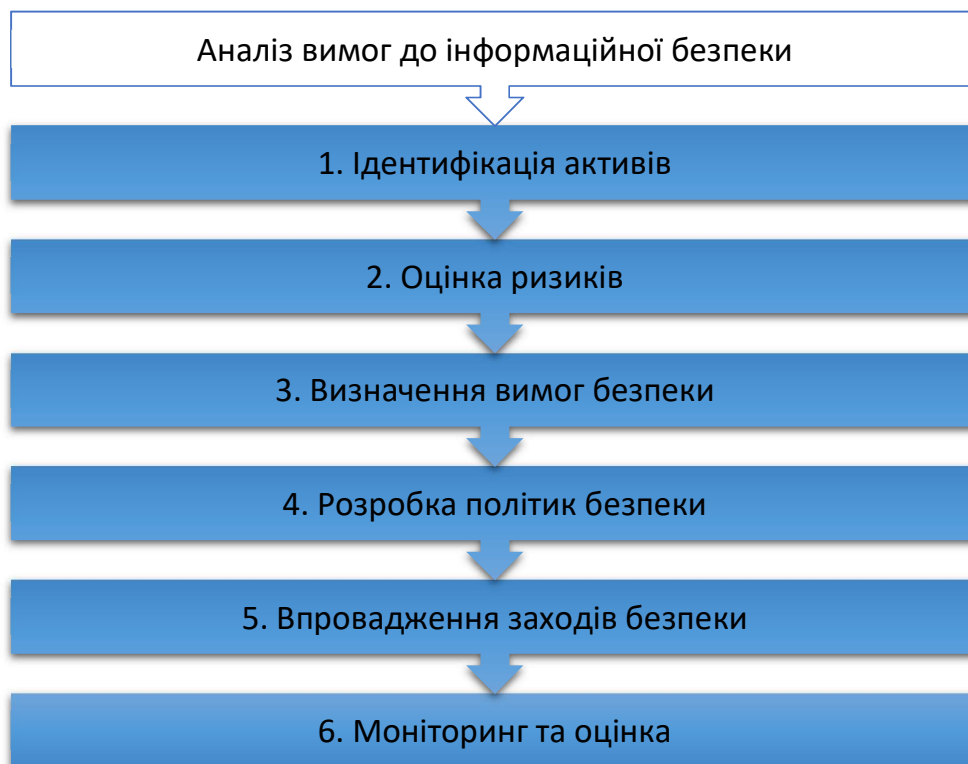


Рисунок 2.2 – Ключові кроки, які слід врахувати при аналізі вимог до інформаційної безпеки

1. Ідентифікація активів

Перший крок полягає у визначенні всіх інформаційних активів, які потребують захисту. Це можуть бути бази даних, документи, електронні записи, інтелектуальна власність, персональні дані клієнтів та співробітників тощо.

2. Оцінка ризиків

Другий крок включає оцінку ризиків, асоційованих з кожним активом. Це включає аналіз потенційних загроз (наприклад, кібератаки, фізичне пошкодження, втрата даних) та визначення вразливостей, які можуть бути використані для зловмисних дій.

3. Визначення вимог безпеки

На основі оцінки ризиків розробляються вимоги до інформаційної безпеки. Це може включати вимоги до шифрування даних, аутентифікації користувачів, контролю доступу, захисту від вірусів та шкідливого програмного забезпечення, а також фізичного захисту інформаційних систем.

4. Розробка політик безпеки

На основі визначених вимог формулюються політики безпеки, які встановлюють правила та процедури для захисту інформаційних активів. Політики безпеки повинні бути чітко документовані та доступні всім зацікавленим сторонам.

5. Впровадження заходів безпеки

Цей крок включає реалізацію технічних, адміністративних та фізичних заходів для виконання політик безпеки. Це може включати встановлення брандмауерів, антивірусного програмного забезпечення, систем відеоспостереження, проведення навчань для співробітників тощо.

6. Моніторинг та оцінка

Останній крок полягає у постійному моніторингу та оцінці ефективності впроваджених заходів безпеки. Це включає регулярні аудити, тестування на проникнення та оновлення систем безпеки для відповідності сучасним загрозам.

Інструменти та технології:

- Шифрування: Використання технологій шифрування для захисту даних від несанкціонованого доступу.
- Брандмауери та антивірусне програмне забезпечення: Захист мережі та комп'ютерів від зловмисних атак та програм.
- Системи управління ідентифікацією та доступом (IAM): Контроль доступу до ресурсів на основі ролей користувачів.
- Фізичний захист: Заходи, такі як контроль доступу в будівлю, відеоспостереження, захист від несанкціонованого фізичного доступу.

Аналіз вимог до інформаційної безпеки є невід'ємною частиною стратегії захисту будь-якої організації, що дозволяє не тільки захистити важливі дані, але й забезпечити довіру клієнтів та партнерів [27].

2.7 Висновки

Завершивши дослідження основних аспектів автоматизації системи управління збірними вантажоперевезеннями, можна перейти до наступного важливого етапу дослідження. Враховуючи зростаючу складність та важливість

логістичних операцій у сучасному бізнес-середовищі, особливо актуальним стає питання ефективності управління збірними вантажоперевезеннями. Тому наступний розділ роботи присвячений глибокому аналізу та порівнянню існуючих систем управління збірними вантажоперевезеннями. Будуть розглянуті різні підходи та технології, які використовуються в існуючих системах управління, з метою виявлення їхніх сильних та слабких сторін, а також потенціалу для оптимізації та підвищення ефективності.

3 АНАЛІЗ І ПОРІВНЯННЯ ІСНУЮЧИХ СИСТЕМ УПРАВЛІННЯ ЗБІРНИМИ ВАНТАЖОПЕРЕВЕЗЕННЯМИ

3.1 Загальний огляд систем управління збірними вантажоперевезеннями

Збірні вантажоперевезення вимагають координації між різними видами транспорту, складськими операціями та кінцевими клієнтами, що робить управління ними складним та багатогранним завданням. В цьому контексті, використання спеціалізованих систем управління може значно підвищити ефективність та оптимізувати весь ланцюг поставок.

Для цього аналізу було обрано кілька провідних систем управління збірними вантажоперевезеннями, які широко використовуються на ринку та мають різні функціональні можливості та стратегічні переваги. Вибір цих систем базується на їхній популярності, технологічній зрілості, а також на відгуках користувачів і експертів у галузі логістики. Такий підхід дозволяє зробити об'єктивне порівняння та виявити ключові особливості кожної системи.

3.1.1 SAP Logistics Execution System (LES)

SAP Logistics Execution System (LES) розроблений компанією SAP SE, яка є одним зі світових лідерів у галузі виробництва програмного забезпечення для управління бізнес-процесами [28]. Компанія SAP відома своїми інноваційними рішеннями, які допомагають підприємствам ефективно управляти складними бізнес-операціями та аналізувати великі обсяги даних. Ця система забезпечує інтегроване управління складськими процесами та транспортуванням, що дозволяє компаніям оптимізувати свої логістичні операції та підвищити загальну ефективність.

Основні можливості SAP LES включають:

- Управління складом (Warehouse Management): SAP LES надає потужні інструменти для управління складськими запасами, включаючи

автоматизацію складських операцій, оптимізацію розміщення товарів та ефективне управління складськими потоками.

- Транспортне управління (Transportation Management): Система дозволяє планувати, оптимізувати та виконувати транспортні операції, забезпечуючи ефективне управління витратами на транспортування та підвищення рівня обслуговування клієнтів.
- Моніторинг та відстеження: SAP LES забезпечує повну видимість всіх логістичних процесів, дозволяючи відстежувати статус замовлень, вантажів та транспортних засобів у реальному часі.
- Інтеграція з іншими системами SAP: LES тісно інтегрований з іншими модулями SAP, такими як SAP ERP та SAP Supply Chain Management, що дозволяє забезпечити єдину інформаційну базу для всіх логістичних та бізнес-процесів.

На рисунках 3.1-3.4 представлено деякі з функціональних можливостей системи SAP LES.

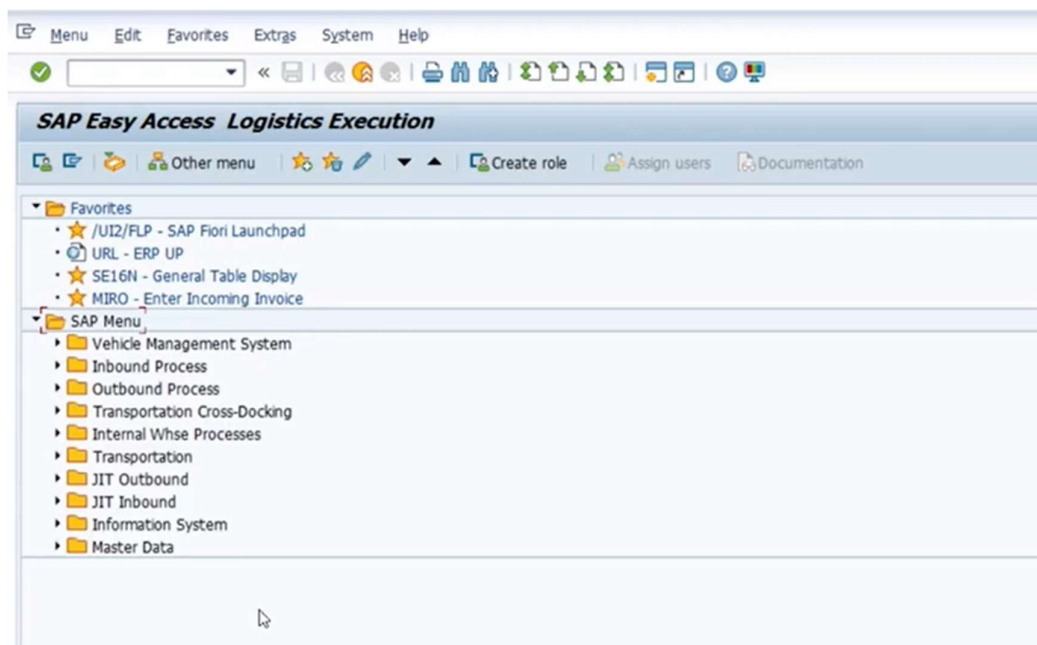


Рисунок 3.1 – Головний екран системи SAP LES

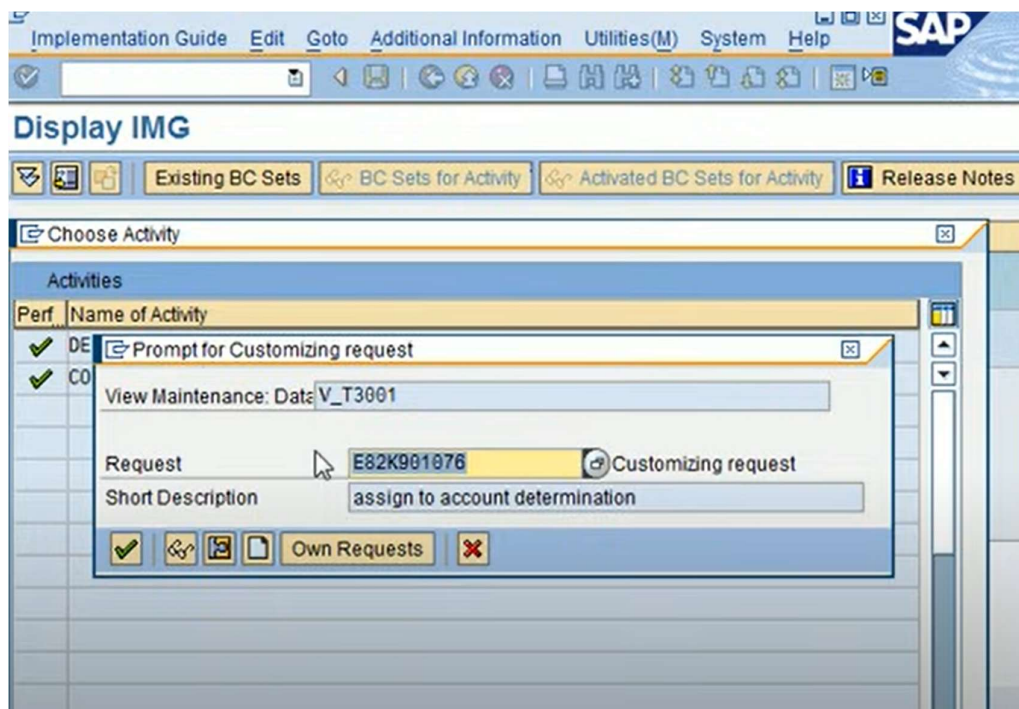


Рисунок 3.2 – Налаштування заявки в системі SAP LES

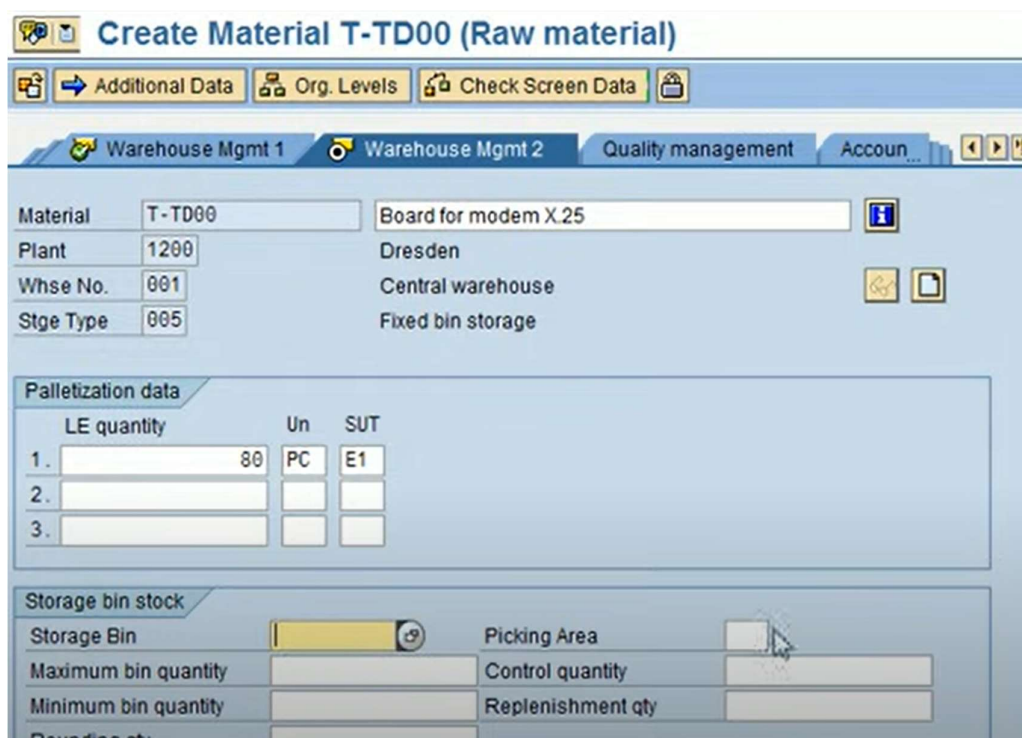


Рисунок 3.3 – Створення заявки в системі SAP LES

ShPt	Pick Date	Total Weight	WUn	Volume	VUn	ProcTime	Nr	Items
1000	01/16/2013	2,800	KG	7,500	M3	1.25	1	
80016995	01/18/2013	2,800	KG	7,500	M3			A

Рисунок 3.4 – Відображення запланованого завантаження в системі SAP LES

Переваги використання SAP LES включають:

- Підвищення продуктивності: Автоматизація складських та транспортних операцій дозволяє значно знизити час на обробку замовлень та підвищити загальну продуктивність.
- Зниження витрат: Оптимізація логістичних процесів допомагає знизити витрати на складське господарство та транспортування.
- Покращення обслуговування клієнтів: Швидке та точне виконання замовлень, а також можливість надання клієнтам актуальної інформації про статус їх замовлень.

Цінова політика SAP LES може варіюватися в залежності від конкретних потреб бізнесу, обсягу ліцензій та рівня необхідної підтримки. SAP пропонує гнучкі варіанти ліцензування, включаючи постійні ліцензії та підписки на хмарні сервіси. Ціни можуть змінюватися в залежності від регіону та специфіки впровадження системи на підприємстві. Компанія також пропонує консультаційні послуги та технічну підтримку, що також може впливати на загальну вартість рішення.

Для отримання точної інформації про ціни та доступні опції, потенційним клієнтам рекомендується звернутися безпосередньо до представників SAP або авторизованих дистриб'юторів, які можуть надати детальні консультації та підготувати індивідуальну пропозицію відповідно до специфіки бізнесу та вимог до системи.

SAP LES є відмінним вибором для великих компаній, які шукають інтегроване рішення для управління всіма аспектами своїх логістичних операцій. Його функціональність та інтеграція з іншими системами SAP роблять його одним з найбільш потужних інструментів у галузі логістики.

3.1.2 Oracle Transportation Management (OTM)

Oracle Transportation Management (OTM) є частиною широкого портфеля рішень для управління ланцюгами поставок від корпорації Oracle [29]. Ця система спеціалізується на оптимізації та автоматизації процесів транспортування та логістики, що дозволяє компаніям знижувати витрати, підвищувати продуктивність та покращувати обслуговування клієнтів.

Корпорація Oracle, що є одним з провідних світових постачальників рішень для управління даними, ланцюгами поставок та бізнес-процесами, відома своїми інноваційними технологіями та широким портфелем продуктів, що включає різноманітні програмні рішення для різних аспектів бізнесу, включаючи ERP системи, бази даних, хмарні сервіси та інші.

Основні можливості Oracle Transportation Management включають:

- Планування та оптимізація маршрутів: OTM використовує передові алгоритми для планування маршрутів та забезпечення максимальної ефективності транспортних операцій.
- Управління вантажними перевезеннями: Система дозволяє керувати всіма аспектами вантажних перевезень, включаючи вибір перевізників, управління тарифами та виконання контрактів.
- Відстеження та видимість вантажів: OTM надає інструменти для реального відстеження вантажів та забезпечення видимості всіх етапів транспортного процесу.
- Фінансове управління транспортуванням: Система включає функції для управління витратами на транспортування, виставлення рахунків та обробки платежів.

- Інтеграція з іншими системами: OTM може інтегруватися з іншими ERP-системами та рішеннями для управління ланцюгами поставок, забезпечуючи єдину інформаційну базу для всіх логістичних процесів.

На рисунках 3.5-3.9 представлено деякі з функціональних можливостей системи Oracle Transportation Management.

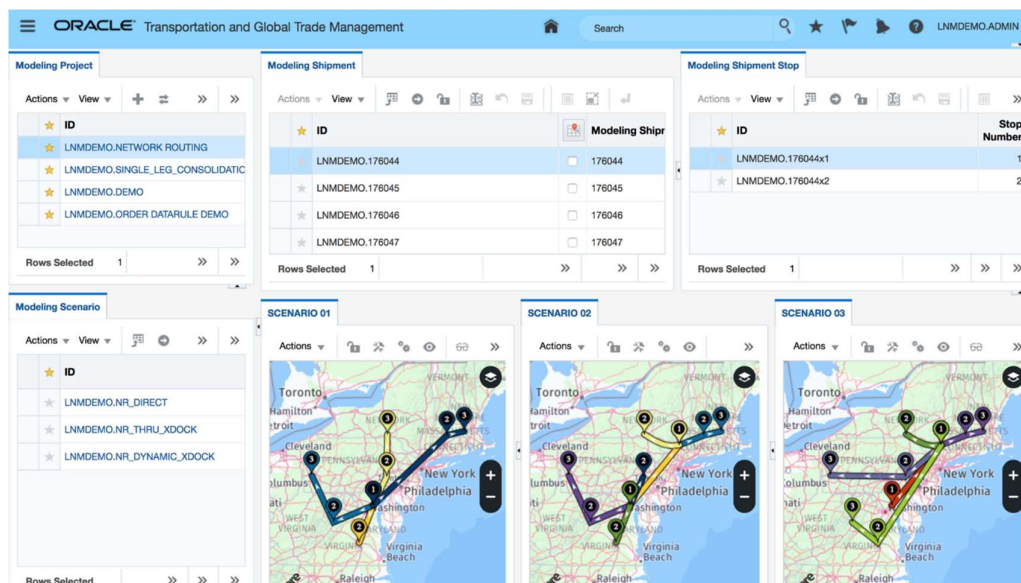


Рисунок 3.5 – Моделювання шляху вантажоперевезень в системі Oracle Transportation Management

Scenario Bulk Plan Comparison

Metrics

Performance

Bulk Plan ID	20181129-0003	20181129-0004	20181129-0002
Description	Source Changed to Boston DC	Source Changed to PHL DC	Source Location retained at BWI DC
Parameter Set ID	CONS_ALLOW_MULTISTOP	CONS_ALLOW_MULTISTOP	CONS_ALLOW_MULTISTOP
Query Name	LNMDemo.BWI_SOURCED_ORDERS	LNMDemo.BWI_SOURCED_ORDERS	LNMDemo.BWI_SOURCED_ORDERS
Modeling Project ID	ORDER DATARULE DEMO	ORDER DATARULE DEMO	ORDER DATARULE DEMO
Modeling Scenario ID	SRC_CHANGED_TO_BOS_DC	SRC_CHANGED_TO_PHL_DC	SRC_RETAINED_AS_BWI
Results			
Status	COMPLETED	COMPLETED	COMPLETED
Total Weighted Cost	3,470.48 USD	2,571.38 USD	2,489.00 USD
Total Weight	34,189.00 LB	34,189.00 LB	34,189.00 LB
Total Volume	660.00 CUFT	660.00 CUFT	660.00 CUFT
Total Distance	1,658.70 MI	1,210.69 MI	1,169.50 MI

Рисунок 3.6 – Розрахунок вартості плану маршруту в системі Oracle Transportation Management

ORACLE

Data Rule Definition ☆ 1 of 1 New Finished

* Data Rule Definition ID
SOURCE LOCATION CHANGE

Description

Domain Name
LNMDEMO

Rule Group

Table Name
ORDER_RELEASE

Include All Columns

Data Rule Parameters

Column Name	Parameter Name	Description	Fixed Values	Save
ASSIGNED_SELL_ITINERARY_GID	EARLY DELIVERY DATE			
BUY_ITINERARY_PROFILE_GID	LATE DELIVERY			
DELIVERY_IS_APPT	SOURCE LOCATION GID	Source Location ID		
DEST_LOCATION_GID				
DROPOFF_ROUTING_SEQUENCE_GID				
EARLY_DELIVERY_DATE				
EARLY_PICKUP_DATE				
EQUIPMENT_GROUP_GID				
EQUIPMENT_GROUP_PROFILE_GID				
FIXED_ITINERARY_GID				
FIXED_SELL_ITINERARY_GID				
IS_IGNORE_LOCATION_CALENDAR				
IS_SPLITTABLE				
LATE_DELIVERY_DATE				
LATE_PICKUP_DATE				
LEG_CONSOLIDATION_GROUP_GID				
MODE_PROFILE_GID				
PICKUP_IS_APPT				
PICKUP_ROUTING_SEQUENCE_GID				

Рисунок 3.7 – Можливість персоналізувати базу даних в системі Oracle Transportation Management

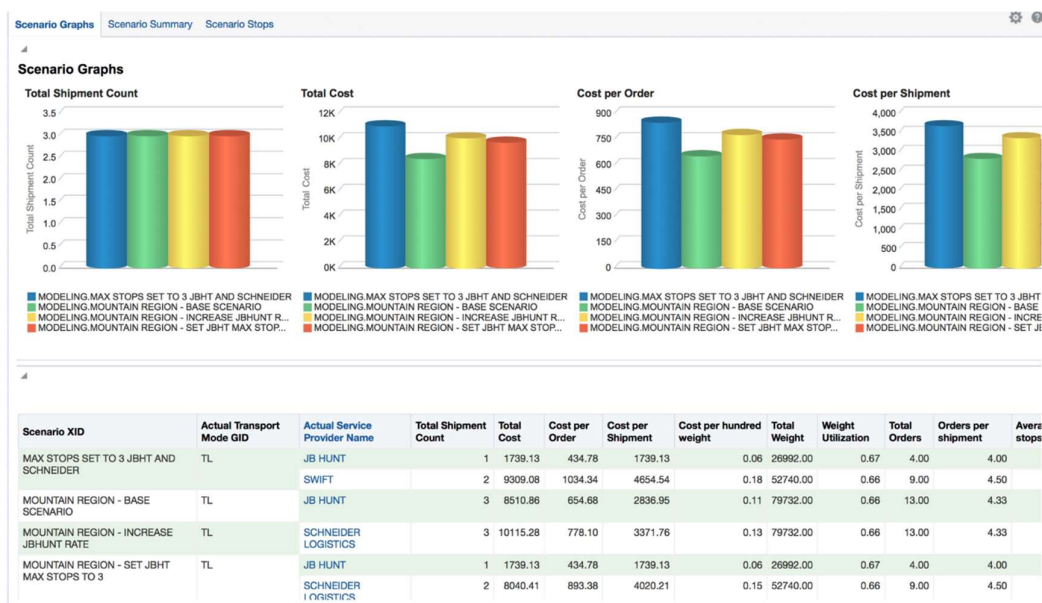


Рисунок 3.8 – Відображення статистики по виконаним вантажоперевезенням в системі Oracle Transportation Management

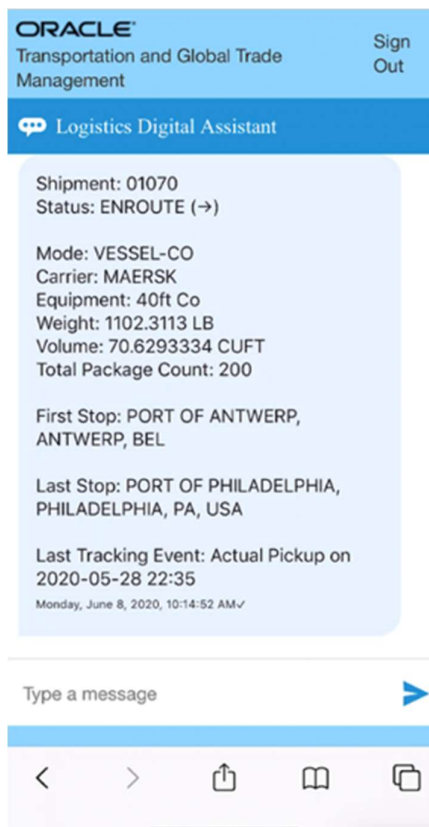


Рисунок 3.9 – Використання чат-бот асистента в мобільній версії системи Oracle Transportation Management

Переваги використання Oracle Transportation Management:

- Зниження транспортних витрат: Оптимізація маршрутів та ефективне управління тарифами допомагають знизити загальні витрати на транспортування.
- Підвищення ефективності: Автоматизація транспортних процесів сприяє швидшому та точнішому виконанню замовлень.
- Покращення обслуговування клієнтів: Точне відстеження вантажів та своєчасне інформування клієнтів про статус їх замовлень забезпечує високий рівень задоволеності клієнтів.

Ціни на Oracle Transportation Management можуть варіюватися в залежності від обраного пакету послуг, обсягу ліцензій та рівня підтримки. Oracle пропонує як постійні ліцензії, так і підписки на хмарні сервіси, що дозволяє компаніям обрати найбільш відповідний варіант фінансування. За детальною інформацією

про ціни та консультаціями звертайтеся безпосередньо до представників Oracle або їхніх авторизованих дистриб'юторів.

3.1.3 JDA Transportation Management

JDA Transportation Management є частиною комплексного набору рішень для управління ланцюгами поставок, розробленою компанією JDA Software, яка нині відома під брендом Blue Yonder [30]. Ця система спеціалізується на оптимізації всіх аспектів транспортних операцій, від планування до виконання, допомагаючи компаніям знижувати витрати та покращувати ефективність та обслуговування клієнтів.

Основні можливості JDA Transportation Management включають:

- Планування та оптимізація маршрутів: Система використовує передові алгоритми для створення оптимальних маршрутів, що мінімізують витрати та час доставки.
- Управління перевізниками: JDA дозволяє керувати відносинами з перевізниками, включаючи вибір перевізників, управління тарифами та контрактами.
- Відстеження вантажів: Система забезпечує повну видимість вантажів у реальному часі, дозволяючи відслідковувати їх статус та місцезнаходження.
- Аналітика та звітність: JDA Transportation Management включає інструменти для аналізу даних та генерації звітів, що допомагають приймати обґрунтовані рішення та покращувати процеси.
- Інтеграція з іншими системами: Система може інтегруватися з іншими ERP-системами та рішеннями для управління ланцюгами поставок, що забезпечує єдину інформаційну базу для всіх логістичних процесів.

На рисунках 3.10-3.14 представлено деякі з функціональних можливостей системи JDA Transportation Management.

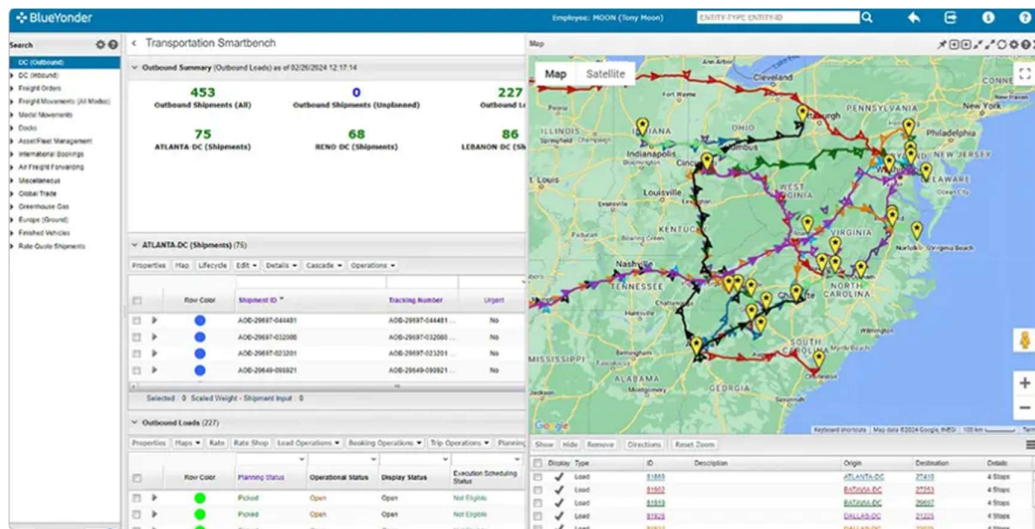


Рисунок 3.10 – Створення маршрутів вантажоперевезень в системі JDA Transportation Management

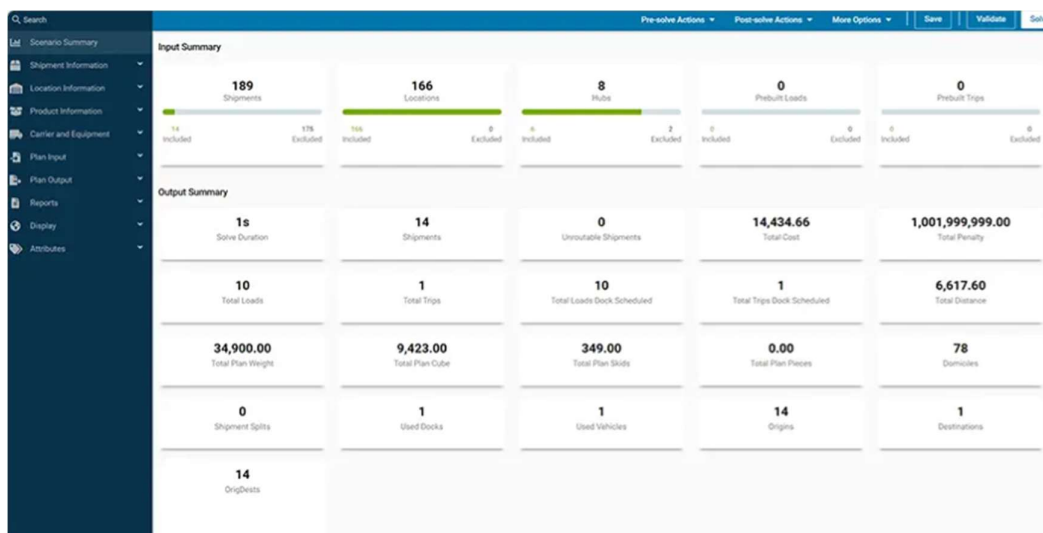


Рисунок 3.11 – Відображення поточної статистики вантажоперевезень в системі JDA Transportation Management

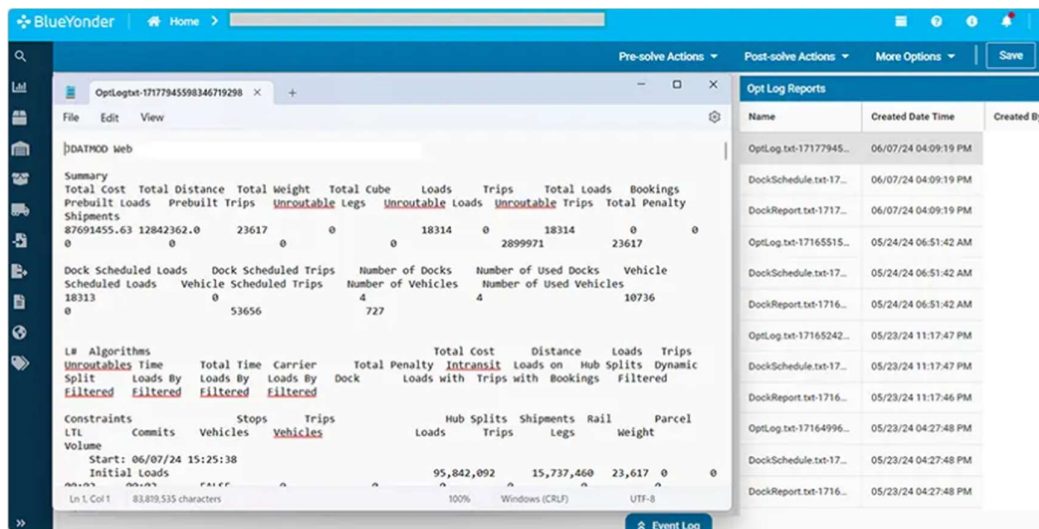


Рисунок 3.12 – Логування інформації про вантажоперевезення в системі JDA Transportation Management



Рисунок 3.13 – Тривимірна візуалізація завантаження вантажного автомобіля в системі JDA Transportation Management

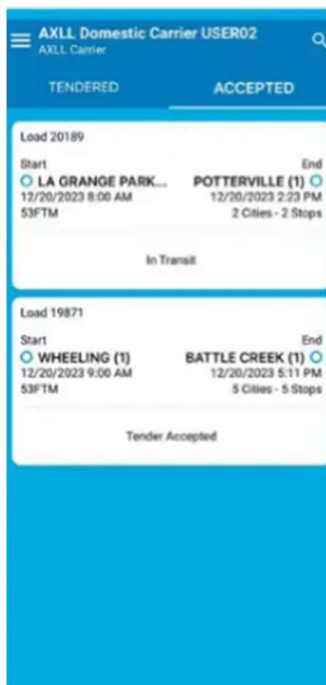


Рисунок 3.14 – Відображення прийнятих заявок на вантажоперевезення в мобільній версії системи JDA Transportation Management

Переваги використання JDA Transportation Management:

- Зниження транспортних витрат: Оптимізація маршрутів та ефективне управління перевізниками допомагають знизити витрати на транспортування.
- Підвищення ефективності: Автоматизація процесів та точне планування сприяють швидкому та ефективному виконанню замовлень.
- Покращення обслуговування клієнтів: Відстеження вантажів у реальному часі та точне інформування клієнтів про статус їх замовлень забезпечує високий рівень задоволеності клієнтів.

Вартість JDA Transportation Management може коливатися в залежності від вибраного комплекту послуг, кількості ліцензій та рівня обслуговування. Компанія Blue Yonder пропонує різні варіанти ліцензування, включаючи постійні ліцензії та підписки на хмарні рішення. Для отримання детальної інформації про цінові умови та консультації, рекомендуємо звернутися безпосередньо до представників Blue Yonder або їх авторизованих дилерів.

3.1.4 Manhattan Associates Transportation Management

Manhattan Associates Transportation Management є частиною інтегрованого портфеля рішень для управління ланцюгами поставок, розробленого компанією Manhattan Associates [31]. Ця система забезпечує комплексне управління транспортними процесами, допомагаючи компаніям оптимізувати доставку, знизити витрати та підвищити рівень задоволеності клієнтів.

Основні можливості Manhattan Associates Transportation Management включають:

- Планування та оптимізація маршрутів: Система використовує передові алгоритми для створення ефективних маршрутів, що забезпечують мінімізацію витрат і часу доставки.
- Управління перевізниками: Дозволяє керувати відносинами з перевізниками, включаючи вибір, управління тарифами та контрактами.
- Відстеження вантажів: Забезпечує повну видимість вантажів у реальному часі, дозволяючи відслідковувати їх статус і місцезнаходження.
- Аналітика та звітність: Включає інструменти для аналізу даних та генерації звітів, що допомагають приймати обґрунтовані рішення та покращувати процеси.
- Інтеграція з іншими системами: Може інтегруватися з іншими ERP-системами та рішеннями для управління ланцюгами поставок, забезпечуючи єдину інформаційну базу для всіх логістичних процесів.

На рисунках 3.15-3.18 представлено деякі з функціональних можливостей системи Manhattan Associates Transportation Management.

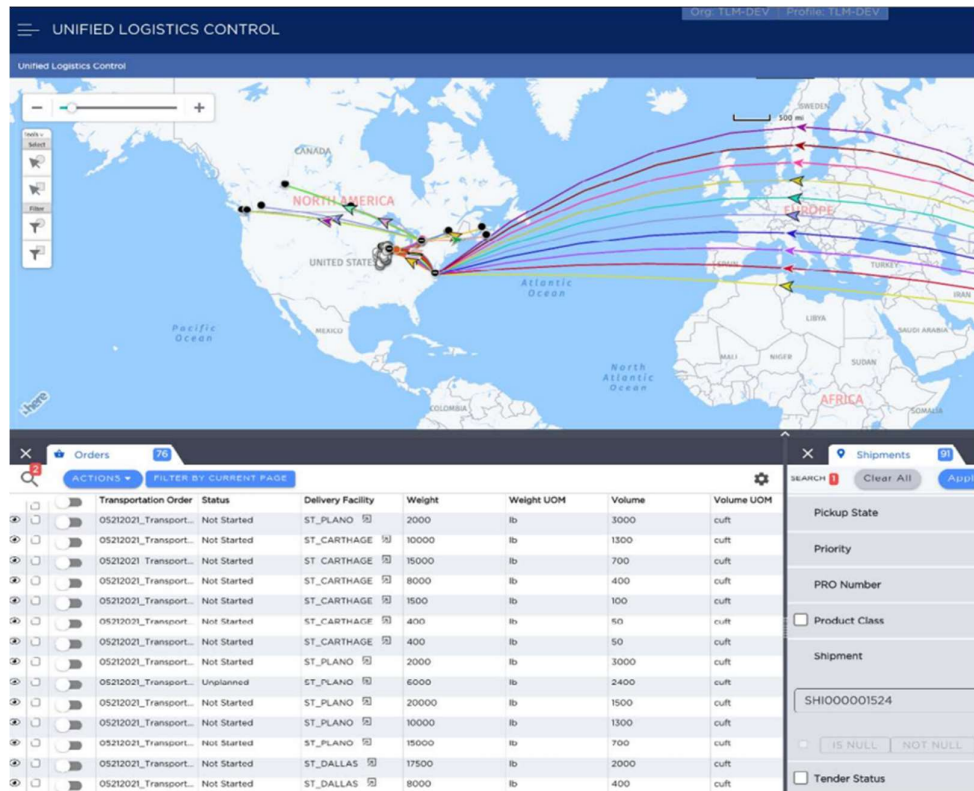


Рисунок 3.15 – Відображення запланованих маршрутів в системі Manhattan Associates Transportation Management



Рисунок 3.16 – Моделювання базових метрик для порівняння маршрутів в системі Manhattan Associates Transportation Management

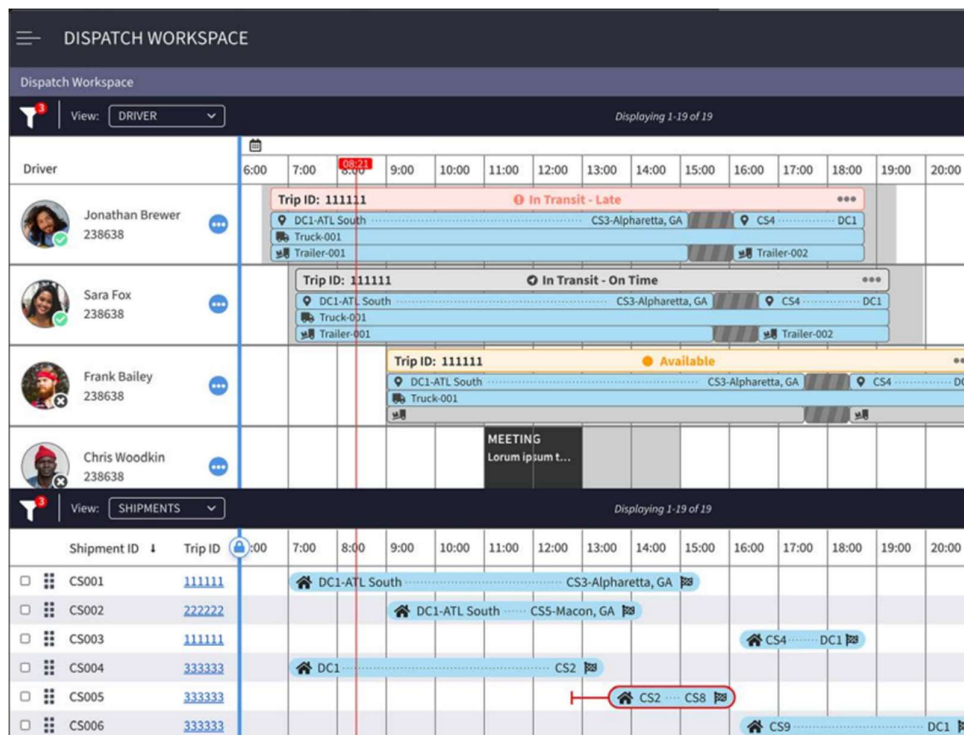


Рисунок 3.17 – Управління водіями в системі Manhattan Associates Transportation Management

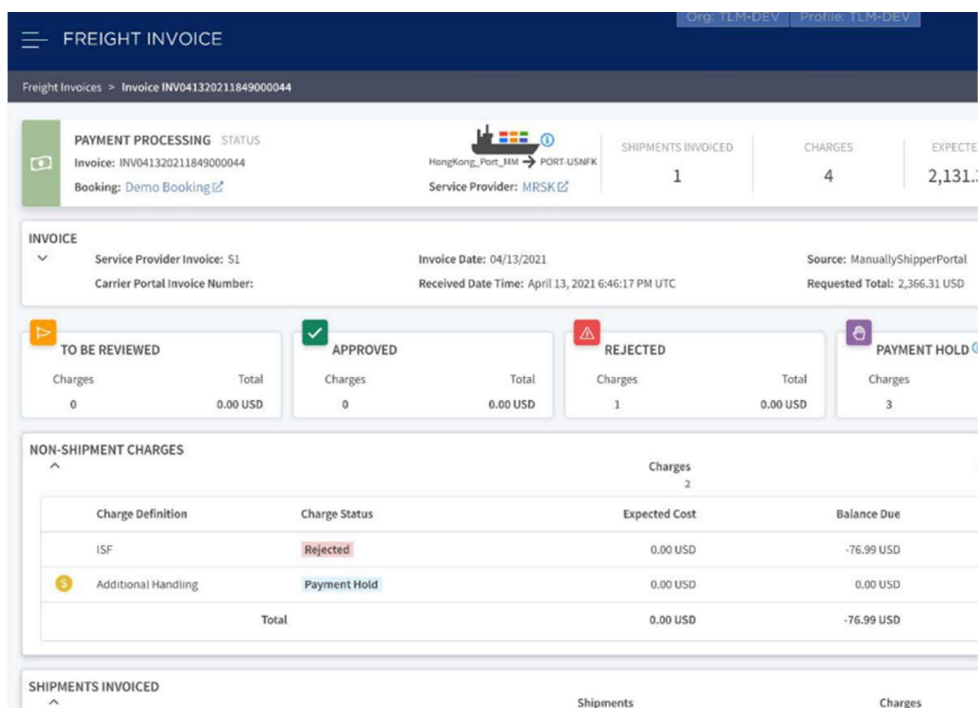


Рисунок 3.18 – Управління заявками на вантажоперевезення в системі Manhattan Associates Transportation Management

Переваги використання Manhattan Associates Transportation Management:

- Зниження транспортних витрат: Оптимізація маршрутів та ефективне управління перевізниками допомагають знизити витрати на транспортування.
- Підвищення ефективності: Автоматизація процесів та точне планування сприяють швидкому та ефективному виконанню замовлень.
- Покращення обслуговування клієнтів: Відстеження вантажів у реальному часі та точне інформування клієнтів про статус їх замовлень забезпечує високий рівень задоволеності клієнтів.

Ціни на Manhattan Associates Transportation Management можуть варіюватися залежно від обраного пакету послуг, обсягу ліцензій та рівня підтримки. Компанія пропонує як постійні ліцензії, так і підписки на хмарні рішення. За детальною інформацією про цінові умови та консультації, рекомендується звернутися безпосередньо до представників Manhattan Associates або їхніх авторизованих дистриб'юторів.

3.2 Порівняльний аналіз ефективності та обмежень кожної системи

Для зручності порівняння основних характеристик та показників систем SAP Logistics Execution System (LES), Oracle Transportation Management (OTM), JDA (Blue Yonder) Transportation Management та Manhattan Associates Transportation Management, наведемо їх у вигляді зведеної таблиці 3.1.

Таблиця 3.1 - Порівняння основних характеристик та показників розглянутих систем управління збірними вантажоперевезеннями

Показник / Система	SAP LES	Oracle OTM	JDA (Blue Yonder) TM	Manhattan Associates TM
Інтеграція з іншими системами	Висока (з іншими продуктами SAP)	Висока	Середня	Висока

Оптимізація маршрутів	Доступна	Розширені можливості	Розширені можливості	Розширені можливості
Управління перевізниками	Базові функції	Розширені можливості	Розширені можливості	Розширені можливості
Відстеження вантажів	Базові функції	Розширені можливості	Розширені можливості	Розширені можливості
Аналітика та звітність	Доступна	Розширені можливості	Розширені можливості	Розширені можливості
Масштабованість	Висока	Висока	Висока	Висока
Підходить для	Великих підприємств	Великих та середніх підприємств	Всіх типів підприємств	Великих та середніх підприємств
Складність впровадження	Висока	Середня	Середня	Середня
Вартість впровадження та підтримки	Висока	Висока	Середня	Висока
Підтримка хмарних рішень	Обмежена	Повна	Повна	Повна

Таблиця 3.2 - Порівняння другорядних характеристик та показників розглянутих систем управління збірними вантажоперевезеннями (продовження)

Показник / Система	SAP LES	Oracle OTM	JDA (Blue Yonder) TM	Manhattan Associates TM
Мобільні додатки	Ні	Так	Так	Так
Мульти-модальні перевезення	Так	Так	Так	Так
Штучний інтелект для прогнозування	Ні	Так	Так	Так
Глобальна підтримка	Так	Так	Так	Так
Автоматизація процесів	Обмежена	Розширена	Розширена	Розширена

Підсумовуючи порівняльний аналіз систем SAP Logistics Execution System (LES), Oracle Transportation Management (OTM), JDA (Blue Yonder) Transportation

Management та Manhattan Associates Transportation Management, можна виділити кілька ключових аспектів, які варто врахувати при виборі системи управління транспортом.

Перш за все, важливо звернути увагу на технологічні можливості кожної системи. Oracle OTM, JDA (Blue Yonder) TM та Manhattan Associates TM пропонують повну підтримку хмарних рішень, що робить їх більш гнучкими та доступними для різних типів бізнесу. Вони також мають мобільні додатки, що дозволяє керувати логістичними процесами в режимі реального часу з будь-якого місця.

Щодо інноваційних технологій, таких як штучний інтелект для прогнозування та автоматизації, Oracle OTM, JDA (Blue Yonder) TM та Manhattan Associates TM виходять на передові позиції, пропонуючи розширені можливості, які можуть значно покращити ефективність та точність логістичних операцій.

SAP LES, хоча і є потужним рішенням, має деякі обмеження, такі як відсутність мобільних додатків та обмежена підтримка хмарних рішень, що може бути критичним для компаній, які шукають більшу гнучкість та мобільність.

Враховуючи мульти-модальні перевезення, всі чотири системи пропонують підтримку, що є важливим для компаній, які займаються різними видами доставки.

Загалом, вибір системи залежить від специфічних потреб бізнесу, бюджету, а також від того, наскільки важливі інноваційні технології та гнучкість управління. Oracle OTM, JDA (Blue Yonder) TM та Manhattan Associates TM пропонують більш передові технологічні можливості, тоді як SAP LES може бути більш підходящим для компаній, які вже використовують інші продукти SAP і шукають рішення з високою інтеграцією.

3.3 Висновки

На основі проведеного аналізу можна зробити висновок про те, які системи управління транспортом найкраще відповідають потребам сучасних логістичних компаній, забезпечуючи їм необхідні технологічні можливості та гнучкість. Вибір

правильної системи є критичним кроком для оптимізації логістичних процесів та підвищення ефективності вантажоперевезень.

Переходячи до наступного розділу, увага буде зосереджена на проектуванні автоматизованої системи управління збірними вантажоперевезеннями. Цей розділ буде включати детальний аналіз вимог до системи, розробку архітектури та визначення ключових функціональних можливостей, які повинні бути реалізовані для досягнення оптимальної продуктивності та ефективності в управлінні збірними вантажами.

4 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ЗБІРНИМИ ВАНТАЖОПЕРЕВЕЗЕННЯМИ

4.1 Визначення вимог до системи

Проектування ефективної автоматизованої системи управління збірними вантажоперевезеннями починається з чіткого визначення вимог до системи. Ці вимоги поділяються на кілька категорій, включаючи функціональні, нефункціональні, технічні та бізнес-вимоги.

Функціональні вимоги:

1. Планування маршрутів: Система повинна автоматично генерувати оптимальні маршрути доставки, враховуючи такі параметри, як відстань, час в дорозі, вартість перевезення, та доступність транспортних засобів.
2. Управління замовленнями: Можливість приймати та обробляти замовлення на перевезення, включаючи інтеграцію з системами клієнтів для автоматичного обміну даними.
3. Відстеження вантажів: Забезпечення повної видимості вантажів у реальному часі з можливістю інформування клієнтів про статус доставки.
4. Управління ресурсами: Оптимізація використання транспортних засобів та водійського складу, включаючи планування графіків роботи та обслуговування.

Нефункціональні вимоги:

1. Надійність: Система повинна бути стабільною та забезпечувати високий рівень доступності.
2. Масштабованість: Можливість легкого масштабування системи для підтримки зростання обсягів перевезень та розширення географії діяльності.
3. Безпека: Захист даних та системи від несанкціонованого доступу та інших загроз.

Технічні вимоги:

1. Інтеграція: Сумісність з існуючими ІТ-системами, включаючи ERP, CRM та інші логістичні платформи.
2. Інтерфейс користувача: Інтуїтивно зрозумілий та зручний інтерфейс для всіх категорій користувачів.
3. Підтримка мобільних пристроїв: Доступність мобільних додатків для водіїв та логістичних менеджерів.

Бізнес-вимоги:

1. Вартісна ефективність: Система повинна пропонувати рішення, що забезпечують оптимальне співвідношення ціни та якості.
2. ROI (повернення інвестицій): Чітке розуміння того, як система допоможе знизити витрати та збільшити доходи [32-34].

З урахуванням цих вимог, наступним кроком у проектуванні системи буде розробка її архітектури, що дозволить реалізувати зазначені функціональні та нефункціональні можливості.

4.2 Розробка архітектури системи

Розробка архітектури автоматизованої системи управління збірними вантажоперевезеннями є ключовим етапом, який визначає як система буде функціонувати, які технології будуть використані, та як вона інтегруватиметься з іншими системами. Архітектура системи повинна бути гнучкою, масштабованою та забезпечувати високий рівень безпеки та надійності.

Компоненти системи та їх взаємодію можна представити у вигляді UML-діаграми компонентів, що зображена на рисунку 4.1, а саме:

1. Модуль управління замовленнями: Забезпечує прийом та обробку замовлень на перевезення, включаючи введення даних, перевірку та підтвердження замовлень.
2. Модуль планування маршрутів: Використовує алгоритми оптимізації для створення ефективних маршрутів доставки, враховуючи різні параметри, такі як час, відстань, вартість та доступність транспорту.

3. Модуль відстеження вантажів: Надає інформацію про місцезнаходження вантажів у реальному часі, використовуючи GPS та інші технології відстеження.
4. Модуль управління ресурсами: Включає управління транспортними засобами, водіями та іншими ресурсами, необхідними для виконання перевезень.
5. Модуль звітності та аналітики: Забезпечує збір та аналіз даних про ефективність перевезень, допомагаючи приймати обґрунтовані управлінські рішення.

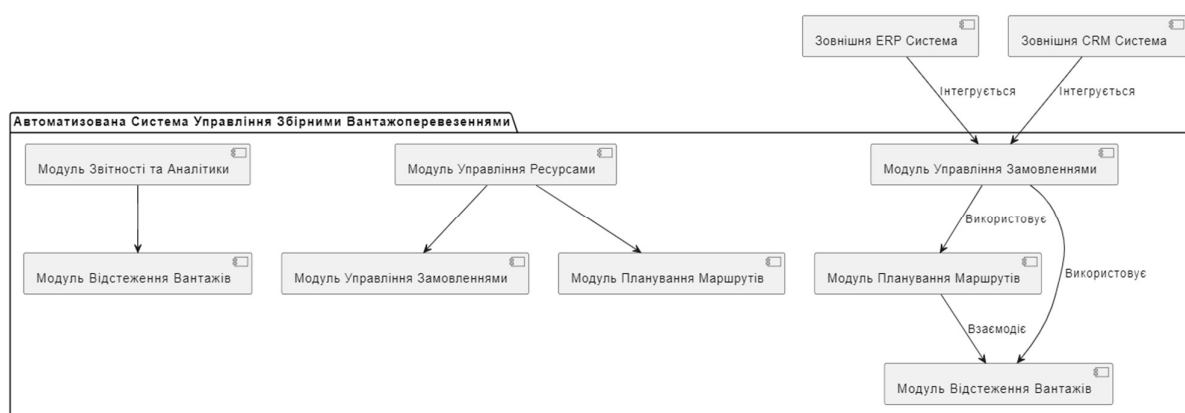


Рисунок 4.1 – UML-діаграма компонентів автоматизованої системи управління збірними вантажоперевезеннями

Технологічну структуру в свою чергу можна представити у вигляді UML-діаграми розгортання, що зображена на рисунку 4.2.

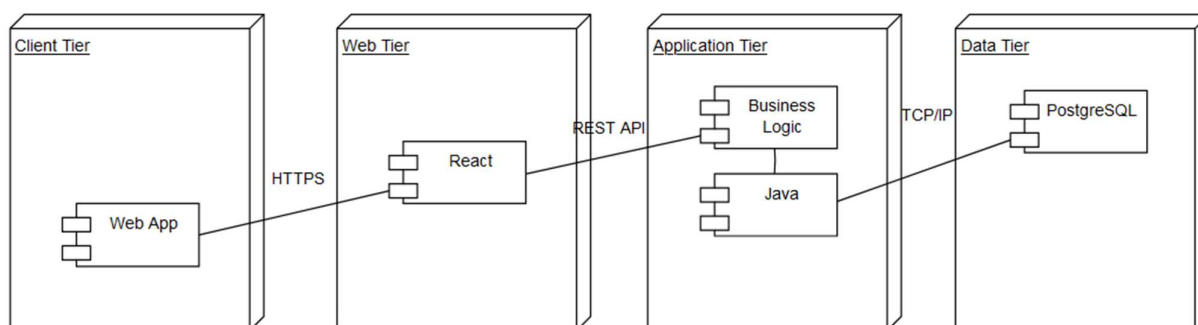


Рисунок 4.2 – UML-діаграми розгортання технологічної структури автоматизованої системи управління збірними вантажоперевезеннями

Компонентами UML-діаграми розгортання є:

- База даних: Використання реляційної бази даних для зберігання та обробки великих обсягів даних, що забезпечує надійність та консистентність даних.
- Backend: Розробка на мові програмування Java, з використанням Spring Framework для розробки серверної частини, що дозволяє ефективно управляти бізнес-логікою та забезпечує високу продуктивність.
- Frontend: Створення користувацьких інтерфейсів з використанням HTML, CSS та JavaScript, а також фреймворку React, що дозволяє створювати інтуїтивно зрозумілі та відгукові інтерфейси.
- Інтеграція: Використання RESTful API для інтеграції з Google Maps, що дозволяє візуалізувати маршрути вантажоперевезень та оптимізувати логістичні процеси.
- Безпека: Застосування сучасних методів шифрування та аутентифікації, включаючи OAuth та SSL/TLS, для захисту даних та системи від несанкціонованого доступу [35].

Дана архітектура створює міцну основу для розробки та впровадження автоматизованої системи управління збірними вантажоперевезеннями, яка може ефективно вирішувати завдання, поставлені перед сучасними логістичними компаніями.

4.3 Проектування бази даних

У процесі розробки будь-якої системи ключовим аспектом є вибір типу бази даних. Для автоматизованої системи управління збірними вантажоперевезеннями було обрано реляційну базу даних з огляду на її здатність забезпечувати високий рівень консистентності та надійності даних. Реляційні бази даних відомі своєю структурованістю та строгими схемами, що дозволяє легко встановлювати зв'язки

між різними типами даних, що є критично важливим для комплексних логістичних операцій. Також, реляційні бази даних підтримують транзакції, що є необхідним для забезпечення цілісності даних під час виконання різних операцій, які можуть одночасно впливати на кілька таблиць [36, 37].

Проектування бази даних для автоматизованої системи управління збірними вантажоперевезеннями відбувалося в кілька етапів. Спочатку було проведено аналіз вимог до системи, що дозволило визначити ключові сутності, які повинні бути представлені в базі даних. Ці сутності включають вантажі, транспортні засоби, замовлення, користувачів та компанії.

На наступному етапі було розроблено концептуальну схему, яка відображає зв'язки між сутностями, що представлено на рисунку 4.3.

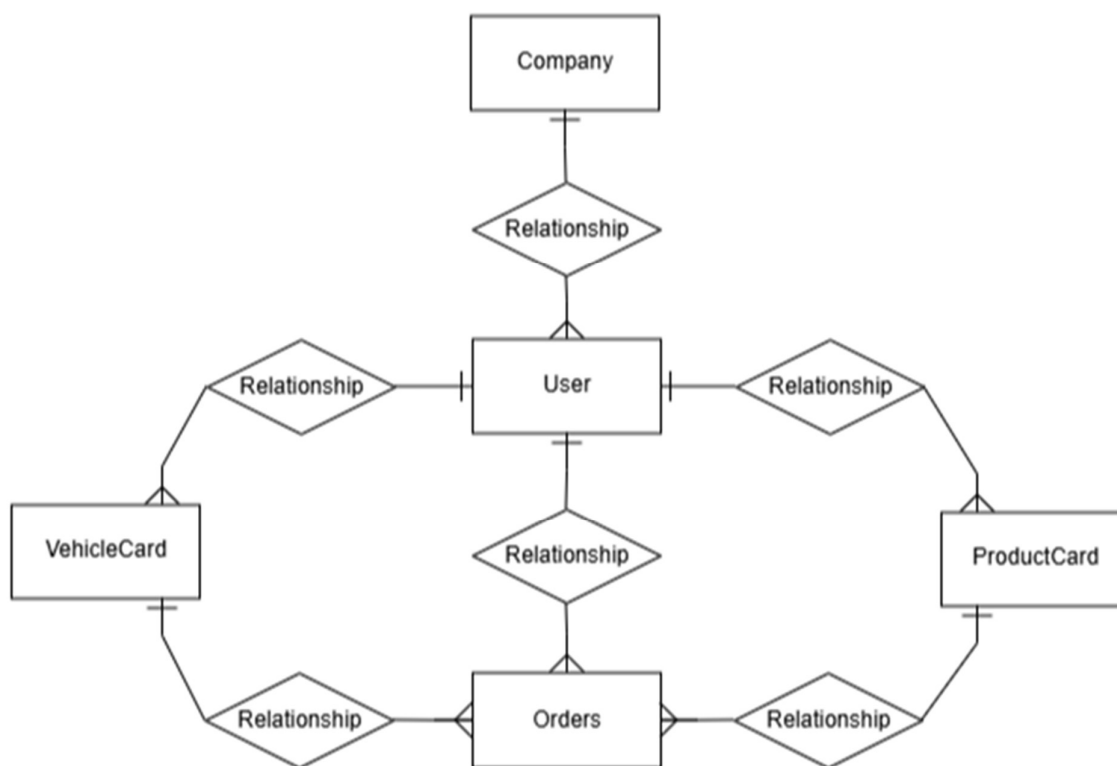


Рисунок 4.3 – Концептуальна схема бази даних автоматизованої системи управління збірними вантажоперевезеннями

На даній схемі бази даних представлено п'ять основних сутностей: User, Company, VehicleCard, ProductCard та Orders. Компанія (Company) може мати багато співробітників (User), що вказує на зв'язок один до багатьох між Company

та User. Користувачі (User) можуть володіти багатьма вантажними картками (ProductCard) та картками транспортних засобів (VehicleCard), що також представлено як зв'язки один до багатьох. Кожна вантажна картка (ProductCard) та кожна картка транспортного засобу (VehicleCard) можуть бути пов'язані з багатьма замовленнями (Orders), що відображено зв'язками один до багатьох. Користувачі (User) також можуть створювати замовлення (Orders), що вказує на ще один зв'язок один до багатьох між User та Orders.

Наступним етапом, на основі концептуальної схеми, було створено логічну схему, яка вже включала визначення таблиць, їхніх атрибутів та зв'язків між ними.

Логічна схема структури бази даних представлена у вигляді ER-діаграми на рисунку 4.4.

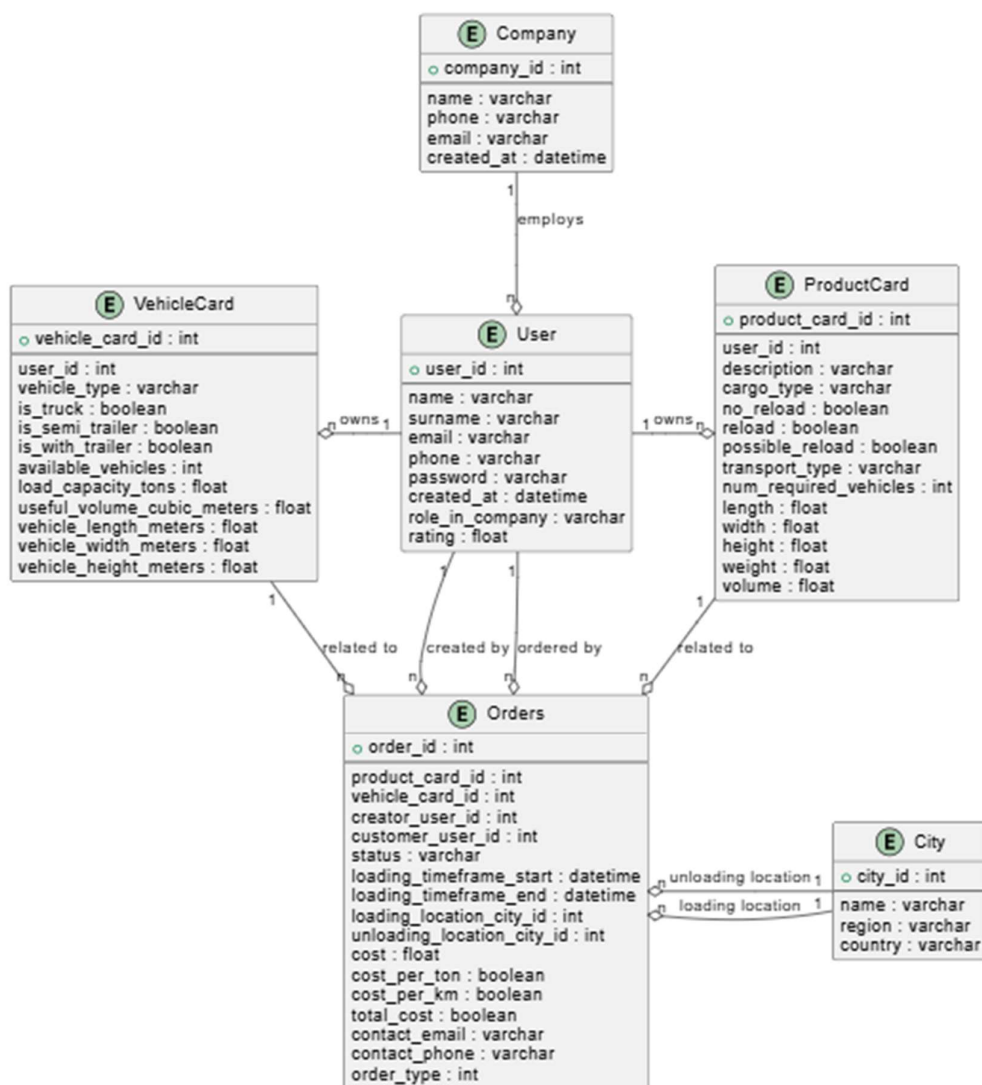


Рисунок 4.4 – ER-діаграма структури бази даних автоматизованої системи управління збірними вантажоперевезеннями

На діаграмі зображено наступні сутності та зв'язки між ними:

1. Company:

- Кожна компанія має унікальний ідентифікатор (`company\_id`), назву, телефон, електронну адресу та дату створення.
- Компанія може мати багато співробітників (`User`), що відображено зв'язком один до багатьох.

2. User:

- Кожен користувач має унікальний ідентифікатор (`user\_id`), ім'я, прізвище, електронну адресу, телефон, пароль, дату створення, роль у компанії та рейтинг.
- Користувач може володіти багатьма вантажними картками (`ProductCard`) та картками транспортних засобів (`VehicleCard`), що відображено зв'язками один до багатьох.
- Користувач може створювати та замовляти багато замовлень (`Orders`), що також відображено зв'язками один до багатьох.

3. City:

- Кожне місто має унікальний ідентифікатор (`city\_id`), назву, регіон та країну.
- Місто може бути місцем завантаження або розвантаження для багатьох замовлень (`Orders`), що відображено зв'язками один до багатьох.

4. ProductCard:

- Кожна вантажна картка має унікальний ідентифікатор (`product\_card\_id`), опис, тип вантажу, параметри перевантаження, тип транспорту, кількість необхідних транспортних засобів, розміри та вагу.
- Кожна вантажна картка може бути пов'язана з багатьма замовленнями (`Orders`), що відображено зв'язком один до багатьох.

5. VehicleCard:

- Кожна картка транспортного засобу має унікальний ідентифікатор ('vehicle_card_id'), тип транспортного засобу, наявність вантажівки, напівпричепа, причепа, кількість доступних транспортних засобів, вантажопідйомність, корисний об'єм, розміри.
- Кожна картка транспортного засобу може бути пов'язана з багатьма замовленнями ('Orders'), що відображено зв'язком один до багатьох.

6. Orders:

- Кожне замовлення має унікальний ідентифікатор ('order_id'), ідентифікатори вантажної та транспортної карток, ідентифікатори користувача-створювача та замовника, статус, часові рамки завантаження та розвантаження, місця завантаження та розвантаження, вартість, тип оплати, контактну електронну адресу та телефон.

Завершальним етапом проектування бази даних стало фізичне проектування, під час якого було визначено конкретні типи даних для атрибутів, індекси для підвищення продуктивності запитів, а також обмеження цілісності, такі як первинні ключі та зовнішні ключі, що забезпечують правильність зв'язків між таблицями. Цей етап також включав налаштування параметрів зберігання та доступу до даних, що є важливим для забезпечення високої продуктивності та масштабованості системи.

Для реалізації бази даних було обрано PostgreSQL, оскільки ця система управління базами даних відома своєю високою надійністю, підтримкою великої кількості даних та розширеними можливостями управління транзакціями. PostgreSQL також пропонує великий набір розширень та інструментів для оптимізації роботи, що дозволяє ефективно налаштувати базу даних під конкретні потреби системи [38].

У рамках налаштування PostgreSQL для автоматизованої системи управління збірними вантажоперевезеннями було зосереджено увагу на оптимізації процесів читання та запису, що є критично важливим для подібних систем, де швидкість обробки даних може значно впливати на загальну продуктивність. Було впроваджено індексацію ключових полів, що часто

використовуються у запитах, для зменшення часу відповіді бази даних. Також було налаштовано реплікацію даних для забезпечення високої доступності та надійності системи, що дозволяє уникнути втрати даних у разі збоїв або відмов одного з серверів.

Ці налаштування дозволили створити міцну та ефективну базу даних, яка забезпечує високий рівень продуктивності та надійності для автоматизованої системи управління збірними вантажоперевезеннями.

4.4 Розробка UML-діаграм

Розробка UML-діаграм є важливою для проектування та реалізації програмних систем, оскільки дозволяє архітекторам, розробникам та зацікавленим сторонам мати чітке візуальне представлення структури та взаємодії компонентів системи. Тому окрім діаграми компонентів та діаграми розгортання, будуть розроблені ще UML-діаграма варіантів використання, діаграма класів та послідовностей.

4.4.1 Use Case UML-діаграма

Use Case UML-діаграма (діаграма варіантів використання) — це візуальне представлення функціональних вимог системи, яке показує різні способи взаємодії користувачів (акторів) з системою для досягнення конкретних цілей. Ця діаграма допомагає визначити взаємозв'язки між акторами та основними функціями системи, а також сприяє кращому розумінню зовнішньої поведінки та взаємодій у системі [39].

Use Case UML-діаграма для користувача автоматизованої системи управління збірними вантажоперевезеннями представлена на рисунку 4.5.



Рисунок 4.5 – Use Case UML-діаграма для користувача автоматизованої системи управління збірними вантажоперевезеннями

Дана UML-діаграма варіантів використання представляє різні функціональні можливості, які користувачі можуть виконувати в системі. Основні актори включають звичайних користувачів та адміністраторів.

1. Управління особистими даними (Manage Personal Data): Користувачі можуть редагувати, створювати та видаляти свої особисті дані.
2. Управління даними компанії (Manage Company Data): Тільки адміністратори мають доступ до редагування, створення та видалення даних компанії. Цей варіант використання розширюється для адміністраторів.
3. Управління даними продуктів (Manage Product Data): Користувачі можуть редагувати, створювати, видаляти та архівувати дані про продукти.
4. Управління даними транспортних засобів (Manage Vehicle Data): Користувачі можуть редагувати, створювати, видаляти та архівувати дані про транспортні засоби.
5. Управління замовленнями (Manage Orders): Користувачі можуть редагувати, створювати, видаляти, архівувати замовлення та переглядати існуючі замовлення.
6. Перегляд статистики транспортування (View Transportation Statistics): Користувачі можуть переглядати статистичні дані, пов'язані з транспортуванням.
7. Розрахунок відстані між містами (Calculate Distance Between Cities): Користувачі можуть використовувати цю функцію для визначення відстані між містами.

Кожен варіант використання розбивається на підваріанти, що дозволяє користувачам виконувати специфічні дії в рамках більш широких категорій.

4.4.2 Class UML-діаграма

Class UML-діаграма (діаграма класів) є одним із інструментів моделювання, яка використовується для візуалізації структури системи, показуючи класи, їх атрибути, методи та взаємозв'язки між ними [40]. На рисунку 4.6 представлена

UML-діаграма класів автоматизованої системи управління збірними вантажоперевезеннями.

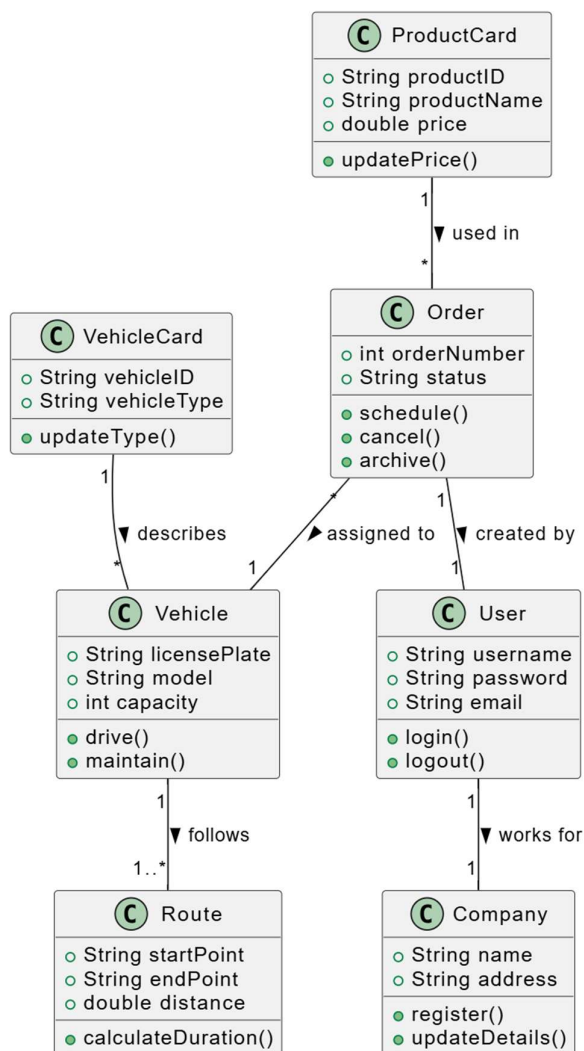


Рисунок 4.6 – Class UML-діаграма автоматизованої системи управління збірними вантажоперевезеннями

Діаграма класів для автоматизованої системи управління збірними вантажоперевезеннями включає наступні класи та їх взаємозв'язки:

1. User: Цей клас представляє користувачів системи. Він містить атрибути для зберігання імені користувача, пароля та електронної пошти. Методи 'login()' та 'logout()' дозволяють користувачам входити в систему та виходити з неї.

2. **Company:** Клас, що представляє компанію, яка використовує систему. Він включає атрибути для назви та адреси компанії. Методи ``register()`` та ``updateDetails()`` дозволяють реєструвати компанію в системі та оновлювати її деталі.
3. **Vehicle:** Клас для транспортних засобів, які використовуються для перевезення вантажів. Включає атрибути для номера автомобіля, моделі та вантажопідйомності. Методи ``drive()`` та ``maintain()`` відповідають за керування транспортним засобом та його технічне обслуговування.
4. **Route:** Клас, що описує маршрути перевезення. Включає атрибути для початкової та кінцевої точки маршруту, а також відстані між ними. Метод ``calculateDuration()`` дозволяє розрахувати тривалість поїздки.
5. **Order:** Клас для управління замовленнями на перевезення. Містить атрибути для номера замовлення та його статусу. Методи ``schedule()``, ``cancel()`` та ``archive()`` дозволяють планувати, скасовувати та архівувати замовлення.
6. **ProductCard:** Клас для управління інформацією про продукти. Включає атрибути для ідентифікатора продукту, назви та ціни. Метод ``updatePrice()`` дозволяє оновлювати ціну продукту.
7. **VehicleCard:** Клас для управління інформацією про транспортні засоби. Включає атрибути для ідентифікатора транспортного засобу та його типу. Метод ``updateType()`` дозволяє оновлювати тип транспортного засобу.

Взаємозв'язки між класами:

- Кожен **User** працює на одну **Company**.
- Кожен **Vehicle** може слідувати одному або більше **Route**.
- Кожне **Order** може бути призначене одному **Vehicle** і створене одним **User**.
- Кожна **ProductCard** може бути використана в одному або більше **Order**.
- Кожна **VehicleCard** описує один або більше **Vehicle**.

Дана діаграма класів відображає структуру та взаємодії в системі управління збірними вантажоперевезеннями, забезпечуючи чітке розуміння ролей та відповідальностей кожного компонента.

4.4.3 Sequence UML-діаграма

Sequence UML-діаграма (діаграма послідовності) використовується для візуалізації порядку взаємодії між об'єктами в рамках певного сценарію та показує, як об'єкти спілкуються між собою залежно від часу [41]. На рисунку 4.7 представлена UML-діаграма послідовності процесу створення замовлення на перевезення в автоматизованій системі управління збірними вантажоперевезеннями.

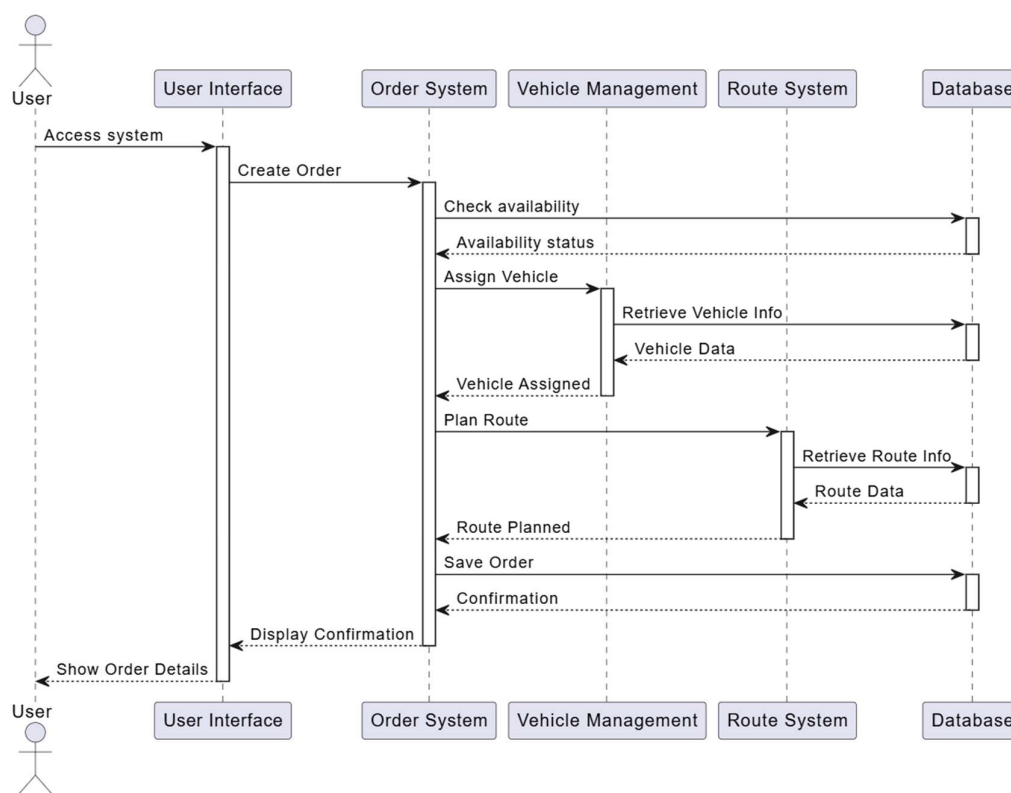


Рисунок 4.7 – UML-діаграма послідовності процесу створення замовлення на перевезення в автоматизованій системі управління збірними вантажоперевезеннями

Згідно діаграми користувач спочатку входить в систему через інтерфейс користувача і ініціює створення нового замовлення. Система управління замовленнями звертається до бази даних для перевірки доступності ресурсів і, після отримання підтвердження, призначає транспортний засіб через систему

управління транспортними засобами. Наступним кроком є планування маршруту, для чого система управління замовленнями координується з системою планування маршрутів. Після планування маршруту та призначення транспортного засобу, замовлення зберігається в базі даних. Завершується процес відображенням деталей замовлення користувачеві через інтерфейс користувача, що підтверджує успішне створення замовлення.

4.5 Висновки

Розробивши діаграми та структуру бази даних, можна перейти до наступного етапу розробки автоматизованої системи, що включає написання коду, інтеграцію компонентів системи та тестування функціоналу. Завдяки чітко визначеним діаграмам, отримано зрозумілу візуалізацію архітектури системи та взаємодій між компонентами, що спрощує процес кодування та знижує ризик помилок. Визначення структури бази даних дозволяє ефективно організувати зберігання та доступ до даних, що є критично важливим для стабільності та швидкодії системи.

5 РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ

5.1 Вибір технологій розробки

У розділі 5.1 розглядається вибір технологій, які були використані для розробки автоматизованої системи управління збірними вантажоперевезеннями. Для серверної частини було обрано мову програмування Java версії 11, оскільки вона забезпечує високу продуктивність, надійність та широкі можливості для розробки масштабованих систем. Використання Spring Framework версії 5 дозволило ефективно управляти бізнес-логікою та інтеграцією з іншими компонентами системи.

Для розробки клієнтської частини було обрано JavaScript з використанням бібліотеки React версії 16. Це забезпечило створення інтерактивних та відгукових користувацьких інтерфейсів. React виявився зручним у використанні для створення складних інтерфейсів, які вимагають динамічного оновлення даних без перезавантаження сторінки [41, 42].

Для забезпечення взаємодії між клієнтською та серверною частинами було використано RESTful API, реалізоване за допомогою Spring Boot версії 2. Це дозволило забезпечити легкість інтеграції та високу масштабованість системи. API було налаштовано для забезпечення безпечного з'єднання між різними компонентами системи через HTTPS, використовуючи SSL/TLS сертифікати для шифрування даних.

База даних була реалізована за допомогою PostgreSQL версії 12, що забезпечило надійне зберігання даних та високу продуктивність при роботі з великими обсягами інформації. Використання реляційної моделі дозволило ефективно організувати дані та забезпечити їх цілісність. Інтеграція бази даних з серверною частиною була виконана через JDBC, що дозволило забезпечити стабільне та ефективне з'єднання.

Таким чином, вибір технологій та їх версій був зумовлений потребами проекту та вимогами до продуктивності, надійності та масштабованості системи.

Цілісна інтеграція всіх компонентів забезпечила створення надійної та ефективної системи управління збірними вантажоперевезеннями.

Для написання коду було використано інтегроване середовище розробки (IDE) IntelliJ IDEA, яке забезпечує широкі можливості для розробки на Java та JavaScript. IntelliJ IDEA відома своїм зручним інтерфейсом, потужними інструментами для рефакторингу та дебагінгу, а також вбудованою підтримкою для Spring та React. Це значно спростило процес розробки та дозволило зосередитись на реалізації бізнес-логіки без необхідності витратити час на налаштування середовища [43].

Крім того, для забезпечення консистентності розробки та легкості розгортання системи використовувалась технологія контейнеризації Docker. Docker дозволив упакувати додаток та всі його залежності в контейнери, що забезпечило легке та швидке розгортання системи на будь-якій платформі. Це також сприяло легкій масштабованості системи, оскільки контейнери можна легко розмножувати та розподіляти на різних вузлах. Використання Docker Compose допомогло управляти множиною контейнерів, що використовуються в системі, забезпечуючи їхнє автоматичне розгортання та взаємодію.

У системі використовувалася множина контейнерів Docker, кожен з яких відповідав за певний аспект функціональності системи. Один з контейнерів містив серверну частину, реалізовану на Spring Boot, яка відповідала за обробку бізнес-логіки та взаємодію з базою даних. Інший контейнер використовувався для бази даних PostgreSQL, забезпечуючи надійне зберігання та управління даними. Також був окремий контейнер для клієнтської частини, розробленої на React, який забезпечував відображення інтерфейсу користувача та взаємодію з серверною частиною через RESTful API [44-46].

Використання цих контейнерів значно підвищило ефективність розробки та розгортання системи, забезпечуючи високий рівень ізоляції компонентів та легкість масштабування.

5.2 Реалізація функціональності системи

Функціональності роботи автоматизованої системи описується злагодженою роботою бекенд та фронтенд частин, а також інтеграцією з базою даних PostgreSQL для забезпечення ефективної роботи над управлінням вантажоперевезеннями.

Основні компоненти бекенду включають:

- Сервіси для обробки замовлень: Класи Java, які відповідають за прийом, обробку та збереження даних замовлень в базі даних.
- Контролери для API: Контролери Spring MVC, які приймають HTTP запити від фронтенду та відправляють відповіді [47].

Для підвищення функціональності бекенду системи управління вантажоперевезеннями, було додано додаткові компоненти та функції такі як:

- Модуль аутентифікації та авторизації, що використовує Spring Security для перевірки користувачів та надання доступу до ресурсів на основі їхніх ролей.

```
@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .csrf().disable()
            .authorizeRequests()
            .antMatchers("/api/public/**").permitAll()
            .antMatchers("/api/admin/**").hasRole("ADMIN")
            .anyRequest().authenticated()
            .and()
            .httpBasic();
    }
}
```

Цей конфігураційний клас встановлює політику безпеки, де ендпоінти в `/api/admin/` доступні лише для користувачів з роллю ADMIN, а інші запити вимагають аутентифікації.

- Логування запитів та відповідей для поліпшення здатності відстежувати та аналізувати діяльність в системі, що було впроваджено за допомогою AOP (Aspect-Oriented Programming) [48].

```
@Aspect
@Component
public class LoggingAspect {

    @Around("execution(* com.example.transportation.controller.*.*(..))")
    public Object logControllerAccess(ProceedingJoinPoint joinPoint) throws Throwable {

        long startTime = System.currentTimeMillis();

        Object result = joinPoint.proceed();

        long endTime = System.currentTimeMillis();

        Logger logger = LoggerFactory.getLogger(joinPoint.getTarget().getClass());

        logger.info("Method {} executed in {} ms", joinPoint.getSignature().getName(), (endTime -
            startTime));

        return result;
    }
}
```

Цей аспект логує час виконання кожного методу в контролерах, що допомагає ідентифікувати потенційні затримки або проблеми з продуктивністю.

- Модуль аналітики, який збирає та обробляє дані про замовлення, маршрути та час доставки.

```
@Service
public class AnalyticsService {

    public DeliveryAnalytics calculateDeliveryMetrics() {
        List<Order> orders = orderRepository.findAll();
        double averageDeliveryTime = orders.stream()
            .mapToDouble(Order::getDeliveryTime)
            .average()
            .orElse(0.0);
    }
}
```

```

        return new DeliveryAnalytics(averageDeliveryTime);
    }
}

```

Цей сервіс вираховує середній час доставки на основі даних замовлень, що дозволяє адміністрації системи вживати заходів для покращення ефективності служби доставки.

Для фронтенду системи управління вантажоперевезеннями, використано React.js разом з додатковими бібліотеками для створення інтерактивного користувацького інтерфейсу [49, 50]. Наприклад:

- Навігаційна панель (Navbar), що дозволяє користувачам легко переміщатися по різних частинах додатку. Вона містить посилання на головну сторінку, сторінку замовлень, контакти та інші важливі розділи.

```

import React from 'react';
import { Link } from 'react-router-dom';

function Navbar() {
    return (
        <nav>
            <ul>
                <li><Link to="/">Home</Link></li>
                <li><Link to="/orders">Orders</Link></li>
                <li><Link to="/contact">Contact</Link></li>
            </ul>
        </nav>
    );
}

export default Navbar;

```

- Маршрутизація (Routing) за допомогою бібліотеки React Router, що дозволяє визначити маршрути для різних компонентів, які мають відображатися на певних URL-адресах.

```

import React from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

```

```

import Home from './components/Home';
import Orders from './components/Orders';
import Contact from './components/Contact';

function App() {
  return (
    <Router>
      <Navbar />
      <Switch>
        <Route path="/" exact component={Home} />
        <Route path="/orders" component={Orders} />
        <Route path="/contact" component={Contact} />
      </Switch>
    </Router>
  );
}

export default App;

```

- Форми для введення даних реалізовані за допомогою хуків [51].

```

import React, { useState } from 'react';

function OrderForm() {
  const [order, setOrder] = useState({ description: "", status: 'Pending' });

  const handleSubmit = (event) => {
    event.preventDefault();
    fetch('/api/orders', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(order)
    }).then(() => {
      alert('Order submitted!');
      setOrder({ description: "", status: 'Pending' });
    }).catch(error => {
      console.error('Error submitting order:', error);
    });
  };

  return (
    <form onSubmit={handleSubmit}>
      <label>

```



```

    Description:
    <input
      type="text"
      value={order.description}
      onChange={e => setOrder({ ...order, description: e.target.value })}
    />
  </label>
  <label>
    Status:
    <select
      value={order.status}
      onChange={e => setOrder({ ...order, status: e.target.value })}
    >
      <option value="Pending">Pending</option>
      <option value="Completed">Completed</option>
    </select>
  </label>
  <button type="submit">Submit</button>
</form>
);
}

export default OrderForm;

```

Ці основні та інші можливості налаштування бекенд та фронтенд частин разом із правильно структуризованою та інтегрованою базою даних забезпечують стабільну та ефективну взаємодію користувачів з системою управління вантажоперевезеннями. Більш детальний лістинг програмного коду представлено в додатку Б.

5.3 Тестування системи

Робота з системою починається тільки після авторизації користувача, щоб захистити дані від небажаного несанкціонованого доступу. Форма авторизації представлена на рисунку 5.1.

Увійти

Емейл(логін):

Введіть ваш емейл або логін

Пароль:

.....

[Забули пароль?](#)

Увійти

Ще не зареєстровані? [Зареєструватися](#)

Рисунок 5.1 – Форма авторизації в системі

На рисунках 5.2 та 5.3 представлено відображення інформації з профілю користувача та компанії відповідно. Користувач має можливість редагувати власні дані, в той час як дані компанії може редагувати лише адміністратор компанії.

Автоперевезення

[мої перевезення](#) | [лодати вантаж](#) | [лодати транспорт](#) | [пошук вантажів і транспорту](#)

Д. Дмитро

Організація | **Профіль**

Профіль

Інформація про мій профіль

Основна інформація

Моя основна інформація - ім'я, прізвище і місто в якому я перебуваю

Ім'я* Дмитро

Прізвище* Дмитро

Зберегти

Пароль

Паролі зберігаються в зашифрованому вигляді, ми не можемо показати його в налаштуваннях. Якщо ви хочете змінити пароль натисніть 'Змінити пароль'

Змінити пароль

Контактна інформація

Ви можете використовувати цю контактну інформацію при розміщенні вантажів і транспорту.

Телефон: +380(6...)

E-Mail: ...mail.com

Рисунок 5.2 – Відображення інформації про користувача

Автоперевезення

мої перевезення | додати вантаж | додати транспорт | пошук вантажів і транспорту

ДД Дмитро

Організація | Профіль

Організація
Інформація про мою організацію

Загальна інформація
Загальна інформація про мою організацію або бізнес. Ця інформація доступна для перегляду іншим користувачам.

ДМИТРО ДМИТРО
📍 Україна 📅 06/10/2024

Загальний Рейтинг Ⓜ
☆☆☆☆

Контактна інформація
Ви можете використовувати цю контактну інформацію при розміщенні вантажів і транспорту.

Телефон: +380(6) [redacted] E-Mail: [redacted]@mail.com

Рисунок 5.3 – Відображення інформації про компанію

На рисунку 5.4 зображено процес додавання картки вантажу, яка може зберігатися як артефакт компанії для повторного використання при наступних створеннях заявок з вантажем. Створену раніше картку товару та заявку можуть редагувати та видаляти лише адміністратори компанії.

Автоперевезення

мої перевезення | додати вантаж | додати транспорт | пошук вантажів і транспорту

ДД Дмитро

Додавання вантажу

Тип вантажу: Тип транспорту:

вага вантажу (т): Кількість машин:

об'єм вантажу (м³):

☐ Без довантаження (окреме авто) ☐ Довантаження ☐ Можл. довантаження

довжина: м ширина: м висота: м

Розмістити

Рисунок 5.4 – Додавання картки вантажу

На рисунку 5.5 представлено процес створення заявки з вантажем. Форма має додаткові поля, що були описані як атрибути таблиці Orders в розділі про розробку структури бази даних.

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Додавання заявки на перевезення вантажу
вкажіть будь-ласка населені пункти завантаження - розвантаження, параметри вантажу та контактну інформацію

Завантаження: з 26.11.2024 до 27.11.2024

Нас. пункт завантаження: Вінниця

Розвантаження

Нас. пункт розвантаження: Чоп

Тип вантажу:

вага вантажу (т): 0,3

об'єм вантажу (м³): 2

Тип транспорту: крита

Кількість машин: 1

☐ Без довантаження (окремe авто) ☐ Довантаження ☐ Можл. дозавантаження

довжина: м ширина: м висота: м

Вартість: 2000 грн всього

E-mail: .ail.com

Телефон: +380(6)

Додаткова інформація: [вказати](#)

Розмістити

Рисунок 5.5 – Створення заявки з вантажем

На рисунках 5.6 та 5.7 аналогічно зображено додавання картки транспортного засобу та створення заявки з транспортним засобом відповідно. Після розміщення заявок, вони одразу стають доступними для перегляду та замовлення всіма авторизованими користувачами системи.

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Розміщення транспорту

Тип транспорту: тент

☒ вантажівка ☐ напівпричіп ☐ з причепом

Кількість машин: 1

Характеристики транспортного засобу:

Вантажопідйомність (т): 6

Корисний об'єм (м³): 43

довжина: 6,8 м ширина: 2,45 м висота: 2,6 м

Розмістити

Рисунок 5.6 – Додавання картки транспортного засобу

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Розміщення інформації про вільний транспорт
вкажіть бажані пункти завантаження - розвантаження, параметри транспортного засобу та контактну інформацію

Завантаження: з 26.11.2024 до 27.11.2024

Місце завантаження: Вінниця

Розвантаження

Місце розвантаження: Київська обл.

Тип транспорту:
☒ вантажівка
☐ напівпричіп
☐ з причепом

Характеристики транспортного засобу:
Вантажопідйомність (т):
Корисний об'єм (м³):

Кількість машин: довжина: м ширина: м висота: м

Вартість: валюта не вказана за кілометр

E-mail: **Телефон:** +380()

Додаткова інформація: [вказати](#)

Рисунок 5.7 – Створення заявки з транспортним засобом

На рисунку 5.8 представлено відображення активних перевезень (заявок) користувача. Детальну інформацію про них та поточний стан можна переглянути, відкривши картку заявки.

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Мої активні перевезення
Активні перевезення, доступні іншим користувачам в пошуку

🕒 5 годин тому

📅 26.11–27.11 🚚 тент 📊 6 т 📦 43 м³

Вінниця (UA) — Київська обл. (UA)

(дов=6,8м шир=2,45м вис=2,6м)

🕒 9 годин тому

📅 26.11–27.11 🚚 крита 📊 0,3 т 📦 2 м³

Вінниця (UA) — Чоп (UA) ~ 642 км

двері 2 000 грн
3,12 грн/км

Рисунок 5.8 – Відображення активних перевезень користувача

На рисунку 5.9 представлено допоміжний функціонал для користувачів системи – розрахунок відстаней між містами. Цей функціонал дозволяє побудову маршруту з великої кількості проміжних пунктів та автоматичне обрахування тривалості та відстані маршруту. Інтегрована карта від Google прокладе необхідний маршрут, відкрившись у новому вікні.

The screenshot displays a web application interface for 'Автоперевезення' (Auto Transport). At the top, there are navigation links: 'мої перевезення', 'додати вантаж', 'додати транспорт', and 'пошук вантажів і транспорту'. A user profile 'ДД Дмитро' is visible in the top right corner.

The main section is titled 'Розрахунок відстаней між містами України' (Distance calculation between cities in Ukraine). It features two input fields for locations:

- Звідки** (From): 'Вінниця, Вінницька область, Україна' (Vinnytsia, Vinnytsia region, Ukraine). Below it is a '+ Додати пункт' (Add point) button.
- Куди** (To): 'Київ, Київська область, Україна' (Kyiv, Kyiv region, Ukraine). Below it is a '+ Додати пункт' (Add point) button.

 A 'Розрахувати' (Calculate) button is positioned below the input fields.

Below the input fields, the 'Деталі маршруту:' (Route details:) section shows:

- Route: 'Вінниця (UA) — Київ (UA)'
- Distance: 'Відстань за маршрутом ~ 267 км' (Distance along the route ~ 267 km)
- Estimated travel time: 'Приблизний час поїздки ~ 3 ч 17 м' (Estimated travel time ~ 3 h 17 m)

 A visual route representation shows a line connecting 'Вінниця (UA)' and 'Київ (UA)' with the distance and time labels.

At the bottom, there is a map view with a button 'Дивитися маршрут на карті' (View route on map).

Рисунок 5.9 – Відображення розрахунку відстаней між містами

Також для пришвидшення пошуку вантажів та транспортних засобі система має віджети для пошуку, що представлені на рисунку 5.10, а також віджет з цінами на останні вантажні перевезення зареєстровані, в системі. Останній з згаданих віджетів оновлює дані одразу після появи нових заявок.

Пошук попутних вантажів

Звідки: Україна (UA) ▼
всі обл. ▼

Куди: Україна (UA) ▼
всі обл. ▼

Знайти

Пошук попутних машин

Звідки: Україна (UA) ▼
всі обл. ▼

Куди: Україна (UA) ▼
всі обл. ▼

Знайти

Рисунок 5.10 – Віджети для швидкого пошуку вантажів та транспортних засобів

Ціни на вантажні перевезення

Дніпро — Запоріжжя 4 т	
3 800 грн	44,71 грн/км
Львів — Одеса 22 т	
24 500 грн	29,77 грн/км

Показати всі ціни

Рисунок 5.11 – Віджет з цінами на останні вантажні перевезення зареєстровані в системі

5.4 Висновки

Під час тестування системи було перевірено всі ключові функції, включаючи авторизацію, управління профілями, створення та управління заявками, а також допоміжні інструменти, такі як розрахунок відстаней і пошукові віджети. Наразі система налаштована лише на українські населені пункти, проте в майбутньому можливе розширення географічного охоплення, що дозволить здійснювати міжнародні перевезення з використанням тих же інструментів і віджетів.

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи було проведено дослідження та аналіз системи управління збірними вантажоперевезеннями з метою підвищення ефективності логістичних процесів та оптимізації взаємодії з іншими логістичними платформами. Метою роботи було створення інструменту, спрямованого на оптимізацію управління та автоматизацію ключових процесів.

Проведено аналіз основних аспектів автоматизації системи управління збірними вантажоперевезеннями, досліджено поточні методи управління, особливості інтеграції з іншими логістичними платформами та можливості автоматизації процесів відстеження та контролю. Також було проаналізовано алгоритми оптимізації маршрутів та систему управління ризиками.

Проведено аналіз та порівняння існуючих систем управління збірними вантажоперевезеннями, де були вивчені їхні функціональні можливості, переваги та обмеження.

Виконано проектування автоматизованої системи, включаючи визначення вимог, розробку архітектури системи, проектування бази даних та розробку UML-діаграм.

Реалізовано функціональність системи та проведено її тестування, результати якого підтвердили ефективність розробленої системи та її придатність для використання в управлінні збірними вантажоперевезеннями. Робота виконана в повному обсязі, відповідає поставленим завданням та меті, а створена система за всіма характеристиками заслуговує бути впровадженою у сферу автоматизації управління збірними вантажоперевезеннями.

ПЕРЕЛІК ПОСИЛАНЬ

1. Дуна Н., Матвієнко А. Перспективи розвитку українського ринку автомобільних вантажоперевезень: євроінтеграційний аспект // Науковий вісник Ужгородського національного університету. – 2022. – Випуск 44. – С. 21–29.
2. Шевченко Г.І., Бондаренко О.М. Екологічні аспекти у системах збірних вантажоперевезень // Екологія та промисловість України. – 2021. – № 2. – С. 45-49.
3. Фариняк Д. Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями. [Електронний ресурс]. – URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/author/submission/22506>
4. Мороз П.Ю., Литвиненко І.Л. Інтеграція інформаційних систем у логістиці: виклики та рішення // Вісник економіки транспорту і промисловості. – 2022. – № 3. – С. 74-78.
5. Білик М.О., Кузнєцова Т.В. Використання PostgreSQL для управління даними // Інформаційні технології в транспорті. – 2017. – № 1. – С. 58-62.
6. Стасюк О. Ринки вантажних та пасажирських перевезень в Україні: проблеми та тенденції // Ефективна економіка. – 2020. – № 9. – С. 1–19. – Режим доступу: http://www.economy.nayka.com.ua/pdf/9_2020/56.pdf. – Дата звернення: 10.09.2024.
7. Parker, R. & Lewis, H. Global Trends in Consolidated Transportation // International Journal of Logistics Management. – 2008. – Vol. 19, No. 1. – Pp. 45-62.
8. Грищенко В.І. Інформаційні технології в логістиці: переваги та ризики // Журнал управління ланцюгами постачання. – 2013. – № 3. – С. 34-42.
9. Козак Л.В., Мельник Л.Г. ERP системи в логістиці: переваги та виклики // Журнал управління ланцюгами постачання. – 2018. – № 2. – С. 30-36.
10. Овчарова В.І. Оптимізація складських процесів за допомогою WMS // Логістика сьогодні. – 2012. – № 1. – С. 42-47.

- 11.Рисунок взаємодії компонентів у сфері управління вантажними перевезеннями [Електронний ресурс] – Режим доступу: <https://de.linkedin.com/in/karthik-appari>. – Дата звернення: 28.09.2024.
- 12.Thompson, J., & Lee, M. Logistics Innovation in Asia: The Impact of Technology on Supply Chain Management // Asian Journal of Logistics. – 2023. – Vol. 35, No. 3. – Pp. 145-160.
- 13.Anderson, R., & Petrov, I. Global Standards and Local Practices in Ukrainian Logistics // European Journal of Logistics. – 2021. – Vol. 29, No. 1. – Pp. 77-92.
- 14.Williams, S. & Carter, D. Integrating Logistics Systems through API Technology // International Journal of Logistics and IT Integration. – 2021. – Vol. 12, No. 2. – Pp. 234-249.
- 15.Martin, R. & Thompson, H. The Role of EDI in Modern Logistics // Logistics Technology Review. – 2020. – Vol. 15, No. 4. – Pp. 112-127.
- 16.Ковальчук О.М. Стратегії інтеграції логістичних систем: переваги та виклики // Журнал сучасної логістики. – 2022. – № 3. – С. 78-92.
- 17.Якайтис І.Б. Стратегічні вектори розвитку міжнародних перевезень в Україні. Інвестиції: практика та досвід. № 21-22, 2020. С. 124-128.
- 18.Краснікова О.М. Безпека даних у логістичних системах: проблеми та рішення // Журнал інформаційної безпеки. – 2022. – № 2. – С. 34-42.
- 19.Генсіцький Л.Д. Управління змінами при інтеграції ІТ-систем: стратегії та практики // Журнал управління змінами. – 2023. – № 1. – С. 54-65.
- 20.Мельник Л.Г. Сучасні технології відстеження в логістиці. – Київ: Видавництво "Логос", 2021.
- 21.Кравченко В.А. Інтеграція та автоматизація логістичних систем. – Харків: Форт, 2022.
- 22.Goldsmith, J. Advanced Algorithms for Route Optimization // Journal of Computational Logistics. – 2021. – Vol. 17, No. 3. – Pp. 45-62.
- 23.Thompson, R. & Lee, S. Simulation and Testing of Routing Algorithms Using MATLAB // International Journal of Simulation Systems. – 2022. – Vol. 29, No. 1. – Pp. 134-150.

24. Patel, A. & Kumar, V. Optimization Techniques in Logistics: From Theory to Practice. – Springer, 2021.
25. Brown, A. Risk Management in Business: Theory and Practice. – Cambridge University Press, 2021.
26. Johnson, R. Tools for Effective Risk Management // Journal of Risk Analysis. – 2022. – Vol. 22, No. 4. – Pp. 210-225.
27. Taylor, M. & Harris, P. Implementing Information Security Measures: A Practical Guide. – Wiley, 2022.
28. Система SAP LES [Електронний ресурс] – Режим доступу: https://help.sap.com/docs/SAP_ERP/62ee292c419c41a9ab9609d73af0aa37/7c5c5e53118d4308e10000000a174cb4.html?version=6.18.latest. – Дата звернення: 25.09.2024.
29. Система Oracle OTM [Електронний ресурс] – Режим доступу: <https://www.oracle.com/scm/logistics/transportation-management/>. – Дата звернення: 26.09.2024.
30. Система JDA (Blue Yonder) TM [Електронний ресурс] – Режим доступу: <https://blueyonder.com/solutions/transportation-management>. – Дата звернення: 27.09.2024.
31. Система Manhattan Associates TM [Електронний ресурс] – Режим доступу: <https://www.manh.com/solutions/supply-chain-management-software/transportation-management>. – Дата звернення: 28.09.2024.
32. Іваненко В.М. Проектування автоматизованих систем управління логістикою. – Київ: Техніка, 2022.
33. Кашубська О.С. Інноваційні технології в логістиці. – Львів: Астролябія, 2021.
34. Ковальчук Я.Б. Системи управління вантажоперевезеннями: від теорії до практики. – Одеса: Морська друкарня, 2023.
35. Козлов Д.О. Використання UML-діаграм для проектування розподілених систем // Науковий журнал "Інформаційні технології та системи". – 2022. – Випуск 12. – С. 45-59.
36. Brown, A. & Davis, B. Relational Database Design and Implementation. – 4th Edition. – Elsevier, 2021.

37. Thompson, S. Database Systems: The Complete Book. – 2nd Edition. – Pearson, 2022.
38. Braker, M. PostgreSQL: Up and Running. – 3rd Edition. – O'Reilly Media, 2023.
39. Rosenberg, L. & Scott, K. Use Case Driven Object Modeling with UML: Theory and Practice. – Apress, 2021.
40. Pilone, D. & Pitman, N. UML 2.0 in a Nutshell. – O'Reilly Media, 2022.
41. Miles, R. Learning UML 2.0: A Pragmatic Introduction to UML. – O'Reilly Media, 2022.
42. Banker, K. Pro React 16. – Apress, 2021.
43. Smith, J. Mastering IntelliJ IDEA: Maximize IDE productivity. – Packt Publishing, 2023.
44. Turnbull, J. "Containerization and Docker: Revolutionizing Software Deployment" // Journal of Software Engineering and Applications. – 2022. – Vol. 15, No. 3. – Pp. 112-128.
45. Richards, M. "Comparative Analysis of Microservices and Service-Oriented Architectures" // International Journal of Advanced Computer Science. – 2022. – Vol. 12, No. 4. – Pp. 234-250.
46. Newman, S. "The Impact of Microservices on Enterprise Architecture" // Journal of System Architecture. – 2023. – Vol. 59, No. 1. – Pp. 41-56.
47. Wallace, C. & Gutz, S. Effective REST Services via Spring. – Apress, 2021.
48. Kiczales, G. Aspect-Oriented Programming. – MIT Press, 2022.
49. Banks, A. & Porcello, E. Learning React: Modern Patterns for Developing React Apps. – 2nd Edition. – O'Reilly Media, 2022.
50. Fedosejev, A. React.js Essentials. – Packt Publishing, 2021.
51. Hahn, S. Hooks at a Glance: Building Dynamic React Apps with Hooks. – Independent, 2023.
52. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів <https://iq.vntu.edu.ua/repository/getfile.php/6077.pdf>

ДОДАТКИ

Додаток А
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: «Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями»

Тип роботи: магістерська кваліфікаційна робота
(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше (вказати))

Підрозділ кафедра комп'ютерних систем управління
(кафедра, факультет (інститут), навчальна група)

Керівник Гришук Т.В., доцент, кафедри КСУ
(прізвище, ініціали, посада)

Показники звіту подібності

Turnitin	
Оригінальність	97%
Загальна схожість	3%

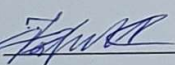
Аналіз звіту подібності (відмітити потрібне)

✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

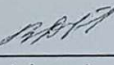
Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор  Дмитро ФАРИНЯК
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

Особа, відповідальна за перевірку  Володимир ДУБОВОЙ
(підпис) (прізвище, ініціали)

Експерт _____
(за потреби) (підпис) (прізвище, ініціали, посада)

Додаток Б
(обов'язковий)

ВНТУ

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., професор

 В'ячеслав КОВТУН

“ 14 ” 10 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

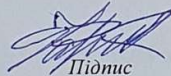
на виконання магістерської кваліфікаційної роботи

Розробка автоматизованої системи для ефективного управління

збірними вантажоперевезеннями

08-33.МКР.009.00.000 ТЗ

Студент групи ЗАКІТР-23м

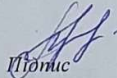
 Підпис

Дмитро ФАРИНЯК

Ім'я ПРИЗВИЩЕ

Керівник

к.т.н., доц. каф. КСУ

 Підпис

Тетяна ГРИЩУК

Ім'я ПРИЗВИЩЕ

Вінниця 2024

1. Назва та галузь застосування

1.1. Назва – Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями.

1.2. Галузь застосування – Інформаційні технології у сфері логістики, спеціалізовані платформи для автоматизації процесів планування, відстеження та управління збірними вантажоперевезеннями.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ № 310 від 17-09-2024 р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є розробка та впровадження автоматизованої системи управління збірними вантажоперевезеннями, спрямованої на підвищення ефективності логістичних процесів, зниження витрат та покращення якості обслуговування клієнтів, що дозволить компаніям оптимізувати управління вантажопотоками, забезпечити високий рівень інтеграції з іншими системами та підвищити загальну конкурентоспроможність на ринку.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Якайтис І.Б. Стратегічні вектори розвитку міжнародних перевезень в Україні. Інвестиції: практика та досвід. № 21-22, 2020. С. 124-128.
2. Ананко І.М. Оцінка ефективності функціонування логістичних систем на АТП / І.М. Ананко // Економіка транспортного комплексу. - 2014. - №24. - с. 125-138.
3. Turnbull, J. "Containerization and Docker: Revolutionizing Software Deployment" // Journal of Software Engineering and Applications. – 2022. – Vol. 15, No. 3. – Pp. 112-128..

5. Вимоги до розробки.

5.1. Перелік головних етапів виконання розробки:

- визначити аспекти функціонування системи управління збірними вантажоперевезеннями;
- дослідити функціональні можливості існуючих аналогічних систем у сфері логістики;
- скласти вимоги до автоматизованої системи, що розробляється;
- спроектувати та розробити систему у відповідності до поставлених вимог та функцій:
 - забезпечення доступу користувачів до системи та їхніх персональних акаунтів;
 - управління маршрутами та графіками вантажоперевезень;
 - відстеження статусу вантажів у реальному часі;

- звітність та аналітика для оптимізації логістичних процесів;
- інтеграція з іншими системами та сервісами для забезпечення комплексного управління ланцюгом поставок.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- підтримка веб-платформи.

5.2.2. Умови експлуатації системи:

- робота системи у режимі 24/7 з високою доступністю та надійністю;
- система повинна бути масштабованою для обробки великих обсягів даних та здатною адаптуватися до зростаючих потреб бізнесу.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

- Дослідження об'єкта автоматизації та постановка задач дослідження 03.09.2024р.
- Дослідження основних аспектів автоматизації системи управління збірними вантажоперевезеннями 19.09.2024р.
- Аналіз і порівняння існуючих систем управління збірними вантажоперевезеннями 01.10.2024р.
- Проектування автоматизованої системи управління збірними вантажоперевезеннями 15.11.2024р.
- Апробація результатів дослідження 20.11.2024р.
- Оформлення пояснювальної записки, графічного матеріалу і презентації 29.11.2024р.

6.2 Графічні матеріали:

- Use-case UML-діаграми 04.10. 2024 р.
- ER-діаграма структури бази даних 18.10. 2024 р.
- вигляд екранів розробленого web-додатку 18.11. 2024 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «20»__11__ 2024 р.
- 7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «1»__12__ 2024 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «12»__12__ 2024 р.

Додаток В (вибірковий)

Лістинг програмного забезпечення

Spring Boot Модель Order

```
package com.example.transportation.model;

import javax.persistence.*;
import java.util.Date;

@Entity
@Table(name = "orders")
public class Order {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
```

```
@Column(nullable = false)
private String description;
```

```
@Column(nullable = false)
private Date orderDate;
```

```
@Column(nullable = false)
private String status;
```

// Getters and Setters

```
public Long getId() {
    return id;
}
```

```
public void setId(Long id) {
    this.id = id;
}
```

```
public String getDescription() {
    return description;
}
```

```
public void setDescription(String description) {
    this.description = description;
}
```

```
public Date getOrderDate() {
    return orderDate;
}
```

```
public void setOrderDate(Date orderDate) {
    this.orderDate = orderDate;
}
```

```
public String getStatus() {
    return status;
}
```

```
public void setStatus(String status) {
    this.status = status;
}
```

Репозиторій OrderRepository

```
package com.example.transportation.repository;
```

```
import com.example.transportation.model.Order;
import
```

```
org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

```
@Repository
```

```
public interface OrderRepository extends
JpaRepository<Order, Long> {
}
```

Контролер OrderController

```
package com.example.transportation.controller;
```

```
import com.example.transportation.model.Order;
import
```

```
com.example.transportation.repository.OrderRepository;
import
```

```
org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
```

```
import java.util.List;
```

```
@RestController
```

```
@RequestMapping("/api/orders")
```

```
public class OrderController {
```

```

@Autowired
private OrderRepository orderRepository;

@GetMapping
public List<Order> getAllOrders() {
    return orderRepository.findAll();
}

@PostMapping
public Order createOrder(@RequestBody Order
order) {
    return orderRepository.save(order);
}

@PutMapping("/{id}")
public ResponseEntity<Order>
updateOrder(@PathVariable Long id, @RequestBody Order
orderDetails) {
    Order order = orderRepository.findById(id)
        .orElseThrow(() -> new
RuntimeException("Order not found with id " + id));

    order.setDescription(orderDetails.getDescription());
    order.setStatus(orderDetails.getStatus());

    order.setOrderDate(orderDetails.getOrderDate());
    final Order updatedOrder =
orderRepository.save(order);
    return ResponseEntity.ok(updatedOrder);
}

```

React Компонент App

```

import React from 'react';
import { BrowserRouter as Router, Route, Switch }
from 'react-router-dom';

import Home from './components/Home';
import Orders from './components/Orders';
import Navbar from './components/Navbar';

function App() {
    return (
        <Router>
            <Navbar />
            <Switch>
                <Route path="/" exact
component={Home} />
                <Route path="/orders"

```

```

component={Orders} />
            </Switch>
        </Router>
    );
}

```

export default App;

Компонент Navbar

```

import React from 'react';
import { Link } from 'react-router-dom';

function Navbar() {
    return (
        <nav>
            <ul>
                <li><Link to="/">Home</Link></li>
                <li><Link
to="/orders">Orders</Link></li>
            </ul>
        </nav>
    );
}

```

export default Navbar;

Компонент Orders

```

import React, { useState, useEffect } from 'react';

function Orders() {
    const [orders, setOrders] = useState([]);

    useEffect(() => {
        fetch('/api/orders')
            .then(response => response.json())
            .then(data => setOrders(data))
            .catch(error => console.error('Error fetching
orders:', error));
    }, []);

    return (
        <div>
            <h1>Orders</h1>
            <ul>
                {orders.map(order => (
                    <li key={order.id}>{order.description}
- {order.status}</li>
                ))}
            </ul>
        </div>
    );
}

```

```

    </ul>
  </div>
);
}

export default Orders;

Cepbic OrderService
  package com.example.transportation.service;

  import com.example.transportation.model.Order;
  import
com.example.transportation.repository.OrderRepository;
  import
org.springframework.beans.factory.annotation.Autowired;
  import org.springframework.stereotype.Service;

  import java.util.List;

  @Service
  public class OrderService {
    @Autowired
    private OrderRepository orderRepository;

    public List<Order> getAllOrders() {
      return orderRepository.findAll();
    }

    public Order saveOrder(Order order) {
      return orderRepository.save(order);
    }

    public Order updateOrder(Long id, Order
orderDetails) {
      Order order = orderRepository.findById(id)
        .orElseThrow(() -> new
RuntimeException("Order not found with id " + id));

      order.setDescription(orderDetails.getDescription());
      order.setStatus(orderDetails.getStatus());

      order.setOrderDate(orderDetails.getOrderDate());
      return orderRepository.save(order);
    }
  }

```

Винятки CustomException

```

package com.example.transportation.exception;

public class CustomException extends
RuntimeException {
  public CustomException(String message) {
    super(message);
  }
}

Компонент OrderDetails
  import React, { useState, useEffect } from 'react';

  function OrderDetails({ match }) {
    const [order, setOrder] = useState(null);

    useEffect(() => {
      fetch(`/api/orders/${match.params.id}`)
        .then(response => response.json())
        .then(data => setOrder(data))
        .catch(error => console.error('Error fetching
order:', error));
    }, [match.params.id]);

    return (
      <div>
        {order ? (
          <div>
            <h1>Order Details</h1>
            <p>Description:
{order.description}</p>
            <p>Status: {order.status}</p>
            <p>Order Date: {order.orderDate}</p>
          </div>
        ) : (
          <p>Loading...</p>
        )}
      </div>
    );
  }

export default OrderDetails;

Маршрутизація для OrderDetailsc
  import React from 'react';
  import { BrowserRouter as Router, Route, Switch }
from 'react-router-dom';
  import Home from './components/Home';

```

```

import Orders from './components/Orders';
import OrderDetails from './components/OrderDetails';
import Navbar from './components/Navbar';

function App() {
  return (
    <Router>
      <Navbar />
      <Switch>
        <Route path="/" exact
component={Home} />
        <Route path="/orders" exact
component={Orders} />
        <Route path="/orders/:id"
component={OrderDetails} />
      </Switch>
    </Router>
  );
}

```

export default App;

Контролер для статистики StatisticsController

```

package com.example.transportation.controller;

import
com.example.transportation.service.OrderService;
import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController
@RequestMapping("/api/statistics")
public class StatisticsController {
  @Autowired
  private OrderService orderService;

  @GetMapping("/order-status")
  public Map<String, Long>
getStatusStatistics() {

```

```

    return orderService.getOrderStatusStatistics();
  }
}

```

Метод у OrderService для статистики

```

package com.example.transportation.service;

import
com.example.transportation.repository.OrderRepository;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Map;
import java.util.stream.Collectors;

@Service
public class OrderService {
  @Autowired
  private OrderRepository orderRepository;

  public Map<String, Long>
getStatusStatistics() {
    return orderRepository.findAll().stream()
      .collect(Collectors.groupingBy(Order::getStatus,
        Collectors.counting()));
  }
}

Компонент Statistics
import React, { useState, useEffect } from 'react';

function Statistics() {
  const [stats, setStats] = useState({});

  useEffect(() => {
    fetch('/api/statistics/order-status')
      .then(response => response.json())
      .then(data => setStats(data))
      .catch(error => console.error('Error fetching
statistics:', error));
  }, []);

  return (
    <div>
      <h1>Order Statistics</h1>
      <ul>

```

```

        {Object.entries(stats).map(([status, count])
=> (
            <li key={status}>{status}: {count}</li>
        ))}
    </ul>
</div>
);
}

```

```
export default Statistics;
```

Додавання маршруту для статистики

```

import React from 'react';
import { BrowserRouter as Router, Route, Switch }
from 'react-router-dom';
import Home from './components/Home';
import Orders from './components/Orders';
import      OrderDetails      from
'./components/OrderDetails';
import Statistics from './components/Statistics';
import Navbar from './components/Navbar';

function App() {
    return (
        <Router>
            <Navbar />
            <Switch>
                <Route      path="/"      exact
component={Home} />
                <Route      path="/orders"  exact
component={Orders} />
                <Route      path="/orders/:id"
component={OrderDetails} />
                <Route      path="/statistics"
component={Statistics} />
            </Switch>
        </Router>
    );
}

```

```
export default App;
```

Методи в OrderService для нових запитів

```

package com.example.transportation.service;

import com.example.transportation.model.Order;
import
com.example.transportation.repository.OrderRepository;

```

```

import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Date;
import java.util.List;

@Service
public class OrderService {

    @Autowired
    private OrderRepository orderRepository;

    public List<Order> findOrdersByStatus(String
status) {

        return orderRepository.findByStatus(status);
    }

    public List<Order> findOrdersByDateRange(Date
startDate, Date endDate) {

        return
orderRepository.findByOrderDateBetween(startDate,
endDate);
    }
}

```

Контролер для фільтрації замовлень OrderSearchController

```

package com.example.transportation.controller;

import com.example.transportation.model.Order;
import
com.example.transportation.service.OrderService;
import
org.springframework.beans.factory.annotation.Autowired;
import
org.springframework.format.annotation.DateTimeFormat;
import org.springframework.web.bind.annotation.*;

import java.util.Date;
import java.util.List;

@RestController
@RequestMapping("/api/search/orders")
public class OrderSearchController {

    @Autowired
    private OrderService orderService;

    @GetMapping("/status")
    public List<Order>

```

```

getOrdersByStatus(@RequestParam String status) {
    return orderService.findOrdersByStatus(status);
}

@GetMapping("/date-range")
public List<Order> getOrdersByDateRange(
    @RequestParam @DateTimeFormat(iso =
DateTimeFormat.ISO.DATE) Date startDate,
    @RequestParam @DateTimeFormat(iso =
DateTimeFormat.ISO.DATE) Date endDate) {
    return
orderService.findOrdersByDateRange(startDate, endDate);
}
}

```

docker-compose.yml

```

version: '3.8'
services:
  db:
    image: postgres:13
    environment:
      POSTGRES_DB: transportation_db
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
    ports:
      - "5432:5432"
    volumes:
      - pgdata:/var/lib/postgresql/data

  backend:
    build:
      context: ./backend

```

dockerfile: Dockerfile

```

ports:
  - "8080:8080"
depends_on:
  - db

```

frontend:

```

build:
  context: ./frontend
  dockerfile: Dockerfile
ports:
  - "3000:3000"
depends_on:
  - backend

```

volumes:

pgdata:

```

# PostgreSQL Configuration
spring.datasource.url=jdbc:postgresql://localhost:54
32/transportation_db
spring.datasource.username=user
spring.datasource.password=password

# JPA Configuration
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.P
ostgreSQLDialect

```

Додаток Г
(обов'язковий)

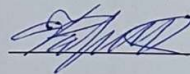
ІЛЮСТРОВАНА ЧАСТИНА

«Розробка автоматизованої системи для ефективного управління збірними вантажоперевезеннями»

1. UML-діаграма компонентів автоматизованої системи управління збірними вантажоперевезеннями
2. UML-діаграми розгортання технологічної структури автоматизованої системи управління збірними вантажоперевезеннями
3. Концептуальна схема бази даних автоматизованої системи управління збірними вантажоперевезеннями
4. ER-діаграма структури бази даних автоматизованої системи управління збірними вантажоперевезеннями
5. Use Case UML-діаграма для користувача автоматизованої системи управління збірними вантажоперевезеннями
6. Class UML-діаграма автоматизованої системи управління збірними вантажоперевезеннями
7. UML-діаграма послідовності дій користувача в автоматизованій системі управління збірними вантажоперевезеннями
8. Форма авторизації в системі
9. Відображення інформації про користувача
10. Відображення інформації про компанію
11. Додавання картки вантажу
12. Створення заявки з вантажем
13. Додавання картки транспортного засобу
14. Створення заявки з транспортним засобом
15. Відображення активних перевезень користувача
16. Відображення розрахунку відстаней між містами
17. Віджети для швидкого пошуку вантажів та транспортних засобів
18. Віджет з цінами на останні вантажні перевезення зареєстровані в системі

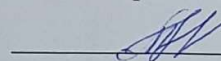
Виконав: студент 2-го курсу,
групи ЗАКІТР-23м
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка

(шифр і назва спеціальності)



Дмитро ФАРИНЯК
(ім'я та прізвище)

Керівник: к.т.н., доц., каф. КСУ



Тетяна ГРИЩУК
(ім'я та прізвище)

« 24 » 12 2024 р.

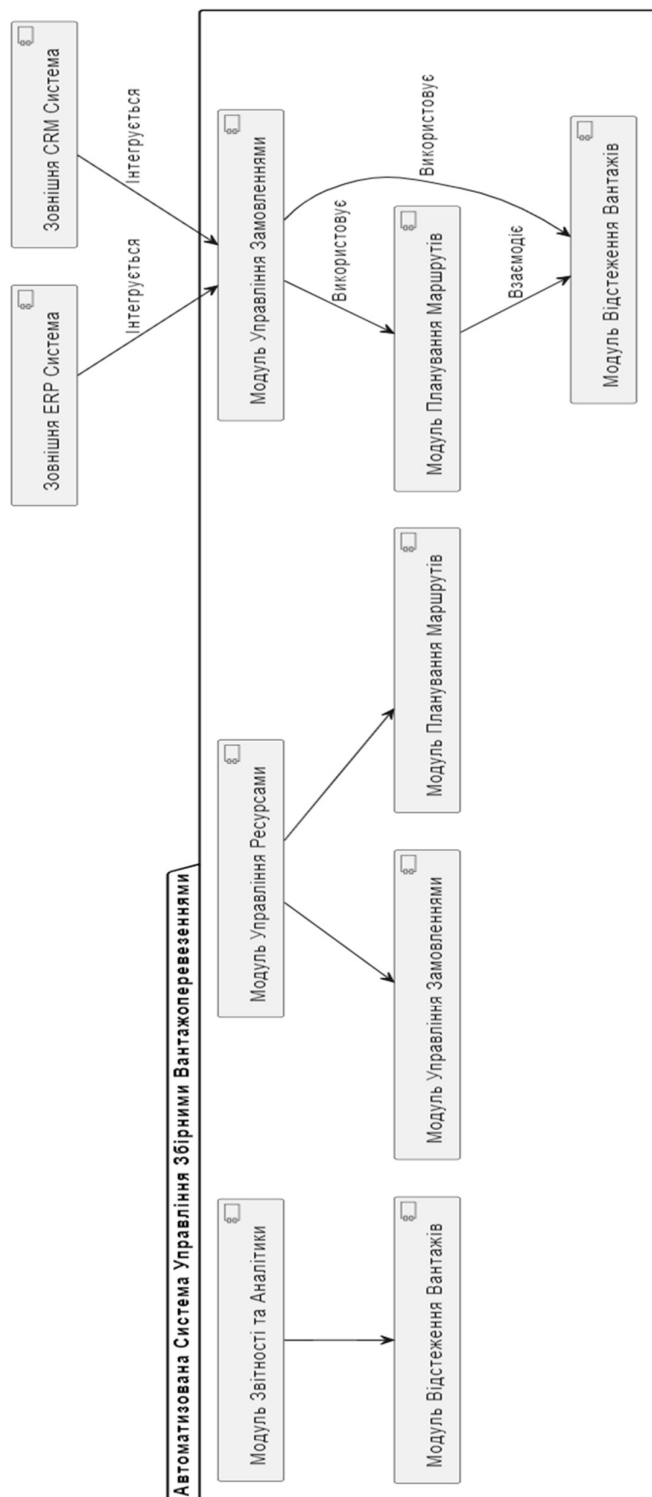
Опонент: к.т.н., доцент., каф. АІТ



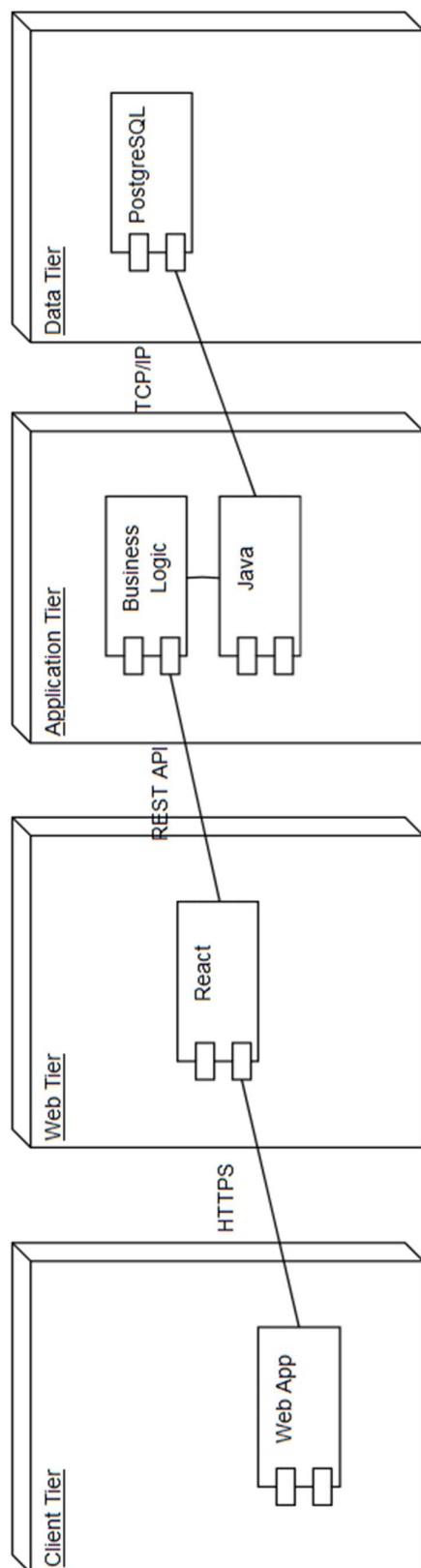
Володимир ГАРМАШ
(ім'я та прізвище)

« 14 » 12 2024 р.

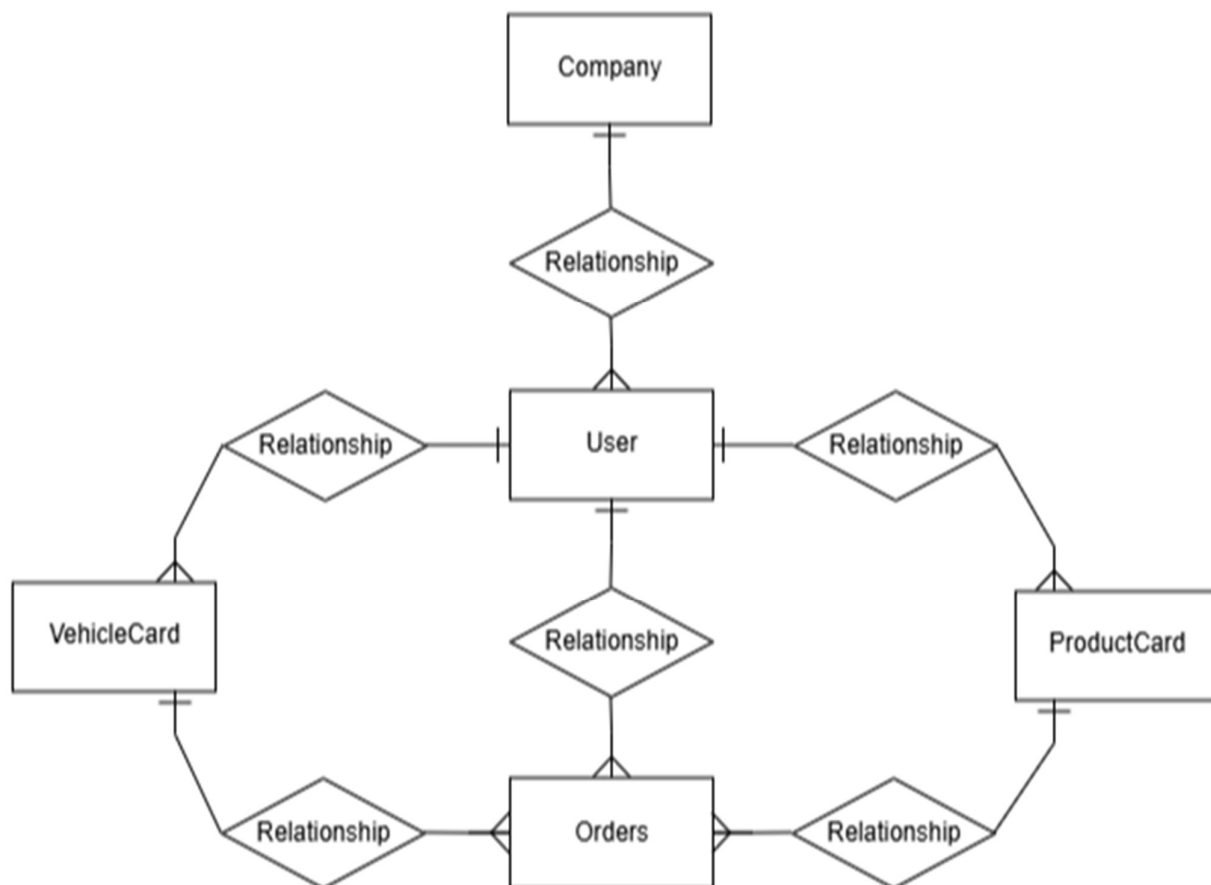
UML-діаграма компонентів автоматизованої системи управління збірними вантажоперевезеннями



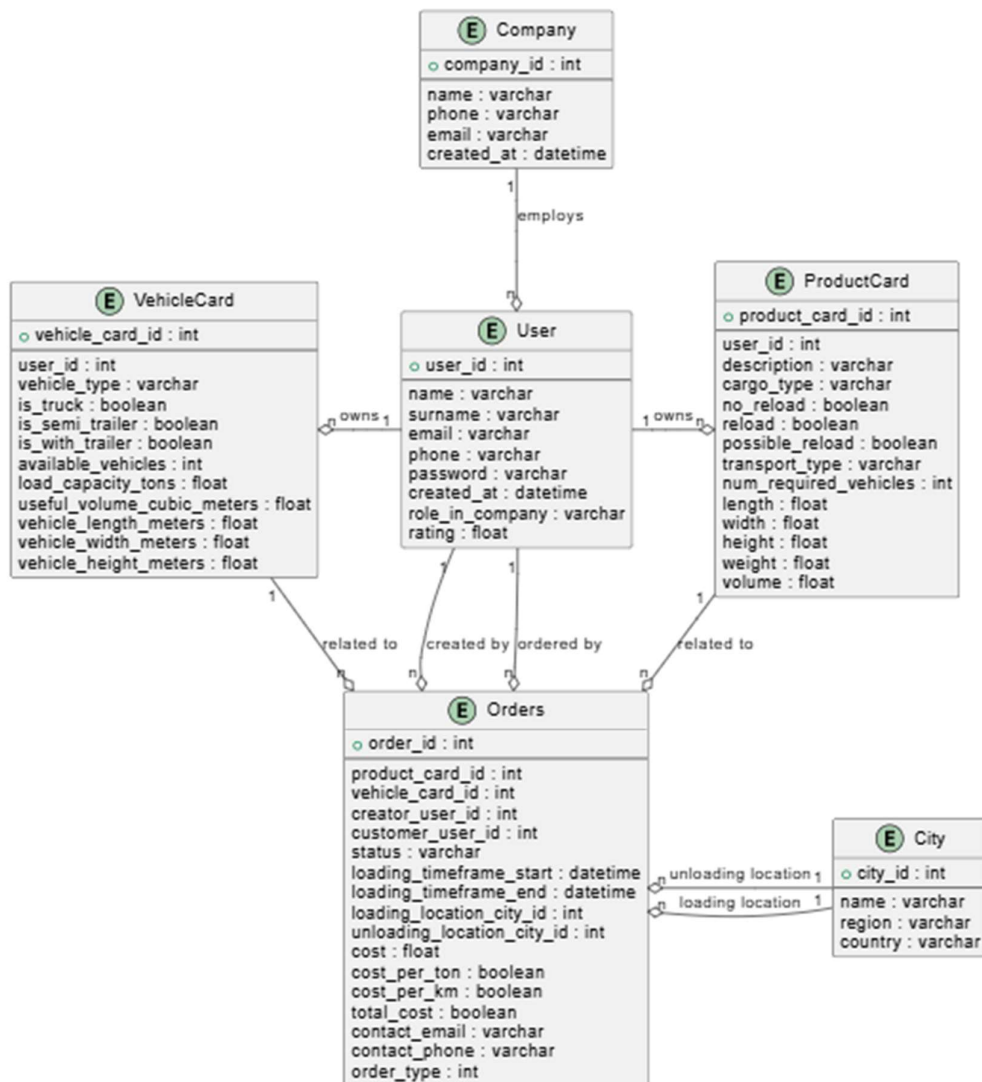
UML-діаграми розгортання технологічної структури автоматизованої системи
управління збірними вантажоперевезеннями



Концептуальна схема бази даних автоматизованої системи управління збірними
вантажоперевезеннями



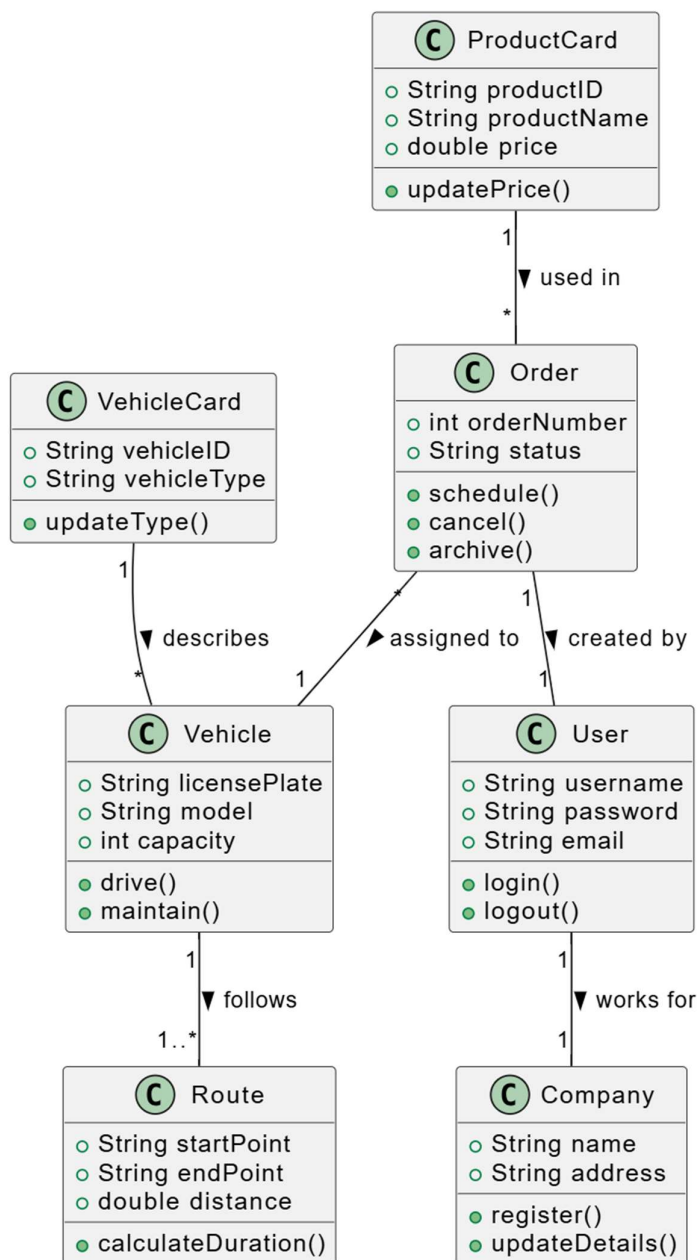
ER-діаграма структури бази даних автоматизованої системи управління збірними вантажоперевезеннями



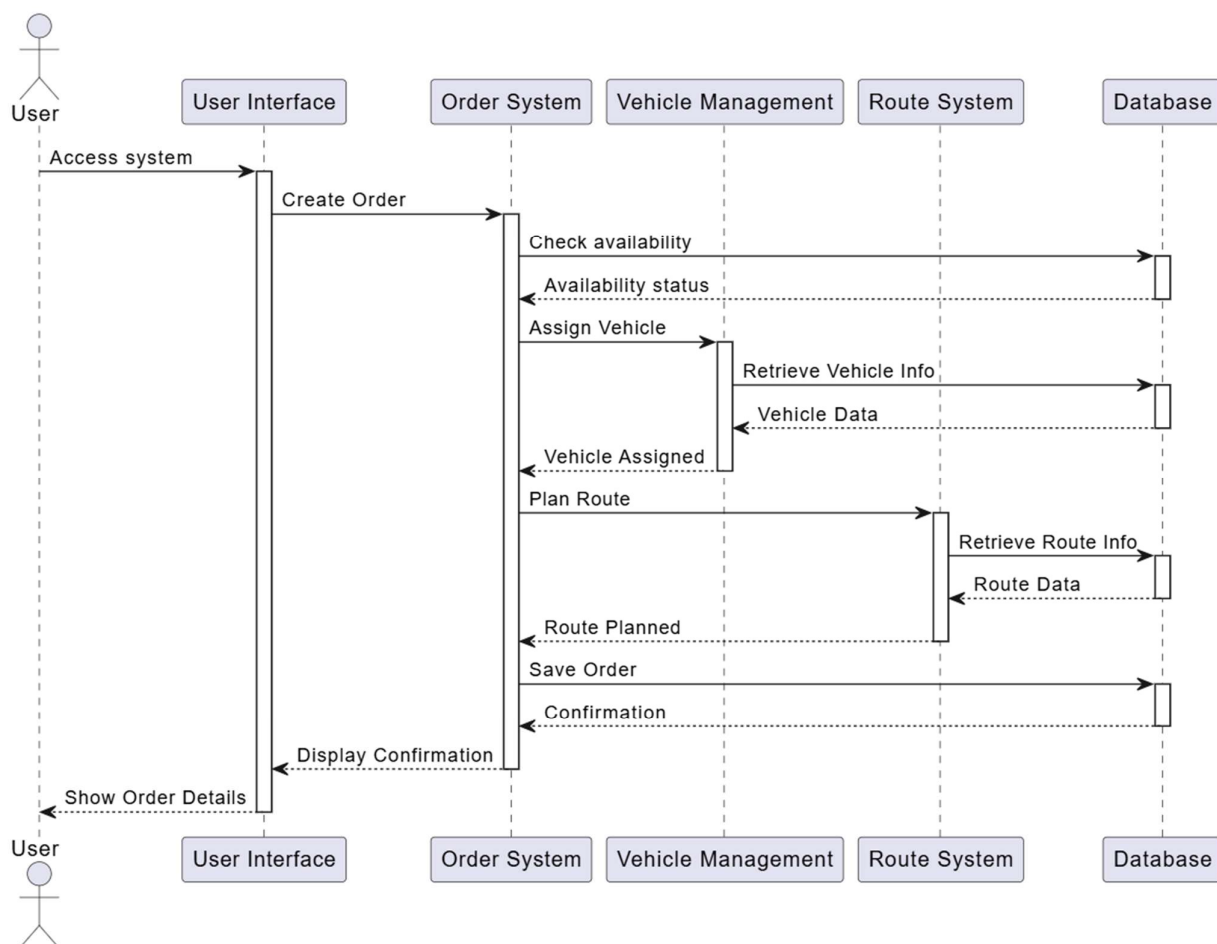
Use Case UML-діаграма для користувача автоматизованої системи управління збірними вантажоперевезеннями



Class UML-діаграма автоматизованої системи управління збірними вантажоперевезеннями



UML-діаграма послідовності дій користувача в автоматизованій системі
управління збірними вантажоперевезеннями



Форма авторизації в системі

**Увійти**

Емейл(логін):

Пароль:

[Забули пароль?](#)Ще не зареєстровані? [Зареєструватися](#)

Відображення інформації про користувача

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Організація

Профіль

Профіль

Інформація про мій профіль

Основна інформація

Моя основна інформація - ім'я, прізвище і місто в якому я перебуваю

Ім'я*

Дмитро

Прізвище*

Дмитро

Зберегти

Пароль

Паролі зберігаються в зашифрованому вигляді, ми не можемо показати його в налаштуваннях. Якщо ви хочете змінити пароль натисніть 'Змінити пароль'

Змінити пароль

Контактна інформація

Ви можете використовувати цю контактну інформацію при розміщенні вантажів і транспорту.

Телефон:

+380(6)

E-Mail:

mail.com

Відображення інформації про компанію

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Організація

Профіль

Організація

Інформація про мою організацію

Загальна інформація

Загальна інформація про мою організацію або бізнес. Ця інформація доступна для перегляду іншим користувачам.

ДМИТРО ДМИТРО

Україна 06/10/2024

Загальний Рейтинг ?

☆☆☆☆☆

Контактна інформація

Ви можете використовувати цю контактну інформацію при розміщенні вантажів і транспорту.

Телефон:

+380(6)

E-Mail:

mail.com

Автоперевезення

мої перевезення

додати вантаж

додати транспорт

пошук вантажів і транспорту

ДД

Дмитро

Додавання вантажу

Тип вантажу:

двері

вага вантажу (т):

0,3

об'єм вантажу (м³):

2

☐ Без довантаження (окреме авто)

☐ Довантаження

☐ Можл. довантаження

Тип транспорту:

крита

Кількість машин:

1

довжина:

м

ширина:

м

висота:

м

Розмістити

Додавання заявки на перевезення вантажу

вказуйте будь-ласка населені пункти завантаження - розвантаження, параметри вантажу та контактну інформацію

Завантаження: з 26.11.2024 до 27.11.2024

Розвантаження

Нас. пункт завантаження: Вінниця

Нас. пункт розвантаження: Чоп

Тип вантажу:

вага вантажу (т): 0,3

об'єм вантажу (м³): 2

☐ Без довантаження (окреме авто) ☐ Довантаження ☐ Мохл. дозавантаження

Тип транспорту: крита

Кількість машин: 1

довжина: м ширина: м висота: м

Вартість: 2000 грн

Всього

E-mail: mail.com

Телефон: +380(6)

Додаткова інформація: [вказати](#)

Розмістити

Автоперевезення

мої перевезення | додати вантаж | додати транспорт | пошук вантажів і транспорту

ДД Дмитро

Розміщення транспорту

Тип транспорту:

тент

☒ вантажівка

☐ напівпричіп

☐ з причепом

вантажівка

напівпричіп

з причепом

Кількість машин:

1

Характеристики транспортного засобу:

Вантажопідйомність (т):

6

Корисний об'єм (м³):

43

довжина:

6,8

м

ширина:

2,45

м

висота:

2,6

м

Розмістити

Створення заявки з транспортним засобом

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Розміщення інформації про вільний транспорт

вказіть бажані пункти завантаження - розвантаження, параметри транспортного засобу та контактну інформацію

Завантаження: з 26.11.2024 до 27.11.2024

Місце завантаження: Вінниця

Розвантаження

Місце розвантаження: Київська обл.

Тип транспорту:

тент

☒ вантажівка

☐ напівпричіп

☐ з причепом



Кількість машин: 1

Характеристики транспортного засобу:

Вантажопідйомність (т): 6

Корисний об'єм (м³): 43

довжина: 6,8 м

ширина: 2,45 м

висота: 2,6 м

Вартість: валюта не вказана за кілометр

E-mail: mail.com

Телефон: +380(

Додаткова інформація: [вказати](#)

Розмістити

Відображення активних перевезень користувача

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [лодати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Активні

Архів

Мої активні перевезення

Активні перевезення, доступні іншим користувачам в пошуку

5 годин тому

26.11–27.11  тент  6 т  43 м³

Вінниця (UA) — Київська обл. (UA)

(дов=6,8м шир=2,45м вис=2,6м)

Докладніше

9 годин тому

26.11–27.11  крита  0,3 т  2 м³

Вінниця (UA) — Чоп (UA) ~ 642 км

двері

2 000 грн
3,12 грн/км

Докладніше

Відображення розрахунку відстаней між містами

Автоперевезення

[мої перевезення](#) | [додати вантаж](#) | [додати транспорт](#) | [пошук вантажів і транспорту](#)

ДД Дмитро

Розрахунок відстаней між містами України

А

Звідки

Вінниця, Вінницька область, Україна

Додати пункт

В

Куди

Київ, Київська область, Україна

Додати пункт

Розрахувати

Деталі маршруту:

Вінниця (UA) — Київ (UA)

Відстань за маршрутом ~ 267 км

Приблизний час поїздки ~ 3 ч 17 м

Вінниця (UA)

Київ (UA)

~ 267 км

~ 3 ч 17 м

Дивитися маршрут на карті

Віджети для швидкого пошуку вантажів та транспортних засобів

Пошук попутних вантажів

Звідки:

Куди:

Пошук попутних машин

Звідки:

Куди:

Віджет з цінами на останні вантажні перевезення зареєстровані в системі

Ціни на вантажні перевезення	
Дніпро — Запоріжжя 4 т	
3 800 грн	44,71 грн/км
Львів — Одеса 22 т	
24 500 грн	29,77 грн/км
Показати всі ціни	