

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут  
«Інститут інформаційних технологій в економіці»**

**Кафедра інформаційних систем в економіці**

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА  
«Інформаційні управляючі системи і технології»**

|               |                               |
|---------------|-------------------------------|
| галузь знань  | 12 Інформаційні<br>технології |
| спеціальність | 122 Комп'ютерні науки         |

Форма навчання: очна (денна)

**КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА**

на тему: «Розроблення інформаційної системи управління для транспортного підприємства»

здобувача Шевченка Вячеслава Анатолійовича  
(ПІБ, підпис)

Науковий керівник: к.т.н., доцент Граняк Валерій Федорович  
(науковий ступінь, учене звання, ПІБ)

\_\_\_\_\_  
(підпис)

**Робота допущена до захисту перед екзаменаційною комісією з атестації  
здобувачів вищої освіти (ЕК)**

Завідувач кафедри: доцент, к.е.н., Тішков Б.О.  
(науковий ступінь, учене звання, ПІБ)

\_\_\_\_\_  
(підпис)

**Київ 2025**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут «Інститут інформаційних технологій в економіці»**

**Кафедра інформаційних систем в економіці**

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА  
«Інформаційні управляючі системи і технології»**

галузь знань                      12 Інформаційні  
технології

спеціальність                122 Комп'ютерні науки

ПОГОДЖЕНО:

Керівник проєктної групи (гарант)  
освітньо-професійної програми

ЗАТВЕРДЖУЮ:

Завідувач кафедри інформаційних  
систем в економіці

\_\_\_\_\_ Устенко С.В.  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

\_\_\_\_\_ Тішков Б.О.  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

здобувача вищої освіти Шевченка Вячеслава Анатолійовича  
(прізвище, ім'я, по батькові)

\_\_\_\_\_ денної \_\_\_\_\_ форми навчання  
очної (денної), заочної, дистанційної

на підготовку кваліфікаційної магістерської роботи  
на тему: «Розроблення інформаційної системи управління для транспортного підприємства»

Тему затверджено наказом ректора Університету від « \_\_\_\_ » \_\_\_\_\_ 202\_ р. № \_\_\_\_\_

**Кваліфікаційна магістерська робота виконується на матеріалах**

---

**План кваліфікаційної магістерської роботи**

**Розділ I** Дослідження та аналіз підходів до створення інформаційних управляючих систем предметної області

**Розділ II** Характеристика інформаційних управляючих систем, методів і моделей

**Розділ III** Розроблення проєктних рішень та їх реалізація

**Об'єкт дослідження:** процеси управління транспортним підприємством за допомогою інформаційних систем.

**Предмет дослідження:** методи, моделі та технології розроблення інформаційної управляючої системи для транспортного підприємства

**Мета кваліфікаційної магістерської роботи:** підвищення ефективності управління логістикою, оптимізація маршрутів перевезень і покращення аналітичних можливостей підприємства шляхом розробки спеціалізованої інформаційної системи управління.

**Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:**

### **У розділі I**

- Дослідити особливості функціонування транспортного підприємства.
- Проаналізувати існуючі інформаційні управляючі системи у транспортній галузі.
- Визначити переваги та недоліки сучасних підходів до автоматизації управління.
- Обґрунтувати вибір методів і технологій для створення інформаційної системи.

### **У розділі II**

- Розробити структуру інформаційної управляючої системи транспортного підприємства.
- Дослідити методи та моделі, що застосовуються в таких системах.
- Оцінити ефективність існуючих технологічних рішень у цій сфері.
- Визначити оптимальні методи обробки та аналізу транспортних даних.

### **У розділі III**

- Спроекувати базу даних для інформаційної управляючої системи.
- Розробити базу знань та механізми інтелектуального аналізу даних.
- Вибрати архітектуру та технології для створення програмного забезпечення.
- Виконати тестування розробленої системи та створити інструкцію користувача.
- Визначити вимоги до технічного забезпечення та реалізувати систему в тестовому середовищі.

**Завдання підготував**

**науковий керівник**

(підпис)

Граняк В.Ф.

(ініціали, прізвище)

«\_\_\_\_» \_\_\_\_\_ 2025 р.

**Завдання одержав**

**здобувач**

(підпис)

Шевченко В.А.

(ініціали, прізвище)

«\_\_\_\_» \_\_\_\_\_ 2025 р.

## Реферат

Кваліфікаційна магістерська робота містить 90 сторінок, 8 таблиць, 46 рисунків, список використаних джерел з 54 найменувань, 1 додаток.

Об'єкт дослідження – процеси управління транспортним підприємством за допомогою інформаційних систем

Предмет дослідження – методи, моделі та технології розроблення інформаційної управляючої системи.

Мета роботи – підвищення ефективності управління логістикою, оптимізація маршрутів перевезень і покращення аналітичних можливостей підприємства шляхом розробки спеціалізованої інформаційної системи управління.

Основні завдання дослідження:

- аналіз існуючих інформаційних управляючих систем у транспортній галузі;
- визначення переваг і недоліків підходів до автоматизації управління;
- обґрунтування вибору методів і технологій для створення системи;
- розробка структури та характеристика інформаційної управляючої системи;
- проектування бази даних, бази знань, механізмів аналізу даних;
- вибір архітектури та інструментальних засобів розробки ПЗ;
- тестування системи та створення керівництва користувача;
- визначення вимог до технічного забезпечення та реалізація системи.

Практична значущість дослідження – розробка інформаційної системи, що підвищує ефективність управління підприємством, оптимізує обробку даних і сприяє прийняттю рішень.

Рік виконання роботи – 2025.

Рік захисту – 2025.

Ключові слова: інформаційна система, транспорт, управління, база даних, інтелектуальний аналіз, автоматизація.

## Зміст

|                                                                                                                   |    |
|-------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....                                                                                   | 4  |
| ВСТУП.....                                                                                                        | 5  |
| РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ<br>ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 8  |
| 1.1 Аналіз існуючих інформаційних управляючих систем у транспортній сфері 8                                       |    |
| 1.1.1 Дослідження транспортного підприємства як предметної області .....                                          | 10 |
| 1.1.2 Дослідження існуючих інформаційних управляючих систем для<br>транспортної галузі.....                       | 14 |
| 1.2 Обґрунтування вибору підходів і технологій для створення інформаційної<br>управляючої системи .....           | 18 |
| РОЗДІЛ 2. ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ<br>СИСТЕМ, МЕТОДІВ І МОДЕЛЕЙ .....                             | 22 |
| 2.1 Структура і характеристика системи управління транспортним<br>підприємством .....                             | 22 |
| 2.2 Методи та моделі в інформаційних управляючих системах і технологіях....                                       | 28 |
| РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ.....                                                      | 32 |
| 3.1 Проєктування бази даних для інформаційної управляючої системи .....                                           | 32 |
| 3.2 Проєктування бази знань та засобів інтелектуального аналізу даних .....                                       | 36 |
| 3.2.1 Реалізація інтелектуального аналізу великих даних для управління<br>транспортном.....                       | 38 |
| 3.2.2 Проєктування та реалізація бази знань системи.....                                                          | 40 |
| 3.3 Програмне забезпечення .....                                                                                  | 43 |
| 3.3.1 Проєктування програмного забезпечення.....                                                                  | 46 |
| 3.3.2 Вибір архітектури програмного забезпечення.....                                                             | 55 |
| 3.3.3 Інструментальні засоби розробки прикладного програмного забезпечення<br>.....                               | 58 |
| 3.3.4 Тестування програмного забезпечення .....                                                                   | 61 |
| 3.3.5 Керівництво (інструкція) користувача .....                                                                  | 72 |
| 3.4 Технічне забезпечення.....                                                                                    | 76 |
| 3.4.1 Обґрунтування вибору технічного забезпечення .....                                                          | 76 |
| 3.4.2 Ресурсні вимоги до технічного забезпечення .....                                                            | 78 |
| 3.5 Реалізація інформаційної управляючої системи.....                                                             | 80 |

|                                 |    |
|---------------------------------|----|
| ВИСНОВКИ.....                   | 83 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 85 |
| ДОДАТОК А ЛІСТИНГ ПРОГРАМИ..... | 91 |

## **ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

ІСУ – Інформаційні системи управління  
ІУС – Інформаційні управляючі системи  
ІС – Інформаційні системи  
ТМС – Системи управління транспортом  
WMS – Системи управління складом  
ГІС – Геоінформаційні системи  
ІТС – Інтелектуальні транспортні системи  
БД – База даних  
СКБД – Система керування базами даних

## ВСТУП

**Актуальність теми.** Сучасні транспортні підприємства функціонують у складному й динамічному середовищі, що вимагає ефективних рішень для управління логістикою, перевезеннями, технічним обслуговуванням і фінансовими потоками. Автоматизація управлінських процесів дозволяє підвищити продуктивність, знизити витрати, оптимізувати маршрути перевезень і мінімізувати людський фактор. У зв'язку з цим розроблення інформаційної системи управління для транспортного підприємства є актуальною науково-прикладною задачею, що має вагоме значення для розвитку галузі.

Дослідження в галузі інформаційних управляючих систем для транспорту активно ведуться як вітчизняними, так і зарубіжними науковцями. Зокрема, питання автоматизації транспортних процесів розглядалися у працях В.М. Назаренка [1], який аналізує сучасні технології управління логістикою, О.В. Грінченка [2], що досліджує математичні методи моделювання транспортних потоків, та Р. Ді Юрія [3], який розробив підходи до оптимізації перевезень у великих містах на основі штучного інтелекту. Незважаючи на значний науковий внесок, питання інтеграції інтелектуальних технологій у транспортні інформаційні системи, розроблення ефективних баз знань для підтримки прийняття рішень, а також адаптація програмного забезпечення під конкретні потреби підприємств залишаються недостатньо висвітленими.

У цьому контексті дане дослідження спрямоване на створення комплексної інформаційної управляючої системи, що забезпечить автоматизацію основних управлінських процесів транспортного підприємства з використанням сучасних технологій обробки даних, машинного навчання та інтелектуального аналізу.

**Наукова проблема:** Сучасні інформаційні управляючі системи для транспортної галузі мають низку обмежень, серед яких: відсутність єдиних стандартів для інтеграції даних між різними підсистемами підприємства; недостатнє використання методів штучного інтелекту для прогнозування та

оптимізації маршрутів; низька адаптивність до змінних умов експлуатації транспорту; обмежені можливості аналітики на основі великих масивів даних.

Розв'язання цієї проблеми вимагає розроблення інформаційної системи управління, що враховуватиме потреби транспортних підприємств і забезпечуватиме гнучке, масштабоване та ефективне управління ресурсами.

**Мета роботи** – підвищення ефективності управління логістикою, оптимізація маршрутів перевезень і покращення аналітичних можливостей підприємства шляхом розробки спеціалізованої інформаційної системи управління.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

У першому розділі: дослідити особливості функціонування транспортного підприємства; проаналізувати існуючі інформаційні управляючі системи у транспортній галузі; визначити переваги та недоліки сучасних підходів до автоматизації управління; обґрунтувати вибір методів і технологій для створення інформаційної системи.

У другому розділі: розробити структуру інформаційної управляючої системи транспортного підприємства; дослідити методи та моделі, що застосовуються в таких системах; оцінити ефективність існуючих технологічних рішень у цій сфері; визначити оптимальні методи обробки та аналізу транспортних даних.

У третьому розділі: спроектувати базу даних для інформаційної управляючої системи; розробити базу знань та механізми інтелектуального аналізу даних; вибрати архітектуру та технології для створення програмного забезпечення; виконати тестування розробленої системи та створити інструкцію користувача; визначити вимоги до технічного забезпечення та реалізувати систему в тестовому середовищі.

**Об'єкт дослідження** – процеси управління транспортним підприємством за допомогою інформаційних систем.

**Предмет дослідження** – методи, моделі та технології розроблення інформаційної управляючої системи для транспортного підприємства.

Для досягнення мети дослідження використано такі **методи**:

- системний аналіз – для оцінки існуючих інформаційних систем управління у транспортній галузі;
- експертні методи – для визначення ключових вимог до програмного забезпечення;
- методи баз даних – для проєктування сховища інформації та структури знань;
- методи машинного навчання – для реалізації інтелектуального аналізу транспортних даних;
- імітаційне моделювання – для оцінки ефективності інформаційної системи.

**Апробація результатів:** Розроблена інформаційна система пройшла тестування в умовах моделювання типових транспортних процесів з використанням імітаційного моделювання на основі статистичних даних одного з українських транспортних підприємств. Її концепція була представлена на науково-практичній конференції «Сучасні інформаційні технології та системи в управлінні», де отримала схвальні відгуки від учасників та фахівців галузі. Отримані результати можуть бути впроваджені у діяльність транспортних підприємств для оптимізації управлінських процесів і покращення якості надання послуг.

**Структура кваліфікаційної магістерської роботи:** Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Перший розділ присвячений аналізу предметної області та існуючих інформаційних систем для транспортної галузі.

Другий розділ містить характеристики методів, моделей і технологій, що використовуються в інформаційних управляючих системах.

Третій розділ описує процес розроблення проєктних рішень, створення бази даних, реалізацію інтелектуального аналізу та тестування програмного забезпечення.

У висновках узагальнено результати дослідження та визначено перспективи подальшого розвитку інформаційних управляючих систем у транспортній сфері.

## РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Аналіз існуючих інформаційних управляючих систем у транспортній сфері

Сучасна транспортна галузь характеризується високим рівнем динаміки та складності, що вимагає ефективних підходів до управління. Інформаційні системи управління (ІСУ) відіграють ключову роль у забезпеченні оперативності, точності та ефективності процесів на транспортних підприємствах [4]. Вони сприяють оптимізації логістичних операцій, підвищенню безпеки перевезень та зниженню витрат.

ІСУ в транспортній сфері охоплюють широкий спектр рішень, спрямованих на автоматизацію та оптимізацію різних аспектів діяльності підприємств [5]. Найбільш поширені системи описані в табл. 1.1.

Таблиця 1.1

Найбільш поширені ІСУ в транспортній сфері

| № | Назва                                | Опис                                                                                                                                                                                                                                                                                                                  |
|---|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2                                    | 3                                                                                                                                                                                                                                                                                                                     |
| 1 | Системи управління транспортом (TMS) | Ці системи забезпечують планування, виконання та моніторинг транспортних операцій. Вони дозволяють оптимізувати маршрути, відстежувати рух транспортних засобів у реальному часі та управляти вантажопотоками. Використання TMS сприяє зниженню витрат на перевезення та підвищенню рівня обслуговування клієнтів [6] |
| 2 | Системи управління складом (WMS)     | Хоча основна функція цих систем полягає в управлінні складськими операціями, інтеграція WMS з TMS забезпечує безперервний потік інформації між складом і транспортом, що підвищує ефективність всього логістичного ланцюга [7]                                                                                        |
| 3 | Геоінформаційні системи (ГІС)        | ГІС використовуються для аналізу та візуалізації просторових даних, що є критично важливим у транспортній сфері. Вони допомагають у плануванні маршрутів, аналізі трафіку та управлінні інфраструктурою. Застосування ГІС сприяє підвищенню точності та оперативності прийняття рішень [8]                            |

## Продовження таблиці 1.1.

| 1 | 2                                        | 3                                                                                                                                                                                                                                               |
|---|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Інтелектуальні транспортні системи (ITS) | ITS об'єднують різні технології для покращення безпеки, ефективності та екологічності транспортних систем. Вони включають системи управління дорожнім рухом, інформаційні системи для пасажирів та системи моніторингу стану інфраструктури [9] |

«Джерело»: розроблено автором на підставі [6, 7, 8, 9]

Впровадження цих систем на транспортних підприємствах дозволяє досягти значних переваг [10]. Зокрема, автоматизація процесів управління транспортом сприяє зниженню операційних витрат, підвищенню точності планування та покращенню контролю за виконанням перевезень. Інтеграція різних ІСУ забезпечує єдиний інформаційний простір, що дозволяє оперативно обмінюватися даними між різними підрозділами підприємства та приймати обґрунтовані управлінські рішення [11].

Однак, незважаючи на очевидні переваги, впровадження ІСУ в транспортній сфері стикається з низкою викликів [12]. Серед них можна виділити, описані в табл. 1.2.

Таблиця 1.2

## Виклики ІСУ в транспортній сфері

| № | Назва                                       | Опис                                                                                                                                                            |
|---|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Високі витрати на впровадження та підтримку | Розробка та інтеграція ІСУ вимагає значних фінансових вкладень, що може бути обтяжливим для деяких підприємств [13]                                             |
| 2 | Складність інтеграції з існуючими системами | Багато транспортних підприємств використовують застарілі системи, інтеграція яких з новими ІСУ може бути технічно складною та вимагати додаткових ресурсів [14] |
| 3 | Проблеми з безпекою даних                   | Зростання обсягу цифрової інформації підвищує ризики кіберзагроз, що вимагає впровадження надійних заходів безпеки та постійного моніторингу [15]               |

«Джерело»: розроблено автором на підставі [13, 14, 15]

Для подолання цих викликів підприємствам рекомендується проводити детальний аналіз потреб та можливостей перед впровадженням ІСУ, забезпечувати

належне навчання персоналу та впроваджувати сучасні засоби захисту інформації [16].

Інформаційні системи управління є невід'ємною складовою сучасних транспортних підприємств [17]. Вони сприяють підвищенню ефективності операцій, оптимізації ресурсів та покращенню якості обслуговування клієнтів. Незважаючи на наявні виклики, правильний підхід до впровадження та використання ІСУ дозволяє досягти значних конкурентних переваг та забезпечити стійкий розвиток підприємства в умовах сучасного ринку.

### **1.1.1 Дослідження транспортного підприємства як предметної області**

Транспортні підприємства відіграють ключову роль у забезпеченні ефективного функціонування економіки, забезпечуючи перевезення вантажів і пасажирів, що сприяє розвитку торгівлі, промисловості та інших секторів [18]. Дослідження транспортного підприємства як предметної області є важливим для розуміння його структури, функцій та процесів, що дозволяє розробляти ефективні інформаційні системи управління [19].

Транспортне підприємство — це організація, основною діяльністю якої є надання транспортних послуг з перевезення вантажів або пасажирів [20]. Залежно від виду транспорту, такі підприємства поділяються на автомобільні, залізничні, морські, річкові, авіаційні та змішані, які комбінують кілька видів транспорту для забезпечення комплексних логістичних рішень. Кожен вид транспорту має свої унікальні особливості, що впливають на організацію перевезень, технічне оснащення, управління процесами та вимоги до інфраструктури. Наприклад, автомобільні перевезення вирізняються високою гнучкістю маршрутів і можливістю доставки "від дверей до дверей", тоді як залізничні перевезення ефективні для транспортування великих обсягів вантажів на значні відстані.

Структура транспортного підприємства (рис. 1.1) зазвичай включає підрозділи, що описані в табл. 1.3.

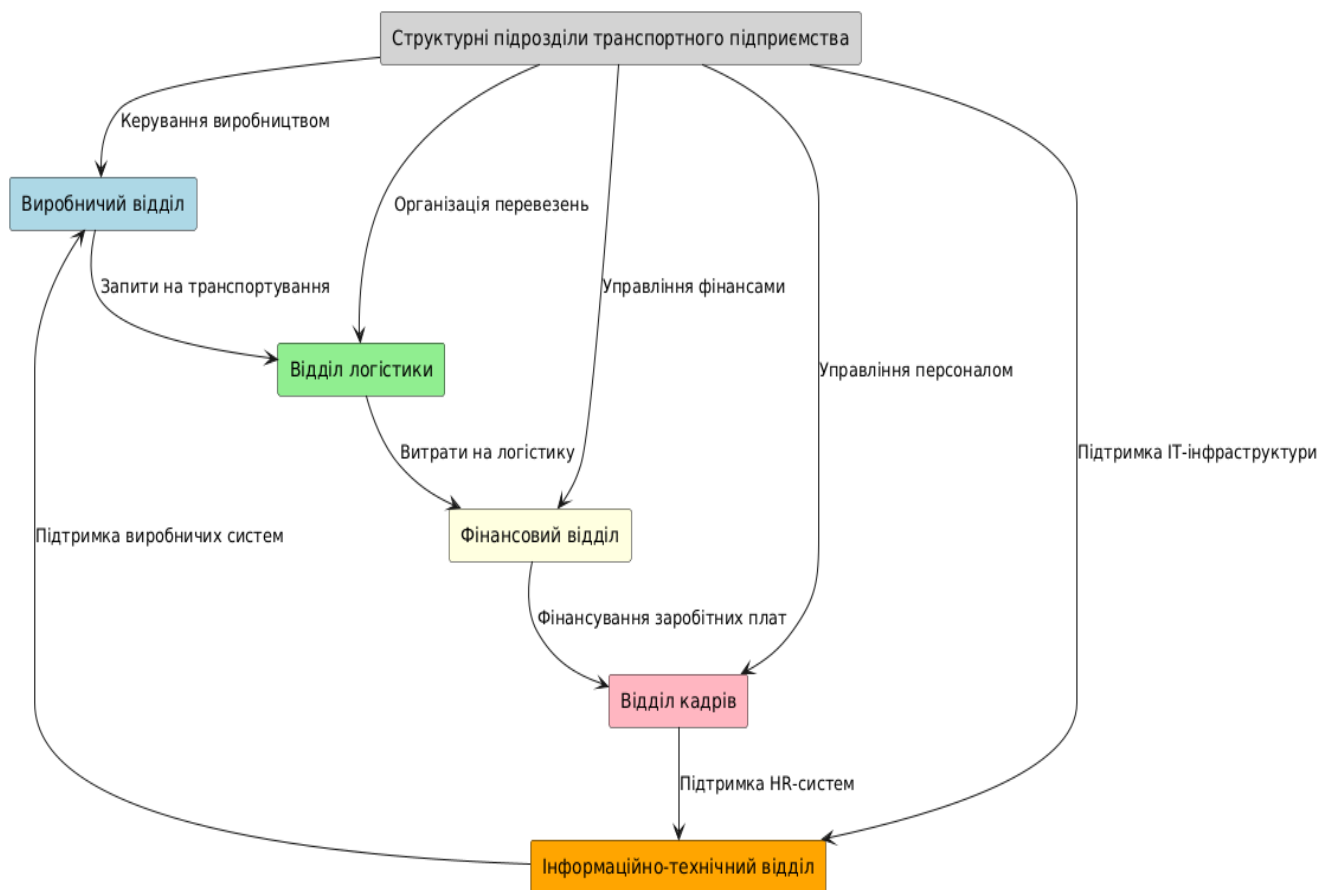


Рисунок 1.1 – Структурна схема транспортного підприємства

«Джерело»: розроблено автором на підставі [21, 22, 23]

Таблиця 1.3

### Структурні підрозділи транспортного підприємства

| № | Назва                         | Опис                                                                                                                           |
|---|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 1 | Виробничий відділ             | відповідає за організацію та виконання перевезень, технічне обслуговування та ремонт транспортних засобів [24]                 |
| 2 | Відділ логістики              | займається плануванням маршрутів, оптимізацією завантаження та розвантаження, координацією з клієнтами та постачальниками [25] |
| 3 | Фінансовий відділ             | здійснює облік доходів і витрат, бюджетування, фінансовий аналіз та звітність [26]                                             |
| 4 | Відділ кадрів                 | відповідає за підбір, навчання та управління персоналом [27]                                                                   |
| 5 | Інформаційно-технічний відділ | забезпечує підтримку інформаційних систем, зв'язок та обробку даних [28]                                                       |

«Джерело»: розроблено автором на підставі [24, 25, 26, 27, 28]

Основні процеси транспортного підприємства представлені в табл. 1.4.

Таблиця 1.4

## Основні процеси транспортного підприємства

| № | Назва                             | Опис                                                                                                                      |
|---|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 1 | Організація перевезень            | прийом замовлень, планування маршрутів, розподіл транспортних засобів та контроль за виконанням перевезень [29]           |
| 2 | Технічне обслуговування та ремонт | регулярне обслуговування транспортних засобів, діагностика та усунення несправностей, плановий та капітальний ремонт [30] |
| 3 | Логістика та складування          | управління складськими запасами, координація завантаження та розвантаження, оптимізація логістичних ланцюгів [31]         |
| 4 | Фінансовий облік та звітність     | ведення бухгалтерського обліку, складання фінансової звітності, аналіз фінансових показників [32]                         |
| 5 | Управління персоналом             | підбір та навчання кадрів, оцінка ефективності роботи, мотивація та розвиток персоналу [33]                               |

«Джерело»: розроблено автором на підставі [29, 30, 31, 32, 33]

Для ефективного управління транспортним підприємством необхідна своєчасна та точна інформація про всі нюанси його діяльності (рис. 1.2). Одним із ключових елементів є дані про транспортні засоби, зокрема їх технічні характеристики, поточний стан, а також графіки технічного обслуговування та ремонту [34]. Важливу роль відіграє інформація щодо маршрутів і розкладів руху, що включає оптимальні шляхи перевезень, часові графіки та умови на дорогах [35].

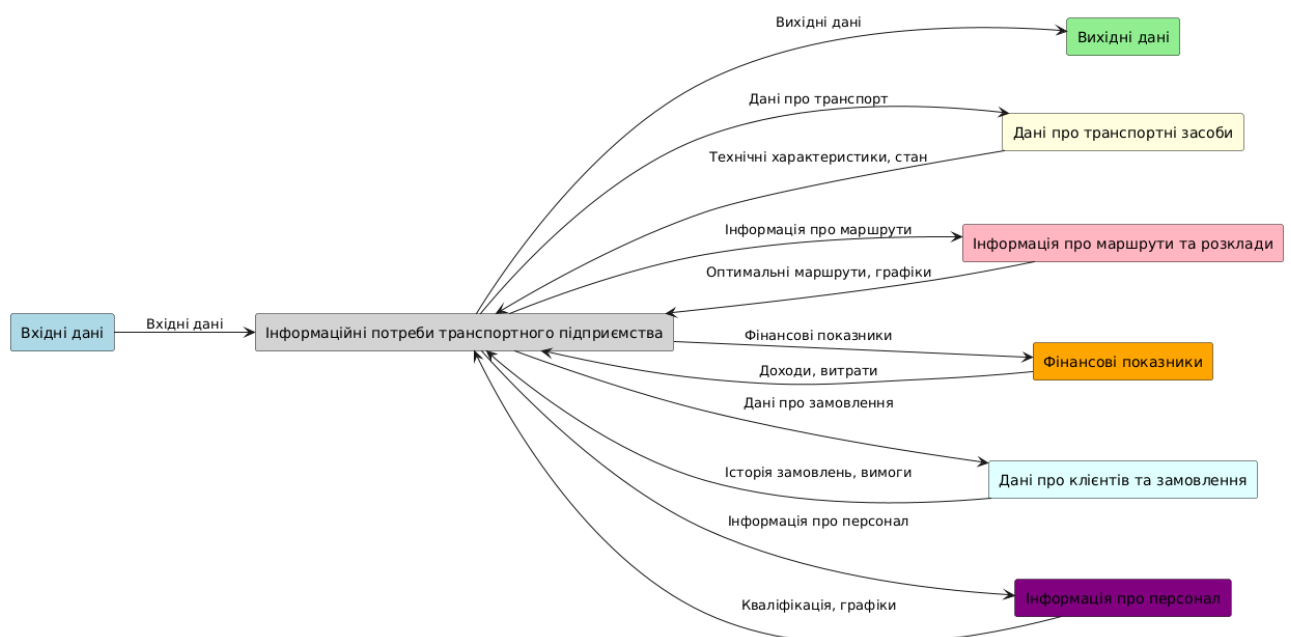


Рисунок 1.2 – Інформаційні потреби транспортного підприємства

«Джерело»: розроблено автором на підставі [34, 35, 36]

Не менш значущими є фінансові показники, які дозволяють оцінювати рівень доходів, витрат, прибутковість та загальну рентабельність підприємства. Також необхідно мати актуальні відомості про клієнтів та їхні замовлення, що містять контактні дані, історію співпраці, а також специфічні вимоги та побажання щодо послуг [36]. Окрім цього, для забезпечення стабільної роботи підприємства важливо контролювати інформацію про персонал, яка охоплює рівень кваліфікації працівників, їхні робочі графіки та результати атестацій.

Застосування сучасних інформаційних систем (рис. 1.3) дає змогу автоматизувати ключові процеси управління транспортним підприємством, підвищуючи ефективність його діяльності та скорочуючи витрати [37]. Завдяки цифровим технологіям стає можливим постійний моніторинг транспортних засобів, що дозволяє відстежувати їхнє місцезнаходження, технічний стан та експлуатаційні параметри в режимі реального часу. Управління замовленнями також набуває нового рівня ефективності, оскільки автоматизовані системи забезпечують швидкий прийом і обробку заявок, контроль за виконанням перевезень та налагодження взаємодії з клієнтами [38].



Рисунок 1.3 – Використання інформаційних систем в управлінні транспортними підприємствами

«Джерело»: розроблено автором на підставі [37, 38, 39]

Розрахунок та планування маршрутів можуть здійснюватися за допомогою алгоритмів, що враховують безліч факторів, таких як дорожня ситуація, погодні

умови та завантаженість транспортних шляхів. Фінансова діяльність підприємства значно оптимізується завдяки автоматизованим інструментам обліку, які спрощують ведення бухгалтерських операцій, підготовку звітності та аналіз фінансових показників [39]. Окрім цього, управління персоналом також інтегрується в інформаційні системи, що дає змогу ефективно контролювати робочий час, формувати графіки змін та оцінювати продуктивність співробітників.

Дослідження транспортного підприємства як предметної області дозволяє глибше зрозуміти його структуру, процеси та інформаційні потреби. Це є основою для розробки ефективних інформаційних систем управління, які сприяють підвищенню ефективності роботи підприємства, оптимізації ресурсів та покращенню якості наданих послуг.

### **1.1.2 Дослідження існуючих інформаційних управляючих систем для транспортної галузі**

Транспортна галузь є однією з найважливіших складових економіки, і її ефективність прямо залежить від рівня автоматизації та управлінських технологій [40]. Останнім часом інформаційні системи управління в транспортній сфері набувають дедалі більшого значення, адже вони дозволяють оптимізувати роботу підприємств, забезпечити своєчасність виконання операцій, знижувати витрати і підвищувати рівень безпеки. Тому вивчення існуючих підходів до розробки та впровадження таких систем є необхідним кроком для побудови ефективної та гнучкої інфраструктури, здатної задовольнити сучасні вимоги ринку.

Інформаційні системи управління для транспортних підприємств є багатофункціональними комплексами, що охоплюють різні етапи процесу перевезень (рис. 1.4): планування маршрутів, моніторинг транспорту, управління запасами, логістику та взаємодію з клієнтами [41]. Розробка таких систем вимагає комплексного підходу, здатного охопити усі аспекти діяльності підприємства, від обробки заявок клієнтів до аналізу та оптимізації маршрутів транспорту [42].

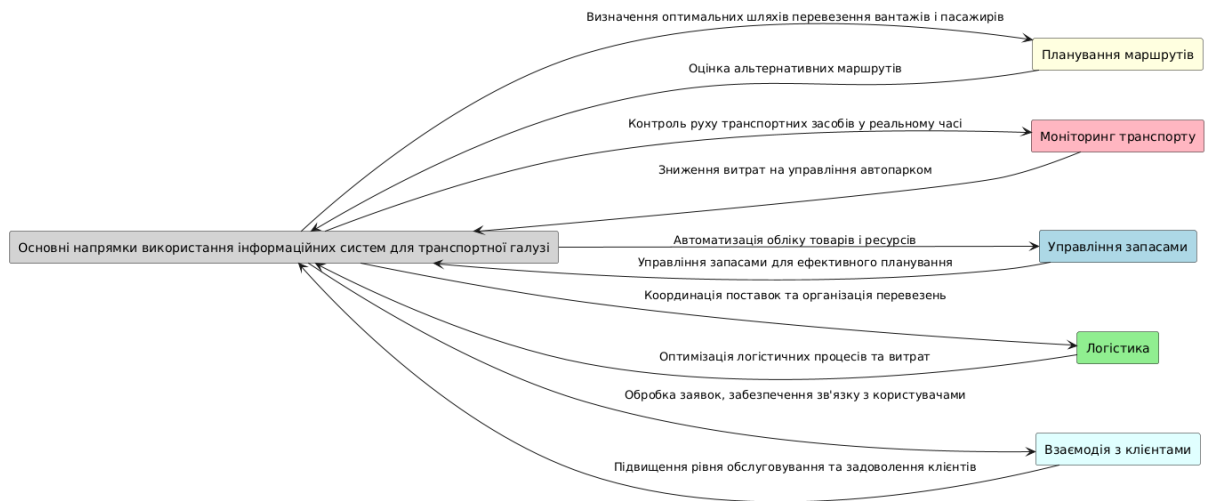


Рисунок 1.4 – Основні напрямки використання інформаційних систем

«Джерело»: розроблено автором на підставі [41, 42]

Одним із найбільш відомих типів інформаційних систем для транспортних компаній (рис. 1.5) є системи управління автопарком [43]. Ці системи забезпечують моніторинг транспортних засобів у реальному часі, включаючи GPS-навігацію, контроль пального, швидкості та технічного стану автомобілів. Вони дозволяють зменшити витрати на експлуатацію автопарку, підвищити ефективність роботи водіїв та забезпечити кращу взаємодію між різними підрозділами підприємства [44]. Наприклад, система Fleet Complete пропонує рішення для відстеження, управління і аналізу даних про транспорт, що дає змогу оптимізувати робочі процеси і приймати більш обґрунтовані рішення.



Рисунок 1.5 – Основні типи інформаційних систем для транспортної галузі

«Джерело»: розроблено автором на підставі [43, 44, 45]

Ще одним важливим напрямом є інформаційні системи для логістики та доставки. Це спеціалізовані платформи, які використовуються для планування маршрутів, управління запасами, а також для прогнозування термінів доставки та обробки замовлень. Одним з таких рішень є система Transporeon, яка надає платформи для керування ланцюгами поставок, взаємодії з перевізниками, моніторингу вантажоперевезень у реальному часі. Завдяки такому програмному забезпеченню зменшуються затримки в доставці, оптимізуються транспортні маршрути, знижуються витрати на логістику.

У контексті залізничного транспорту можна згадати системи для управління рухом поїздів та автоматизації роботи станцій. Такі платформи включають моніторинг складів, управління перевезеннями та планування графіків руху. Однією з таких систем є SIRCA, що використовується в залізничних компаніях для автоматизації планування маршрутів і моніторингу транспорту. Вони дозволяють зменшити людський фактор і підвищити ефективність використання залізничних ліній.

Інтелектуальні транспортні системи (ITS) є наступним рівнем розвитку інформаційних систем у транспортній галузі. Вони використовують новітні технології для управління дорожнім рухом, зменшення заторів та підвищення безпеки. Це рішення включає використання датчиків, камер спостереження, розпізнавання номерних знаків, системи керування світлофорами [45]. ITS дозволяють не тільки керувати дорожнім рухом, але й надавати водіям інформацію про ситуацію на дорогах у реальному часі. Наприклад, система ATMS (Advanced Traffic Management Systems) вміє адаптувати режим руху світлофорів, прогнозуючи затори та сприяючи більш плавному руху транспорту.

Ще однією важливою категорією є системи управління пасажирським транспортом. Ці системи сприяють поліпшенню обслуговування пасажирів, надаючи точну інформацію про час прибуття автобусів, тролейбусів або трамваїв, а також дозволяючи здійснювати оплату проїзду через мобільні додатки. Один з прикладів таких систем — Транспортні карти та системи на основі мобільних додатків для громадського транспорту, які застосовуються в багатьох великих

містах. Вони дозволяють зменшити черги в касах, покращити облік пасажирів та здійснювати більш точне планування маршрутів.

Важливо розглянути переваги та недоліки існуючих ІС для транспортних підприємств (рис. 1.6) [46].



Рисунок 1.6 – Переваги та виклики використання інформаційних систем для транспортних підприємств

«Джерело»: розроблено автором на підставі [46]

Переваги використання інформаційних управляючих систем для транспортних підприємств значні. Вони дозволяють знизити витрати, підвищити продуктивність, покращити обслуговування клієнтів, а також забезпечити високий рівень безпеки на дорогах і в транспортних засобах. Проте, впровадження таких систем також стикається з рядом труднощів, таких як необхідність значних інвестицій у інфраструктуру, інтеграцію з існуючими системами та підготовку персоналу [46].

Загалом, існуючі інформаційні системи для управління транспортом суттєво підвищують ефективність роботи транспортних підприємств і забезпечують якісне обслуговування клієнтів. Однак для досягнення найкращих результатів необхідно інтегрувати різні типи технологій у єдину систему, враховуючи специфіку кожного виду транспорту і вимоги до безпеки та ефективності. З розвитком новітніх технологій, таких як штучний інтелект, автоматизація та Інтернет речей, можна очікувати ще більше інноваційних рішень, які сприятимуть подальшому розвитку транспортної галузі.

## 1.2 Обґрунтування вибору підходів і технологій для створення інформаційної управляючої системи

Сучасне транспортне підприємство, яке забезпечує надання послуг з перевезення, складування, митного оформлення та страхування вантажів, стикається з численними викликами, пов'язаними з ефективним управлінням своїми операціями. Інтеграція інформаційних технологій у такі бізнес-процеси стає необхідною умовою для досягнення високої конкурентоспроможності і забезпечення стабільної роботи підприємства. Для цього транспортні компанії потребують складних інформаційних систем, які забезпечують автоматизацію різних бізнес-процесів і дозволяють інтегрувати їх у єдину управляючу структуру. Вибір підходів і технологій для створення таких систем (рис. 1.7) вимагає ретельного підходу, що поєднує зручність розробки, високу продуктивність, безпеку і масштабованість [47]. У цьому контексті важливою є роль правильного вибору програмних інструментів, які будуть відповідати специфічним вимогам бізнесу.

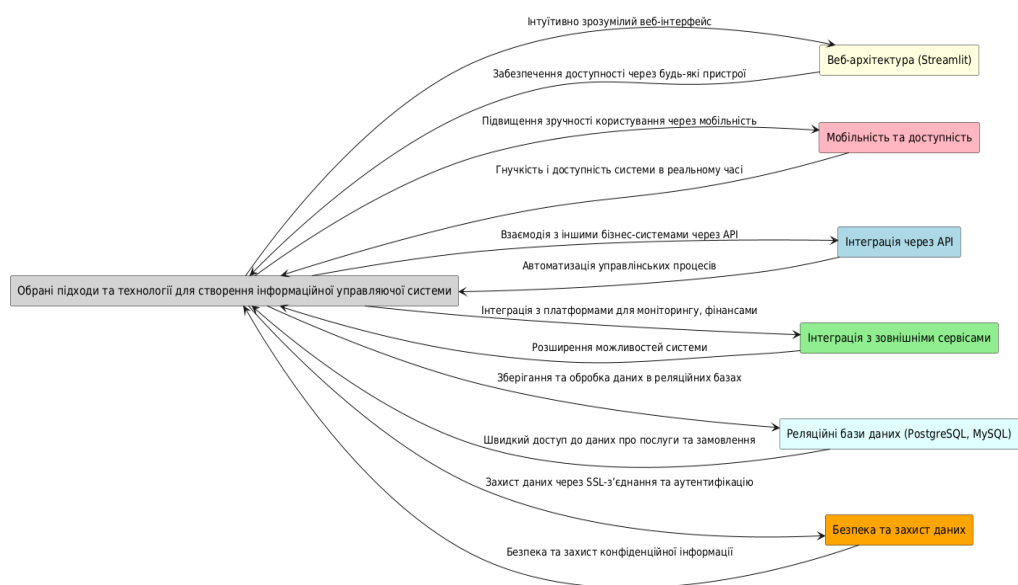


Рисунок 1.7 – Обраних підходів та технологій для створення інформаційної управляючої системи

«Джерело»: розроблено автором на підставі [47, 48, 49, 50, 51, 52, 43]

Для створення ефективної інформаційної системи для транспортного підприємства важливо звернути увагу на особливості роботи в цій галузі. Важливими факторами, що визначають вибір технологій, є потреба в інтерактивному інтерфейсі, швидкому доступі до даних, інтеграції з іншими бізнес-системами та забезпеченні надійного захисту даних. Враховуючи ці вимоги, розробка системи з використанням веб-технологій, зокрема, Streamlit, виглядає оптимальним рішенням для ефективного управління процесами на підприємстві [48].

Одним із ключових аспектів при розробці системи є інтеграція різних компонентів: управлінських функцій, взаємодії з клієнтами, моніторингу послуг і замовлень, а також забезпечення безпеки даних [49]. Вибір платформи для веб-розробки, яка дозволяє швидко створювати додатки, що підтримують інтерактивність та зручність користування, є надзвичайно важливим. Платформа Streamlit, яка дозволяє розробникам створювати інтуїтивно зрозумілі інтерфейси без необхідності глибоких знань у веб-розробці, стала ідеальним вибором для реалізації такого рішення [50].

Завдяки інтеграції з Python, Streamlit дозволяє ефективно обробляти великі обсяги даних у реальному часі [51]. Це є важливим для транспортних компаній, де необхідно забезпечити актуальність і точність інформації про замовлення, послуги та стан перевезень. Крім того, завдяки своїй простоті у використанні та гнучкості, Streamlit забезпечує швидку розробку веб-додатків, що відповідають вимогам транспортного підприємства.

З огляду на специфіку бізнес-процесів у транспортній галузі, важливою є здатність системи інтегруватися з іншими сервісами та платформами. Це дає змогу автоматизувати багато процесів, зокрема обробку замовлень, моніторинг вантажоперевезень, фінансову звітність, а також роботу з клієнтами. Використання API для інтеграції з зовнішніми сервісами дозволяє забезпечити автоматичне оновлення даних і зменшити кількість ручних операцій, що значно підвищує ефективність роботи системи.

Крім того, для зберігання та обробки даних використовуються реляційні бази даних, такі як PostgreSQL або MySQL [52]. Ці технології дозволяють структурувати великі обсяги даних, що стосуються послуг, замовлень і клієнтів, і забезпечують швидкий доступ до необхідної інформації. Реляційні бази даних є стабільним і перевіреним рішенням для зберігання важливих даних, що дозволяє підприємствам ефективно керувати інформацією, підтримувати її цілісність і забезпечувати швидкий доступ до даних.

Не менш важливим є питання безпеки. Оскільки система буде працювати з чутливою інформацією, такою як персональні дані користувачів та фінансові транзакції, необхідно забезпечити високий рівень захисту цих даних. Для цього використовуються сучасні методи безпеки, зокрема використання SSL-з'єднань для захисту передавання даних та методи аутентифікації за допомогою токенів для забезпечення контролю доступу до системи [53]. Це дозволяє гарантувати захист інформації та знизити ризики, пов'язані з її викраденням або неправомірним доступом.

Транспортне підприємство, яке надає широкий спектр послуг, потребує ефективною інтеграції своєї інформаційної системи з іншими бізнес-платформами. Вибір технологій, що дозволяють забезпечити таку інтеграцію, має важливе значення для оптимізації роботи підприємства. Використання API для зв'язку з різними зовнішніми системами дає змогу автоматизувати багато процесів і забезпечити безперебійну роботу системи.

Один із найбільш важливих напрямків інтеграції — це взаємодія з фінансовими платформами [54]. Враховуючи, що підприємства часто працюють з великою кількістю транзакцій, автоматизація процесів, пов'язаних з оплатою послуг, звітністю та управлінням рахунками, є необхідною для забезпечення високої ефективності бізнесу. Використання API для інтеграції з фінансовими платформами дозволяє автоматично обробляти платежі, генерувати звіти про доходи та витрати, а також синхронізувати фінансові дані з іншими складовими інформаційної системи.

Іншим важливим напрямком інтеграції є моніторинг вантажоперевезень і звітність. Для транспортних підприємств важливо мати актуальну інформацію про місцезнаходження вантажів, стан перевезень і прогнозування можливих затримок. Використання зовнішніх систем моніторингу дозволяє не тільки покращити обслуговування клієнтів, а й оптимізувати внутрішні процеси, мінімізуючи витрати на управління транспортним парком і перевезеннями. Інтеграція з такими сервісами через API забезпечує безперервний потік актуальної інформації.

Інформаційна система повинна бути не лише ефективною, але й масштабованою, щоб відповідати на вимоги росту підприємства. Технології, які були обрані для розробки цієї системи, дозволяють безперешкодно розширювати її функціональність та забезпечувати обробку більших обсягів даних. Реляційні бази даних, зокрема PostgreSQL і MySQL, добре підходять для роботи з великими масивами даних і можуть підтримувати зростання обсягів інформації без втрати продуктивності [52].

Streamlit дозволяє швидко реагувати на зміни в бізнес-процесах, оскільки це гнучка платформа, яка дозволяє розробляти нові функції та інтерфейси без значних затримок [50]. Завдяки використанню цієї технології можна швидко адаптувати систему під нові вимоги та зміни в бізнесі. Це дозволяє підприємству зберігати конкурентні переваги і адаптувати свою діяльність до вимог часу.

Розробка інформаційної управляючої системи для транспортного підприємства є складним і багатоетапним процесом, що вимагає ретельного підбору підходів і технологій. Вибір сучасних інструментів для створення системи, таких як Streamlit для веб-розробки, реляційні бази даних для зберігання даних, API для інтеграції з іншими сервісами, а також застосування методів безпеки для захисту даних, дозволяє забезпечити високу ефективність роботи підприємства. Всі ці технології взаємодіють між собою, створюючи потужну та гнучку платформу для управління різними бізнес-процесами. Завдяки правильному вибору технологій підприємство отримує можливість не лише оптимізувати існуючі процеси, але й масштабувати свою діяльність у відповідь на зміни на ринку.

## **РОЗДІЛ 2. ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ, МЕТОДІВ І МОДЕЛЕЙ**

### **2.1 Структура і характеристика системи управління транспортним підприємством**

Розвиток транспортних компаній у сучасному світі є складним і багатогранним процесом, що включає не лише фізичне перевезення вантажів чи пасажирів, але й організацію та оптимізацію численних бізнес-процесів, пов'язаних із цими послугами. Важливим елементом ефективної роботи транспортного підприємства є впровадження інформаційних систем управління, які дозволяють автоматизувати численні операції, забезпечити оперативність прийняття рішень і мінімізувати людські помилки.

У рамках цієї роботи розглядається система, що дозволяє управляти різними аспектами діяльності транспортного підприємства, зокрема перевезеннями, замовленнями, складським зберіганням, митними процедурами та страхуванням вантажів. Така інформаційна система повинна бути гнучкою, масштабованою та надійною, що дозволяє менеджерам та адміністраторам ефективно керувати процесами, отримувати важливу аналітичну інформацію та підтримувати високий рівень обслуговування клієнтів.

Інформаційна система для транспортного підприємства має бути побудована таким чином, щоб забезпечувати інтеграцію всіх важливих процесів, які включають планування, виконання та моніторинг перевезень, управління замовленнями, аналітику діяльності та інші функції. Для цього система повинна мати кілька основних модулів (рис. 2.1), що відповідають різним напрямкам роботи підприємства.

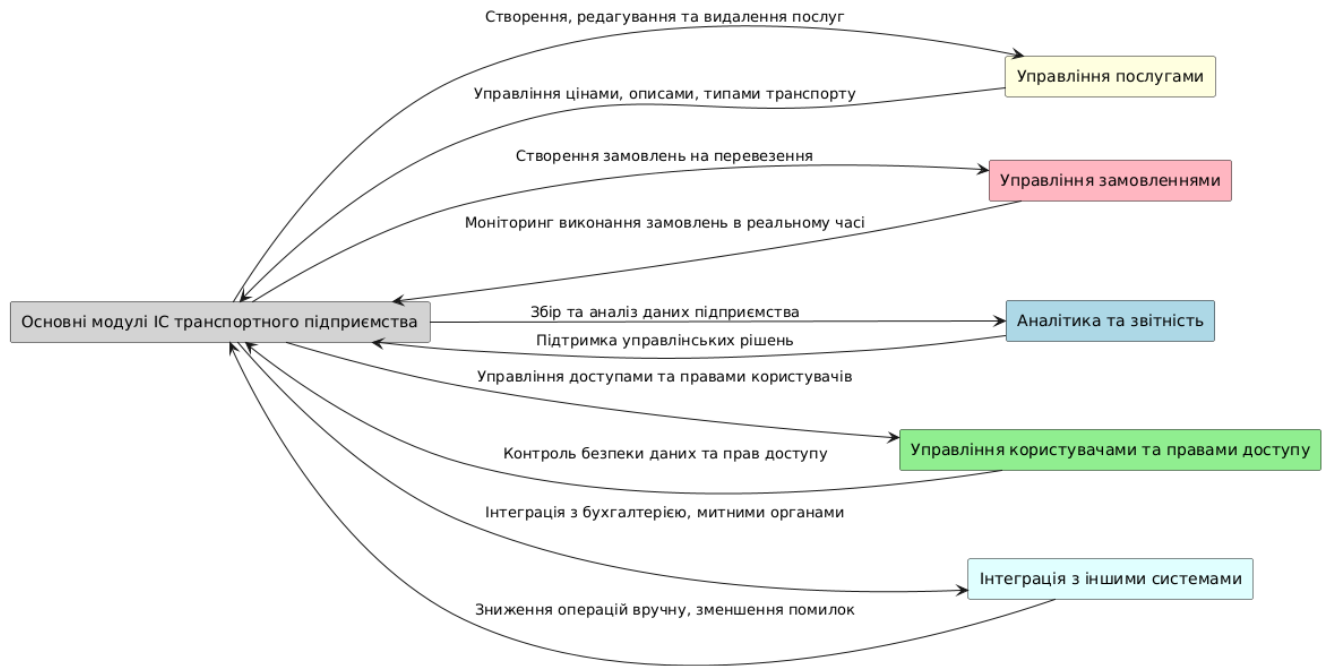


Рисунок 2.1 – Основні модулі інформаційної системи транспортного підприємства

«Джерело»: [розроблено автором]

Основні модулі, які повинні бути частиною інформаційної системи управління, описані в табл. 2.1.

Таблиця 2.1

#### Основні модулі інформаційної системи управління

| № | Назва                   | Опис                                                                                                                                                                                                                                                                                                                                                                                            |
|---|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2                       | 3                                                                                                                                                                                                                                                                                                                                                                                               |
| 1 | Управління послугами    | Цей модуль дозволяє створювати, редагувати та видаляти різноманітні послуги підприємства, такі як міжміські та міжнародні перевезення, складське зберігання, митне оформлення, страхування вантажів. Користувачі можуть управляти цінами, описами послуг, обирати тип транспорту та інші параметри. Це дозволяє забезпечити ефективну організацію роботи компанії, адаптуючи її до потреб ринку |
| 2 | Управління замовленнями | Цей компонент системи дозволяє клієнтам створювати замовлення на перевезення вантажів або пасажирів, а менеджерам — слідкувати за їх виконанням. Важливим аспектом є автоматизація процесів оформлення замовлень, їх моніторинг у реальному часі, а також інтеграція з модулями оплати та звітності                                                                                             |

Продовження таблиці 2.1

| 1 | 2                                           | 3                                                                                                                                                                                                                                                                                             |
|---|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3 | Аналітика та звітність                      | Цей модуль забезпечує збір та аналіз даних про роботу підприємства, таких як ефективність перевезень, рівень задоволення клієнтів, фінансові показники, зростання доходів. Дані з цього модуля можуть бути використані для ухвалення важливих управлінських рішень і покращення якості послуг |
| 4 | Управління користувачами та правами доступу | Кожен користувач, залежно від своєї ролі в системі, має доступ до різних функцій. Це дозволяє ефективно контролювати процеси управління та забезпечити безпеку даних підприємства                                                                                                             |
| 5 | Інтеграція з іншими системами               | Важливою характеристикою є можливість інтеграції з зовнішніми системами, такими як бухгалтерія, склади, митні органи, що дозволяє знизити кількість операцій, що виконуються вручну, і зменшити ризик помилок                                                                                 |

«Джерело»: [розроблено автором]

У системі управління транспортним підприємством модуль управління послугами (рис. 2.2) є ключовим для забезпечення ефективності роботи компанії. Він дозволяє створювати нові послуги, редагувати вже існуючі та видаляти непотрібні. Кожна послуга має набір атрибутів, таких як назва, опис, тип транспорту, ціни та інші параметри, що дозволяють персоналізувати пропозицію для клієнтів.



Рисунок 2.2 – Блок-схема модулю управління послугами

«Джерело»: [розроблено автором на підставі коду програми]

Завдяки інтеграції цього модуля з іншими системами підприємства, менеджери мають змогу оперативно оновлювати інформацію щодо цін та умов

перевезення. Зокрема, система передбачає можливість вибору типу транспорту для кожної послуги, що дає змогу точно налаштувати цінову політику та адаптувати пропозиції під конкретні умови замовлення.

Модуль управління замовленнями (рис. 2.3) є основним для взаємодії з клієнтами. Він дозволяє створювати замовлення на перевезення, обирати послуги, які необхідно використати, а також моніторити статус виконання замовлення. Завдяки інтеграції з іншими модулями, такими як управління послугами та аналітика, цей модуль дозволяє автоматично обирати оптимальний тип транспорту та організовувати маршрут перевезення.

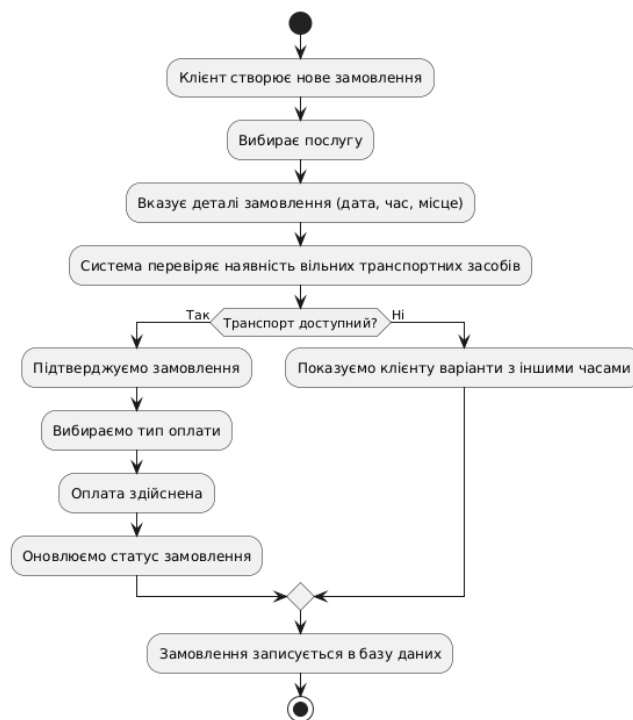


Рисунок 2.3 – Блок-схема модулю управління замовленнями

«Джерело»: [розроблено автором на підставі коду програми]

Система надає можливість не тільки створювати замовлення, а й редагувати його на всіх етапах виконання, що дає змогу оперативно реагувати на зміни в умовах перевезення. Для кожного замовлення створюється детальний запис, що містить усю необхідну інформацію — від умов оплати до прогнозованих термінів доставки.

Модуль аналітики (рис. 2.4) дозволяє збирати та обробляти дані про діяльність підприємства, такі як ефективність перевезень, витрати на транспортування, рівень задоволення клієнтів та інші важливі показники. Зібрані дані можуть бути представлені у вигляді звітів, графіків та діаграм, що дозволяє менеджерам краще розуміти ситуацію і приймати обґрунтовані рішення для покращення роботи компанії.



Рисунок 2.4 – Блок-схема модулю аналітики та звітності

*«Джерело»: [розроблено автором на підставі коду програми]*

Особливістю даного модуля є його інтеграція з іншими компонентами системи, що дозволяє автоматично оновлювати інформацію у звітності та надавати точні дані для аналізу.

Модуль управління користувачами та правами доступу (рис. 2.5) забезпечує гнучке управління доступом до різних функцій системи. Кожен користувач має визначену роль (клієнт, менеджер, адміністратор), яка дозволяє обмежувати або надавати доступ до певних функцій системи. Це дозволяє зберегти конфіденційність даних та знизити ризики неправомірного доступу.

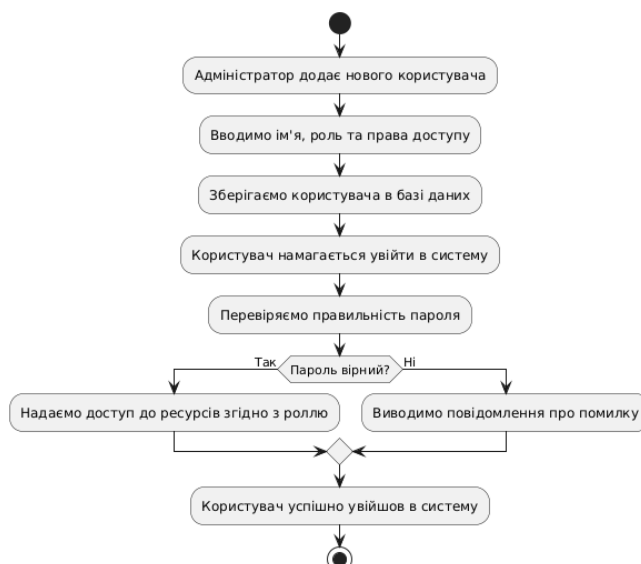


Рисунок 2.5 – Блок-схема модулю управління користувачами та правами доступу

«Джерело»: [розроблено автором на підставі коду програми]

Завдяки цьому модулю система стає більш безпечною та зручною в управлінні, оскільки кожен користувач може працювати лише з тими даними, які йому необхідні.

Інтеграція та безпека системи (рис. 2.6): інформаційна система для транспортного підприємства повинна бути розроблена з урахуванням потреб у безпеці та конфіденційності даних. Всі чутливі дані, такі як фінансові операції або особисті дані клієнтів, повинні бути захищені від несанкціонованого доступу. Це може бути досягнуто за допомогою використання шифрування, двофакторної автентифікації та інших засобів захисту.

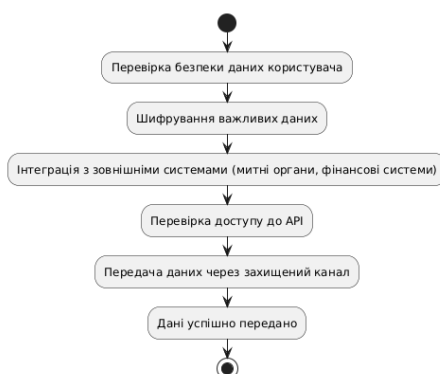


Рисунок 2.6 – Блок-схема модулю інтеграції та безпеки системи

«Джерело»: [розроблено автором на підставі коду програми]

Інтеграція з іншими системами, такими як митні органи чи транспортні компанії, має на увазі наявність спеціальних API або інших механізмів обміну даними, що дозволяє зменшити кількість ручних операцій та помилок у процесі роботи.

Інформаційна система управління для транспортного підприємства є важливим інструментом, що дозволяє автоматизувати більшість операцій, забезпечити ефективне управління різними аспектами діяльності компанії та покращити взаємодію з клієнтами. Вона включає численні модулі, що охоплюють всі етапи перевезення — від створення послуг до управління замовленнями та аналізу ефективності. За допомогою такої системи можна оптимізувати процеси, знижувати витрати та підвищувати рівень обслуговування, що має позитивний вплив на загальний результат діяльності транспортного підприємства.

## **2.2 Методи та моделі в інформаційних управляючих системах і технологіях**

Інформаційні управляючі системи (ІУС) є важливим інструментом для ефективного управління діяльністю підприємств у різних сферах, включаючи транспортні компанії. Вони дозволяють автоматизувати процеси, знижувати ризики помилок, покращувати взаємодію з клієнтами та забезпечувати зручний доступ до різноманітних даних. Методи та моделі, що використовуються в ІУС, створюють основу для їх ефективної роботи. У розглянутому випадку транспортної компанії такі системи можуть охоплювати різні аспекти діяльності: від керування замовленнями до управління послугами та логістикою. Детальне розуміння методів та моделей, застосованих у цих системах, є ключовим для розробки та вдосконалення інформаційної інфраструктури підприємства.

Основні методи та моделі ІУС (рис.2.7), що застосовуються в транспортній сфері, включають моделювання процесів, побудову графів та мереж, теорії оптимізації та планування, а також використання спеціалізованих алгоритмів для обробки великих обсягів даних. Для правильного функціонування ІУС важливими

є як методи збору та обробки інформації, так і способи інтеграції різних елементів, що забезпечують безперервний процес обміну даними.

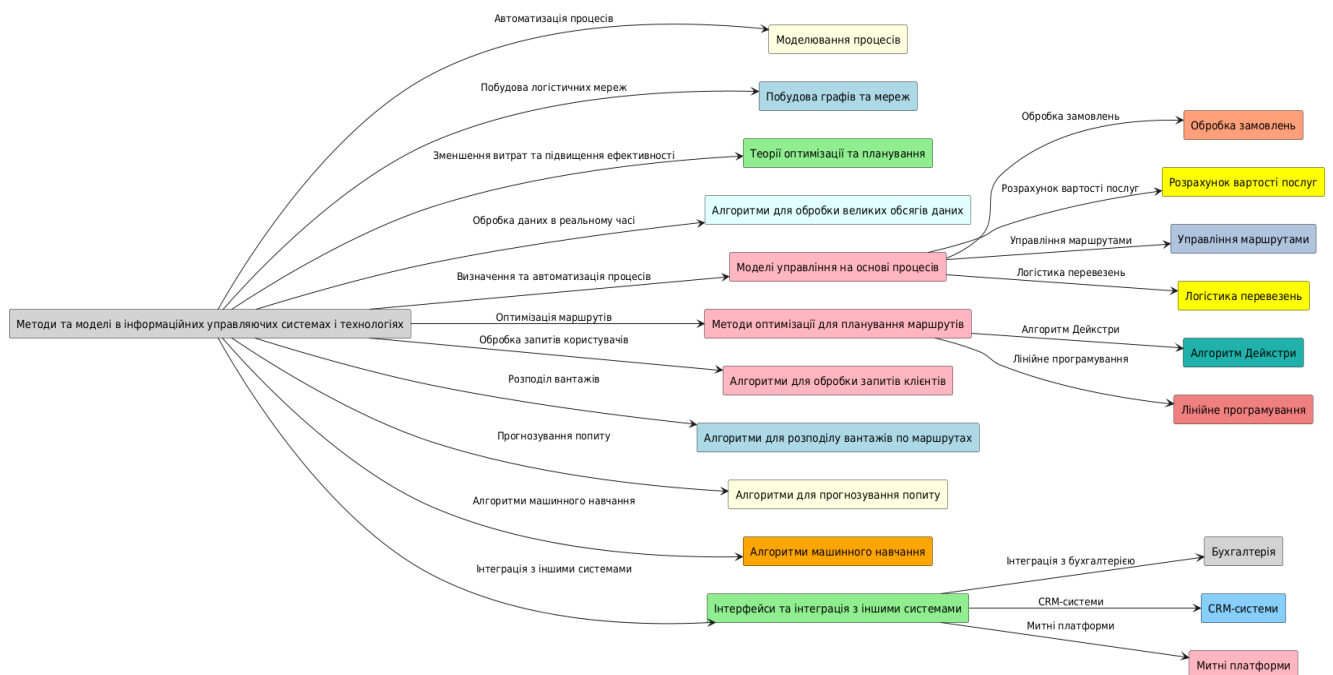


Рисунок 2.7 – Методи та моделі в інформаційних управлюючих системах і технологіях

«Джерело»: [розроблено автором на підставі коду програми]

Інформаційні управлюючі системи побудовані на різних моделях, які дозволяють забезпечити інтеграцію та взаємодію між усіма компонентами системи. Однією з таких моделей є модель управління на основі процесів. Вона передбачає чітке визначення всіх процесів підприємства та їх автоматизацію через ІУС, що сприяє оптимізації діяльності. Наприклад, для транспортної компанії це може включати такі процеси, як обробка замовлень, розрахунок вартості послуг, управління маршрутами та логістика перевезень.

Основною перевагою цієї моделі є здатність відслідковувати стан кожного процесу в реальному часі та швидко коригувати стратегії при зміні умов. Тому моделі управління на основі процесів забезпечують високу гнучкість і адаптивність системи до змінюваних зовнішніх та внутрішніх умов. Вони дозволяють створювати системи, здатні реагувати на нові вимоги та умови діяльності

підприємства, що є особливо важливим для таких сфер, як транспортні послуги, де адаптація до змін є критично важливою.

Одним із важливих напрямків у розвитку ІУС є використання методів оптимізації, зокрема, для планування маршрутів, управління запасами та ресурсами, що дозволяє знижувати витрати та збільшувати ефективність. Відповідно до особливостей діяльності транспортних компаній, одним із ключових елементів є оптимізація маршрутів доставки. Для цього застосовуються методи лінійного програмування, алгоритми пошуку найкоротшого шляху (наприклад, алгоритм Дейкстри) та методи планування з використанням графів.

Завдяки таким методам можливе оперативне планування перевезень з урахуванням різних параметрів: відстані, часу, вартості та типу транспорту. Наприклад, для міжміських або міжнародних перевезень, де можливе перетинання кордонів, враховуються не лише витрати на перевезення, а й митні тарифи, правила перевезення вантажів та інші фактори. Завдяки використанню цих алгоритмів, система здатна запропонувати найбільш оптимальні варіанти для користувачів, знижуючи їх витрати та покращуючи якість обслуговування.

Використання алгоритмів є невід'ємною частиною інформаційних систем, особливо при обробці великих обсягів даних. У системах для транспортних компаній застосовуються різні типи алгоритмів для автоматизації процесів обробки замовлень, розрахунку цін та управління послугами. Серед них можна виділити алгоритми для обробки запитів користувачів, розподілу вантажів по маршрутах, а також для прогнозування попиту на послуги.

Алгоритми машинного навчання можуть бути ефективно використані для прогнозування трендів на ринку транспортних послуг, аналізу попиту та пропозиції, визначення найбільш популярних маршрутів, а також для динамічної оптимізації тарифів з урахуванням сезонності, конкуренції та інших ринкових факторів. Це дозволяє створювати інтелектуальні системи, які самостійно адаптуються до змін в умовах бізнесу, мінімізуючи витрати та підвищуючи прибутковість. Наприклад, алгоритми кластеризації можуть допомогти ідентифікувати географічні зони з високим попитом, а прогнозні моделі, такі як

регресійний аналіз або нейронні мережі, здатні передбачати пікові періоди замовлень. Крім того, ці алгоритми можуть бути застосовані для прогнозування необхідних обсягів транспортування, що сприяє оптимальному розподілу транспортних засобів, зменшенню простоїв і підвищенню ефективності використання ресурсів. Також машинне навчання може допомогти у виявленні аномалій, таких як нестандартні замовлення або шахрайські дії, забезпечуючи додатковий рівень безпеки та надійності.

Система управління має забезпечувати зручний та інтуїтивно зрозумілий інтерфейс для взаємодії з користувачами, включаючи клієнтів, менеджерів і адміністраторів, а також для інтеграції з іншими системами. Важливою складовою є безшовна інтеграція з зовнішніми платформами, такими як бухгалтерські системи, CRM-системи, митні платформи, логістичні сервіси та бази даних геолокації. Наприклад, інтеграція з API митних служб може автоматизувати процеси оформлення документів, а підключення до CRM дозволяє відстежувати історію взаємодії з клієнтами та персоналізувати пропозиції. Вбудовані інтерфейси надають менеджерам та адміністраторам інструменти для ефективного управління послугами, моніторингу статусів замовлень у реальному часі, аналізу ключових показників ефективності (KPI) та швидкого реагування на зміни.

Методи та моделі, що застосовуються в інформаційних управляючих системах, є важливими для забезпечення ефективного функціонування транспортного підприємства. Використання моделей управління на основі процесів, оптимізаційних методів, алгоритмів для прогнозування та аналізу даних дозволяє значно підвищити ефективність операцій, знизити витрати та покращити обслуговування клієнтів. Інтеграція з іншими системами та впровадження сучасних технологій, таких як машинне навчання, відкриває нові горизонти для розвитку транспортних компаній та забезпечує їм конкурентні переваги на ринку. Завдяки таким підходам, компанії можуть адаптуватися до змінюваних умов і забезпечити високий рівень обслуговування для своїх клієнтів.

## РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ

### 3.1 Проєктування бази даних для інформаційної управляючої системи

Розробка ефективної бази даних є важливим етапом у створенні інформаційної системи для транспортного підприємства, адже вона визначає, як будуть зберігатися, організовуватися та оброблятися дані про послуги, замовлення, користувачів та інші аспекти діяльності компанії. Правильне проєктування бази даних гарантує її надійність, масштабованість та ефективність при використанні, що критично для забезпечення безперебійної роботи системи в майбутньому.

Проєктування бази даних є багатогранним процесом, який охоплює кілька ключових етапів, кожен із яких відіграє важливу роль у створенні ефективної та масштабованої системи. На рівні інфологічного моделювання розробники визначають основні сутності, їхні атрибути та зв'язки між ними, створюючи концептуальну модель даних, яка відображає бізнес-логіку системи. Цей етап є фундаментом для подальшого проєктування, оскільки забезпечує чітке розуміння структури даних і їхньої взаємодії. Наприклад, у наданому коді моделі User, Service і Order чітко визначені з відповідними полями та зв'язками, що відповідає принципам інфологічного моделювання.

Проєктування інфологічної моделі (рис. 3.1) є ключовим етапом створення бази даних, що визначає основні сутності предметної області, їх атрибути та взаємозв'язки. Для транспортної компанії це включає зберігання даних про замовлення, послуги, клієнтів, працівників, транспортні засоби, тарифи, координати відправлення та доставки, статуси замовлень і аналітичні показники. Надана модель у коді (сутності User, Service, Order) відображає ці вимоги, забезпечуючи ефективне управління бізнес-процесами, підтримку операційної діяльності та можливість аналізу даних для оптимізації логістичних операцій.

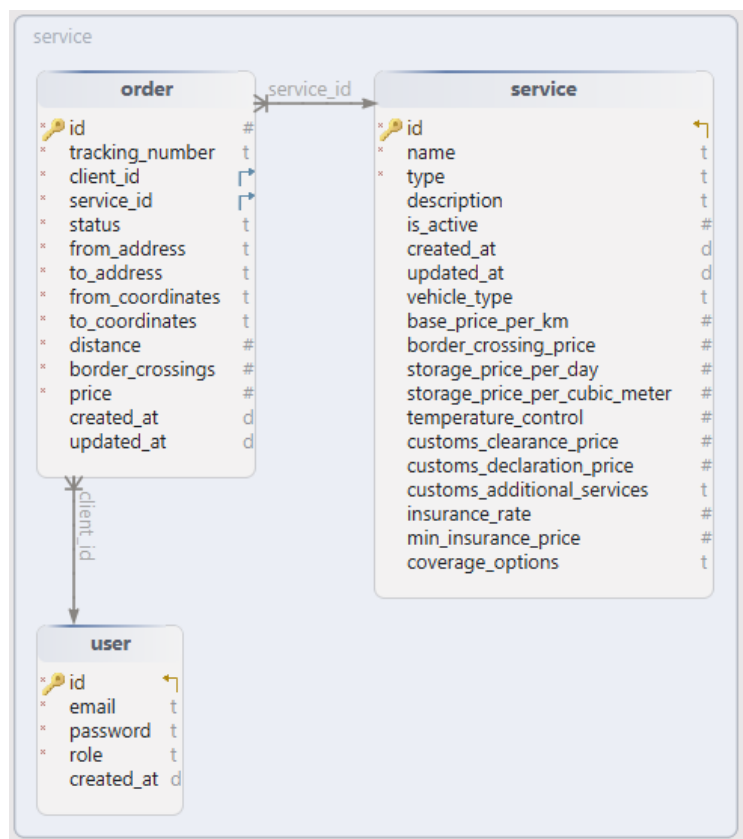


Рисунок 3.1 – Модель БД

«Джерело»: [розроблено автором на підставі реалізованого проекту]

Основні сутності для реалізованої системи описані в табл. 3.1.

Таблиця 3.1

Основні сутності БД для системи управління для транспортного підприємства

| № | Назва              | Опис                                                                                                                                                                                                                                                            |
|---|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2                  | 3                                                                                                                                                                                                                                                               |
| 1 | Користувач (user)  | дані про клієнтів, працівників або менеджерів, які використовують систему для взаємодії з іншими елементами бізнесу. Включають інформацію про email, пароль, роль користувача (клієнт або менеджер)                                                             |
| 2 | Послуга (service)  | описує доступні послуги, які надаються транспортною компанією, такі як міжміські та міжнародні перевезення, складське зберігання, митне оформлення, страхування вантажів. Ця сутність містить деталі послуг, такі як тип транспорту, тарифи, описи, доступність |
| 3 | Замовлення (order) | ця сутність зберігає інформацію про конкретні замовлення клієнтів. Вона містить дані про адреси доставки, координати, ціну, тип послуги та статус                                                                                                               |

«Джерело»: [розроблено автором]

Між сутностями існують певні зв'язки. Наприклад, кожен замовлення пов'язано з конкретним клієнтом та послугою. Для забезпечення цілісності даних використовуються зовнішні ключі, які з'єднують таблиці `order`, `service` та `user`. Для зберігання даних про послугу та її тарифи передбачені різноманітні поля, такі як ціна за км, ціна за перетин кордону, температура для складського зберігання тощо.

Для забезпечення нормалізації даних база даних приведена до третьої нормальної форми (3НФ). Наприклад, всі атрибути, які можна розкласти на незалежні елементи, виносяться в окремі таблиці. Це дозволяє уникнути дублювання даних і полегшує підтримку цілісності та консистентності бази.

На цьому етапі варто також зазначити, що вибір типів даних для кожної сутності здійснюється з урахуванням вимог щодо обсягу збережених даних, швидкості їх обробки та масштабу системи. Наприклад, для полів з цінами використовуються типи даних `FLOAT` або `DECIMAL`, для текстових полів — `VARCHAR` з обмеженням довжини.

Вибір системи керування базами даних (СКБД) є важливим кроком, оскільки він впливає на продуктивність, доступність і масштабованість інформаційної системи. Для даної транспортної компанії оптимальним вибором є використання реляційної бази даних, оскільки вона дозволяє ефективно зберігати та обробляти структуровані дані з чітко визначеними взаємозв'язками між сутностями.

Однією з найбільш популярних реляційних СКБД є PostgreSQL, яка є відкритим продуктом і підтримує високий рівень масштабованості та ефективність при роботі з великими обсягами даних. Її особливості, такі як підтримка транзакцій, можливість використання індексів, а також підтримка складних запитів, роблять її ідеальним вибором для даної системи. Крім того, PostgreSQL має зручні інструменти для інтеграції з іншими сервісами та системами, що може бути корисним при подальшому розширенні системи.

Проектування фізичної моделі бази даних здійснюється з урахуванням вибору СКБД та її можливостей. В таблицях створюються індекси для полів, які часто використовуються у запитах (наприклад, для поля `tracking_number` в таблиці

order або email в таблиці user). Це дозволяє прискорити пошук і покращити загальну продуктивність системи. Крім того, для забезпечення цілісності даних використовуються обмеження на зовнішні ключі, унікальність полів та обмеження на значення атрибутів.

Фізичне проектування також передбачає визначення типів даних для кожної колонки та максимальних розмірів полів, що забезпечує оптимальне використання пам'яті та ефективну обробку запитів.

На етапі обрахунку прогнозованого обсягу бази даних необхідно врахувати кількість записів, які будуть зберігатися в таблицях за певний час, а також обсяг даних, які будуть зберігатися для кожного запису. Для цього потрібно провести розрахунки, що допоможуть оцінити розмір бази даних і необхідні ресурси для її підтримки.

Прогнозування обсягів бази даних враховує кілька факторів, таких як кількість користувачів, послуг, замовлень, а також очікувану частоту змін даних (наприклад, кількість нових замовлень щомісяця). Наприклад, якщо щомісяця створюється 1000 нових замовлень, кожне з яких містить 100 байтів інформації, то для одного місяця додатковий обсяг бази складе 100 МБ. З урахуванням зростання числа записів за кілька місяців можна прогнозувати обсяг бази на кілька років вперед.

Прогнозні розрахунки обсягу БД дозволяють спланувати ресурси для зберігання даних, а також підготувати інфраструктуру для масштабування системи в майбутньому.

Проектування бази даних є важливим кроком у розробці інформаційної системи для транспортного підприємства. Інфологічне моделювання дозволяє точно визначити структуру даних та їх взаємозв'язки, що є основою для ефективної роботи системи. Вибір СКБД, таких як PostgreSQL, дає змогу забезпечити високу продуктивність та надійність бази даних, що важливо для будь-якої комерційної діяльності. Прогнозування обсягів бази даних дозволяє передбачити зростання системи і забезпечити її ефективну роботу в майбутньому.

### 3.2 Проектування бази знань та засобів інтелектуального аналізу даних

Інформаційна система управління транспортним підприємством вимагає ретельно спроектованої бази знань, що забезпечує ефективну обробку, аналіз та використання даних. Враховуючи складність логістичних процесів, важливим є застосування інтелектуальних засобів аналізу для оптимізації перевезень, зменшення витрат і підвищення якості обслуговування клієнтів. Впровадження таких технологій дозволяє автоматизувати процеси ухвалення рішень, виявляти закономірності в масивах даних та прогнозувати ключові показники ефективності.

База знань інформаційної системи транспортного підприємства виконує функцію центрального сховища структурованих та неструктурованих даних. Вона включає відомості про замовлення, послуги, клієнтів, транспортні засоби, маршрути та тарифи. Дані організовані у вигляді реляційної структури, що дозволяє ефективно виконувати запити, забезпечувати їх узгодженість і цілісність. Одним із ключових принципів її побудови є нормалізація, яка допомагає уникнути дублювання інформації та підвищує швидкість обробки запитів.

Важливим компонентом є система класифікації послуг, що включає міжміські, міжнародні перевезення, митне оформлення, складське зберігання та страхування вантажів. Використання класифікаторів дозволяє автоматизувати процеси вибору оптимального тарифу, підбору транспорту та формування відповідних комерційних пропозицій.

Для покращення процесу ухвалення рішень застосовуються методи інтелектуального аналізу даних, що базуються на машинному навчанні, статистичних моделях та аналітичних алгоритмах. Зокрема, використовується алгоритм кластеризації для групування схожих замовлень, що дозволяє оптимізувати маршрути перевезень. Методи прогнозного аналізу на основі регресійних моделей допомагають визначати попит на послуги в різних регіонах та сезонах.

Особлива увага приділяється аналізу історичних даних для виявлення закономірностей, які можуть вплинути на вартість та ефективність

транспортування. Наприклад, аналіз часових рядів дозволяє прогнозувати коливання попиту залежно від часу року, економічної ситуації та інших зовнішніх факторів. Використання таких підходів сприяє зменшенню витрат за рахунок раціонального розподілу ресурсів.

Розробка засобів обробки знань передбачає інтеграцію експертних систем, що допомагають менеджерам ухвалювати оптимальні рішення. Експертна система оцінює різні параметри перевезення, зокрема вартість, швидкість, ризики та доцільність використання певного типу транспорту.

Використання алгоритмів оптимізації забезпечує динамічний розподіл транспортних ресурсів залежно від навантаження, часу доби та поточних замовлень. Наприклад, застосування генетичних алгоритмів дозволяє знаходити оптимальні маршрути перевезення з урахуванням обмежень, таких як час доставки, доступність транспорту та вимоги до зберігання товарів.

Система підтримки рішень інтегрована з базою знань, що дозволяє автоматично формувати рекомендації для диспетчерів та менеджерів. Зокрема, на основі аналізу історичних даних система може пропонувати найбільш економічно вигідні варіанти транспортування або нагадувати про необхідність оновлення тарифів.

Для контролю за ефективністю роботи системи використовується модуль моніторингу, що забезпечує візуалізацію ключових показників діяльності компанії. Графічні інтерфейси дозволяють відображати статистичні дані у вигляді графіків, діаграм та теплових карт, що полегшує аналіз і сприяє прийняттю зважених рішень.

Моніторинг транспортних засобів здійснюється за допомогою GPS-трекерів, дані яких інтегруються у систему управління. Це дозволяє в реальному часі відстежувати переміщення вантажів, оперативно реагувати на затримки та забезпечувати прозорість логістичних процесів.

Проектування бази знань та засобів інтелектуального аналізу даних є ключовим етапом у розробленні інформаційної системи управління транспортним підприємством. Оптимізація структури даних, застосування методів аналізу та використання автоматизованих систем підтримки рішень забезпечують

ефективність роботи підприємства, зменшують витрати та підвищують якість обслуговування клієнтів.

Реалізація запропонованих рішень дозволяє підвищити рівень автоматизації та мінімізувати людський фактор у процесах планування, аналізу та прогнозування логістичних операцій. Подальше вдосконалення системи може включати інтеграцію штучного інтелекту для розпізнавання патернів та адаптацію алгоритмів оптимізації відповідно до змін у ринкових умовах.

### **3.2.1 Реалізація інтелектуального аналізу великих даних для управління транспортом**

У сучасному світі транспортні підприємства стикаються з необхідністю обробки та аналізу величезних обсягів даних, що виникають у процесі їхньої діяльності. Застосування інтелектуального аналізу великих даних (Big Data) дозволяє підвищити ефективність управління транспортними процесами, оптимізувати логістичні операції та приймати обґрунтовані рішення на основі отриманих знань.

Інтелектуальний аналіз даних (Data Mining) охоплює процеси збору, обробки та інтерпретації великих обсягів інформації з метою виявлення прихованих закономірностей та тенденцій. У транспортній галузі це може включати аналіз маршрутів, часу доставки, витрат палива, поведінки водіїв та інших параметрів, що впливають на ефективність перевезень.

Застосування методів інтелектуального аналізу даних дозволяє виявити неочевидні взаємозв'язки між різними факторами, що впливають на роботу транспортного підприємства. Наприклад, аналіз даних про рух транспортних засобів може допомогти визначити оптимальні маршрути, зменшити час простою та підвищити загальну ефективність логістичних операцій.

Розробка інформаційної системи управління для транспортного підприємства передбачає інтеграцію модулів інтелектуального аналізу даних для підтримки прийняття рішень. Це включає збір даних з різних джерел, їхню обробку

та аналіз з використанням сучасних алгоритмів машинного навчання та статистичних методів.

Одним із ключових етапів є створення модулів для збору та зберігання даних. Це можуть бути дані про замовлення, маршрути, стан транспортних засобів, витрати палива та інші параметри, що впливають на роботу підприємства. Зібрані дані зберігаються у базі даних, що забезпечує їхню доступність для подальшого аналізу.

Наступним кроком є розробка алгоритмів для обробки та аналізу даних. Це можуть бути методи кластеризації для групування схожих маршрутів, регресійний аналіз для прогнозування витрат палива або нейронні мережі для виявлення складних взаємозв'язків між різними параметрами.

Результати аналізу можуть бути представлені у вигляді звітів, графіків або інтерактивних панелей, що дозволяють менеджерам підприємства швидко отримувати необхідну інформацію для прийняття рішень. Наприклад, система може надавати рекомендації щодо оптимальних маршрутів, прогнозувати можливі затримки або виявляти аномалії у роботі транспортних засобів.

Впровадження інтелектуального аналізу даних у транспортну інформаційну систему надає низку переваг, що описані в табл. 3.2.

Таблиця 3.2

Переваги впровадження інтелектуального аналізу даних у транспортну інформаційну систему

| № | Назва                 | Опис                                                                                                                                                            |
|---|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Оптимізація маршрутів | Аналіз даних про трафік, погодні умови та інші фактори дозволяє визначити найбільш ефективні маршрути для перевезень, що зменшує час доставки та витрати палива |
| 2 | Підвищення безпеки    | Моніторинг поведінки водіїв та стану транспортних засобів допомагає виявляти потенційні ризики та запобігати аваріям                                            |
| 3 | Прогнозування попиту  | Аналіз історичних даних про замовлення дозволяє прогнозувати майбутній попит на послуги та планувати ресурси відповідно                                         |
| 4 | Зниження витрат       | Оптимізація операційних процесів та ефективне використання ресурсів сприяють зменшенню загальних витрат підприємства                                            |

«Джерело»: [розроблено автором]

Незважаючи на очевидні переваги, впровадження інтелектуального аналізу даних у транспортну галузь супроводжується низкою викликів. Серед них – необхідність обробки великих обсягів даних у реальному часі, забезпечення безпеки та конфіденційності інформації, а також інтеграція з існуючими системами управління.

Перспективи розвитку включають впровадження більш складних алгоритмів машинного навчання, використання даних з нових джерел (наприклад, Інтернету речей) та підвищення рівня автоматизації процесів прийняття рішень.

Інтеграція інтелектуального аналізу великих даних у інформаційну систему управління транспортним підприємством дозволяє підвищити ефективність операцій, знизити витрати та покращити якість обслуговування клієнтів. Сучасні технології аналізу даних надають потужні інструменти для підтримки прийняття рішень та забезпечення конкурентних переваг у динамічному середовищі транспортної галузі.

Використання інтелектуального аналізу даних стає невід'ємною частиною сучасного управління транспортними підприємствами, сприяючи їхньому сталому розвитку та адаптації до змінних умов ринку.

### **3.2.2 Проєктування та реалізація бази знань системи**

Інформаційна система управління транспортним підприємством забезпечує автоматизацію ключових бізнес-процесів, зокрема обробку замовлень, управління послугами, аналітику та адміністрування користувачів. Важливою складовою цієї системи є база знань, яка містить структуровану інформацію про логістичні послуги, тарифи, типи транспорту, складські можливості та митне оформлення. Проєктування та реалізація бази знань визначають ефективність функціонування програмного комплексу, його здатність до розширення та адаптації під змінні бізнес-вимоги.

Архітектура бази знань зображена на рис. 3.2.

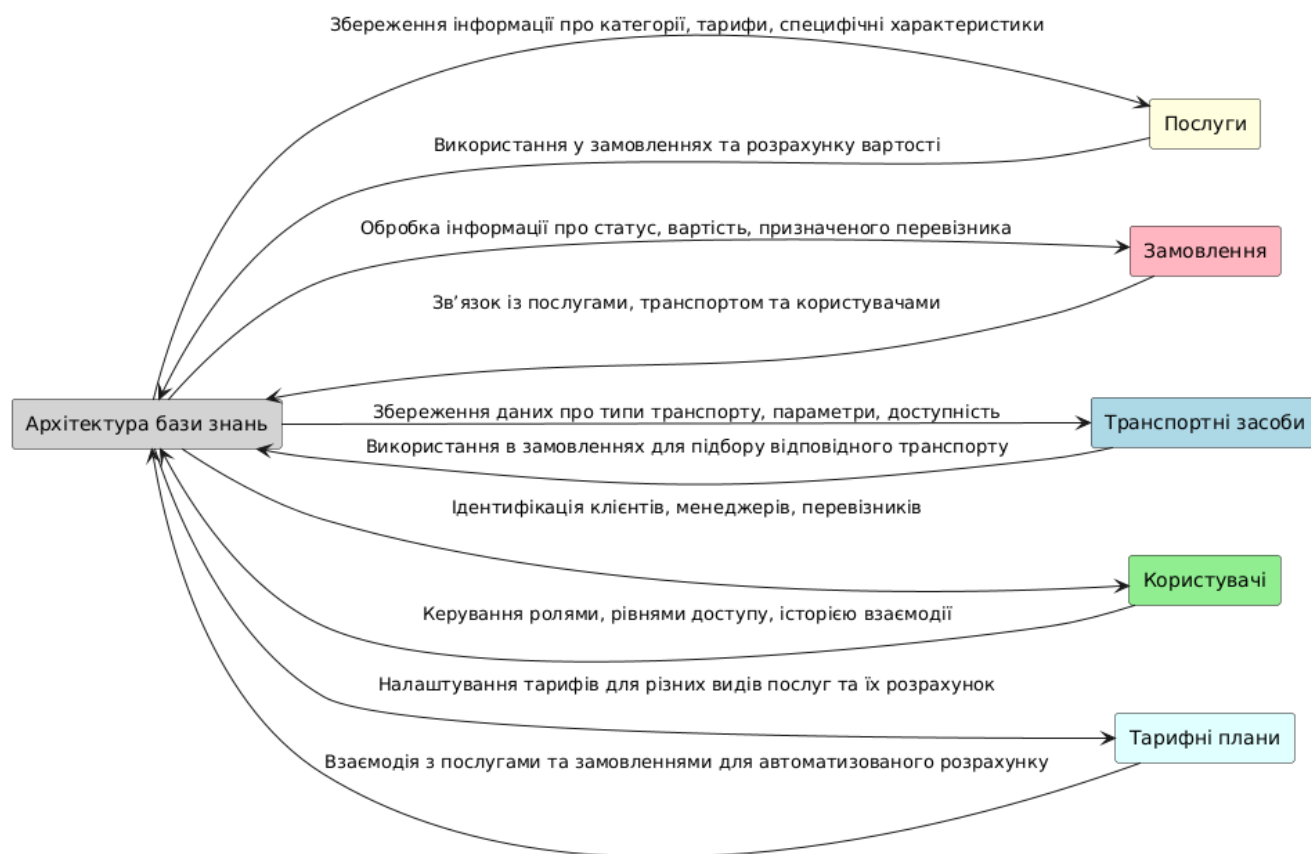


Рисунок 3.2 – Архітектура бази знань

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

База знань формується як сукупність взаємопов'язаних даних, що зберігаються в реляційній структурі та використовуються для прийняття управлінських рішень. Основними сутностями є послуги, замовлення, транспортні засоби, користувачі та тарифні плани. Для забезпечення ефективного доступу до інформації застосовується система ідентифікаторів, що дозволяє швидко взаємодіяти з необхідними даними та здійснювати їх фільтрацію відповідно до запитів користувачів.

Дані про послуги зберігають інформацію про їхню категорію, назву, опис, тарифи та специфічні характеристики. Наприклад, для вантажних перевезень передбачено збереження інформації про тип транспорту, вартість перевезення за кілометр та додаткові збори, такі як оплата перетину кордону. Складське

зберігання містить дані про вартість зберігання на день або за кубічний метр, а також інформацію про наявність температурного контролю. Для митного оформлення передбачено базову вартість послуг, витрати на декларацію та перелік додаткових послуг.

Інформація про послуги організована у вигляді вкладених сутностей. Для збереження та обробки використовується система взаємопов'язаних таблиць, що забезпечує логічну ієрархію та взаємозв'язок між категоріями даних. Наприклад, таблиця `services` містить базову інформацію про послугу, тоді як допоміжні таблиці, такі як `transport_services`, `storage_services`, `customs_services` та `insurance_services`, доповнюють її специфічними атрибутами для кожного типу діяльності.

Формування структури бази знань передбачає чітке визначення атрибутів, їхніх типів даних та правил зв'язку між таблицями. Наприклад, поле `vehicle_type` в таблиці `transport_services` містить значення, що відповідають категоріям вантажного транспорту, які представлені у словнику відповідності між типами та графічними позначеннями. Це забезпечує зручне відображення інформації для користувачів та зменшує ймовірність помилок при внесенні даних.

Доступ до бази знань реалізується через API-запити, що дозволяють додавати нові послуги, оновлювати інформацію та отримувати дані про вже існуючі пропозиції. Використовується механізм аутентифікації та авторизації, що забезпечує контроль доступу до даних залежно від ролі користувача. Менеджер підприємства має можливість створювати, змінювати та деактивувати послуги, тоді як клієнти можуть лише переглядати доступні варіанти та оформлювати замовлення.

Обробка запитів передбачає використання ключових параметрів, які дозволяють отримувати вибірки даних відповідно до заданих критеріїв. Наприклад, для відображення всіх активних послуг виконується запит із фільтрацією за полем `is_active`, що дозволяє виключити з перегляду деактивовані пропозиції.

Для підтримання актуальності даних реалізовано механізм автоматичного оновлення інформації. Менеджери мають можливість змінювати характеристики

послуг, а система забезпечує збереження змін та їхнє відображення в реальному часі. При внесенні коригувань проводиться перевірка правильності введених даних, що дозволяє уникнути конфліктів у базі знань.

Для видалення інформації передбачена функція деактивації, що дозволяє тимчасово приховати певні послуги без втрати історичних записів. Це особливо важливо для збереження інформації про минулі транзакції та забезпечення аналітичного аналізу на основі попередніх даних.

Проектування та реалізація бази знань для інформаційної системи управління транспортним підприємством є важливим етапом створення ефективного програмного забезпечення. Використання реляційної структури даних забезпечує зручність збереження та швидкість доступу до необхідної інформації. Інтеграція бази знань з інтерфейсом користувача дозволяє автоматизувати процеси управління послугами, що сприяє підвищенню продуктивності роботи підприємства. Механізми аутентифікації та авторизації гарантують безпеку даних, а системи перевірки та оновлення забезпечують їхню достовірність.

Таким чином, реалізація бази знань не лише впорядковує інформаційні потоки всередині компанії, а й дозволяє оперативно адаптуватися до змін ринку, оптимізуючи взаємодію між менеджерами та клієнтами.

### **3.3 Програмне забезпечення**

У процесі розробки програмного забезпечення для транспортної компанії, основною метою є створення ефективного, функціонального та інтегрованого рішення, яке зможе забезпечити повноцінну автоматизацію управлінських і операційних процесів підприємства. У цьому контексті програмне забезпечення стає основним інструментом для менеджерів, водіїв, клієнтів і адміністрації компанії, забезпечуючи зручний доступ до різноманітних функцій, таких як управління послугами, замовленнями, відслідковуванням вантажів, а також аналітика та звітність.

Система повинна відповідати вимогам безпеки, зручності у використанні та надійності, адже вона буде опрацьовувати значну кількість даних і забезпечувати взаємодію між різними користувачами. У своїй основі програмне забезпечення для управління транспортною компанією має включати кілька основних компонентів: систему реєстрації та аутентифікації користувачів, інтерактивні інтерфейси для введення й відображення даних, а також можливість взаємодії з іншими компонентами системи, такими як база даних, сервіси для зберігання інформації про замовлення та послуги, а також інтерфейси для менеджерів.

Програмний продукт для транспортної компанії реалізовано за допомогою фреймворка Streamlit, який є сучасним інструментом для розробки веб-додатків, що дозволяють зручний доступ до бізнес-даних у режимі реального часу. Завдяки цьому, створення веб-додатків є швидким і зручним, без необхідності залучення великої кількості сторонніх бібліотек або складної інфраструктури. Це дозволяє значно зменшити витрати на розробку та обслуговування системи.

Програмне забезпечення має кілька основних сторінок, які виконують різні функції в межах одного додатка. Кожна сторінка забезпечує певний функціонал, зокрема для клієнтів, менеджерів і адміністраторів. Окремо виділяється механізм для управління послугами компанії, замовленнями, а також можливість перегляду та зміни статусу наявних послуг. Крім того, в додатку реалізовано функцію для створення замовлень, їх обробки та управління доставкою.

Однією з основних складових програмного забезпечення є сторінка для управління послугами, яка призначена для менеджерів транспортної компанії. Вона дозволяє створювати нові послуги для міжміських та міжнародних перевезень, організовувати складське зберігання вантажів, а також забезпечувати митне оформлення та страхування вантажів. Створення нової послуги включає введення назви, опису послуги, вибір типу транспорту, а також встановлення базових тарифів і умов.

Крім цього, на сторінці для управління послугами реалізовано можливість перегляду існуючих послуг із детальним описом, а також наявність кнопок для активації або деактивації послуг, що дозволяє менеджерам швидко змінювати

доступність послуг для клієнтів. Для кожної послуги передбачено функціонал для внесення змін, наприклад, коригування ціни, додавання нових характеристик або видалення послуги з системи. Це дозволяє легко налаштовувати пропозиції компанії під змінювані умови на ринку та потреби клієнтів.

Іншим важливим компонентом програмного забезпечення є механізм для управління замовленнями. Клієнти можуть створювати замовлення на різні послуги, що включають перевезення вантажів, організацію складського зберігання, митне оформлення чи страхування вантажів. Кожне замовлення має статус, який можна відстежувати в реальному часі, що дає змогу відстежувати процес виконання замовлення і забезпечувати точність виконання умов договору.

Менеджери мають доступ до сторінки з усіма замовленнями, де можуть переглядати їхній статус, координувати з клієнтами, а також перевіряти деталі замовлення для уточнення та коригування деталей. Система дозволяє швидко реагувати на будь-які зміни чи проблеми, що виникають під час виконання замовлення, таким чином забезпечуючи високий рівень обслуговування клієнтів.

Для підтримки прийняття управлінських рішень програмне забезпечення включає розділ аналітики, який дозволяє збирати статистичні дані про виконані замовлення, популярність послуг, середній дохід і витрати компанії. Завдяки цьому менеджери можуть отримати необхідну інформацію для оцінки ефективності роботи компанії, виявлення слабких місць у процесах і формування стратегії для подальшого розвитку бізнесу.

Розділ аналітики включає в себе різноманітні графіки, діаграми та таблиці, що дозволяють візуалізувати результати роботи компанії. Дані в цьому розділі зберігаються в базі даних і оновлюються в режимі реального часу, що забезпечує актуальність та точність інформації.

Програмне забезпечення, створене для управління транспортною компанією, базується на використанні Python та бібліотеки Streamlit, що дозволяє швидко створювати інтерактивні веб-додатки з мінімальними зусиллями. Використання цієї бібліотеки дозволяє створити гнучкий інтерфейс, який можна

легко адаптувати під потреби користувача. Всі дані зберігаються в базі даних, що дозволяє забезпечити надійне зберігання та обробку інформації.

Для забезпечення безпеки доступу до системи використовуються механізми аутентифікації, що забезпечують захист персональних даних клієнтів і співробітників компанії. Для цього передбачено використання токенів доступу, що дозволяють перевіряти автентичність користувачів і контролювати їхні права доступу до різних частин системи.

Програмне забезпечення для транспортної компанії створене з метою автоматизації та оптимізації управлінських процесів, що дозволяє покращити ефективність роботи компанії, знизити витрати та підвищити рівень обслуговування клієнтів. Завдяки зручному інтерфейсу та інтеграції з іншими системами, програмне забезпечення дозволяє легко здійснювати моніторинг послуг, замовлень і фінансових показників, що сприяє швидкому прийняттю рішень і адаптації до змінюваних умов ринку.

Таким чином, розробка програмного забезпечення для транспортної компанії стала важливим етапом у процесі оптимізації бізнес-процесів, що дозволяє значно покращити якості обслуговування клієнтів і забезпечити гнучкість у відповіді на нові виклики ринку.

### **3.3.1 Проєктування програмного забезпечення**

Інформаційна система управління для транспортного підприємства є важливим елементом для оптимізації внутрішніх процесів компанії, зокрема у сфері управління послугами, замовленнями та аналізу діяльності. Проєктування програмного забезпечення для такої системи включає в себе етапи планування, моделювання, вибору архітектурних рішень та реалізації. Основна мета цієї системи — забезпечити інтерактивність і зручність використання для всіх типів користувачів: клієнтів і менеджерів, автоматизувати процеси обробки замовлень, обслуговування клієнтів, управління послугами та здійснення аналітики.

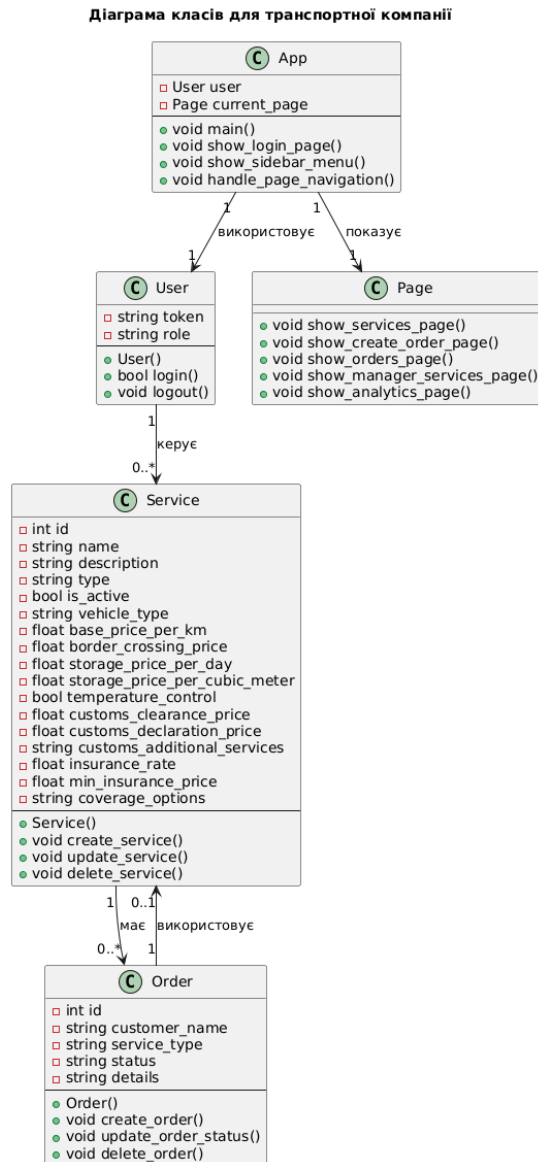
При розробці програмного забезпечення для транспортної компанії важливим є правильний вибір архітектури системи. Враховуючи специфіку діяльності підприємства, система повинна бути гнучкою, масштабованою та зручною для користувачів. У нашому випадку програмне забезпечення створено на основі веб-технологій, що дозволяє користувачам взаємодіяти з платформою через браузер на різних пристроях.

Вибір технології для розробки інтерфейсу був здійснений на основі популярності та зручності для створення інтерактивних веб-додатків — фреймворк Streamlit. Цей інструмент дозволяє без особливих труднощів реалізувати компоненти, необхідні для роботи з користувачами, а також інтегрувати необхідні функції без великих витрат часу на налаштування.

Інтерфейс передбачає використання вкладок, що дозволяє менеджеру та клієнту взаємодіяти з різними модулями системи, зокрема для управління послугами, створення замовлень та їх обробки. Крім того, система підтримує ролі користувачів, що дозволяє доступ до різних функціональних блоків залежно від типу користувача (клієнт або менеджер).

Архітектура системи побудована на модульному підході, що забезпечує чіткий розподіл функціональних обов'язків між компонентами, сприяючи ефективному управлінню даними та операційними процесами. Кожен модуль системи спроектовано для виконання специфічних завдань, що підвищує гнучкість, масштабованість та зручність підтримки програмного забезпечення. Зокрема, серверна частина (server.py) відповідає за обробку запитів, управління базою даних та забезпечення безпеки через JWT-аутентифікацію, тоді як клієнтська частина (client.py) реалізована з використанням Streamlit для створення інтерактивного веб-інтерфейсу. Скрипт seed\_db.py забезпечує початкове наповнення бази даних тестовими даними, що полегшує розробку та тестування.

Діаграма класів (рис. 3.3) ілюструє основні об'єкти системи, їх атрибути та взаємозв'язки, а також допомагає візуалізувати структуру програмного забезпечення.



**Рисунок 3.3 – Діаграма класів**

*«Джерело»: [розроблено автором на підставі реалізованого проєкту]*

Однією з головних складових програмного забезпечення є можливість управління послугами. Це дозволяє менеджеру ефективно створювати, редагувати, активувати/деактивувати або видаляти різноманітні послуги, пов'язані з транспортними перевезеннями, зберіганням, митним оформленням та страхуванням вантажів. Для кожної з цих послуг передбачено створення форми, в якій менеджер може ввести всі необхідні параметри, такі як назва, опис, тип транспорту, ціна за кілометр або день, температура зберігання тощо.

Кожна послуга є структурованою одиницею даних, яка зберігається у базі даних і може бути легко оновлена або видалена. Завдяки чітко визначеній структурі

даних, кожен елемент системи може бути зв'язаний з іншими компонентами, що дозволяє підтримувати актуальність інформації.

Реалізація таких функцій вимагає інтеграції з сервером для зберігання інформації про послуги та можливості для керування цими послугами. База даних використовується для забезпечення надійного зберігання даних, таких як базові ціни, транспортні засоби, типи послуг тощо. Для підвищення ефективності роботи системи передбачено використання API для віддаленого доступу до даних і можливості створення нових послуг або оновлення існуючих через інтерфейс користувача.

Для кожної з функцій системи були розроблені відповідні алгоритми, що забезпечують обробку даних і виконання операцій, зокрема, створення, редагування та видалення послуг. Діаграма варіантів використання (рис. 3.4) показує основні функції, які доступні користувачам системи та взаємодії між різними учасниками процесу.

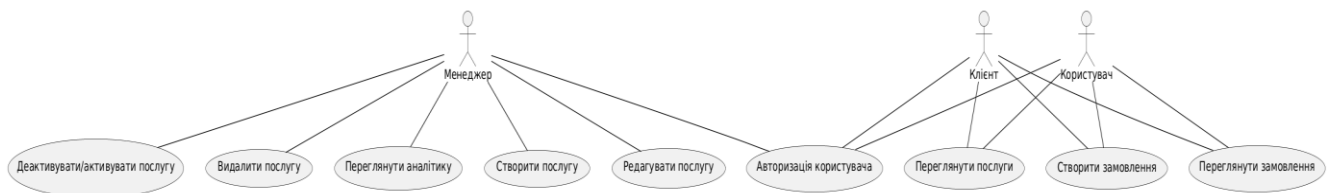


Рисунок 3.4 – Діаграма варіантів використання

*«Джерело»: [розроблено автором на підставі реалізованого проєкту]*

Ще однією важливою складовою інформаційної системи є модуль для обробки замовлень. Це дозволяє клієнтам компанії створювати замовлення на транспортні послуги, вибрати тип перевезення, вказувати місце відправлення та призначення, обирати додаткові послуги (наприклад, страхування вантажів або митне оформлення) і сплачувати за послугу.

Для забезпечення зручності використання користувачам надано інтуїтивно зрозумілий інтерфейс, що включає форми для введення даних замовлення. Замовлення створюються в декілька кроків, зокрема вибір послуги, вказівка даних вантажу, а також можливість підтвердити замовлення за допомогою кнопки

"створити замовлення". Після цього система зберігає інформацію про замовлення, прив'язує її до клієнта і дозволяє менеджеру здійснювати подальше управління цим замовленням (наприклад, зміна статусу або підтвердження виконання).

Особливістю такого підходу є наявність автоматичних сповіщень для клієнта та менеджера, що дозволяють своєчасно отримати інформацію про зміни у статусі замовлення. Завдяки цьому процес обробки замовлення стає більш прозорим та ефективним, знижуючи ймовірність помилок або затримок.

У системі для клієнтів створено зручний інтерфейс для оформлення замовлень і вибору послуг. Діаграма станів (рис. 3.5) ілюструє можливі стани замовлення в процесі обробки та зміни статусу, що допомагає зрозуміти логіку роботи системи в реальному часі.

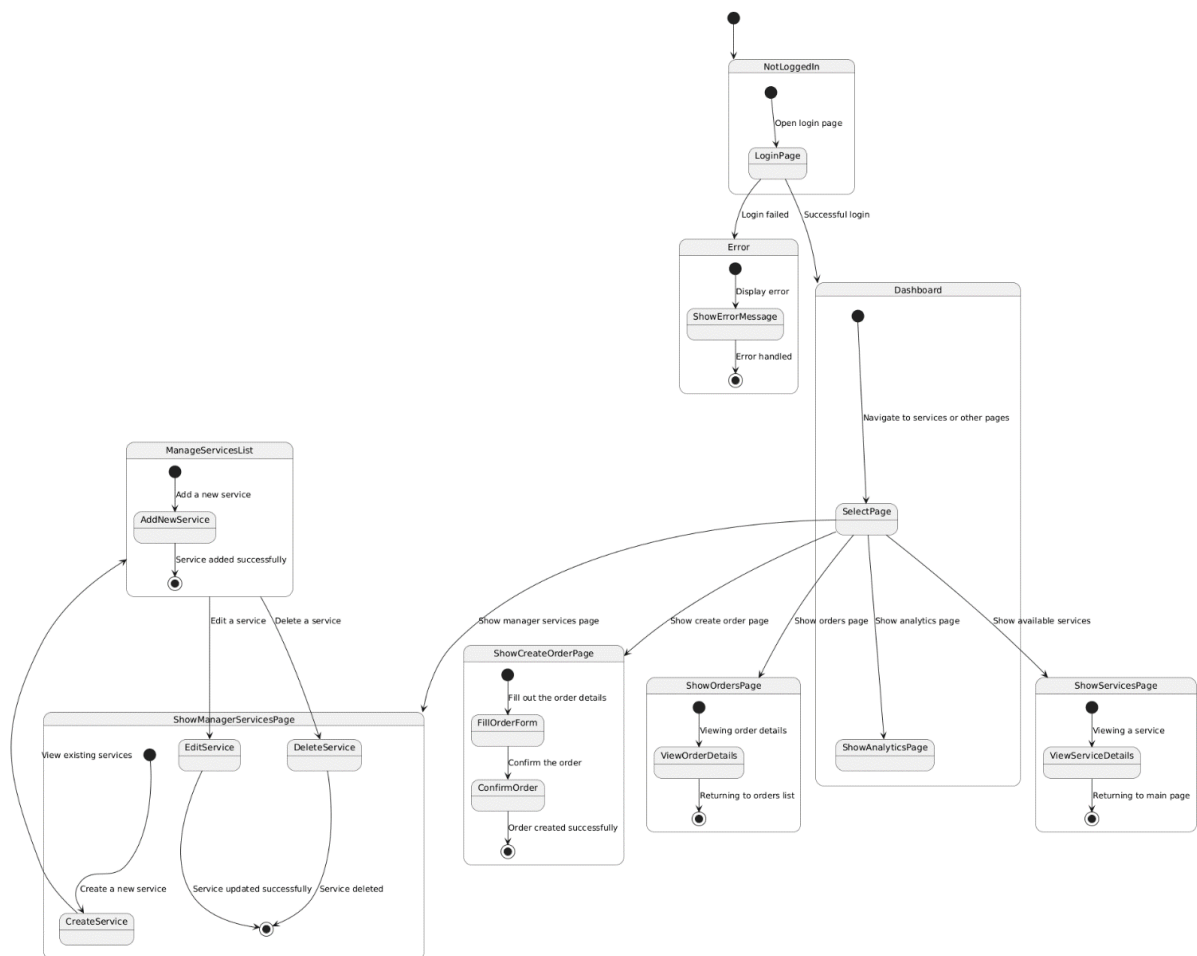


Рисунок 3.5 – Діаграма станів

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Для того щоб підприємство могло здійснювати контроль над своєю діяльністю та ефективно приймати управлінські рішення, в систему була інтегрована функція аналітики. Завдяки цьому модулю менеджери мають змогу отримувати статистичні дані, що стосуються замовлень, активності послуг, а також витрат на перевезення та інші важливі показники.

Реалізація аналітичних функцій здійснюється за допомогою спеціальних графіків, діаграм та таблиць, що дозволяє візуалізувати дані і отримувати важливу інформацію в зручному для розуміння вигляді. Це дає змогу оцінювати ефективність роботи різних послуг та знаходити найбільш вигідні стратегії для розвитку бізнесу.

Для аналізу даних та генерації звітів система використовує різноманітні інструменти візуалізації. Діаграма компонентів (рис. 3.6) відображає основні програмні блоки, що відповідають за обробку та зберігання даних для аналітичних цілей.

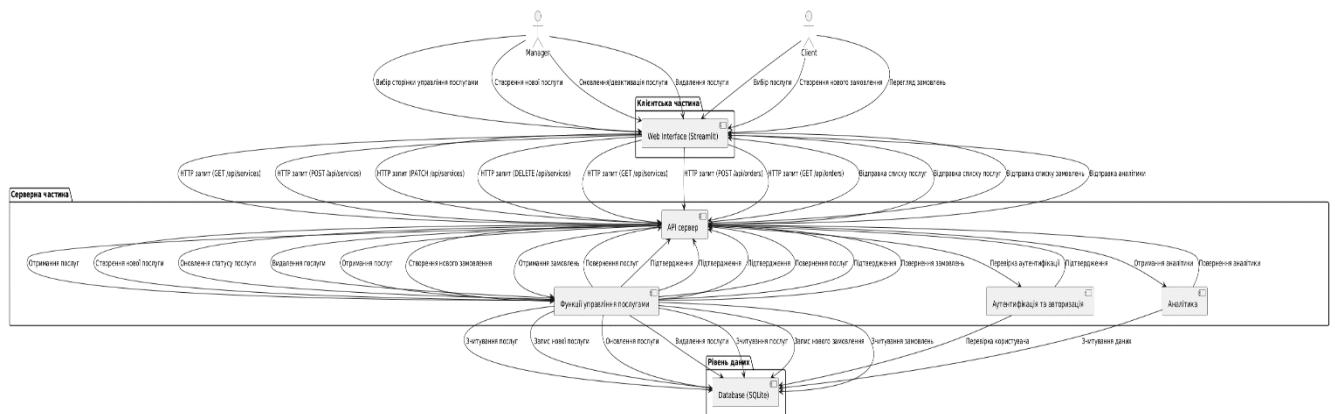


Рисунок 3.6 – Діаграма компонентів

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Для реалізації цієї інформаційної системи були застосовані сучасні підходи до розробки веб-додатків. Такі технології, як Python, Streamlit і API, дозволяють забезпечити зручність і швидкість розробки, а також забезпечити необхідну інтеграцію для роботи з базою даних і сторонніми сервісами.

Уся система має бути адаптивною та легко налаштовуваною, щоб згодом можна було розширювати функціонал без значних змін в основній архітектурі. Це дозволяє додавати нові послуги, можливості для клієнтів і менеджерів, а також підключати додаткові модулі для управління іншими аспектами діяльності підприємства.

Програмне забезпечення реалізовано з використанням новітніх веб-технологій, що забезпечують високу ефективність та гнучкість системи. Діаграма кооперацій (рис. 3.7) демонструє взаємодію компонентів системи та їх спільну роботу для виконання основних операцій.

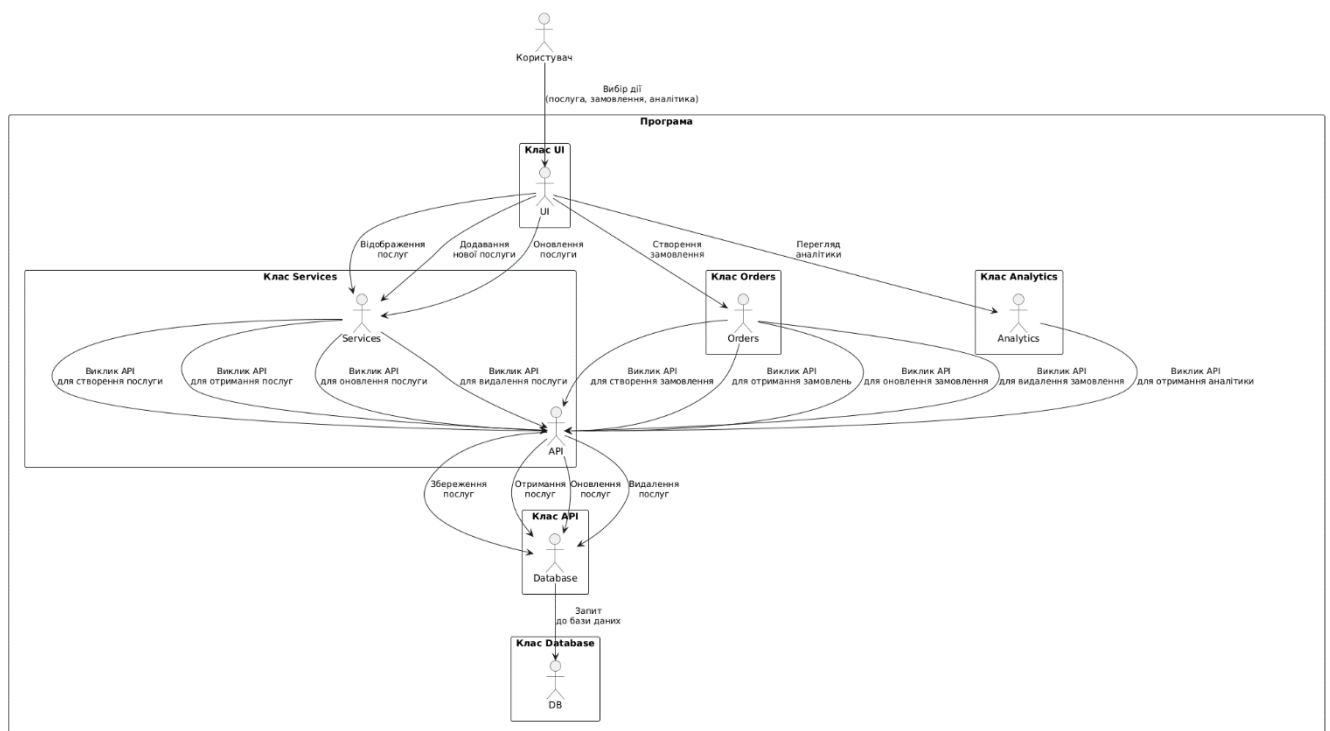


Рисунок 3.7 – Діаграма кооперацій

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Проектування програмного забезпечення передбачає чітке визначення компонентів, функцій і взаємодій, що забезпечують успішне впровадження та ефективність системи. Діаграма діяльності (рис. 3.8) відображає основні етапи виконання бізнес-процесів у системі, а діаграма розгортання (рис. 3.9) показує

структуру фізичних компонентів програмного забезпечення в реальній інфраструктурі підприємства.

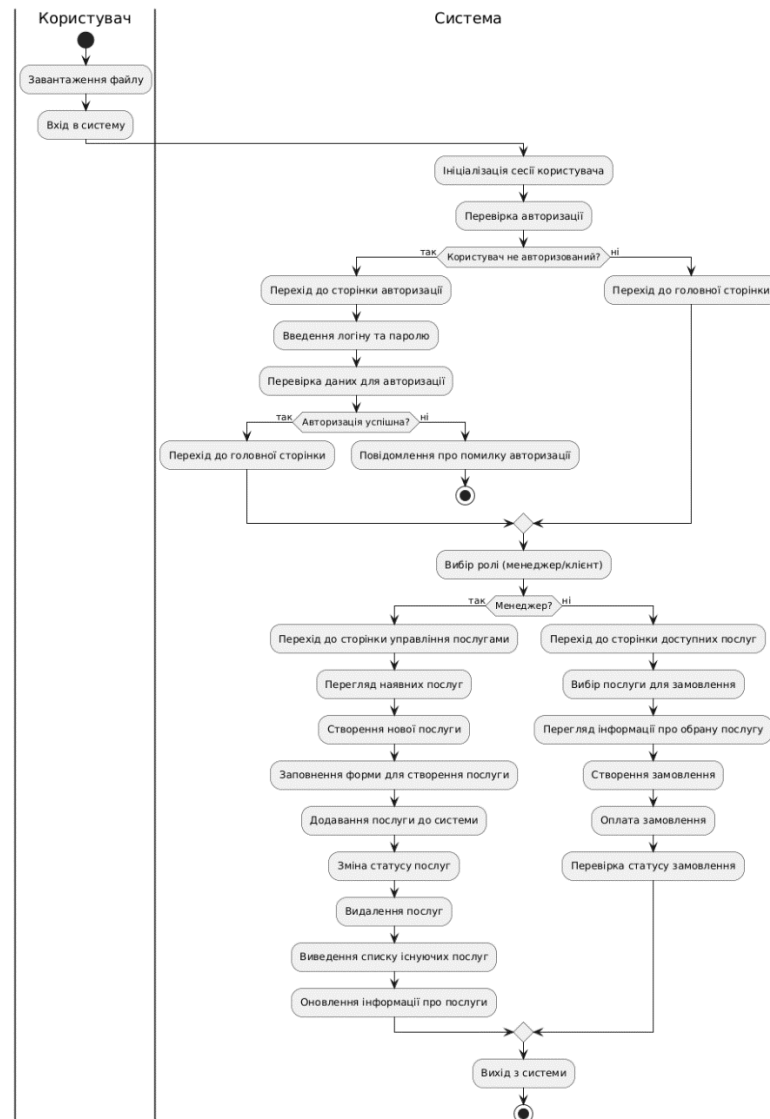


Рисунок 3.8 – Діаграма діяльності

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

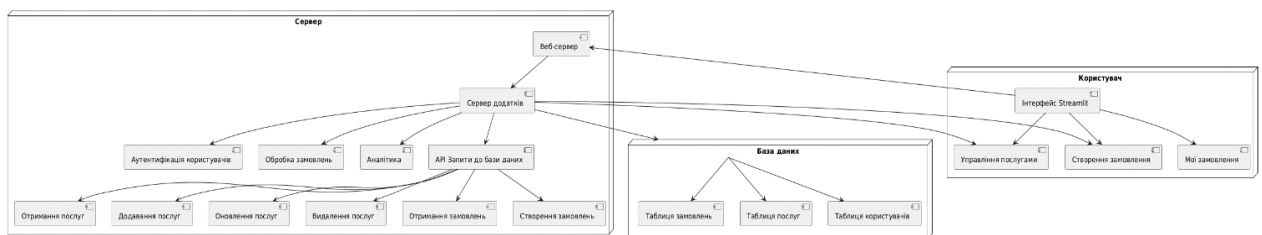


Рисунок 3.9 – Діаграма розгортання

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Діаграма послідовностей (рис. 3.10) допомагає візуалізувати порядок виконання основних операцій та взаємодію між об'єктами системи під час обробки запитів.



Рисунок 3.10 – Діаграма послідовностей

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Проектування програмного забезпечення для транспортного підприємства передбачає створення інтегрованої системи, що дозволяє автоматизувати

управління послугами, обробку замовлень та здійснення аналітики. Система повинна бути гнучкою, масштабованою та інтуїтивно зрозумілою для всіх користувачів. Інтеграція сучасних веб-технологій забезпечує високу ефективність і зручність у використанні, що сприяє підвищенню ефективності роботи підприємства та покращенню взаємодії з клієнтами.

Для успішного розвитку системи потрібно зберігати принципи масштабованості, гнучкості та зручності, а також регулярно оновлювати програмне забезпечення для відповідності вимогам ринку та розвитку технологій.

### **3.3.2 Вибір архітектури програмного забезпечення**

Розробка програмного забезпечення для управління транспортним підприємством вимагає особливої уваги до вибору архітектури, оскільки вона визначає ефективність, масштабованість та гнучкість майбутньої системи. Оскільки підприємства цього профілю мають специфічні вимоги до роботи з великими обсягами даних, постійними змінами в управлінських процесах і необхідністю забезпечення взаємодії різних частин інформаційної системи, архітектура повинна бути продумана таким чином, щоб гарантувати безпеку, високий рівень обробки даних та зручність для кінцевого користувача. У цій частині буде розглянуто найбільш доцільну архітектуру програмного забезпечення, яка найбільш ефективно відповідає вимогам транспортного підприємства та забезпечує оптимальну інтеграцію з іншими службами і системами.

При проектуванні інформаційної системи для транспортного підприємства важливо орієнтуватися на архітектуру, яка забезпечить ефективну взаємодію різних компонентів та максимальну зручність для користувачів. Система повинна обробляти велику кількість даних, які постійно змінюються, зокрема, щодо послуг, замовлень, а також організовувати процеси для менеджерів та клієнтів.

З огляду на специфіку бізнесу та можливість інтеграції з іншими цифровими інструментами, оптимальним вибором є архітектура на основі веб-сервісів, з використанням багатошарової клієнт-серверної моделі. Клієнтська частина, яка

представлена веб-інтерфейсом, забезпечує взаємодію кінцевих користувачів з системою, надаючи їм зручний доступ до функцій. Серверна частина відповідає за обробку запитів, виконання бізнес-логіки та доступ до бази даних.

Одним із важливих елементів цієї архітектури є розділення користувачів на різні ролі, такі як клієнти та менеджери, що дозволяє налаштувати доступ до різних частин системи, відповідно до рівня повноважень користувача. Крім того, кожен користувач має доступ до інтерфейсу, що відповідає його потребам, зокрема:

- Клієнти можуть переглядати послуги, створювати замовлення та переглядати статус своїх замовлень.
- Менеджери можуть адмініструвати послуги, керувати замовленнями та виконувати аналітику для покращення ефективності роботи підприємства.

Такий підхід дозволяє зберігати контроль над доступом до інформації, водночас спрощуючи управлінські функції та взаємодію з кінцевими користувачами.

У рамках реалізованої архітектури, система розподіляється на кілька основних компонентів, кожен з яких відповідає за виконання певних завдань (табл. 3.3).

Таблиця 3.3

## Завдання компонентів реалізованої архітектури

| № | Компонент             | Завдання                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2                     | 3                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 1 | Інтерфейс користувача | Це частина, яка взаємодіє безпосередньо з кінцевими користувачами — клієнтами та менеджерами. Вона включає в себе всі елементи взаємодії, як-от форми для створення замовлень, перегляд послуг, налаштування профілю користувача. Основною платформою для реалізації цього компонента є веб-додаток на основі фреймворку Streamlit, що дозволяє ефективно реалізувати інтерактивний інтерфейс з мінімальними затратами часу на розробку |
| 2 | Бізнес-логіка         | Бізнес-логіка відповідає за обробку запитів та виконання розрахунків, наприклад, для створення нових послуг або замовлень. Вона складається з функцій, що забезпечують додавання, оновлення та видалення послуг, а також перевірку введених даних на коректність. Це дозволяє гарантовано уникнути помилок у бізнес-процесах                                                                                                            |

Продовження таблиці 3.3

| 1 | 2                  | 3                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3 | База даних         | База даних є важливою складовою архітектури і використовується для зберігання всіх даних, пов'язаних із послугами, замовленнями, користувачами та їх ролями. У даному випадку передбачається використання реляційної бази даних, що дозволяє ефективно зберігати та організовувати інформацію. Дані зберігаються в таблицях, кожна з яких відповідає за певну категорію інформації, таку як послуги, замовлення або профілі користувачів |
| 4 | API для інтеграції | Оскільки в рамках системи передбачається робота з різними сторонніми сервісами (наприклад, для митного оформлення, страхівки або оплати), необхідним є розробка API для інтеграції з іншими зовнішніми сервісами. Це дозволить автоматизувати багато процесів, зокрема, оновлення статусу замовлення або отримання інформації про тарифи                                                                                                 |

«Джерело»: [розроблено автором]

Безпека є однією з головних вимог при виборі архітектури. Система повинна забезпечити надійний захист даних, що зберігаються у базі даних, а також гарантувати захищений доступ до інформації на всіх етапах взаємодії з користувачем. Одним із важливих елементів є аутентифікація та авторизація користувачів, що забезпечує доступ до певних розділів системи залежно від їхніх ролей. У системі передбачено використання токенів для авторизації користувачів, що додає додатковий рівень захисту.

Також важливою частиною є шифрування даних, що передаються між клієнтом і сервером, щоб запобігти можливим атакам через відкриті канали зв'язку. Для цього слід використовувати сучасні стандарти шифрування, зокрема HTTPS для передачі даних між клієнтом та сервером.

Одним з важливих критеріїв при виборі архітектури є її здатність адаптуватися до змінних умов і масштабу підприємства. Транспортні компанії можуть швидко змінювати свої послуги або впроваджувати нові функціональні можливості, тому система повинна бути легко адаптованою до змін. Модульний підхід до розробки дозволяє вносити зміни до окремих частин системи без необхідності її повного переписування. Також важливо забезпечити можливість масштабування, що дозволить обробляти великі обсяги даних в разі зростання підприємства.

Вибір архітектури програмного забезпечення для інформаційної системи управління транспортним підприємством має вирішальне значення для ефективної роботи всієї системи. Веб-сервісна архітектура на основі багатошарової клієнт-серверної моделі є оптимальним рішенням, оскільки вона забезпечує високу гнучкість, масштабованість, зручність для користувачів та можливість інтеграції з іншими сервісами. Використання сучасних стандартів безпеки та можливість адаптації до змінних умов дозволяють системі бути надійною і готовою до майбутніх викликів у процесі розвитку транспортного підприємства.

### **3.3.3 Інструментальні засоби розробки прикладного програмного забезпечення**

Інструментальні засоби розробки програмного забезпечення є важливим компонентом, що визначає ефективність процесу створення інформаційної системи. Вибір таких засобів залежить від вимог, які пред'являються до системи, специфіки бізнес-процесів підприємства, а також від технічних вимог, таких як стабільність, масштабованість, безпека та інтерфейс. Система управління для транспортного підприємства має на меті автоматизувати різноманітні процеси, такі як управління послугами, обробка замовлень, ведення обліку вантажів, формування звітів та аналітики. Усі ці функції потребують відповідного програмного забезпечення, для розробки якого застосовуються різні інструменти, що дозволяють реалізувати необхідні можливості в межах доступних технологій.

Ключовими інструментами для розробки прикладного програмного забезпечення є мови програмування, фреймворки, середовища розробки, бібліотеки, системи контролю версій та бази даних. Вибір кожного з цих елементів залежить від функціональних вимог, що ставляться до системи, а також від рівня команди розробників, їхнього досвіду та можливостей для інтеграції з іншими технологіями.

Для розробки програмного забезпечення для транспортного підприємства використовувалися сучасні мови програмування, зокрема Python, яка є однією з

найпопулярніших мов для створення веб-застосунків та автоматизації різноманітних бізнес-процесів. Python має значний набір бібліотек, що забезпечують швидкий розвиток програмних продуктів. У поєднанні з фреймворком Streamlit Python дозволяє створювати веб-застосунки з інтерактивним інтерфейсом, що є важливим для управлінських панелей, які активно використовуються в інформаційних системах для транспорту.

Streamlit є бібліотекою Python, що спеціалізується на швидкому створенні веб-додатків для аналітики та візуалізації даних. Завдяки своїй простоті і зручному інтерфейсу, Streamlit дозволяє розробникам швидко створювати інтерфейси для взаємодії з користувачами, що є критично важливим для систем, що мають обробляти великі обсяги інформації, такі як замовлення, послуги та дані про вантажі.

Однією з ключових частин програмного забезпечення для транспортного підприємства є база даних, яка повинна ефективно зберігати інформацію про послуги, замовлення, клієнтів, тарифи, документи та інші операції. Для розробки таких систем широко використовуються реляційні бази даних, такі як PostgreSQL або MySQL, оскільки вони дозволяють зберігати і маніпулювати даними з високою ефективністю та забезпечують надійність у зберіганні інформації.

PostgreSQL, зокрема, є однією з найпопулярніших баз даних, що використовується для зберігання великих обсягів даних завдяки своїм високим характеристикам надійності і масштабованості. Вона дозволяє легко створювати таблиці для збереження різних типів даних, а також підтримує складні запити і транзакції, що є важливими для управління замовленнями та послугами в рамках інформаційної системи транспортного підприємства.

Інтеграція з іншими системами є невід'ємною частиною розробки програмного забезпечення для бізнесу. У контексті інформаційної системи для транспортного підприємства важливою є можливість взаємодії з зовнішніми сервісами, такими як платіжні системи, митні органи або постачальники послуг з транспортної логістики. Для цього використовуються спеціалізовані API, які

дозволяють безпосередньо підключатися до цих систем і обробляти дані в реальному часі.

Інтеграція API забезпечує автоматизацію багатьох процесів, наприклад, при обробці замовлень чи митного оформлення вантажів. Таке рішення дозволяє знизити людський фактор, а також зробити взаємодію з клієнтами більш ефективною, автоматизуючи процеси і скорочуючи час на обробку.

Тестування та деплоймент є важливими етапами на шляху до створення якісного програмного продукту. Для забезпечення надійності та ефективності розробленої інформаційної системи використовуються різноманітні інструменти для тестування. Вони допомагають перевірити функціональність, безпеку та сумісність системи з різними платформами.

Для автоматизації тестування у системах управління часто використовуються такі інструменти, як `pytest` та `unittest`. Ці бібліотеки дозволяють ефективно перевіряти коректність роботи кожної складової системи, тестуючи її окремі компоненти та знижуючи ймовірність появи помилок на етапі експлуатації.

Для деплойменту і розгортання використовуються такі інструменти, як `Docker` та `Kubernetes`. Вони дозволяють розробникам швидко налаштовувати та деплоїти застосунки в різних середовищах, забезпечуючи легкість масштабування і управління додатками.

Вибір правильних інструментальних засобів розробки програмного забезпечення є важливим етапом у процесі створення інформаційної системи для транспортного підприємства. Системи, побудовані з використанням таких інструментів, як `Python`, `Streamlit`, `PostgreSQL`, `API`, а також сучасних методів тестування і деплойменту, дозволяють забезпечити ефективну автоматизацію бізнес-процесів і швидке реагування на зміни в умовах конкурентного ринку. В результаті, реалізоване програмне забезпечення здатне підтримувати високий рівень якості послуг, підвищувати ефективність роботи підприємства та знижувати операційні витрати.

### 3.3.4 Тестування програмного забезпечення

Тестування програмного забезпечення є критично важливим етапом у процесі його розробки, що забезпечує перевірку функціональності та надійності програмних рішень. У випадку з розробкою інформаційної системи для транспортного підприємства тестування дозволяє визначити, наскільки ефективно система справляється з поставленими завданнями, а також забезпечує її стабільність і відсутність помилок, які можуть вплинути на роботу користувачів та забезпечення всіх функцій системи. Основним завданням є забезпечення безперебійної роботи програми, яка включає в себе автоматизацію процесів управління послугами, замовленнями та аналітикою для клієнтів і менеджерів.

Тестування програмного забезпечення включає кілька етапів, кожен з яких передбачає перевірку окремих компонентів системи, а також її взаємодії з іншими частинами. З цією метою проводиться покрокове виконання програми, яке забезпечує виявлення потенційних помилок, збоїв або некоректної роботи функціональних блоків. Метою цього етапу є підтвердження відповідності програмного забезпечення вимогам, що ставляться до системи, і забезпечення високої якості кінцевого продукту. Кожен з етапів тестування дозволяє надавати зворотний зв'язок розробникам для подальшої оптимізації та покращення системи. У рамках цього тестування буде розглянуто процес запуску програми, перевірка взаємодії різних частин системи та її стійкість під час виконання основних функцій.

#### Крок 1. Підготовка до тестування

Перед початком тестування необхідно провести налаштування середовища для запуску програмного забезпечення. Це включає в себе встановлення всіх необхідних бібліотек, налаштування інтерфейсу для взаємодії з програмою, а також перевірку працездатності основних модулів системи. На цьому етапі важливо забезпечити правильність налаштувань, адже навіть незначні помилки можуть вплинути на подальше тестування та призвести до некоректних результатів. Зокрема, в даному випадку необхідно переконатися, що всі дані, що стосуються послуг, замовлень та аналітики, зберігаються та обробляються коректно. Кожен

етап тестування має бути розділений на функціональні компоненти, які перевіряються незалежно один від одного.

## Крок 2. Запуск програми

Після налаштування середовища наступним етапом є запуск програми, що дозволяє перевірити її загальну працездатність. На цьому етапі програма повинна запускатися без помилок і з коректним відображенням інтерфейсу. Необхідно перевірити, чи відображаються всі компоненти інтерфейсу, а також чи працюють основні функції: вхід користувача (рис. 3.11), відображення послуг (рис. 3.12), замовлення та управління ними. Важливо, щоб система коректно визначала роль користувача, будь то клієнт чи менеджер, і забезпечувала доступ до відповідних сторінок в залежності від цієї ролі. Крім того, на цьому етапі перевіряється наявність усіх необхідних вкладок, таких як "Послуги", "Замовлення", "Аналітика" для менеджера і так далі.

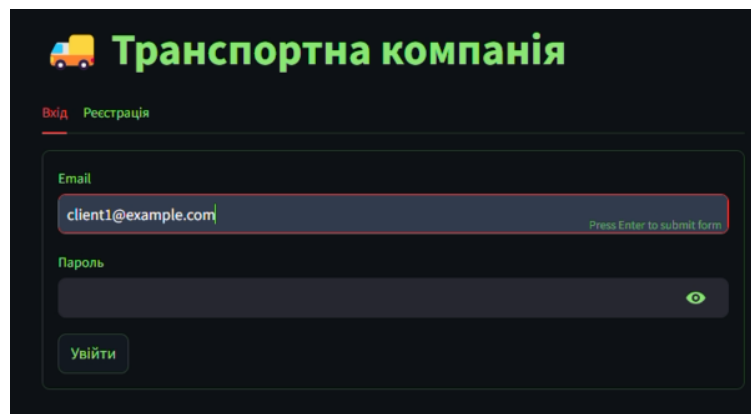


Рисунок 3.11 – Вікно входу користувача

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

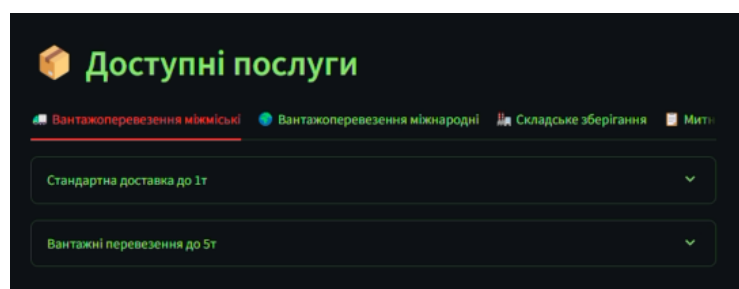


Рисунок 3.12 – Відображення доступних послуг

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

### Крок 3. Авторизація користувача та перевірка ролей

Один з основних елементів роботи програми — це правильне визначення ролі користувача після авторизації (рис. 3.13, рис. 3.14). Система повинна коректно обробляти введення логіну та паролю, а також перевіряти наявність у користувача відповідної ролі. Тестування цього етапу включає перевірку функціональності обробки логінів і паролів, валідацію їх коректності, а також перевірку доступу до сторінок в залежності від ролі. Користувач повинен отримати доступ тільки до тих функцій, які дозволяються його роллю (рис. 3.15, рис. 3.16). Наприклад, клієнт повинен мати доступ лише до створення замовлення і перегляду своїх замовлень, а менеджер — до управління послугами і перегляду аналітики.

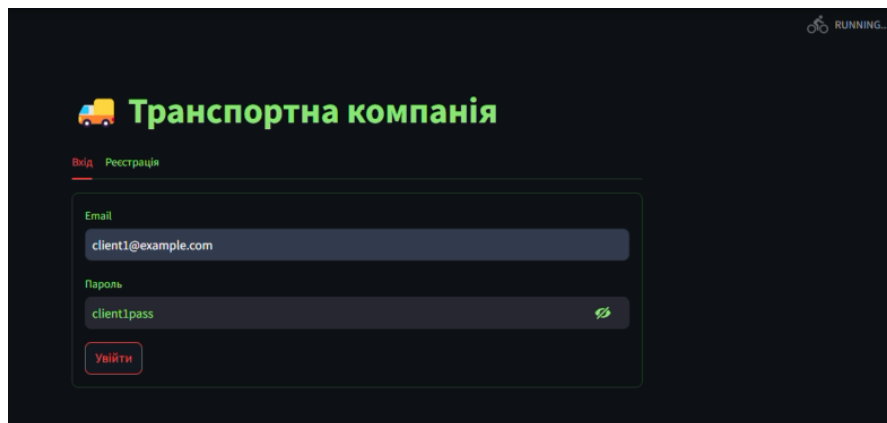


Рисунок 3.13 – Вхід клієнта (введення логіну та паролю)

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

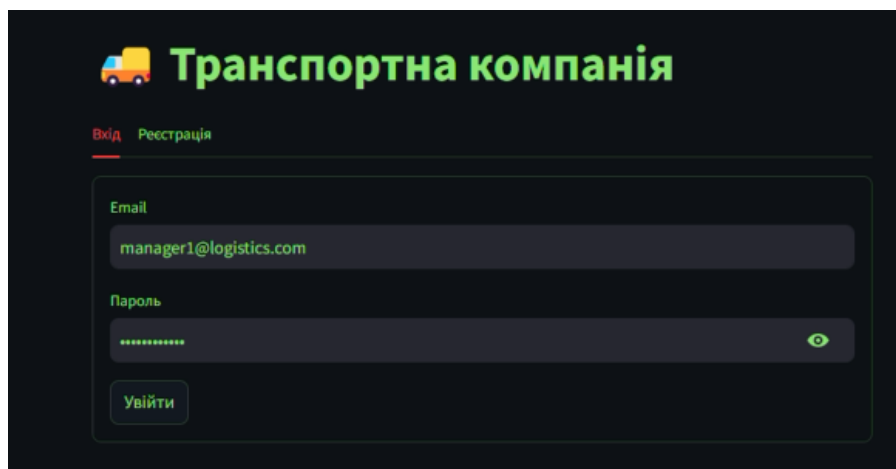


Рисунок 3.14 – Вхід менеджера (введення логіну та паролю)

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

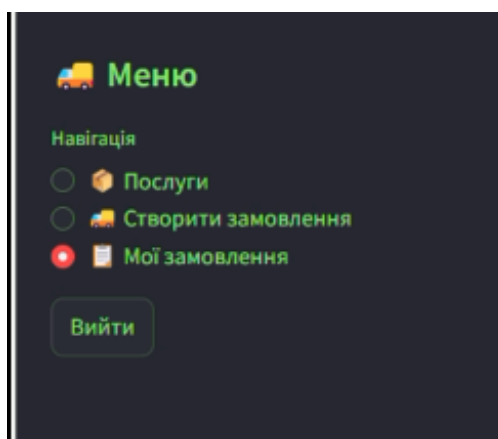


Рисунок 3.15 – Основні вкладки (меню) клієнта

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

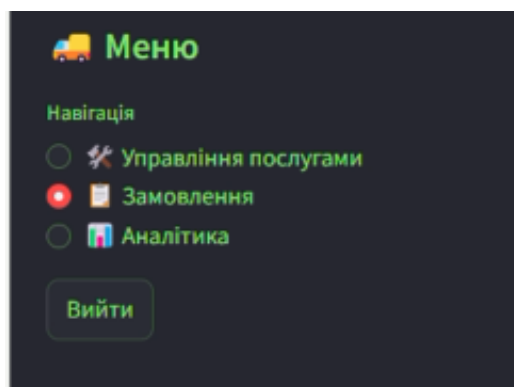


Рисунок 3.16 – Основні вкладки (меню) менеджера

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Крок 4. Управління послугами для менеджера (рис. 3.17, рис.3.18, рис. 3.19, рис. 3.20, рис. 3.21)

Наступний етап тестування передбачає перевірку функціональності управління послугами для менеджера. Це включає створення нових послуг, таких як перевезення, складське зберігання, митне оформлення та страхування вантажів. Тестування повинно перевіряти правильність заповнення форм, перевірку введених даних і коректність збереження даних в системі. Важливим нюансом є перевірка взаємодії різних форм, наприклад, вибір типу транспорту та ціни за кілометр. Крім того, необхідно тестувати функції активації і деактивації послуг, а також їх видалення, щоб переконатися, що ці операції виконуються правильно без порушення даних.

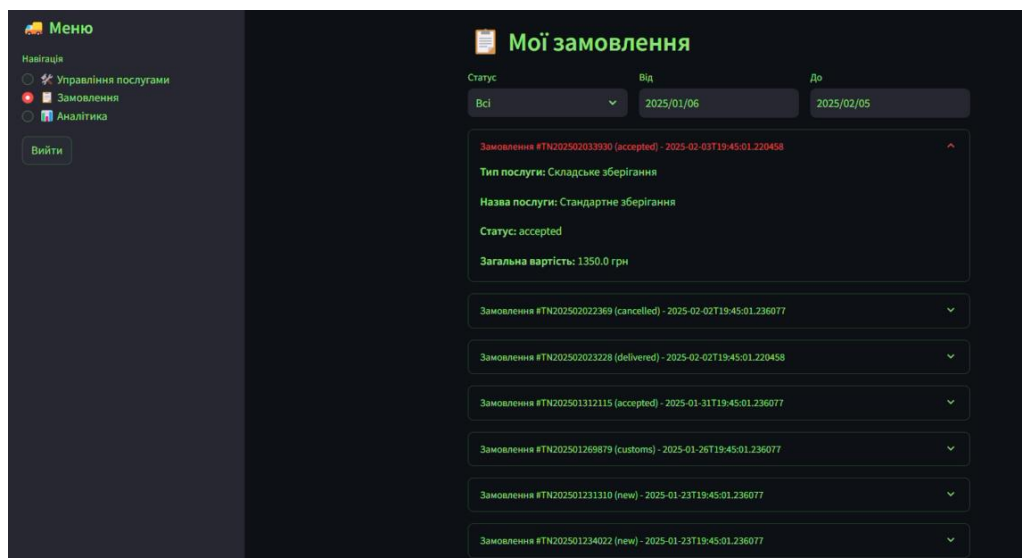


Рисунок 3.17 – Відображення замовлень на сторінці менеджера «Джерело»: [розроблено автором на підставі реалізованого проєкту]

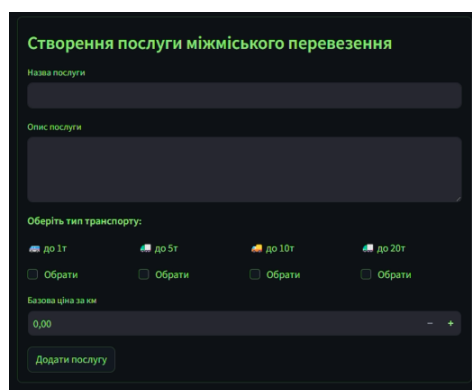


Рисунок 3.18 – Управління замовленнями – створення міжміського перевезення «Джерело»: [розроблено автором на підставі реалізованого проєкту]

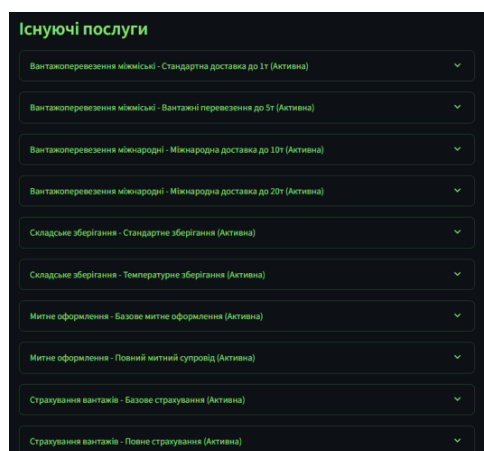


Рисунок 3.19 – Управління замовленнями – вибір існуючої послуги «Джерело»: [розроблено автором на підставі реалізованого проєкту]

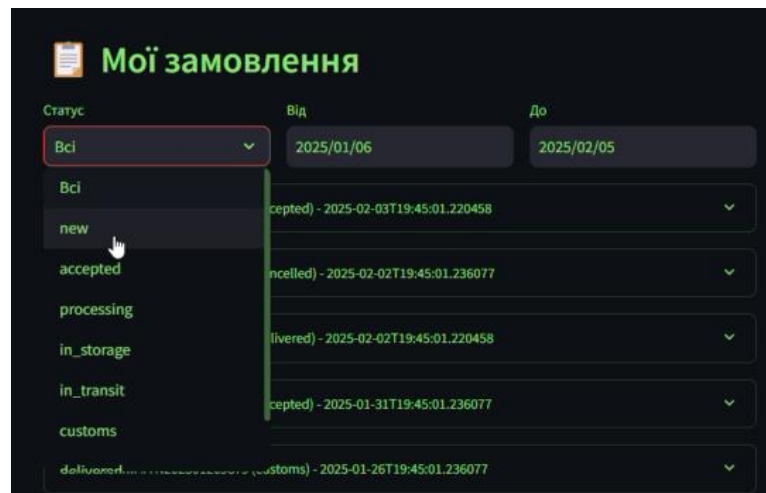


Рисунок 3.20 – Сортуння замовлень

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

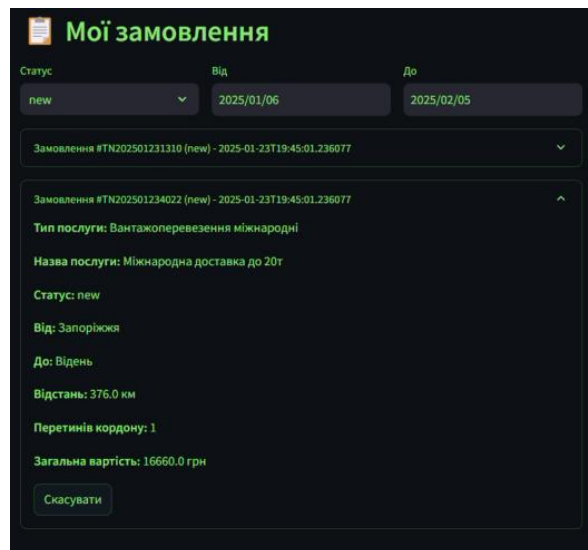


Рисунок 3.21 – Перегляд деталей замовлення

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Крок 5. Перевірка створення замовлень для клієнтів (рис. 3.22, рис. 3.23, рис. 3.24, рис. 3.25, рис. 3.26, рис. 3.27)

Тестування функції створення замовлення є важливою частиною тестування клієнтської частини системи. Клієнт повинен мати можливість вибрати необхідну послугу, вказати дані про вантаж, вибрати тип транспорту і отримати відповідну ціну за перевезення. Важливо перевірити, чи коректно відображаються ці дані в системі, а також чи генерується правильний рахунок. Крім того, слід тестувати, чи

можуть клієнти переглядати свої замовлення, а також чи коректно працює система управління замовленнями, зокрема їх редагування, скасування або завершення.

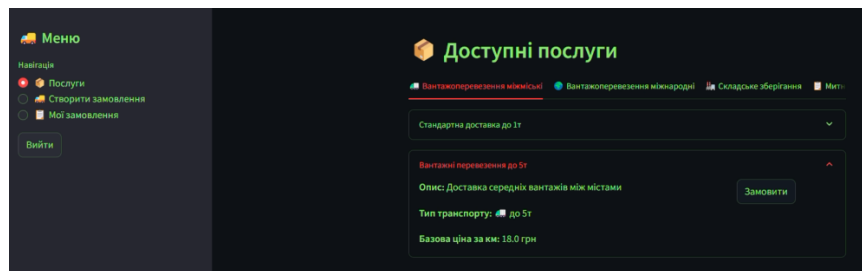


Рисунок 3.22 – Меню послуг для користувача

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Рисунок 3.23 – Форма замовлення послуги для користувача

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Рисунок 3.24 – Заповнена форма замовлення послуги для користувача

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

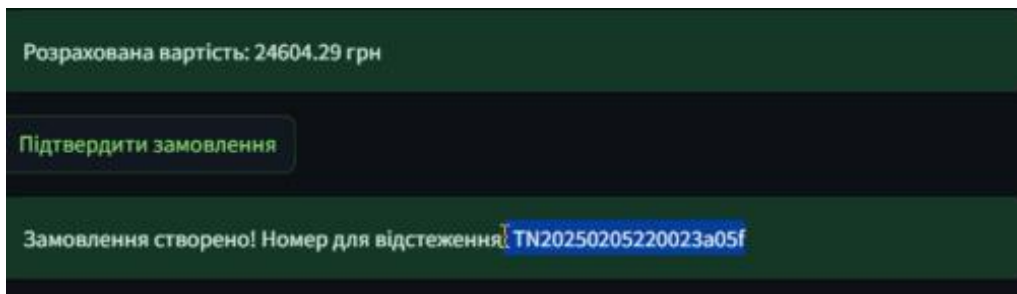


Рисунок 3.25 – Розрахована вартість замовлення та номер для відстеження

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

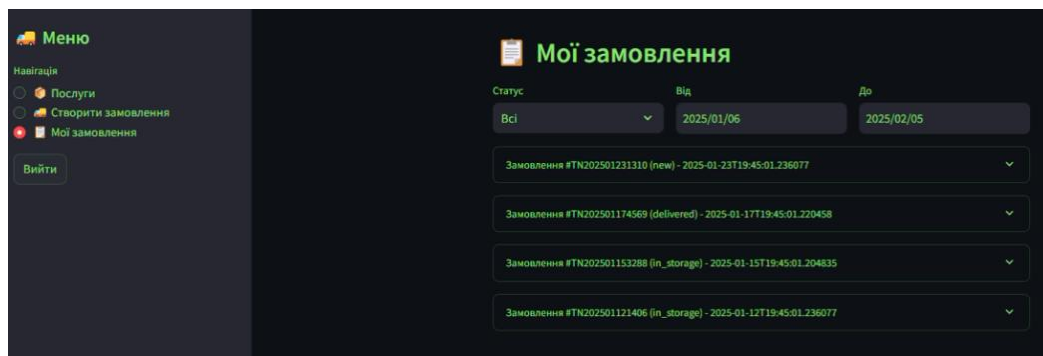


Рисунок 2.26 – Відображення замовлень клієнта

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

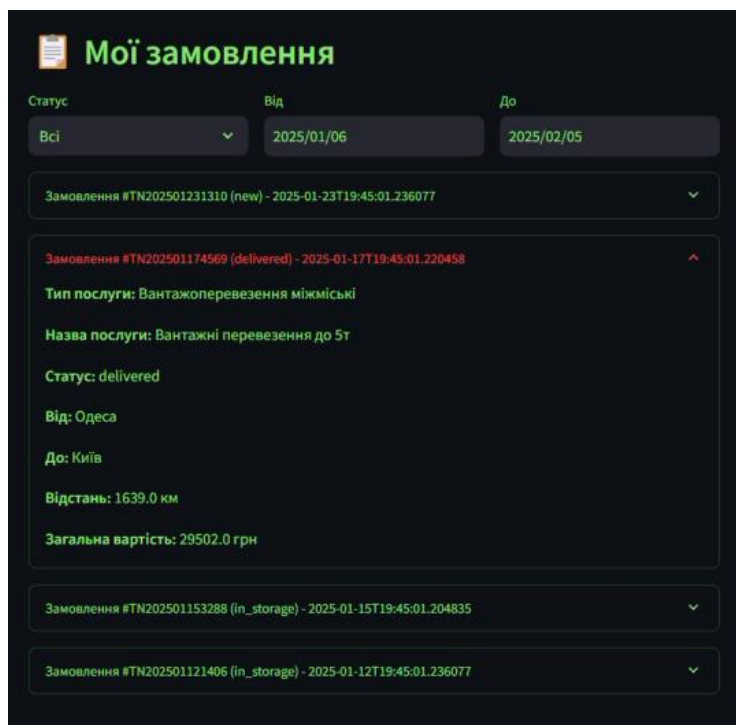


Рисунок 3.27 – Перегляд деталей замовлення

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

Крок 6. Перевірка аналітики та звітів (рис. 3.28, рис. 3.29, рис. 3.30, рис. 3.31, рис. 3.32)

Для менеджера важливо мати можливість переглядати аналітику щодо виконаних замовлень, доходів і витрат. Тестування цього розділу включає перевірку коректності збору і відображення даних. Це може бути графічне відображення даних або таблиці, що містять потрібну інформацію. Важливо переконатися, що система відображає актуальні дані і дозволяє менеджеру виконувати базову аналітику для прийняття рішень.

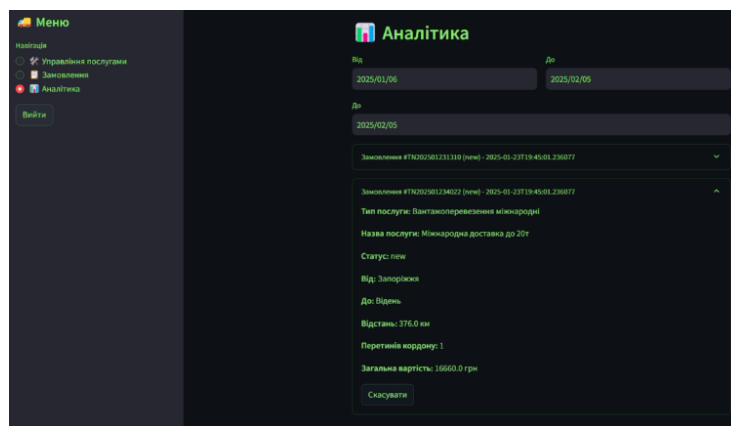


Рисунок 3.28 – Перегляд Аналітики

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

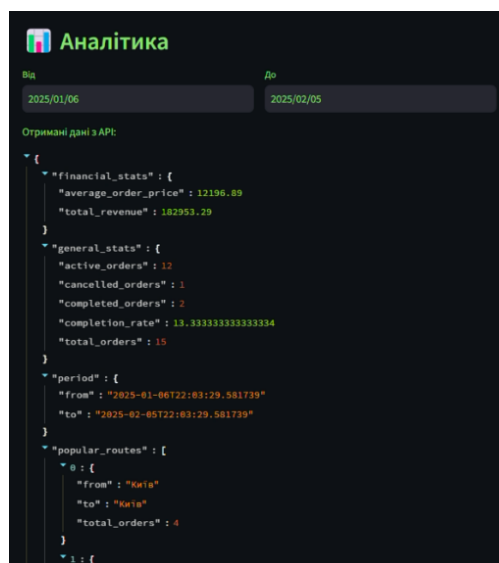


Рисунок 3.29 – Отримані дані з API

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

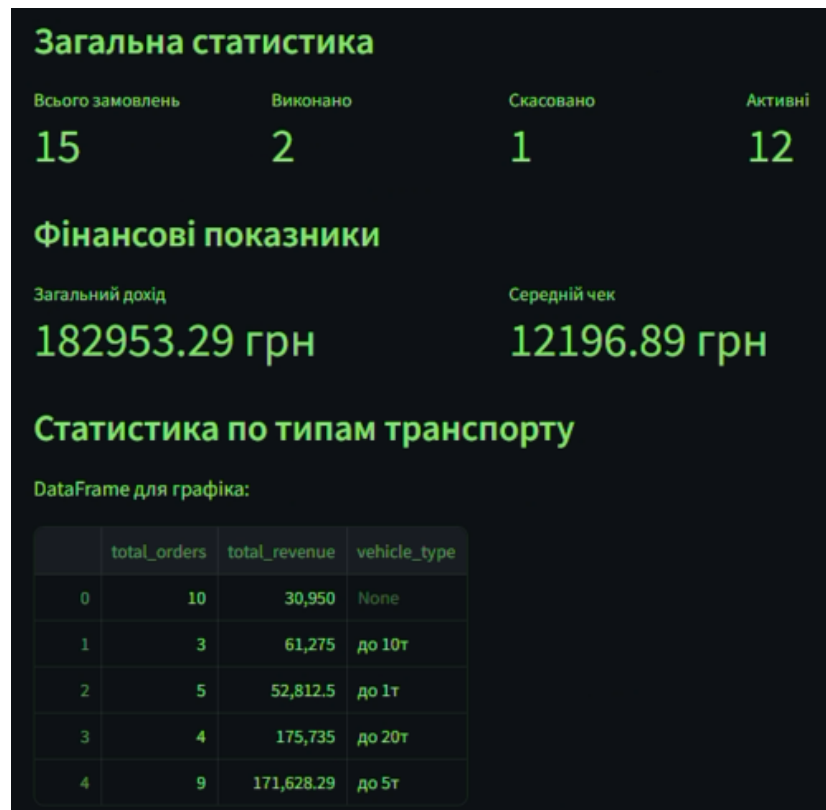


Рисунок 3.30 – Статистичні дані

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

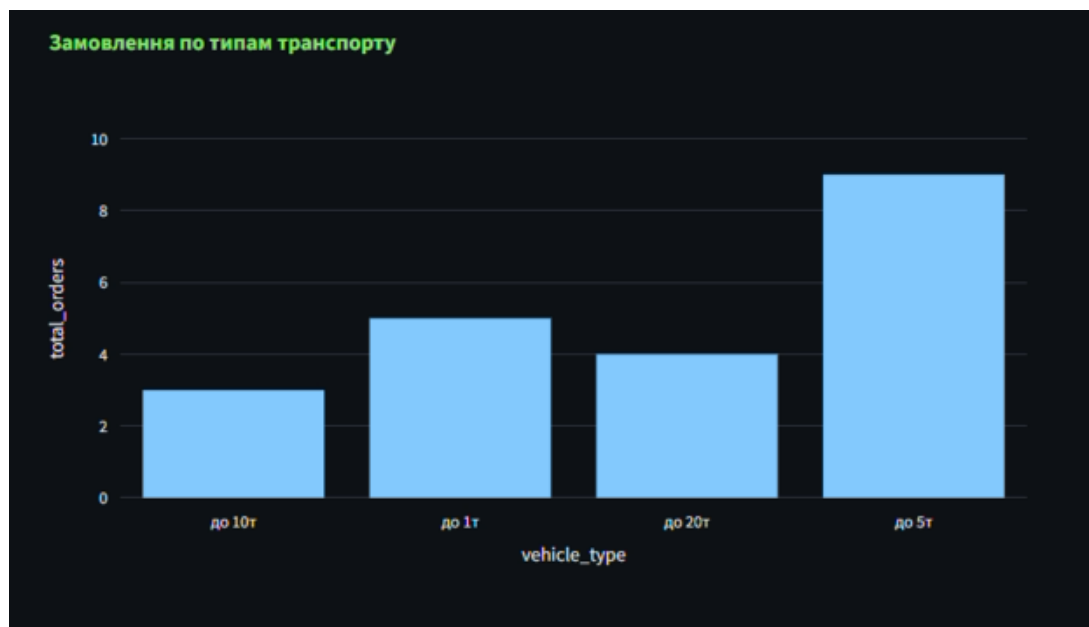


Рисунок 3.31 – Графічне відображення аналітичних даних

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

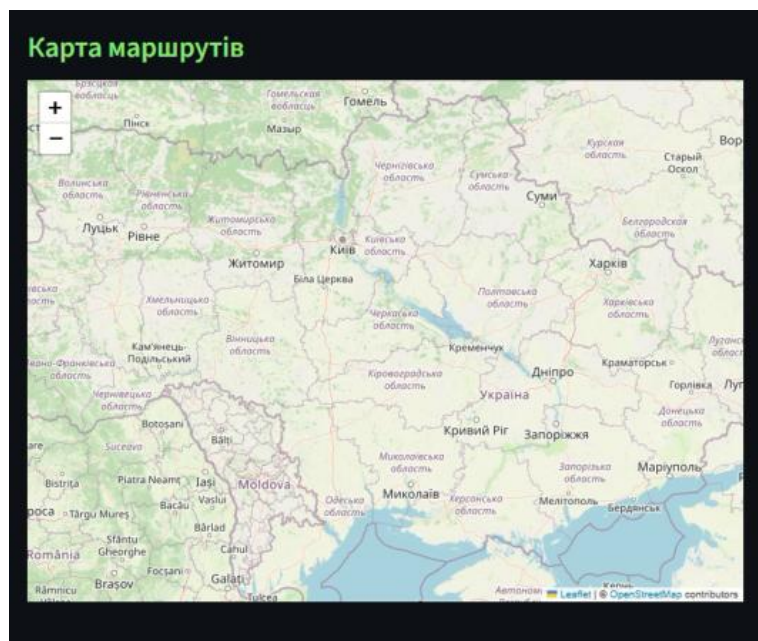


Рисунок 3.31 – Карта маршрутів

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

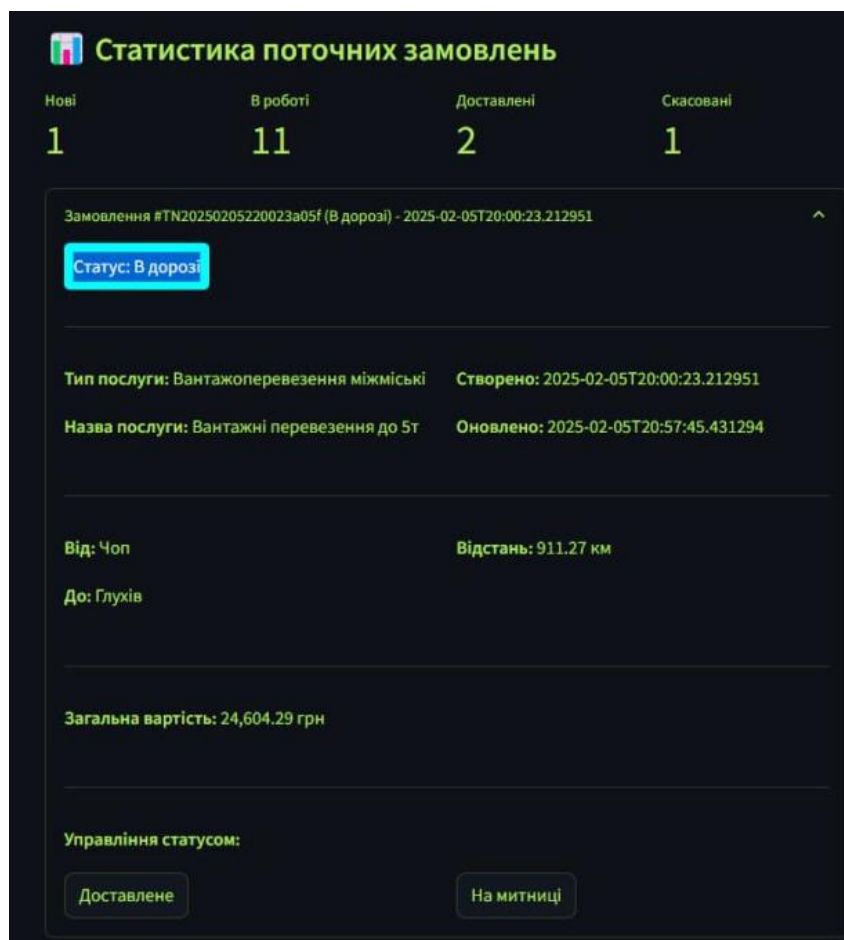


Рисунок 3.32 – Статистика поточних замовлень

«Джерело»: [розроблено автором на підставі реалізованого проєкту]

### Крок 7. Завершення тестування та аналіз результатів

Після завершення тестування кожної з функцій необхідно провести аналіз результатів тестування. Це дозволяє визначити, чи всі функції працюють належним чином, чи є помилки в програмному забезпеченні і на яких етапах вони виникають. Аналіз результатів допомагає виявити будь-які недоліки, які потребують виправлення, а також сприяє покращенню якості програмного забезпечення.

Тестування програмного забезпечення є необхідним етапом у процесі його розробки, який гарантує, що кінцевий продукт відповідає вимогам користувачів і працює без помилок. Покрокове виконання програми дозволяє не тільки виявити можливі дефекти, але й забезпечити стабільність і надійність системи в реальних умовах експлуатації. Тестування, що включає перевірку різних компонентів системи, дозволяє максимально підготувати програму до реального використання, забезпечивши безпеку і ефективність всіх функцій.

### 3.3.5 Керівництво (інструкція) користувача

Розробка програмного забезпечення для управління транспортним підприємством передбачає створення зручного, функціонального інтерфейсу, який дозволяє користувачам здійснювати операції, пов'язані з організацією та управлінням перевезень, замовленнями, обробкою послуг і аналізом даних. Для забезпечення ефективної роботи з програмним продуктом важливо мати чітке та зрозуміле керівництво користувача, яке допоможе в навігації по інтерфейсу та оптимальному використанні всіх можливостей системи.

#### ***1. Авторизація користувача***

Перед початком роботи з програмою кожен користувач повинен авторизуватися в системі. Для цього потрібно виконати наступні дії:

##### ***1.1 Вхід до системи:***

- Користувач відкриває основний інтерфейс системи, де відображається форма для введення даних для входу.
- У форму необхідно ввести логін та пароль.

– Після введення правильних даних користувач натискає кнопку "Вхід", що надає доступ до основних функцій програми.

### *1.2 Роль користувача:*

Після авторизації система автоматично визначає роль користувача. Користувач може бути або менеджером, або клієнтом. В залежності від ролі, доступні різні функціональні можливості.

### *1.3 Вихід із системи:*

Для завершення роботи користувач може натискати кнопку "Вийти", що призведе до виведення із системи. Після цього всі дані сесії будуть очищені.

## **2. Управління послугами**

Модуль управління послугами є однією з ключових частин системи. Він дозволяє менеджерам додавати нові послуги, редагувати існуючі, а також видаляти або змінювати їх статус (активувати чи деактивувати). Для ефективного використання цього модуля необхідно виконати наступні кроки:

### *2.1 Перехід до управління послугами:*

- Для того, щоб увійти на сторінку управління послугами, необхідно у бічному меню вибрати розділ "  Управління послугами".
- Відкриється сторінка з вкладками для різних категорій послуг, таких як "Міжміські перевезення", "Міжнародні перевезення", "Складське зберігання" та інші.

### *2.2 Створення нової послуги:*

- Вибір відповідної вкладки залежно від типу послуги.
- Користувач переходить до форми для створення нової послуги. Для цього потрібно вказати:

- ✓ Назву послуги.
- ✓ Опис послуги.
- ✓ Вибір типу транспорту, що відповідатиме послугі.
- ✓ Вартість послуги, наприклад, базову ціну за кілометр для перевезень або ціну за зберігання для складських послуг.

- Після заповнення всіх полів користувач натискає кнопку "Додати послугу". Якщо форма заповнена коректно, система створює нову послугу в базі даних.

### *2.3 Редагування та активація/деактивація послуг:*

- У списку існуючих послуг відображаються всі активні та неактивні послуги.
- Для редагування послуги необхідно натиснути на її назву. Відкриється форма редагування, в якій можна змінити параметри послуги, такі як ціна, тип транспорту, опис тощо.
- Для зміни статусу послуги (активувати чи деактивувати) користувач натискає на відповідну кнопку біля послуги.

### *2.4 Видалення послуги:*

Якщо послуга більше не потрібна, можна її видалити. Для цього поряд із послугою є кнопка "Видалити". Після натискання кнопки послуга буде безповоротно видалена з системи.

## **3. Створення замовлення**

Користувач може створити замовлення на перевезення або інші послуги, вибравши відповідний розділ у меню. Для цього необхідно виконати наступні кроки:

### *3.1 Перехід до створення замовлення:*

У бічному меню вибрати розділ "  Створити замовлення", що відкриває форму для заповнення даних замовлення.

### *3.2 Заповнення даних замовлення:*

- Вибір послуги, що підходить для замовлення (наприклад, міжміське перевезення).
- Вказати параметри перевезення, такі як відстань, тип вантажу, розміри вантажу, особливі вимоги (якщо є).

– Після заповнення всіх даних користувач натискає кнопку "Створити замовлення", і система обробляє заявку, автоматично підраховуючи вартість замовлення відповідно до встановлених тарифів.

#### ***4. Перегляд та управління замовленнями***

Після створення замовлення користувач може переглядати його статус та деталі:

##### ***4.1 Перегляд замовлення:***

– У розділі "  Мої замовлення" користувач може побачити всі свої замовлення, включаючи поточний статус (активне, виконано, скасовано).

– Для детального перегляду потрібно натискати на конкретне замовлення. Відкриється його детальна інформація, включаючи всі вказані параметри та історію змін.

##### ***4.2 Редагування та скасування замовлення:***

Якщо потрібно внести зміни до замовлення, користувач може редагувати його деталі або скасувати замовлення, натиснувши відповідні кнопки.

#### ***5. Аналітика та звітність***

Модуль аналітики дозволяє користувачам отримувати важливу інформацію щодо виконання замовлень, використання послуг та інших показників діяльності підприємства.

##### ***5.1 Перехід до аналітики:***

У бічному меню менеджер може вибрати розділ "  Аналітика", що надає доступ до різноманітних звітів і статистики.

##### ***5.2 Фільтрація та перегляд даних:***

– Користувач може фільтрувати дані за різними критеріями, такими як період часу, типи послуг, статуси замовлень тощо.

– Результати будуть відображатися у вигляді графіків, таблиць або інших зручних форматів для подальшого аналізу.

Правильне та ефективне використання інформаційної системи управління транспортним підприємством допомагає значно оптимізувати робочі процеси,

зменшити час, витрачений на ручне введення даних, і підвищити загальну ефективність підприємства. Кожен етап, починаючи від авторизації до управління послугами та замовленнями, ретельно продуманий для забезпечення зручності і простоти роботи з системою. Системи такого типу повинні бути інтуїтивно зрозумілими, а також містити чіткі та детальні інструкції для користувачів, що дозволяють максимально ефективно використати доступні функції.

### **3.4 Технічне забезпечення**

Розробка ефективної інформаційної системи управління для транспортного підприємства потребує чіткого визначення технічного забезпечення, яке буде виконувати роль основи для стабільної роботи та високої ефективності таких систем. Технічне забезпечення включає в себе вибір програмного та апаратного забезпечення, а також вимоги до ресурсів, які необхідні для оптимальної роботи всієї інфраструктури. Для цього необхідно врахувати специфіку функціонування транспортних компаній, де обробка великих обсягів інформації, взаємодія з різними видами послуг і забезпечення безперебійної роботи системи є важливими складовими для забезпечення бездоганної роботи організації.

#### **3.4.1 Обґрунтування вибору технічного забезпечення**

Вибір технічного забезпечення для інформаційної системи управління транспортним підприємством залежить від кількох ключових факторів: специфікації функціональних вимог до системи, потреби в обробці та зберіганні даних, масштабування проекту, забезпечення високої доступності і надійності, а також зручність користувацького інтерфейсу.

Основними складовими технічного забезпечення є серверне обладнання, програмне забезпечення та бази даних.

## **1. Серверне обладнання та хостинг**

Для забезпечення ефективної роботи інформаційної системи управління транспортною компанією необхідно вибрати відповідну інфраструктуру серверів, яка підтримуватиме високу доступність і масштабованість. Використання хмарних платформ, таких як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP), дає можливість масштабувати ресурси відповідно до зростання обсягів даних та навантаження на систему. Хмарне рішення також забезпечує високу доступність, швидкий доступ до даних і знижує витрати на фізичне обладнання.

Програмна частина системи повинна базуватися на серверній архітектурі, що підтримує паралельну обробку запитів і дозволяє інтегрувати різні модулі в єдину систему. Вибір хмарної платформи забезпечує надійність і швидке відновлення в разі збоїв, що є критичним для транспортних компаній, де час є важливим фактором.

## **2. Програмне забезпечення**

Програмне забезпечення є важливою складовою інформаційної системи. У даному випадку для розробки системи управління транспортним підприємством вибір ліцензійних або відкритих платформ для розробки веб-застосунків є оптимальним рішенням. Використання таких мов програмування, як Python і бібліотеки для розробки веб-додатків (Streamlit, Django або Flask), дозволяє швидко і ефективно реалізувати інтерфейси користувача для менеджерів і клієнтів, а також інтегрувати бази даних для зберігання інформації про послуги, замовлення і клієнтів.

Для інтеграції з базами даних використовується система управління реляційними базами даних (MySQL, PostgreSQL або SQLite). Реляційні бази даних гарантують збереження структурованої інформації з підтримкою транзакцій, що є важливим для ефективного управління замовленнями, послугами і взаємодією з клієнтами.

### **3. Інтерфейс користувача**

Інтерфейс повинен бути інтуїтивно зрозумілим для користувачів з різним рівнем кваліфікації, тому для його розробки вибрано використання сучасних фреймворків, таких як Streamlit. Цей інструмент дає змогу створювати прості в користуванні, інтерактивні веб-додатки з функціями створення форм, перегляду списків послуг, управління замовленнями тощо.

### **4. Безпека і захист даних**

Вибір технічного забезпечення також включає в себе заходи безпеки для захисту особистої інформації клієнтів і фінансових даних. Для цього необхідно використовувати шифрування даних на всіх етапах взаємодії з системою (передача даних, зберігання, обробка). Система повинна відповідати вимогам GDPR або аналогічним стандартам для захисту персональних даних.

#### **3.4.2 Ресурсні вимоги до технічного забезпечення**

Для забезпечення безперебійної роботи інформаційної системи необхідно визначити вимоги до апаратних і програмних ресурсів. Враховуючи специфіку задач, що постають перед системою, можна виділити кілька основних категорій ресурсів.

##### ***1. Серверні ресурси***

Ресурси для хостингу системи повинні відповідати наступним вимогам:

- Процесор: для ефективної обробки запитів необхідний багатоцільовий процесор з кількома ядрами. Враховуючи можливі пикові навантаження при взаємодії великої кількості користувачів з системою (особливо в умовах багатьох одночасних замовлень), рекомендується використовувати сервери з багатоядерними процесорами, такими як Intel Xeon або AMD EPYC.
- Оперативна пам'ять (RAM): для швидкої обробки запитів і даних система повинна мати достатній обсяг оперативної пам'яті. Мінімальний обсяг RAM для базового сервера складає 16 ГБ, проте для забезпечення стабільної роботи при великій кількості запитів доцільно використовувати 32 ГБ або більше.

– Місце на диску: система має зберігати значні обсяги даних, тому необхідно передбачити достатню кількість місця на серверних дисках. Мінімальний обсяг для базових даних може бути від 100 ГБ, залежно від обсягу даних, що обробляються. Для великих баз даних рекомендується використовувати диски SSD, оскільки вони забезпечують значно швидший доступ до даних.

## ***2. Мережеві ресурси***

Оскільки система повинна бути доступною для користувачів у будь-який час, важливим є забезпечення високої пропускної здатності каналу зв'язку. Для стабільної роботи необхідно мати мережеві з'єднання з пропускною здатністю не менше 1 Гбіт/с, а для більших обсягів трафіку можна застосовувати швидкість 10 Гбіт/с.

## ***3. Ресурси для зберігання даних***

Для зберігання великих обсягів даних (зокрема, історії замовлень, послуг, користувачів) слід використовувати надійні і швидкі сховища даних. Використання розподілених сховищ або хмарних рішень дасть змогу масштабувати систему відповідно до потреб підприємства.

## ***4. Програмне забезпечення для моніторингу та управління***

Для забезпечення стабільної роботи інформаційної системи необхідно інтегрувати інструменти моніторингу та управління серверним обладнанням і програмним забезпеченням. Це дозволить своєчасно виявляти потенційні проблеми в роботі системи і оперативно вживати заходів для їх усунення.

Вибір технічного забезпечення для інформаційної системи управління транспортним підприємством є важливим етапом, що впливає на ефективність і стабільність всієї системи. Правильний вибір серверного обладнання, програмних платформ та інтерфейсів користувача дозволить забезпечити безперебійну роботу системи при високих навантаженнях. Окрім того, важливою складовою є безпека та захист даних, що гарантує конфіденційність і безпеку інформації клієнтів. Ресурсні вимоги до технічного забезпечення забезпечать можливість масштабування системи відповідно до зростання компанії та її потреб.

### 3.5 Реалізація інформаційної управляючої системи

У сучасних умовах функціонування транспортних підприємств значну роль відіграє автоматизація процесів управління. Однією з ключових складових такої автоматизації є впровадження інформаційних систем, здатних оптимізувати операційну діяльність, знижувати витрати, підвищувати ефективність і зручність для всіх учасників. Високоякісна інформаційна система забезпечує швидкий доступ до даних, дозволяє своєчасно реагувати на зміни в умовах бізнесу та підвищує загальну продуктивність компанії.

Однією з основних складових такої системи є розробка функціональних інтерфейсів для різних типів користувачів, зокрема для менеджерів та клієнтів. Створення цих інтерфейсів дозволяє автоматизувати управлінські функції, зокрема, управління послугами, створення замовлень, обробку заявок, а також моніторинг існуючих послуг і аналітику.

Розроблена система складається з кількох основних модулів, кожен з яких відповідає за конкретний функціональний блок підприємства. Ці модулі включають:

- Управління послугами: дозволяє менеджеру створювати, редагувати та видаляти різні послуги, що надаються підприємством.
- Створення та управління замовленнями: дозволяє клієнтам створювати нові замовлення, переглядати свої заявки, а також стежити за їх статусом.
- Аналітика: надає менеджерам необхідні дані для прийняття обґрунтованих управлінських рішень на основі зібраної інформації про послуги та замовлення.

Кожен з цих модулів спрямований на автоматизацію і оптимізацію бізнес-процесів. Розглянемо детальніше реалізацію ключових функцій цієї системи.

Для керівників підприємства передбачено інтерфейс для управління послугами, які надаються клієнтам. Відповідно до ролі користувача (менеджер або клієнт), система адаптує функціонал для виконання відповідних дій.

Менеджери можуть створювати нові послуги за допомогою форм для заповнення інформації, такої як назва послуги, опис, тип транспорту, ціна за кілометр та інші важливі параметри. Зокрема, для кожної послуги можна вибрати тип транспорту (від легкових автомобілів до вантажівок різного тоннажу) за допомогою графічних елементів (іконок), що робить інтерфейс зручним і наочним.

Форма для введення даних включає такі поля:

- Назва послуги.
- Опис послуги.
- Тип транспорту.
- Базова ціна за кілометр.
- Для міжнародних перевезень додатково включена можливість вказати ціну за перетин кордону.
- Для складського зберігання передбачено введення вартості зберігання на день і за кубічний метр.

Після введення необхідних даних, менеджер може надіслати форму, що автоматично зберігає послугу в базі даних, а також виводить її у списку існуючих послуг. Крім того, існує можливість деактивації або активації послуги, що дозволяє швидко змінювати доступність послуги для клієнтів.

Для клієнтів система передбачає простий інтерфейс для створення замовлень. Клієнт має змогу вибрати одну з існуючих послуг, що відповідають його вимогам, та оформити замовлення. При цьому система автоматично розраховує вартість на основі параметрів замовлення (наприклад, відстань для перевезення або тривалість зберігання товару).

Важливим елементом є підтвердження замовлення та отримання статусу замовлення, що дозволяє клієнту стежити за процесом виконання. Система надає можливість переглядати історію замовлень, що дозволяє користувачам зберігати важливу інформацію для подальшого використання.

Модуль аналітики забезпечує менеджерів інформацією для прийняття рішень на основі даних про виконання послуг та їх популярність серед клієнтів.

Завдяки цьому менеджер має можливість переглядати статистику щодо кожної послуги, її ефективності та фінансових показників.

Дані для аналітики збираються автоматично на основі всіх операцій, що проводяться через систему, і виводяться у зручному вигляді для подальшого аналізу. Це включає графіки, таблиці та інші візуальні інструменти, що допомагають менеджерам робити обґрунтовані прогнози та коригувати стратегію розвитку компанії.

Для розробки даної системи використовувалась мова програмування Python з використанням фреймворку Streamlit, що дозволяє швидко створювати інтерактивні веб-додатки. Всі дані зберігаються у базі даних, до якої забезпечено доступ через API. Використання таких бібліотек, як `st.form`, `st.columns` та інших інтерактивних елементів допомагає створити зручний та інтуїтивно зрозумілий інтерфейс для користувачів різних ролей.

Розроблена інформаційна система для управління транспортним підприємством є потужним інструментом для автоматизації операційної діяльності. Вона дозволяє менеджерам і клієнтам взаємодіяти з послугами у реальному часі, забезпечує ефективне управління замовленнями та надає можливість здійснювати аналітичний моніторинг виконання бізнес-процесів. Завдяки використанню сучасних технологій, таких як Python і Streamlit, система відповідає вимогам швидкості, зручності та надійності, що є важливими складовими для успішного функціонування транспортного підприємства в умовах сучасного ринку.

## ВИСНОВКИ

У результаті проведеного дослідження було розроблено концептуальну модель інформаційної системи управління для транспортного підприємства, що ґрунтується на використанні сучасних інформаційних технологій, зокрема машинного навчання та інтелектуального аналізу даних. Досягнуті результати підтверджують, що впровадження такої системи може суттєво покращити ефективність управлінських процесів, знизити витрати та оптимізувати процеси перевезення, а також підвищити гнучкість і адаптивність транспортних підприємств до змінюваних умов експлуатації.

Поставлені завдання, що стосуються дослідження існуючих інформаційних систем у транспортній сфері та аналізу їх недоліків і переваг, були виконані на високому рівні. На основі цього аналізу було обґрунтовано вибір методів і технологій, які є найбільш доцільними для створення ефективної інформаційної системи для транспортних підприємств. Виявлено, що існуючі рішення не забезпечують достатньо високої інтеграції між різними підсистемами, а також не завжди використовують потенціал сучасних технологій для оптимізації перевезень і прогнозування транспортних потоків.

У процесі розробки структури майбутньої інформаційної системи, особлива увага була приділена створенню ефективної бази даних і знань, що забезпечує оперативне управління інформацією та аналіз даних на основі великих масивів. Завдяки застосуванню методів машинного навчання вдалося реалізувати інтелектуальні механізми аналізу, які дозволяють прогнозувати і оптимізувати маршрути перевезень в реальному часі. Цей підхід дозволяє суттєво знизити витрати на логістичні операції, що є одним з головних завдань для транспортних підприємств у сучасних умовах.

Розроблені програмні засоби відповідають вимогам, що постають перед сучасними транспортними підприємствами, і забезпечують зручний інтерфейс для користувачів. Вибір архітектури програмного забезпечення був обґрунтований з урахуванням вимог до гнучкості, масштабованості та надійності системи.

Тестування програмного забезпечення в умовах моделювання транспортних процесів показало його високу ефективність, точність прогнозів і стабільність у роботі з великими обсягами даних.

Таким чином, основним результатом роботи є створення інформаційної системи, яка здатна значно покращити управління транспортним підприємством, знизити витрати, підвищити ефективність логістики та забезпечити оптимізацію маршрутизації. Розроблена система відповідає вимогам сучасного розвитку транспортної галузі та може бути використана як основа для подальшого впровадження в конкретних транспортних підприємствах.

Виконання поставлених завдань показало, що питання інтеграції інтелектуальних технологій у транспортні інформаційні системи має великий потенціал для подальшого розвитку. У процесі дослідження були розглянуті ключові проблеми, зокрема недостатнє використання методів штучного інтелекту для оптимізації маршрутів та прогнозування, а також обмеження в адаптивності існуючих систем до змінних умов. Рішення цих проблем є важливим кроком для створення більш гнучких і ефективних інструментів для управління транспортними підприємствами.

У подальших дослідженнях можна розглянути можливість інтеграції цієї системи з іншими галузями, зокрема з системами управління запасами, фінансами та технічним обслуговуванням. Це дозволить створити більш комплексну і інтегровану систему, здатну забезпечити максимальну ефективність на всіх етапах діяльності транспортного підприємства. Крім того, подальша робота може бути спрямована на вдосконалення інтерфейсу користувача, що дозволить зробити систему ще більш доступною і зручною для кінцевих користувачів.

Таким чином, результати дослідження демонструють значний потенціал сучасних інформаційних технологій для вдосконалення управління транспортними підприємствами, а також визначають напрямки для подальших наукових і практичних розробок у цій сфері.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Назаренко В. М. Управління логістичними процесами на транспортних підприємствах. Київ: Наукова думка, 2021. 200 с.
2. Грінченко О. В. Методи моделювання транспортних потоків: монографія. Харків: ХНУМГ ім. О. М. Бекетова, 2020. 180 с.
3. Di Yurio R. Artificial Intelligence in Transport Management. Cham: Springer, 2019. 250 p. DOI: 10.1007/978-3-030-12345-6.
4. Taiwo Justice Olorunlana. Using Management Information Systems (MIS) to Improve Business Processes. Journal of Business and Management, 2023, № 4, pp. 15–25.
5. Rokicki T., Ziółkowska P. Integrated IT systems in logistics company management. Ekonomika i Organizacja Logistyki, 2020, Vol. 5, No. 1, pp. 55–64. DOI: 10.22367/eiol.2020.5.1.4.
6. Що таке система управління транспортом TMS? [Електронний ресурс]. URL: <https://quantum-int.com/news/shho-take-sistema-upravlinnya-transportom-tms/> (дата звернення: 06.02.2024).
7. Warehouse Management System, WMS [Електронний ресурс]. URL: <https://www.it.ua/knowledge-base/technology-innovation/warehouse-management-system-wms> (дата звернення: 06.02.2024).
8. Геоінформаційні системи (ГІС) [Електронний ресурс]. URL: <http://www.znannya.org/?view=gis> (дата звернення: 06.02.2024).
9. Інтелектуальна транспортна система [Електронний ресурс]. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Інтелектуальна\\_транспортна\\_система](https://uk.wikipedia.org/wiki/Інтелектуальна_транспортна_система) (дата звернення: 06.02.2024).
10. Kadłubek M., Thalassinos E., Domagała J., Grabowska S., Saniuk S. Intelligent Transportation System Applications and Logistics Resources for Logistics Customer Service in Road Freight Transport Enterprises. Smart and Sustainable Cities and Communities, 2022, pp. 123–145. DOI: 10.1007/978-3-030-98745-9\_8.

11. Mwiinga P. Management Information Systems & Computer Software Packages. Real Deal, 2023, No. 2, pp. 45–60.
12. Sierpiński G., Masoumi H., Macioszek E. Challenges and Solutions for Present Transport Systems. Cham: Springer, 2023. 300 p. DOI: 10.1007/978-3-031-23456-7.
13. Waqar A., Alshehri A. H., Alanazi F., Alotaibi S., Almujiabah H. R. Evaluation of challenges to the adoption of intelligent transportation system for urban smart mobility. Research in Transportation Business & Management, 2023, Vol. 49, Article 100789. DOI: 10.1016/j.rtbm.2023.100789.
14. 5 Most Common Challenges in TMS Implementation and How to Overcome Them [Електронний ресурс]. URL: <https://uniqtms.com/5-most-common-challenges-in-tms-implementation-and-how-to-overcome-them/> (дата звернення: 06.02.2024).
15. Top Cyber Security Challenges in the Transport Sector [Електронний ресурс]. URL: <https://www.tmc3.co.uk/insights/cyber-security-challenges-in-the-transport-sector> (дата звернення: 06.02.2024).
16. Ngesa J. Tackling Security and Privacy Challenges in the Realm of Big Data Analytics. World Journal of Advanced Research and Reviews, 2024, Vol. 21, No. 2, pp. 123–135. DOI: 10.30574/wjarr.2024.21.2.0123.
17. Хітров І. Стратегічні напрями розвитку транспортного підприємства в умовах невизначеності. Сучасні технології в машинобудуванні та транспорті, 2024, № 2 (23), с. 268–275. DOI: 10.36910/2312-1505-2024-2-23-268.
18. Stanković M. The Economic Importance of Transportation Sector. Knowledge International Journal, 2021, Vol. 47, No. 1, pp. 143–146.
19. Кашканов В. А., Кашканов А. А., Кужел В. П. Інформаційні системи і технології на автомобільному транспорті: навчальний посібник. Вінниця: ВНТУ, 2020. 220 с.
20. Transport of passengers and goods by transport enterprises [Електронний ресурс]. Statistics Estonia. URL: <https://www.stat.ee/en/find-statistics/methodology-and-quality/esms-metadata/22029> (дата звернення: 06.02.2024).

21. Організація виробничого процесу в просторі і часі [Електронний ресурс]. URL: <http://feb.tsatu.edu.ua/ebook/mn/ov/page6.html> (дата звернення: 06.02.2024).

22. Організація автомобільних вантажних перевезень. Ч. 1. Електронний посібник [Електронний ресурс]. URL: [https://vukladach.pp.ua/MyWeb/manual/transportni\\_texnologii/ORGANIZASIA\\_VAN\\_TAGNUX\\_AVTOMOBILNUX\\_PEREVEZEN/1/1.htm](https://vukladach.pp.ua/MyWeb/manual/transportni_texnologii/ORGANIZASIA_VAN_TAGNUX_AVTOMOBILNUX_PEREVEZEN/1/1.htm) (дата звернення: 06.02.2024).

23. Організація обслуговуючих господарств. Організація транспортного господарства підприємства. Значення, завдання і структура транспортного господарства [Електронний ресурс]. URL: [https://pidru4niki.com/15950210/ekonomika/organizatsiya\\_obsługovuyuchih\\_gospodarstv](https://pidru4niki.com/15950210/ekonomika/organizatsiya_obsługovuyuchih_gospodarstv) (дата звернення: 06.02.2024).

24. Виробнича дільниця [Електронний ресурс]. Вікіпедія. URL: [https://uk.wikipedia.org/wiki/Виробнича\\_дільниця](https://uk.wikipedia.org/wiki/Виробнича_дільниця) (дата звернення: 06.02.2024).

25. Кошій О. В. Розвиток організаційних структур управління логістикою на вітчизняних підприємствах. Економічний форум, 2022, № 1 (1), с. 49–57. DOI: 10.36910/6775-2308-8559-2022-1-49.

26. Організаційне забезпечення системи управління фінансами [Електронний ресурс]. URL: <https://buklib.net/books/22959/> (дата звернення: 06.02.2024).

27. Кадрове діловодство в компанії: що потрібно знати кадровику-початківцю [Електронний ресурс]. URL: <https://vchasno.ua/kadrove-dilovodstvo/> (дата звернення: 06.02.2024).

28. Про затвердження Положення про Відділ інформаційних технологій, охорони праці та матеріально-технічного забезпечення: наказ Міністерства охорони здоров'я України від 19 квітня 2018 р. № 758. URL: <https://zakon.rada.gov.ua/laws/show/z0587-18> (дата звернення: 06.02.2024).

29. Moraglio M., Kopper C. The Organization of Transport: A History of Users, Industry, and Public Policy. London: Routledge, 2015. 280 p. DOI: 10.4324/9781315776521.

30. Технічне обслуговування автомобіля [Електронний ресурс]. URL: <https://car-service.pro/uk/tehnicheskoe-obsluzhivanie-avtomobilya-ua> (дата звернення: 06.02.2024).

31. Складська логістика — що це? [Електронний ресурс]. URL: <https://skladovka.ua/skladaska-logistyka-shho-cze/> (дата звернення: 06.02.2024).

32. Організація фінансового обліку: етапи, документи та залучені фахівці [Електронний ресурс]. URL: <https://finacademy.net/ua/materials/article/organizaciya-finansovogo-obliku> (дата звернення: 06.02.2024).

33. Управління персоналом: особливості та основні етапи процесу [Електронний ресурс]. URL: <https://sunone.com.ua/articles-uk/upravlinnya-personalom-osoblivosti-ta-osnovni-etapi-procesu/> (дата звернення: 06.02.2024).

34. Zhai Y. The Impact of Intelligent Transportation Technology on The Operational Efficiency of Public Transportation Enterprises. *Academic Journal of Science and Technology*, 2024, Vol. 12, No. 3, pp. 134–138. DOI: 10.54097/ajst.v12i3.1234.

35. Maksimenko G. Transportation optimization best practices [Електронний ресурс]. URL: <https://adexin.com/blog/transport-optimization-best-practices/> (дата звернення: 06.02.2024).

36. Sadaf M., Iqbal Z., Javed A. R., Saba I., Krichen M., Majeed S., Raza A. Connected and Automated Vehicles: Infrastructure, Applications, Security, Critical Challenges, and Future Aspects. *Information and Communication Technologies*, 2023, pp. 100–120. DOI: 10.1007/978-3-031-24567-8\_6.

37. Ajiga D., Okeleke P. A., Folorunsho S., Ezeigweneme C. The role of software automation in improving industrial operations and efficiency. *International Journal of Industrial Engineering*, 2024, Vol. 15, No. 1, pp. 45–60.

38. Ebiesuwa S., Gegeleso B., Falana T., Adegbnjo A. Impact of Information Systems on Operational Efficiency: A Comprehensive Analysis. *Indian Journal of Computer Science and Engineering*, 2023, Vol. 14, No. 4, pp. 661–673. DOI: 10.21817/indjcse/2023/v14i4/231404056.

39. Automation is Transforming the Logistics Industry [Електронний ресурс]. URL: <https://www.ioscm.com/blog/automation-is-transforming-the-logistics-industry/> (дата звернення: 06.02.2024).
40. Sied A. A Study on Essential of Effective Transportation System for Supply Chain Efficiency, Cost Reduction and Enhancing Customer Satisfaction. *International Journal of Supply Chain Management*, 2024, Vol. 13, No. 2, pp. 78–90.
41. Oladimeji D., Gupta K., Kose N. A., Gundogan K., Ge L., Liang F. Smart Transportation: An Overview of Technologies and Applications. *Evolution of IoT and IIoT: Opportunities, Challenges, and Applications*, 2023, pp. 150–170. DOI: 10.1201/9781003374039-8.
42. Турченко І. В. Пріоритетні напрями використання інформаційних технологій у сфері функціонування транспортної інфраструктури України. *New Ukrainian Law*, 2025, № 1, с. 50–60.
43. Key technologies for the digitalisation of transport [Електронний ресурс]. European Commission. URL: <https://digital-strategy.ec.europa.eu/en/policies/technologies-digitalisation-transport> (дата звернення: 06.02.2024).
44. Transportation Management System: Benefits, Features, and Main Providers [Електронний ресурс]. URL: <https://www.altexsoft.com/blog/transportation-management-system/> (дата звернення: 06.02.2024).
45. Yigitcanlar T., Downie A. T., Mathews S., Fatima S., MacPherson J., Behara K. N. S., Paz A. Digital technologies of transportation-related communication: Review and the state-of-the-art. *Transportation Research Interdisciplinary Perspectives*, 2024, Vol. 23, Article 100987. DOI: 10.1016/j.trip.2024.100987.
46. Advantages and Disadvantages of IT in Logistics [Електронний ресурс]. URL: <https://www.aeologic.com/blog/advantages-and-disadvantages-of-it-in-logistics/> (дата звернення: 06.02.2024).
47. The Ultimate Guide to Developing a Management Information System [Електронний ресурс]. URL: <https://cygnis.co/blog/developing-management-information-system/> (дата звернення: 06.02.2024).

48. A faster way to build and share data apps URL: <https://streamlit.io/> (дата звертання 06.02.2024)
49. Integration [Електронний ресурс]. TechTarget. URL: <https://www.techtarget.com/searchcustomerexperience/definition/integration> (дата звернення: 06.02.2024).
50. Streamlit documentation [Електронний ресурс]. URL: <https://docs.streamlit.io/> (дата звернення: 06.02.2024).
51. Python 3.13.2 documentation [Електронний ресурс]. URL: <https://docs.python.org/3/> (дата звернення: 06.02.2024).
52. MySQL [Електронний ресурс]. URL: <https://dev.mysql.com/doc/> (дата звернення: 06.02.2024).
53. Керування ідентифікацією і доступом [Електронний ресурс]. URL: <https://pupenasan.github.io/TI40/Лекц/cloudauth.html> (дата звернення: 06.02.2024).
54. Орехова А., Харченко В. Інтеграція систем і захист інформації як напрями удосконалення управління інформаційним середовищем підприємства в умовах діджиталізації. Економіка та суспільство, 2024, № 60, с. 123–130. DOI: 10.32782/2524-0072/2024-60-16.

## ДОДАТОК А

## ЛІСТИНГ ПРОГРАМИ

server.py

```

from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime, timedelta
import jwt
from functools import wraps
import uuid
from geopy.distance import geodesic
from geopy.geocoders import Nominatim
import json
from datetime import datetime, timezone
app = Flask(__name__)
app.config['SECRET_KEY'] = 'ваш_секретний_ключ'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///transport_company.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
# Моделі бази даних
class User(db.Model):
    """Модель користувача"""
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password = db.Column(db.String(255), nullable=False)
    role = db.Column(db.String(20), nullable=False) # 'client' або 'manager'
    created_at = db.Column(db.DateTime, default=datetime.now(timezone.utc))
    orders = db.relationship('Order', backref='client', lazy=True)
class Service(db.Model):
    """Модель послуги"""
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    type = db.Column(db.String(50), nullable=False)
    description = db.Column(db.Text, nullable=True)
    is_active = db.Column(db.Boolean, default=True)
    created_at = db.Column(db.DateTime, default=datetime.now(timezone.utc))
    updated_at = db.Column(db.DateTime, default=datetime.now(timezone.utc),
onupdate=datetime.now(timezone.utc))
    # Поля для перевезень
    vehicle_type = db.Column(db.String(20), nullable=True) # до 1т, до 5т, до 10т, до 20т
    base_price_per_km = db.Column(db.Float, nullable=True)
    border_crossing_price = db.Column(db.Float, nullable=True)
    # Поля для складського зберігання
    storage_price_per_day = db.Column(db.Float, nullable=True) # ціна за день зберігання
    storage_price_per_cubic_meter = db.Column(db.Float, nullable=True) # ціна за м³
    temperature_control = db.Column(db.Boolean, default=False) # наявність температурного
режиму
    # Поля для митного оформлення
    customs_clearance_price = db.Column(db.Float, nullable=True) # базова ціна оформлення
    customs_declaration_price = db.Column(db.Float, nullable=True) # ціна декларації
    customs_additional_services = db.Column(db.Text, nullable=True) # додаткові послуги
    # Поля для страхування
    insurance_rate = db.Column(db.Float, nullable=True) # відсоток від вартості вантажу
    min_insurance_price = db.Column(db.Float, nullable=True) # мінімальна вартість страховки
    coverage_options = db.Column(db.Text, nullable=True) # варіанти покриття
class Order(db.Model):
    """Модель замовлення"""
    id = db.Column(db.Integer, primary_key=True)
    tracking_number = db.Column(db.String(20), unique=True, nullable=False)

```

```

client_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
service_id = db.Column(db.Integer, db.ForeignKey('service.id'), nullable=False)
status = db.Column(db.String(50), nullable=False, default='new') # new, accepted,
processing, in_storage, in_transit, customs, delivered, cancelled
from_address = db.Column(db.String(255), nullable=False)
to_address = db.Column(db.String(255), nullable=False)
from_coordinates = db.Column(db.String(50), nullable=False) # "lat,lng"
to_coordinates = db.Column(db.String(50), nullable=False) # "lat,lng"
distance = db.Column(db.Float, nullable=False) # в кілометрах
border_crossings = db.Column(db.Integer, nullable=False, default=0)
price = db.Column(db.Float, nullable=False)
created_at = db.Column(db.DateTime, default=datetime.utcnow)
updated_at = db.Column(db.DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

# Допоміжні функції
def token_required(f):
    """Декоратор для перевірки JWT токена"""
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.headers.get('Authorization')
        if not token:
            return jsonify({'message': 'Токен відсутній!'}), 401
        try:
            data = jwt.decode(token, app.config['SECRET_KEY'], algorithms=["HS256"])
            current_user = User.query.get(data['user_id'])
        except:
            return jsonify({'message': 'Недійсний токен!'}), 401
        return f(current_user, *args, **kwargs)
    return decorated

def calculate_distance_and_borders(from_coords, to_coords):
    """Розрахунок відстані та кількості перетинів кордону"""
    # Розпарсимо координати
    from_lat, from_lng = map(float, from_coords.split(','))
    to_lat, to_lng = map(float, to_coords.split(','))
    # Розрахуємо відстань
    distance = geodesic((from_lat, from_lng), (to_lat, to_lng)).kilometers
    # Тут має бути логіка визначення кількості перетинів кордону
    # Спрощений варіант - якщо відстань більше 1000 км, вважаємо що є перетин кордону
    border_crossings = 1 if distance > 1000 else 0
    return round(distance, 2), border_crossings

def get_coordinates(address):
    """Отримання координат за адресою"""
    geolocator = Nominatim(user_agent="transport_company")
    location = geolocator.geocode(address)
    if location:
        return f"{location.latitude},{location.longitude}"
    return None

# Маршрути аутентифікації
@app.route('/register', methods=['POST'])
def register():
    """Реєстрація нового користувача"""
    data = request.get_json()
    if User.query.filter_by(email=data['email']).first():
        return jsonify({'message': 'Користувач вже існує!'}), 409
    hashed_password = generate_password_hash(data['password'])
    new_user = User(
        email=data['email'],
        password=hashed_password,
        role=data['role']
    )
    try:
        db.session.add(new_user)
        db.session.commit()
        return jsonify({'message': 'Користувача успішно зареєстровано!'})
    except Exception as e:

```

```

        db.session.rollback()
        return jsonify({'message': f'Помилка при реєстрації: {str(e)}'}), 400
@app.route('/login', methods=['POST'])
def login():
    """Вхід користувача"""
    data = request.get_json()
    user = User.query.filter_by(email=data['email']).first()
    if user and check_password_hash(user.password, data['password']):
        token = jwt.encode(
            {'user_id': user.id, 'exp': datetime.utcnow() + timedelta(hours=24)},
            app.config['SECRET_KEY'],
            algorithm="HS256"
        )
        return jsonify({'token': token, 'role': user.role})
    return jsonify({'message': 'Невірні дані!'}), 401
# Маршрути для послуг
@app.route('/services', methods=['GET'])
def get_services():
    """Отримання списку послуг"""
    show_all = request.args.get('show_all', '').lower() == 'true'
    token = request.headers.get('Authorization')
    query = Service.query
    if not show_all:
        query = query.filter_by(is_active=True)
    elif token:
        try:
            data = jwt.decode(token, app.config['SECRET_KEY'], algorithms=["HS256"])
            current_user = User.query.get(data['user_id'])
            if current_user.role != 'manager':
                query = query.filter_by(is_active=True)
        except:
            query = query.filter_by(is_active=True)
    services = query.order_by(Service.created_at.desc()).all()
    return jsonify([
        # Basic fields
        'id': s.id,
        'name': s.name,
        'type': s.type,
        'description': s.description,
        'is_active': s.is_active,
        'created_at': s.created_at,
        'updated_at': s.updated_at,
        # Transport fields
        'vehicle_type': s.vehicle_type,
        'base_price_per_km': s.base_price_per_km,
        'border_crossing_price': s.border_crossing_price,
        # Storage fields
        'storage_price_per_day': s.storage_price_per_day,
        'storage_price_per_cubic_meter': s.storage_price_per_cubic_meter,
        'temperature_control': s.temperature_control,
        # Customs fields
        'customs_clearance_price': s.customs_clearance_price,
        'customs_declaration_price': s.customs_declaration_price,
        'customs_additional_services': s.customs_additional_services,
        # Insurance fields
        'insurance_rate': s.insurance_rate,
        'min_insurance_price': s.min_insurance_price,
        'coverage_options': s.coverage_options
    ] for s in services])
@app.route('/services', methods=['POST'])
@token_required
def create_service(current_user):
    """Створення нової послуги"""
    if current_user.role != 'manager':
        return jsonify({'message': 'Недостатньо прав!'}), 403

```

```

data = request.get_json()
new_service = Service(
    name=data['name'],
    type=data['type'],
    description=data.get('description', ''),
    base_price_per_km=data['base_price_per_km'],
    vehicle_type=data['vehicle_type'],
    border_crossing_price=data['border_crossing_price']
)
try:
    db.session.add(new_service)
    db.session.commit()
    return jsonify({'message': 'Послугу створено!', 'id': new_service.id})
except Exception as e:
    db.session.rollback()
    return jsonify({'message': f'Помилка при створенні послуги: {str(e)}'}), 400
@app.route('/services/<int:service_id>', methods=['PUT'])
@token_required
def update_service(current_user, service_id):
    """Оновлення послуги"""
    if current_user.role != 'manager':
        return jsonify({'message': 'Недостатньо прав!'}), 403
    service = Service.query.get_or_404(service_id)
    data = request.get_json()
    try:
        for key, value in data.items():
            if hasattr(service, key):
                setattr(service, key, value)
        db.session.commit()
        return jsonify({'message': 'Послугу оновлено!'})
    except Exception as e:
        db.session.rollback()
        return jsonify({'message': f'Помилка при оновленні послуги: {str(e)}'}), 400
@app.route('/services/<int:service_id>', methods=['DELETE'])
@token_required
def delete_service(current_user, service_id):
    """Видалення (деактивація) послуги"""
    if current_user.role != 'manager':
        return jsonify({'message': 'Недостатньо прав!'}), 403
    service = Service.query.get_or_404(service_id)
    service.is_active = False
    try:
        db.session.commit()
        return jsonify({'message': 'Послугу деактивовано!'})
    except Exception as e:
        db.session.rollback()
        return jsonify({'message': f'Помилка при деактивації послуги: {str(e)}'}), 400
# Маршрути для замовлень
@app.route('/orders', methods=['POST'])
@token_required
def create_order(current_user):
    """Створення нового замовлення"""
    data = request.get_json()
    service = Service.query.get_or_404(data['service_id'])
    if not service.is_active:
        return jsonify({'message': 'Послуга не активна!'}), 400
    # Отримання координат
    from_coords = get_coordinates(data['from_address'])
    to_coords = get_coordinates(data['to_address'])
    if not from_coords or not to_coords:
        return jsonify({'message': 'Не вдалося визначити координати!'}), 400
    # Розрахунок відстані та перетинів кордону
    distance, border_crossings = calculate_distance_and_borders(from_coords, to_coords)
    # Розрахунок вартості
    vehicle_coefficients = {'до 1т': 1.0, 'до 5т': 1.5, 'до 10т': 2.0, 'до 20т': 2.5}

```

```

coefficient = vehicle_coefficients.get(service.vehicle_type, 1.0)
price = (service.base_price_per_km * distance * coefficient +
        (border_crossings * service.border_crossing_price))
# Створення замовлення
new_order = Order(
    tracking_number=f"TN{datetime.now().strftime('%Y%m%d%H%M%S')}{uuid.uuid4().hex[:4]}",
    client_id=current_user.id,
    service_id=service.id,
    from_address=data['from_address'],
    to_address=data['to_address'],
    from_coordinates=from_coords,
    to_coordinates=to_coords,
    distance=distance,
    border_crossings=border_crossings,
    price=round(price, 2)
)
try:
    db.session.add(new_order)
    db.session.commit()
    return jsonify({
        'message': 'Замовлення створено!',
        'tracking_number': new_order.tracking_number,
        'price': new_order.price
    })
except Exception as e:
    db.session.rollback()
    return jsonify({'message': f'Помилка при створенні замовлення: {str(e)}'}), 400
@app.route('/orders/<tracking_number>', methods=['GET'])
@token_required
def get_order(current_user, tracking_number):
    """Отримання інформації про замовлення"""
    order = Order.query.filter_by(tracking_number=tracking_number).first_or_404()
    # Перевірка прав доступу
    if current_user.role != 'manager' and order.client_id != current_user.id:
        return jsonify({'message': 'Недостатньо прав!'}), 403
    return jsonify({
        'tracking_number': order.tracking_number,
        'status': order.status,
        'from_address': order.from_address,
        'to_address': order.to_address,
        'distance': order.distance,
        'border_crossings': order.border_crossings,
        'price': order.price,
        'created_at': order.created_at,
        'updated_at': order.updated_at
    })
@app.route('/orders/<tracking_number>/status', methods=['PUT'])
@token_required
def update_order_status(current_user, tracking_number):
    """Оновлення статусу замовлення"""
    if current_user.role != 'manager':
        return jsonify({'message': 'Недостатньо прав!'}), 403
    order = Order.query.filter_by(tracking_number=tracking_number).first_or_404()
    data = request.get_json()
    # Перевірка валідності статусу
    valid_statuses = ['new', 'accepted', 'processing', 'in_storage',
                     'in_transit', 'customs', 'delivered', 'cancelled']
    if data['status'] not in valid_statuses:
        return jsonify({'message': 'Невірний статус!'}), 400
    # Перевірка можливості скасування
    if data['status'] == 'cancelled' and order.status != 'new':
        return jsonify({'message': 'Замовлення можна скасувати тільки в статусі "new"!'}), 400
    try:
        order.status = data['status']

```

```

        db.session.commit()
        return jsonify({'message': 'Статус замовлення оновлено!'})
    except Exception as e:
        db.session.rollback()
        return jsonify({'message': f'Помилка при оновленні статусу: {str(e)}'}), 400
@app.route('/orders/cancel/<tracking_number>', methods=['POST'])
@token_required
def cancel_order(current_user, tracking_number):
    """Скасування замовлення клієнтом"""
    order = Order.query.filter_by(tracking_number=tracking_number).first_or_404()
    # Перевірка прав доступу
    if order.client_id != current_user.id:
        return jsonify({'message': 'Недостатньо прав!'}), 403
    # Перевірка можливості скасування
    if order.status != 'new':
        return jsonify({'message': 'Замовлення можна скасувати тільки в статусі "new"!'}), 400
    try:
        order.status = 'cancelled'
        db.session.commit()
        return jsonify({'message': 'Замовлення скасовано!'})
    except Exception as e:
        db.session.rollback()
        return jsonify({'message': f'Помилка при скасуванні замовлення: {str(e)}'}), 400
@app.route('/orders', methods=['GET'])
@token_required
def get_orders(current_user):
    """Отримання списку замовлень"""
    # Базовий запит з join до таблиці послуг
    query = Order.query.join(Service, Order.service_id == Service.id)
    # Фільтрація за користувачем для клієнтів
    if current_user.role != 'manager':
        query = query.filter(Order.client_id == current_user.id)
    # Фільтрація за статусом
    status = request.args.get('status')
    if status:
        query = query.filter(Order.status == status)
    # Фільтрація за датою
    date_from = request.args.get('date_from')
    date_to = request.args.get('date_to')
    if date_from:
        query = query.filter(Order.created_at >= datetime.fromisoformat(date_from))
    if date_to:
        query = query.filter(Order.created_at <= datetime.fromisoformat(date_to))
    # Сортування
    sort_by = request.args.get('sort_by', 'created_at')
    sort_order = request.args.get('sort_order', 'desc')
    if hasattr(Order, sort_by):
        sort_column = getattr(Order, sort_by)
        if sort_order == 'desc':
            query = query.order_by(sort_column.desc())
        else:
            query = query.order_by(sort_column.asc())
    orders = query.all()
    return jsonify([
        {
            'tracking_number': o.tracking_number,
            'service_type': Service.query.get(o.service_id).type, # Додаємо тип послуги
            'service_name': Service.query.get(o.service_id).name, # Додаємо назву послуги
            'status': o.status,
            'from_address': o.from_address,
            'to_address': o.to_address,
            'distance': o.distance,
            'border_crossings': o.border_crossings,
            'price': o.price,
            'cargo_type': getattr(o, 'cargo_type', None),
            'cargo_value': getattr(o, 'cargo_value', None),
        }
    ])

```

```

        'cargo_volume': getattr(o, 'cargo_volume', None),
        'storage_days': getattr(o, 'storage_days', None),
        'created_at': o.created_at.isoformat() if o.created_at else None,
        'updated_at': o.updated_at.isoformat() if o.updated_at else None
    } for o in orders])
# Маршрути для аналітики
@app.route('/analytics/orders', methods=['GET'])
@token_required
def get_orders_analytics(current_user):
    """Отримання аналітики по замовленням"""
    if current_user.role != 'manager':
        return jsonify({'message': 'Недостатньо прав!'}), 403
    # Отримання параметрів періоду
    date_from = request.args.get('date_from',
                                  (datetime.now() - timedelta(days=30)).isoformat())
    date_to = request.args.get('date_to', datetime.now().isoformat())
    # Базовий запит
    query = Order.query.filter(
        Order.created_at.between(date_from, date_to)
    )
    # Загальна статистика
    total_orders = query.count()
    completed_orders = query.filter_by(status='delivered').count()
    cancelled_orders = query.filter_by(status='cancelled').count()
    active_orders = total_orders - completed_orders - cancelled_orders
    # Фінансова статистика
    total_revenue = sum(order.price for order in query.all())
    avg_order_price = total_revenue / total_orders if total_orders > 0 else 0
    # Статистика по типам транспорту
    transport_stats = db.session.query(
        Service.vehicle_type,
        db.func.count(Order.id).label('total_orders'),
        db.func.sum(Order.price).label('total_revenue')
    ).join(Service).group_by(Service.vehicle_type).all()
    # Популярні напрямки
    popular_routes = db.session.query(
        Order.from_address,
        Order.to_address,
        db.func.count(Order.id).label('total_orders')
    ).group_by(Order.from_address, Order.to_address)\
    .order_by(db.func.count(Order.id).desc())\
    .limit(10).all()
    return jsonify({
        'period': {
            'from': date_from,
            'to': date_to
        },
        'general_stats': {
            'total_orders': total_orders,
            'completed_orders': completed_orders,
            'cancelled_orders': cancelled_orders,
            'active_orders': active_orders,
            'completion_rate': (completed_orders / total_orders * 100) if total_orders > 0
        },
        'financial_stats': {
            'total_revenue': round(total_revenue, 2),
            'average_order_price': round(avg_order_price, 2)
        },
        'transport_stats': [{
            'vehicle_type': t[0],
            'total_orders': t[1],
            'total_revenue': round(t[2], 2) if t[2] else 0
        } for t in transport_stats],
        'popular_routes': [{

```

```

        'from': r[0],
        'to': r[1],
        'total_orders': r[2]
    } for r in popular_routes]
    })
@app.route('/price-calculator', methods=['POST'])
def calculate_delivery_price():
    """Калькулятор вартості доставки"""
    data = request.get_json()
    service = Service.query.get_or_404(data['service_id'])
    # Отримання координат
    from_coords = get_coordinates(data['from_address'])
    to_coords = get_coordinates(data['to_address'])
    if not from_coords or not to_coords:
        return jsonify({'message': 'Не вдалося визначити координати!'}), 400
    # Розрахунок відстані та перетинів кордону
    distance, border_crossings = calculate_distance_and_borders(from_coords, to_coords)
    # Розрахунок вартості
    vehicle_coefficients = {'до 1т': 1.0, 'до 5т': 1.5, 'до 10т': 2.0, 'до 20т': 2.5}
    coefficient = vehicle_coefficients.get(service.vehicle_type, 1.0)
    price = (service.base_price_per_km * distance * coefficient +
             (border_crossings * service.border_crossing_price))
    return jsonify({
        'from_address': data['from_address'],
        'to_address': data['to_address'],
        'distance': distance,
        'border_crossings': border_crossings,
        'base_price_per_km': service.base_price_per_km,
        'vehicle_type': service.vehicle_type,
        'vehicle_coefficient': coefficient,
        'border_crossing_price': service.border_crossing_price,
        'total_price': round(price, 2)
    })
if __name__ == '__main__':
    with app.app_context():
        db.create_all()
    app.run(debug=True)

```

## seed\_db.py

```

from server import app, db, User, Service, Order
from werkzeug.security import generate_password_hash
from datetime import datetime, timedelta, timezone
import random
def seed_database():
    # Очистка бази даних
    with app.app_context():
        db.drop_all()
        db.create_all()
    # Створення менеджерів
    managers = [
        {
            'email': 'manager1@logistics.com',
            'password': 'manager1pass',
            'role': 'manager'
        },
        {
            'email': 'manager2@logistics.com',
            'password': 'manager2pass',
            'role': 'manager'
        }
    ]

```

```

]
# Створення клієнтів
clients = [
    {
        'email': 'client1@example.com',
        'password': 'client1pass',
        'role': 'client'
    },
    {
        'email': 'client2@example.com',
        'password': 'client2pass',
        'role': 'client'
    },
    {
        'email': 'client3@example.com',
        'password': 'client3pass',
        'role': 'client'
    },
    {
        'email': 'client4@example.com',
        'password': 'client4pass',
        'role': 'client'
    },
    {
        'email': 'client5@example.com',
        'password': 'client5pass',
        'role': 'client'
    }
]
# Додавання користувачів до БД
for manager in managers:
    user = User(
        email=manager['email'],
        password=generate_password_hash(manager['password']),
        role=manager['role']
    )
    db.session.add(user)
for client in clients:
    user = User(
        email=client['email'],
        password=generate_password_hash(client['password']),
        role=client['role']
    )
    db.session.add(user)
db.session.commit()
# Створення послуг
services = [
    # Міжміські перевезення
    {
        'name': 'Стандартна доставка до 1т',
        'type': 'Вантажоперевезення міжміські',
        'description': 'Доставка малогабаритних вантажів між містами',
        'vehicle_type': 'до 1т',
        'base_price_per_km': 12.5,
        'border_crossing_price': 0
    },
    {
        'name': 'Вантажні перевезення до 5т',
        'type': 'Вантажоперевезення міжміські',
        'description': 'Доставка середніх вантажів між містами',
        'vehicle_type': 'до 5т',
        'base_price_per_km': 18.0,
        'border_crossing_price': 0
    },
    # Міжнародні перевезення

```

```

{
  'name': 'Міжнародна доставка до 10т',
  'type': 'Вантажоперевезення міжнародні',
  'description': 'Міжнародні перевезення великих вантажів',
  'vehicle_type': 'до 10т',
  'base_price_per_km': 25.0,
  'border_crossing_price': 2500
},
{
  'name': 'Міжнародна доставка до 20т',
  'type': 'Вантажоперевезення міжнародні',
  'description': 'Міжнародні перевезення великогабаритних вантажів',
  'vehicle_type': 'до 20т',
  'base_price_per_km': 35.0,
  'border_crossing_price': 3500
},
# Складське зберігання
{
  'name': 'Стандартне зберігання',
  'type': 'Складське зберігання',
  'description': 'Зберігання вантажів у стандартних умовах',
  'storage_price_per_day': 150,
  'storage_price_per_cubic_meter': 50,
  'temperature_control': False
},
{
  'name': 'Температурне зберігання',
  'type': 'Складське зберігання',
  'description': 'Зберігання вантажів з контролем температури',
  'storage_price_per_day': 250,
  'storage_price_per_cubic_meter': 80,
  'temperature_control': True
},
# Митне оформлення
{
  'name': 'Базове митне оформлення',
  'type': 'Митне оформлення',
  'description': 'Стандартне оформлення документів',
  'customs_clearance_price': 5000,
  'customs_declaration_price': 1500,
  'customs_additional_services': 'Консультація\nПідготовка документів'
},
{
  'name': 'Повний митний супровід',
  'type': 'Митне оформлення',
  'description': 'Повний пакет митних послуг',
  'customs_clearance_price': 8000,
  'customs_declaration_price': 2000,
  'customs_additional_services': 'Консультація\nПідготовка документів\nМитний
брокер\nСертифікація'
},
# Страхування
{
  'name': 'Базове страхування',
  'type': 'Страхування вантажів',
  'description': 'Базове покриття ризиків',
  'insurance_rate': 0.5,
  'min_insurance_price': 1000,
  'coverage_options': 'Пошкодження\nВтрата\nКрадіжка'
},
{
  'name': 'Повне страхування',
  'type': 'Страхування вантажів',
  'description': 'Максимальне покриття ризиків',
  'insurance_rate': 1.2,

```

```

        'min_insurance_price': 2500,
        'coverage_options': 'Пошкодження\nВтрата\nКрадіжка\nФорс-мажор\nЗатримка'
    }
]

# Додавання послуг до БД
for service_data in services:
    service = Service(**service_data)
    db.session.add(service)
db.session.commit()

# Створення замовлень
# Отримуємо ID користувачів та послуг
client_ids = [user.id for user in User.query.filter_by(role='client').all()]
service_ids = [service.id for service in Service.query.all()]
# Статуси для замовлень
statuses = ['new', 'accepted', 'processing', 'in_storage',
            'in_transit', 'customs', 'delivered', 'cancelled']
# Адреси для замовлень
cities_ua = ['Київ', 'Львів', 'Харків', 'Одеса', 'Дніпро', 'Запоріжжя']
cities_eu = ['Варшава', 'Берлін', 'Прага', 'Відень', 'Будапешт']
# Створення 30 замовлень
for _ in range(30):
    # Вибір випадкового клієнта
    client_id = random.choice(client_ids)
    # Вибір послуги з нерівномірним розподілом
    # Робимо деякі послуги більш популярними
    weights = [3, 3, 2, 2, 2, 1, 1, 1, 1, 1] # Ваги для кожної послуги
    service_id = random.choices(service_ids, weights=weights, k=1)[0]
    service = Service.query.get(service_id)
    # Генерація даних замовлення в залежності від типу послуги
    if "Вантажоперевезення" in service.type:
        if "міжнародні" in service.type.lower():
            from_address = random.choice(cities_ua)
            to_address = random.choice(cities_eu)
            border_crossings = 1
        else:
            from_address = random.choice(cities_ua)
            to_address = random.choice(cities_ua)
            while to_address == from_address:
                to_address = random.choice(cities_ua)
            border_crossings = 0
        distance = random.randint(300, 2000)
        price = service.base_price_per_km * distance + (border_crossings *
service.border_crossing_price)
    elif service.type == "Складське зберігання":
        from_address = random.choice(cities_ua)
        to_address = from_address
        distance = 0
        price = service.storage_price_per_day * random.randint(5, 30)
    else: # Митне оформлення та страхування
        from_address = "Офіс компанії"
        to_address = "Офіс компанії"
        distance = 0
        if service.type == "Митне оформлення":
            price = service.customs_clearance_price +
service.customs_declaration_price
        else:
            cargo_value = random.randint(50000, 500000)
            price = max(cargo_value * service.insurance_rate / 100,
service.min_insurance_price)
    # Створення замовлення
    order_date = datetime.now(timezone.utc) - timedelta(days=random.randint(0, 60))
    order = Order(
        tracking_number=f"TN{order_date.strftime('%Y%m%d')}{random.randint(1000,
9999)}",

```

```

        client_id=client_id,
        service_id=service_id,
        status=random.choice(statuses),
        from_address=from_address,
        to_address=to_address,
        from_coordinates="0,0", # Спрощено для прикладу
        to_coordinates="0,0",   # Спрощено для прикладу
        distance=distance,
        border_crossings=border_crossings if "Вантажоперевезення" in service.type else
0,

        price=price,
        created_at=order_date,
        updated_at=order_date + timedelta(days=random.randint(1, 5))
    )
    db.session.add(order)
    db.session.commit()
    print("База даних успішно заповнена тестовими даними!")
if __name__ == "__main__":
    seed_database()

```

## client.py

```

import streamlit as st
import requests
import json
from datetime import datetime, timedelta
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from PIL import Image
import folium
from streamlit_folium import folium_static
# Налаштування
BASE_URL = 'http://localhost:5000'
# Іконки для типів транспорту
VEHICLE_ICONS = {
    'до 1т': '🚚',
    'до 5т': '🚛',
    'до 10т': '🚚',
    'до 20т': '🚛'
}
# Кольори для статусів замовлень
STATUS_COLORS = {
    'new': 'blue',
    'accepted': 'orange',
    'processing': 'yellow',
    'in_storage': 'purple',
    'in_transit': 'cyan',
    'customs': 'brown',
    'delivered': 'green',
    'cancelled': 'red'
}
# Функції для роботи з API
def api_request(endpoint, method='GET', data=None, token=None):
    """Універсальна функція для запитів до API"""
    headers = {'Authorization': token} if token else {}
    url = f"{BASE_URL}/{endpoint}"
    try:
        if method == 'GET':
            response = requests.get(url, headers=headers)
        elif method == 'POST':

```

```

        response = requests.post(url, json=data, headers=headers)
    elif method == 'PUT':
        response = requests.put(url, json=data, headers=headers)
    elif method == 'DELETE':
        response = requests.delete(url, headers=headers)
    response.raise_for_status()
    return response.json()
except requests.exceptions.RequestException as e:
    st.error(f"Помилка API: {str(e)}")
    return None

def login(email, password):
    """Функція для входу"""
    return api_request('login', 'POST', {'email': email, 'password': password})

def register(email, password, role):
    """Функція для реєстрації"""
    return api_request('register', 'POST', {
        'email': email,
        'password': password,
        'role': role
    })

def get_services(token, show_all=False):
    """Отримання списку послуг"""
    return api_request(f'services{"?show_all=true" if show_all else ""}', token=token)

def create_service(token, data):
    """Створення нової послуги"""
    return api_request('services', 'POST', data, token)

def update_service(token, service_id, data):
    """Оновлення послуги"""
    return api_request(f'services/{service_id}', 'PUT', data, token)

def delete_service(token, service_id):
    """Видалення послуги"""
    return api_request(f'services/{service_id}', 'DELETE', token=token)

def calculate_price(data):
    """Розрахунок вартості доставки"""
    return api_request('price-calculator', 'POST', data)

def create_order(token, data):
    """Створення замовлення"""
    return api_request('orders', 'POST', data, token)

def get_orders(token, **filters):
    """Отримання списку замовлень"""
    query_params = '&'.join(f'{k}={v}' for k, v in filters.items() if v)
    return api_request(f'orders?{query_params}', token=token)

def get_order_details(token, tracking_number):
    """Отримання деталей замовлення"""
    return api_request(f'orders/{tracking_number}', token=token)

def update_order_status(token, tracking_number, status):
    """Оновлення статусу замовлення"""
    return api_request(f'orders/{tracking_number}/status', 'PUT', {'status': status}, token)

def cancel_order(token, tracking_number):
    """Скасування замовлення"""
    return api_request(f'orders/cancel/{tracking_number}', 'POST', token=token)

def get_analytics(token, date_from=None, date_to=None):
    """Отримання аналітики"""
    query_params = {}
    if date_from:
        query_params['date_from'] = date_from
    if date_to:
        query_params['date_to'] = date_to
    return api_request('analytics/orders', token=token, data=query_params)

# Функції інтерфейсу
def show_login_page():
    """Сторінка входу"""
    st.title("🚚 Транспортна компанія")
    tab1, tab2 = st.tabs(["Вхід", "Реєстрація"])

```

```

with tab1:
    with st.form("login_form"):
        email = st.text_input("Email")
        password = st.text_input("Пароль", type="password")
        submit = st.form_submit_button("Увійти")
        if submit and email and password:
            response = login(email, password)
            if response and 'token' in response:
                st.session_state.token = response['token']
                st.session_state.role = response['role']
                st.success("Успішний вхід!")
                st.rerun()

with tab2:
    with st.form("register_form"):
        email = st.text_input("Email")
        password = st.text_input("Пароль", type="password")
        role = st.selectbox("Роль", ["client", "manager"])
        submit = st.form_submit_button("Зареєструватися")
        if submit and email and password:
            response = register(email, password, role)
            if response and 'message' in response:
                st.success(response['message'])

def show_services_page():
    """Сторінка послуг"""
    st.header("📦 Доступні послуги")
    services = get_services(st.session_state.token)
    if not services:
        st.warning("Немає доступних послуг")
        return
    # Групуємо послуги за типом
    service_types = {
        "Вантажоперевезення міжміські": "🚚",
        "Вантажоперевезення міжнародні": "🌐",
        "Складське зберігання": "📦",
        "Митне оформлення": "📋",
        "Страхування вантажів": "🔒"
    }
}

# Створюємо вкладки для різних типів послуг
tabs = st.tabs([f"{icon} {name}" for name, icon in service_types.items()])
for tab, (service_type, icon) in zip(tabs, service_types.items()):
    with tab:
        # Фільтруємо послуги за типом
        type_services = [s for s in services if s['type'] == service_type]
        if not type_services:
            st.info(f"Немає доступних послуг типу {service_type}")
            continue
        for service in type_services:
            with st.expander(f"{service['name']}"):
                col1, col2 = st.columns([3, 1])
                with col1:
                    st.write(f"**Опис:** {service.get('description', 'Опис відсутній')}")
                    # Показуємо специфічні деталі в залежності від типу послуги
                    if "Вантажоперевезення" in service_type:
                        vehicle_icons = {
                            'до 1т': '🚚',
                            'до 5т': '🚚',
                            'до 10т': '🚚',
                            'до 20т': '🚚'
                        }
                    }
                    if service.get('vehicle_type'):
                        st.write(f"**Тип транспорту:** {vehicle_icons.get(service['vehicle_type'], '')} {service['vehicle_type']}")
                if service.get('base_price_per_km'):

```

```

        st.write(f"***Базова ціна за км:**
{service['base_price_per_km']} грн")
        if "міжнародні" in service_type and
service.get('border_crossing_price'):
            st.write(f"***Ціна за перетин кордону:**
{service['border_crossing_price']} грн")
        elif service_type == "Складське зберігання":
            if service.get('storage_price_per_day'):
                st.write(f"***Ціна за день:**
{service['storage_price_per_day']} грн")
            if service.get('storage_price_per_cubic_meter'):
                st.write(f"***Ціна за м³:**
{service['storage_price_per_cubic_meter']} грн")
            if 'temperature_control' in service:
                st.write(f"***Температурний режим:** {'Так' if
service['temperature_control'] else 'Ні'}")
            elif service_type == "Митне оформлення":
                if service.get('customs_clearance_price'):
                    st.write(f"***Базова ціна оформлення:**
{service['customs_clearance_price']} грн")
                if service.get('customs_declaration_price'):
                    st.write(f"***Ціна декларації:**
{service['customs_declaration_price']} грн")
                if service.get('customs_additional_services'):
                    st.write(f"***Додаткові послуги:**
{service['customs_additional_services']}")
            elif service_type == "Страхування вантажів":
                if service.get('insurance_rate'):
                    st.write(f"***Страховий тариф:** {service['insurance_rate']}%")
                if service.get('min_insurance_price'):
                    st.write(f"***Мінімальна вартість:**
{service['min_insurance_price']} грн")
                if service.get('coverage_options'):
                    st.write(f"***Варіанти покриття:**
{service['coverage_options']}")
            with col2:
                if st.button("Замовити", key=f"order_{service['id']}"):
                    st.session_state.selected_service = service
                    st.session_state.current_page = 'create_order'
                    st.rerun()

def show_create_order_page():
    """Сторінка створення замовлення"""
    service = st.session_state.selected_service
    if not service:
        st.warning("Спочатку оберіть послугу!")
        st.session_state.current_page = 'services'
        st.rerun()
    return

st.header(f"🚚 Замовлення послуги: {service['name']}")
# Показуємо деталі обраної послуги
st.subheader("Деталі послуги")
st.write(f"***Тип послуги:** {service['type']}")
# Зберігаємо дані форми в session_state
if 'form_data' not in st.session_state:
    st.session_state.form_data = {}
with st.form("order_form"):
    # Різні форми для різних типів послуг
    if "Вантажоперевезення" in service['type']:
        st.write(f"***Тип транспорту:** {service['vehicle_type']}")
        col1, col2 = st.columns(2)
        with col1:
            from_address = st.text_input("Адреса відправлення")
        with col2:
            to_address = st.text_input("Адреса доставки")

```

```

cargo_weight = st.number_input("Вага вантажу (кг)", min_value=0.0)
cargo_description = st.text_area("Опис вантажу")
# Зберігаємо введені дані
form_data = {
    'from_address': from_address,
    'to_address': to_address,
    'cargo_weight': cargo_weight,
    'cargo_description': cargo_description
}
elif service['type'] == "Складське зберігання":
    cargo_volume = st.number_input("Об'єм вантажу (м³)", min_value=0.0)
    storage_days = st.number_input("Кількість днів зберігання", min_value=1)
    temperature_required = st.checkbox("Потрібен температурний режим")
    cargo_description = st.text_area("Опис вантажу")
    form_data = {
        'cargo_volume': cargo_volume,
        'storage_days': storage_days,
        'temperature_required': temperature_required,
        'cargo_description': cargo_description
    }
elif service['type'] == "Митне оформлення":
    cargo_type = st.text_input("Тип вантажу")
    cargo_value = st.number_input("Вартість вантажу", min_value=0.0)
    additional_services = st.multiselect(
        "Додаткові послуги",
        options=service.get('customs_additional_services', '').split('\n')
    )
    form_data = {
        'cargo_type': cargo_type,
        'cargo_value': cargo_value,
        'additional_services': additional_services
    }
elif service['type'] == "Страхування вантажів":
    cargo_value = st.number_input("Вартість вантажу", min_value=0.0)
    cargo_type = st.text_input("Тип вантажу")
    if service.get('coverage_options'):
        coverage = st.selectbox(
            "Варіант покриття",
            options=service['coverage_options'].split('\n')
        )
    form_data = {
        'cargo_value': cargo_value,
        'cargo_type': cargo_type,
        'coverage': coverage if 'coverage' in locals() else None
    }
# Загальні поля для всіх типів послуг
contact_phone = st.text_input("Контактний телефон")
additional_info = st.text_area("Додаткова інформація")
form_data.update({
    'contact_phone': contact_phone,
    'additional_info': additional_info
})
submitted = st.form_submit_button("Розрахувати вартість")
if submitted:
    st.session_state.form_data = form_data
# Якщо форма була відправлена і дані збережені
if 'form_data' in st.session_state and st.session_state.form_data:
    with st.spinner("Розрахунок вартості..."):
        calculation_data = {
            'service_id': service['id'],
            **st.session_state.form_data
        }
    # Отримуємо розрахунок вартості
    price_data = calculate_price(calculation_data)
    if price_data:

```

```

st.success(f"Розрахована вартість: {price_data['total_price']} грн")
# Кнопка підтвердження замовлення поза формою
if st.button("Підтвердити замовлення"):
    order_data = {
        'service_id': service['id'],
        **st.session_state.form_data,
        'calculated_price': price_data['total_price']
    }
    response = create_order(st.session_state.token, order_data)
    if response and 'tracking_number' in response:
        st.success(
            f"Замовлення створено! "
            f"Номер для відстеження: {response['tracking_number']}"
        )
        # Очищаємо дані форми та вибрану послугу
        st.session_state.form_data = {}
        st.session_state.selected_service = None
        st.session_state.current_page = 'orders'
        st.rerun()

def show_orders_page():
    """Сторінка перегляду замовлень"""
    st.header("📄 Мої замовлення")
    # Фільтри
    col1, col2, col3 = st.columns(3)
    with col1:
        status_filter = st.selectbox(
            "Статус",
            ["Bci", "new", "accepted", "processing", "in_storage",
             "in_transit", "customs", "delivered", "cancelled"]
        )
    with col2:
        date_from = st.date_input("Від", value=datetime.now() - timedelta(days=30))
    with col3:
        date_to = st.date_input("До", value=datetime.now())
    # Отримання замовлень
    filters = {}
    if status_filter != "Bci":
        filters['status'] = status_filter
    if date_from:
        filters['date_from'] = date_from.isoformat()
    if date_to:
        filters['date_to'] = date_to.isoformat()
    orders = get_orders(st.session_state.token, **filters)
    if not orders:
        st.info("Немає замовлень за вказаними критеріями")
        return
    # Відображення замовлень
    for order in orders:
        with st.expander(
            f"Замовлення #{order.get('tracking_number', 'N/A')} "
            f"({order.get('status', 'невідомо')}) - {order.get('created_at', 'дата "
            f"невідомо')}"
        ):
            # Основна інформація
            st.write(f"**Тип послуги:** {order.get('service_type', 'Не вказано')}")
            st.write(f"**Назва послуги:** {order.get('service_name', 'Не вказано')}")
            st.write(f"**Статус:** {order.get('status', 'Не вказано')}")
            # Показуємо деталі замовлення в залежності від типу послуги
            service_type = order.get('service_type', '')
            if "Вантажоперевезення" in service_type:
                st.write(f"**Від:** {order.get('from_address', 'Не вказано')}")
                st.write(f"**До:** {order.get('to_address', 'Не вказано')}")
                st.write(f"**Відстань:** {order.get('distance', 0)} км")
                if order.get('border_crossings', 0) > 0:

```

```

        st.write(f"""Перетинів кордону:** {order['border_crossings']}""")
    elif service_type == "Складське зберігання":
        if order.get('cargo_volume'):
            st.write(f"""Об'єм:** {order['cargo_volume']} м³""")
        if order.get('storage_days'):
            st.write(f"""Днів зберігання:** {order['storage_days']}""")
    elif service_type == "Митне оформлення":
        if order.get('cargo_type'):
            st.write(f"""Тип вантажу:** {order['cargo_type']}""")
        if order.get('cargo_value'):
            st.write(f"""Вартість вантажу:** {order['cargo_value']} грн""")
    elif service_type == "Страхування вантажів":
        if order.get('cargo_value'):
            st.write(f"""Вартість вантажу:** {order['cargo_value']} грн""")
    # Загальна вартість для всіх типів
    st.write(f"""Загальна вартість:** {order.get('price', 0)} грн""")
    # Можливість скасування для нових замовлень
    if order.get('status') == 'new':
        if st.button("Скасувати", key=f"cancel_{order['tracking_number']}"):
            response = cancel_order(st.session_state.token, order['tracking_number'])
            if response:
                st.success("Замовлення скасовано!")
                st.rerun()

def show_analytics_page():
    """Сторінка аналітики"""
    st.header("📊 Аналітика")
    # Період аналізу
    col1, col2 = st.columns(2)
    with col1:
        date_from = st.date_input(
            "Від",
            value=datetime.now() - timedelta(days=30)
        )
    with col2:
        date_to = st.date_input(
            "До",
            value=datetime.now()
        )
    analytics = get_analytics(
        st.session_state.token,
        date_from=date_from.isoformat(),
        date_to=date_to.isoformat()
    )
    if not analytics:
        st.warning("Немає даних для аналізу")
        return
    # Вивід отриманих даних для дебагу
    st.write("Отримані дані з API:", analytics)
    # Загальна статистика
    if 'general_stats' in analytics:
        st.subheader("Загальна статистика")
        col1, col2, col3, col4 = st.columns(4)
        with col1:
            st.metric("Всього замовлень", analytics['general_stats']['total_orders'])
        with col2:
            st.metric("Виконано", analytics['general_stats']['completed_orders'])
        with col3:
            st.metric("Скасовано", analytics['general_stats']['cancelled_orders'])
        with col4:
            st.metric("Активні", analytics['general_stats']['active_orders'])
    # Фінансова статистика
    if 'financial_stats' in analytics:
        st.subheader("Фінансові показники")
        col1, col2 = st.columns(2)

```

```

        with col1:
            st.metric("Загальний дохід", f"{analytics['financial_stats']['total_revenue']}
грн")
        with col2:
            st.metric("Середній чек", f"{analytics['financial_stats']['average_order_price']}
грн")
# Графік по типам транспорту
if 'transport_stats' in analytics and analytics['transport_stats']:
    st.subheader("Статистика по типам транспорту")
    transport_df = pd.DataFrame(analytics['transport_stats'])
    # Вивід датафрейму для дебагу
    st.write("DataFrame для графіка:", transport_df)
    if not transport_df.empty:
        try:
            fig = px.bar(
                transport_df,
                x='vehicle_type',
                y='total_orders',
                title='Замовлення по типам транспорту'
            )
            st.plotly_chart(fig)
            # Окремий графік для доходів
            fig2 = px.bar(
                transport_df,
                x='vehicle_type',
                y='total_revenue',
                title='Дохід по типам транспорту'
            )
            st.plotly_chart(fig2)
        except Exception as e:
            st.error(f"Помилка при створенні графіка: {str(e)}")
# Популярні маршрути
if 'popular_routes' in analytics and analytics['popular_routes']:
    st.subheader("Популярні маршрути")
    routes_df = pd.DataFrame(analytics['popular_routes'])
    # Вивід датафрейму для дебагу
    st.write("DataFrame для маршрутів:", routes_df)
    if not routes_df.empty:
        try:
            fig = px.bar(
                routes_df,
                x='total_orders',
                y='from',
                title='Топ маршрутів за кількістю замовлень',
                orientation='h'
            )
            st.plotly_chart(fig)
        except Exception as e:
            st.error(f"Помилка при створенні графіка маршрутів: {str(e)}")

def show_manager_services_page():
    """Сторінка управління послугами для менеджера"""
    st.header("🚚 Управління послугами")
    # Словник з іконками транспорту
    vehicle_icons = {
        'до 1т': '🚗', # або URL зображення
        'до 5т': '🚚', # або URL зображення
        'до 10т': '🚛', # або URL зображення
        'до 20т': '🚚', # або URL зображення
    }
    # Вкладки для різних типів послуг
    tabs = st.tabs([
        "🚚 Міжміські перевезення",
        "🌐 Міжнародні перевезення",
    ])

```

```

        "🏠 Складське зберігання",
        "📄 Митне оформлення",
        "🚚 Страхування вантажів"
    ])
    # 1. Міжміські перевезення
    with tabs[0]:
        with st.form("local_transport_form"):
            st.subheader("Створення послуги міжміського перевезення")
            name = st.text_input("Назва послуги")
            description = st.text_area("Опис послуги")
            # Вибір типу транспорту з іконками
            st.write("**Оберіть тип транспорту:**")
            cols = st.columns(4)
            vehicle_type = None
            for i, (vtype, icon) in enumerate(vehicle_icons.items()):
                with cols[i]:
                    st.write(icon, vtype)
                    if st.checkbox("Обрати", key=f"local_{vtype}"):
                        vehicle_type = vtype
            base_price_per_km = st.number_input("Базова ціна за км", min_value=0.0, step=0.1,
            key="local_price")
            if st.form_submit_button("Додати послугу"):
                if not vehicle_type:
                    st.error("Оберіть тип транспорту!")
                    return
                service_data = {
                    'name': name,
                    'type': "Вантажоперевезення міжміські",
                    'description': description,
                    'vehicle_type': vehicle_type,
                    'base_price_per_km': base_price_per_km,
                    'border_crossing_price': 0
                }
                response = create_service(st.session_state.token, service_data)
                if response:
                    st.success("Послугу успішно додано!")
                    st.rerun()
    # 2. Міжнародні перевезення
    with tabs[1]:
        with st.form("international_transport_form"):
            st.subheader("Створення послуги міжнародного перевезення")
            name = st.text_input("Назва послуги")
            description = st.text_area("Опис послуги")
            st.write("**Оберіть тип транспорту:**")
            cols = st.columns(4)
            vehicle_type = None
            for i, (vtype, icon) in enumerate(vehicle_icons.items()):
                with cols[i]:
                    st.write(icon, vtype)
                    if st.checkbox("Обрати", key=f"int_{vtype}"):
                        vehicle_type = vtype
            base_price_per_km = st.number_input("Базова ціна за км", min_value=0.0, step=0.1,
            key="int_price")
            border_crossing_price = st.number_input("Ціна за перетин кордону", min_value=0.0,
            step=100.0)
            if st.form_submit_button("Додати послугу"):
                if not vehicle_type:
                    st.error("Оберіть тип транспорту!")
                    return
                service_data = {
                    'name': name,
                    'type': "Вантажоперевезення міжнародні",
                    'description': description,
                    'vehicle_type': vehicle_type,

```

```

        'base_price_per_km': base_price_per_km,
        'border_crossing_price': border_crossing_price
    }
    response = create_service(st.session_state.token, service_data)
    if response:
        st.success("Послугу успішно додано!")
        st.rerun()
# 3. Складське зберігання
with tabs[2]:
    with st.form("storage_form"):
        st.subheader("Створення послуги складського зберігання")
        name = st.text_input("Назва послуги")
        description = st.text_area("Опис послуги")
        storage_price_per_day = st.number_input("Ціна за день зберігання", min_value=0.0,
step=1.0)
        storage_price_per_cubic_meter = st.number_input("Ціна за кубічний метр",
min_value=0.0, step=1.0)
        temperature_control = st.checkbox("Наявність температурного режиму")
        if st.form_submit_button("Додати послугу"):
            service_data = {
                'name': name,
                'type': "Складське зберігання",
                'description': description,
                'storage_price_per_day': storage_price_per_day,
                'storage_price_per_cubic_meter': storage_price_per_cubic_meter,
                'temperature_control': temperature_control
            }
            response = create_service(st.session_state.token, service_data)
            if response:
                st.success("Послугу успішно додано!")
                st.rerun()
# 4. Митне оформлення
with tabs[3]:
    with st.form("customs_form"):
        st.subheader("Створення послуги митного оформлення")
        name = st.text_input("Назва послуги")
        description = st.text_area("Опис послуги")
        customs_clearance_price = st.number_input("Базова ціна оформлення", min_value=0.0,
step=100.0)
        customs_declaration_price = st.number_input("Ціна декларації", min_value=0.0,
step=100.0)
        customs_additional_services = st.text_area("Додаткові послуги")
        if st.form_submit_button("Додати послугу"):
            service_data = {
                'name': name,
                'type': "Митне оформлення",
                'description': description,
                'customs_clearance_price': customs_clearance_price,
                'customs_declaration_price': customs_declaration_price,
                'customs_additional_services': customs_additional_services
            }
            response = create_service(st.session_state.token, service_data)
            if response:
                st.success("Послугу успішно додано!")
                st.rerun()
# 5. Страхування вантажів
with tabs[4]:
    with st.form("insurance_form"):
        st.subheader("Створення послуги страхування")
        name = st.text_input("Назва послуги")
        description = st.text_area("Опис послуги")
        insurance_rate = st.number_input(
            "Відсоток від вартості вантажу",
            min_value=0.0,
            max_value=100.0,

```

```

        step=0.1
    )
    min_insurance_price = st.number_input(
        "Мінімальна вартість страховки",
        min_value=0.0,
        step=100.0
    )
    coverage_options = st.text_area("Варіанти покриття")
    if st.form_submit_button("Додати послугу"):
        service_data = {
            'name': name,
            'type': "Страхування вантажів",
            'description': description,
            'insurance_rate': insurance_rate,
            'min_insurance_price': min_insurance_price,
            'coverage_options': coverage_options
        }
        response = create_service(st.session_state.token, service_data)
        if response:
            st.success("Послугу успішно додано!")
            st.rerun()

# Список існуючих послуг
st.subheader("Існуючі послуги")
services = get_services(st.session_state.token, show_all=True)
if services:
    for service in services:
        with st.expander(
            f"{service['type']} - {service['name']} "
            f"({'Активна' if service['is_active'] else 'Неактивна'})"
        ):
            st.write(f"**Опис:** {service['description']}")

            if "Вантажоперевезення" in service['type']:
                if 'vehicle_type' in service:
                    st.write(f"**Тип транспорту:** "
                        {vehicle_icons.get(service['vehicle_type'], '')} {service['vehicle_type']}")
                    st.write(f"**Базова ціна за км:** {service.get('base_price_per_km', 0)} "
                        грн")

                    if service['type'] == "Вантажоперевезення міжнародні":
                        st.write(f"**Ціна за перетин кордону:** "
                            {service.get('border_crossing_price', 0)} грн")
                    elif service['type'] == "Складське зберігання":
                        st.write(f"**Ціна за день:** {service.get('storage_price_per_day', 0)} "
                            грн")

                        st.write(f"**Ціна за м³:** {service.get('storage_price_per_cubic_meter', "
                            0)} грн")

                        st.write(f"**Температурний режим:** {'Так' if "
                            service.get('temperature_control') else 'Ні'})")
                    elif service['type'] == "Митне оформлення":
                        st.write(f"**Базова ціна оформлення:** "
                            {service.get('customs_clearance_price', 0)} грн")
                        st.write(f"**Ціна декларації:** {service.get('customs_declaration_price', "
                            0)} грн")

                        st.write(f"**Додаткові послуги:** "
                            {service.get('customs_additional_services', '')}")
                    elif service['type'] == "Страхування вантажів":
                        st.write(f"**Страховий тариф:** {service.get('insurance_rate', 0)}%")
                        st.write(f"**Мінімальна вартість:** {service.get('min_insurance_price', "
                            0)} грн")

                        st.write(f"**Варіанти покриття:** {service.get('coverage_options', '')}")

# Кнопки управління
col1, col2 = st.columns(2)
with col1:
    if st.button("Деактивувати" if service['is_active'] else "Активувати",

```

```

        key=f"toggle_{service['id']}"):
    response = update_service(
        st.session_state.token,
        service['id'],
        {'is_active': not service['is_active']}
    )
    if response:
        st.success("Статус послуги змінено!")
        st.rerun()

    with col2:
        if st.button("Видалити", key=f"delete_{service['id']}"):
            response = delete_service(st.session_state.token, service['id'])
            if response:
                st.success("Послугу видалено!")
                st.rerun()

def main():
    """Головна функція додатку"""
    st.set_page_config(
        page_title="Транспортна компанія",
        page_icon="🚚",
        layout="wide"
    )
    # Ініціалізація стану сесії
    if 'token' not in st.session_state:
        st.session_state.token = None
    if 'role' not in st.session_state:
        st.session_state.role = None
    if 'current_page' not in st.session_state:
        st.session_state.current_page = 'services'
    if 'selected_service' not in st.session_state:
        st.session_state.selected_service = None
    # Якщо користувач не авторизований
    if not st.session_state.token:
        show_login_page()
        return
    # Бічне меню
    st.sidebar.title("🚚 Меню")
    # Меню для клієнта
    if st.session_state.role == 'client':
        menu_options = {
            "services": "📦 Послуги",
            "create_order": "🚚 Створити замовлення",
            "orders": "📋 Мої замовлення"
        }
    # Меню для менеджера
    else:
        menu_options = {
            "manage_services": "🔧 Управління послугами",
            "orders": "📋 Замовлення",
            "analytics": "📊 Аналітика"
        }
    # Вибір сторінки
    st.session_state.current_page = st.sidebar.radio(
        "Навігація",
        list(menu_options.keys()),
        format_func=lambda x: menu_options[x]
    )
    # Кнопка виходу
    if st.sidebar.button("Вийти"):
        st.session_state.token = None
        st.session_state.role = None
        st.session_state.current_page = 'services'
        st.rerun()

```

```
# Відображення відповідної сторінки
if st.session_state.current_page == 'services':
    show_services_page()
elif st.session_state.current_page == 'create_order':
    show_create_order_page()
elif st.session_state.current_page == 'orders':
    show_orders_page()
elif st.session_state.current_page == 'manage_services':
    show_manager_services_page()
elif st.session_state.current_page == 'analytics':
    show_analytics_page()
if __name__ == "__main__":
    main()
```