

**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 «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології» спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЕКТ

на тему

**ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ВЕБ-СИСТЕМИ ДЛЯ
ЛОГІСТИЧНОГО ПІДПРИЄМСТВА З ОБСЛУГОВУВАННЯ ГРУЗОВИХ
АВТОМОБІЛІВ**

здобувача Скорика Владара Володимировича

Науковий керівник:

к.е.н., доцент

_____ Лозовик Ю.М.

**Проект допущений до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти
в.о. завідувача кафедри:**

к.е.н., доцент.

_____ Тішков Б.О.

Київ 2024

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

галузь знань 12 «Інформаційні технології» спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи(гарант)
освітньо-професійної програми

Іванченко Г.Ф.
“ ” 2024 р.

ЗАТВЕРДЖУЮ:

в.о. завідувача кафедри

Тішков Б.О.
“ ” 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти ***Скорика Владара Володимировича***
очної (денної) форми навчання

на підготовку випускного бакалаврського проекту на тему:
«Проектування інформаційної веб-системи для логістичного підприємства
з обслуговування вантажних автомобілів»

Тему затверджено наказом ректора Університету від « » лютого 2024 р.
№.

Кваліфікаційний бакалаврський проект виконується на матеріалах
дослідження предметної області джерела

План кваліфікаційного бакалаврського проекту

Розділ I. Характеристика та аналіз предметної галузі

Розділ II. Розробка вимог і моделювання інформаційної системи

Розділ III. Проектування та реалізація компонентів системи

Об'єкт дослідження. Процеси управління логістичними операціями та забезпечення зручного обміну інформацією між клієнтами, логістичними спеціалістами та партнерами.

Предмет дослідження. Проектування та розробка веб-системи для логістичних компаній, яка включає управління замовленнями, розподіл завдань, моніторинг транспорту, облік запасів та забезпечення можливості обміну цією інформацією між клієнтами, логістичними спеціалістами та партнерами.

Мета кваліфікаційного бакалаврського проекту. Розробка та реалізація веб-системи, яка надає логістичним компаніям зручні та ефективні інструменти для організації та оптимізації логістичних процесів і обслуговування вантажних автомобілів

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I. Описати особливості предметної області та об'єкта дослідження. Дослідити практичний досвід застосування інформаційних систем у відповідній галузі.

У розділі II. Виконати аналіз та розробити специфікацію вимог до системи. Описати формулювання задачі та алгоритми роботи системи. Провести моделювання системи.

У розділі III. Спроектувати та реалізувати компоненти системи.

Завдання підготував
науковий керівник _____

Лозовик Юрій Миколайович

“ _____ ” _____ 2024
р.

Завдання одержав
студент _____

Скорик Владар Володимирович

Відгук

наукового керівника про кваліфікаційний бакалаврський проект
студента Навчально-наукового інституту
“Інститут інформаційних технологій в економіці” галузі знань 12 “Інформаційні
технології” спеціальності 122 “Комп’ютерні науки”

Скорики Владара Володимировича

на тему:

“ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ВЕБ-СИСТЕМИ ДЛЯ ЛОГІСТИЧНОГО ПІДПРИЄМСТВА З ОБСЛУГОВУВАННЯ ГРУЗОВИХ АВТОМОБІЛІВ”

1. **Актуальність теми:** розробка інформаційної системи важлива для підприємства з обслуговування вантажних автомобілів, які зацікавлені в удосконаленні наявних сервісів шляхом інтеграції сервісного обслуговування із послугами по замовленню запчастин.
2. **Позитивні риси кваліфікаційного бакалаврського проекту:**
здійснено аналіз реалізації програмних продуктів для підприємств з сервісного обслуговування автомобілів, визначені їхні сильні та слабкі сторони, обґрунтовано необхідність створення нової веб-системи з використанням сучасного стеку технологій, таких як: Spring та JS, HTML та CSS.
3. **Наявність самостійних розробок:**
 - наявність необхідних діаграм проектування системи;
 - опис вхідних і вихідних даних інформаційної моделі;
 - розроблені прототип серверної частини та прототип клієнтської частини;
 - використання сучасного стеку технологій.
4. **Наявність недоліків:**
 - відсутні посилання на використані джерела;
 - важко відокремити зміст автора від запозичень;
 - відсутність повністю робочого прототипу системи;
 - більшість веб-сторінок розроблені зі статичними даними;
 - серверна частина не зв’язана з клієнтською;
 - відсутній код серверної частини застосунку;
 - слабка реалізація програмного продукту;
 - база даних більше підходить до реалізації онлайн магазину, ніж до підприємства, яке займається наданням сервісних послуг вантажним автомобілям.
5. **Загальна оцінка кваліфікаційного бакалаврського проекту та його допущення до захисту перед Екзаменаційною комісією:**

Висвітлене вище дозволяє констатувати, що дипломний проєкт відповідає встановленим вимогам, що висуваються до таких видів робіт, а його автор, Скорик В.В., володіє навиками проектування та розробки веб-системи, яка надає логістичним компаніям зручні та ефективні інструменти для організації та оптимізації логістичних процесів і обслуговування вантажних автомобілів. Дипломна робота рекомендується до захисту в ДЕК і може бути оцінена на «добре».

4 червня 2024 р.

Науковий керівник

доцент, кандидат економічних наук

Лозовик Ю. М.

Рецензія

на кваліфікаційний бакалаврський проєкт

Скорики Владара Володимировича

тема

«Проектування веб-системи для логістичного підприємства з обслуговування грузових автомобілів»

Актуальність теми кваліфікаційного бакалаврського проєкту і доцільність його розроблення. У сучасному логістичному середовищі підприємства стикаються з численними викликами, пов'язаними з обслуговуванням грузових автомобілів. Веб-система, що оптимізує ці процеси, є важливим інструментом для покращення ефективності та організації роботи. Потреба в такій системі зростає з кожним роком, що робить розробку цього проєкту актуальним завданням.

Якість проведеного дослідження. Автор проєкту глибоко розуміє специфіку проблем, з якими стикаються логістичні підприємства при обслуговуванні грузових автомобілів, та ретельно дослідив можливі шляхи їх вирішення. Методи дослідження чітко описані, а використовувані інструменти базуються на надійних академічних джерелах.

Позитивні риси кваліфікаційного бакалаврського проєкту. Перш за все, автором виконано значний аналіз потреб і вимог логістичних підприємств, що сприяло розробці функціональних можливостей системи, які найбільш ефективно відповідають їхнім потребам. Дизайн інтерфейсу також добре продуманий і забезпечує зручну навігацію та користувацький досвід.

Зауваження. У проєкті можна було більш детально описати процес розробки системи, зосередившись на виборі технологій і архітектурних рішень.

Практична значимість висновків і рекомендацій. Сформовані висновки є чіткими та повністю описують переваги та недоліки розробленої системи, а також можливі шляхи для її вдосконалення. Рекомендації щодо подальшого розвитку системи також є цінними для майбутніх досліджень у цій галузі.

ВСП «Фаховий коледж інформаційних
систем і технологій КНЕУ ім. В. Гетьмана»

завідуюча відділення «Програмування» _____ Світлана ЯНЧУК
підпис, ПІБ

Місце печатки організації, де працює рецензент

АНОТАЦІЯ

кваліфікаційного бакалаврського проекту студента 4 курсу навчально-наукового інституту «Інститут інформаційних технологій в економіці»

Скорика Владара Володимировича, виконаної на тему:

«Проектування інформаційної веб-системи для логістичного підприємства з обслуговування грузових автомобілів»

Київ: кафедра інформаційних систем в економіці, 2024 р.

Випускний бакалаврський проект присвячений актуальній темі оптимізації логістичних процесів та обслуговування вантажних автомобілів, а також забезпеченню зручного обміну інформацією між клієнтами, логістичними спеціалістами та партнерами.

Кваліфікаційний бакалаврський проект складається з трьох розділів, логічно пов'язаних між собою.

У першому розділі надається характеристика предметної галузі, обґрунтовується потреба у створенні нової інформаційної системи, а також наводиться перелік існуючих програмних засобів.

Другий розділ присвячений обґрунтуванню методологій проектування та моделювання системи, аналізу вимог та опису функціональності системи. Тут розглядається формулювання задачі та алгоритми роботи системи.

У третьому розділі детально розглядається інформаційне, технічне, програмне та організаційне забезпечення системи. Описується структура інформаційного забезпечення, обґрунтовується вибір технічних засобів та програмного забезпечення для реалізації серверної та клієнтської частини системи. Також тут зазначені інформаційні масиви, які є складовою розв'язання поставленої задачі.

РЕФЕРАТ

Кваліфікаційний бакалаврський проект містить 60 сторінок роботи та 99 сторінок загалом, 12 таблиць, 61 рисунок, список літератури з 8 найменувань, 3 додатки.

Проектування інформаційної веб-системи для логістичного підприємства з обслуговування грузових автомобілів

Ключові слова: проектування, розробка, клієнт-сервер, CASE-засоби, Windows, графічний інтерфейс, API-запити.

В контексті логістичного підприємства сучасні логістичні сервіси стали необхідним елементом щоденного функціонування, сприяючи зручному веденню логістичних операцій. Ці сервіси дозволяють ефективно відслідковувати рух вантажів та оптимізувати ланцюжок постачання, використовуючи компоненти, такі як база даних, файловий сервер, система оплати, користувацьке клієнтське програмне забезпечення та інші, які були впроваджені в рамках проектування для вдосконалення логістичних процесів та забезпечення їхньої ефективної управлінської підтримки.

Для проектування такої системи було використано низку CASE-засобів, а саме: Lucidchart, Enterprise Architect. Для розробки було обрано платформу IntelliJ IDEA. Ця платформа передбачає розробку як користувацьких додатків так і серверних. Для розробки користувацького інтерфейсу було використано фреймворк SpringBoot.

В результаті проектування було отримано проект, який виконує необхідні вимоги. Для використання проекту необхідне його встановлення серверного додатку і клієнтського на різні машини. Також необхідно забезпечити безперебійну роботу сервера. Такий проект має досить високу ефективність роботи, адже користувач в будь-який момент може отримати доступ до системи.

Рік виконання випускного бакалаврського проекту: 2024р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, ВЕЛИЧИН І ТЕРМІНІВ.....	3
ВСТУП.....	4
РОЗДІЛ 1. Характеристика та аналіз предметної галузі.....	4
1.1. Характеристика предметної галузі та об'єкта дослідження	6
1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі	7
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	4
2.1. Аналіз і специфікація вимог до інформаційної системи	14
2.1.1.Бізнес-вимоги.....	14
2.1.2.Нефункціональні вимоги.....	16
2.1.3.Функціональні вимоги.....	18
2.2. Постановка та алгоритм розв'язання задачі	19
2.2.1.Постановка задачі.....	19
2.2.2.Вихідна інформація	20
2.2.3.Вхідна інформація.....	22
2.2.4.Алгоритм розв'язання задачі.....	25
2.3. Моделювання інформаційної системи.....	28
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	39
3.1. Інформаційне забезпечення.....	39
3.2. Технічне забезпечення.....	55
3.3. Програмне забезпечення	56
3.4. Результати реалізації інформаційної системи	59
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	66
ДОДАТКИ	71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, ВЕЛИЧИН І ТЕРМІНІВ

CSS	Cascading Style Sheets (Каскадні таблиці стилів)
JSP	Java Server Pages
RAD	Rapid Application Development
ООП	Об'єктно-орієнтований підхід
ОС	Операційна система
HTML	HyperText Markup Language (Мова гіпертекстової розмітки)
MVC	Model-View-Controller (Модель-представлення-контролер)
NET	.NET (Платформа розробки програмного забезпечення)
PDF	Portable Document Format (Портативний формат документа)
SQL	Structured Query Language (Мова структурованих запитів)
UML	Unified Modeling Language (Уніфікована мова моделювання)
ПЗ	Програмне Забезпечення
ПК	Персональний Комп'ютер
СКБД	Система Керування Базами Даних
ІС	Інформаційна система
MS WORD	Microsoft Word
IntelliJ IDEA	IntelliJ IDEA

ВСТУП

У сучасному бізнес-середовищі логістика та обслуговування грузових автомобілів стають стратегічно важливими сегментами підприємницької діяльності. Швидкий та ефективний обмін інформацією в цих галузях може суттєво підвищити продуктивність та конкурентоспроможність підприємств.

Метою даного проекту є розробка інформаційної веб-системи, спрямованої на оптимізацію логістичних процесів та обслуговування грузових автомобілів. Система має за мету полегшити взаємодію між клієнтами, логістичними спеціалістами та партнерами, забезпечуючи зручний та ефективний обмін інформацією.

Тема даного проекту полягає у дослідженні методів та засобів розробки веб-системи для управління логістичними процесами та обслуговування вантажних автомобілів. У рамках дослідження буде проведено аналіз існуючих рішень, розробка та тестування запропонованої системи, а також оцінка її ефективності у реальних умовах експлуатації.

Предметом дослідження є інформаційні системи у сфері логістики та обслуговування транспортних засобів. Основну увагу приділено інтеграції різних компонентів системи для забезпечення оптимального функціонування логістичних процесів, управління транспортом та взаємодії з клієнтами та партнерами.

Практичне значення. Розроблена програма може функціонувати як веб-сервіс, що надає послуги логістичного обслуговування. Це стане у нагоді як логістичним компаніям, які зможуть оптимізувати свої процеси, так і водіям та клієнтам, що зможуть зручно взаємодіяти з системою.

Використані інструментальні засоби розробки. Під час створення даного проекту використовувались інструменти розробки, такі як середовище IntelliJ IDEA, система управління базами даних MySQL, CASE-засіб LucidChart. Для підготовки та оформлення пояснювальної записки використовувався текстовий редактор Microsoft Word 2019.

Структура роботи. Сам проект включає вступ, розділ, присвячений дослідженню комплексу задач предметної області, порівняльну характеристику програмних систем, схему інформаційних взаємозв'язків задач, опис бази даних, діаграму компонентів, аналіз вхідної та вихідної інформації, висновки та додатки.

РОЗДІЛ 1. Характеристика та аналіз предметної галузі

1.1. Характеристика предметної галузі та об'єкта дослідження

Сервіс для логістичного підприємства – це система, призначена для полегшення взаємодії між клієнтами, логістичними спеціалістами та партнерами, сприяючи оптимізації логістичних процесів та обслуговування грузових автомобілів. За основу взяті ідеї форуму, де клієнти можуть розміщувати свої замовлення, описувати деталі та вибирати логістичних спеціалістів зі списку. Логістичні спеціалісти, у свою чергу, отримують доступ до усіх замовлень та можуть надсилати рекомендації щодо обробки.

Майбутніми користувачами можуть бути замовники, які шукають ефективний спосіб здійснення та контролю над транспортними послугами. Також, логістичні компанії, які бажають оптимізувати свої процеси та забезпечити зручну взаємодію з клієнтами, можуть використовувати цю систему. Крім того, це може стати додатковим джерелом доходу для логістичних спеціалістів, які мають можливість обробляти більше замовлень та отримувати додаткові вигоди. Для підприємств різного масштабу ця система відкриває можливість ефективного планування маршрутів, зменшення витрат на паливо та зниження ризиків втрати вантажу через прозорий облік та моніторинг усіх етапів транспортування.

Система повинна вирішувати проблему оптимізованого та безпечного обміну інформацією між усіма учасниками логістичного процесу. Це особливо актуально в умовах зростання конкуренції та підвищення вимог до швидкості та якості обслуговування. У період локдауну та обмежень, цей сервіс може стати ключовим інструментом для надання логістичних послуг та підтримки комунікації між учасниками ланцюга постачання.

1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Вивчення існуючих інформаційних систем для логістичного підприємства є актуальним завданням, оскільки цей сегмент ринку є досить новим. Раніше ця тематика не привертала належної уваги, але з плином часу в Україні з'явилися різноманітні сервіси, які пов'язані з логістикою грузових автомобілів.

Для порівняння було обрано такі ІС: «MercuryGate TMS», «Oracle Transportation Management», «Samsara» та «Descartes Systems Group».

MercuryGate TMS - це інноваційна транспортно-логістична система, що надає розширені рішення для управління та оптимізації логістичних процесів у сфері транспорту та поставок. Виробник, компанія MercuryGate International, вивів на ринок цей інтегрований пакет програм, спрямований на поліпшення ефективності та продуктивності логістичних операцій. MercuryGate TMS дозволяє ефективно управляти транспортними операціями, включаючи планування маршрутів, відстеження вантажів та оптимізацію використання транспортних ресурсів.

Система спрощує процеси створення та обробки електронних накладних, що сприяє автоматизації та зменшенню паперової робочої документації. MercuryGate TMS інтегрується з митними службами та допомагає забезпечити відповідність з усіма регуляторними вимогами та правилами. Надає великий набір аналітичних інструментів для моніторингу та аналізу логістичних даних, що дозволяє приймати обґрунтовані рішення. Система дозволяє управляти запасами, визначати оптимальні рівні запасів та автоматизувати використання складського простору.

Легко інтегрується з іншими системами та має гнучку архітектуру для розширення функціональності залежно від потреб користувача. Інтегрується з іншими бізнес-системами для максимальної зручності та спрощення управління. Також оптимізує витрати та час, що дозволяє збільшити ефективність логістичних процесів. Надає рішення, що можна адаптувати до різних типів

логістичних бізнесів та їхніх потреб. Забезпечує повний обсяг аналітики та звітності для детального аналізу та прийняття стратегічних рішень.

MercuryGate TMS є потужним рішенням для підприємств, що мають високі вимоги до управління транспортно-логістичними операціями та бажають оптимізувати свої логістичні процеси.

Дизайн представлено на рис. 1.1.

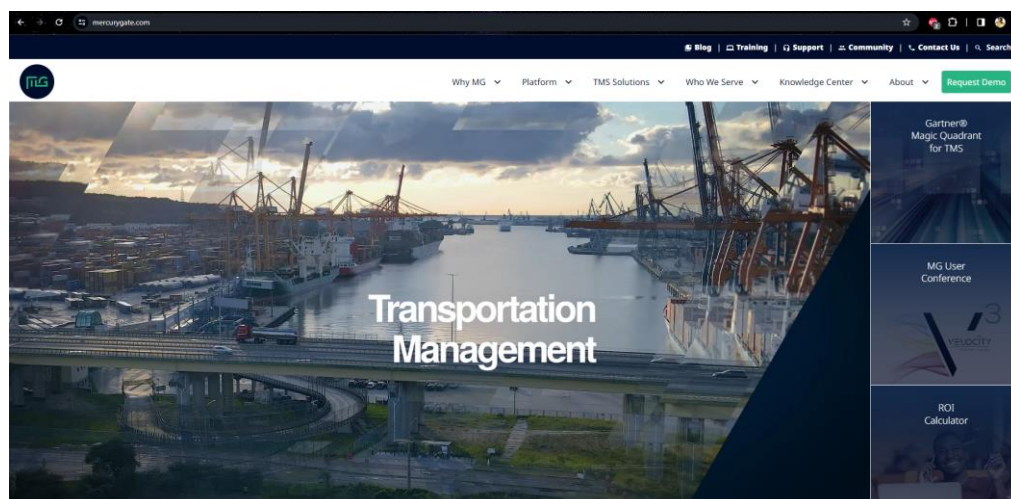


Рисунок 1.1 – Головна сторінка сайту MercuryGate TMS

Oracle Transportation Management (OTM) є потужною та інтегрованою системою управління транспортом, розробленою Oracle Corporation. Це рішення спрямоване на оптимізацію та ефективне управління вантажними перевезеннями та логістичними процесами у великих ланцюгах постачання. Важливо відзначити, що Oracle Transportation Management є частиною більшої екосистеми продуктів Oracle, які охоплюють різні сфери бізнесу.

OTM надає інструменти для планування маршрутів, оптимізації вантажів, управління транспортними ресурсами та розподілу завдань. Система ж дозволяє в режимі реального часу відстежувати рух вантажів, контролювати події та управляти виконанням перевезень. Є легко інтегрованою для інших систем, таких як системи управління складами, системи управління замовленнями, та надає потужні механізми комунікації з постачальниками та перевізниками. Може бути розширено за допомогою спеціалізованих модулів для вирішення конкретних завдань, таких як управління міжнародними перевезеннями, обробка

вантажів в об'ємах, та інші.

Також дозволяє ефективно керувати витратами на перевезення, оптимізуючи маршрути та використовуючи економії масштабу. Надає розширені аналітичні засоби та можливості створення звітів для вивчення та вдосконалення логістичних процесів. Система підтримує глобальні логістичні виклики, дозволяючи оптимізувати та управляти перевезеннями на різних континентах.

Oracle Transportation Management ставить своїм завданням полегшити та оптимізувати логістичні процеси, забезпечуючи компаніям засоби для вирішення складних завдань управління транспортом та ланцюгом постачання.

Дизайн представлено на рис. 1.2.

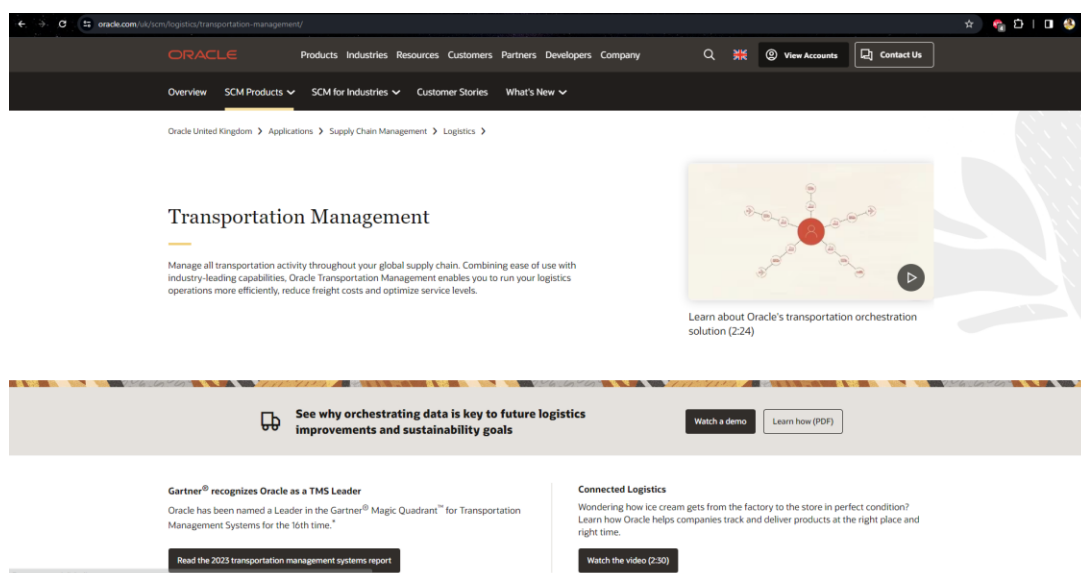


Рисунок 1.2 – Сторінка сайту Oracle Transportation Management

Samsara — це технологічна компанія, яка спеціалізується на інтернеті речей (IoT) і надає рішення для відстеження та управління флотами та інфраструктурою.

Samsara надає рішення для відстеження руху транспортних засобів, таких як грузовики, автобуси або легкові автомобілі. Це включає в себе GPS-відстеження, моніторинг швидкості та використання пального. Компанії можуть використовувати Samsara для відстеження та контролю за станом вантажів у реальному часі. Також дозволяє вимірювати та аналізувати витрати пального, що є важливим елементом ефективного управління транспортним флотом.

Крім рішень для транспортних засобів, Samsara пропонує різноманітні IoT-сенсори для моніторингу робочих процесів, таких як температурний контроль, відстеження активності пристроїв та інші. Система включає в себе можливості відеомоніторингу, які можуть бути використані для безпеки та внутрішнього контролю.

Надає інструменти для аналізу даних, зокрема використовуючи великий обсяг інформації, щоб допомогти компаніям зробити кращі стратегічні рішення. Рішення Samsara також можуть включати функції для відстеження робочого часу працівників та забезпечення безпеки на робочому місці.

Дизайн представлено на рис. 1.3.

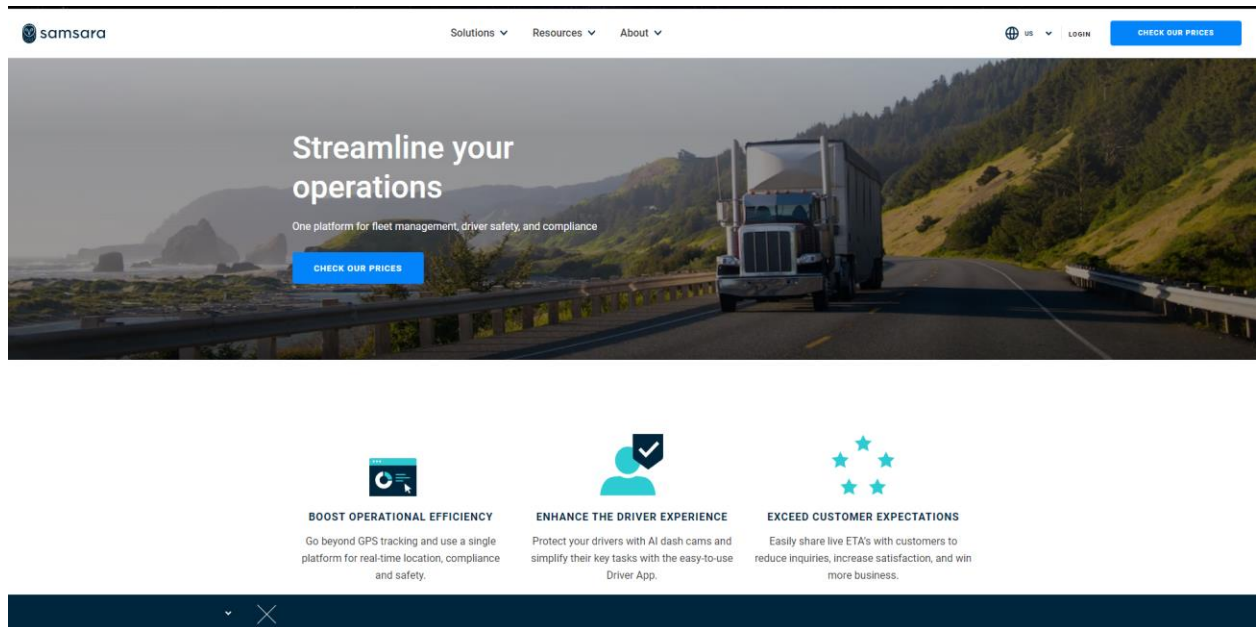


Рисунок 1.3 – Сторінка сайту Samsara

Descartes Systems Group - це канадська компанія, що спеціалізується на наданні рішень у сфері логістики та управління цепочками постачання. Її програмні продукти орієнтовані на оптимізацію та автоматизацію різноманітних логістичних процесів для підприємств різних галузей.

Ключовим компонентом їхнього підходу є глобальна логістична мережа, із забезпеченням підключення та обміном даними між різними сторонами цепочок постачання. Це може включати виробників, перевізників, роздрібних продавців і багато інших. Пропонує інструменти для оптимізації маршрутів, мобільних

рішень та телематики для вирішення завдань з управління транспортним флотом. Продукти Descartes включають рішення для управління складами, які допомагають оптимізувати процеси відправлення та отримання товарів. Спеціалізовані інструменти для вирішення питань митної декларації та виконання вимог щодо регуляторної відповідності.

Descartes надає доступ до світового торговельного контенту та іншої інформації, що допомагає клієнтам приймати обґрунтовані рішення в сфері логістики та торгівлі. Також має інструменти для відстеження та моніторингу руху товарів в реальному часі для покращення видимості в цепочці постачання. Наявні аналітичні інструменти для обробки та аналізу даних, спрямовані на забезпечення ефективного управління логістикою та постачанням.

Descartes Systems Group є великим гравцем у сфері логістики та цепочок постачання, і їхні рішення використовуються компаніями різного масштабу та галузей для оптимізації їхніх бізнес-процесів. Більш детальну інформацію можна знайти на офіційному веб-сайті компанії чи інших надійних джерелах.

Дизайн представлено на рис. 1.4.

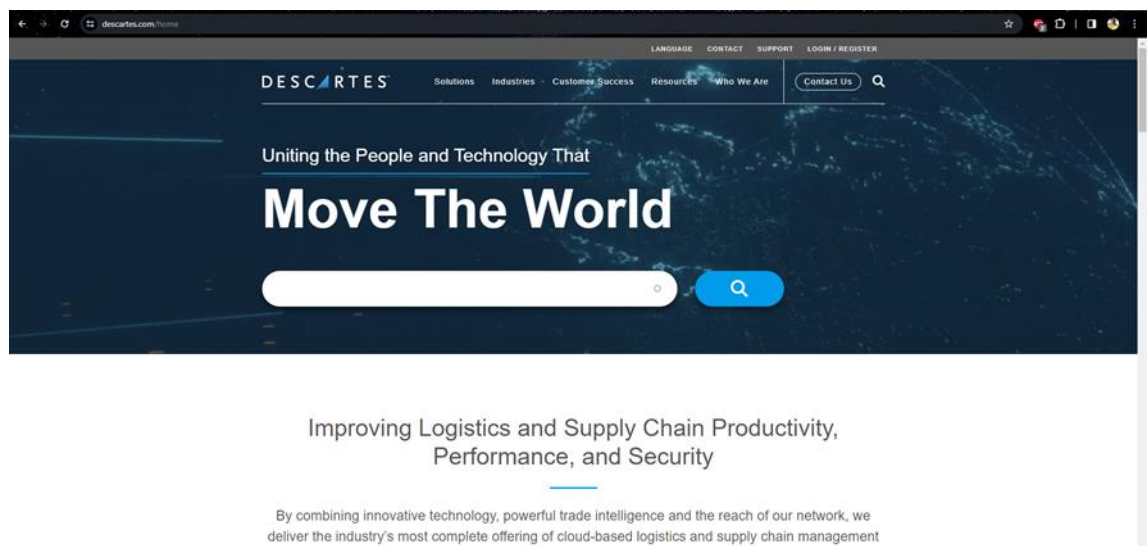


Рисунок 1.4 – Головна сторінка сайту Descartes Systems Group Порівняння можливостей даних систем наведено в табл. 1.1.

Всі чотири системи надають рішення для глобального логістичного управління та мають функціонал для маршрутизації, відстеження, аналітики та звітності.

Таблиця 1.1 – Порівняльна таблиця характеристик систем

Характеристика	MercuryGate TMS	Oracle Transportation Management (OTM)	SAP Integrated Business Planning (IBP)	Descartes Systems Group
Глобальна логістична мережа	Так	Так	Так	Так
Маршрутизація та транспорт	Так	Так	Так	Так
Управління складами	Є функціонал, але менше виражений	Так	Є функціонал, але менше виражений	Так
Митна декларація та відповідність	Так	Так	Так	Так
Світовий торговельний контент	Обмежений доступ, більше внутрішніх рішень	Так	Не надається	Так
Відстеження та моніторинг	Так	Так	Так	Так
Аналітика та звітність	Так	Так	Так	Так
Інтеграція з іншими системами	Підтримується	Широкі можливості інтеграції з іншими продуктами Oracle	Інтеграція з іншими продуктами SAP	Є можливості інтеграції з іншими системами та службами
Написати листа у службу підтримки	Так	Так	Ні	Ні

Oracle Transportation Management і SAP IBP відзначаються вираженим функціоналом управління складами та митною декларацією.

Descartes Systems Group відомий своєю глобальною логістичною мережею та можливостями інтеграції з іншими системами. В кожній системі наявні якісь певні переваги та обмеження, які необхідно врахувати при виборі системи для конкретного підприємства. Одними з ключових характеристик є висока функціональність, інтеграційна готовність та зручність у використанні. Ці рішення створені для ефективного управління логістичними викликами та надають комплексні інструменти для оптимізації та контролю над логістичними процесами на підприємствах різних розмірів.

Навчання та підтримка користувачів:

- Недолік: Недостатні ресурси для користувачів MercuryGate TMS можуть призводити до довгого освоєння системи.
- Покращення: Розвиток онлайн-курсів та розширення служби підтримки для полегшення вивчення системи.

Інтеграція:

- Недолік: Oracle SCM Cloud SAP потребує більш простої інтеграції з іншими системами.
- Покращення: Розробка стандартизованих інтерфейсів для полегшення обміну даними.

Масштабованість:

- Недолік: Integrated Business Planning (IBP) не завжди легко масштабується на великі підприємства.
- Покращення: Розробка додаткових функцій для ефективного масштабування системи.

Аналіз відгуків користувачів:

- Недолік: Manhattan Associates TMS може не завжди ефективно враховувати відгуки користувачів.
- Покращення: Запровадження систематичного аналізу відгуків для активного врахування побажань користувачів.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Аналіз і специфікація вимог до інформаційної системи

2.1.1. Бізнес-вимоги

Бізнес-вимоги – це специфікації, які, після реалізації, приносять цінність і визначають характеристики запропонованої системи для користувача. Визначення бізнес-вимог під час етапу розробки системи є критичним процесом, оскільки вони встановлюють основні цілі та очікування бізнесу від системи.

Бізнес-вимоги для інформаційної системи наведено далі:

1. Ефективне замовлення та обробка послуг: Система повинна забезпечувати швидке та ефективне замовлення послуг обслуговування грузових автомобілів клієнтами через веб-інтерфейс.
2. Точність та надійність даних: Бізнес вимагає, щоб система забезпечувала точність та надійність даних про замовлення, рух транспорту та статус обслуговування.
3. Гнучкість та масштабованість: Система повинна бути гнучкою та масштабованою, щоб вона могла легко адаптуватися до змін у вимогах бізнесу та розширюватися разом з ростом підприємства.
4. Інтеграція з іншими системами: Бізнес вимагає, щоб система була здатна інтегруватися з іншими внутрішніми та зовнішніми системами, такими як системи управління складами, платіжні системи та інші.
5. Забезпечення безпеки даних: З міркувань конфіденційності та захисту даних клієнтів і підприємства, система повинна мати ефективну систему безпеки для захисту від несанкціонованого доступу та витоку інформації.
6. Моніторинг та звітність: Система повинна надавати засоби для моніторингу та звітності про ефективність обслуговування, витрати на послуги та інші ключові показники,

7. Підтримка клієнтського сервісу: Бізнес вимагає, щоб система забезпечувала зручність та ефективність спілкування з клієнтами через різноманітні канали, включаючи онлайн-чат, електронну пошту та телефон.

8. Аналіз та оптимізація процесів: Система повинна надавати інструменти для аналізу процесів обслуговування, щоб ідентифікувати можливості для оптимізації та покращення ефективності.

9. Забезпечення зручного інтерфейсу користувача: Вимагається, щоб інтерфейс користувача був інтуїтивно зрозумілим та зручним у використанні для клієнтів, менеджерів та адміністраторів системи.

10. Підтримка мобільності: Бізнес вимагає, щоб система підтримувала доступ через мобільні пристрої, щоб клієнти та персонал могли зручно користуватися сервісом навіть під час пересування.

Далі наведено діаграми бізнес-вимог та бізнес-цілей кожної особи яка взаємодіє з системою що представлені на рис. 2.1 -2.6.

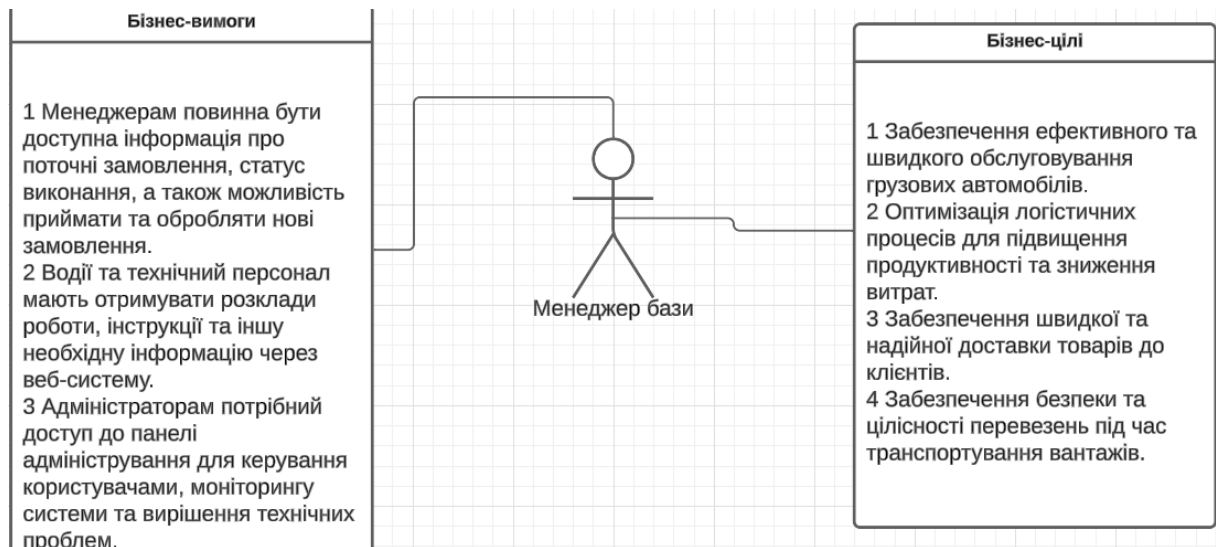


Рисунок 2.1 – Бізнес-цілі та бізнес-вимоги менеджера бази

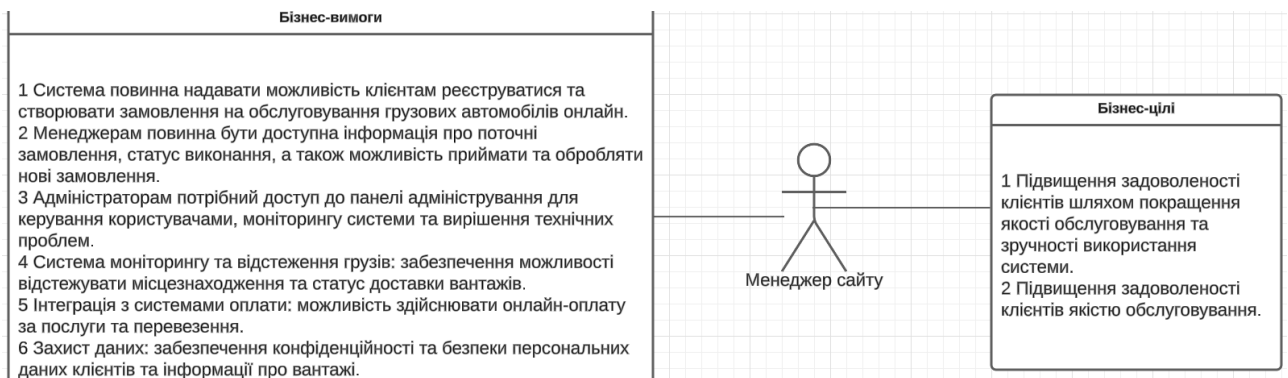


Рисунок 2.2 – Бізнес-цілі та бізнес-вимоги менеджера сайту

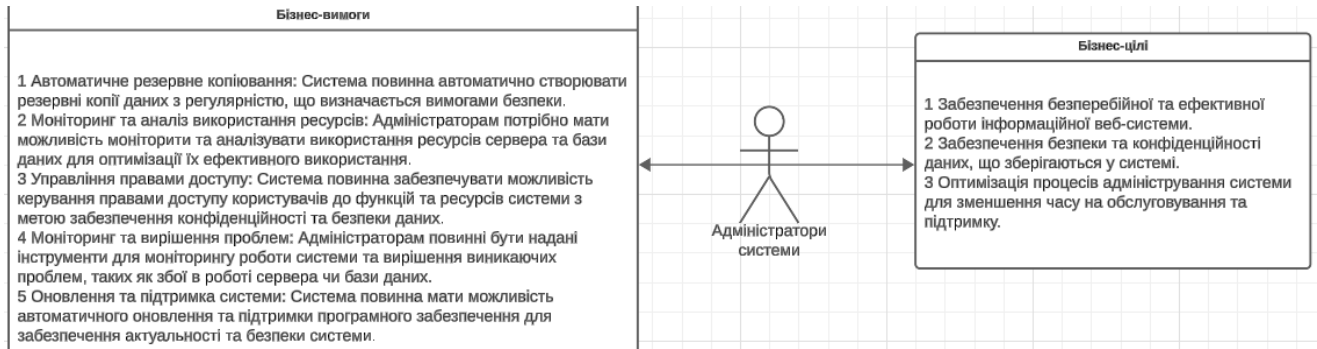


Рисунок 2.3 – Бізнес-цілі та бізнес-вимоги адміністратора системи

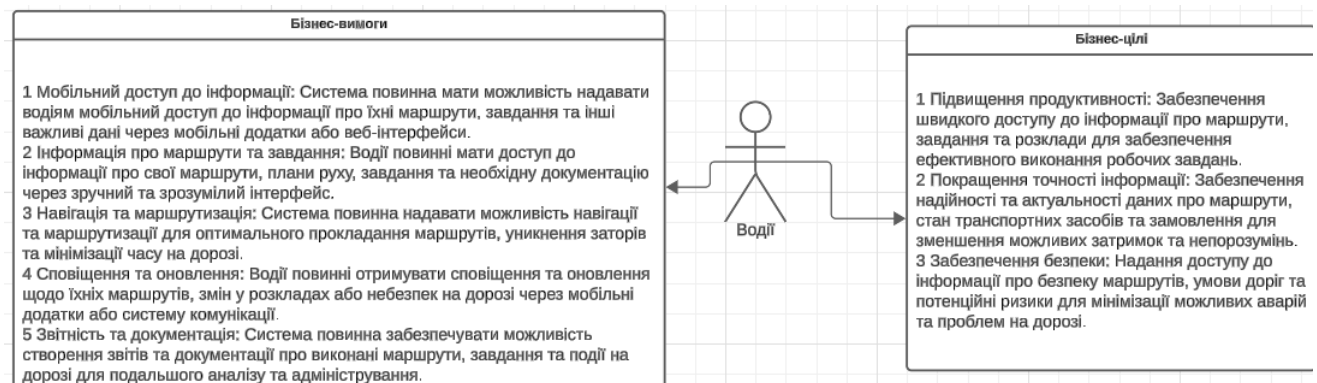


Рисунок 2.4 – Бізнес-цілі та бізнес-вимоги водіїв

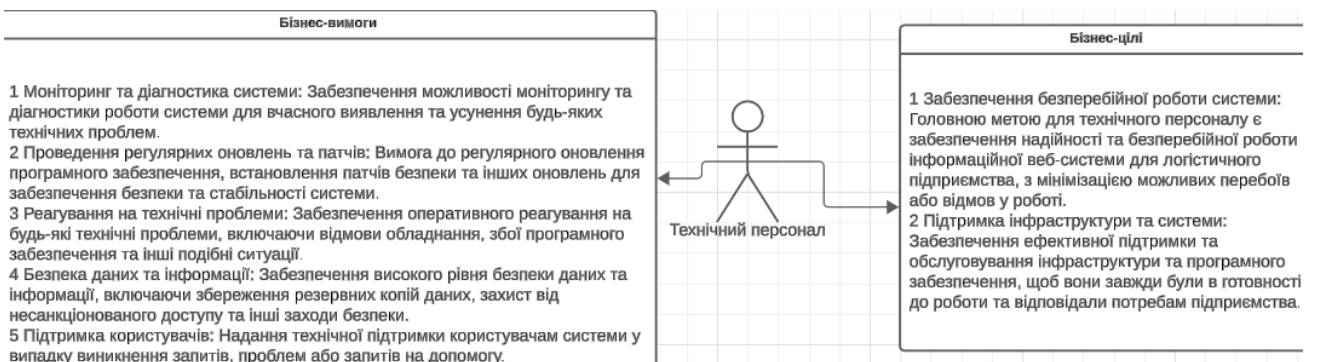


Рисунок 2.5 – Бізнес-цілі та бізнес-вимоги технічного персоналу

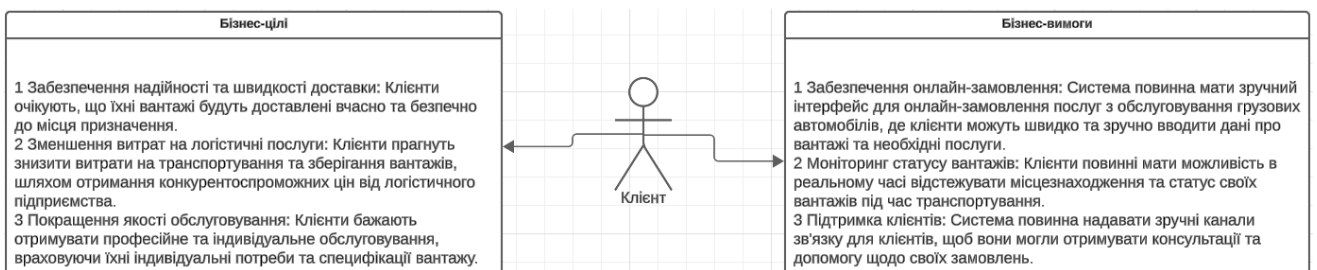


Рисунок 2.6 – Бізнес-цілі та бізнес-вимоги клієнтів

2.1.2. Нефункціональні вимоги

Нефункціональні вимоги – це обмеження, що визначають якісні характеристики системи.

Система повинна гарантувати високий рівень захисту конфіденційності, цілісності та доступності даних, зокрема, у випадку обробки конфіденційної інформації про клієнтів, вантажі та транспортні засоби. Швидкодія системи має бути на високому рівні, особливо під час обробки великого обсягу даних про рух вантажів та транспортних засобів. Система повинна бути легко масштабованою для підтримки зростаючого обсягу даних та користувацького трафіку з часом без втрати продуктивності. Крім того, важливо, щоб система коректно відображалася на різних пристроях (ПК, планшетах, мобільних телефонах) та підтримувала різні веб-браузери. Потрібна підтримка мовних версій для різних регіонів та країн, а також врахування міжнародних стандартів та правил. Інтерфейс користувача повинен бути інтуїтивно зрозумілим та простим для використання навіть для людей з обмеженими навичками у користуванні комп'ютером. Система повинна залишатися стійкою до навантажень та перебоїв у мережі, щоб забезпечити неперервну доступність для користувачів у всі часи. Крім того, необхідна можливість ведення журналу подій та аудиту для відстеження дій користувачів та забезпечення відповідності законодавству. Система повинна мати механізми які дозволять швидко виявити та виправити помилки, а також надання підтримки користувачам у разі виникнення запитань або проблем. Важливо мати механізми для систематичного резервного копіювання даних та можливість швидкого їх відновлення при втраті чи пошкодженні.

Загальні нефункціональні вимоги можна оглянути на рис. 2.7.

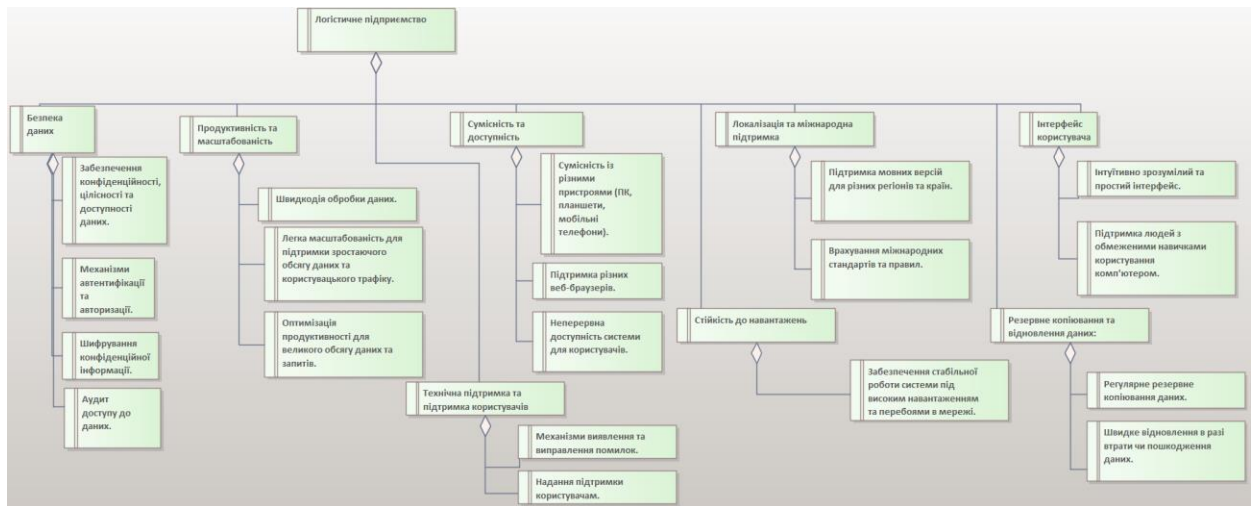


Рисунок 2.7 – Діаграма нефункціональних вимог

2.1.3. Функціональні вимоги

Функціональні вимоги – це описи того, як система має діяти. Вони визначають, що система повинна виконувати для задоволення потреб або очікувань користувача. Функціональні вимоги можна розглядати як функції, які помічає користувач. В них наявна різниця від нефункціональних вимог, які визначають внутрішні аспекти роботи системи.

Система повинна надавати можливість користувачам реєструватися, включаючи клієнтів, менеджерів та адміністраторів. Клієнти повинні мати змогу створювати, переглядати, редагувати та видаляти замовлення на обслуговування вантажних автомобілів. Система повинна також надавати можливість відстеження руху транспортних засобів, що перебувають у процесі обслуговування, включаючи відображення маршруту та статусу доставки. Менеджерам необхідно мати можливість призначати та керувати роботою водіїв та технічного персоналу. Система також повинна включати календар та графік роботи, що відображає доступні та зайняті часові інтервали для обслуговування вантажних автомобілів. Облік матеріалів та запасів також має бути реалізованим в системі, включаючи контроль рівня запасів. Крім того, система повинна забезпечувати можливість для генерування різноманітних звітів та аналітичних даних про роботу підприємства, такі як звіти про витрати та ефективність

обслуговування. Система повинна також надсилати сповіщення клієнтам щодо статусу їхніх замовлень та менеджерам про важливі події або зміни в графіку роботи. Щодо оплати, система має підтримувати можливість онлайн-оплати послуг та інтеграцію з різними платіжними системами. Не менш важливою є можливість клієнтів залишати відгуки та оцінки щодо якості обслуговування, що допомагатиме вдосконалювати процеси підприємства. Загальні функціональні вимоги можна оглянути на рис. 2.8.

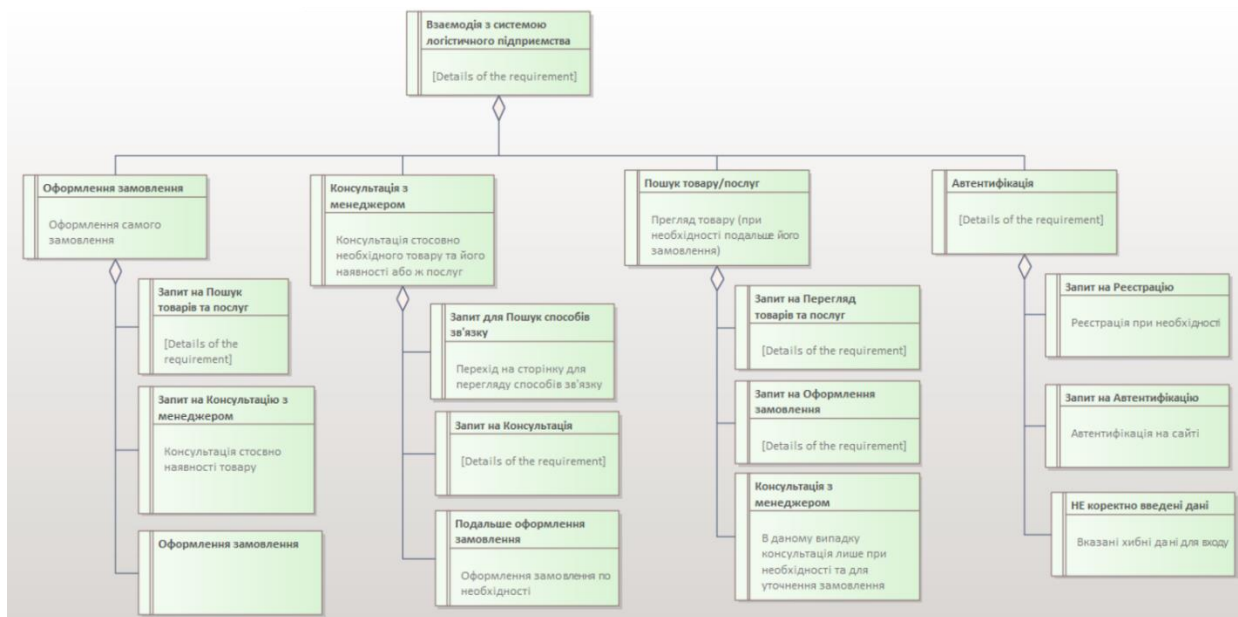


Рисунок 2.8 – Діаграма функціональних вимог

2.2. Постановка та алгоритм розв'язання задачі

2.2.1. Постановка задачі

Важливим аспектом та складовою формування загальної концепції веб-системи для логістичного підприємства є інформаційна модель. Під час розробки цієї моделі важливо розуміти майбутні взаємозв'язки між даними, які входять у систему, та кінцевий результат, оброблений самою системою. Мета нашого проекту полягає в наданні користувачам більш розширених можливостей управління та оптимізації логістичних процесів, забезпечуючи якісне та

ефективне використання системи. Інформаційну модель задачі можна оглянути на рис. 2.9.

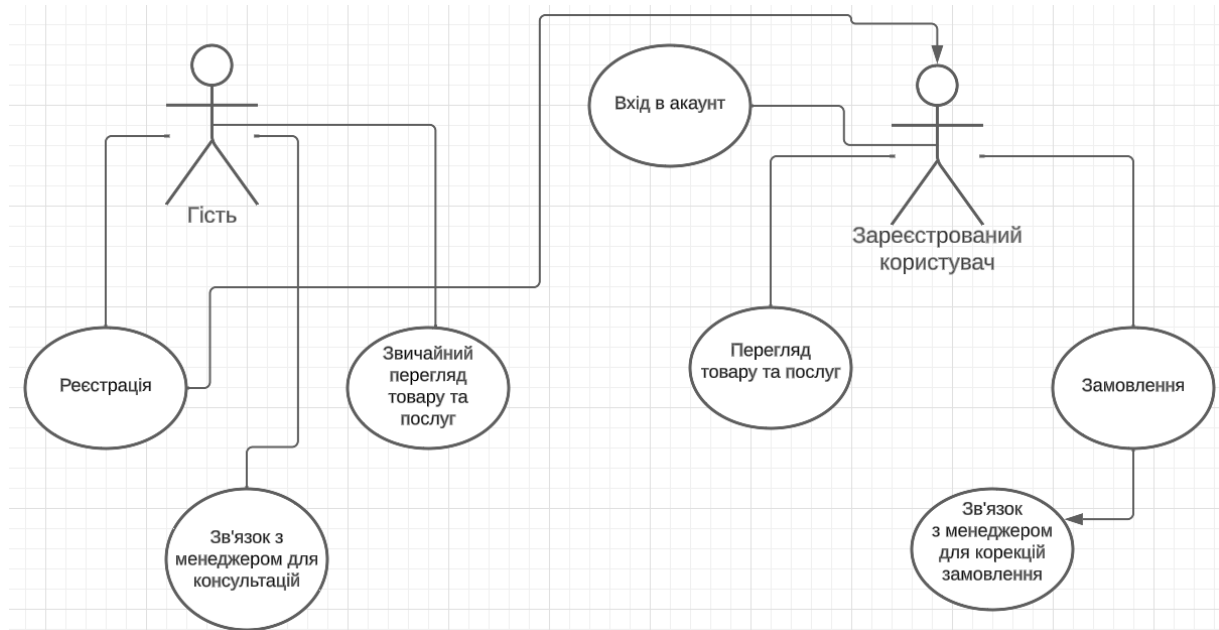


Рисунок 2.9 – Інформаційна модель задачі

2.2.2. Вихідна інформація

Перелік і опис вихідних повідомлень наведено у табл. 2.1.

Таблиця 2.1. – Перелік та опис вихідних повідомлень

Назва вихідного повідомлення	Ідентифікатор	Форма подання і вимоги до неї	Періодичність звітування	Термін звітування і допустимий час затримки	Отримувач інформації
Повідомлення про створення нового замовлення	MES-ORDER	Електронний лист, SMS, сповіщення	Щоденно	Термін видання - негайно, допустимий час затримки – 5 хвилин	Клієнт, менеджер замовлень
Повідомлення про статус доставки	MES-DELIVERY	Електронний лист, SMS	Щоденно	Термін видання - згідно графіку доставки, допустимий час затримки – 15 хвилин	Клієнт, відділ логістики

Сповіщення про відгук клієнта	MES-REVIEW	Електронний лист	Після кожного покупцевого відгуку	Термін видання - негайно, допустимий час затримки – 10 хвилин	Адміністратор, інші користувачі
Повідомлення про акції та знижки	MES-PROMO	Електронний лист, сповіщення	Раз на тиждень	Термін видання - згідно графіку акцій, допустимий час затримки – 30 хвилин	Підписники, клієнти
Сповіщення про найближчі заходи та події	MES-EVENT	Електронний лист, SMS, сповіщення	Місячно	Термін видання - перед подією, допустимий час затримки – 1 година	Клієнти, підписники
Повідомлення про надходження нових товарів	MES-NEW-PROD	Електронний лист, сповіщення	Раз на тиждень	Термін видання - негайно при надходженні, допустимий час затримки – 15 хвилин	Клієнти, менеджери закупівель

Система логістичного підприємства використовує реляційну базу даних для зберігання та управління інформацією про товари та клієнтів. Вона генерує різноманітні вихідні повідомлення, які є повідомленнями про наявний стан бази даних або передачі результатів операцій.

Вихідні повідомлення представляють собою інформаційні повідомлення, що формуються системою для взаємодії з іншими завданнями або для відображення на екрані за запитом користувача. Ці повідомлення можуть мати різноманітні типи, залежно від призначення та форми подання. Основні категорії вихідних повідомлень включають:

Машинограми для друку: Це документи, які автоматично формуються системою та містять інформацію про товари, замовлення, резервування та інші події. Ці документи служать як довідковий матеріал для звітності та аналізу

роботи системи.

Вихідні файли з результатами завдань: Це файли, які містять результати виконання досліджуваних завдань та можуть бути корисними для використання в інших системах для подальшої обробки. Наприклад, це можуть бути файли з даними про розподіл товарів, які використовуються для бухгалтерського обліку.

2.2.3. Вхідна інформація

Створення облікового запису клієнта - це процес, під час якого новий клієнт подає необхідну інформацію для реєстрації в системі управління автомобільним концерном. Це повідомлення передає дані, введені клієнтом, в базу даних для подальшого формування особистого облікового запису.

Оновлення інформації про клієнта - це дія, коли клієнт системи управління автомобільним концерном оновлює свої особисті дані, такі як контактна інформація, адреса тощо. Це повідомлення оновлює дані про клієнта в базі даних, забезпечуючи актуальність інформації.

Створення нових категорій товарів - це дія, коли адміністратор чи відповідальна особа системи визначають нові категорії для класифікації товарів, наприклад, "Легкові автомобілі", "Вантажівки", "Мотоцикли" тощо. Це повідомлення додає нові категорії до бази даних, що спрощує класифікацію товарів та полегшує їх пошук.

Додавання нової продукції - це дія, коли адміністратор чи відповідальна особа системи додають новий товар, такий як автомобіль, запчастина чи аксесуар. Це повідомлення додає новий запис про продукт до бази даних, зберігаючи його характеристики, такі як модель, ціна, наявність тощо.

Придбання продукції - це дія, коли клієнт здійснює покупку товару і отримує доступ до інформації про своє замовлення. Це повідомлення оновлює дані про покупку в базі даних, реєструючи придбання та надаючи клієнту відповідний доступ до інформації про його замовлення.

Створення нових облікових записів менеджерів - це дія, коли адміністратор

чи старший менеджер системи створюють облікові записи для нових менеджерів, які відповідають за взаємодію з клієнтами, ведення продажів чи інші аспекти управління. Це повідомлення додає нові облікові записи менеджерів до бази даних разом із доступом до системи управління та для початку виконання своїх обов'язків.

Перелік і опис усіх вхідних повідомлень наведено в табл. 2.2.

Таблиця 2.2 – Перелік та опис вхідних повідомлень

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
Створення облікового запису клієнта	ACC-CRE	Веб-форма	Щоденно	Клієнт
Оновлення інформації про клієнта	INFO-UPD	Веб-форма	Щоденно	Клієнт
Створення нових категорій товарів	CAT-CRE	Веб-форма	Раз на місяць	Менеджер
Додавання нового товару	PROD-ADD	Веб-форма	Щотижня	Менеджер
Здійснення покупки	PURCHASE	Веб-форма	Щоденно	Клієнт
Створення нових облікових записів менеджерів	MNG-CRE	Веб-форма	Раз на місяць	Адміністратор
Запит інформації про послугу	SERV-REQ	Веб-форма	Щоденно	Клієнт
Створення нової послуги	SERV-CRE	Веб-форма	Раз на місяць	Менеджер
Оновлення тарифів на послуги	TARIFF-UPD	Веб-форма	Щоквартально	Адміністратор
Запит інформації про тарифи	TARIFF-REQ	Веб-форма	Щоденно	Клієнт
Додавання нових транспортних засобів	VEHICLE-ADD	Веб-форма	Раз на місяць	Менеджер

Оновлення інформації про транспортні засоби	VEHICLE-UPD	Веб-форма	Щоквартально	Менеджер
--	-------------	-----------	--------------	----------

2.2.4. Алгоритм розв'язання задачі

Математичний опис.

Індекс задоволеності користувача (IS):

$$IS = \frac{\text{Повна кількість}}{\text{Загальна кількість користувачів}} \times 100$$

Індекс ефективності процесів (IEP):

$$IEP = \frac{\text{Сума виконаних процесів}}{\text{Загальна кількість процесів}} \times 100$$

Індекс доступності інформації (IAI):

$$IAI = \frac{\text{Частота успішних}}{\text{Загальна кількість запитів}} \times 100$$

Індекс автоматизації (IA):

$$IA = \frac{\text{Кількість автоматизованих процесів}}{\text{Загальна кількість процесів}} \times 100$$

Індекс забезпечення безпеки (IS):

$$IS = \frac{\text{Кількість безпечних операцій}}{\text{Загальна кількість операцій}} \times 100$$

Рисунок 2.10 – Математичні формули

Алгоритм розв'язання задачі. Для демонстрації роботи системи було розроблено діаграму, яка зображують алгоритми дій замовлення користувачем та його взаємодію із системою рис. 2.11.

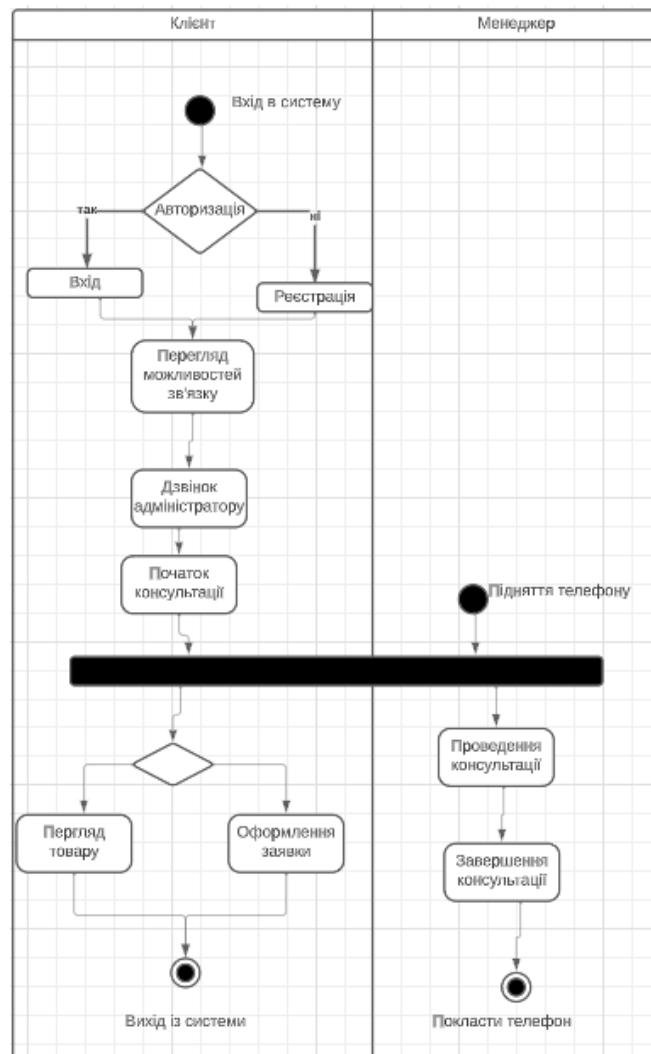


Рисунок 2.11 – Алгоритм замовлення користувачем

Під час входу в систему користувач може обрати: увійти за допомогою вже існуючого акаунту, при цьому йому будуть надані права згідно з роллю, або зареєструвати новий акаунт. Враховано можливість введення некоректних даних клієнтом під час входу. На рис.2.12 представлена діаграма процесу авторизації.

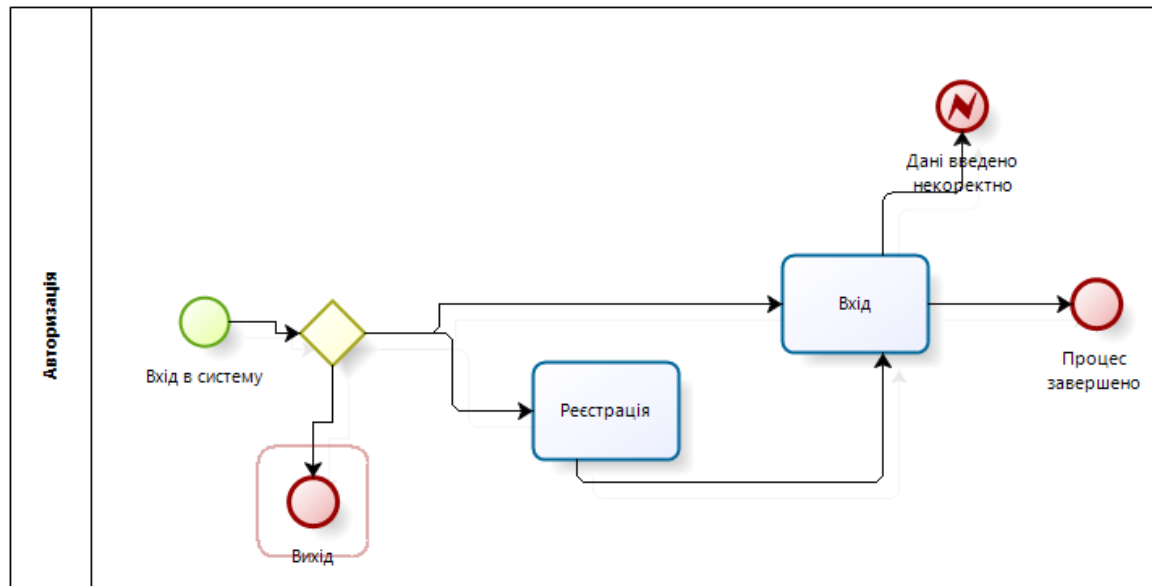


Рисунок 2.12 – Процес авторизації

Коли клієнт вже оформлює замовлення під час зв'язку з менеджером, він може одразу перейти до оформлення замовлення або повернутися до замовлення необхідного товару чи рудагування певних послуг. Після чого буде прийматись рішення з оформленням замовлення в даний момент або ж можливо відкласти його. На рис.2.13 представлена діаграма процесу замовлення.

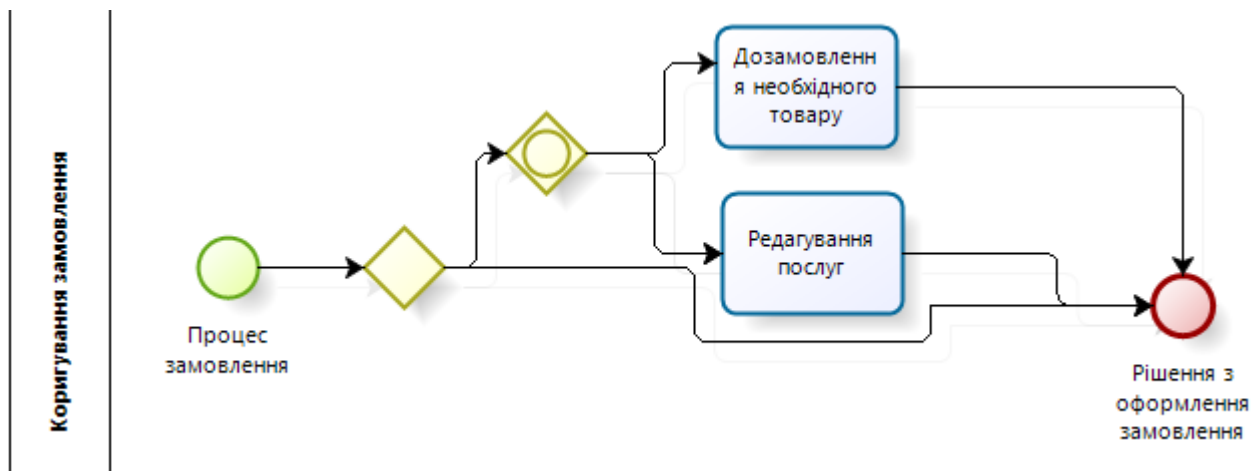


Рисунок 2.13 – Процес замовлення клієнтом

В діаграмі на рис. 2.14 показано два процеси: клієнт та адміністратор. Процес виконання роботи адміністратором поділений на дві частини: менеджер і адміністратор. А для процесу клієнта є два під-процеси: авторизація та вибір товару для покупки. Процес покупки є повторюваним. В діаграмі ми маємо певні послідовні дії, бази даних, та приєднання їх до процесів, в яких вони використовуються.

Після успішної замовлення клієнт має змогу відредагувати разом з менеджером замовлення. Після звернення менеджер спробує вирішити проблеми які могли виникнути, за допомогою внесення змін до бази даних, але може виникнути проблема із замовленнями для клієнта певних запчастин. На рис.2.14 представлена загальна діаграма взаємодії системи.

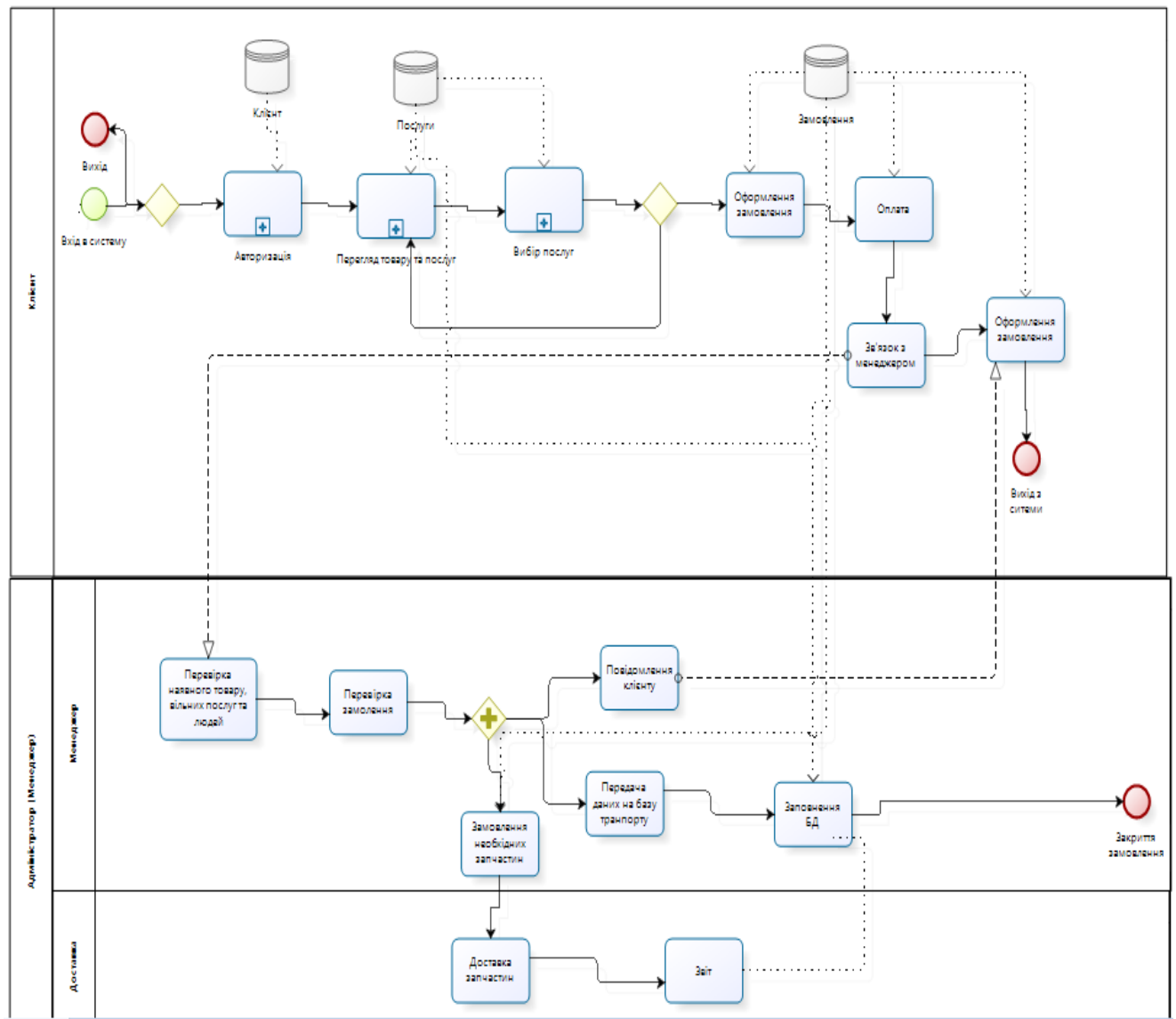


Рисунок 2.14 – Діаграма взаємодії

2.3. Моделювання інформаційної системи

Моделювання поведінки системи в контексті логістичного підприємства може бути важливим аспектом для забезпечення ефективного та оптимального функціонування всіх процесів.

Ось розширене пояснення моделі поведінки системи:

- Створення та реєстрація замовлень: система дозволяє клієнтам створювати та реєструвати замовлення на обслуговування вантажних автомобілів; клієнти можуть вказувати деталі замовлення, такі як тип вантажу, маршрут, дату та час доставки.
- Управління роботою та розподіл завдань: менеджери мають можливість призначати та керувати роботою водіїв та технічного персоналу для виконання замовлень; система автоматично розподіляє завдання серед водіїв та технічного персоналу на основі їхньої доступності та робочого графіку.
- Моніторинг та відстеження руху транспорту: система надає інтерфейс для моніторингу та відстеження руху транспортних засобів, які здійснюють доставку вантажів; клієнти та менеджери можуть переглядати маршрут транспортного засобу та відстежувати статус доставки в реальному часі.
- Повідомлення та сповіщення: система автоматично надсилає сповіщення клієнтам про статус їхніх замовлень, такі як підтвердження замовлення, підтвердження доставки або затримка; менеджери отримують сповіщення про важливі події або зміни в графіку роботи.
- Генерація звітів та аналітика: система забезпечує можливість генерувати різноманітні звіти та аналітичні дані про роботу підприємства; звіти включають витрати, ефективність обслуговування, прогрес виконання завдань та інші важливі показники.
- Облік та управління запасами: система веде облік матеріалів, необхідних для обслуговування автомобілів, та контролює рівень запасів; автоматичне повідомлення про необхідність поповнення запасів при досягненні

мінімального рівня.

На діаграмі прецедентів системи (рис. 2.15) зображено діаграму в якій наведені можливості зареєстрованого користувача та гостя системи. Показано лише дві дії, перегляд товару або ж його оформлення у замовлення, бо є можливість бути гостем системи або ж зареєстрованим користувачем. Якщо ж просто гість то можна лише переглянути товар чи послуги та окремо поспілкуватись з менеджером, або ж як зареєстрований користувач може оформити замовлення та потім також провести консультацію для уточнення в замовленні або його переробці. зображено на рис. 2.15.

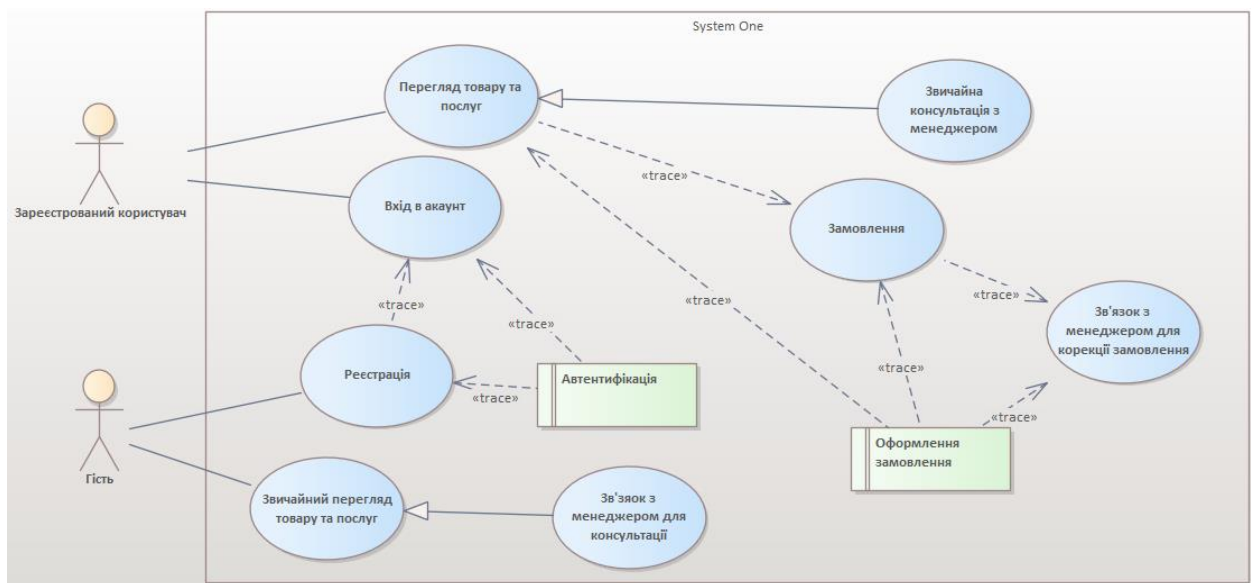


Рисунок 2.15 – UML діаграма прецедентів (UseCase)

На діаграмі дій замовлення користувачем (рис. 2.16) зображено алгоритм дій користувача при замовленні товару чи послуг. Спочатку йде перегляд товару чи послуг, потім його вибір і після цього замовлення. Замовлення йде до БД і після цього з користувачем зв'язується менеджер для корекції замовлення, якщо допомога менеджера була потрібна взагалі. Тому від менеджера замовлення йде до бази для початку робіт, а якщо ж все було проведено без менеджера то замовлення йде на пряму до технічної бази.

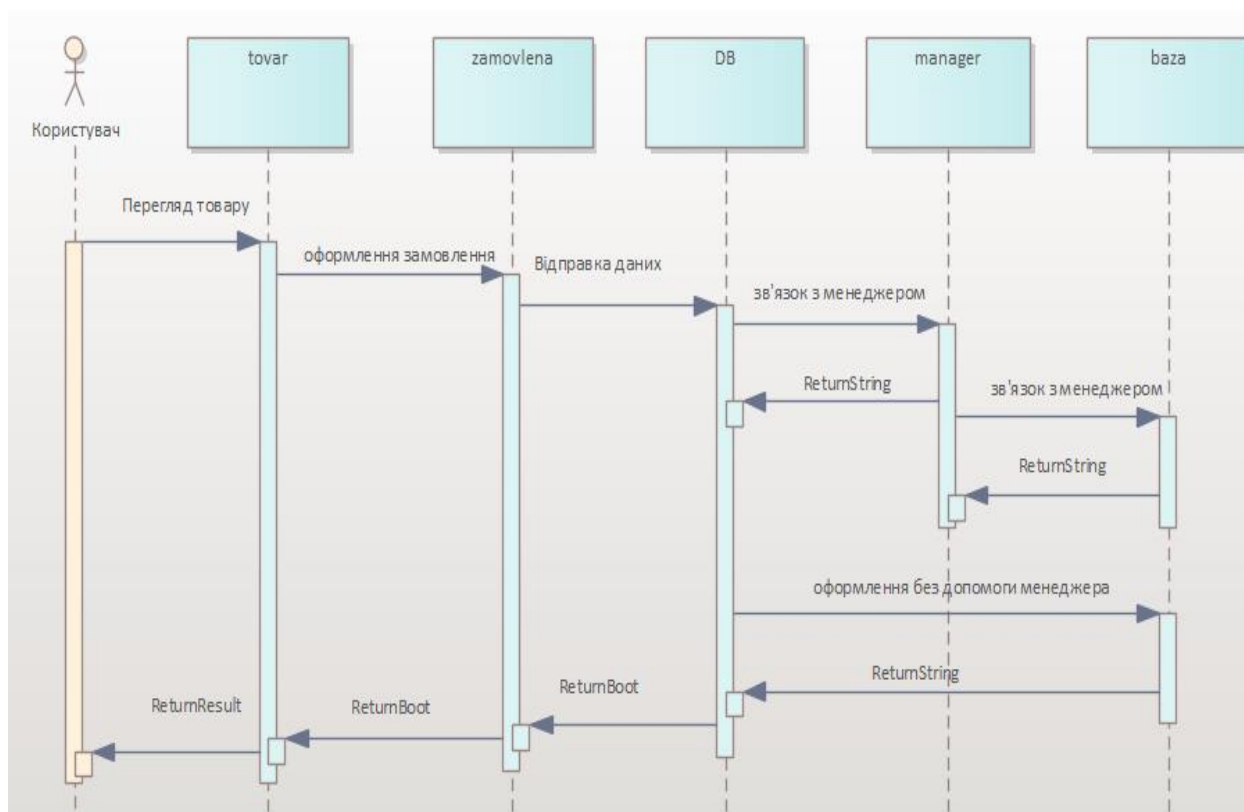


Рисунок 2.16 – Діаграма дій замовлення користувачем

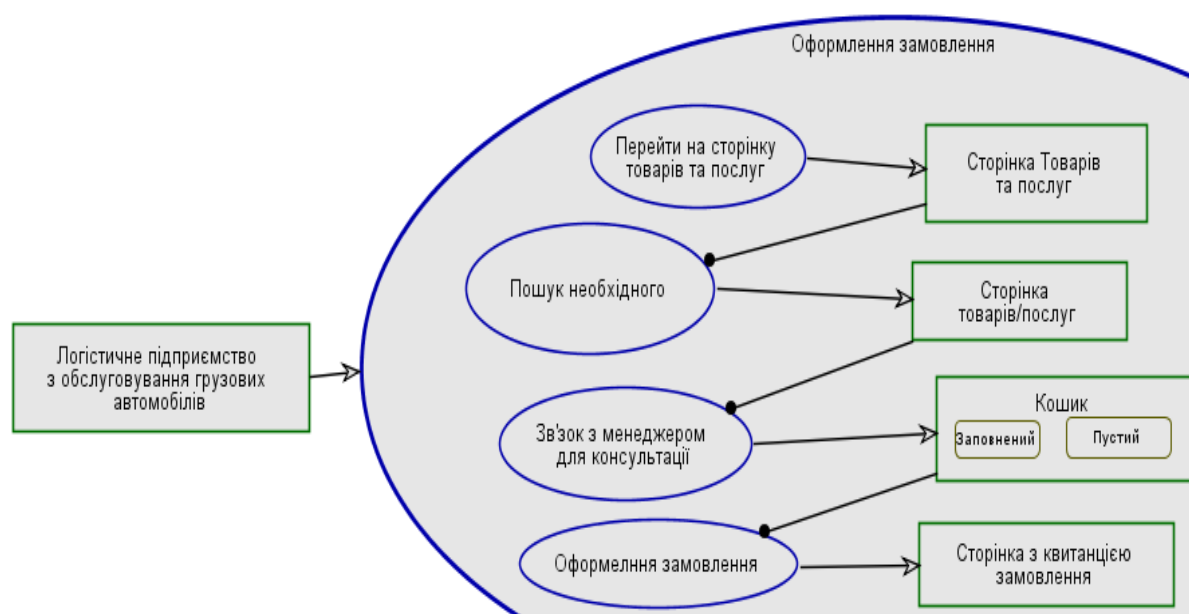


Рисунок 2.17 – Перехід користувачів по сторінках



Рисунок 2.18 – Перехід користувачів по сторінках при консультації з менеджером

На діаграмі дій входу в систему (рис. 2.19) зображено алгоритм дій користувача при вході в свій акаунт. Спочатку йде перехід на сайт, а потім вхід в саму систему. Відправка даних йде до бази даних, а також йде перевірка паролю користувача. Після чого користувачу надається доступ в систему через його акаунт.

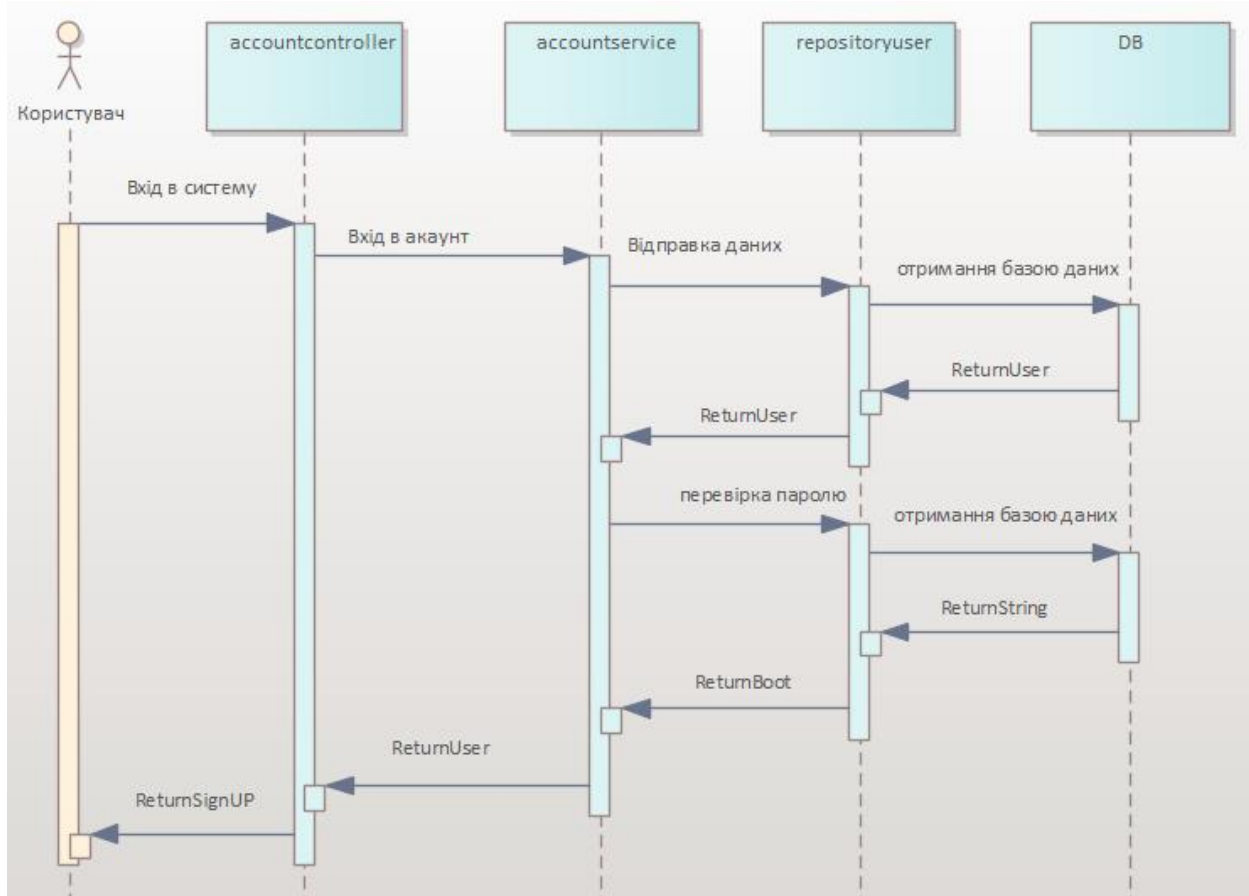


Рисунок 2.19 – Діаграма дій входу в систему

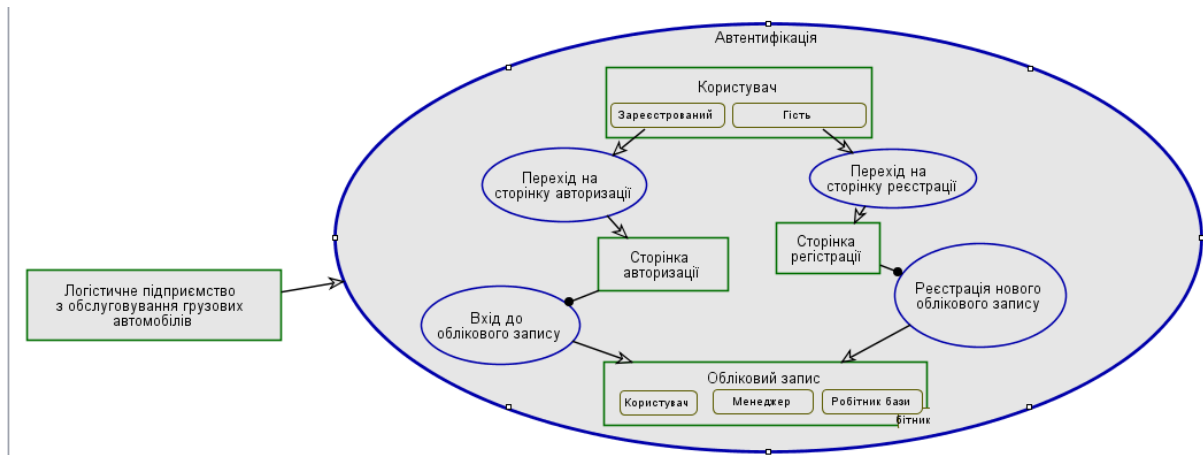


Рисунок 2.20 – Перехід користувача по сторінках

На діаграмі дій реєстрації в систему (рис. 2.21) зображено алгоритм дій реєстрації нового користувача в системі. Спочатку йде перехід на сайт, а потім реєстрація в саму систему. Відправка даних про створеного користувача йде до бази даних. Після чого користувачу надається доступ в систему через його новий акаунт. Також є можливість перегляду сайту без реєстрації.

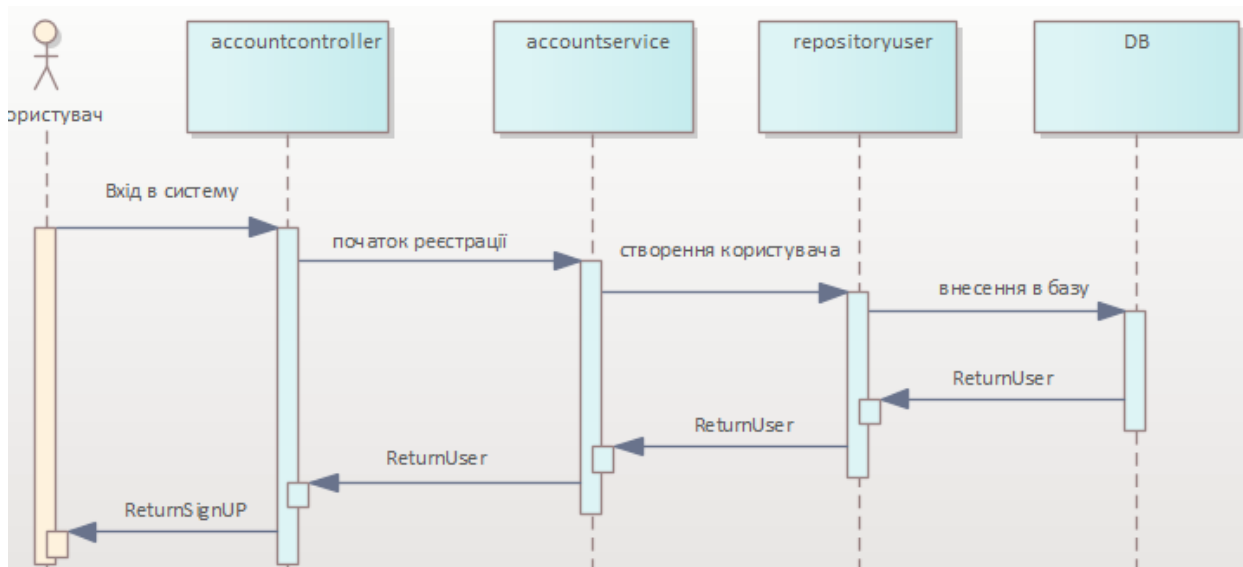


Рисунок 2.21 – Діаграма дій реєстрації в системі

Моделювання структури системи логістичного підприємства включає в себе опис основних компонентів та їх взаємодію для забезпечення ефективного функціонування всього підприємства.

Клієнтський інтерфейс - взаємодія з системою через веб-інтерфейс або

мобільний додаток. Цей інтерфейс надає клієнтам можливість реєстрації, створення та перегляду замовлень, відстеження статусу доставки та надання відгуків та оцінок.

Через панель адміністратора самі ж адміністратори системи керують основними налаштуваннями та мають доступ до управління користувачами, замовленнями та іншими важливими аспектами системи через панель адміністратора.

За допомогою менеджерського інтерфейсу менеджери взаємодіють з системою через спеціальний інтерфейс, який дозволяє їм керувати роботою водіїв та технічного персоналу, розподіляти завдання та відстежувати робочий процес.

Централізована база даних зберігає інформацію про користувачів, замовлення, транспортні засоби, матеріали, запаси та інші важливі дані. База даних забезпечує доступ до інформації для всіх компонентів системи та забезпечує консистенцію та цілісність даних.

Система моніторингу та відстеження - відстежує рух транспортних засобів, відображає їхні маршрути та статус доставки. Вона інтегрована з головною системою та забезпечує актуальну інформацію про рух транспорту.

Система сповіщень автоматично надсилає сповіщення клієнтам та менеджерам про важливі події, такі як підтвердження замовлення, затримки в доставці або зміни в графіку роботи.

Система інтеграції з платіжними системами забезпечує можливість онлайн-оплати послуг та інтеграцію з різними платіжними системами для зручності клієнтів.

Діаграма класів в системі моделює структуру системи з точки зору об'єктів, які в ній присутні, їхні взаємовідносини та властивості. Вона описує класи об'єктів у системі та їхні взаємозв'язки. В діаграмі класів можуть бути відображені наступні елементи:

Класи які представляють основні сутності системи, такі як користувачі, замовлення, транспортні засоби тощо.

Атрибути - властивості класів, які визначають їхні характеристики. Наприклад, клас "Користувач" може мати атрибути "ім'я", "прізвище", "електронна пошта" тощо.

Методи як операції, котрі можуть бути виконані над об'єктами класу. Наприклад, клас "Замовлення" може мати методи "створити", "редагувати", "видалити" тощо.

Взаємозв'язки між класами показують, як класи взаємодіють один з одним. Наприклад, взаємозв'язок між класами "Клієнт" і "Замовлення" може показувати, що в кожного клієнта може бути наявним багато замовлень.

Спадкування або ж наслідування вказує, що один клас успадковує властивості та методи іншого класу.

Інтерфейси визначають контракти, які клас повинен реалізувати. Наприклад, інтерфейс "Платіжна система" може визначати методи для обробки платежів.

Далі наведено діаграму класів системи рис. 2.22 та діаграму структури системи рис 2.23.

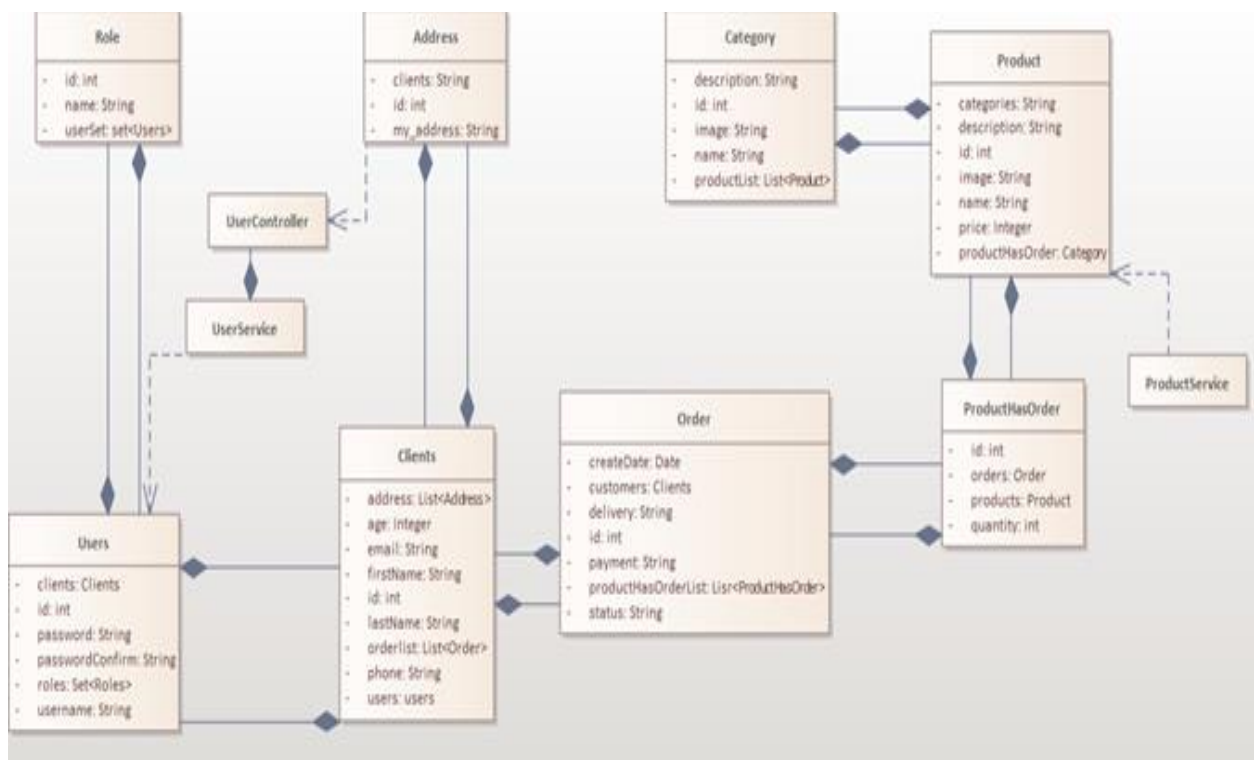


Рисунок 2.22 – Діаграма класів системи

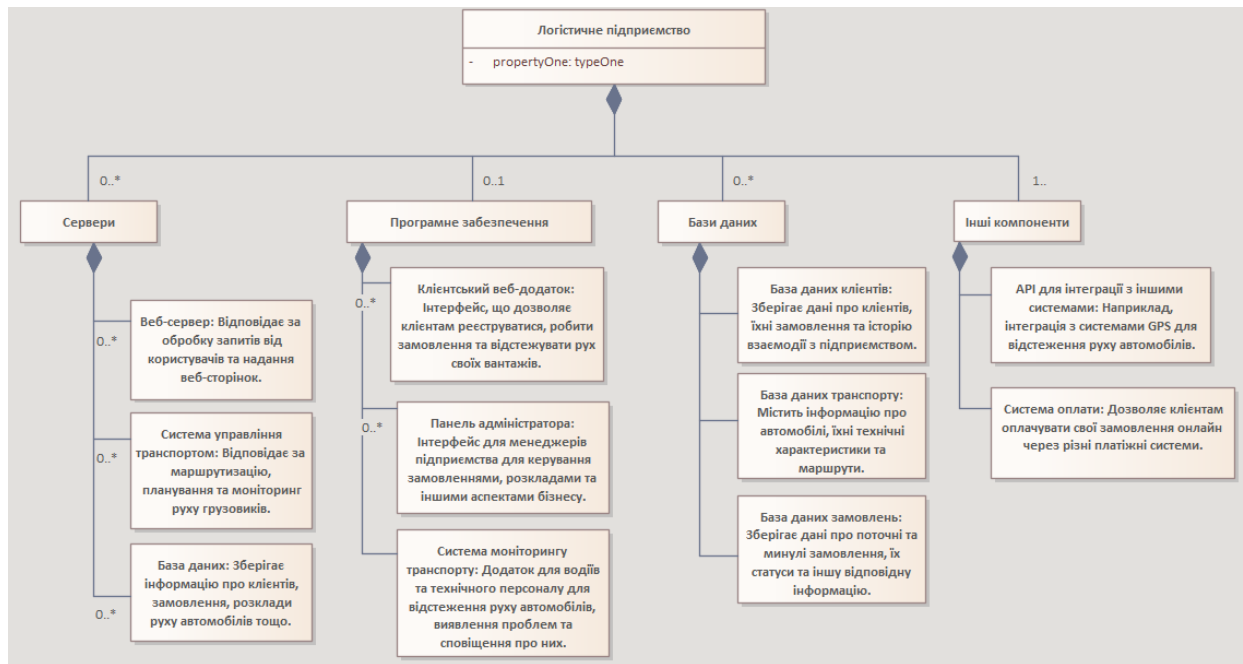


Рисунок 2.23 – Діаграма структури системи

На рисунку 2.24 зображено приклад роботи API компоненту ІС. На цій діаграмі показано фрагмент залежності компонентів системи, для оформлення замовлення. API приймає запит з вказаним товаром і це все надсилається в базу. Так само під час замовлення.

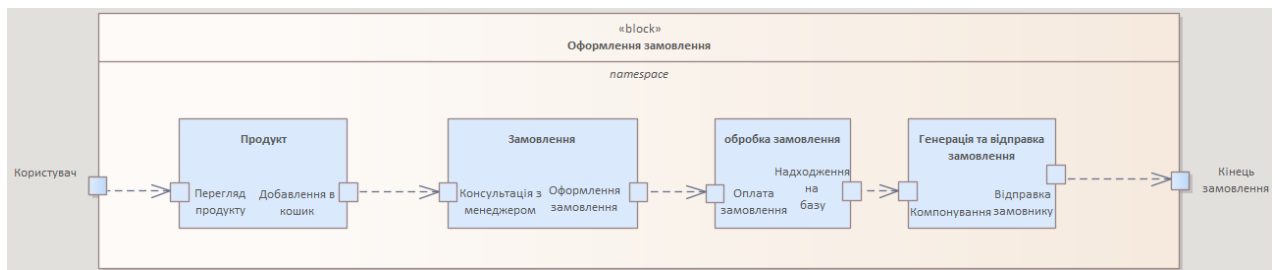


Рисунок 2.24 – Діаграма внутрішніх блоків

Розподіл вимог за компонентами системи.

Для наочності, які вимоги належать до яких прецедентів і на якій екранній формі вони повинні бути реалізовані, була створена діаграма трасування на рис. 2.25.

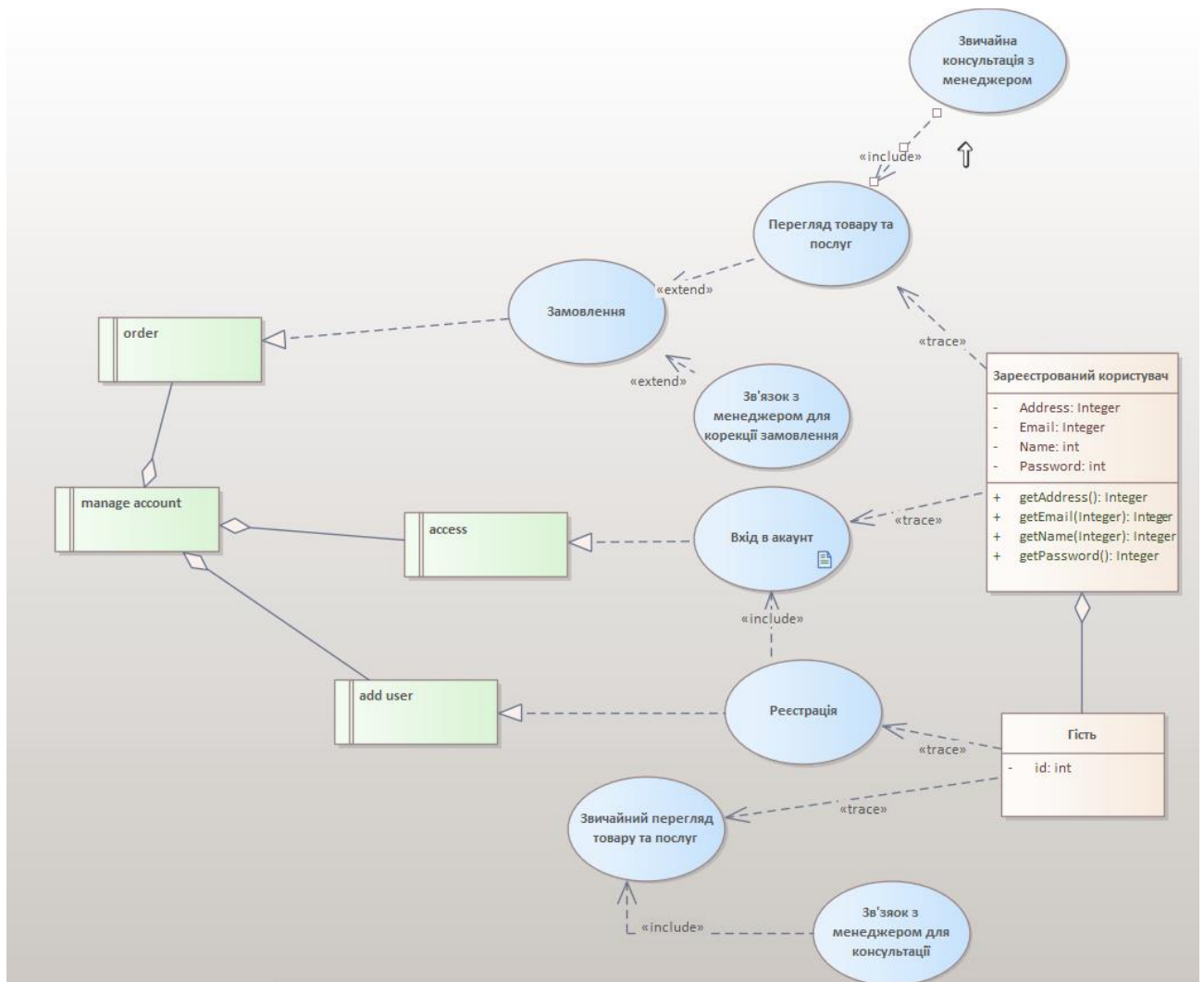


Рисунок 2.25 – Діаграма трасування в системі

Трасування вимог - це процес відстеження вимог від їх виникнення до реалізації та тестування. У системі логістичного підприємства трасування вимог має декілька важливих цілей:

Забезпечення відповідності допомагає переконатися, що кожен елемент програмного забезпечення відповідає конкретним вимогам, встановленим у специфікації вимог. Це дозволяє забезпечити, що функціональність системи відповідає очікуванням користувачів і бізнес-потребам.

Виявлення пропущених вимог допомагає виявити вимоги, які можуть бути пропущені або не реалізовані в системі. Це дозволяє уникнути ситуацій, коли деякі важливі аспекти функціональності не були враховані при проектуванні та розробці.

[illegible]

Рисунок 2.27 – Таблиця трасування в системі

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1. Інформаційне забезпечення

Далі наведено склад та структуру інформаційного забезпечення системи для логістичного підприємства з обслуговування грузових автомобілів. Інформаційне забезпечення цієї системи включає наступні компоненти:

- Серверна інфраструктура складає це самі ж сервери, які забезпечують та є необхідними для зберігання та обробки даних, та управління користувачами з наданням доступу до системи через мережу.
- База даних містить інформація про транспортні засоби, вантажі, маршрути, розклади обслуговування, дані водіїв та інші важливі дані підприємства.
- Клієнтські додатки призначені для взаємодії користувачів з системою. Вони забезпечують можливість перегляду та редагування даних, створення заявок на обслуговування автомобілів, спілкування зі службою підтримки та інші функції.
- Мережеві пристрої забезпечують зв'язок між сервером і клієнтськими додатками, для передавання даних через мережу та взаємодіяти з транспортними засобами на дорозі.
- Основні принципи, які керують організацією інформаційного забезпечення системи для логістичного підприємства з обслуговування грузових автомобілів, включають:
 - Принцип безпеки – це забезпечення конфіденційності, цілісності та доступності даних системи, використання механізмів автентифікації, авторизації та шифрування для захисту від несанкціонованого доступу.
 - Принцип надійності стабільної роботи системи, виявлення та виправлення помилок, резервне копіювання даних для уникнення їх втрати та

забезпечення безперебійності сервісу для клієнтів.

– Принцип зручності використання, тобто розробка зрозумілого інтерфейсу для користувачів, із забезпеченням зручності в навігаційному середовищі та функцій для зручного та ефективного використання системи.

Структура бази даних та її таблиць. Далі наведено структуру таблиць бази даних можна побачити у таб. 3.1 – 3.9.

Таблиця 3.1 – «Користувачі»

Найменування	Ідентифікатор у програмі	Умовне позначення у формулах	Формат	Бізнес-правила	Логічні чи семантичні зв'язки	Найменування
Ім'я користувача	Username	UN	Varchar	Довжина: 0-45	Первинний ключ, Значення: не нуль, Обов'язкове поле, Індексне поле	Ім'я користувача
Пароль	Password	UP	Varchar	Довжина: 225	Значення: не нуль, Обов'язкове поле	Пароль
Ім'я	FirstName	FN	Varchar	Довжина: 45	-	Ім'я
Прізвище	LastName	LN	Varchar	Довжина: 45	-	Прізвище
Електронна пошта	Email	UE	Varchar	Довжина: 45	-	Електронна пошта
Посада	Position	POS	Varchar	Довжина: 45	-	Посада
Телефон	Phone	PH	Varchar	Довжина: 20	-	Телефон
Адреса	Address	AD	Varchar	Довжина: 255	-	Адреса
Роль	Role	RL	Varchar	Довжина: 45	-	Роль

Таблиця 3.2 – «Замовлення»

Найменування	Ідентифікатор у програмі	Умовне позначення у формулах	Формат	Бізнес-правила	Логічні чи семантичні зв'язки
ID замовлення	OrderID	OID	Integer	Первинний ключ	Значення: не нуль, Обов'язкове поле, Індексне поле
ID клієнта	CustomerID	CID	Integer	-	Індексне поле
ID товару	ProductID	PID	Integer	-	Індексне поле
ID послуги	ServiceID	SID	Integer	-	Індексне поле
Кількість товару	Quantity	QTY	Integer	-	-
Кількість послуги	ServiceQuantity	SQTY	Integer	-	-
Ціна товару	Price	PR	Decimal	-	-
Ціна послуги	ServicePrice	SPR	Decimal	-	-
Дата замовлення	OrderDate	OD	Date	-	Індексне поле
Статус	Status	ST	Varchar	Довжина: 45	-
Тариф на перевезення	TransportRate	TR	Decimal		
Тариф на доставку	DeliveryRate	DR	Decimal		

Таблиця 3.3 – «Товари»

Найменування	Ідентифікатор у програмі	Умовне позначення у формулах	Формат	Бізнес-правила	Логічні чи семантичні зв'язки
--------------	--------------------------	------------------------------	--------	----------------	-------------------------------

ID товару	ProductID	PID	Integer	Первинний ключ	Значення: не нуль, Обов'язкове поле, Індексне поле
Найменування	Name	NM	Varchar	Довжина: 255	-
Опис	Description	DESC	Text	-	-
Ціна	Price	PR	Decimal	-	-
Кількість на складі	Quantity	QTY	Integer	-	-
Виробник	Manufacturer	MFR	Varchar	Довжина: 100	-
Модель	Model	MDL	Varchar	Довжина: 100	-
Рік випуску	Year	YR	Integer	-	-
Країна виробництва	Country	CNT	Varchar	Довжина: 100	-
Ціна за транспортуванн я	DeliveryRate	DR	Decimal		

Таблиця 3.4 – «Послуги»

Найменування послуги	Ідентифікатору програмі	Умовне позначення у формулах	Формат	Бізнес-правила	Логічні чи семантичні зв'язки
ID послуги	ServiceID	SID	Integer	Первинний ключ	Значення: не нуль, Обов'язкове поле, Індексне поле
Найменування послуги	ServiceName	SNM	Varchar	Довжина: 255	-
Опис послуги	Description	DESC	Text	-	-

Ціна	Price	PR	Decimal	-	-
Тривалість	Duration	DUR	Integer	-	-
Категорія	Category	CAT	Varchar	Довжина: 100	-

Таблиця 3.5 – «Марка авто»

Марка автомобіля	Ідентифікатор у програмі	Умовне позначення у формулах	Формат	Бізнес-правила	Логічні чи семантичні зв'язки
ID марки	BrandID	BID	Integer	Первинний ключ	Значення: не нуль, Обов'язкове поле, Індексне поле
Назва марки	BrandName	BN	Varchar	Довжина: 255	-
Модель	Model	MDL	Varchar	Довжина: 255	-
Рік випуску	Year	YR	Integer	-	-
Потужність	Power	PWR	Integer	-	-
Максимальна вантажність	MaxLoad	ML	Decimal	-	-

Таблиця 3.6 – «Зв'язок між товарами і замовленнями»

Поле	Тип даних	Опис
product_has_order_id	INT	Унікальний ідентифікатор зв'язку товару з замовленням

product_id	INT	Ідентифікатор товару (запчастини або послуги)
order_id	INT	Ідентифікатор замовлення
quantity	INT	Кількість одиниць товару в замовленні

Таблиця 3.7 – «Зв'язок між користувачами і їх ролями в системі»

Поле	Тип даних	Опис
fix_user_role_id	INT	Унікальний ідентифікатор зв'язку між користувачем і роллю
user_id	INT	Ідентифікатор користувача
role_id	INT	Ідентифікатор ролі користувача

Таблиця 3.8 – «Замовлення на обслуговування

Поле	Тип даних	Опис	Поле	Тип даних	Опис
fix_order_id	INT	Унікальний ідентифікатор замовлення	fix_order_id	INT	Унікальний ідентифікатор замовлення
user_id	INT	Ідентифікатор користувача, який зробив замовлення	user_id	INT	Ідентифікатор користувача, який зробив замовлення

auto_id	INT	Ідентифікатор грузового автомобіля, який буде обслуговува тись	auto_id	INT	Ідентифікатор грузового автомобіля, який буде обслуговуватись
service_id	INT	Ідентифікатор послуги, яку потрібно виконати	service_id	INT	Ідентифікатор послуги, яку потрібно виконати
parts_id	INT	Ідентифікатор запчастини, яку потрібно встановити (за потреби)	parts_id	INT	Ідентифікатор запчастини, яку потрібно встановити (за потреби)
order_date	DATE	Дата замовлення	order_date	DATE	Дата замовлення
status	VARCHAR	Статус замовлення	status	VARCHAR	Статус замовлення
additional_notes	TEXT	Додаткові примітки або вимоги замовлення	additional_ notes	TEXT	Додаткові примітки або вимоги замовлення

Таблиця 3.9 – «Адреси замовлень»

Поле	Тип даних	Опис
address_id	INT	Унікальний ідентифікатор адреси
user_id	INT	Ідентифікатор користувача, якому належить адреса

street	VARCHAR	Назва вулиці
city	VARCHAR	Місто
postal_code	VARCHAR	Поштовий індекс
country	VARCHAR	Країна

Вибір СКБД. MySQL – це вільна система управління реляційними базами даних (СУБД), яка була розроблена як альтернатива комерційним системам і стала однією з найпоширеніших у світі. Ця СУБД використовується для створення динамічних веб-сторінок, забезпечуючи високу підтримку для різних мов програмування.

Після визначення з базою даних необхідно було створити модель бази даних яка продемонстрована на рис 3.1.

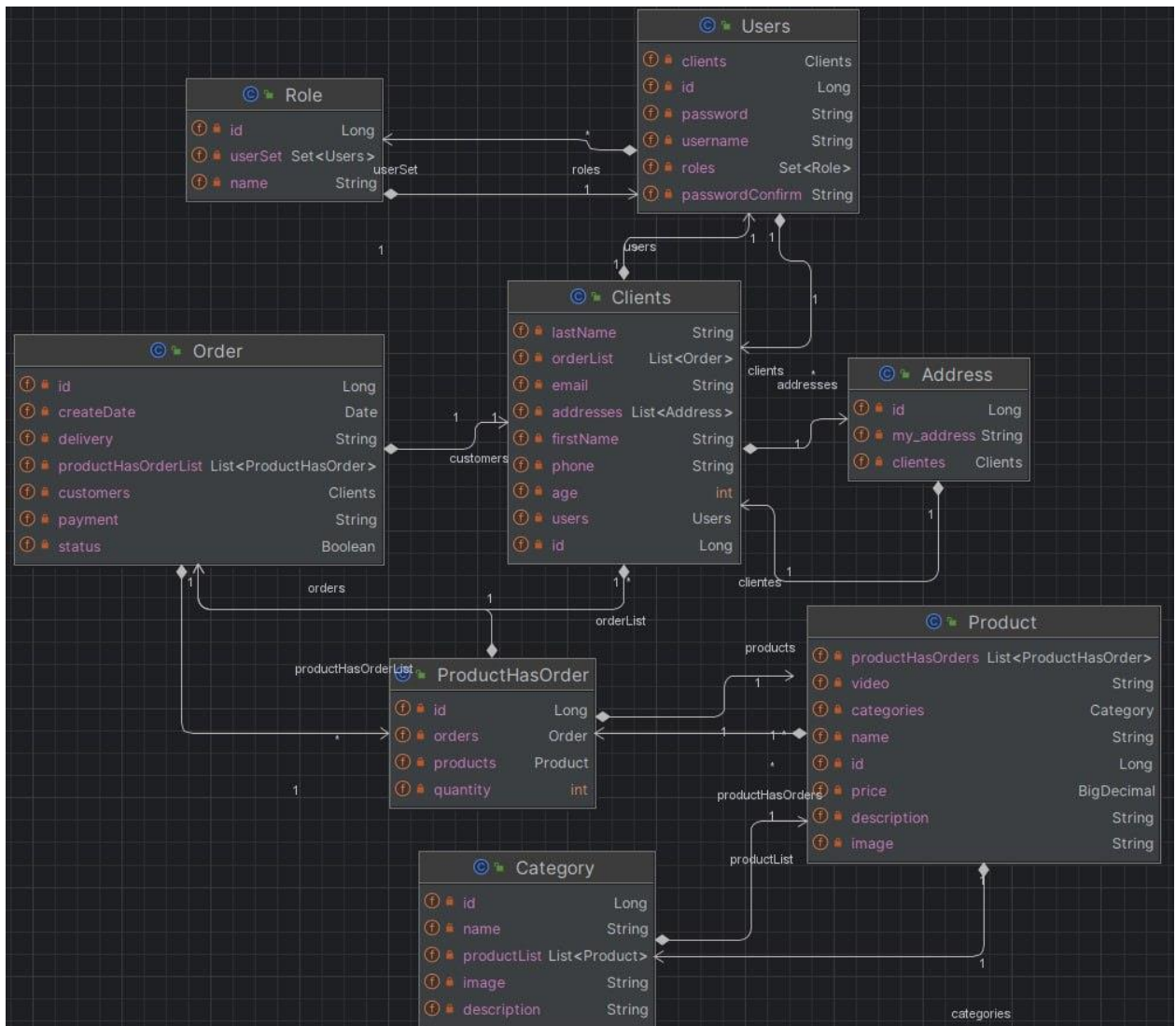


Рисунок 3.1 – Схема бази даних в IntelliJ Idea

MySQL володіє рядом властивостей, які роблять її ефективним інструментом для зберігання, обслуговування та отримання даних. Вона відзначається високою швидкістю, стійкістю та простотою використання. Розроблена з урахуванням підвищення продуктивності обробки великих баз даних, система працює на багатьох платформах, зокрема на UNIX-системах, де виявляється її повний потенціал завдяки підтримці багатоканальності.

Основні можливості сервера MySQL включають простоту для встановлення та подальшого використання, підтримку необмеженої кількості користувачів, велику кількість рядків у таблицях, високу швидкість виконання команд і наявність ефективної системи безпеки.

Обрання MySQL в якості СУБД для проекту обумовлено такими факторами, як масштабованість, переносність на різні платформи, зв'язаність та безпека. MySQL є відкритим кодом, що додатково підкреслює його важливість у сфері управління базами даних.

На основі цієї схеми було створено базу даних в програмі MySQL Workbench. Тут є усі необхідні колекції. Отриману базу даних наведено нижче(рис 3.2).

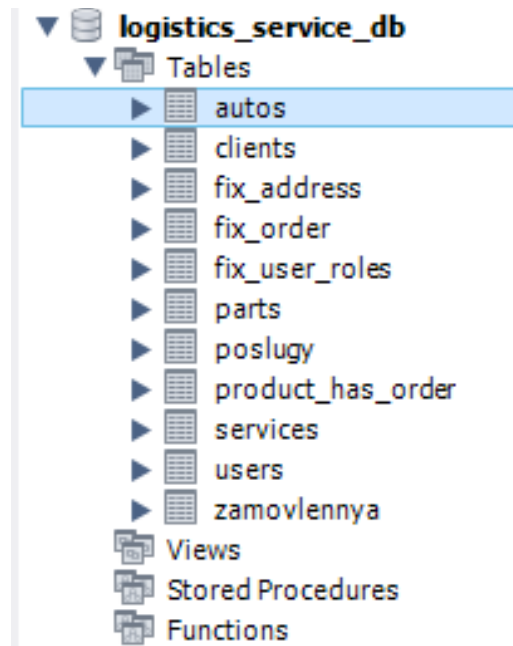


Рисунок 3.2 -База даних побудована в MySQL Workbench

Даталогічна модель бази даних.

Для створення даталогічної моделі було використано MySql Workbench.

Результат створення зображено на рис. 3.3.

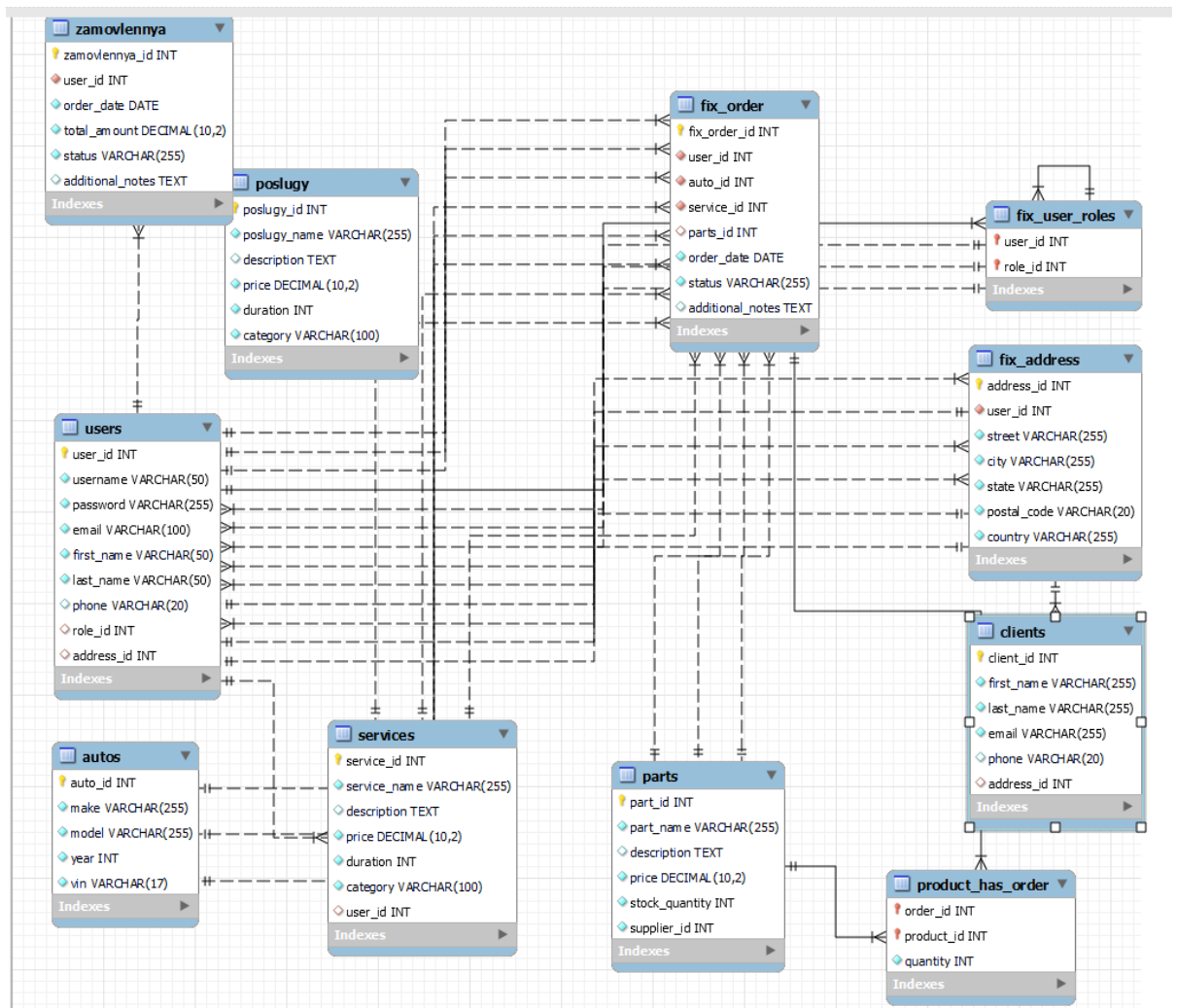


Рисунок 3.3 – Даталогічна модель бази даних в MySQL Workbench

На діаграмі класів (рис. 3.4) серверної частини системи, відповідно до підходу MVC, представлені моделі та контролери, але відсутні View. Класи діаграми можна поділити на три частини. Класи діаграми можна поділити на три частини:

1. Дані – всі елементи, що стосуються зберігання даних;
2. Бізнес-логіка – охоплює всі класи, які займаються обчисленнями та виконанням алгоритмів обробки даних;
3. Інтерфейси – до цієї частини відносяться класи, в яких збирається інформація з системи та передається до користувача (клієнта).

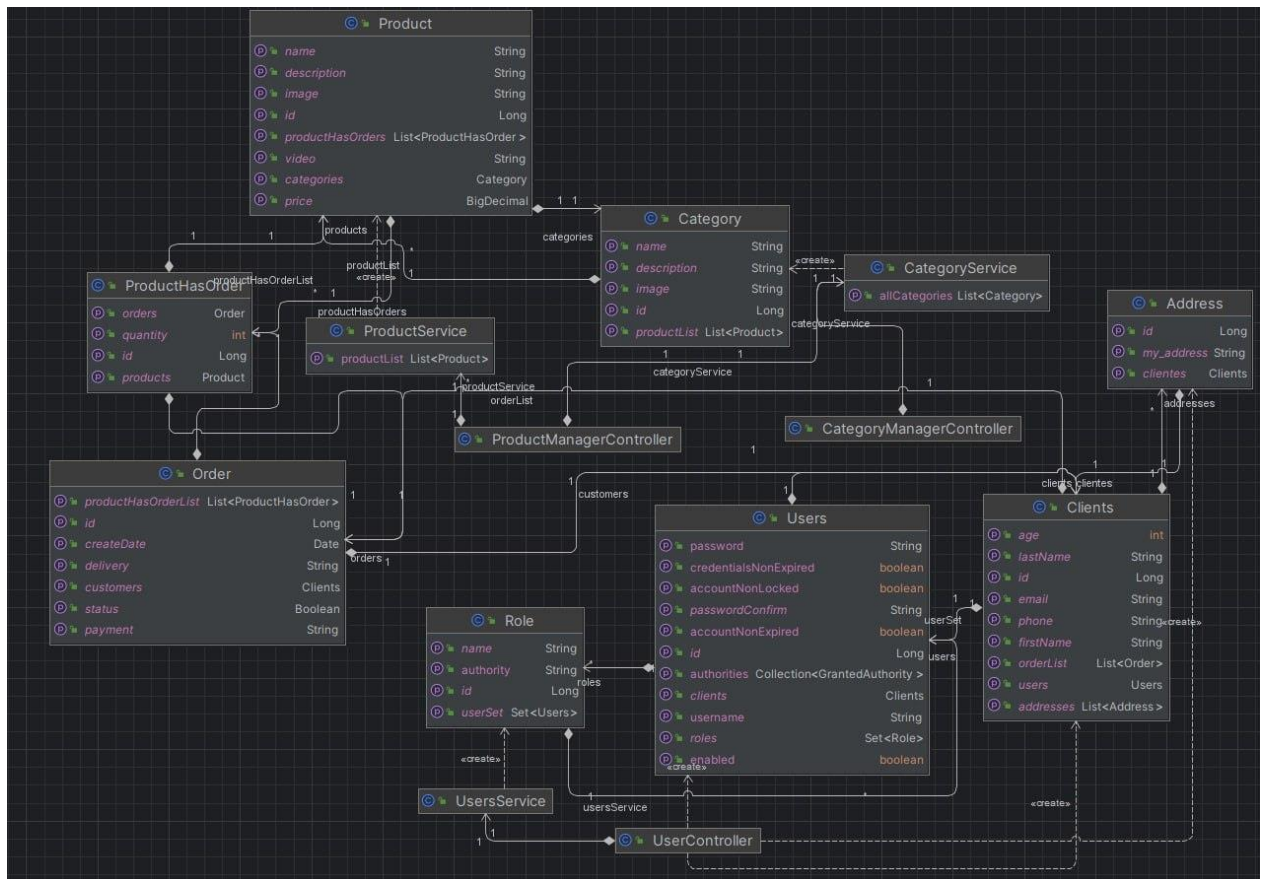


Рисунок 3.4 – Діаграма класів бази даних

Для кодування даних в базі даних використовується автоматичне порядкове кодування.

Результати перетворення ER-моделей в необхідні схеми бази даних для цієї системи на основі СУБД MySQL та вміст файлу згенерованої моделі даних представлено у «Додатку А» та «Додатку В»..

Основні таблиці бази даних наведені на рис. 3.5 – 3.17.

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table x

Table Name: clients

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
client_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
first_name	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
last_name	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
email	VARCHAR(255)	☐	☒	☒	☐	☐	☐	☐	☐	
phone	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
address_id	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Рисунок 3.5 – Таблица «clients»

fix_role - Table x

Table Name: fix_role

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	BIGINT	☒	☒	☐	☐	☐	☐	☒	☐	
name	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Рисунок 3.6 – Таблица «fix_role»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table f

Table Name: parts

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
part_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
part_name	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
description	TEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
price	DECIMAL(10,2)	☐	☒	☐	☐	☐	☐	☐	☐	
stock_quantity	INT	☐	☒	☐	☐	☐	☐	☐	☐	
supplier_id	INT	☐	☒	☐	☐	☐	☐	☐	☐	

Рисунок 3.7 – Таблица «parts»

category - Table ×

Table Name:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	BIGINT	☒	☒	☐	☐	☐	☐	☒	☐	
description	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
image	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
name	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Рисунок 3.8 – Таблица «category»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table

Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
service_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
service_name	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
description	TEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
price	DECIMAL(10,2)	☐	☒	☐	☐	☐	☐	☐	☐	
duration	INT	☐	☒	☐	☐	☐	☐	☐	☐	
category	VARCHAR(100)	☐	☒	☐	☐	☐	☐	☐	☐	
user_id	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Рисунок 3.9 – Таблица «services»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table fix_order - Table **fix_user_roles - Table** ×

Table Name: Schema: **logistics_ser**

Charset/Collation: Engine: **InnoDB**

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
user_id	INT	☒	☒	☐	☐	☐	☐	☐	☐	
role_id	INT	☒	☒	☐	☐	☐	☐	☐	☐	

Рисунок 3.10 – Таблица «fix_user_roles»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table **fix_order - Table** x

Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
fix_order_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
user_id	INT	☐	☒	☐	☐	☐	☐	☐	☐	
auto_id	INT	☐	☒	☐	☐	☐	☐	☐	☐	
service_id	INT	☐	☒	☐	☐	☐	☐	☐	☐	
parts_id	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
order_date	DATE	☐	☒	☐	☐	☐	☐	☐	☐	
status	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
additional_notes	TEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Рисунок 3.11 – Таблица «fix_orders»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table **fix_address - Table** x

Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
address_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
user_id	INT	☐	☒	☐	☐	☐	☐	☐	☐	
street	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
city	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
state	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
postal_code	VARCHAR(20)	☐	☒	☐	☐	☐	☐	☐	☐	
country	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	

Рисунок 3.12 – Таблица «fix_address»

SQL File 4* autos logistics_service_db.autos **autos - Table** x

Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
auto_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
make	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
model	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
year	INT	☐	☒	☐	☐	☐	☐	☐	☐	
vin	VARCHAR(17)	☐	☒	☒	☐	☐	☐	☐	☐	

Рисунок 3.13 – Таблица «autos»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table fi

Table Name: poslugy

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
poslugy_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
poslugy_name	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
description	TEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
price	DECIMAL(10,2)	☐	☒	☐	☐	☐	☐	☐	☐	
duration	INT	☐	☒	☐	☐	☐	☐	☐	☐	
category	VARCHAR(100)	☐	☒	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Рисунок 3.14 – Таблица «poslugy»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table

Table Name: product_has_order

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
order_id	INT	☒	☒	☐	☐	☐	☐	☐	☐	
product_id	INT	☒	☒	☐	☐	☐	☐	☐	☐	
quantity	INT	☐	☒	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Рисунок 3.15 – Таблица «product_has_order»

SQL File 4* autos logistics_service_db.autos autos - Table clients - Table fix_address - Table

Table Name: users

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
user_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
username	VARCHAR(50)	☐	☒	☐	☐	☐	☐	☐	☐	
password	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
email	VARCHAR(100)	☐	☒	☐	☐	☐	☐	☐	☐	
first_name	VARCHAR(50)	☐	☒	☐	☐	☐	☐	☐	☐	
last_name	VARCHAR(50)	☐	☒	☐	☐	☐	☐	☐	☐	
phone	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
role_id	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
address_id	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

Рисунок 3.16 – Таблица «users»

istics_service_db.autos autos - Table clients - Table fix_address - Table fix_order - Table fix_user_role

Table Name: zamovlennyya

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
zamovlennyya_id	INT	☒	☒	☐	☐	☐	☐	☒	☐	
user_id	INT	☐	☒	☐	☐	☐	☐	☐	☐	
order_date	DATE	☐	☒	☐	☐	☐	☐	☐	☐	
total_amount	DECIMAL(10,2)	☐	☒	☐	☐	☐	☐	☐	☐	
status	VARCHAR(255)	☐	☒	☐	☐	☐	☐	☐	☐	
additional_notes	TEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Рисунок 3.17 – Таблиця «zamovlennyya»

3.2. Технічне забезпечення

На цій діаграмі (рис. 3.18) зображено головні компоненти системи:

1) Клієнтська частина, яка реалізована у вигляді веб-сайту, де розміщений весь функціонал системи та візуальне оформлення.

2) Back частина, складається з двох компонентів: PlatformAPI - це API яке забезпечує основну функціональність сайту, авторизація користувачів, відображення товару, послуг та ін. Deliver – це доставка необхідного при потребі, також об'єкти зберігаються в MainDB.

3) Server - ця частина являє собою дві бази даних (MainDB, Layout), які зберігають в собі всю потрібну інформацію(клієнти, замовлення...) для нормального функціонування самої системи в цілому. Layout – це вже передача з БД до транспортної бази про замовлення.

Структуру комплексу технічних засобів наведено на рис. 3.22.

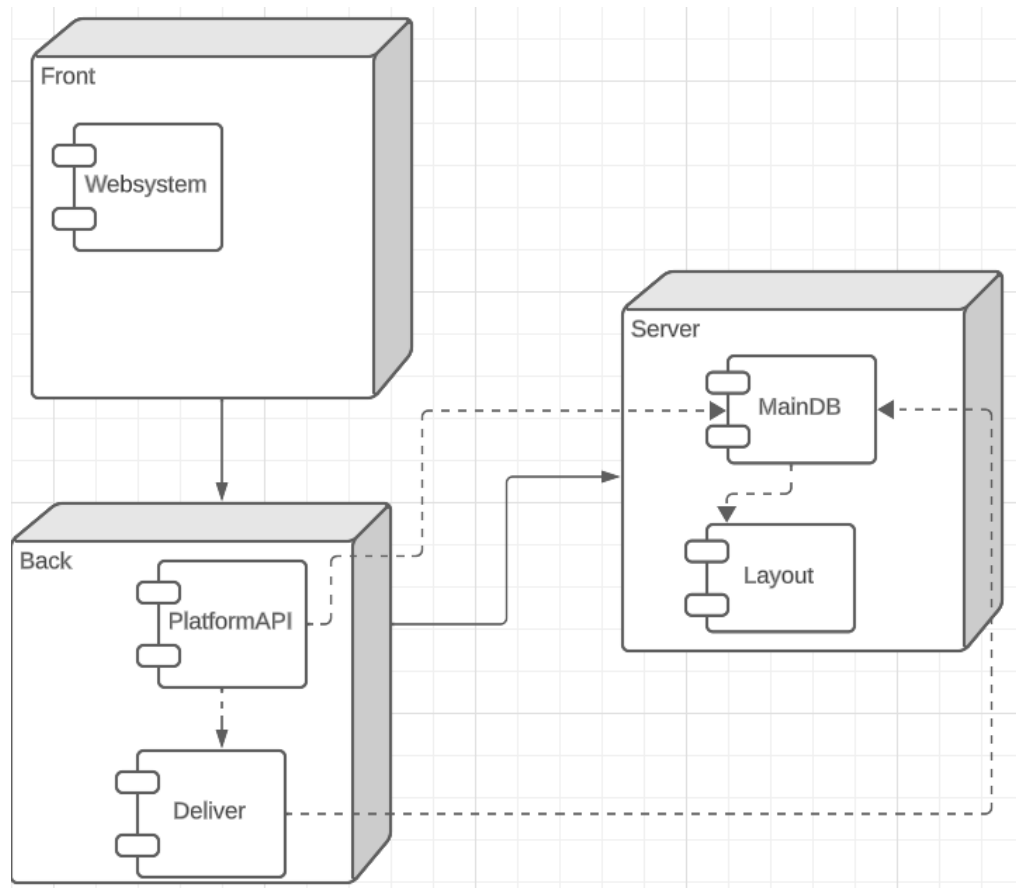


Рисунок 3.18 – UML діаграма розгортання

Для розробки клієнтських та серверних компонентів системи було використано мову програмування – Java, бібліотеку jQuery, мову розмітки – HTML, мову стилів – CSS та скриптову мову програмування – JavaScript.

3.3. Програмне забезпечення

Структура програмного забезпечення. HTTP є протоколом для передачі гіпертекстової інформації, який використовується як основа для функціонування сучасних веб-сайтів. Він дозволяє запитувати та отримувати дані з віддалених систем, таких як файли та веб-сторінки. Проект буде реалізовано з використанням цього протоколу.

Системне програмне забезпечення. Під час розробки системи використовувалися різноманітні засоби та середовища для забезпечення ефективності та якості процесу.

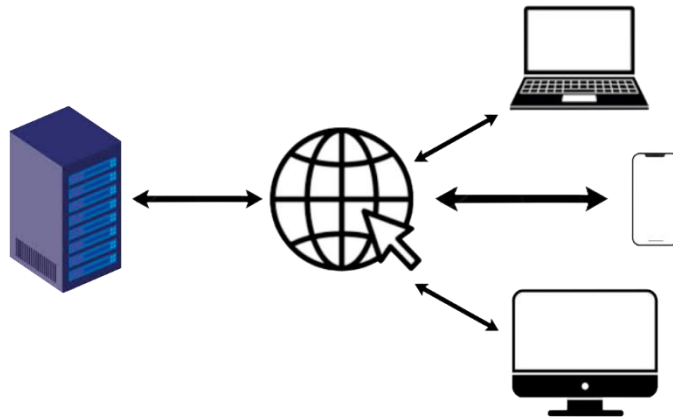


Рисунок 3.19 – Архітектура Клієнт-серверу

Операційною системою була використана Windows, яка виступала основною платформою для розробки та виконання проекту. Вона є надійною та зручною робочою середою із забезпеченими відповідними умови для роботи з іншими засобами.

Одним з ключових засобів розробки була IntelliJ IDEA – потужне інтегроване середовище розробки, що надало розробникам багатофункціональну платформу для створення веб-додатків.

Для редагування та набору коду також використовувався IntelliJ IDEA – легкий та розширюваний текстовий редактор. Середовище забезпечило зручне редагування коду в такому ж зручному інтерфейсі, що підтримує в собі різні мови програмування та засоби розробки.

Також додаткові інструменти та середовища були використаними під час розробки, такі як MySQL Workbench для роботи з базою даних MySQL, Enterprise Architect для моделювання архітектури системи, LucidChart для створення діаграм та схем.

Ці засоби та середовища були зручними та продуктивними в розробці проекту, дозволяючи розробнику швидко та з комфортом розробляти систему на різних етапах розробки та управління проектом.

Під час написання кваліфікаційного бакалаврського проекту були використані такі технології: HTML, CSS, JavaScript, Bootstrap, MySQL.

HTML та CSS використовувалися для створення структури та вигляду веб-сторінок, причому HTML відповідає за розміщення елементів, а CSS – за їх стилізацію та оформлення.

JavaScript (JS) додавав інтерактивність та динамічні можливості веб-сторінкам, включаючи функції, перевірку даних, маніпуляцію з DOM-елементами та комунікацію з сервером.

Bootstrap служив як фреймворк для швидкого та ефективного створення адаптивного та мобільного дизайну веб-сторінок, забезпечуючи готові компоненти, сітку та стилі для спрощення процесу розробки та забезпечення єдності в оформленні.

MySQL використовувався як система управління базами даних, забезпечуючи зберігання та управління інформацією, необхідною для функціонування додатків.

Використання цих технологій дозволило розробити веб-систему для логістичного підприємства зі зручним інтерфейсом для комфортного використання та надійним зберіганням даних.

Програмна документація. Для розробки системи був обраний шаблон проектування MVC (Model-View-Controller). Його функціонування та взаємодію з іншими компонентами можна побачити на діаграмі компонентів (див. рис. 3.20).

Для використання веб-системи користувачам необхідно мати:

- Комп'ютер або інший пристрій такі як ноутбук, планшет або смартфон з операційною системою котра підтримує відкриття веб-додатків.
- Веб-браузер. Для роботи з веб-додатком необхідні сучасні веб-браузери Google Chrome, Mozilla Firefox, Microsoft Edge або Safari. Браузер

повинен підтримувати HTML5, CSS3 та JavaScript.

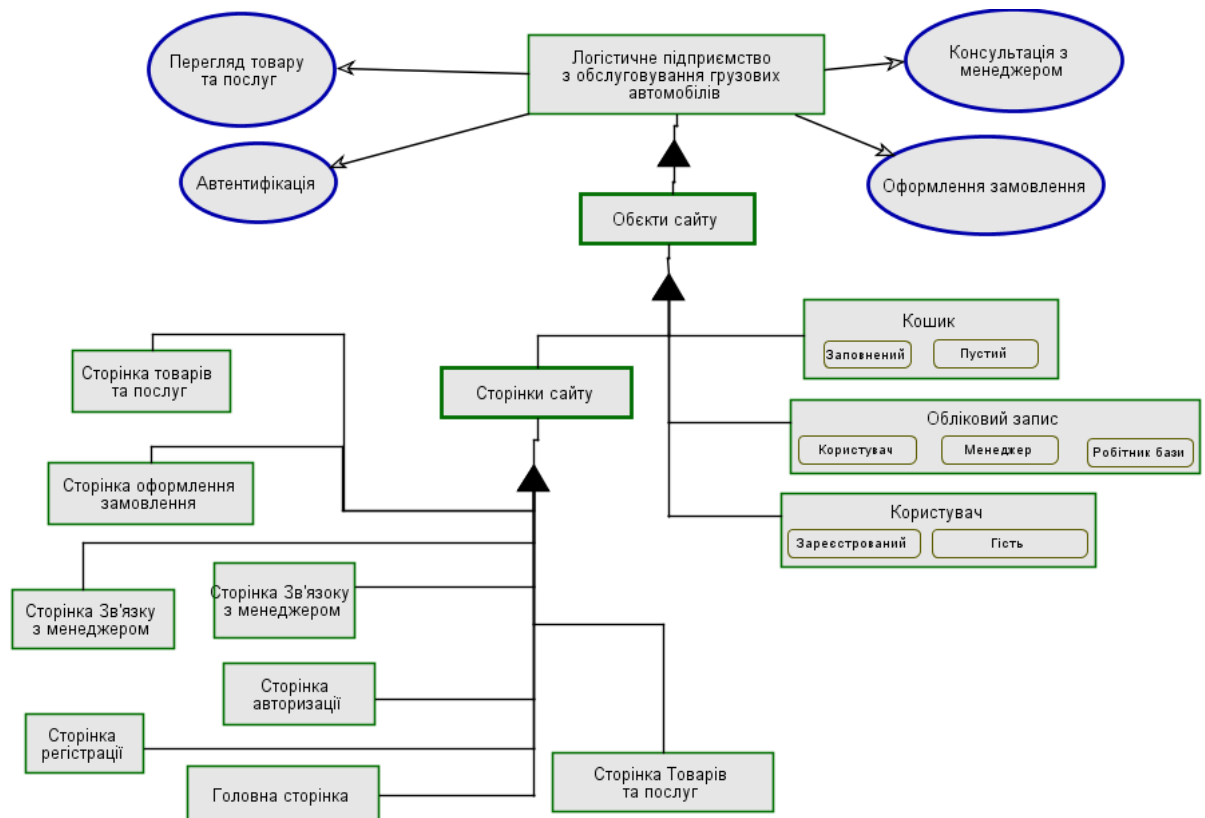


Рисунок 3.20 – Діаграма компонентів

- Інтернет-з'єднання. Стабільне підключення до Інтернету є обов'язковою умовою для доступу до веб-системи.
- Акаунт користувача. Створений обліковий запис в користувача або клієнта для входу і доступу до повного набору можливостей системи у вигляді оформлення замовлення та відстеження замовлень.

3.4. Результати реалізації інформаційної системи

Для перевірки відповідності створена система з поставленими вимогами та проведено її контрольний запуск.

Після того як був запуснений сервер та відкритий локальний хост у браузері нам

відкривається головна сторінка (див. рис. 3.21). На данній сторінці немає у собі ніякої функціональності крім авторизації та реєстрації, а також перехід по сторінках сайту.

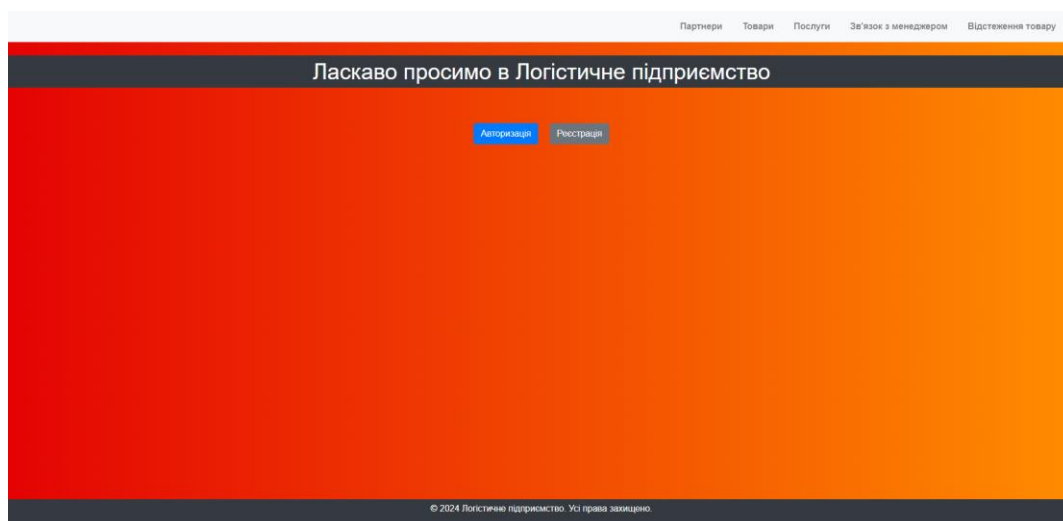


Рисунок 3.21 – Головна сторінка

Для того щоб почати повноцінно користуватися системою необхідно першим ділом створити обліковий запис. Для цього необхідний перехід на сторінку реєстрації (див. рис. 3.22), заповнимо форму та натиснемо на кнопку «Зареєструватися».

Рисунок 3.22 – Сторінка реєстрації

Може існувати ситуація при якій користувач не заповнив якесь поле або заповнив його не правильно. Для цього було додано простий JavaScript з перевіркою (див. рис. 3.23).

sadgmail.com

Адрес электронной почты должен содержать символ "@". В адресе "sadgmail.com" отсутствует символ "@".

@gmail.com

Введите часть адреса до символа "@". Адрес "@gmail.com" неполный.

Рисунок 3.23 – Поле Email не заповнене або не коректно заповнене

Якщо правильно заповнено усі поля (див. рис. 3.24) буде здійснено перехід до сторінки з товарами.

Логістичне підприємство

Авторизація Партнери Товари Послуги Зв'язок з менеджером Відстеження товару

Реєстрація

hello

sad@gmail.com

Vladar

Skoryk

+38043435224

Вул. Київська

Зареєструватися

Рисунок 3.24 – Коректно заповнена форма реєстрації

Після реєстрації можна перейти на сторінку авторизації (див. рис. 3.25)

Логістичне підприємство

Партнери Товари Послуги Зв'язок з менеджером Відстеження товару

Авторизація

Увійти

Рисунок 3.25 – Сторінка авторизації

Після авторизації одразу відбувається на сторінку товарів (див. рис. 3.26). Але на цей раз є додаткова вкладка кошику на головному меню.

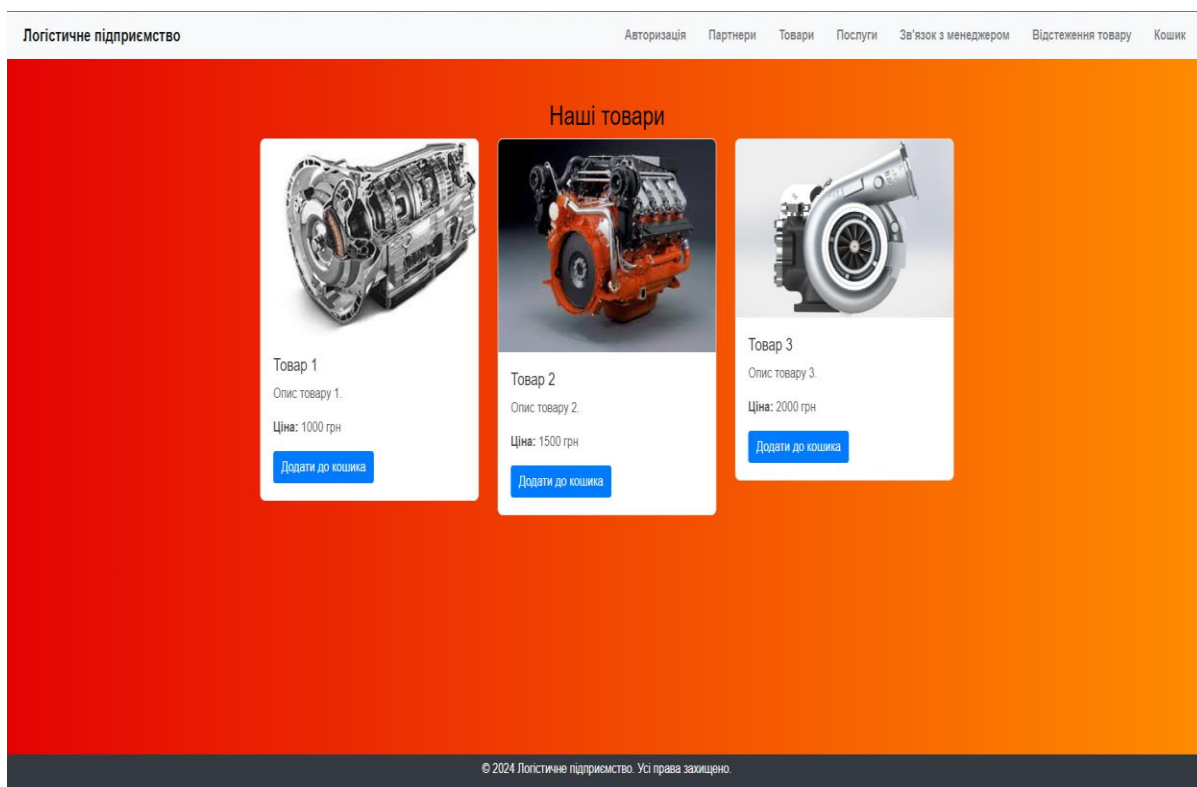


Рисунок 3.26 – Сторінка товарів після авторизації

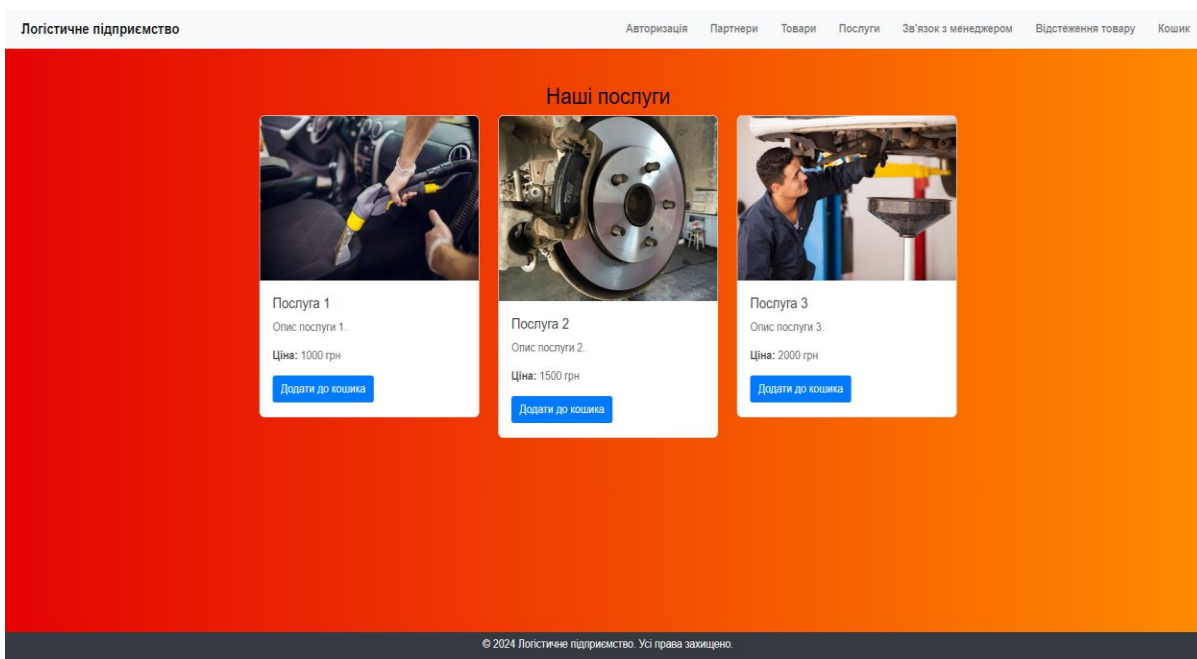


Рисунок 3.27 – Сторінка з послугами

При додаванні товарів до кошику вони добавляються в базу і після переходу на сторінку кошику нам показує все що ми обрали (див. рис. 3.28).

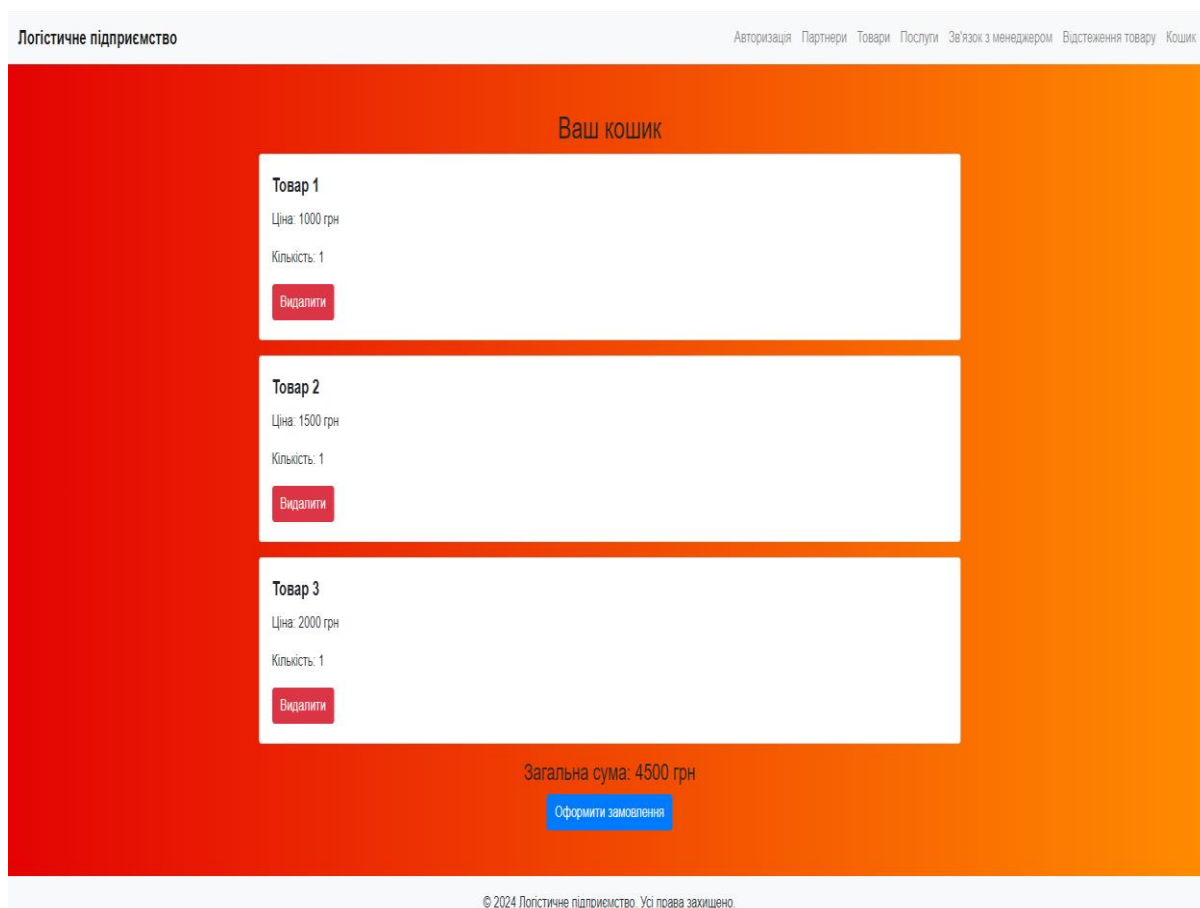


Рисунок 3.28 – Сторінка кошику з товарами

Також можна і далі добавляти товари до кошику (див. рис. 3.29).

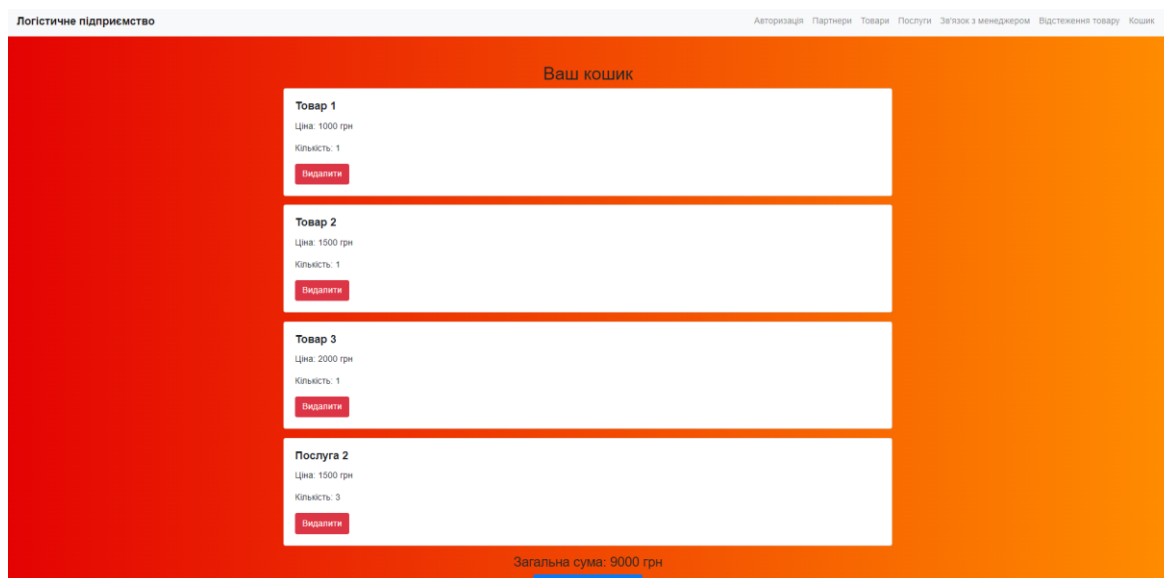


Рисунок 3.29 – Сторінка з більшою кількістю товарів у кошику

Також є можливість зв'язку з менеджером для консультацій (див. рис.

3.30).

Логістичне підприємство Головна Послуги Товари Зв'язок з менеджером Відстеження товару

Зв'язок з менеджером

Для зв'язку з менеджером, будь ласка, використовуйте один з наступних номерів:

- Менеджер 1: +1234567890
- Менеджер 2: +1987654321
- Менеджер 3: +1122334455

Ім'я:

Email:

Повідомлення:

Надіслати

Рисунок 3.30 – Сторінка для зв'язку з менеджером

Зазначу також, що ця система є лише прототипом і потребує подальшого вдосконалення. Більшість запланованого функціоналу ще не реалізовано або взагалі відсутній. Тому під час використання програми можливі деякі помилки у роботі певних функцій, які не були виявлені під час тестування.

ВИСНОВКИ

Під час розробки дипломного проекту "Проектування інформаційної веб-системи логістичного підприємства для обслуговування грузових автомобілів", було проведено дослідження та розробка інформаційної системи для даного логістичного підприємства.

Основною метою проекту було спрощення процесу замовлення та надання допомоги клієнтам щодо ремонту або перевезення товарів. На підставі проведених досліджень та аналізу користувачів були визначені функціональні, нефункціональні та бізнес-вимоги, а також розроблена архітектура системи. Для реалізації проекту були використані сучасні інформаційні технології, такі як HTML, CSS, JavaScript, SQL та інші. В результаті була створена веб-система, яка забезпечує зручний та ефективний спосіб замовлення послуг або товарів для грузових автомобілів.

Під час розробки були враховані потреби користувачів та забезпечено зручний інтерфейс для роботи з системою. В проекті присутні кілька переваг, зокрема: легка доступність та зрозумілість, швидкий доступ до необхідного. Проте, основним недоліком є те, що для використання системи потрібний доступ до мережі Інтернет.

Якщо усунути недоліки та допрацювати функціонал системи, це допоможе користувачам у зручності використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. MySql [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/MySQL>
2. MySql [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mysql.com/>
3. Lucidchart [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: <https://www.lucidchart.com/>
4. HTML [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>
5. Java [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Java>
6. Oracle [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oracle.com/cis/scm/logistics/transportation-management/>
7. Descartes [Електронний ресурс] – Режим доступу до ресурсу: <https://www.descartes.com/home>
8. MercuryGate TMS [Електронний ресурс] – Режим доступу до ресурсу: <https://mercurysgate.com/>
9. Designing Data-Intensive Applications [Електронний ресурс] / Martin Kleppmann – Режим доступу: <https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/>
10. Clean Code: A Handbook of Agile Software Craftsmanship [Електронний ресурс] / Robert C. Martin – Режим доступу: <https://www.oreilly.com/library/view/clean-code-a/9780136083238/>
11. Patterns of Enterprise Application Architecture [Електронний ресурс] / Martin Fowler – Режим доступу: <https://www.oreilly.com/library/view/patterns-of-enterprise/0321127420/>

12. The Pragmatic Programmer [Електронний ресурс] / Andrew Hunt, David Thomas – Режим доступу: <https://www.oreilly.com/library/view/the-pragmatic-programmer/9780135956977/>
13. Journal of Computer Information Systems [Електронний ресурс] – Режим доступу: <https://www.tandfonline.com/loi/ucis20>
14. Journal of Logistics Management [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/journal/journal-of-logistics-management>
15. IEEE Xplore Digital Library [Електронний ресурс] – Режим доступу: <https://ieeexplore.ieee.org/>
16. ACM Digital Library [Електронний ресурс] – Режим доступу: <https://dl.acm.org/>
17. MySQL Documentation [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/>
18. Spring Framework Documentation [Електронний ресурс] – Режим доступу: <https://spring.io/projects/spring-framework>
19. IntelliJ IDEA Documentation [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/idea/documentation/>
20. Wikipedia, статті про Java, HTML, Lucidchart [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/>
21. Medium, різні статті про веб-розробку та логістику [Електронний ресурс] – Режим доступу: <https://medium.com>
22. DZone, різні блоги про програмування та проектування систем [Електронний ресурс] – Режим доступу: <https://dzone.com>
23. Dev.to, різні блоги про програмування та проектування систем [Електронний ресурс] – Режим доступу: <https://dev.to>

24. Coursera, курс "Full Stack Web Development" [Електронний ресурс] – Режим доступу: <https://www.coursera.org/specializations/full-stack-web-development>
25. Gartner, звіти про інформаційні системи в логістиці [Електронний ресурс] – Режим доступу: <https://www.gartner.com/>
26. Oracle Transportation Management [Електронний ресурс] – Режим доступу: <https://www.oracle.com/cis/scm/logistics/transportation-management/>
27. Descartes [Електронний ресурс] – Режим доступу: <https://www.descartes.com/home>
28. "Distributed Systems: Principles and Paradigms" [Електронний ресурс] / Andrew S. Tanenbaum, Maarten Van Steen – Режим доступу: <https://www.pearson.com/store/p/distributed-systems-principles-and-paradigms/P100000003632>
29. "RESTful Web Services" [Електронний ресурс] / Leonard Richardson, Sam Ruby – Режим доступу: <https://www.oreilly.com/library/view/restful-web-services/9780596529260/>
30. "Building Microservices: Designing Fine-Grained Systems" [Електронний ресурс] / Sam Newman – Режим доступу: <https://www.oreilly.com/library/view/building-microservices/9781491950340/>
31. "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" [Електронний ресурс] / Gene Kim, Kevin Behr, George Spafford – Режим доступу: <https://www.oreilly.com/library/view/the-phoenix-project/9781942788294/>
32. "Effective Java" [Електронний ресурс] / Joshua Bloch – Режим доступу: <https://www.oreilly.com/library/view/effective-java/9780134686098/>
33. Google Scholar [Електронний ресурс] – Режим доступу: <https://scholar.google.com/>

34. ScienceDirect [Электронный ресурс] – Режим доступа: <https://www.sciencedirect.com/>
35. SpringerLink [Электронный ресурс] – Режим доступа: <https://link.springer.com/>
36. Wiley Online Library [Электронный ресурс] – Режим доступа: <https://onlinelibrary.wiley.com/>
37. ProQuest Dissertations & Theses [Электронный ресурс] – Режим доступа: <https://www.proquest.com/products-services/dissertations/>
38. MIT OpenCourseWare [Электронный ресурс] – Режим доступа: <https://ocw.mit.edu/>
39. Logistics Management Journal [Электронный ресурс] – Режим доступа: <https://www.logisticsmgmt.com/>
40. Transportation Research Part E: Logistics and Transportation Review [Электронный ресурс] – Режим доступа: <https://www.journals.elsevier.com/transportation-research-part-e-logistics-and-transportation-review>
41. Journal of Business Logistics [Электронный ресурс] – Режим доступа: <https://onlinelibrary.wiley.com/journal/21581592>
42. IBM Supply Chain Solutions [Электронный ресурс] – Режим доступа: <https://www.ibm.com/supply-chain>
43. SAP Transportation Management [Электронный ресурс] – Режим доступа: <https://www.sap.com/products/transportation-management.html>
44. TMW Systems [Электронный ресурс] – Режим доступа: <https://www.tmwsystems.com/>
45. Blue Yonder (JDA Software) [Электронный ресурс] – Режим доступа: <https://blueyonder.com/>

46. Manhattan Associates [Електронний ресурс] – Режим доступу:
<https://www.manh.com/>

47. McKinsey & Company, звіти про цифрову трансформацію в логістиці
[Електронний ресурс] – Режим доступу: <https://www.mckinsey.com/>

48. Logistics Bureau [Електронний ресурс] – Режим доступу:
<https://www.logisticsbureau.com/>

ДОДАТКИ

ДОДАТОК А

Додаток А.1 – Код створення структури бази даних

```
USE logistics_service_db;
```

```
CREATE TABLE users (
```

```
    user_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    username VARCHAR(50) NOT NULL,
```

```
    password VARCHAR(255) NOT NULL,
```

```
    email VARCHAR(100) NOT NULL,
```

```
    first_name VARCHAR(50) NOT NULL,
```

```
    last_name VARCHAR(50) NOT NULL,
```

```
    phone VARCHAR(20),
```

```
    role_id INT,
```

```
    address_id INT,
```

```
    FOREIGN KEY (role_id) REFERENCES fix_user_roles(role_id),
```

```
    FOREIGN KEY (address_id) REFERENCES fix_address(address_id)
```

```
);
```

```
CREATE TABLE fix_order (
```

```
    fix_order_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    user_id INT NOT NULL,
```

```
    auto_id INT NOT NULL,
```

```
    service_id INT NOT NULL,
```

```
    parts_id INT DEFAULT NULL,
```

```
    order_date DATE NOT NULL,
```

```
    status VARCHAR(255) NOT NULL,
```

```
    additional_notes TEXT,
```

```
    FOREIGN KEY (user_id) REFERENCES users(user_id),
```

```
    FOREIGN KEY (auto_id) REFERENCES autos(auto_id),
```

```
    FOREIGN KEY (service_id) REFERENCES services(service_id),
```

```
    FOREIGN KEY (parts_id) REFERENCES parts(parts_id)
```

```
);
```

```
CREATE TABLE fix_address (  
    address_id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,  
    street VARCHAR(255) NOT NULL,  
    city VARCHAR(255) NOT NULL,  
    state VARCHAR(255) NOT NULL,  
    postal_code VARCHAR(20) NOT NULL,  
    country VARCHAR(255) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

```
CREATE TABLE services (  
    service_id INT AUTO_INCREMENT PRIMARY KEY,  
    service_name VARCHAR(255) NOT NULL,  
    description TEXT,  
    price DECIMAL(10, 2) NOT NULL,  
    duration INT NOT NULL,  
    category VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE fix_user_roles (  
    user_id INT NOT NULL,  
    role_id INT NOT NULL,  
    PRIMARY KEY (user_id, role_id),  
    FOREIGN KEY (user_id) REFERENCES users(user_id),  
    FOREIGN KEY (role_id) REFERENCES roles(role_id)  
);
```

```
CREATE TABLE product_has_order (  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL,  
    PRIMARY KEY (order_id, product_id),  
    FOREIGN KEY (order_id) REFERENCES fix_order(fix_order_id),
```

```
FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

```
CREATE TABLE autos (
    auto_id INT AUTO_INCREMENT PRIMARY KEY,
    make VARCHAR(255) NOT NULL,
    model VARCHAR(255) NOT NULL,
    year INT NOT NULL,
    vin VARCHAR(17) NOT NULL UNIQUE
);
```

```
CREATE TABLE poslugy (
    poslugy_id INT AUTO_INCREMENT PRIMARY KEY,
    poslugy_name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    duration INT NOT NULL,
    category VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE parts (
    part_id INT AUTO_INCREMENT PRIMARY KEY,
    part_name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    stock_quantity INT NOT NULL,
    supplier_id INT NOT NULL,
    FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id)
);
```

```
CREATE TABLE zamovlennya (
    zamovlennya_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    order_date DATE NOT NULL,
```

```
total_amount DECIMAL(10, 2) NOT NULL,  
status VARCHAR(255) NOT NULL,  
additional_notes TEXT,  
FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

```
CREATE TABLE clients (  
    client_id INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(255) NOT NULL,  
    last_name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    phone VARCHAR(20),  
    address_id INT,  
    FOREIGN KEY (address_id) REFERENCES fix_address(address_id)  
);}  
USE logistics_service_db;
```

-- Наповнення таблиці clients

```
INSERT INTO clients (first_name, last_name, email, phone, address_id)  
VALUES  
    ('John', 'Doe', 'john.doe@example.com', '1234567890', 1),  
    ('Jane', 'Smith', 'jane.smith@example.com', '0987654321', 2),  
    ('Bob', 'Brown', 'bob.brown@example.com', '1122334455', 3);
```

```
INSERT INTO fix_address (user_id, street, city, state, postal_code, country)  
VALUES  
    (1, '123 Main St', 'Springfield', 'IL', '62701', 'USA'),  
    (2, '456 Elm St', 'Metropolis', 'NY', '10001', 'USA'),  
    (3, '789 Oak St', 'Gotham', 'NJ', '07001', 'USA');
```

```
INSERT INTO services (service_name, description, price, duration, category)  
VALUES  
    ('Interior Cleaning', 'Complete interior cleaning service', 50.00, 120, 'Cleaning'),  
    ('Exterior Wash', 'Full exterior wash', 30.00, 60, 'Cleaning'),
```

```
('Oil Change', 'Engine oil change and filter replacement', 75.00, 90, 'Maintenance');
```

```
INSERT INTO fix_user_roles (user_id, role_id)
```

```
VALUES
```

```
(1, 1),
```

```
(2, 2),
```

```
(3, 3);
```

```
INSERT INTO product_has_order (order_id, product_id, quantity)
```

```
VALUES
```

```
(1, 1, 2),
```

```
(2, 2, 1),
```

```
(3, 3, 4);
```

```
INSERT INTO autos (make, model, year, vin)
```

```
VALUES
```

```
('Volvo', 'FH16', 2020, '1HTMKADN43H561234'),
```

```
('Scania', 'R450', 2019, '1XKAD49X6DJ123456'),
```

```
('Mercedes-Benz', 'Actros', 2021, '1FTSW21P54EC98765');
```

```
INSERT INTO poslugy (poslugy_name, description, price, duration, category)
```

```
VALUES
```

```
('Salon Cleaning', 'Complete salon cleaning', 40.00, 90, 'Cleaning'),
```

```
('Truck Wash', 'Exterior wash of truck', 35.00, 60, 'Cleaning'),
```

```
('Part Replacement', 'Replacement of truck parts', 100.00, 120, 'Maintenance');
```

```
INSERT INTO parts (part_name, description, price, stock_quantity, supplier_id)
```

```
VALUES
```

```
('Brake Pad', 'High quality brake pads', 25.00, 100, 1),
```

```
('Air Filter', 'Durable air filter', 15.00, 150, 2),
```

```
('Oil Filter', 'Engine oil filter', 10.00, 200, 3);
```

```
INSERT INTO zamovlennyya (user_id, order_date, total_amount, status, additional_notes)
```

```
VALUES
```

```
(1, '2024-06-01', 200.00, 'Pending', 'Urgent order, please expedite'),  
(2, '2024-06-02', 150.00, 'Completed', 'Regular service'),  
(3, '2024-06-03', 300.00, 'In Progress', 'Include tire rotation');
```

```
INSERT INTO fix_order (user_id, auto_id, service_id, parts_id, order_date, status,  
additional_notes)
```

```
VALUES
```

```
(1, 1, 1, 1, '2024-06-01', 'Pending', 'Replace brake pads and clean interior'),  
(2, 2, 2, NULL, '2024-06-02', 'Completed', 'Complete exterior wash'),  
(3, 3, 3, 2, '2024-06-03', 'In Progress', 'Oil change and replace air filter');
```

```
INSERT INTO users (username, password, email, first_name, last_name, phone, role_id,  
address_id)
```

```
VALUES
```

```
('jdoe', 'password123', 'jdoe@example.com', 'John', 'Doe', '1234567890', 1, 1),  
(jsmith', 'password123', 'jsmith@example.com', 'Jane', 'Smith', '0987654321', 2, 2),  
(bbrown', 'password123', 'bbrown@example.com', 'Bob', 'Brown', '1122334455', 3, 3);
```

Додаток Б - Таблиці бази даних, заповнені тестовими даними

	auto_id	make	model	year	vin
▶	1	Volvo	FH16	2020	1HTMKADN43H561234
	2	Scania	R450	2019	1XKAD49X6DJ123456
	3	Mercedes-Benz	Actros	2021	1FTSW21P54EC98765
*	NULL	NULL	NULL	NULL	NULL

Рисунок Б.1 – Таблиця «autos»

	client_id	first_name	last_name	email	phone	address_id
▶	1	John	Doe	john.doe@example.com	1234567890	1
	2	Jane	Smith	jane.smith@example.com	0987654321	2
	3	Bob	Brown	bob.brown@example.com	1122334455	3
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.2 – Таблиця «clients»

	address_id	user_id	street	city	state	postal_code	country
▶	1	1	123 Main St	Springfield	IL	62701	USA
	2	2	456 Elm St	Metropolis	NY	10001	USA
	3	3	789 Oak St	Gotham	NJ	07001	USA
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.3 – Таблиця «fix_adress»

	fix_order_id	user_id	auto_id	service_id	parts_id	order_date	status	additional_notes
▶	1	1	1	1	1	2024-06-01	Pending	Replace brake pads and clean interior
	2	2	2	2	NULL	2024-06-02	Completed	Complete exterior wash
	3	3	3	3	2	2024-06-03	In Progress	Oil change and replace air filter
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.4 – Таблиця «fix_order»

	user_id	role_id
▶	1	1
	2	2
	3	3
*	NULL	NULL

Рисунок Б.5 – Таблиця «fix_user_role»

	part_id	part_name	description	price	stock_quantity	supplier_id
▶	1	Brake Pad	High quality brake pads	25.00	100	1
	2	Air Filter	Durable air filter	15.00	150	2
	3	Oil Filter	Engine oil filter	10.00	200	3
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.6 – Таблиця «parts»

	poslugy_id	poslugy_name	description	price	duration	category
▶	1	Salon Cleaning	Complete salon cleaning	40.00	90	Cleaning
	2	Truck Wash	Exterior wash of truck	35.00	60	Cleaning
	3	Part Replacement	Replacement of truck parts	100.00	120	Maintenance
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.7 – Таблица «poslugi»

	order_id	product_id	quantity
▶	1	1	2
	2	2	1
	3	3	4
*	NULL	NULL	NULL

Рисунок Б.8 – Таблица «product_has_order»

	service_id	service_name	description	price	duration	category	user_id
▶	1	Interior Cleaning	Complete interior cleaning service	50.00	120	Cleaning	NULL
	2	Exterior Wash	Full exterior wash	30.00	60	Cleaning	NULL
	3	Oil Change	Engine oil change and filter replacement	75.00	90	Maintenance	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.9 – Таблица «services»

	user_id	username	password	email	first_name	last_name	phone	role_id	address_id
▶	1	jdoe	password123	jdoe@example.com	John	Doe	1234567890	1	1
	2	jsmith	password123	jsmith@example.com	Jane	Smith	0987654321	2	2
	3	bbrown	password123	bbrown@example.com	Bob	Brown	1122334455	3	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.10 – Таблица «users»

	zamovlennaya_id	user_id	order_date	total_amount	status	additional_notes
▶	1	1	2024-06-01	200.00	Pending	Urgent order, please expedite
	2	2	2024-06-02	150.00	Completed	Regular service
	3	3	2024-06-03	300.00	In Progress	Include tire rotation
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок Б.11 – Таблица «zamovlennaya»

Додаток В

Додаток В.1 - Код веб-сторінок HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Логістичне підприємство</title>

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <link rel="stylesheet" href="1.css">

</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#"></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="partners.html">Партнери</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="products.html">Товари</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="services.html">Послуги</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html">Зв'язок з менеджером</a>
        </li>
      </ul>
    </div>
  </nav>
```

```

        <li class="nav-item">
            <a class="nav-link" href="tracking.html">Відстеження товару</a>
        </li>
    </ul>
</div>
</nav>

<header class="text-center my-4">
    <h1>Ласкаво просимо в Логістичне підприємство</h1>
</header>

<main class="text-center my-4">
    <p>Вітаємо на нашому сайті! Будь ласка, авторизуйтесь або зареєструйтесь
для доступу до наших послуг.</p>
    <a href="login.html" class="btn btn-primary mx-2">Авторизація</a>
    <a href="register.html" class="btn btn-secondary mx-2">Реєстрація</a>
</main>

<footer class="text-center my-4">
    <p>&copy; 2024 Логістичне підприємство. Усі права захищено.</p>
</footer>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"><
/script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script
>
</body>
</html>

```

```

body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    background: linear-gradient(to right, #e40303, #ff8c00); /* Рожевий до

```

```
оранжевого */
    -webkit-background-clip: text;
    color: transparent;
}

header {
    background-color: #343a40; /* Dark Gray */
    color: white;
    padding: 5px;
    text-align: center;
}

header h1 {
    margin: 0;
}
h2{color:black;
}
div.product{color:black;
}
main {
    margin: 20px;
}

footer {
    background-color: #343a40; /* Dark Gray */
    color: white;
    text-align: center;
    padding: 5px;
    position: fixed;
    bottom: 0;
    width: 100%;
}

/* Навігаційні стилі */
nav {
    display: flex;
```

```
    justify-content: center;
    background-color: #343a40; /* Dark Gray */
    padding: 10px;
}
```

```
nav a {
    color: white;
    margin: 0 10px;
    text-decoration: none;
    font-weight: bold;
    transition: color 0.3s ease-in-out;
}
```

```
nav a:hover {
    color: #28a745; /* Green */
}
```

```
/* Стили для товарів та корзини */
#products {
    display: flex;
    justify-content: space-around;
    flex-wrap: wrap;
}
```

```
.product {
    background-color: #fff;
    border: 1px solid #ddd;
    border-radius: 8px;
    padding: 20px;
    margin: 10px;
    text-align: center;
    transition: transform 0.3s ease-in-out;
}
```

```
.product:hover {
    transform: scale(1.05);
}
```

```
}
```

```
.cart-item {  
    background-color: #28a745; /* Green */  
    color: white;  
    padding: 10px;  
    margin: 10px;  
    border-radius: 5px;  
    display: flex;  
    justify-content: space-between;  
}
```

```
.cart-item button {  
    background-color: #dc3545; /* Red */  
    color: white;  
    border: none;  
    padding: 5px 10px;  
    border-radius: 3px;  
    cursor: pointer;  
    transition: background-color 0.3s ease-in-out;  
}
```

```
.cart-item button:hover {  
    background-color: #c82333; /* Darker Red */  
}
```

```
/* Стили для розділів і заголовків */
```

```
.section-title {  
    font-size: 24px;  
    margin-bottom: 15px;  
    color: #007bff; /* Blue */  
}
```

```
.card {  
    margin-bottom: 20px;  
    border: 1px solid #ddd;  
    border-radius: 8px;
```

```
    overflow: hidden;
    transition: transform 0.3s ease-in-out;
}
```

```
.card:hover {
    transform: scale(1.05);
}
```

```
.card img {
    width: 100%;
    height: auto;
}
```

```
.card-body {
    padding: 20px;
}
```

```
.card-title {
    font-size: 20px;
    margin-bottom: 10px;
    color: #333; /* Dark Gray */
}
```

```
.card-text {
    font-size: 16px;
    color: #555; /* Medium Gray */
}
```

```
<!DOCTYPE html>
<html lang="uk">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Товари</title>

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <link rel="stylesheet" href="1.css">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="index.html">Логістичне підприємство</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="login.html">Авторизація</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="partners.html">Партнери</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="products.html">Товари</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="services.html">Послуги</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html">Зв'язок з менеджером</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="tracking.html">Відстеження товару</a>
        </li>
      </ul>
    </div>
  </nav>
```

```

    <li class="nav-item">
      <a class="nav-link" href="cart.html">Кошик</a>
    </li>
  </ul>
</div>
</nav>

<div class="container my-5">
  <h2 class="text-center">Наші товари</h2>
  <div class="row" id="products">
    <div class="col-md-4">
      <div class="card mb-4">
        
        <div class="card-body">
          <h5 class="card-title">Товар 1</h5>
          <p class="card-text">Опис товару 1.</p>
          <p class="card-text"><strong>Ціна:</strong> 1000 грн</p>
          <button class="btn btn-primary" onclick="addToCart('Товар 1',
1000)">Додати до кошика</button>
        </div>
      </div>
    </div>
    <div class="col-md-4">
      <div class="card mb-4">
        
        <div class="card-body">
          <h5 class="card-title">Товар 2</h5>
          <p class="card-text">Опис товару 2.</p>
          <p class="card-text"><strong>Ціна:</strong> 1500 грн</p>
          <button class="btn btn-primary" onclick="addToCart('Товар 2',
1500)">Додати до кошика</button>
        </div>
      </div>
    </div>
    <div class="col-md-4">
      <div class="card mb-4">
        

```

```

        <div class="card-body">
            <h5 class="card-title">Товар 3</h5>
            <p class="card-text">Опис товару 3.</p>
            <p class="card-text"><strong>Ціна:</strong> 2000 грн</p>
            <button class="btn btn-primary" onclick="addToCart('Товар 3',
2000)">Додати до кошика</button>
        </div>
    </div>
</div>
    <!-- Додайте більше товарів за потреби -->
</div>
</div>

<footer class="text-center my-4">
    <p>&copy; 2024 Логістичне підприємство. Усі права захищено.</p>
</footer>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"><
/script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script
>
<script>
function addToCart(productName, productPrice) {
    let cart = JSON.parse(localStorage.getItem('cart')) || [];
    const product = { name: productName, price: productPrice, quantity: 1 };

    // Check if the product is already in the cart
    const existingProduct = cart.find(item => item.name === productName);

    if (existingProduct) {
        existingProduct.quantity++;
    } else {
        cart.push(product);
    }
}

```

```
        localStorage.setItem('cart', JSON.stringify(cart));
        alert('Товар додано до кошика');
    }
</script>
</body>
</html>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Кошик</title>

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
    <link rel="stylesheet" href="koshik.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="index.html">Логістичне підприємство</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="login.html">Авторизація</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="partners.html">Партнери</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="products.html">Товари</a>
                </li>
            </ul>
        </div>
    </nav>
</body>
</html>
```

```
</li>
<li class="nav-item">
  <a class="nav-link" href="services.html">Послуги</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="contact.html">Зв'язок з менеджером</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="tracking.html">Відстеження товару</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="cart.html">Кошик</a>
</li>
</ul>
</div>
</nav>

<div class="container my-5">
  <h2 class="text-center">Ваш кошик</h2>
  <div id="cart-items"></div>
  <div class="text-center">
    <button class="btn btn-primary" onclick="checkout()">Оформити
замовлення</button>
  </div>
</div>

<footer class="text-center my-4">
  <p>&copy; 2024 Логістичне підприємство. Усі права захищено.</p>
</footer>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"><
/script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script
>
```

```

<script>
function loadCart() {
    let cart = JSON.parse(localStorage.getItem('cart')) || [];
    let cartItems = document.getElementById('cart-items');
    cartItems.innerHTML = "";

    if (cart.length === 0) {
        cartItems.innerHTML = '<p class="text-center">Ваш кошик
порожній</p>';
        return;
    }

    let total = 0;
    cart.forEach(item => {
        total += item.price * item.quantity;
        cartItems.innerHTML += `
        <div class="card mb-3">
            <div class="card-body">
                <h5 class="card-title">${item.name}</h5>
                <p class="card-text">Ціна: $ {item.price} грн</p>
                <p class="card-text">Кількість: $ {item.quantity}</p>
                <button class="btn btn-danger"
onclick="removeFromCart('${item.name}')">Видалити</button>
            </div>
        </div>
        `;
    });

    cartItems.innerHTML += `<h4 class="text-center">Загальна сума: $ {total}
грн</h4>`;
}

function removeFromCart(productName) {
    let cart = JSON.parse(localStorage.getItem('cart')) || [];
    cart = cart.filter(item => item.name !== productName);
    localStorage.setItem('cart', JSON.stringify(cart));
    loadCart();
}

```

```
}
```

```
function checkout() {  
    alert('Дякуємо за покупку!');  
    localStorage.removeItem('cart');  
    loadCart();  
}
```

```
    window.onload = loadCart;  
</script>  
</body>  
</html>
```

```
/* 1.css */  
body {  
    font-family: 'Arial', sans-serif;  
}
```

```
.navbar-brand {  
    font-weight: bold;  
}
```

```
.card-title {  
    font-size: 1.25rem;  
    font-weight: bold;  
}
```

```
.card-text {  
    font-size: 1rem;  
}
```

```
footer {  
    background-color: #f8f9fa;  
    padding: 1rem;  
}
```

```
.btn-primary {  
  background-color: #007bff;  
  border-color: #007bff;  
}
```

```
#cart-items .card {  
  margin-bottom: 1rem;  
}
```

```
body {  
  font-family: 'Arial', sans-serif;  
  margin: 0;  
  padding: 0;  
  background: linear-gradient(to right, #e40303, #ff8c00); /* Рожевий до  
оранжевого */  
}
```

Имя пользователя:
Інформаційних систем в економіці Шкуратовська Те...

ID проверки:
1016332861

Дата проверки:
08.06.2024 00:30:25 EEST

Тип проверки:
Doc vs Internet + Library

Дата отчета:
08.06.2024 07:01:44 EEST

ID пользователя:
100005745

Название файла: Скорик В.ІН-403

Количество страниц: 65 Количество слов: 7269 Количество символов: 59071 Размер файла: 13.04 MB ID файла: 1016132834

Обнаружены модификации текста (могут влиять на процент совпадений)

18.8%
Совпадения

Наибольшее совпадение: 9.33% с источником из Библиотеки (ID файла: 1015232085)

3.38% Источники из Интернета 196 Страница 67

18.7% Источники из Библиотеки 390 Страница 68

0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

0% Исключений

Нет исключенных источников

Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 14

Подозрительное форматирование 13 страниц