

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.
«_____» 20____ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: «Розробка застосунку для автоматизації діяльності логістичної компанії «ТрансВей»»

Виконав: студент групи 4151

_____ Салов В.О.
(підпис, ПІБ)

Керівник роботи:

Доцент кафедри ПЗАС, к.т.н. _____

(посада, науковий ступень, вчене звання)

_____ Фаріонова Т.А.
(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
 Навчально-науковий інститут комп'ютерних наук та управління проектами
 Кафедра програмного забезпечення автоматизованих систем
 Спеціальність 121 «Інженерія програмного забезпечення»
 Освітня програма «Інженерія програмного забезпечення»



«ЗАТВЕРДЖУЮ»
 Гарант освітньої програми
 _____ доц. Макарова Л.М.

(підпис)

« 08 » 04 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту _____ Салову Владиславу Олеговичу _____
 (Прізвище, ім'я, по батькові)

1. Тема роботи: розробка застосунку для автоматизації діяльності логістичної компанії «ТрансВей»

Керівник роботи доцент кафедри ПЗАС, к.т.н. Фаріонова Тетяна Анатоліївна
 Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р.

3. Вихідні дані по роботі:

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний слайд, 2. Актуальність розробки ПЗ, 3. Мета, задачі та об'єкт кваліфікаційної роботи, 4. Аналіз вимог до ПЗ, 5. Архітектура програмного забезпечення, 6. Проєкт пз, 6 Проєкт програмного забезпечення (DFD нульового рівня), 7. Проєкт програмного забезпечення (DFD 1 - го рівня), 8-9 Проєкт програмного забезпечення (DFD 2 - го рівня), 10. Концептуальна модель програмного забезпечення, 11 Реалізація ПЗ. Вибір засобів розробки, 12 Реалізація ПЗ. Кількісні характеристики програмного забезпечення 13-14 Результати розробки ПЗ, 15 Висновки, 16 Заклучний слайд

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Термін виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	ПП*
5.	Розробка проекту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	16.06.2025	

* - за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

Студент


 (Підпис)

Салова В.О.

(ПІБ)

Керівник роботи


 (Підпис)

Фаріонова Т.А.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена вирішенню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці застосунку для автоматизації діяльності логістичної компанії «ТрансВей».

Об'єктом кваліфікаційної роботи є процес розробки застосунку для автоматизації діяльності логістичної компанії «ТрансВей». Метою роботи є підвищення ефективності обробки логістичної інформації та зниження впливу людського чинника шляхом автоматизації основних бізнес-процесів. У ході роботи проведено аналіз предметної області, сформульовано постановку задачі на розробку застосунку для автоматизації діяльності логістичної компанії «ТрансВей» та складено технічне завдання. Було розроблено програмне забезпечення для автоматизації діяльності логістичної компанії «ТрансВей» та представлено частину коду. Проведено опис аспектів охорони праці при роботі застосунку для автоматизації діяльності логістичної компанії «ТрансВей».

Під час розробки програмного забезпечення було використано методи структурного проектування таких як DFD та ER-моделювання, інструменти для програмування (Flask, PostgreSQL, HTML/CSS, Python). Було проведено тестування програмного забезпечення.

Кваліфікаційна робота викладена на 92 сторінки друкованого тексту, містить 27 рисунків, 20 таблиць, 5 додатків та список використаних джерел із 14 найменувань.

ABSTRACT

The qualification work is devoted to solving the practical problem of software engineering, characterized by complexity and uncertainty of conditions, namely the development of an application for automating the activities of the logistics company TransWay.

The object of the qualification work is the process of developing an application for automating the activities of the logistics company TransWay. The purpose of the work is to increase the efficiency of logistics information processing and reduce the impact of the human factor by automating key business processes. In the course of the work, we analyzed the subject area, formulated the task of developing an application for automating the activities of the logistics company TransWay and drew up the terms of reference. Software for automating the activities of the logistics company TransVey was developed and part of the code was presented. A description of the aspects of labor protection in the operation of the application for automating the activities of the logistics company “TransWay” was carried out.

During the software development, structural design methods such as DFD and ER modeling, programming tools (Flask, PostgreSQL, HTML/CSS, Python) were used. The software was tested.

The qualification work is presented on 92 pages of printed text, contains 27 figures, 20 tables, 5 appendices and a list of references of 14 titles.

Зміст

ВСТУП.....	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ ДЛЯ ЛОГІСТИЧНОЇ КОМПАНІЇ «ТРАНСВЕЙ»	9
1.1 Аналіз практичної задачі інженерії програмного забезпечення	9
1.2 Аналіз існуючого програмного забезпечення та обґрунтування розробки власного ПЗ для автоматизації діяльності логістичної компанії «ТрансВей»	10
1.3 Постановка задачі на розробку програмного застосунку для автоматизації діяльності логістичної компанії «ТрансВей»	17
2 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ ЛОГІСТИЧНОЇ ДІЯЛЬНОСТІ КОМПАНІЇ «ТРАНСВЕЙ»	19
2.1 Розробка архітектури програмного забезпечення	19
2.2 Розробка проекту програмного забезпечення	22
2.2.1 Багаторівневе подання ПЗ	22
2.2.2 DFD діаграма рівня 1	29
2.2.3 DFD діаграми рівня 2	30
2.2.4 Побудова таблиці відповідності потоків на діаграмах всіх рівнів	32
2.2.5 Розробка мініспецифікацій основних функціональних сценаріїв ПЗ	39
2.2.6 Концептуальна модель програмного забезпечення	46
2.2.7 Реалізація програмного забезпечення	48
2.2.8 Реалізація модулів програмного забезпечення	49
2.3 Тестування та випробування програмного забезпечення	52
2.3.1 Підхід до тестування	52
2.3.2 Таблиця прикладів тестових випадків	53
2.3.3 Висновки щодо тестування	55
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	56
3.1 Загальний опис отриманих результатів	56
3.2 Частково реалізовані або неописані компоненти	56
3.3 Кількісні та якісні характеристики ПЗ	57
3.4 Апробація програмного забезпечення	58

3.5 Результати випробувань	58
4 ОХОРОНА ПРАЦІ У ВІДЛІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	64
4.1 Характеристика робочого місця розробника ПЗ.....	64
4.2 Шкідливі та небезпечні виробничі фактори при розробці ПЗ.....	65
4.3 Заходи щодо забезпечення охорони праці при роботі з комп'ютером...	66
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДАДАТОК А – Технічне завдання на розробку застосунку для автоматизації діяльності логістичної компанії «ТрансВей»	71
ДОДАТОК Б – Вихідний текст програмних модулів.....	76
ДОДАТОК В – Опис програмного забезпечення.....	79
ДОДАТОК Г – Інструкція користувача	82
ДОДАТОК Д – Програма та методика випробування програмного забезпечення	89

ВСТУП

Сучасна галузь логістики вимагає постійної оптимізації процесів перевезення вантажів, управління клієнтами, документообігу та аналітики. Однак, у багатьох компаніях, включно з «ТрансВей», ці процеси залишаються частково неавтоматизованими, що призводить до дублювання інформації, затримок і людських помилок.

Практична задача інженерії програмного забезпечення, що розглядається в кваліфікаційній роботі, полягає в розробці програмного забезпечення для автоматизації діяльності логістичної компанії «ТрансВей» в умовах, що характеризуються багатоваріантністю сценаріїв, змінністю даних та великою кількістю інформаційних потоків. Як зазначено в [1], розробка ПЗ передбачає побудову адаптивних і масштабованих програмних систем, здатних підтримувати комплексну взаємодію користувачів з даними.

Як показує практика на основі аналізу існуючих аналогів [2, 3], встановлено, що ці рішення або мають надлишкову функціональність, або не відповідають специфіці українського ринку. Це обґрунтовує доцільність розробки власного програмного забезпечення, повністю адаптованого під потреби компанії «ТрансВей».

Дана кваліфікаційна робота присвячена розв'язанню практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці застосунку для автоматизації діяльності логістичної компанії «ТрансВей»

Метою кваліфікаційної роботи є розробка програмного застосунку для автоматизації діяльності логістичної компанії «ТрансВей», що дозволить підвищити ефективність роботи за рахунок автоматизації основних облікових логістичною процесів, зменшити час на обробку інформації та мінімізують імовірність виникнення помилок при введенні та обробки даних.

Для досягнення поставленої мети в ході дипломного проектування необхідно виконати наступні задачі:

- проаналізувати практичну задачу інженерії програмного забезпечення для автоматизації діяльності логістичної компанії «ТрансВей»;
- проаналізувати аналоги існуючого ПЗ для автоматизації логістичної діяльності та обґрунтувати необхідність розробки застосунку для логістичної компанії «ТрансВей»;
- сформулювати постанову задачі для розробки застосунку для автоматизації діяльності логістичної компанії «ТрансВей»;
- розробити проект програмного забезпечення для автоматизації логістичної діяльності компанії «ТрансВей»;
- виконати всі етапи проектування програмного забезпечення (ескізні, технічні, робочі);
- розглянути питання охорони праці у відділі розробки програмного забезпечення;
- оформити необхідну програмну документацію.

Об'єктом є процес розробки програмного забезпечення для автоматизації бізнес-процесів у сфері логістики, а саме – логістичної компанії «ТрансВей».

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ ДЛЯ ЛОГІСТИЧНОЇ КОМПАНІЇ «ТРАНСВЕЙ»

1.1 Аналіз практичної задачі інженерії програмного забезпечення

У межах предметної галузі логістики розглядається задача автоматизації діяльності компанії «ТрансВей», яка спеціалізується на наданні послуг вантажоперевезення по Україні. Компанія виконує повний цикл логістичних операцій - від прийому замовлення до доставки вантажу кінцевому споживачеві. Основними напрямками її діяльності є планування маршрутів, керування транспортними засобами, облік вантажів, контроль термінів поставки, взаємодія з клієнтами та оформлення супровідної документації.

У сучасних умовах логістичні компанії, зокрема «ТрансВей», стикаються з низкою викликів, серед яких: велика кількість ручної роботи, що знижує загальну продуктивність; складність в обробці великого обсягу інформації; низька оперативність реагування на зміну умов доставки або маршруту; неузгодженість даних між різними підрозділами компанії; а також відсутність актуальної аналітичної інформації для ухвалення управлінських рішень.

З огляду на ці проблеми, виникає нагальна потреба у впровадженні сучасного програмного забезпечення для автоматизації основних логістичних процесів компанії. Практична задача, розв'язувана в межах цієї кваліфікаційної роботи, полягає у створенні програмного забезпечення, за допомогою якого можливе наступне:

- автоматизувати обробку інформації про клієнтів, вантажі, маршрути та транспортні засоби;
- оптимізувати процеси планування, доставки, обліку логістичних

операцій та документообігу;

- мінімізувати кількість помилок під час введення даних завдяки перевіркам та автоматичному заповненню форм;
- забезпечити користувачів (диспетчерів, водіїв, менеджерів) зручними інструментами для контролю, аналізу, звітності та ухвалення рішень в реальному часі;
- забезпечити централізований доступ до актуальної та узгодженої інформації для всіх структурних підрозділів компанії.

Очікується, що впровадження такого програмного рішення сприятиме підвищенню ефективності роботи компанії, скороченню витрат на логістичні операції, покращенню якості сервісу для клієнтів та зміцненню конкурентних позицій компанії на ринку транспортно-логістичних послуг.

1.2 Аналіз існуючого програмного забезпечення та обґрунтування розробки власного ПЗ для автоматизації діяльності логістичної компанії «ТрансВей»

Для ефективної реалізації програмного проєкту доцільно проаналізувати вже існуюче ПЗ, що призначене для автоматизації логістичних процесів. У цьому підрозділі розглядаються популярні програмні рішення, які використовуються у сфері логістики, визначаються їхні переваги та недоліки, а також обґрунтовується необхідність розробки власного програмного забезпечення, адаптованого до специфіки діяльності компанії «ТрансВей».

RePost [2] - це впровадження цифрових рішень для оптимізації процесів доставки, обробки замовлень, управління складами, маршрутами та обслуговування клієнтів. RePost є українським поштово-логістичним оператором, який надає послуги доставки документів, вантажів та посилок по Україні.

Функціональні можливості програмного забезпечення:

Цифрові сервіси для клієнтів:

- особистий кабінет та мобільний застосунок;
- онлайн-створення та відстеження експрес-накладних;
- SMS- та email-інформування клієнтів.

Інтеграція з e-commerce:

- плагіни для CMS систем (наприклад, WooCommerce, OpenCart);
- API для інтеграції з інтернет-магазинами.

Внутрішня логістика:

- автоматизоване сортування посилок;
- оптимізація маршрутів доставки за допомогою спеціалізованих програм;
- система моніторингу кур'єрів у реальному часі.

Складська логістика:

- впровадження WMS (Warehouse Management System) для обліку товарів;
- створення зонованого зберігання та автоматизованої обробки замовлень.

Екрані форми ПЗ RePost представлені на рис. 1.1-1.3 [2]

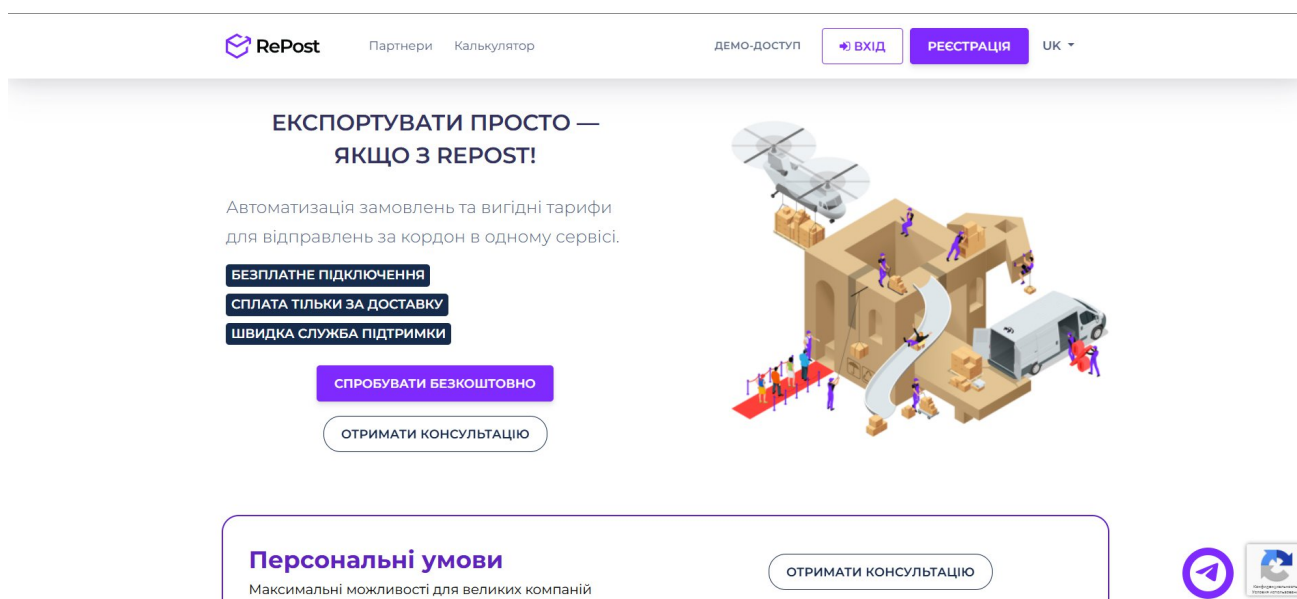


Рисунок 1.1– Головна сторінка сайту [2]

Рисунок 1.2 – Калькулятор цін [2]

ВІДГУКИ НАШИХ КЛІЄНТІВ

Сергій Лапчук
Менеджер відділу відправок
AncientSmithy

Із задоволенням порекомендую RePost всім, хто розраховує на якісний сервіс доставки. Сподобалось все, починаючи від інтуїтивного інтерфейсу сервісу до широкого вибору логістики. Заявки оброблять швидко (включно з вибором DHL кур'єра) і найголовніше, не потрібно самостійно займатися оформленням документації!

Катерина Швадрінова
Менеджер Be Woody

Сервісом користуюсь доволі тривалий час. Подобається швидкість відповідей, допомога в вирішенні всіх питань, клієнторієнтованість. Також задоволена зручним сайтом, швидко можна отримати накладну, прорахувати доставку, продивитись різні види доставок. Є синхронізація з Етсі, що дуже зручно. Можна одразу відслідкувати свою посилку,

Петюр Олександр
Партнер бренду виробів з дерева Woodzo

Repost класний і зручний сервіс. Шлях нашого бренду Woodzo на Етсі почався відносно недавно і разом з тим нам порекомендували цей сервіс для відправки наших замовлень закордон. Спочатку це здавалося чимось складним, так як багато чого незрозуміло, але менеджери Repost все чітко пояснили. Іно швидко лає.

Рисунок 1.3 – Сторінка з відгуками [2]

Переваги програмного застосунку RePost:

- підвищення швидкості обслуговування клієнтів;
- зменшення впливу людського фактору та як наслідок й помилок;
- зручний процес отримання аналітики що до вантажів та

проведених операцій;

- існує модуль оптимізації логістичних витрат.

Недоліки:

- висока вартість впровадження та супроводу ПЗ, що обмежує їх доступність для малих компаній;
- необхідність навчання персоналу для роботи з новими програмами;
- невисока надійність: можливі збої або зависання ПЗ, особливо при великому навантаженні;
- залежність від стійкості інтернет-зв'язку та надійності серверної інфраструктури;
- обмежена гнучкість у налаштуванні систем під нетипові кейси клієнтів.

SAP S/4HANA Logistics [3]. Пропонує комплексні програмні рішення для управління ланцюгами постачань, включаючи модулі для управління транспортом, складом та іншими логістичними процесами.

Функціональні можливості:

- управління ланцюгами постачання: планування, закупівлі, виробництво, доставка;
- інтеграція з фінансами: контроль витрат, бюджетування, фінансова звітність;
- аналітика в реальному часі: використання SAP HANA для обробки великих обсягів даних;
- інтерфейс SAP Fiori: зручний доступ до функцій через веб-інтерфейс.

Екрані форми програмного забезпечення SAP S/4HANA Logistics представлені на рисунках 1.4-1.6 [3]

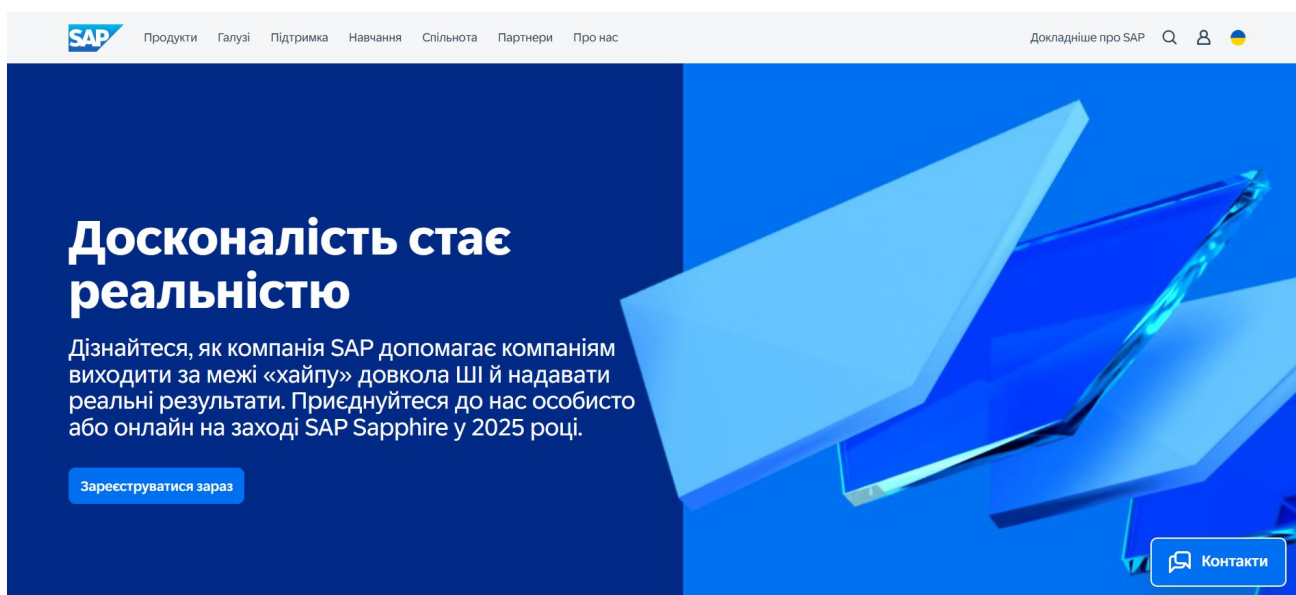


Рисунок 1.4 – Головна сторінка сайту [3]

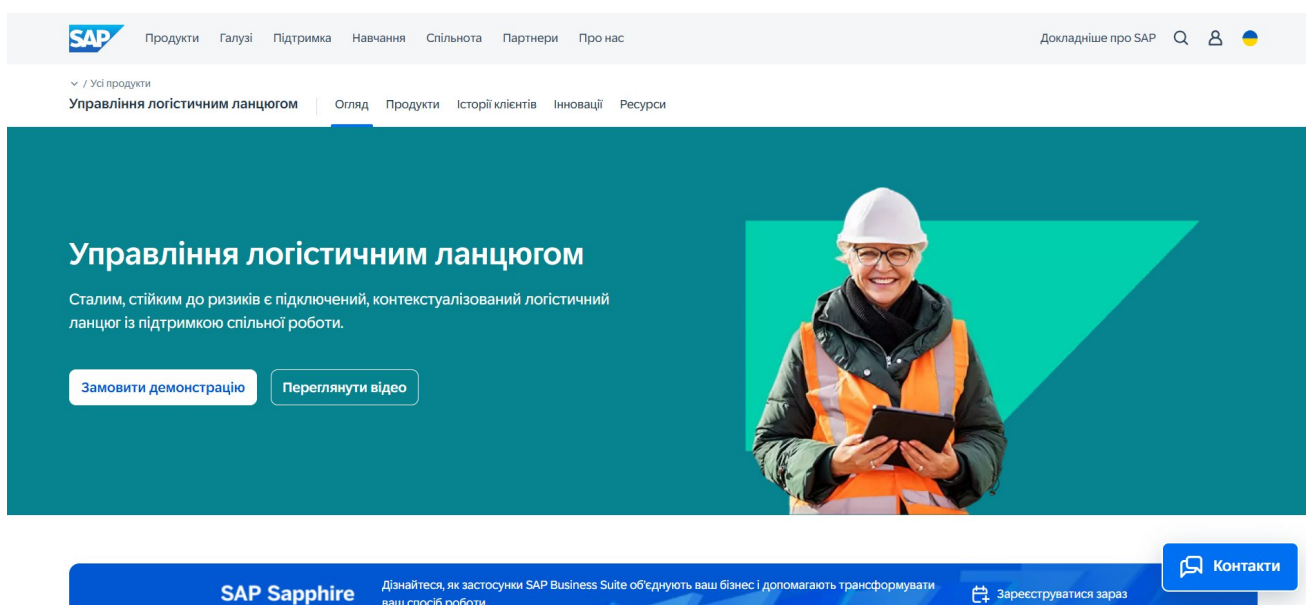


Рисунок 1.5 – Сторінка управління логістичним ланцюгом [3]

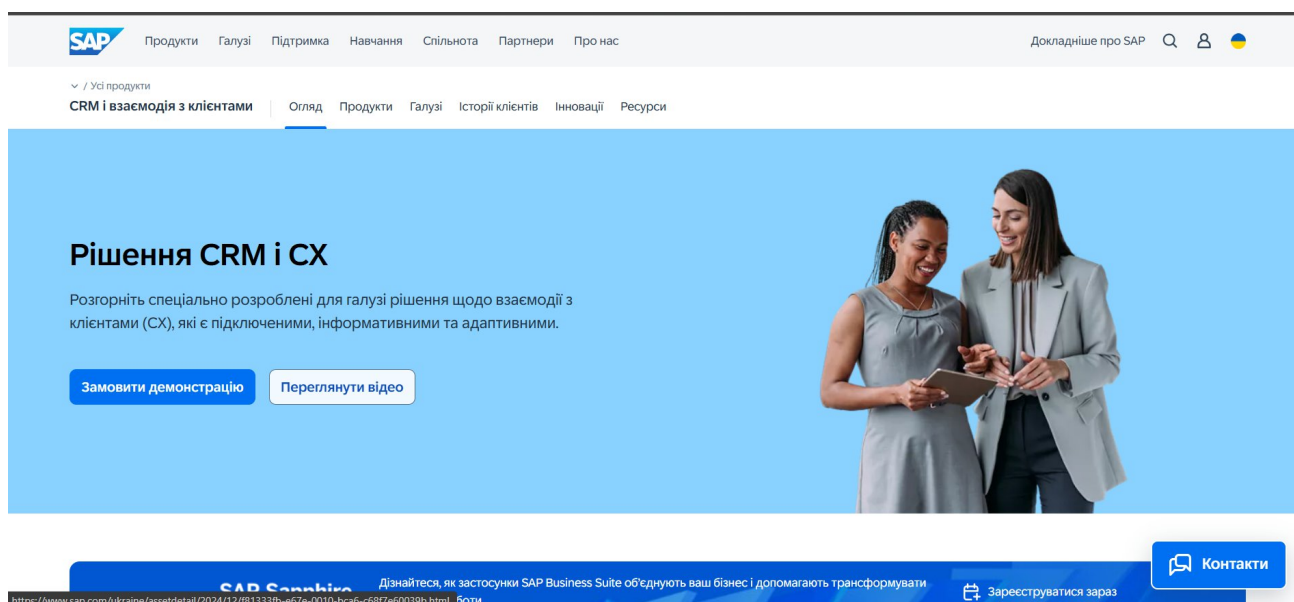


Рисунок 1.6 – Сторінка взаємодії з клієнтами [3]

Переваги:

- комплексне рішення для глобального управління логістикою;
- автоматизація всіх рівнів: від планування до реалізації ланцюгів постачання;
- надійність, масштабованість, можливість хмарного розгортання;
- підтримка глибокої аналітики та BI-інструментів.

Недоліки:

- дуже висока вартість ліцензування та впровадження;
- необхідність участі зовнішніх консультантів;
- надлишковість для малого бізнесу;
- високі вимоги до ресурсів (сервери, IT-персонал).

У ході аналізу предметної області логістики та існуючих програмних рішень було виявлено, що більшість готових ПЗ мають надлишковий або, навпаки, обмежений функціонал, не повністю відповідають потребам компанії «ТрансВей» і вимагають суттєвих ресурсів таких як:

- готові програмні рішення, мають широкий набір функцій, які часто перевищують потреби невеликої логістичної компанії. Це ускладнює

використання застосунку, підвищує витрати на навчання персоналу та потребує дорогого обслуговування;

- хмарні сервіси, часто не адаптовані до українського ринку, не підтримують локалізацію, інтеграцію з внутрішніми документами, а також мають обмежений або платний доступ до критично важливих функцій;

- впровадження чужих рішень передбачає постійну залежність від сторонніх розробників, зниження рівня безпеки даних і відсутність повного контролю над функціональністю програмної системи.

З урахуванням специфіки діяльності компанії «ТрансВей», яка вимагає:

- чіткої структуризації даних про клієнтів, вантажі та транспорт;
- обліку страхування та географічних параметрів доставки;
- зручної системи формування звітів і рахунків-фактур;
- можливості адаптації до майбутніх змін та розширення функціоналу,

розробка власного програмного забезпечення є найбільш доцільним рішенням.

Переваги розробки власного ПЗ:

- повна відповідність функціоналу реальним потребам компанії;
- гнучкість в адаптації інтерфейсу та логіки роботи;
- відсутність зайвих витрат на ліцензії та техпідтримку сторонніх рішень;

- можливість розширення та підтримки внутрішніми силами у майбутньому;

- забезпечення інформаційної безпеки та автономності компанії.

Таким чином, створення власного ПЗ дозволить компанії підвищити ефективність роботи, зменшити витрати та забезпечити надійну підтримку логістичних процесів.

1.3 Постановка задачі на розробку програмного застосунку для автоматизації діяльності логістичної компанії «ТрансВей»

Грунтуючись на результатах проведеного аналізу практичної задачі інженерії програмного забезпечення з метою підвищення автоматизації логістичних процесів компанії «ТрансВей», було прийняте рішення розробити власний програмний застосунок. Виходячи з аналізу задачі інженерії програмного забезпечення з розробки застосунку для автоматизації діяльності логістичної компанії «ТрансВей», яку має розв'язувати кваліфікаційна робота, для логістичної галузі, сформулюємо постановку задачі: розробити програмний застосунок для автоматизації діяльності логістичної компанії «ТрансВей», яке підвищить ефективність роботи компанії та буде задовольняти нижче функціональним та нефункціональним вимогам:

Функціональні вимоги до програмного забезпечення:

- додавання, редагування та видалення інформації про клієнтів, вантажі та транспорт;
- формування рахунків-фактур на основі введених даних;
- генерація звітів про перевезення (з фільтрами за періодом, клієнтом, маршрутом тощо);
- ведення історії замовлень;
- можливість залишення відгуку клієнтом.

Нефункціональні вимоги до програмного забезпечення:

- простий та інтуїтивно зрозумілий інтерфейс;
- підтримка української мови;
- час відповіді програмної системи не перевищує 1 секунди при стандартному запиті.

Вимоги до складу і параметрів технічних засобів:

- мінімальні ресурси клієнтської машини: 4 ГБ ОЗП;
- процесор з тактовою частотою 1 ГГц.

Вимоги до інформаційної та програмної сумісності:

- СУБД: SQLite / PostgreSQL;
- підтримка експорту у PDF, CSV;
- програма повинна функціонувати під управлінням ОС Windows 10+;
- підтримка браузерного доступу (опційно через вебінтерфейс).

Детальні вимоги до програмного забезпечення наведені у технічному завданні, що міститься в додатку А - Технічне завдання.

2 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ ЛОГІСТИЧНОЇ ДІЯЛЬНОСТІ КОМПАНІЇ «ТРАНСВЕЙ»

Аналіз вимог програмного забезпечення було розглянуто у попередньому розділі цієї роботи.

2.1 Розробка архітектури програмного забезпечення

Для розробки застосунку обрано багаторівневу клієнт-серверну архітектуру із чітким розподілом на рівень представлення, бізнес-логіки та доступу до даних. Такий підхід дозволяє забезпечити масштабованість, повторне використання компонентів та спрощує супровід програмної системи.

Обґрунтування вибору архітектури полягає в тому, що застосунок працює через веб-інтерфейс, що зумовлює необхідність підтримки клієнт-серверної моделі. Крім того, розділення відповідальностей за рівнями забезпечує можливість ефективного тестування та внесення змін до логіки роботи програмного застосунку без потреби втручання в інтерфейс або механізми збереження даних. Використання Flask як серверного фреймворку разом із PostgreSQL як системи управління базами даних сприяє досягненню високої продуктивності при мінімальних витратах ресурсів.

Програмне забезпечення має модульну архітектуру:

- Модуль автентифікації - забезпечує доступ лише для авторизованих користувачів.
- Модуль управління замовленнями - реєстрація, редагування та перегляд заявок.
- Модуль логістики - розрахунок оптимального маршруту.
- Модуль аналітики - формування звітів про виконані перевезення.

Діаграми компонентів надають концептуальну картину взаємодії між різними програмними системами. Можуть бути присутні як аспекти логічного,

так і фізичного моделювання. Крім того, компоненти автономні.. Вони містять конструкції будь-якої складності та є самодостатніми [4].

Архітектура програмного забезпечення - спосіб структурування програмної або обчислювальної системи, абстракція елементів застосунку на певній фазі її роботи [5].

Діаграма компонентів архітектури програмного застосунку представлена рис. 2.1:

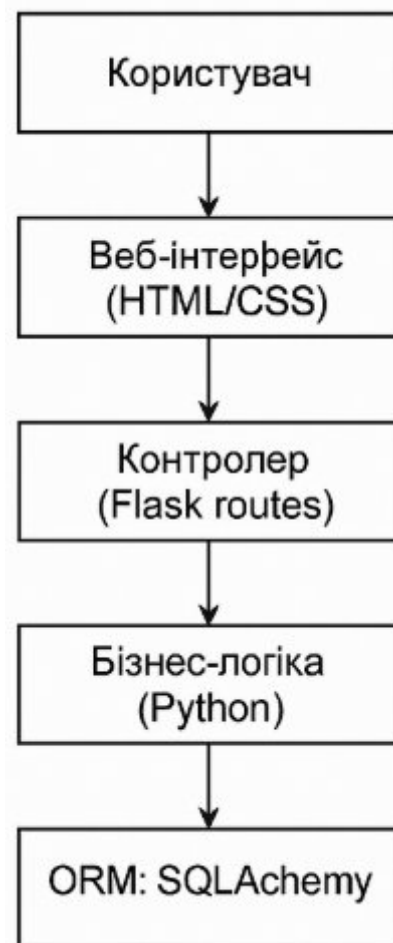


Рисунок 2.1 - Діаграма компонентів архітектури програмного застосунку

Компонент розглядається як єдине ціле, що виконує закінчену функцію і застосовується самостійно або в складі комплексу [6].

Розглянемо компоненти програмного застосунку та їх специфікація (табл. 2.1):

Таблиця 2.1 - Компоненти програмного застосунку та їх специфікація

Компонент	Призначення
1	2
UserInterface	Приймає введення від користувача, відображає дані з бекенду
FlaskController	Обробляє HTTP-запити, викликає відповідні сервіси
BusinessLogic	Реалізує логіку обробки клієнтів, вантажів, рахунків, звітів
ORM Layer	Забезпечує об'єктно-реляційне відображення бази даних
Database	Зберігає інформацію про клієнтів, вантажі, транспорт, рахунки

Діаграма розгортання - це тип діаграми, який визначає фізичне обладнання, на якому працюватиме програмний застосунок. Він також визначає, як програмне забезпечення розгортається на базовому обладнанні. Він прив'язує частини програмного забезпечення до пристрою, який збирається її виконувати [7].

Програмний застосунок розгортається на одному сервері або в хмарному середовищі. Компоненти взаємодіють між собою за допомогою HTTP/REST-запитів.

Нижче наведено діаграму розгортання програмного застосунку (рис. 2.2)

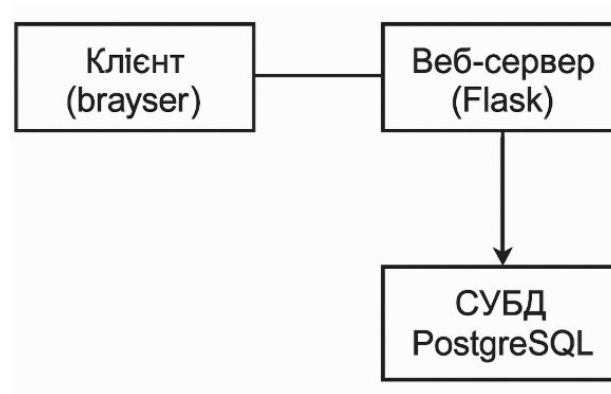


Рисунок 2.2 – Діаграма розгортання

Взаємодія компонентів відбувається таким чином: користувач взаємодіє через веб-інтерфейс (HTML-сторінки), запити передаються на сервер Flask, який викликає відповідні функції бізнес-логіки, а всі дані зберігаються в базі даних PostgreSQL за допомогою SQLAlchemy.

2.2 Розробка проекту програмного забезпечення

Проектування програмного забезпечення для автоматизації діяльності логістичної компанії «ТрансВей» виконано із застосуванням структурної методології, яка дозволяє подати програмну систему у вигляді ієрархічної сукупності взаємопов'язаних компонентів. Такий підхід забезпечує логічну прозорість, спрощує розуміння архітектури програмного забезпечення, а також сприяє перевірці відповідності вимогам, визначеним у постановці задачі.

2.2.1 Багаторівневе подання ПЗ

Контекстна діаграма нульового рівня є ключовим кроком у розробці програмного забезпечення [5,6]. Ця високорівнева діаграма допомагає зрозуміти зовнішнє середовище застосунку, зокрема акторів, що взаємодіють з програмною системою, та внутрішні та зовнішні ресурси, які вона використовує.

Контекстна діаграма нульового рівня допомагає зрозуміти, які інформаційні потоки входять до програмного застосунку від зовнішніх акторів та які вихідні потоки існують у програмній системі для взаємодії зі зовнішніми акторами. Вона також допомагає зрозуміти, які процеси та ресурси залучені до застосунку та як вони взаємодіють між собою.

Контекстна діаграма нульового рівня не надає докладного опису процесів та ресурсів, що використовуються в програмній системі. Вона лише надає загальну картину взаємодії програми з акторами та зовнішніми ресурсами. Це важливий етап у розробці програмного забезпечення, який сприяє налагодженню ефективної співпраці між програмною системою та зовнішніми акторами.

Контекстна діаграма нульового рівня є найбільш загальним типом діаграми у моделюванні бізнес-процесів. Вона відображає зовнішнє середовище, в якому діє організація, та програмне забезпечення, що взаємодіють з цим середовищем.

Контекстна діаграма представлена на рисунку 2.3

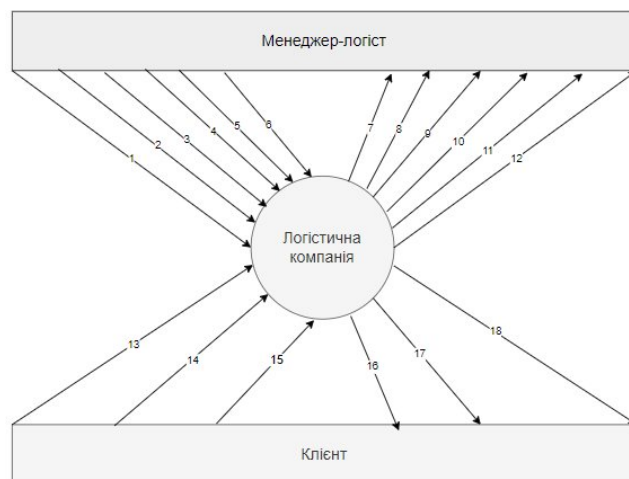


Рисунок 2.3 – Контекстна діаграма

Потоки даних є механізмами, що використовуються для моделювання передачі інформації (або фізичних компонентів) з однієї частини програмної

системи в іншу. Потoki зображуються на схемі іменованими стрілками, орієнтація яких вказує напрямok руху інформації [8].

Нижче наведено назви потоків даних контекстної діаграми:

1. Запит на обробку інформації про клієнта-відправника
2. Запит на обробку інформації про клієнта-отримувача
3. Запит на обробку інформації про вантаж клієнта-відправника
4. Запит на обробку інформації про транспорт
5. Запит на формування звітів
6. Запит на формування рахунку-фактур
7. Підтвердження обробки інформації про клієнта-відправника
8. Підтвердження обробки інформації про клієнта-отримувача
9. Підтвердження обробки інформації про вантаж клієнта-відправника
10. Підтвердження обробки інформації про транспорт
11. Звіти
12. Рахунок-фактура
13. Запит на обробку інформації про клієнта-відправника
14. Запит на обробку інформації про вантаж клієнта-відправника
15. Запит на отримання рахунку-фактури
16. Підтвердження обробки інформації про клієнта-відправника
17. Підтвердження обробки інформації про вантаж клієнта-відправника
18. Рахунок-фактура

Словник даних це структурований опис усіх інформаційних потоків, що використовуються в програмній системі або на діаграмах потоків даних (DFD). Він містить визначення кожного потоку, включно з його назвою, змістом, типами даних, форматами, джерелом та призначенням.

Словник даних контекстної діаграми представлений у таблиці 2.2

Таблиця 2.2 – Словник даних контекстної діаграми

Найменування потоку	Складові потоку
1	2
Запит на обробку інформації про клієнта-відправника	- Запит на додавання інформації про клієнта-відправника: (П.І.Б. клієнта відправника, Номер телефону відправника, Електронна пошта відправника (не обов'язково), код клієнта-відправника, інформація про документи,); - Запит на редагування інформації про клієнта-відправника: (П.І.Б. клієнта відправника, Номер телефону відправника, Електронна пошта відправника (не обов'язково), код клієнта-відправника, інформація про документи,); - Запит на видалення інформації про клієнта-відправника: (код клієнта-відправника).
Запит на обробку інформації про клієнта-отримувача.	- Запит на додавання інформації про клієнта-отримувача: (П.І.Б. клієнта отримувача, номер телефону отримувача, електронна пошта отримувача (не обов'язково), інформацію про спосіб оплати, код клієнта-отримувача); - Запит на редагування інформації про клієнта-отримувача: (П.І.Б. клієнта отримувача, номер телефону отримувача, електронна пошта отримувача (не обов'язково), інформацію про спосіб оплати, код клієнта-отримувача); - Запит на додавання відгуку про вантаж (Код вантажу, код клієнта-отримувача, код відгуку, текст з відгуком). - Запит на видалення інформації про клієнта-отримувача: (код клієнта-отримувача)
Запит на обробку інформації про вантаж	- Запит на додавання інформації про вантаж: (Код відправника, код отримувача, код вантажу, код транспорту, адреса відправлення, адреса отримання, розмір, вага, об'єм, назва, кількість, вартість перевезення, страхування вантажу, термін доставки, дата відправлення та прибуття, країна виробництва)

Продовження таблиці 2.2

1	2
	- Запит на редагування інформації про вантаж клієнта-відправника: (Код відправника, код отримувача, код вантажу, адреса відправлення, адреса отримання, розмір, вага, об'єм, назва, кількість, вартість перевезення, страхування вантажу, термін доставки, дата відправлення та прибуття, країна виробництва) - Запит на видалення інформації про вантаж клієнта-відправника: (код вантажу)
Запит на обробку інформації про транспорт, який буде використовуватись для перевезення вантажу.	- Запит на додавання інформації про транспорт: (Тип транспорту, номер транспортного засобу, код транспорту); - Запит на редагування інформації про транспорт, який буде використовуватись для перевезення вантажу: (Тип транспорту, номер транспортного засобу, код транспорту, вантажопідйомність); - Запит на додавання інформації про характеристики транспорту: (Вид палива, вантажопідйомність, втрата палива, код характеристики); - Запит на видалення інформації про транспорт: (код транспорту);
Запит на формування звітів вантажоперевезень	- Натискання кнопки “Звіт вантажоперевезень”.
Запит на формування рахунку-фактури	- Вантаж_ID

Продовження таблиці 2.2

1	2
Підтвердження обробки інформації про клієнта-відправника	Підтвердження додавання нового клієнта-відправника: - повідомлення про додавання нового клієнта-відправника. Підтвердження оновлення інформації про клієнта-відправника: - повідомлення про оновлення інформації про клієнта-відправника. Підтвердження видалення клієнта-відправника: - повідомлення про видалення клієнта-відправника.
Підтвердження обробки інформації про клієнта-отримувача	Підтвердження додавання нового клієнта-отримувача: - повідомлення про додавання нового клієнта-отримувача. Підтвердження оновлення інформації про клієнта-відправника: - повідомлення про оновлення інформації про клієнта-відправника. Підтвердження додавання відгуку про вантаж - повідомлення про додавання відгуку. Підтвердження видалення клієнта-отримувача: - повідомлення про видалення клієнта-отримувача.
Підтвердження обробки інформації про вантаж клієнта	Підтвердження додавання нового вантажу: - повідомлення про додавання нового вантажу. Підтвердження оновлення інформації про вантаж: - повідомлення про оновлення інформації про вантаж. Підтвердження видалення інформації про вантаж: - повідомлення про видалення вантажу.

Кінець таблиці 2.2

1	2
Підтвердження обробки інформації про транспорт, який буде використовувати для перевезення вантажів	Підтвердження додавання нового транспорту для перевезення вантажу: - повідомлення про вибір транспорту для перевезення вантажу. Підтвердження оновлення транспорту для перевезення вантажу: - повідомлення про оновлення транспорту для перевезення вантажу. Підтвердження додавання інформацій про характеристики транспорту: - повідомлення про характеристики транспорту Підтвердження видалення інформації про транспорт для перевезення вантажу: - повідомлення про видалення транспорту для перевезення вантажу.
Рахунок-фактура	- ПІБ клієнта-отримувача - номер рахунку для оплати - вартість перевезення - дата
віт вантажоперевезень	Формування звітів вантажоперевезень: - Клієнт-відправник - Клієнт-отримувач - Розмір - Вага - Об'єм - Назва вантажу - Кількість - Адреса відправлення - Адреса отримання - Вартість - Інформація про страхування - Термін доставки - Дата відправлення - Дата прибуття - Вид транспорту - Країна виробництва

2.2.2 DFD діаграма рівня 1

Після опису програмного застосунку в цілому проводиться розбиття її на фрагменти. Цей процес називається функціональною декомпозицією, а діаграми, які описують кожен фрагмент і взаємодію фрагментів, називаються діаграмами декомпозиції [9].

На наступному рівні деталізації було виділено такі основні процеси:

- Обробити клієнтів - відправників
- Обробити клієнтів – отримувачів
- Обробити вантаж
- Обробка інформацію про транспорт
- Формування рахунку – фактури
- Формування звітів

На рис. 2.4 представлена декомпозиція контекстної діаграми (DFD 1-го рівня)

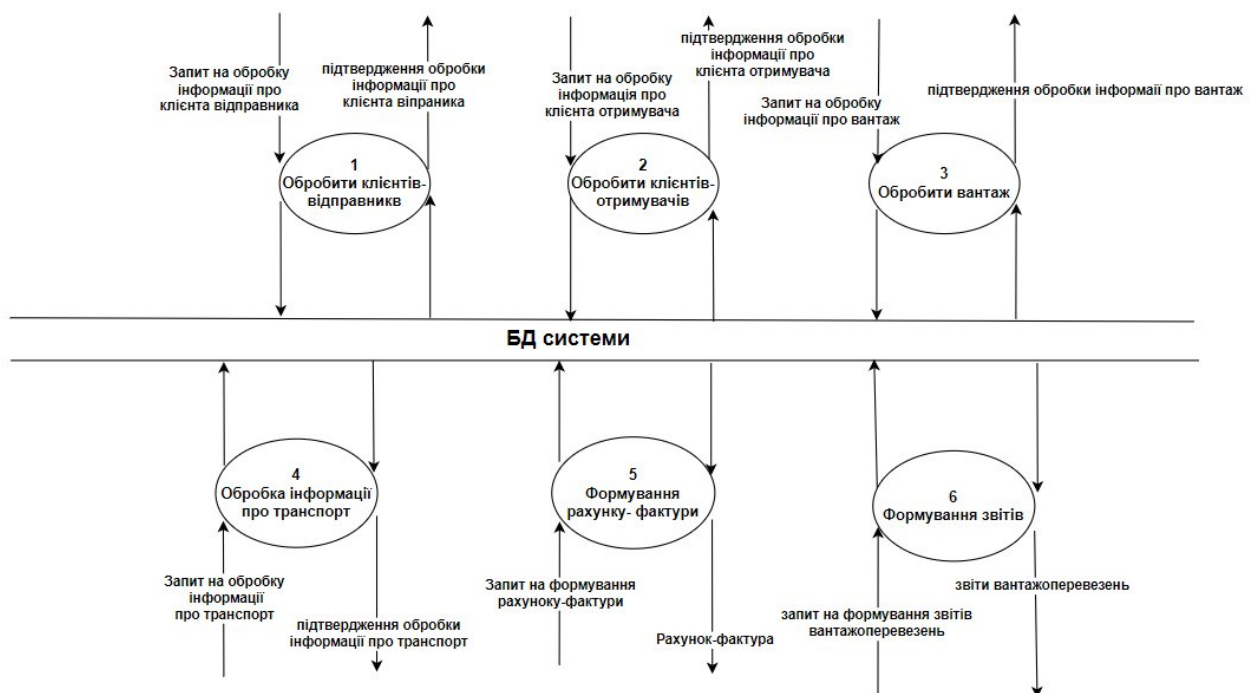


Рисунок 2.4 – Декомпозиція контекстної діаграми (DFD 1-го рівня)

2.2.3 DFD діаграми рівня 2

DFD рівня 2 - це докладне розгортання одного з процесів, зображених на рівні 1. Якщо на рівні 1 маємо загальні функціональні блоки, то рівень 2 дозволяє глибше розкрити логіку одного з них, представивши внутрішню структуру цього процесу [9].

На рівні 2 деталізуються ключові процеси. Наприклад, процес «Обробка клієнта-відправника» розбито на:

- Додати нового клієнта-відправника
- Оновити дані про клієнта-відправника
- Видалити клієнта-відправника

Цей підхід дозволяє глибше аналізувати логіку програмного застосунку.

На рис.2.5 представлено декомпозицію процесу “1.0 Обробити клієнтів-відправників”

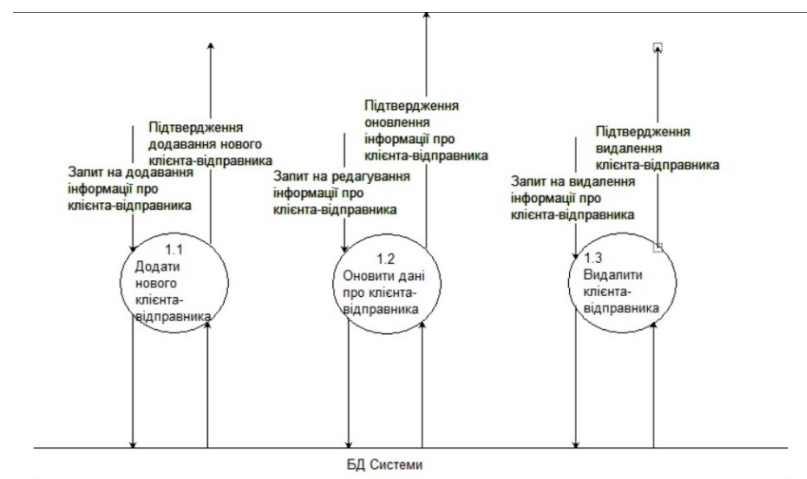


Рисунок 2.5 – DFD 2-го рівня. Декомпозиція процесу “ Обробити клієнтів-відправників ”

На рис.2.6 представлено декомпозицію процесу “2.0 Обробити клієнтів-отримувачів”

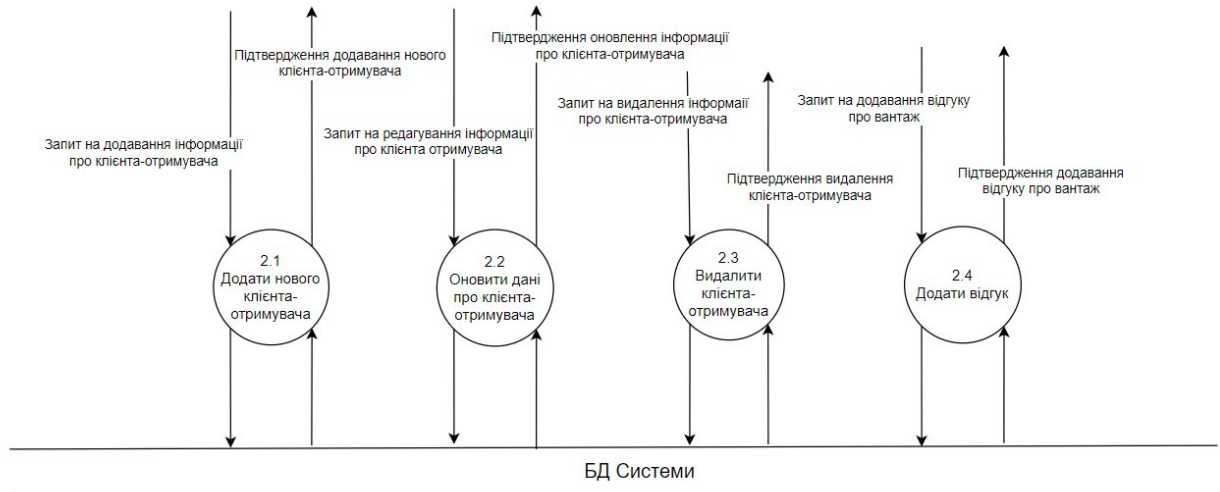


Рисунок 2.6 – DFD 2-го рівня. Декомпозиція процесу “ Обробити клієнтів-отримувачів ”

На рис.2.7 представлено декомпозицію процесу “3.0 Обробити вантаж”

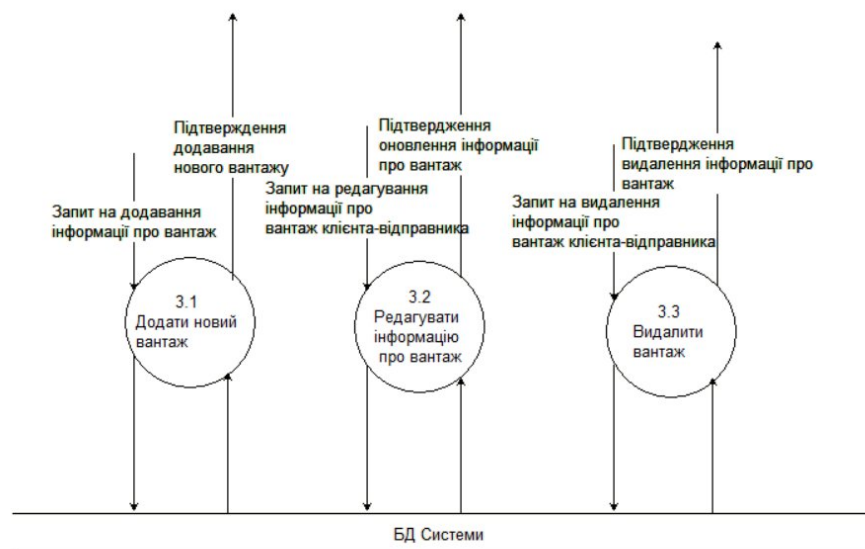


Рисунок 2.7 – DFD 2-го рівня. Декомпозиція процесу “ Обробити вантаж”

На рис.2.8 представлено декомпозицію процесу “4.0 Обробка інформації про транспорт”

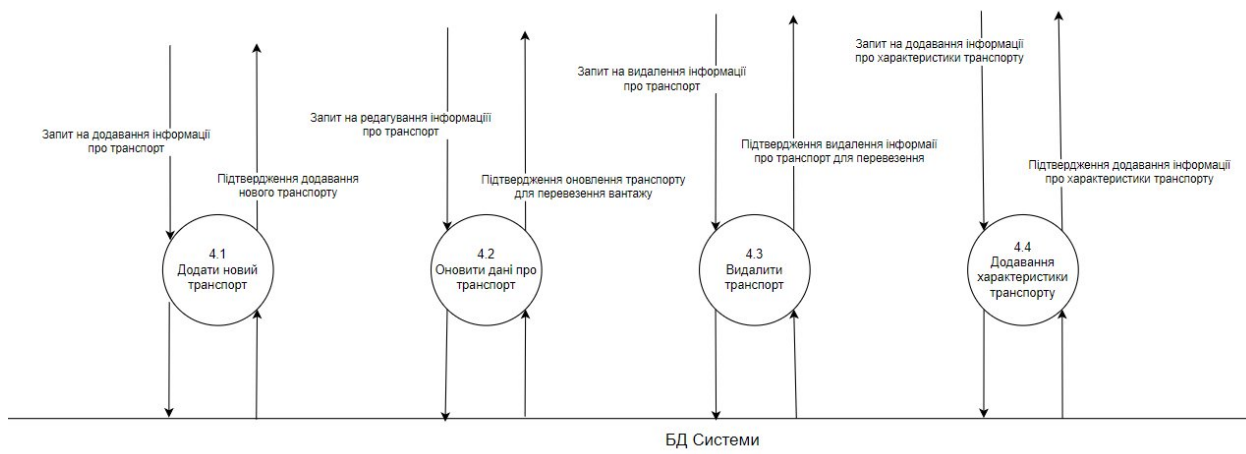


Рисунок 2.8 – DFD 2-го рівня. Декомпозиція процесу “ Обробка інформації про транспорт”

2.2.4 Побудова таблиці відповідності потоків на діаграмах всіх рівнів.

Для забезпечення логічної послідовності моделювання програмного застосунку було побудовано таблицю відповідності потоків на діаграмах усіх рівнів. Така таблиця дозволяє простежити, як інформаційні потоки, що зображені на контекстній діаграмі (0 рівень), деталізуються на діаграмах 1 та 2 рівня. Це забезпечує цілісність та узгодженість розробленої функціональної моделі програмного забезпечення. Таблиці наведені окремо для кожного потоку нульового рівня і містять відповідні потоки 1 та 2 рівнів з описом їх атрибутів. Такий підхід дозволяє зручно аналізувати і відстежувати шлях обробки інформації в програмній системі.

Нижче наведено таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про клієнта відправника» (табл. 2.3)

Таблиця 2.3 - Таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про клієнта відправника»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на обробку інформації про клієнта відправника	Запит на додавання інформації про клієнта-відправника	П.І.Б., номер телефону, email (необов'язково), код клієнта, документи
	Запит на редагування інформації про клієнта-відправника	П.І.Б., номер телефону, email, код клієнта, документи
	Запит на видалення інформації про клієнта-відправника	Код клієнта-відправника

Далі наведено таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про клієнта-отримувача» (табл. 2.4)

Таблиця 2.4 - Таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про клієнта-отримувача»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на обробку інформації про клієнта-отримувача	Запит на додавання інформації про клієнта-отримувача	П.І.Б., номер телефону, email, спосіб оплати, код
	Запит на редагування інформації	П.І.Б., номер телефону, email, спосіб оплати, код
	Запит на видалення інформації	Код
	Запит на додавання відгуку про вантаж	Код вантажу, код клієнта, код відгуку, текст

Нижче наведено таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про вантаж клієнта-відправника» (табл. 2.5)

Таблиця 2.5 - Таблицю відповідності для DFD – 0 рівня Запит на обробку інформації про вантаж клієнта-відправника»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на обробку інформації про вантаж	Запит на додавання інформації про вантаж	Код відправника/отримувача, вантажу, транспорту; адреси, розмір, вага, об'єм, назва, кількість, вартість, страхування, терміни, дати, країна
	Запит на редагування інформації про вантаж	ті самі атрибути
	Запит на видалення інформації	Код вантажу

Також наведено таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про транспорт» (табл. 2.6)

Таблиця 2.6 - Таблицю відповідності для DFD – 0 рівня «Запит на обробку інформації про транспорт»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на обробку інформації про транспорт	Запит на додавання інформації про транспорт	Тип, номер, код транспорту
	Запит на редагування інформації	Тип, номер, код, вантажопідйомність
	Запит на видалення інформації	Код транспорту

Кінець таблиці 2.6

1	2	3
	Запит на додавання характеристик транспорту	Вид палива, вантажопідйомність, витрата палива, код характеристики

Нижче наведено таблицю відповідності для DFD – 0 рівня «Запит на формування рахунку-фактури» (табл. 2.7.)

Таблиця 2.7 - Таблицю відповідності для DFD – 0 рівня «Запит на формування рахунку-фактури»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на формування рахунку-фактури	–	Вантаж_ID

Наведено ще одну таблицю відповідності для DFD – 0 рівня «Запит на формування звітів вантажоперевезень» (табл. 2.8)

Таблиця 2.8 - Таблицю відповідності для DFD – 0 рівня «Запит на формування звітів вантажоперевезень»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на формування звітів вантажоперевезень	–	Клієнти, розміри, вага, об'єм, адреси, назва, кількість, вартість, страхування, терміни, дати, транспорт, країна

Наведено таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про клієнта-відправника» (табл. 2.9)

Таблиця 2.9 - Таблицю відповідності для DFD – 0 рівня «Запит на формування звітів вантажоперевезень»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Підтвердження обробки інформації про клієнта-відправника	Підтвердження додавання нового клієнта-відправника	Повідомлення про додавання
	Підтвердження оновлення інформації про клієнта-відправника	Повідомлення про оновлення
	Підтвердження видалення клієнта-відправника	Повідомлення про видалення

Нижче наведено таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про клієнта-отримувача» (табл. 2.10)

Таблиця 2.10 - Таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про клієнта-отримувача»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Підтвердження обробки інформації про клієнта-отримувача	Підтвердження додавання нового клієнта-отримувача	Повідомлення про додавання
	Підтвердження оновлення інформації про клієнта-отримувача	Повідомлення про оновлення
	Підтвердження видалення клієнта-отримувача	Повідомлення про видалення

Нижче наведено таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про вантаж клієнта-відправника» (табл. 2.11)

Таблиця 2.11 – Таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про вантаж клієнта-відправника»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Підтвердження обробки інформації про вантаж	Підтвердження додавання нового вантажу	Повідомлення про додавання
	Підтвердження оновлення інформації про вантаж	Повідомлення про оновлення
	Підтвердження видалення інформації про вантаж	Повідомлення про видалення

Наведено таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про транспорт» (табл. 2.12)

Таблиця 2.12 - Таблицю відповідності для DFD – 0 рівня «Підтвердження обробки інформації про транспорт»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Підтвердження обробки інформації про транспорт	Підтвердження додавання нового транспорту	Повідомлення про вибір транспорту
	Підтвердження оновлення транспорту	Повідомлення про оновлення
	Підтвердження видалення транспорту	Повідомлення про видалення
	Підтвердження додавання характеристик транспорту	Повідомлення про характеристики транспорту

Нижче наведено таблицю відповідності для DFD – 0 рівня «Запит на отримання рахунку-фактури» (табл. 2.13)

Таблиця 2.13 - Таблицю відповідності для DFD – 0 рівня «Запит на отримання рахунку-фактури»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Запит на отримання рахунку-фактури	–	ID вантажу

Наведено таблицю відповідності для DFD – 0 рівня «Рахунок-фактура» (табл. 2.14)

Таблиця 2.14 - Таблицю відповідності для DFD – 0 рівня «Рахунок-фактура»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Рахунок-фактура	–	ПІБ клієнта-отримувача, номер рахунку, вартість, дата

Нижче наведено таблицю відповідності для DFD – 0 рівня «Звіт вантажоперевезень» (табл. 2.15)

Таблиця 2.15 - Таблицю відповідності для DFD – 0 рівня «Звіт вантажоперевезень»

Потік 1 рівня	Потік 2 рівня	Атрибути потоку
1	2	3
Формування звіту	–	Клієнт-відправник, Клієнт-отримувач, розмір, вага, об'єм, назва, кількість, адреси, вартість, страхування, терміни, дати, транспорт, країна

2.2.5 Розробка мініспецифікацій основних функціональних сценаріїв ПЗ

На цьому етапі проєктування сформовано мініспецифікації основних функціональних сценаріїв, які описують поведінку програмного забезпечення в типових ситуаціях. Кожна мініспецифікація включає призначення функції, перелік вхідних та вихідних даних, умови виконання і процедуру обробки. Це дозволяє забезпечити структуровану реалізацію функцій, підвищити якість тестування та забезпечити відповідність програмного забезпечення вимогам користувача.

Додавання нового клієнта-відправника (процес 1.1)

Призначення: Додати нового клієнта-відправника до бази даних.

Вхідні дані:

- П.І.Б. клієнта-відправника;
- Номер телефону відправника;
- Ідентифікатор відправника;
- Інформація про документи;

Вихідні дані: Повідомлення про додавання нового клієнта-відправника.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Додати".

Процедура обробки: При введенні даних правильного формату присвоюється унікальний номер запису і дані заносяться в таблицю "Клієнти-відправники" бази даних "Логістична компанія".

Оновлення даних клієнта-відправника (процес 1.2)

Призначення: Оновити дані клієнта-відправника в базі даних.

Вхідні дані:

- П.І.Б. клієнта-відправника;
- Номер телефону відправника;
- Ідентифікатор відправника;
- Інформація про документи;

Вихідні дані: Повідомлення про оновлення даних клієнта-відправника.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Зберегти".

Процедура обробки: При введенні даних правильного формату оновлюються необхідні дані про клієнта-відправника в таблиці "Клієнти-відправники" бази даних "Логістична компанія".

Видалення клієнта-відправника (процес 1.3)

Призначення: Видалити клієнта-відправника з бази даних.

Вхідні дані:

- Ідентифікатор клієнта-відправника.

Вихідні дані: Повідомлення про видалення клієнта-відправника.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Видалити".

Процедура обробки: При введенні даних правильного формату відбувається видалення даних про клієнта-відправника з таблиці "Клієнти-відправники" бази даних "Логістична компанія".

Додавання нового клієнта-отримувача (процес 2.1)

Призначення: Додати нового клієнта-отримувача до бази даних.

Вхідні дані:

- П.І.Б. клієнта-отримувача;

- Ідентифікатор отримувача;
- Адреса отримувача;
- Контактна інформація;
- Інформація про спосіб оплати;

Вихідні дані: Повідомлення про додавання нового клієнта-отримувача.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Додати" .

Процедура обробки: При введенні даних правильного формату присвоюється унікальний номер запису і дані заносяться в таблицю "Клієнти-отримувачі" бази даних "Логістична компанія".

Оновлення даних клієнта-отримувача (процес 2.2)

Призначення: Оновити дані клієнта-отримувача в базі даних.

Вхідні дані:

- П.І.Б. клієнта-отримувача;
- Ідентифікатор отримувача;
- Адреса отримувача;
- Контактна інформація;
- Інформація про спосіб оплати;

Вихідні дані: Повідомлення про оновлення даних клієнта-отримувача.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Зберегти".

Процедура обробки: При введенні даних правильного формату оновлюються необхідні дані про клієнта-отримувача в таблиці "Клієнти-отримувачі" бази даних "Логістична компанія".

Видалення клієнта-отримувача (процес 2.3)

Призначення: Видалити клієнта-отримувача з бази даних.

Вхідні дані:

- Ідентифікатор клієнта-отримувача.

Вихідні дані: Повідомлення про видалення клієнта-отримувача.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Видалити".

Процедура обробки: При введенні даних правильного формату відбувається видалення даних про клієнта-отримувача з таблиці "Відгук" бази даних "Логістична компанія".

Додавання відгуку (процес 2.4)

Призначення: Додати відгук до бази даних

Вхідні дані:

- Ідентифікатор клієнта-отримувача
- Ідентифікатор вантажу
- Ідентифікатор відгуку
- Текст з відгуком

Вихідні дані: Повідомлення про додавання нового відгуку

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Додати відгук".

Процедура обробки: При введенні даних правильного формату присвоюється унікальний текст вантажу і дані заносяться в таблицю "Вантажі" бази даних "Логістична компанія".

Додавання нового вантажу (процес 3.1)

Призначення: Додати новий вантаж до бази даних.

Вхідні дані:

- Ідентифікатор клієнта-відправника;
- Ідентифікатор клієнта-отримувача;
- Ідентифікатор транспорту;
- Інформація про адресу відправлення;
- Інформація про адресу отримання;
- Ідентифікатор вантажу;
- Вартість;
- Вага;
- Об'єм;

- Назва;
- Кількість;
- Терміни доставки;
- Дата відправлення;
- Дата прибуття;
- Інформація про страхування;
- Країна виробництва;

Вихідні дані: Повідомлення про додавання нового вантажу.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Додати".

Процедура обробки: При введенні даних правильного формату присвоюється унікальний номер вантажу і дані заносяться в таблицю "Вантажі" бази даних "Логістична компанія".

Редагування інформації про вантаж (процес 3.2)

Призначення: Оновити дані про вантаж в базі даних.

Вхідні дані:

- Ідентифікатор клієнта-відправника;
- Ідентифікатор клієнта-отримувача;
- Ідентифікатор транспорту;
- Інформація про адресу відправлення;
- Інформація про адресу отримання;
- Ідентифікатор вантажу;
- Вартість;
- Вага;
- Об'єм;
- Назва;
- Кількість;
- Терміни доставки;
- Дата відправлення;
- Дата прибуття;

- Інформація про страхування;
- Країна виробництва;

Вихідні дані: Повідомлення про оновлення даних про вантаж.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Зберегти".

Процедура обробки: При введенні даних правильного формату оновлюються необхідні дані про вантаж в таблиці "Вантажі" бази даних "Логістична компанія".

Видалення вантажу (процес 3.3)

Призначення: Видалити вантаж з бази даних.

Вхідні дані:

- Ідентифікатор вантажу.

Вихідні дані: Повідомлення про видалення вантажу.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Видалити".

Процедура обробки: При введенні даних правильного формату відбувається видалення даних про вантаж з таблиці "Вантажі" бази даних "Логістична компанія".

Додавання нового транспорту (процес 4.1)

Призначення: Додати інформацію про транспорт до бази даних.

Вхідні дані:

- Тип транспорту;
- Ідентифікатор транспорту;
- Номер транспорту;

Вихідні дані: Присвоєння певного вантажу певному транспорту.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Додати".

Процедура обробки: При введенні даних правильного формату присвоюється унікальний номер запису і дані заносяться в таблицю "Транспорт" бази даних "Логістична компанія".

Оновлення даних транспорту (процес 4.2)

Призначення: Оновити дані транспорт в базі даних.

Вхідні дані:

- Тип транспорту;
- Ідентифікатор транспорту;
- Номер транспорту;

Вихідні дані: Повідомлення про оновлення даних транспорту.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Зберегти".

Процедура обробки: При введенні даних правильного формату оновлюються необхідні дані про транспорт в таблиці "Транспорт" бази даних "Логістична компанія".

Видалення транспорту (процес 4.3)

Призначення: Видалити дані про транспорт з бази даних.

Вхідні дані:

- Ідентифікатор транспорту;

Вихідні дані: Повідомлення про видалення даних транспорту.

Умови виконання: Введення даних правильного формату в форму та натискання кнопки "Видалити".

Процедура обробки: При введенні даних правильного формату відбувається видалення даних про транспорт з таблиці "Транспорт" бази даних "Логістична компанія".

Додавання характеристик транспорту (процес 4.4)

Призначення: Додати дані про характеристики транспорту.

Вхідні дані:

- Ідентифікатор характеристики
- Витрата палива
- Вантажопідйомність
- Вид палива

Вихідні дані: Повідомлення про додавання інформації характеристики транспорту

Умови виконання: Введення даних правильного формату в форму та натискання кнопки «Додати».

Процедура обробки: При введенні даних правильного формату відбувається додавання даних характеристики транспорту в таблицю "Характеристики" бази даних "Логістична компанія".

2.2.6 Концептуальна модель програмного забезпечення

Діаграма зв'язку сутностей, також відома як ERD є концептуальною моделлю програмного забезпечення. Це діаграма, яка відображає зв'язок наборів сутностей, що зберігаються в базі даних. Іншими словами, ER-діаграми допомагають пояснити логічну структуру баз даних. ER-діаграми створюються на основі трьох основних понять: сутності, атрибути та зв'язки [10].

Для зберігання інформації розроблено ER-діаграму (рис. 2.9)

2.2.7 Реалізація програмного забезпечення

Розробка програмного забезпечення здійснювалася з використанням перевірених, ефективних та взаємодоповнюючих технологій, що забезпечують гнучкість, масштабованість і простоту підтримки застосунку.

- Python (Flask) був обраний як основна мова та фреймворк серверної частини завдяки своїй лаконічності, активній спільноті розробників і широкій підтримці бібліотек для роботи з базами даних, REST API, безпекою тощо. Flask, як мікрофреймворк, дозволяє гнучко формувати структуру проєкту і не нав'язує жорстких обмежень, що особливо важливо для проєктів середнього масштабу.

- HTML + CSS + Bootstrap використовувалися для створення адаптивного та інтуїтивно зрозумілого інтерфейсу користувача. Bootstrap забезпечує швидке макетування сторінок, кросбраузерну сумісність та підтримку адаптивного дизайну без необхідності ручного налаштування стилів для кожного пристрою.

- PostgreSQL обрано як систему керування базами даних через її стабільність, підтримку транзакцій, розширену SQL-функціональність, хорошу масштабованість і відкритий вихідний код. Вона ідеально підходить для логістичних систем, де важливі цілісність і надійність даних.

- SQLAlchemy як ORM (Object-Relational Mapping) використовується для зручної та безпечної взаємодії між Python-кодом і PostgreSQL. Вона абстрагує SQL-запити у вигляді об'єктів, дозволяючи зменшити ймовірність помилок у запитах та підвищити читабельність і супровідність коду.

Обраний стек реалізує класичну трирівневу архітектуру, що включає клієнтський рівень (веб-інтерфейс), прикладний рівень (логіка обробки даних) та рівень збереження (база даних). Така архітектура є гнучкою до змін, дозволяє легко масштабувати програмний застосунок, спрощує тестування і обслуговування окремих компонентів.

2.2.8 Реалізація модулів програмного забезпечення

У процесі розробки застосунку для автоматизації діяльності логістичної компанії було реалізовано низку ключових модулів, які забезпечують виконання основних бізнес-процесів програмної системи. Кожен модуль відповідає за окрему групу функцій, має чітко визначену зону відповідальності, реалізований із використанням фреймворку Flask та взаємодіє з базою даних PostgreSQL за допомогою ORM-бібліотеки SQLAlchemy.

Серед основних модулів, що входять до складу застосунку, виділяються:

- Модуль реєстрації та авторизації, що забезпечує безпечний доступ до застосунку для зареєстрованих користувачів. У модулі використано механізми хешування паролів на основі бібліотеки `werkzeug.security`.

Модуль реєстрації та авторизації представлено у лістингу 1

Лістинг 1 модуль реєстрації та авторизації

```
from werkzeug.security import generate_password_hash,
check_password_hash

@app.route('/register', methods=['POST'])
def register():
    password = request.form['password']
    hashed_password = generate_password_hash(password,
method='sha256')
    new_user = User(email=request.form['email'],
password=hashed_password)
    db.session.add(new_user)
    db.session.commit()
    return redirect('/login')
```

Модуль керування клієнтами, який реалізує повний набір CRUD-операцій для клієнтів-відправників і клієнтів-отримувачів.

Модуль керування клієнтами представлений у лістингу 2

Лістинг 2 модуль керування клієнтами

```
@app.route('/clients/add', methods=['POST'])
def add_client():
    new_client = Client(
        name=request.form['name'],
        phone=request.form['phone'],
        email=request.form['email']
    )
    db.session.add(new_client)
    db.session.commit()
    return redirect('/clients')
```

Модуль керування вантажами, який дозволяє додавати, редагувати та зберігати інформацію про вантажі з прив'язкою до відповідних клієнтів та транспортних засобів.

Модуль керування вантажами представлений у лістингу 3

Лістинг 3 модуль керування вантажами

```
@app.route('/cargo/add', methods=['POST'])
def add_cargo():
    new_cargo = Cargo(
        sender_id=request.form['sender_id'],
        receiver_id=request.form['receiver_id'],
        description=request.form['description'],
        weight=float(request.form['weight']),
```

```

        value=float(request.form['value'])
    )
    db.session.add(new_cargo)
    db.session.commit()
    return redirect('/cargo')

```

Модуль генерації рахунків-фактур, що формує PDF-документи на основі інформації про доставку конкретного вантажу.

Модуль генерації рахунків-фактур представлений у лістингу 4

Лістинг 4 модуль генерації рахунків-фактур

```

from fpdf import FPDF

def generate_invoice(cargo_id):
    cargo = Cargo.query.get(cargo_id)
    pdf = FPDF()
    pdf.add_page()
    pdf.set_font("Arial", size=12)
    pdf.cell(200, 10, txt=f"Invoice for Cargo ID {cargo.id}",
ln=1)
    pdf.cell(200, 10, txt=f"Receiver: {cargo.receiver.name}",
ln=2)
    pdf.output(f"invoice_{cargo.id}.pdf")

```

Модуль звітності, який забезпечує вибірку даних за часовими або іншими параметрами й формує звіти у зручному вигляді (HTML або CSV).

Модуль звітності представлений у лістингу 5

Лістинг 5 модуль звітності

```
@app.route('/report')
def report():
    start_date = request.args.get('start')
    end_date = request.args.get('end')
    cargo_list = Cargo.query.filter(Cargo.date >= start_date,
    Cargo.date <= end_date).all()
    return render_template('report.html', cargo=cargo_list)
```

Кодова реалізація модулів є лаконічною, структурованою та відповідає принципам MVC, що полегшує її підтримку й розвиток.

2.3 Тестування та випробування програмного забезпечення

2.3.1 Підхід до тестування

Для перевірки якості програмного забезпечення застосовувався комбінований підхід до тестування [11], що включав:

- модульне тестування (unit testing) - тестування окремих функцій і класів;
- інтеграційне тестування - перевірка взаємодії між модулями (наприклад, форма додавання вантажу та запис у базу даних);
- функціональне тестування - перевірка відповідності застосунку вимогам користувача;
- системне тестування - перевірка роботи всієї програмної системи як єдиного цілого;
- регресійне тестування - виконувалося після внесення змін у код для перевірки, що існуюча функціональність не порушена.

Тестування виконувалося вручну та автоматизовано. Для автоматизованих перевірок використовувався фреймворк pytest (для Python), а також вбудовані можливості Postman для тестування API-запитів.

2.3.2 Таблиця прикладів тестових випадків

Для перевірки коректності роботи основних функціональних компонентів програмного забезпечення було розроблено тестові випадки, які охоплюють як позитивні, так і негативні сценарії взаємодії з застосунком. Тестування дозволяє виявити помилки, оцінити відповідність фактичної поведінки очікуваним результатам та забезпечити стабільну роботу застосунку в різних умовах експлуатації.

У таблиці наведено приклади тестових випадків із зазначенням мети тесту, вхідних даних, очікуваного результату та фактичного результату після виконання тесту. Такий формат дозволяє ефективно проводити перевірку і документувати результати для подальшого аналізу та виправлення.

Нижче наведено таблицю прикладів тестових випадків (табл. 2.16)

Таблиця 2.16 – Приклади тестових випадків

№	Дія	Очікуваний результат	Результат
1	2	3	4
1	Реєстрація нового клієнта	Клієнта додано до бази даних, повідомлення про успіх	Клієнт з'явився у БД, повідомлення відображено
2	Вхід користувача з неправильним паролем	Повідомлення про помилку авторизації	Авторизація заблокована, повідомлення виведено
3	Додавання нового вантажу	Запис з'являється в базі, повідомлення про успішне додавання	Дані додано, повідомлення з'явилося на екрані

Кінець таблиці 2.16

4	Генерація рахунку-фактури	Створено PDF-файл із рахунком	Файл PDF сформовано, збережено у системі
5	Формування звіту за період	Виведення таблиці з вантажами, які відповідають умовам	Таблиця з результатами відображена
6	Видалення неіснуючого клієнта	Повідомлення про помилку або ніяких змін	Повідомлення про помилку виведено
7	Додавання відгуку без тексту	Повідомлення про помилку заповнення форми	Форма не прийнята, попередження виведено

Нижче наведено таблицю прикладів тестових випадків з некоректними даними (табл. 2.17)

Таблиця 2.17 – Приклади тестових випадків з некоректними даними

№	Дія	Очікуваний результат	Результат
1	2	3	4
1	Додавання клієнта з порожніми полями	Повідомлення про помилку заповнення	Повідомлення з'являється
2	Додавання вантажу з нечисловою вагою	Повідомлення про помилку типу даних	Помилка показана
3	Додавання вантажу з від'ємною вартістю	Повідомлення про помилку валідації	Помилка показана

Кінець таблиці 2.17

1	2	3	4
4	Генерація рахунку-фактури для неіснуючого вантажу	Повідомлення про відсутність запису	Повідомлення з'являється
5	Додавання транспорту без номера	Повідомлення про помилку	Повідомлення з'являється
6	Введення некоректного email	Повідомлення про недійсний формат email	Помилка валідації
7	Завантаження звіту без вибору діапазону дат	Повідомлення про помилку	Помилка з'являється

2.3.3 Висновки щодо тестування

Усі критично важливі функції було протестовано, помилки виявлені та усунені. Програмне забезпечення стабільно працює відповідно до технічного завдання. Тестування підтвердило:

- відповідність функціональних вимог;
- коректність роботи модулів при реальних сценаріях;
- стійкість до некоректного введення даних;
- стабільність при базовому навантаженні.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В результаті виконання кваліфікаційної роботи було розроблено програмне забезпечення для автоматизації логістичної діяльності компанії «ТрансВей». Застосунок відповідає поставленим вимогам і охоплює повний цикл обробки логістичних даних - від реєстрації клієнтів і вантажів до формування звітів та рахунків-фактур.

3.1 Загальний опис отриманих результатів

Розроблено веб-застосунок із наступною функціональністю:

- додавання, редагування та видалення інформації про клієнтів, вантажі та транспорт;
- формування рахунків-фактур на основі введених даних;
- генерація звітів про перевезення (з фільтрами за періодом, клієнтом, маршрутом тощо);
- ведення історії замовлень;
- можливість залишення відгуку клієнтом.

Застосунок має модульну структуру, що забезпечує можливість подальшого розширення.

3.2 Частково реалізовані або неописані компоненти

Через обмеження обсягу основних розділів, нижче наводяться функції, що були реалізовані, але не отримали повного опису:

- резервне копіювання БД - вручну, через адміністраторську панель;
- адаптивна верстка - забезпечує коректну роботу на екранах мобільних пристроїв;
- збереження звітів у CSV - експорт таблиць за фільтрами у форматі .csv.

3.3 Кількісні та якісні характеристики ПЗ

Для оцінки ефективності розробленого програмного забезпечення було проаналізовано його кількісні та якісні характеристики. Кількісні показники відображають обсяг реалізованої функціональності, продуктивність програмного забезпечення, розміри коду, а також використання ресурсів. Якісні характеристики дозволяють оцінити зручність користування, надійність, адаптивність та відповідність вимогам користувача.

Аналіз цих характеристик є важливим етапом, що дозволяє зробити висновки щодо рівня зрілості застосунку, його готовності до експлуатації, а також виявити напрями можливого вдосконалення.

Нижче наведено таблицю характеристик програмного забезпечення (табл. 3.1).

Таблиця 3.1 – Характеристика програмного забезпечення

Характеристика	Значення
1	2
Кількість рядків вихідного коду	~3 500 (HTML/CSS/JS - 1 200, Python/Flask - 2 300)
Розмір виконуваного проекту	7.4 МБ (архівована версія з усіма залежностями)
Розмір бази даних (початково)	~2 МБ при 200 тестових записах
Час запуску	<1 сек (локальний сервер, Flask Debug mode)
Середній час відповіді програмної системи	0.2–0.5 сек при локальному запиті

Кінець таблиці 3.1

1	2
Ресурси клієнтської машини	Windows 10+, ≥ 4 ГБ RAM, ≥ 1 ГГц CPU
Серверні вимоги (тестово)	Python 3.11, PostgreSQL або SQLite, 512 МБ RAM
Підтримка форматів	HTML, CSV, PDF, JSON
Продуктивність	Програмна система обробляє $\sim 20\text{--}30$ запитів на хвилину без втрати стабільності

3.4 Апробація програмного забезпечення

Програмне забезпечення було апробовано на тестовому середовищі, змодельованому відповідно до умов логістичної компанії «ТрансВей». Для апробації використовувалися вигадані, але реалістичні дані - 200+ записів про клієнтів і вантажі.

Результати апробації:

- працездатність підтверджена на типовому офісному ПК;
- всі основні модулі функціонують коректно;
- час обробки запиту - в межах нормативних вимог;
- користувачі (5 тестових осіб) оцінили інтерфейс як "інтуїтивно зрозумілий".

3.5 Результати випробувань

Програмний застосунок «ТрансВей» було піддане випробуванням згідно програми та методикою випробувань наведених у додатку Г. Нижче наведені результати випробувань

На рисунках 3.1 – 3.6 продемонстровано випробування

На початковій сторінці вас вітає привітання. Тут ви можете обрати одну з трьох кнопок (Профіль, Зробити замовлення, Для менеджера логіста).

Головна сторінка web-додатку представлена на рисунку 3.1:

Ласкаво просимо!

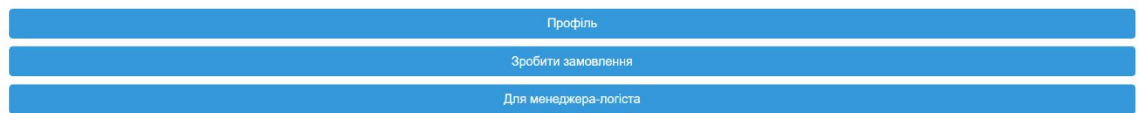
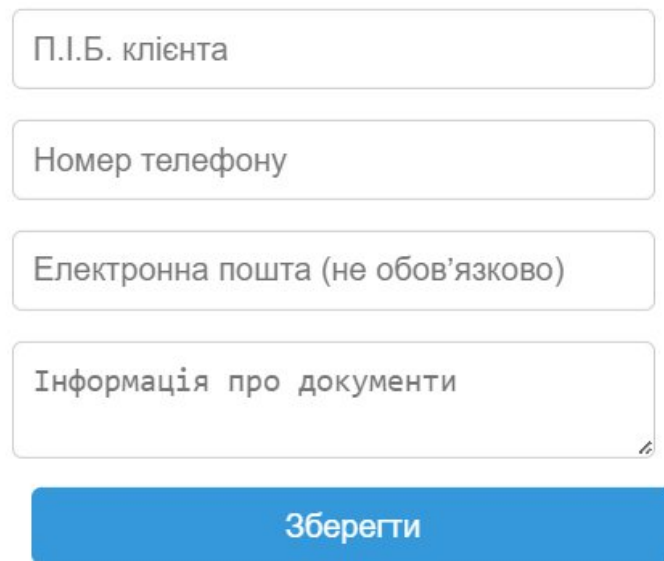


Рисунок 3.1 - Головна сторінка

Кнопка «Профіль» виведе вікно в якому можна ввести особисту інформацію це зображено на рисунку 3.2

Редагування профілю



П.І.Б. клієнта

Номер телефону

Електронна пошта (не обов'язково)

Інформація про документи

Зберегти

Рисунок 3.2 - Редагування профілю

Кнопка «Зробити замовлення» виведе вікно в якому можна ввести інформацію про вантаж це зображено на рисунку 3.3

Зробити замовлення

Адреса відправки
Адреса доставки
Розмір
Вага
Об'єм
Назва вантажу
Кількість
Вартість
Страхування
Термін доставки
Дата відправлення
Дата прибуття
Країна виробництва
Ок

Рисунок 3.3 – Зробити замовлення

Кнопки «Для менеджера логіста» виведе таблицю у якому можна побачити інформацію, яку клієнт вписував у вікні зробити замовлення (рис. 3.3). Це можна побачити на рисунку 3.4

Таблиця для менеджера-логіста

Код вантажу	Код клієнта-отримувача	Код клієнта-відправника	Код транспорту
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Рисунок 3.4 - Таблиця «Для менеджера-логіста»

Кнопки «Рахунок-фактура» виведе таблицю у якому можна побачити інформацію за замовлення вантажу. Це можна побачити на рисунку 3.5

Рахунок-фактура	
ПІБ клієнта-отримувача	Клименко Вадим Олександрович
Номер рахунку для оплати	UA533220010000026209310230139
Вартість перевезення	500 грн
Дата	20.12

Рисунок 3.5 – Таблиця «Рахунок-фактура»

Кнопки «Відгуки» виведе таблицю у якому можна побачити інформацію про відгуки до вантажу. Це можна побачити на рисунку 3.6

Відгуки

Отримувач	Клименко Вадим Олександрович
Назва товару	Масажер для ніг
Відправник	Салов Владислав Олегович
Відгук	Товар прийшов, все чудово

Рисунок 3.6 - Таблиця «Відгуки»

4 ОХОРОНА ПРАЦІ У ВІДЛІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Охорона праці - це комплексна система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів, спрямованих на збереження життя, здоров'я, працездатності й продуктивності працівників у процесі виконання ними трудових обов'язків. У сфері інформаційних технологій охорона праці набуває особливої специфіки, оскільки основні ризики пов'язані не зі шкідливими виробничими умовами в традиційному розумінні, а з тривалим статичним навантаженням, психоемоційним напруженням, а також ергономічними особливостями робочого середовища.

У контексті розробки програмного забезпечення для логістичної компанії «ТрансВей», охорона праці відіграє ключову роль. Працівники, що займаються програмуванням, проєктуванням, тестуванням та супроводом програмного продукту, виконують значну частину своїх обов'язків у сидячому положенні, за монітором комп'ютера. Такий характер роботи потребує ретельного дотримання санітарно-гігієнічних та ергономічних норм для запобігання хронічному перенапруженню, зору, опорно-рухового апарату та нервової системи.

4.1 Характеристика робочого місця розробника ПЗ

Робоче місце програміста, незалежно від того, чи працює він в офісі, чи віддалено, має бути обладнане відповідно до встановлених норм і стандартів. До основних елементів робочого місця належать:

- комп'ютерне обладнання: монітор з достатньою діагоналлю та регулюванням яскравості, клавіатура, миша;
- ергономічні меблі: регульований стілець з підтримкою попереку, зручний стіл на відповідній висоті;

- джерела освітлення: природне (вікна) та штучне (настільні або стельові лампи з нейтральним світлом);
- засоби комунікації та підключення до мережі Інтернет;
- засоби для підтримання температурного та вологісного режимів (кондиціонери, зволожувачі повітря).

Облаштування робочого місця має відповідати вимогам нормативних документів:

- ДСТУ EN ISO 9241 [12] (Ергономічні вимоги до офісної роботи з відеотерміналами);
- ДБН В.2.2-10-2001 [13] (Будинки і споруди. Будинки та приміщення для постійного перебування людей);
- Санітарні правила 3.3.2.007-98 (Гігієнічні вимоги до ПЕОМ та організації праці користувачів) [14].

4.2 Шкідливі та небезпечні виробничі фактори при розробці ПЗ

Розробка програмного забезпечення передбачає вплив низки потенційно шкідливих та небезпечних чинників, які можуть негативно вплинути на здоров'я працівника:

Фізичні фактори:

- тривале перебування у статичному положенні (сидячи);
- недостатнє або занадто яскраве освітлення, яке створює напруження для очей;
- шумове навантаження, особливо в офісах з відкритим плануванням (open space).

Психофізіологічні фактори:

- високий рівень концентрації уваги протягом тривалого часу;
- робота з великими обсягами інформації та кодом;
- стресові ситуації, пов'язані з дедлайнами, помилками або технічними збоями.

Хімічні та електромагнітні фактори:

- можливий вплив електромагнітного випромінювання від моніторів і мережевого обладнання (хоча в межах допустимих норм);
- недостатня вентиляція в приміщенні може призводити до накопичення шкідливих речовин.

4.3 Заходи щодо забезпечення охорони праці при роботі з комп'ютером

Для мінімізації впливу шкідливих чинників і забезпечення безпечних умов праці необхідно впровадити низку заходів:

Організаційні заходи:

- дотримання режиму праці та відпочинку: рекомендовані перерви - 10–15 хвилин після кожної години роботи за ПК;
- чергування розумових та рутинних завдань (програмування, перегляд документації, тестування);
- регулярне проходження інструктажів та навчання з охорони праці, пожежної безпеки та надання першої допомоги.

Санітарно-гігієнічні заходи [14]:

- підтримання температури в межах 20–24°C;
- освітлення з яскравістю не менше 300–500 лк, бажано з можливістю регулювання;
- регулярне провітрювання приміщення, підтримання вологості на рівні 40–60%;
- облаштування робочої зони в місцях, захищених від прямих сонячних променів та відблисків.

Ергономічні заходи:

- встановлення монітора на відстані 60–70 см від очей, верхній край екрана має бути на рівні очей;
- використання офісних крісел з можливістю регулювання висоти та кута нахилу спинки;
- правильне розташування клавіатури і миші - на одному рівні з руками, з можливістю підтримки зап'ясть.

Психологічна підтримка та профілактика емоційного вигорання:

- формування позитивної атмосфери в команді;
- забезпечення розумного навантаження та уникнення понаднормової роботи;
- надання працівникам можливості працювати за гнучким графіком або дистанційно;
- мотиваційні заходи, тренінги з тайм-менеджменту, ментального здоров'я.

Урахування вимог охорони праці є необхідною умовою стабільного функціонування ІТ-підрозділів у логістичних компаніях, таких як «ТрансВей». Якісно організоване робоче середовище не лише знижує ризики професійних захворювань, а й сприяє підвищенню продуктивності праці, зниженню кількості помилок у програмному коді та формує відповідальне ставлення працівників до власного здоров'я та результатів праці.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто основної мети - розроблено застосунок для автоматизації діяльності логістичної компанії «ТрансВей», який спрямований на підвищення ефективності обробки логістичних процесів, зниження впливу людського чинника та оптимізацію внутрішніх бізнес-процесів.

У результаті роботи:

- проаналізовано практичну задачу інженерії програмного забезпечення для автоматизації діяльності логістичної компанії «ТрансВей»;
- проаналізовано аналоги існуючого ПЗ для автоматизації логістичної діяльності та обґрунтувати необхідність розробки застосунку для логістичної компанії «ТрансВей»;
- сформульовано постанову задачі для розробки застосунку для автоматизації діяльності логістичної компанії «ТрансВей»;
- розроблено проект програмного забезпечення для автоматизації логістичної діяльності компанії «ТрансВей»;
- виконано всі етапи проектування програмного забезпечення (ескізні, технічні, робочі);
- розглянуто питання охорони праці у відділі розробки програмного забезпечення;
- оформлена необхідна програмна документація.

Таким чином, робота досягла поставленої мети, всі заплановані завдання були реалізовані, що свідчить про успішне завершення кваліфікаційної роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISO/IEC/IEEE 29148:2018. Systems and software engineering. Life cycle processes. Requirements engineering. [Чинний від 2018-11-28]. Вид. Офіц. Швейцарія, 2018. 9с. (Інформація та документація).
2. Офіційний сайт RePost. URL: <https://repost.global/> (дата звернення: 25.04.2025).
3. Офіційний сайт SAP Logistics. URL : <https://www.sap.com/products/scm/supply-chain-logistics.html> (дата звернення: 25.04.2025).
4. Діаграма компонентів UML. URL: <https://surl.lu/ktqbus> (дата звернення: 15.05.2025).
5. Архітектура програмного забезпечення. URL: <https://surl.lu/cezffj> (дата звернення: 15.05.2025).
6. Програмне забезпечення. URL: <https://surl.lu/zbairo> (дата звернення: 15.05.2025).
7. Діаграма розгортання. URL: <https://www.guru99.com/uk/deployment-diagram-uml-example.html> (дата звернення: 15.05.2025).
8. Потоки даних діаграми. URL: <https://is.gd/Pgb2Ws> (дата звернення: 15.05.2025).
9. Діаграма декомпозиції. URL: <https://is.gd/epEnW3> (дата звернення: 17.05.2025).
10. Модель діаграми зв'язків сутностей (ER) із прикладом СУБД. URL: <https://www.guru99.com/uk/er-diagram-tutorial-dbms.html> (дата звернення: 17.05.2025).
11. Поляков Д.О. Тестування програмного забезпечення: методи і засоби. К.: вид. Ліра-К, 2020. 208 с.
12. ДСТУ EN ISO 9241-210:2019. Ергономіка взаємодії людини і системи. Частина 210. Орієнтоване на користувача проектування

інтерактивних систем. [Чинний від 2020-01-01]. Вид. Офіц. Київ: ДП «УкрНДНЦ», 2020. 36 с.

13. ДБН В.2.2-10-2001. Приміщення житлові та громадські. Основи проектування. [Чинний від 2001-06-04]. Вид. Офіц. Київ, 2001. 171с. (Інформація та документація).

14. ДсанПіН 3.3.2.007-98. Санітарні правила і норми роботи з ПК.[Чинний від 1998-12-10]. Вид. Офіц. Київ: Міністерство охорони здоров'я України, 1998. 22с. (Інформація та документація).

ДАДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку програмного застосунку для автоматизації діяльності
логістичної компанії «ТрансВей»»

А.1 Вступ

Програмне забезпечення призначена для підвищення ефективності роботи компанії «ТрансВей» щодо обробки замовлень, обліку вантажів, контролю доставки та формування фінансової звітності в компанії, що здійснює вантажні перевезення.

А.2 Підстави для розробки

Завдання на завдання на кваліфікаційну роботу практику видане кафедрою ПЗАС.

А.3 Призначення розробки

Розробка призначена для автоматизації процесів логістичної компанії: обліку клієнтів, обробки заявок, управління транспортними засобами, формування звітів і рахунків-фактур.

А.4 Вимоги до програми або програмного виробу

А.4.1 Вимоги до функціональних характеристик

- додавання, редагування та видалення інформації про клієнтів, вантажі та транспорт;
- формування рахунків-фактур на основі введених даних;
- генерація звітів про перевезення (з фільтрами за періодом,

клієнтом, маршрутом тощо);

- ведення історії замовлень;
- можливість залишення відгуку клієнтом.

A.4.2 Вимоги до надійності

- час відповіді застосунку не перевищує 1 секунди при стандартному запиті;

- перевірка коректності вхідних даних;
- можливість резервного копіювання бази даних;
- обробка помилок введення/збереження даних.

A.4.3 Умови експлуатації:

- умови експлуатації програми збігаються з умовами експлуатації технічних засобів на який встановлено програмний застосунок;
- програмний продукт розрахований на користувача із середнім рівнем комп'ютерної грамотності.

A.4.4 Вимоги до складу і параметрів технічних засобів:

- мінімальні ресурси клієнтської машини: 4 ГБ ОЗП;
- процесор з тактовою частотою 1 ГГц.
-

A.4.5 Вимоги до інформаційної та програмної сумісності:

- СУБД: SQLite / PostgreSQL;
- підтримка експорту у PDF, CSV;
- програма повинна функціонувати під управлінням ОС Windows 10+;
- підтримка браузерного доступу (опційно через вебінтерфейс).

A4.6 Вимоги до маркування та пакування

Вимогу до маркування та пакування не висуваються.

A.4.7 Вимоги до транспортування і зберігання

Зберігання файлів розробки на корпоративному сервері або в хмарних сховищах із резервним копіюванням.

A.4.8 Спеціальні вимоги

Спеціальних вимог не передбачається.

A.5 Вимоги до програмної документації:

- Технічне завдання;
- Опис програми;
- Інструкція користувача;
- Програма та методика випробувань;
- Тексти програмних модулів.

A.6 Техніко-економічні показники

Техніко-економічні показники не розраховуються.

А.7 Стадії і етапи розробки

Стадія	Етапи робіт	Термін виконання
Технічне завдання	Збір та аналіз вимог, документування вимог.	15.03.2025
Ескізний проект	Розробка діаграми варіантів використання, розробка специфікацій, розробка діаграм діяльності.	28.03.2025
Технічний проект	Розробка контекстної діаграми, ER-діаграми, DFD – 1 рівня, DFD – 2 рівня	25.04.2025
Робочий проект	Вибір засобів розробки та мов програмування, розробка програмного коду, програмування, тестування, документація	01.04.2025

А.8 Порядок контролю і приймання

На кожному етапі створення програмного забезпечення контролюють процеси аналізу та проектування його складових з урахуванням вимог, викладених у технічному завданні. Кожна фаза розробки має бути представлена у визначені терміни та погоджена із замовником.

Приймання виконується за затвердженою методикою випробувань і програмою тестування, а результати фіксують у протоколі проведення

перевірок. Якщо під час приймання програмного продукту виявлено помилки, складають відповідний акт, у якому їх описують. Цей документ підписують представники замовника й розробника, а затверджують керівники обох сторін. Розробник зобов'язаний упродовж не більше двох тижнів виправити всі виявлені недоліки та повідомити замовника про готовність до повторної перевірки.

ДОДАТОК Б

Вихідні тексти програмних модулів

Б.1 Модуль реєстрації користувача (Flask, Python)

```
from werkzeug.security import generate_password_hash
from flask import request, redirect
from models import User, db

@app.route('/register', methods=['POST'])
def register():
    # Отримання пароля з форми
    password = request.form['password']
    # Хешування пароля для збереження в базі
    hashed_password = generate_password_hash(password,
method='sha256')
    # Створення нового користувача з email та захищеним
паролем
    new_user = User(email=request.form['email'],
password=hashed_password)
    db.session.add(new_user)
    db.session.commit()
    return redirect('/login') # Перенаправлення до сторінки
входу
```

Б.2 Модуль керування клієнтами

```
@app.route('/clients/add', methods=['POST'])
def add_client():
    # Створення нового клієнта на основі даних із форми
    new_client = Client(
        name=request.form['name'],
        phone=request.form['phone'],
```

```

        email=request.form['email']
    )
    db.session.add(new_client)
    db.session.commit()
    return redirect('/clients')

```

Б.3 Модуль додавання вантажу

```

@app.route('/cargo/add', methods=['POST'])
def add_cargo():
    # Додавання нового вантажу з формою, що включає
    ідентифікатори клієнтів і характеристики
    new_cargo = Cargo(
        sender_id=request.form['sender_id'],
        receiver_id=request.form['receiver_id'],
        weight=float(request.form['weight']),
        value=float(request.form['value'])
    )
    db.session.add(new_cargo)
    db.session.commit()
    return redirect('/cargo')

```

Б.4 Генерація рахунку-фактури у PDF

```

from fpdf import FPDF

def generate_invoice(cargo_id):
    cargo = Cargo.query.get(cargo_id)
    pdf = FPDF()
    pdf.add_page()
    pdf.set_font("Arial", size=12)
    pdf.cell(200, 10, txt=f"Invoice for Cargo ID
{cargo.id}", ln=1)

```

```

        pdf.cell(200, 10, txt=f"Receiver:
{cargo.receiver.name}", ln=2)
        pdf.output(f"invoice_{cargo.id}.pdf") # Збереження PDF-
файлу

```

Б.5 Модуль формування звітів

```

@app.route('/report')
def report():
    start_date = request.args.get('start')
    end_date = request.args.get('end')
    # Отримання вантажів, які відповідають діапазону дат
    cargo_list = Cargo.query.filter(Cargo.date >=
start_date, Cargo.date <= end_date).all()
    return render_template('report.html', cargo=cargo_list)

```

ДОДАТОК В

Опис програмного забезпечення

В.1. Загальні відомості

В.1.1 Позначення та найменування програми

Найменування: Застосунок для автоматизації діяльності логістичної компанії

Позначення: TransWayLogisticsApp

В.1.2 Програмне забезпечення, необхідне для функціонування програми

Застосунок є кросплатформним веб-застосунком, що працює у середовищі браузера та серверної частини на Flask. Для його функціонування необхідна операційна система:

- Windows 10 або вище;
- Linux з ядром 4.x або вище;
- Mac OS не нижче версії 10.12.
- Також потрібна наявність:
- інтерпретатора Python (версії не нижче 3.8);
- СУБД PostgreSQL або SQLite;
- веб-браузера (Google Chrome, Mozilla Firefox, Safari, тощо).

В.1.3 Мови програмування

Основні мови програмування, використані при реалізації:

- Python (бекенд, серверна логіка);
- HTML/CSS (представлення);

- JavaScript (клієнтська взаємодія).

В.2 Функціональне призначення

В.2.1 Класи вирішуваних завдань і призначення програми

Програмне забезпечення призначене для автоматизації бізнес-процесів логістичної компанії та реалізує такі функції:

- Додавання, редагування, видалення даних про клієнтів, вантажі, транспорт;
- Генерація рахунків-фактур;
- Формування звітів вантажоперевезень;
- Управління маршрутами доставки;
- Збереження історії замовлень;
- Можливість залишення клієнтських відгуків.

В.2.2 Функціональні обмеження

- Мінімальні апаратні ресурси клієнтського пристрою: 4 ГБ оперативної пам'яті, 1 ГГц процесор;
- Час відповіді на стандартний запит не повинен перевищувати 1 секунди;
- Застосунок функціонує лише за наявності доступу до мережі Інтернет (для клієнт-серверної взаємодії).

В.3 Опис логічної структури

Програмне забезпечення побудовано за принципами багаторівневої клієнт-серверної архітектури із застосуванням шаблону MVC:

- Model (Модель) - реалізована за допомогою ORM SQLAlchemy, відповідає за доступ до бази даних;
 - View (Подання) - HTML/CSS-сторінки, що відображають дані для користувача;
 - Controller (Контролер) - реалізований у Flask, відповідає за обробку HTTP-запитів, керування логікою взаємодії між Model і View.
- Основні модулі:
 - Модуль автентифікації;
 - Модуль клієнтів;
 - Модуль обробки вантажів;
 - Модуль управління транспортом;
 - Модуль аналітики та звітності.

В.4 Технічні засоби та програмні засоби

Апаратні засоби:

- ПК або ноутбук з процесором від 1 ГГц;
 - Оперативна пам'ять від 4 ГБ;
 - Жорсткий диск із вільним місцем не менше 500 МБ.
- ##### Програмні засоби:
- Операційна система: Windows 10+, Linux 4.x+, Mac OS 10.12+;
 - СУБД PostgreSQL або SQLite;
 - Інтерпретатор Python 3.8+;
 - Веб-браузер (для клієнта).

ДОДАТОК Г

Інструкція користувача програмного забезпечення

Г.1 Призначення програми

Функціональне призначення

Програмний продукт призначений для автоматизації облікових та логістичних процесів у компанії «ТрансВей», яка спеціалізується на перевезенні вантажів по території України. Основна мета - підвищення ефективності обробки логістичної інформації, зменшення людського чинника та забезпечення зручного управління даними про клієнтів, вантажі, транспортні засоби, рахунки-фактури та звіти.

Експлуатаційне призначення

Застосунок орієнтовано на використання співробітниками логістичної компанії «ТрансВей», зокрема менеджерами, диспетчерами, адміністраторами та бухгалтерами. Програма покликана полегшити щоденну роботу з обліком вантажоперевезень, документообігом та формуванням звітної документації. Програмне забезпечення може бути використана в комерційній діяльності компаній малого та середнього бізнесу у сфері логістики.

Склад функцій

Програма забезпечує виконання таких функцій:

- реєстрація та авторизація користувачів;
- введення, редагування та видалення інформації про клієнтів-відправників та клієнтів-отримувачів;
- введення, редагування та видалення інформації про вантажі;
- обробка даних про транспортні засоби та їх характеристики;
- формування рахунків-фактур на основі введених даних;
- формування звітів про вантажоперевезення з фільтрацією;

- експорт документів у форматах PDF та CSV;
- збереження відгуків клієнтів;
- інтеграція з email для надсилання рахунків;
- друк рахунків-фактур.

Г.2 Умови виконання програми

Г.2.1 Вимоги до технічних засобів

- Персональний комп'ютер або ноутбук;
- Процесор з тактовою частотою від 1 ГГц і вище;
- Оперативна пам'ять – не менше 4 ГБ;
- Вільне місце на диску – не менше 200 МБ;
- Наявність підключення до локального або глобального серверу БД (PostgreSQL або SQLite).

Г. 2.2 Програмне забезпечення, необхідне для функціонування програми

Операційна система:

- Windows 10 або новіше;

Інше програмне забезпечення:

- Python 3.11+;
- PostgreSQL або SQLite;
- Браузер (Google Chrome, Mozilla Firefox, Edge) - для доступу до інтерфейсу;
- PDF-рідер для перегляду рахунків-фактур.

Запуск програми відбувається через локальний або серверний Flask-додаток з інтерфейсом, доступним через браузер.

Г.2.3 Кліматичні умови експлуатації

Експлуатація програмного продукту має здійснюватися у стандартних офісних умовах:

- Температурний режим: від +10°C до +30°C;
- Вологість: не вище 80%;
- Освітлення робочого місця повинно відповідати нормам ДСТУ EN ISO 9241.

Г.3 Інструкція користувача

На початковій сторінці вас вітає привітання. Тут ви можете обрати одну з трьох кнопок (Профіль, Зробити замовлення, Для менеджера логіста).

Головна сторінка web-додатку представлена на рисунку Г.1:

Ласкаво просимо!

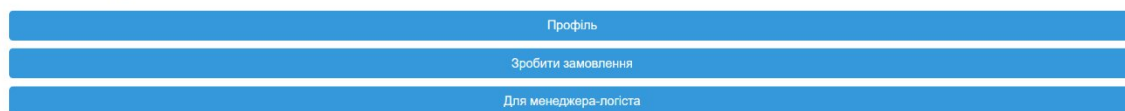
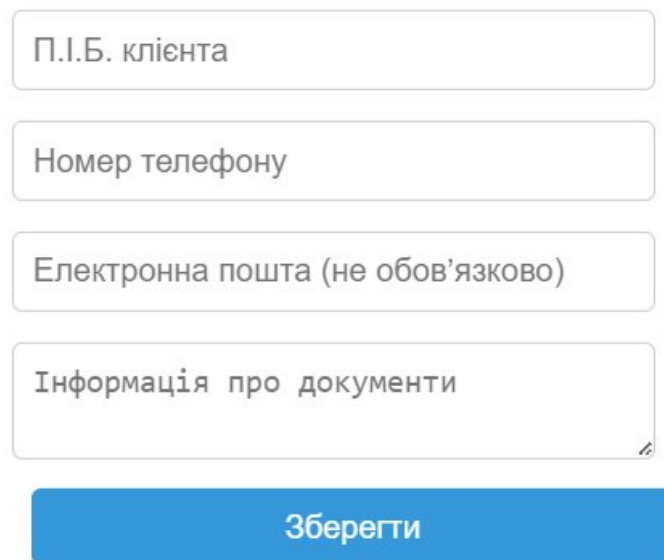


Рисунок Г.1 - Головна сторінка

Кнопка «Профіль» виведе вікно в якому можна ввести особисту інформацію інформація це зображено на рисунку Г.2

Редагування профілю



П.І.Б. клієнта

Номер телефону

Електронна пошта (не обов'язково)

Інформація про документи

Зберегти

Рисунок Г.2 - Редагування профілю

Кнопка «Зробити замовлення» виведе вікно в якому можна ввести інформацію про вантаж це зображено на рисунку Г.3

Зробити замовлення

Адреса відправки
Адреса доставки
Розмір
Вага
Об'єм
Назва вантажу
Кількість
Вартість
Страхування
Термін доставки
Дата відправлення
Дата прибуття
Країна виробництва
Ок

Рисунок Г.3 – Зробити замовлення

Кнопки «Для менеджера логіста» виведе таблицю у якому можна побачити інформацію, яку клієнт вписував у вікні зробити замовлення (рис. Г.3). Це можна побачити на рисунку Г.4

Таблиця для менеджера-логіста

Код вантажу	Код клієнта-отримувача	Код клієнта-відправника	Код транспорту
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Рисунок Г.4 - Таблиця «Для менеджера-логіста»

Кнопки «Рахунок-фактура» виведе таблицю у якому можна побачити інформацію за замовлення вантажу. Це можна побачити на рисунку Г.5

Рахунок-фактура	
ПІБ клієнта-отримувача	Клименко Вадим Олександрович
Номер рахунку для оплати	UA533220010000026209310230139
Вартість перевезення	500 грн
Дата	20.12

Рисунок Г.5 – Таблиця «Рахунок-фактура»

Кнопки «Відгуки» виведе таблицю у якому можна побачити інформацію про відгуки до вантажу. Це можна побачити на рисунку Г.6

Відгуки

Отримувач	Клименко Вадим Олександрович
Назва товару	Масажер для ніг
Відправник	Салов Владислав Олегович
Відгук	Товар прийшов, все чудово

Рисунок Г.6 - Таблиця «Відгуки»

ДОДАТОК Д

Програма та методика випробувань ПЗ

Д.1 Об'єкт випробування

Об'єктом випробування є програмне забезпечення «TransWayApp» - застосунок для автоматизації діяльності логістичної компанії «ТрансВей». Програмне забезпечення розроблене з метою поліпшення ефективності обробки замовлень, керування логістичними операціями та зниження впливу людського чинника.

Д.2 Мета випробування

Метою випробування є перевірка відповідності функціональних можливостей програмного забезпечення вимогам технічного завдання, а також надійності його роботи в типовому середовищі експлуатації.

Основні цілі випробування:

- перевірка роботи CRUD-операцій із даними клієнтів, вантажів та транспорту.
- перевірка формування рахунків-фактур.
- перевірка генерації звітів.
- перевірка стабільності роботи інтерфейсу.
- перевірка швидкодії застосунку при базових запитах.

Д.3 Вимоги до програмного забезпечення

Програмне забезпечення повинно відповідати функціональним та нефункціональним вимогам, викладеним у технічному завданні (див. Додаток

А). Зокрема, час відповіді на типові запити не повинен перевищувати 1 секунди.

Д.4 Вимоги до програмної документації

Документація повинна містити:

- текст програми;
- опис архітектури;
- інструкцію користувача;
- програму та методику випробувань;
- технічне завдання.

Д.5 Засоби і порядок випробувань

Д.5.1 Технічні засоби

- CPU: Intel Core i5-8265U
- RAM: 8 ГБ
- SSD: 256 ГБ
- ОС: Windows 10 x64
- Браузер: Google Chrome 124.x
- Серверне середовище: локальний Flask-сервер

Д.5.2 Програмні засоби

- Python 3.11
- Flask 2.3
- PostgreSQL 15
- SQLAlchemy
- Postman для API-тестів
- JMeter (для перевірки навантаження)

Д.5.3 Порядок випробувань

Випробування проводиться в два етапи:

- Аналіз відповідності програмної документації.
- Функціональне тестування ключових сценаріїв роботи застосунку.

Д.6 Методи випробувань

Д.6.1 Перевірка документації

Оцінюється повнота документації. Зокрема перевіряється наявність технічного завдання, опису програми, інструкції користувача та структури коду.

Д.6.2 Функціональне тестування

Проводиться ручне та автоматизоване тестування за сценаріями, наведеними нижче.

Таблиця Д.1 – Тестування за сценарієм

№	Дія	Очікуваний результат
1	2	3
1	Реєстрація нового клієнта	Клієнта додано до бази даних, повідомлення про успіх
2	Вхід користувача з неправильним паролем	Повідомлення про помилку авторизації

Кінець таблиці Д.1

1	2	3
3	Додавання нового вантажу	Запис з'являється в базі, повідомлення про успішне додавання
4	Генерація рахунку-фактури	Створено PDF-файл із рахунком
5	Формування звіту за період	Виведення таблиці з вантажами, які відповідають умовам
6	Видалення неіснуючого клієнта	Повідомлення про помилку або ніяких змін
7	Додавання відгуку без тексту	Повідомлення про помилку заповнення форми

Функціональне тестування програмного забезпечення проводиться відповідно до наведених у таблиці Д.1 тестових сценаріїв. Очікувані результати використовуються як еталон для оцінювання поведінки системи під час її випробувань.