

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система підтримки процесів обслуговування
транспортного засобу»

Виконав:

студент IV курсу, групи ІС-13

Ткаченко Ігор Андрійович _____

Керівник:

доцент, к.т.н., доцент

Жураковська Оксана Сергіївна _____

Рецензент:

Доцент кафедри ОТ, к.т.н., доцент

Марковський Олександр Петрович _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Ткаченку Ігорю Андрійовичу

1. Тема проєкту «Інформаційна система підтримки процесів обслуговування транспортного засобу», керівник проєкту доцент, к.т.н., доцент Жураковська Оксана Сергіївна, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «9» червня 2025 р.
3. Вихідні дані до проєкту: клієнт-серверна архітектура, мова програмування JavaScript, серверний фреймворк Express.js, база даних PostgreSQL, фронтенд-бібліотека React, REST API, алгоритм оптимізації NSGA-II, контейнеризація через Docker.
4. Зміст пояснювальної записки: опис предметної області, аналіз існуючих рішень, формування вимог до системи, вибір технології розроблення, розробка інформаційної системи, математичне забезпечення.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): структурна схема системи, діаграма варіантів використання, схема бази даних, діаграма розгортання.
6. Дата видачі завдання «7» березня 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Опис предметної області	14.04.2025	
2	Аналіз існуючих рішень	20.04.2025	
3	Формування вимог до системи	25.04.2025	
4	Розробка математичного забезпечення	01.05.2025	
5	Вибір технологій розробки	09.05.2025	
6	Розробка інформаційної системи	13.05.2025	
7	Подання ДП на попередній захист	25.05.2025	
8	Подання ДП н захист	09.06.2025	

Студент

Ігор ТКАЧЕНКО

Керівник

Оксана ЖУРАКОВСЬКА

АНОТАЦІЯ

Інформаційна система підтримки процесів обслуговування транспортного засобу.

Проект містить 71 с. тексту, 19 рисунків, 8 таблиці, посилання на 16 літературних джерел, додатки та 4 конструкторських документи.

ОБЛІК ВИТРАТ ТА ДОХОДІВ, ПЛАНУВАННЯ ІНТЕРВАЛІВ ОБСЛУГОВУВАННЯ, ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ, ТРАНСПОРТНИЙ ЗАСІБ, ІНФОРМАЦІЙНА СИСТЕМА, WEB-ЗАСТОСУНОК.

Об'єктом розробки є підтримка процесів технічного обслуговування транспортних засобів.

Мета розробки – підвищення ефективності управління процесами технічного обслуговування та експлуатації транспортних засобів шляхом автоматизації обліку ремонтних робіт, планових оглядів, витрат і доходів.

У дипломному проекті реалізовано повнофункціональну веб-орієнтовану інформаційну систему, яка забезпечує централізоване зберігання та обробку даних щодо виконаних технічних обслуговувань, фінансових операцій і планування подальших робіт. Система має планування трирівневої архітектури: клієнтська частина забезпечує інтуїтивно зрозумілий інтерфейс користувача, серверна реалізує бізнес-логіку та обробку запитів, а дані зберігаються в надійній реляційній базі даних. Окремий модуль інформаційної системи призначений для планування інтервалів проведення технічного обслуговування, реалізований за допомогою генетичного алгоритму, що дозволяє знижувати витрати на обслуговування без суттєвого збільшення ризику виникнення несправностей.

Отримані результати можуть бути корисними для підприємств, що управляють автопарками, логістичних компаній, державних структур, а також приватних користувачів, сприяючи підвищенню ефективності керування процесами технічного обслуговування транспортних засобів.

SUMMARY

Information system for supporting vehicle maintenance processes.

The project contains 71 pages. text, 19 figures, 8 tables, links to 16 literary sources, annexes and 4 design documents.

Keywords: expense and revenue accounting, information system, maintenance, planning of maintenance intervals, vehicle, web application.

The object of the development is the support of vehicle maintenance processes.

The purpose of the development – improve the efficiency of managing vehicle maintenance and operation processes through automation of repair record-keeping, scheduled inspections, expense and revenue tracking.

In this thesis, a full-featured web-oriented information system has been implemented, providing centralized storage and processing of data on completed maintenance operations, financial transactions, and planning of future tasks. The system employs a three-tier architecture: the client tier offers an intuitive user interface, the server tier implements business logic and request handling, and data is stored in a reliable relational database. A dedicated module creates a plan for maintenance intervals by means of a genetic algorithm, enabling reduction of maintenance costs without a significant increase in the risk of failures.

The results obtained may be valuable for fleet-managing enterprises, logistics companies, government agencies, as well as private users, by enhancing the efficiency of vehicle maintenance management processes.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4						
5	A4	IC13.200БАК.005 ПЗ	Інформаційна система підтримки процесів обслуговування	71		
6			транспортного засобу.			
7			Пояснювальна записка			
8						
9	A3	IC13.200БАК.005 Д1	Інформаційна система підтримки процесів обслуговування	1		
10			транспортного засобу. Структурна			
11			схема системи			
12						
13	A3	IC13.200БАК.005 Д2	Інформаційна система підтримки процесів обслуговування	1		
14			транспортного засобу. Діаграма			
15			варіантів використання			
16						
17	A3	IC13.200БАК.005 Д3	Інформаційна система підтримки процесів обслуговування	1		
18			транспортного засобу.			
19			Схема бази даних			
20						
21	A3	IC13.200БАК.005 Д3	Інформаційна система підтримки процесів обслуговування	1		
22			транспортного засобу. Діаграма			
23			розгортання			
24						
25						
26						
27						
28						

IC13.200БАК.005 ТП

Зм.	Лист	№ докум.	Підпис			
Розробив	Ткаченко			Літ.	Арк.	Аркушів
Перевірив	Жураковська			Т	1	1
				КПІ ім. Ігоря Сікорського Група IC-13		
Затв.						

Інформаційна система підтримки процесів обслуговування транспортного засобу.
Відомість дипломного проєкту

Пояснювальна записка
до дипломного проєкту
на тему: «Інформаційна система підтримки процесів
обслуговування транспортного засобу»

Київ – 2025 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Опис процесу діяльності	7
1.2 Постановка задачі.....	8
1.2.1 Призначення системи	8
1.2.2 Цілі та задачі розробки	9
Висновок до розділу 1	10
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	11
2.1 Програмне рішення «Fuelio».....	11
2.2 Програмне рішення «Car Maintenance Reminder»	13
2.3 Програмне рішення «Simply Auto»	15
2.4 Програмне рішення «Loggy»	17
Висновок до розділу 2	19
3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ	20
3.1 Вимоги до системи в цілому	20
3.1.1 Вимоги до структури та функціонування системи.....	20
3.1.2 Вимоги до надійності.....	21
3.1.3 Вимоги до збереження інформації	22
3.2 Вимоги до функціональних характеристик.....	23
3.3 Вимоги до видів забезпечення.....	24
Висновок до розділу 3	26
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ	27
4.1 Клієнтська частина.....	27
4.2 Серверна частина	29
4.3 База даних	30
4.4 Середовище розробки та інструменти	30

				ІС13.200БАК.005 ПЗ			
Зм.	Лист	№ док.ум.	Підпис				
Розробив		Ткаченко І.А.		Інформаційна система підтримки процесів обслуговування транспортного засобу. Пояснювальна записка	Літ.	Арк.	Аркушів
Перевірив		Жураковська			Т	2	71
					КПІ ім. Ігоря Сікорського Група ІС-13		
Затв.							

Висновок до розділу 4	31
5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	33
5.1 Структура системи	33
5.2 Функціональна модель системи	35
5.3 Модель даних	38
5.4 Передавання та обробка даних	42
5.5 Архітектура програмного забезпечення	44
5.6 Настанова користувача.....	46
Висновок до розділу 5	56
6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	57
6.1 Змістовна постановка задачі	57
6.2 Математична постановка задачі	58
6.3 Обґрунтування методу розв'язання	61
6.4 Опис методу розв'язання.....	63
6.4.1 Принцип роботи алгоритму NSGA-II	63
6.4.2 Приклад застосування алгоритму.....	65
Висновок до розділу 6	68
ВИСНОВКИ.....	70
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СУБД – система управління базами даних.

ACID – Atomicity, Consistency, Isolation, Durability.

API – Application Programming Interface.

CSV – Comma-Separated Values.

DOM – Document Object Model.

FMEA – Failure Mode and Effects Analysis.

HTTP – Hyper Text Transfer Protocol.

IDE – Integrated Development Environment.

JSON – JavaScript Object Notation.

JWT – JSON Web Token.

MUI – Material User Interface.

NSGA – Non-dominated Sorting Genetic Algorithm.

ORM – Object-Relational Mapping.

PDF – Portable Document Format.

POS – Particle Swarm Optimization.

REST – Representational State Transfer.

SPA – Single Page Application.

SQL – Structured Query Language.

URL – Uniform Resource Locator.

UUID – Universally unique identifier.

VIN – Vehicle Identification Number.

					IC13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

ВСТУП

У сучасних умовах інтенсивного розвитку транспортної інфраструктури та підвищених вимог до ефективності логістичних процесів надзвичайно важливою стає автоматизація внутрішніх операцій, пов'язаних з експлуатацією транспортних засобів. Одним з ключових аспектів цього управління є підтримання належного технічного стану транспортних засобів. Значна частина витрат підприємств, що володіють автопарками, припадає саме на технічне обслуговування та ремонт транспортних одиниць. Неefективне управління цими процесами може призводити до неочікуваних простоїв, зниження рівня безпеки, перевитрат ресурсів і, як наслідок, до зменшення прибутковості компанії.

Традиційний підхід до ведення обліку технічного обслуговування, зазвичай пов'язаний із використанням паперової документації або розрізнених електронних таблиць, не забезпечує достатньої прозорості, оперативності й точності інформації. Це ускладнює прийняття управлінських рішень, унеможливорює ефективне планування графіків обслуговування, а також ускладнює аналіз витрат і оцінку рентабельності використання окремих транспортних засобів. Усе це знижує загальний рівень керованості автопарком.

Водночас, стрімкий розвиток інформаційних технологій відкриває широкі можливості для вдосконалення процесів обліку, планування та контролю. Створення комплексних інформаційних систем для управління технічним обслуговуванням транспортних засобів дає змогу об'єднати в єдиному середовищі усі дані щодо виконаних робіт, витратних матеріалів, доходів, запланованих обслуговувань тощо. Такі системи здатні автоматизувати рутинні задачі, мінімізувати вплив людського фактору та покращити якість управління процесами обслуговування на всіх етапах життєвого циклу транспортного засобу.

У перспективі такі інформаційні системи можуть знайти широке застосування не лише у комерційних транспортних підприємствах, але й у державних структурах, сервісних центрах, логістичних компаніях, муніципальних органах та приватних сервісах, що спеціалізуються на наданні послуг технічного

					ІС13.200БАК.005 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

обслуговування. Також системи цього класу можуть бути адаптовані під специфіку окремих галузей – наприклад, вантажних перевезень, пасажирського транспорту або спеціалізованих служб (комунальних, аварійно-рятувальних тощо). Завдяки універсальності та модульності архітектури такі рішення можуть бути масштабованими, інтегрованими з іншими системами обліку, а також розширеними відповідно до потреб користувача.

Метою розробки системи є підвищення ефективності управління процесами технічного обслуговування транспортних засобів шляхом автоматизації обліку ремонтних робіт, планових оглядів, витрат та доходів.

Актуальність теми дипломного проєкту зумовлена не лише технічною необхідністю оптимізації процесів обліку та обслуговування транспортних засобів, але й загальною тенденцією цифровізації управлінських і технологічних процесів у різних галузях. Упровадження інформаційних систем у сфері експлуатації транспорту сприяє підвищенню прозорості операцій, підвищенню безпеки руху, а також сприяє дотриманню технічних регламентів і стандартів.

Результатом розробки має стати готовий до використання інструмент, який дозволяє зменшити кількість помилок в обліку, покращити контроль над витратами та підвищити прозорість усіх операцій, що стосуються технічного обслуговування транспортних засобів.

Дипломний проєкт складається з наступних розділів: вступ, основні розділи, висновки, список використаних джерел із 16 найменувань, 1 додатку. Графічна частина включає 4 кресленики формату А3. Загальний обсяг 71 сторінка.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис процесу діяльності

Процес діяльності у сфері обслуговування та експлуатації транспортних засобів передбачає регулярну фіксацію інформації про всі ремонтні роботи, планові технічні огляди та заміни деталей, а також супутні витрати, пов'язані з їх проведенням. Окрім фіксації минулих подій, процес також передбачає можливість планування майбутніх робіт, що дозволяє завчасно передбачити необхідність технічного обслуговування та мінімізувати ризик виникнення позапланових несправностей.

У приватному використанні транспортного засобу власник самостійно веде систематичний облік виконаних ремонтних робіт, технічних обслуговувань та витрат, що пов'язані з експлуатацією автомобіля. Такий підхід дозволяє сформулювати чітке розуміння поточного технічного стану транспортного засобу, своєчасно планувати подальші технічні обслуговування, запобігати виникненню непередбачених несправностей, а також сприяє раціональному розподілу фінансових ресурсів на утримання автомобіля.

Для підприємців, які мають у своєму розпорядженні декілька транспортних засобів, підтримка процесів обслуговування технічних засобів характеризується додатковою складністю. Це обумовлено необхідністю одночасного контролю технічного стану всіх транспортних засобів та обліку витрат на їх утримання. Окрім стандартного ведення записів про ремонтні роботи, заміни деталей і супутні витрати, власники автопарків також здійснюють облік доходів, які приносять транспортні засоби. За допомогою цих даних підприємці можуть проводити аналіз прибутковості кожної одиниці транспорту та ефективно планувати подальші витрати, пов'язані з їх обслуговуванням та експлуатацією.

У військових підрозділах підтримка процесів обслуговування транспортних засобів набуває особливого значення, зважаючи на специфіку використання техніки у бойових умовах та її критичну роль у виконанні оперативних завдань. У таких умовах існує нагальна потреба у постійному, системному контролі

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

технічного стану кожного транспортного засобу, а також у веденні обліку всіх виконаних та запланованих ремонтних робіт, оскільки від цього залежить боєздатність та оперативна готовність підрозділу. Такий підхід дозволяє оперативно реагувати на будь-які технічні несправності, запобігати аварійним ситуаціям та гарантувати високий рівень готовності техніки до виконання поставлених бойових завдань.

Станом на теперішній час підтримка процесів обслуговування транспортних засобів у більшості випадків здійснюються вручну – за допомогою паперових журналів або окремих електронних таблиць, які не інтегровані в єдину систему. Такий підхід є малоефективним у випадках, коли потрібно відслідковувати великий обсяг інформації, вести облік для кількох транспортних засобів або здійснювати регулярне планування технічних робіт. Відсутність структурованого середовища для внесення та перегляду даних про ремонтні роботи, технічні обслуговування, витрати на паливо та інші супутні витрати спричиняє ризики пропуску регламентних робіт, знижує загальну інформованість користувача про стан об'єкта експлуатації та унеможливорює проведення системного аналізу витрат на утримання транспортного засобу.

1.2 Постановка задачі

1.2.1 Призначення системи

Система призначена для підтримки процесів технічного обслуговування та експлуатації транспортних засобів. Вона забезпечує централізоване зберігання даних про виконані ремонтні роботи, планові технічні огляди, заміни деталей, доходи, а також про супутні витрати (на паливо, запчастини, сервісні послуги тощо).

Найбільш важливим завданням є фіксація виконаних робіт та планування майбутніх обслуговувань на основі регламентних термінів, або пробігу. Це дозволяє своєчасно проводити необхідні роботи, запобігаючи виникненню критичних поломок і знижуючи простої техніки спричинені непередбачуваними

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

ремонтними роботами.

Підтримка процесу технічного обслуговування та експлуатації транспортних засобів передбачає:

- моніторинг загальної інформації кожного транспортного засобу;
- ведення обліку про всі види обслуговування;
- аналіз та контроль витрат на обслуговування транспортного засобу
- аналіз та контроль доходів від транспортних засобів;
- планування проведення систематичних робіт.

1.2.2 Цілі та задачі розробки

Метою розробки системи є підвищення ефективності управління процесами технічного обслуговування транспортних засобів шляхом автоматизації обліку ремонтних робіт, планових оглядів, витрат та доходів. Реалізація даної системи дозволить підвищити контроль стану техніки, забезпечити вчасне обслуговування і зменшити витрати на експлуатацію.

Для досягнення поставленої мети необхідно реалізувати такі задачі:

- провести аналіз предметної області, визначити основні сценарії системи та сформулювати функціональні та нефункціональні вимоги до її реалізації;
- провести аналіз подібних систем з метою обґрунтування актуальності та доцільності створення власної інформаційної системи;
- розробити структуру бази даних, яка забезпечить збереження та доступ до інформації про транспортні засоби, виконані та заплановані ремонтні роботи та витрати;
- реалізувати модуль реєстрації та авторизації користувачів;
- реалізувати ведення журналу для внесення записів про ремонтні роботи, планові технічні обслуговування та заміни деталей;
- реалізувати можливість ведення обліку витрат та доходів, які виникають у процесі експлуатації транспортного засобу, з можливістю категоризації за типами;
- розробити механізм планування майбутніх робіт, що дозволить

					ІС13.200БАК.005 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

встановлювати орієнтовні терміни або пробіг, при досягненні яких потрібно виконати технічне обслуговування;

– реалізувати можливість перегляду історії обслуговування, яка забезпечить доступ до повного переліку раніше виконаних робіт та витрат для кожного транспортного засобу;

– реалізувати модуль планування інтервалів обслуговування;

– провести тестування системи, спрямоване на перевірку її працездатності, відповідності функціональним вимогам та стабільності роботи у межах типових сценаріїв використання.

Висновок до розділу 1

У даному розділі було проведено аналіз предметної області, що охоплює процеси технічного обслуговування та експлуатації транспортних засобів. Проведений аналіз показав, що у більшості випадків ведення обліку ремонтних робіт, витрат та доходів здебільшого здійснюється вручну, що зумовлює низку проблем, зокрема ускладнення доступу до необхідної інформації, ймовірність пропуску планового технічного обслуговування та нераціонального використання фінансових ресурсів.

Розглянуто особливості підтримки процесів обслуговування транспортних засобів для приватних, комерційних та військових цілей. Визначено, що в усіх зазначених сферах існує потреба у впорядкуванні процесів фіксації ремонтів, контролю витрат та планування технічного обслуговування.

На основі проведеного аналізу було визначено призначення системи, сформульовано її мету та перелік задач. Розробка такої інформаційної системи дозволить впорядкувати облік і забезпечити кращий контроль за витратами на утримання транспортних засобів.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

У цьому розділі розглянуто існуючі аналогічні програмні продукти для підтримки процесів обслуговування транспортного засобу. Розгляд аналогів допоможе оцінити поточний стан ринку, визначити сильні сторони існуючих рішень, а також звернути увагу на ті функціональні обмеження, які знижують зручність використання. Це дозволяє чітко сформулювати перелік вимог до майбутньої системи та уникнути повторення типових недоліків.

Подальший аналіз зосереджено на чотирьох поширених системах для обліку обслуговування транспортних засобів: Fuelio, Car Maintenance Reminder, Simply Auto та Drivvo. У наступних частинах розділу буде подано детальний опис кожної з них, з акцентом на їхні функціональні можливості, переваги та недоліки.

2.1 Програмне рішення «Fuelio»

Fuelio [1] – популярний мобільний застосунок, орієнтований насамперед для ведення обліку витрат на паливо та визначення середнього споживання пального транспортним засобом. Програма підтримує роботу з пристроями Android та iOS, має зручний у використанні інтерфейс і надає користувачу можливість вносити дані вручну або за допомогою автоматичного зчитування чеків.

Головною функцією застосунку є ведення статистики пов'язаною з використанням палива – користувач може відстежувати середню витрату пального, вартість кілометра тощо. Крім того, Fuelio дозволяє вести облік для кількох транспортних засобів одночасно, що зручно для користувачів, які у користуванні мають декілька транспортних засобів. Також застосунок надає можливість зберігати дані за допомогою Google Drive або Dropbox, що забезпечує їх безпеку у випадку втрати пристрою чи перевстановлення програми. Наявність часткової української локалізації робить використання застосунку більш зручнішим та зрозумілішим для користувачів з України. На головному екрані мобільного застосунку Fuelio, як показано на рисунку 2.1, користувач може переглядати

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

поточний пробіг транспортного засобу, середню та останню витрату пального, вартість останньої заправки, а також швидко отримувати доступ до розділів для обліку пального й інших витрат.

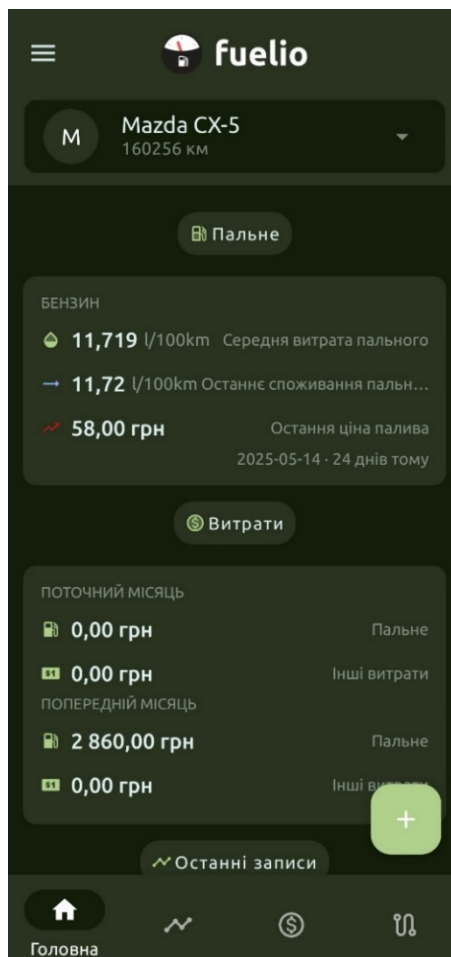


Рисунок 2.1 – Головний екран застосунку Fuelio

Далі розглянемо основні переваги та недоліки системи.

Переваги:

- застосунок є повністю безкоштовним, усі базові функції доступні без оформлення підписки, що робить його доступним для широкого кола користувачів;
- зручна система обліку витрат на паливо, з можливістю точного підрахунку витрат на кілометр і середньої витрати пального кожного транспортного засобу;
- інформація про витрати представлена у вигляді графіків, які дають змогу легко відстежувати зміни в динаміці витрат та детально їх аналізувати;
- підтримка декількох транспортних засобів, що дозволяє вести облік

					IC13.200БАК.005 ПЗ	Арк.
						12
Зм.	Лист	№ докум.	Підпис	Дата		

одночасно для різних транспортних засобів;

- хмарна синхронізація із Google Drive та Dropbox забезпечує збереження та перенесення даних при зміні пристрою;

- часткова українська локалізація покращує зручність використання для українців;

- імпорт та експорт даних у форматі CSV надає можливість резервного копіювання для аналізу в сторонніх системах.

Недоліки:

- додаток зосереджений виключно на обліку витрат на паливо й не охоплює інші важливі аспекти експлуатації авто, такі як обслуговування чи ремонти;

- відсутній журнал технічного обслуговування, де можна було б фіксувати історію пройдених технічних обслуговувань або виконаних ремонтів;

- планування обслуговування реалізовано мінімально: немає можливості налаштувати нагадування за пробігом чи датою та детально описувати заплановані дії;

- застосунок доступний лише у мобільній версії, що обмежує зручність використання для частини користувачів.

2.2 Програмне рішення «Car Maintenance Reminder»

Car Maintenance Reminder [2] – мобільний застосунок, розроблений виключно для операційної системи iOS, функціонал якого зосереджений на нагадуваннях про планові технічні обслуговування. Його основна мета – допомогти користувачам не пропускати важливі сервісні роботи, пов’язані з пробігом або часом експлуатації. Додаток має простий і зрозумілий інтерфейс, дозволяє створювати планові технічні обслуговування та встановлювати інтервали між нагадуваннями.

На головному екрані Car Maintenance Reminder, як зображено на рисунку 2.2, користувач бачить перелік усіх доданих транспортних засобів разом із зазначенням пробігу кожного авто та відповідними іконками для різних типів технічного обслуговування. Кольорова індикація дозволяє оперативно оцінити стан

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

обслуговування для кожного з автомобілів, а також швидко перейти до додавання нового транспортного засобу чи перегляду детальної інформації.



Рисунок 2.2 – Головний екран застосунку Car Maintenance Reminder

Переваги:

- простий інтерфейс і функціонал, який легко освоїти будь-яким користувачам;
- система нагадувань за пробігом або часом дозволяє уникати пропуску обслуговувань;
- є можливість додавати зображення й чеки до обслуговувань, що допомагає фіксувати підтвердження виконаних робіт;
- підтримується імпорт та експорт даних, що дозволяє зберігати резервні копії або переносити інформацію між пристроями;
- користувач може персоналізувати типи нагадувань відповідно до власних потреб.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

Недоліки:

- застосунок не передбачає обліку витрат чи доходів, що обмежує його функціональність у порівнянні з повноцінними системами;
- відсутній журнал обслуговувань із можливістю перегляду історії робіт або аналітики;
- функція планування обмежується лише базовими нагадуваннями, тому немає можливості задавати складні або багатоетапні роботи;
- відсутня українська локалізація;
- працює лише на платформі iOS, що зменшує потенційну аудиторію;
- відсутня підтримка обліку кількох транспортних засобів – що обмежує можливості для власників автопарку або кількох авто.

2.3 Програмне рішення «Simply Auto»

Simply Auto [3] – застосунок, розроблений для ведення повного обліку витрат, пов'язаних з експлуатацією транспортних засобів та для запису технічних обслуговувань. Він охоплює такі аспекти, як заправки, технічне обслуговування, ремонти, а також пропонує зручну систему нагадувань. Окрім стандартного введення даних, користувачі можуть переглядати звіти у вигляді графіків, що допомагає аналізувати витрати та технічні роботи. Додаток доступний як на Android і iOS, так і у вебверсії, що робить його зручним у щоденному користуванні. Завдяки гнучкій архітектурі, додаток підходить як для особистого використання, так і для обліку автопарку в невеликих підприємствах.

На рисунку 2.3 представлено головний екран мобільного застосунку Simply Auto, який містить зведену інформацію про обраний транспортний засіб: витрати на паливо за останні 30 днів, середню витрату пального, кількість проведених сервісних робіт, а також список останніх записів про заправки та обслуговування. Такий підхід до відображення даних дозволяє користувачу швидко оцінити поточний стан витрат і проведених обслуговувань, а також оперативно додавати нові записи.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

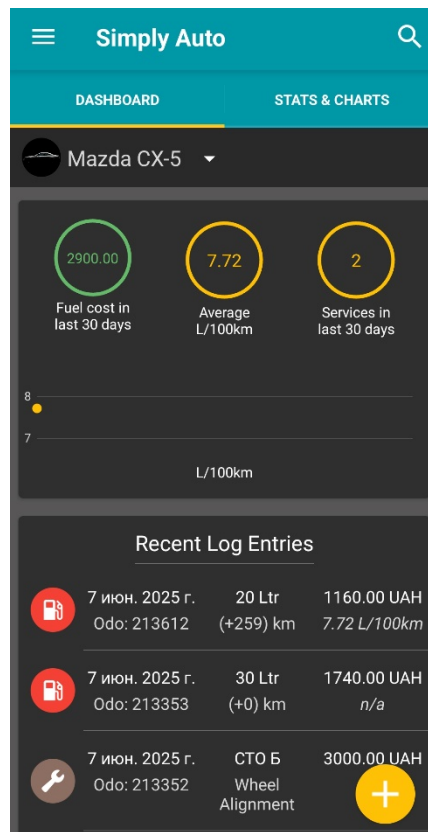


Рисунок 2.3 – Головний екран застосунку Simply Auto

Переваги:

- повний набір функцій для обліку витрат, ремонтів, технічного обслуговування, а також створення нагадувань і перегляду аналітики;
- можливість вести облік одразу для кількох транспортних засобів та водіїв, що є зручним як для сімей, так і для малого бізнесу;
- підтримка вебверсії, що забезпечує зручний доступ до даних з будь-якого пристрою, а також синхронізація між усіма платформами;
- експорт та імпорт даних, зокрема у CSV, і побудова графіків на основі введених даних дають змогу оцінювати витрати в динаміці;
- застосунок дозволяє зберігати дані в хмарі, що зручно при зміні пристрою або для створення резервної копії.

Недоліки:

- частина основних функцій, включаючи синхронізацію, аналітику, а також додавання більше трьох транспортних засобів, доступна лише у платній версії, що обмежує можливості користувачів без підписки;

- доступ до вебзастосунку доступний лише з платною підпискою, що зменшує зручність використання та кількість потенційних користувачів;
- немає можливості вписувати доходи, що обмежує фінансову аналітику при комерційному використанні;
- відсутня українська локалізація, що знижує зручність використання для українців;
- додаток не забезпечує повноцінного аналізу витрат – немає можливості групувати чи порівнювати дані за категоріями;
- інтерфейс не є інтуїтивно зрозумілим і може здаватися надто перенавантаженим для нових користувачів.

2.4 Програмне рішення «Loggy»

Loggy [4] – мобільний застосунок для Android та iOS, орієнтований на зручне ведення історії обслуговування транспортних засобів. Застосунок позиціонується як універсальний інструмент, який підтримує практично всі типи техніки: легкові автомобілі, мотоцикли, вантажівки, спецтехніку, човни та навіть літаки. Програма дозволяє створювати окремі профілі для кожного транспортного засобу та вести повноцінний журнал робіт з можливістю прикріплення фото, відео, чеків та голосових нотаток. Дані синхронізуються в хмарі, що дозволяє безпечно зберігати інформацію та переносити її між пристроями. Також передбачена функція створення нагадувань за пробігом або часом, а також експорт історії обслуговування у форматі PDF, що особливо зручно при продажі авто.

На рисунку 2.4 наведено приклад перегляду історії обслуговування у застосунку Loggy. Користувач може створювати окремий профіль для кожного транспортного засобу, прикріплювати реальні фотографії авто, а всі виконані роботи відображаються у вигляді детального журналу. Для кожного запису можна швидко додати додаткову інформацію чи мультимедійні файли, що спрощує фіксацію всіх деталей обслуговування.

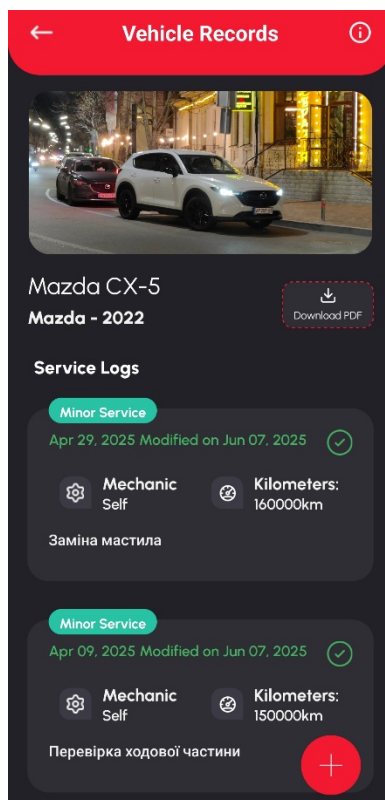


Рисунок 2.4 – Сторінка перегляду історії обслуговування у Loggy

Переваги:

- підтримка необмеженої кількості транспортних засобів різних типів, що робить застосунок універсальним для широкого кола користувачів;
- повністю безкоштовний функціонал без реклами, що забезпечує повний доступ до всього функціоналу без додаткових витрат;
- можливість прикріплення до записів мультимедійних файлів (фото, відео, голосові нотатки), що підвищує зручність та достовірність обліку;
- синхронізація в хмарі та PDF-експорт спрощують збереження історії та надають можливість її передачі іншим користувачам;

Недоліки:

- відсутність вебверсії унеможливорює роботу на портативних комп'ютерах, що зменшує кількість потенційних користувачів;
- інтерфейс доступний лише англійською мовою, що може створювати труднощі для українців;
- відсутній окремий модуль аналітики витрат – аналіз доступний лише через

					IC13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

ручний перегляд окремих записів;

– відсутня функція обліку доходів чи прибутковості, що робить систему менш придатною для користування у підприємницьких цілях;

– відсутня можливість фільтрації та сортування записів обслуговування за категоріями.

Висновок до розділу 2

Проведений аналіз популярних застосунків для підтримки процесів обслуговування транспортного засобу показав, що жоден із них не є повністю універсальним. Кожна система має як свої сильні сторони, так і слабкі, які ускладнюють використання для певних груп користувачів.

Наприклад мобільний застосунок Fuelio чудово підходить для ведення статистики пов'язаної з паливом, але за допомогою нього не можна фіксувати технічні роботи і планувати обслуговування. Car Maintenance Reminder має повний і зручний функціонал нагадувань про майбутні обслуговування, проте не дозволяє вести облік витрат і не підтримує збереження повної історії обслуговування. Simply Auto, хоч і пропонує розширений функціонал, але більшість можливостей відкриває лише після придбання підписки. Крім того, відсутність української мови в інтерфейсі може створити незручності для частини користувачів. Застосунок Loggy приваблює універсальністю та простотою, однак не надає користувачеві аналітики по витратах, не дозволяє вести облік доходів і не підтримує фільтрацію обслуговувань та витрат за категоріями.

Отримані результати аналізу підтверджують необхідність розробки системи для підтримки процесів обслуговування транспортного засобу.

3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

3.1 Вимоги до системи в цілому

3.1.1 Вимоги до структури та функціонування системи

Інформаційна система підтримки процесів обслуговування транспортного засобу створюється як веборієнтована система з чітко визначеною модульною архітектурою. Призначення даної системи – забезпечити зручне та надійне середовище для ведення обліку технічного обслуговування транспортних засобів, планування майбутніх робіт, контролю витрат та доходів і аналізу історії експлуатації. Система має бути гнучкою, стабільною і достатньо простою у використанні, щоб не вимагати спеціальних знань від користувача.

До складу системи мають входити такі логічні модулі:

– модуль керування транспортним засобом: має забезпечувати можливість створення, редагування та перегляд інформації про транспортний засіб;

– модуль обліку ремонтів і технічного обслуговування – повинен давати змогу фіксувати кожне технічне обслуговування, вказуючи його зміст, дату та пов'язані витрати;

– модуль планування робіт: має відповідати за формування графіку обслуговувань на основі пробігу або часу, а також генерувати нагадування для користувача;

– модуль обліку витрат: повинен надавати змогу вести облік витрат відповідно до обраних категорій (паливо, сервіс, запчастини тощо) та формувати аналітичні звіти;

– модуль обліку доходів: має дозволяти фіксувати усі джерела прибутків від експлуатації транспортного засобу (перевезення, оренда, партнерські доходи), а також будувати статистику та звіти щодо прибутковості;

– модуль планування інтервалів обслуговування: має генерувати інтервал проходження технічного обслуговування

– модуль авторизації користувача: реалізує автентифікацію, авторизацію та базову безпеку даних.

					ІС13.200БАК.005 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

Функціонування системи передбачається в безперервному режимі. Вона повинна бути доступною з будь-якого пристрою через вебінтерфейс, працювати у браузері без потреби в додатковому встановленні. Завдяки своїй простій архітектурі, система не потребує потужних обчислювальних ресурсів і може бути розгорнута на більшості хостингових платформ.

Система має бути розширюваною та адаптивною: її архітектура передбачає можливість додавання нових модулів, а також інтеграції з іншими програмними рішеннями через API. Передбачено підтримку обміну даними у форматах, зручних для подальшої обробки та взаємодії з іншими сервісами. Така гнучкість дозволяє ефективно реагувати на зміни потреб користувачів або умов експлуатації.

3.1.2 Вимоги до надійності

Надійність є ключовим фактором успішного використання інформаційної системи, оскільки вона напряму впливає на збереження, точність і безпеку обробки даних. Система повинна стабільно працювати у штатних і нештатних умовах, забезпечуючи користувачів достовірною інформацією незалежно від зовнішніх чинників.

Основні вимоги до надійності включають:

- цілісність даних: система повинна забезпечувати збереження даних у незмінному вигляді, виключаючи спотворення через помилки введення, технічні збої чи стороннє втручання;
- захист від збоїв: при втраті зв'язку, вимкненні живлення або технічному збої інформація не повинна бути втрачена. Застосовуються транзакційні механізми СУБД та черги повторних запитів;
- безпека доступу: автентифікація, авторизація та контроль сесій;
- оновлення та підтримка: система має підтримувати регулярні оновлення без порушення роботи користувачів.

У сукупності ці вимоги забезпечують стабільну та передбачувану роботу системи в умовах реального використання.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

3.1.3 Вимоги до збереження інформації

Збереження інформації є одним із критичних аспектів у роботі інформаційної системи для обліку технічного обслуговування, витрат і доходів транспортних засобів. Усі дані – від загальної інформації про транспортні засоби до детальних записів про обслуговування, витрати, доходи та планування – повинні надійно зберігатися у централізованій реляційній базі даних. Система повинна гарантувати, що всі взаємопов'язані сутності зберігаються в базі даних у взаємозалежному вигляді, з використанням зовнішніх ключів та відповідних обмежень. Структуроване зберігання дозволяє легко отримувати потрібну інформацію, здійснювати пошук, фільтрацію та сортування без втрати продуктивності.

Для забезпечення доступності та зручності обробки даних у системі має використовуватись відкритий та поширений формат, наприклад JSON. Він забезпечує можливість обміну даними, формування звітів, а також полегшує інтеграцію з іншими програмними засобами.

З точки зору безпеки, система має реалізовувати базові принципи контролю доступу: кожен користувач повинен мати змогу працювати лише зі своїми даними. Додаткові механізми, як-от шифрування всіх записів не є критично необхідними, однак можуть бути впроваджені в майбутньому, якщо система масштабуватиметься.

Система повинна зберігати ефективність навіть зі зростанням обсягів даних. Для цього ще на початковому етапі розробки необхідно передбачити реалізацію базових принципів масштабованості, зокрема індексацію найбільш використовуваних полів у базі даних. Це дозволить підтримувати стабільну швидкодію системи без необхідності вдаватися до складних механізмів архівації чи оптимізацій, які є надмірними для нашої системи.

Таким чином, вимоги до збереження інформації у цій системі зводяться до забезпечення надійного централізованого зберігання даних із захистом доступу на рівні користувача, а також підтримки структурованих форматів, придатних для формування звітів і подальшого аналізу.

					ІС13.200БАК.005 ПЗ	Арк.
						22
Зм.	Лист	№ докум.	Підпис	Дата		

3.2 Вимоги до функціональних характеристик

Інформаційна система повинна автоматизувати всі ключові процеси, пов'язані з веденням обліку технічного стану транспортних засобів, витрат і доходів. До функцій, які підлягають автоматизації, належать:

- додавання транспортних засобів;
- редагування, видалення та перегляд інформації про транспортні засоби;
- фіксація технічних обслуговувань і ремонтів;
- планування майбутніх робіт за пробігом або часом;
- облік витрат за категоріями;
- облік доходів за категоріями;
- перегляд звітів по витратах та доходах;
- перегляд історії обслуговувань;
- автентифікація користувача.

Усі перелічені функції мають реалізовуватись як окремі, логічно відокремлені, але пов'язані між собою частини системи. Це дозволяє зберегти гнучкість архітектури, забезпечує простоту підтримки та можливість подальшого розвитку.

Якість реалізації кожної функції повинна відповідати загальним критеріям: система має працювати без помилок, коректно обробляти всі дії користувача та забезпечувати очікуваний результат у відповідності до технічного завдання. Інтерфейс повинен бути інтуїтивно зрозумілим та простим у користуванні навіть для людей без спеціальної технічної підготовки. Функції мають бути адаптованими до подальшого розширення без порушення основної логіки системи. Важливо забезпечити узгодженість роботи модулів системи, зокрема автоматичне оновлення загальної суми витрат при додаванні нових записів технічного обслуговування.

Інформація, яку формує система, повинна відображатися у доступному для користувача форматі, що дозволяє швидко її проаналізувати та використати для подальших рішень. Дані мають виводитися у табличному вигляді, а в разі потреби

					ІС13.200БАК.005 ПЗ	Арк.
						23
Зм.	Лист	№ докум.	Підпис	Дата		

– доповнюватися графічними елементами (діаграми, графіки). Уся інформація повинна бути чітко структурованою з можливістю сортування, фільтрації та групування. Інтерфейс та вихідна інформація повинні бути повністю українською мовою для зручності використання. У всіх розрахунках, пов'язаних із фінансами, система повинна дотримуватися точності до двох знаків після коми. Оновлення вихідної інформації повинно здійснюватися одразу після внесення змін, щоб забезпечити її максимальну актуальність і відповідність поточному стану даних.

Час відгуку на основні дії користувача не повинен перевищувати 2 секунд, що забезпечить комфортну взаємодію із системою в режимі реального часу. У випадках асинхронної обробки, таких як побудова великих звітів або аналітичних таблиць, допустимі короткочасні затримки, які не повинні впливати на загальну швидкодію системи чи створювати відчуття зависання.

Загалом функціональні характеристики мають повністю відповідати визначеним задачам, бути достатньо гнучкими для подальшого розвитку, забезпечувати комфортну роботу для користувача та дозволяти ефективно застосування як у повсякденному житті, так і у спеціалізованих галузях, зокрема в умовах підвищеної відповідальності, які характерні для військових підрозділів.

3.3 Вимоги до видів забезпечення

У процесі розробки системи підтримки процесів обслуговування транспортних засобів важливо забезпечити належну реалізацію чотирьох складових: математичного, інформаційного, програмного та технічного забезпечення.

Математичне забезпечення системи повинно ґрунтуватися на використанні базових, але достатньо ефективних алгоритмів, необхідних для виконання основних функцій. До таких належать сортування, фільтрація, підрахунок кількості обслуговувань, обчислення сум витрат за вказаний період, а також формування загальної статистики на основі внесених даних. Всі обчислення мають бути точними, із дотриманням стандарту двох знаків після коми у фінансових значеннях,

					ІС13.200БАК.005 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

що гарантує коректність статистики та аналітичних висновків. Таким чином забезпечується точність збережених даних та мінімізується ймовірність помилок під час перегляду або аналізу інформації користувачем.

Інформаційне забезпечення визначає склад і структуру даних, необхідних для функціонування системи. У межах розроблюваного застосунку інформація має зберігатися у вигляді таблиць, дані яких повинні бути структуровані таким чином, щоб уникати дублювання, забезпечувати цілісність і логічну зв'язаність. Для цього використовуються первинні та зовнішні ключі, а також нормалізоване представлення записів.

Організація інформації повинна передбачати можливість її ефективної обробки – пошуку, фільтрації, сортування. Структура має бути оптимізованою для регулярного оновлення та розширення. Також важливо передбачити можливість збереження даних у зручних для експорту й обробки форматах, таких як CSV або JSON. Система управління базами даних повинна підтримувати цілісність, доступність та гарантувати безпеку інформації відповідно до потреб користувача.

Програмне забезпечення базується на сучасному відкритому стеку технологій. Серверна частина реалізована за допомогою Node.js, клієнтська – з використанням бібліотеки React. Для зберігання даних застосовується PostgreSQL. Усі обрані інструменти є ліцензійними, активно підтримуються, масштабовані та добре документовані. Код системи має бути читабельним, структурованим і відповідати базовим стандартам програмування. Це дозволяє забезпечити як стабільну роботу системи, так і її подальший розвиток.

Технічне забезпечення системи передбачає використання стандартних, доступних засобів без необхідності у високопродуктивному обладнанні. Система повинна коректно функціонувати на персональних комп'ютерах, ноутбуках чи мобільних пристроях, які мають сучасний браузер та стабільне підключення до мережі. Клієнтська частина запускається без встановлення, безпосередньо через вебінтерфейс. Для розміщення серверної частини достатньо середнього за характеристиками віртуального серверу. Основна вимога – це забезпечення безперервної, стабільної роботи системи без перевантажень, навіть за умов

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

обмежених ресурсів або експлуатації в умовах обмежених технічних можливостей

Висновок до розділу 3

У даному розділі було детально сформовано вимоги до майбутньої інформаційної системи підтримки процесів обслуговування транспортного засобу. На основі аналізу предметної області, цілей розробки та функціонального призначення системи визначено ключові характеристики, яким вона повинна відповідати.

Описано загальну структуру та функціонування системи як модульної веборієнтованої платформи, здатної забезпечити стабільну, безперервну та зручну в користуванні роботу як для приватного, так і для комерційного чи військового застосування. Сформовано перелік логічних модулів, які охоплюють усі ключові аспекти обліку: від реєстрації транспортного засобу та фіксації обслуговувань – до аналізу витрат і доходів.

У розділі також наведено вимоги до функціональних характеристик системи, включно з точністю обробки фінансових даних, форматами представлення інформації, максимально допустимим часом відгуку та зручністю взаємодії користувача з інтерфейсом. Особливу увагу приділено забезпеченню актуальності даних, їх оновленню в реальному часі, а також адаптації до умов із підвищеною відповідальністю, таких як бойові дії.

Окремо проаналізовано вимоги до чотирьох ключових видів забезпечення – математичного, інформаційного, програмного та технічного. Запропоновані підходи до реалізації алгоритмів, структури збереження даних, програмного коду та технічного середовища орієнтовані на досягнення максимальної надійності, масштабованості та простоти впровадження системи.

Таким чином, вимоги, сформовані у цьому розділі, закладають основу для подальшої розробки інформаційної системи, що відповідає сучасним потребам обліку технічного обслуговування та забезпечує її ефективне застосування в широкому спектрі умов.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБЛЕННЯ

Для реалізації інформаційної системи обліку технічного обслуговування транспортних засобів було обрано сучасний технологічний стек, що забезпечує високу продуктивність, легкість підтримки, масштабованість та надійність розробки. Вибір інструментів та технологій базувався на аналізі сучасних тенденцій у веброботці, популярності серед розробників, активності спільноти, а також відповідності до поставлених вимог до системи. Застосування перевірених і добре документованих технологій дозволяє значно скоротити час розробки та забезпечити стабільність роботи системи.

4.1 Клієнтська частина

Клієнтська частина системи реалізована на основі сучасного фреймворку React, який є однією з найпопулярніших JavaScript бібліотек для створення інтерфейсів користувача. React побудований на компонентному підході, що дозволяє розбивати інтерфейс на окремі незалежні блоки, кожен з яких має власну логіку та стан [5]. Це значно полегшує розробку, тести, повторне використання коду та масштабування застосунку. React використовує віртуальний DOM, що забезпечує високу швидкість інтерфейсу, оптимізуючи повторний рендеринг лише тих компонентів, які зазнали змін.

Для організації маршрутизації застосовується React Router DOM – офіційна бібліотека для роутінгу в React. Вона дозволяє створювати багатосторінкові застосунки (SPA) з динамічними маршрутами, передачею параметрів у URL, вкладеними маршрутами та гнучкою системою навігації. Завдяки React Router DOM можна реалізувати зручну навігацію між сторінками без перезавантаження сторінки [6].

З метою забезпечення статичної типізації та зниження ймовірності помилок під час розробки використовується TypeScript. Це надбудова над JavaScript, яка дозволяє задавати типи для змінних, функцій та компонентів. TypeScript виявляє

					IC13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		27

помилки ще на етапі компіляції, забезпечує кращу читабельність і передбачуваність коду, полегшує рефакторинг та дає можливість скористатися перевагами автодоповнення в редакторі коду [7].

Як інструмент збірки обрано Vite – нове покоління білд-системи, що забезпечує надзвичайно швидкий запуск проєкту та гаряче оновлення модулів. Vite використовує сучасні можливості браузера, що дозволяє пришвидшити процес розробки порівняно з традиційними рішеннями на кшталт Webpack [8]. Також Vite підтримує гнучке налаштування плагінів і має хорошу інтеграцію з React.

Для побудови інтерфейсу використовується Material UI (MUI) – повнофункціональна бібліотека компонентів для React, яка реалізує принципи дизайну Google Material. MUI надає понад сотню готових компонентів (таблиці, форми, кнопки, співбесіди, модальні вікна, меню тощо), що дозволяє швидко створити естетично привабливий, уніфікований та зручний у користуванні інтерфейс. Також MUI має вбудовану підтримку адаптивності та темізації, що дозволяє реалізовувати темні та світлі режими без додаткових зусиль [9].

Для стилізації компонентів з MUI використовується бібліотека Emotion. Це CSS-in-JS рішення, яке дозволяє писати стилі безпосередньо в JavaScript-коді з урахуванням логіки компонента. Emotion забезпечує динамічну генерацію класів, підтримку тем, медіа-запитів, стилів на основі пропсів та інших факторів. Це дозволяє створювати індивідуальні стилі компонентів без потреби в окремих CSS-файлах.

Паралельно з MUI застосовується Tailwind CSS – утилітарний фреймворк для швидкої верстки, що базується на використанні набору готових класів. Tailwind дозволяє створювати адаптивну структуру сторінок, сітки, відступи, тіні, кольори тощо напряму в JSX-коді. Такий підхід прискорює процес розробки та дає змогу швидко прототипувати дизайн без переходу до окремого стилізування.

Для роботи з іконками використовуються дві бібліотеки: Lucide React та React Icons. Lucide React надає сучасні, акуратні SVG-іконки з можливістю кастомізації кольору, розміру та стилю, що добре підходить до MUI-стилістики. React Icons дозволяє імпортувати іконки з різних наборів та використовувати їх як

					IC13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

звичайні React-компоненти, що значно спрощує додавання графічних елементів до інтерфейсу.

4.2 Серверна частина

На сервері використовується Node.js – це середовище виконання JavaScript, яке ідеально підходить для створення високопродуктивних вебсистем з великою кількістю одночасних підключень. У контексті розробки системи обліку технічного обслуговування транспортних засобів Node.js дозволяє обробляти велику кількість запитів у режимі реального часу, зберігаючи швидкодію навіть при високому навантаженні. Його неблокуюча модель вводу/виводу особливо ефективна для обробки асинхронних операцій, таких як запис до бази даних, автентифікація користувачів чи надсилання відповідей клієнту [10]. Використання однієї мови на клієнті й сервері спрощує підтримку та дозволяє зменшити кількість помилок при передачі даних. Завдяки своїй подієво-орієнтованій архітектурі Node.js ефективно працює у середовищах з великою кількістю паралельних запитів, що є особливо корисним у системах, де очікується інтенсивний обмін інформацією між користувачем і сервером у реальному часі.

Як фреймворк для побудови REST API застосовується Express.js – мінімалістичний та гнучкий інструмент, який є одним із найпоширеніших рішень для створення серверних застосунків на платформі Node.js [11]. Серед ключових переваг Express.js – це його простота у використанні, що дозволяє швидко розгортати API будь-якої складності. Завдяки своїй модульній архітектурі Express легко адаптується до потреб проєкту: розробник може підключати лише ті функції, які дійсно потрібні. Express також підтримує проміжне програмне забезпечення, яке дозволяє легко додавати обробку помилок, логування, авторизацію або інші функціональні блоки без зміни основного коду. Крім того, його широке поширення у промислових проєктах гарантує наявність великої кількості готових рішень, плагінів та активної спільноти, що полегшує розробку та підтримку.

Для зручного та ефективного керування базою даних використовується ORM

					ІС13.200БАК.005 ПЗ	Арк.
						29
Зм.	Лист	№ докум.	Підпис	Дата		

Sequelize. Цей інструмент дозволяє створювати і управляти моделями даних, автоматично генерувати SQL-запити, а також забезпечує підтримку міграцій та транзакцій [12]. Використання Sequelize спрощує роботу з базою даних і дозволяє уникнути багатьох типових помилок у запитах.

4.3 База даних

У якості системи управління базами даних використовується PostgreSQL – потужне реляційне рішення з відкритим кодом, яке зарекомендувало себе як одне з найбільш надійних і функціональних серед усіх доступних СУБД. PostgreSQL підтримує ACID-транзакції, що забезпечує цілісність даних навіть у випадках збоїв. Завдяки розвиненій системі індексів та можливості створення складних запитів ця СУБД ідеально підходить для обробки великого обсягу інформації без втрати продуктивності [13].

Для потреб системи обліку транспортного обслуговування PostgreSQL є доцільним вибором, адже вона дозволяє ефективно організувати зв'язки між сутностями, забезпечити швидкий пошук та фільтрацію записів, а також гарантує стабільність у довготривалій експлуатації. Її гнучкість, підтримка з боку спільноти, наявність засобів розширення функціоналу та відсутність ліцензійних обмежень роблять PostgreSQL найкращим вибором для реалізації нашої системи.

4.4 Середовище розробки та інструменти

Для розробки було обрано середовище WebStorm – одне з найпотужніших комерційних IDE для JavaScript і TypeScript. Це середовище надає широкі можливості для розробки як фронтенд, так і бекенд частини проекту в єдиному просторі. Серед головних переваг WebStorm – підтримка розумної автопідказки коду, глибокий аналіз структури проекту, вбудовані інструменти для роботи з Git, а також підтримка популярних фреймворків і бібліотек, зокрема React, Node.js, Express та ін. Завдяки цьому середовище значно прискорює розробку, зменшує

					IC13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		30

кількість помилок та дозволяє легко орієнтуватися у великих проєктах. Вибір WebStorm був зумовлений потребою в стабільному та надійному середовищі для швидкої розробки складної інтерактивної системи з чітко розділеними модулями та великою кількістю логіки як на фронтенді, так і на сервері.

Для забезпечення послідовного стилю коду було обрано ESLint –інструмент статичного аналізу коду, що дозволяє не лише автоматично виявляти синтаксичні або логічні помилки на ранньому етапі, а й підтримувати єдиний стиль написання в усьому проєкті. ESLint є надзвичайно гнучким у налаштуванні, з його допомогою можна створювати власні правила або використовувати вже готові конфігурації з відкритого середовища. Він інтегрується безпосередньо з середовищем розробки, що дозволяє бачити помилки безпосередньо під час написання коду. Завдяки цьому інструменту підтримується чистота та послідовність коду, що дозволяє уникати хаотичного стилю програмування й підвищує загальну якість програмного забезпечення.

З метою ізоляції середовища, спрощення розгортання та уніфікації інфраструктури використовується Docker. Це рішення дозволяє розміщувати всі компоненти системи у незалежних контейнерах, кожен з яких містить повний набір необхідних залежностей. Таким чином, середовище розробки, тестування та продакшен не мають відмінностей, що запобігає помилкам, пов'язаним із різницею конфігурацій.

Docker значно спрощує процес встановлення – замість ручного налаштування середовища достатньо запустити заздалегідь підготовлений контейнер. Це економить час, дозволяє швидко повернутися до стабільної версії у випадку помилок і спрощує перенесення проєкту на інший сервер або комп'ютер. Завдяки цьому інструменту забезпечується зручність у розгортанні системи, стабільність її роботи та передбачуваність поведінки застосунку в будь-якому середовищі.

Висновок до розділу 4

У цьому розділі було здійснено обґрунтований вибір технологій та

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

інструментів для реалізації інформаційної системи обліку технічного обслуговування транспортних засобів. Основна увага приділялась забезпеченню надійності, масштабованості, зручності використання та відповідності функціональним вимогам, сформованим у попередньому розділі.

Для середовища розробки обрано JavaScript як універсальну мову програмування, що дозволяє реалізовувати як клієнтську, так і серверну частину системи. Такий підхід спрощує підтримку коду та забезпечує єдину логіку роботи всієї системи. Серверна частина реалізується з використанням Node.js, що забезпечує високу продуктивність і масштабованість. Клієнтська частина створюється за допомогою бібліотеки React, яка забезпечує зручну побудову інтерфейсу користувача та підтримує гнучку компонентну архітектуру.

Для зберігання даних використовується реляційна система керування базами даних PostgreSQL, яка забезпечує надійність, цілісність та ефективну обробку структурованої інформації. Вибір цієї СУБД також обумовлений її сумісністю з обраним технологічним стеком і широкою підтримкою інструментів для розробки та адміністрування.

Загалом, обраний набір технологій є сучасним, перевіреним у практичному використанні та придатним для реалізації завдань, поставлених у межах даної системи.

5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

5.1 Структура системи

Інформаційна система підтримки процесів обслуговування транспортного засобу має чітко визначену трирівневу архітектуру, яка включає клієнтську частину, серверну частину та реляційну базу даних. Така структура дає змогу логічно розподілити відповідальність між компонентами, забезпечити масштабованість і стабільну роботу системи, а також підвищити її гнучкість у супроводі та розширенні функціоналу.

Клієнтська частина реалізована у вигляді вебзастосунку, побудованого з використанням React та супутніх бібліотек. Вона забезпечує зручний і адаптивний інтерфейс для користувача, відповідає за збір вхідних даних, відображення результатів, формування запитів і взаємодію з API. Всі функціональні блоки взаємодіють через модуль роботи з API, який виконує роль комунікаційного інтерфейсу з серверною частиною. Ключові компоненти клієнтської частини згруповано у модулі, кожен з яких реалізує окрему частину логіки взаємодії з користувачем.

Модулі клієнтської частини:

- модуль перегляду транспортних засобів: відповідає за візуалізацію наявних транспортних засобів та доступ до їх детальної інформації;
- модуль перегляду витрат та доходів: відображає фінансові записи, згруповані за категоріями та транспортними засобами;
- модуль перегляду звітів: забезпечує користувачу доступ до аналітики у вигляді діаграм і графіків;
- модуль перегляду обслуговувань: дозволяє переглядати всі виконані технічні обслуговування;
- модуль перегляду нагадувань: показує нагадування щодо майбутніх обслуговувань за пробігом або датою;
- модуль перегляду строків обслуговування – демонструє результати математичної оптимізації інтервалів обслуговувань.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

– FrontEnd фреймворк: керує відображенням та навігацією між сторінками клієнтської частини;

модуль роботи з API: відповідає за надсилання запитів та обробку відповідей із серверної частини.

Серверна частина системи реалізована на основі ExpressJS з використанням мови TypeScript. Вона відповідає за обробку HTTP-запитів, реалізацію бізнес-логіки, валідацію даних, автентифікацію, авторизацію та взаємодію з базою даних через ORM. Серверна архітектура є модульною: кожен функціональний блок реалізовано окремим модулем, що підвищує розділення відповідальностей та полегшує підтримку коду. Комунікація між клієнтською і серверною частинами забезпечується через REST API, а передача структурованих даних здійснюється у форматі JSON.

Модулі серверної частини:

– модуль керування транспортними засобами: опрацьовує запити на створення, зміну або видалення даних про транспортний засіб;

– модуль обліку витрат та доходів: відповідає за логіку створення та обробки фінансових транзакцій;

– модуль звітності та аналітики: реалізує формування агрегованих даних для клієнтської частини;

– модуль планування інтервалів обслуговування: реалізує математичну модель розрахунку строків технічних обслуговувань;

– модуль обліку обслуговувань: здійснює збереження, зміну та повернення інформації про виконані технічні роботи;

– модуль планування нагадувань: керує створенням і оновленням нагадувань для конкретного транспортного засобу;

– модуль авторизації користувача: обробляє реєстрацію, вхід, автентифікацію та керування сесіями;

– API шлюз: розподіляє запити до відповідних модулів системи відповідно до маршрутів.

База даних побудована на основі реляційної СУБД PostgreSQL. У структурі

					ІС13.200БАК.005 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

бази передбачено таблиці для користувачів, транспортних засобів, нагадувань, витрат, доходів, категорій транзакцій та технічного обслуговування. Всі таблиці пов'язані між собою згідно з логікою доменної області системи. Наприклад, один користувач може мати декілька транспортних засобів, кожен із яких має власну історію обслуговувань, витрат і нагадувань. Таке моделювання дозволяє ефективно працювати з великими обсягами інформації та формувати узгоджену звітність.

Таким чином, інформаційна система побудована на основі модульної архітектури з чітко визначеною роллю кожного з її компонентів. Такий підхід забезпечує гнучкість, простоту масштабування та зручність супроводу, а також дозволяє легко інтегрувати нові функціональні можливості без порушення стабільності основного коду. Візуальне подання структурної схеми системи, представлене на кресленику ІС13.200БАК.005 Д1, дозволяє цілісно оцінити логіку побудови системи та взаємозв'язки між її частинами.

5.2 Функціональна модель системи

Функціональна модель розроблюваної інформаційної системи підтримки процесів обслуговування транспортних засобів відображає логіку взаємодії користувача із програмним продуктом, демонструючи, які дії доступні для виконання в системі. Вона покликана формалізувати процеси, що відбуваються у системі, а також забезпечити чітке розуміння її функціональної структури та призначення окремих складових.

Центральним актором у системі виступає користувач, який може бути як неавторизованим, так і вже зареєстрованим. Усі дії в системі можливі лише після проходження процедури автентифікації, що гарантує цілісність, захист і персоніфікацію даних. Користувач, який вперше користується системою, має можливість зареєструватися, після чого створюється обліковий запис, який необхідний для подальшої взаємодії з усіма основними функціональними модулями. Після успішної автентифікації користувач отримує доступ до інтерфейсу, в межах якого реалізується вся основна функціональність: додавання,

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		35

редагування, перегляд та аналіз даних, формування звітів, планування обслуговувань та інші процеси, пов'язані з технічною і фінансовою підтримкою транспортних засобів.

Користувач має доступ до широкого спектру функціональних можливостей, кожна з яких реалізує взаємодію з транспортними засобами, витратами, технічним обслуговуванням, доходами та аналітичними інструментами. Всі ці функції згруповані у відповідні модулі, що полегшує роботу користувача з системою та робить процес роботи з нею зручним і простим.

Зокрема, у межах модуля керування транспортними засобами користувач може виконувати базові операції з об'єктами обліку: переглядати список створених транспортних засобів, додавати нові з відповідними характеристиками, вносити зміни до вже існуючих або видаляти застарілі транспортні засоби. Крім того, реалізовано можливість експорту всієї бази транспортних засобів для подальшого аналізу в сторонніх застосунках. Це дає змогу зберігати та підтримувати актуальну інформацію про автопарк у електронному вигляді, спрощуючи процес її обробки та аналізу.

У системі передбачено модуль керування технічними обслуговуваннями та ремонтами, який дозволяє користувачеві вести облік усіх виконаних робіт для кожного транспортного засобу. Така інформація зберігається у вигляді структурованих записів, які можна використовувати для аналізу історії обслуговування кожного окремого транспортного засобу. Передбачено також можливість редагування й видалення відповідних записів, що дозволяє оперативно оновлювати інформацію. За потреби, користувач може експортувати дані про обслуговування у вигляді звітних таблиць.

Модуль керування витратами дозволяє користувачу зберігати дані про всі фінансові транзакції, пов'язані з утриманням транспортних засобів. До таких витрат можуть належати заправки, купівля запчастин, технічне обслуговування тощо. Кожну витрату можна детально описати, обрати категорію, вказати дату, суму та транспортний засіб, до якого вона належить. Реалізовано можливість фільтрації за різними параметрами, а також експорт даних про витрати для

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		36

подальшого аналізу.

Аналогічно до витрат, система підтримує облік доходів – фінансових надходжень, які пов'язані з експлуатацією транспортного засобу у комерційній діяльності. Користувач може створювати, переглядати, змінювати або видаляти відповідні записи, а також фільтрувати доходи за періодом, транспортним засобом або типом доходу. Присутня функція експорту таблиці доходів для подальшого аналізу чи формування звітної документації.

Важливим компонентом системи є модуль для створення та управління нагадуваннями про технічні обслуговування. Завдяки цьому модулю можна заздалегідь створювати нагадування про необхідні технічні обслуговування транспортних засобів. Користувач самостійно обирає умови нагадування (за часом чи пробігом), а також зміст повідомлення. У разі потреби, нагадування можна змінити або повністю видалити. Коли подія, на яку було створено нагадування, виконується, система автоматично переносить її до переліку проведених обслуговувань. Крім того, система надсилає відповідні повідомлення на електронну пошту користувача, що дозволяє своєчасно відреагувати на заплановані роботи та не допустити пропуску важливих обслуговувань.

Окрім стандартних функцій з ведення обліку, система також містить модуль формування звітів і аналітики, який дозволяє користувачу оцінювати ефективність використання ресурсів та стан транспортних засобів на основі накопичених даних. Його основне призначення – надати користувачу можливість швидко та зручно проаналізувати зібрані дані про витрати, доходи та технічні обслуговування. Звіти формуються у вигляді таблиць та графіків, що дає змогу візуально оцінити ефективність експлуатації транспорту, виявити найбільш затратні періоди, оцінити динаміку прибутковості та приймати виважені рішення щодо подальшого використання ресурсів. Механізми аналітики тісно пов'язані з попередніми модулями, оскільки опираються на вже зібрані дані, і є невід'ємною частиною функціонального циклу системи.

Також слід відзначити модуль розрахунку оптимального плану технічного обслуговування. Для цього користувач зазначає ключові параметри: тип

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		37

обслуговування, орієнтовну вартість, інтервал пробігу або періодичність у часі. Після цього система здійснює обчислення з використанням алгоритму Парето-розкладу, який дозволяє знайти найефективніше співвідношення між частотою обслуговувань, їхньою вартістю та впливом на технічний стан транспортного засобу. Результати подаються у вигляді структурованих таблиць та графіків, що дозволяє користувачу краще планувати обслуговування техніки в довгостроковій перспективі.

Функціональні сценарії взаємодії користувача з системою та логічна структура основних дій відображені на кресленику ІС13.200БАК.005 Д2, який містить діаграму варіантів використання. Такий спосіб організації функціоналу дозволяє користувачам легко орієнтуватися в системі, швидко виконувати необхідні операції та аналізувати інформацію для ефективного керування транспортними засобами.

5.3 Модель даних

Ефективна робота системи значною мірою залежить від правильної організації її структури даних. Для цього було побудовано цілісну реляційну модель даних, яка охоплює всі аспекти функціонування системи та відповідає її логічній і функціональній архітектурі. Основною метою побудови моделі даних є забезпечення цілісності інформації, спрощення доступу до неї, підтримка масштабованості системи в майбутньому та збереження зв'язків між сутностями. Усі сутності системи реалізовано у вигляді таблиць з чітко визначеними полями та встановленими зв'язками між ними за допомогою зовнішніх ключів. Задля підтримки м'якого видалення та відстеження часу створення кожного запису передбачено службові поля `created_at` і `deleted_at`.

Нижче наведено повну структуру даних, які використовуються в базі даних. Кожна з таблиць містить чітко визначені поля, типи даних та призначення. Розроблена структура даних відображає логіку взаємодії користувача з системою, а також формалізує зберігання та обробку даних у відповідності до передбаченого

					ІС13.200БАК.005 ПЗ	Арк.
						38
Зм.	Лист	№ докум.	Підпис	Дата		

функціоналу.

Таблиця користувачів містить дані про зареєстрованих осіб, які мають доступ до системи. Зберігається базова інформація, така як ім'я, електронна пошта та хеш пароля. Також ведеться облік дати створення запису та можливості м'якого видалення користувача при потребі. Структуру таблиці наведено в таблиці 5.1.

Таблиця 5.1 – Структура таблиці Users

Назва поля	Тип даних	Опис
id	UUID	Унікальний ідентифікатор користувача
name	TEXT	Ім'я користувача
email	TEXT	Адреса електронної пошти
password_hash	TEXT	Хешований пароль
created_at	TIMESTAMP	Дата створення запису
deleted_at	TIMESTAMP	Дата м'якого видалення

Таблиця транспортних засобів містить повну інформацію про кожен одиницю техніки, яка належить користувачеві. Зокрема, зберігається тип транспортного засобу, його марка, модель, рік випуску, пробіг, об'єм двигуна та додаткові параметри, як-от номерний знак, VIN-код та тип палива. Кожен запис у таблиці транспортних засобів прив'язаний до конкретного користувача за допомогою зовнішнього ключа. Це дозволяє системі фільтрувати, обробляти та виводити дані по транспортних засобах, що належать певному користувачу, забезпечуючи індивідуалізацію роботи з інформацією. Детальну структуру таблиці наведено в таблиці 5.2.

Таблиця 5.2 – Структура таблиці vehicles

Назва поля	Тип даних	Опис
id	UUID	Унікальний ідентифікатор транспортного засобу
user_id	UUID	Зовнішній ключ на таблицю користувачів
type	TEXT	Тип транспортного засобу
brand	TEXT	Марка транспортного засобу
model	TEXT	Модель транспортного засобу

Назва поля	Тип даних	Опис
year	INTEGER	Рік випуску транспортного засобу
mileage	INTEGER	Пробіг
engine_volume	TEXT	Об'єм двигуна
license_plate	TEXT	Номерний знак
vin	TEXT	VIN-код
fuel_type	TEXT	Тип палива
created_at	TIMESTAMP	Дата створення запису
deleted_at	TIMESTAMP	Дата м'якого видалення

У таблиці обслуговувань зберігається інформація про всі проведені роботи для обраного транспортного засобу. Зокрема, зазначається тип обслуговування, дата, вартість, пробіг під час його виконання та сервісний центр. При необхідності, кожен запис може бути пов'язаний із транзакцією, що відображає витрати на проведення обслуговування. Структуру цієї таблиці представлено в таблиці 5.3.

Таблиця 5.3 – Структура таблиці maintenances

Назва поля	Тип даних	Опис
id	UUID	Унікальний ідентифікатор обслуговування
vehicle_id	UUID	Зовнішній ключ на таблицю транспортних засобів
transaction_id	UUID	Зв'язок з відповідною транзакцією
date	DATE	Дата проведення обслуговування
cost	DECIMAL	Вартість проведеного обслуговування
mileage	INTEGER	Пробіг на момент проведення обслуговування
service_center	TEXT	Назва сервісного центру
notes	TEXT	Примітки, або додаткова інформація
created_at	TIMESTAMP	Дата створення запису
deleted_at	TIMESTAMP	Дата м'якого видалення запису

Таблиця нагадувань дозволяє фіксувати майбутні роботи, які мають бути виконані на певному транспортному засобі. Користувач має змогу зазначити тип обслуговування, заплановану дату або пробіг, а також додати коментар. Після

виконання нагадування створюється відповідний запис в таблиці обслуговувань. Повна структура таблиці наведена у таблиці 5.4.

Таблиця 5.4 – Структура таблиці reminders

Назва поля	Тип даних	Опис
id	UUID	Унікальний ідентифікатор нагадування
vehicle_id	UUID	Зовнішній ключ на таблицю транспортних засобів
reminder_type	TEXT	Тип обслуговування, про яке нагадує система
date	DATE	Запланована дата виконання
mileage	INTEGER	Запланований пробіг
notes	TEXT	Примітки, або додаткова інформація
created_at	TIMESTAMP	Дата створення запису
deleted_at	TIMESTAMP	Дата м'якого видалення запису

Фінансові транзакції, що вносяться до системи, зберігаються в таблиці transactions. Вона охоплює як витрати, так і доходи, дозволяючи зафіксувати дату, суму, категорію та опис кожної операції. Для аналітики та статистики транзакції пов'язуються з конкретними транспортними засобами і категоріями. Структуру таблиці наведено у таблиці 5.5.

Таблиця 5.5 – Структура таблиці transactions

Назва поля	Тип даних	Опис
id	UUID	Унікальний ідентифікатор транзакції
vehicle_id	UUID	Зовнішній ключ на таблицю транспортних засобів
category_id	UUID	Зовнішній ключ на таблицю категорій
date	DATE	Дата проведення фінансової операції
amount	DECIMAL	Сума транзакції
description	TEXT	Примітки, або додаткова інформація
created_at	TIMESTAMP	Дата створення запису
deleted_at	TIMESTAMP	Дата м'якого видалення запису

Таблиця категорій використовується для класифікації фінансових записів.

Кожна категорія визначає, до якого типу вона належить — дохід чи витрата. Це дає змогу більш точно впорядковувати інформацію та полегшує створення звітності. Опис структури цієї таблиці подано у таблиці 5.6.

Таблиця 5.6 – Структура таблиці categories

Назва поля	Тип даних	Опис
id	UUID	Унікальний ідентифікатор категорії
name	TEXT	Назва категорії
type	TEXT	Тип категорії

Взаємозв'язки між усіма таблицями бази даних наочно подані на схемі бази даних, що наведена на кресленику IC13.200БАК.005 ДЗ. Ця діаграма відображає логічну структуру бази даних та схему зв'язків між таблицями.

Структура моделі даних, що була побудована, забезпечує повну відповідність функціональним вимогам системи та дозволяє надійно реалізовувати увесь передбачений функціонал. Вона також створює гнучку основу для подальшого масштабування системи або інтеграції нових модулів у майбутньому.

5.4 Передавання та обробка даних

У розробленій інформаційній системі для обліку технічного обслуговування та витрат на транспортні засоби передавання та обробка даних реалізовані на основі клієнт-серверної архітектури з чітким розподілом обов'язків між фронтендом, сервером і базою даних. Це забезпечує ефективну, масштабовану та структуровану взаємодію між усіма компонентами системи.

Передавання даних ініціюється користувачем через клієнтський інтерфейс, реалізований за допомогою React. Після заповнення форм або взаємодії з інтерфейсом (наприклад, створення нового запису витрат або оновлення пробігу) формується HTTP-запит у форматі JSON. У тілі запиту передаються лише ті поля, які необхідні для виконання конкретної операції. Для кожного виду запиту використовується відповідний HTTP-метод (GET, POST, PATCH або DELETE), а

також вказується URL-адреса, яка відповідає маршруту певного ресурсу.

Отриманий запит надсилається на серверну частину системи, реалізовану на базі ExpressJS. Тут кожен маршрут має свій контролер, який отримує вхідні дані та виконує їх первинну перевірку.

Після успішної валідації дані передаються в сервіси, які реалізують основну бізнес-логіку. Наприклад, у випадку створення нового нагадування або запису витрат, сервіс формує відповідні запити до бази даних, використовуючи інструменти для роботи з PostgreSQL. Для цього застосовується ORM Sequelize, яка інкапсулює логіку взаємодії з таблицями.

Результати обробки (наприклад, створений об'єкт або повідомлення про помилку) передаються назад через контролер, де формується відповідь клієнту. Відповідь також структурується у вигляді JSON-об'єкта, що включає необхідні дані та статус виконання.

На клієнтській стороні відповідь обробляється за допомогою бібліотеки TanStack Query. Вона зберігає отримані дані у кеші, оновлює інтерфейс користувача та дозволяє забезпечити взаємодію без зайвих повторних запитів. Таким чином, уся інформація, яку бачить користувач на екрані, є результатом чітко визначеного ланцюга передачі, обробки й відображення даних.

Взаємодія між компонентами системи дотримується принципу розділення відповідальностей: інтерфейс відповідає за збирання і представлення даних, сервер – за логіку і контроль доступу, а база – за надійне зберігання.

У структурі інформаційних потоків можна виокремити такі ключові етапи:

- формування вхідних даних на клієнті;
- надсилання HTTP-запиту;
- валідація та обробка даних на сервері;
- збереження або читання з бази даних;
- формування відповіді та повернення її клієнту;
- відображення результатів користувачу.

Цей процес забезпечує повний цикл обробки даних та гарантує коректність, безпеку й стабільність функціонування системи.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		43

5.5 Архітектура програмного забезпечення

Архітектура програмного забезпечення є фундаментально важливим елементом, що визначає структуру, взаємодію та принципи організації всіх компонентів інформаційної системи для підтримки процесів технічного обслуговування та експлуатації транспортних засобів. Вона окреслює не лише логічну побудову програмного продукту, але й правила, за якими різні частини системи взаємодіють, забезпечуючи при цьому гнучкість, простоту масштабування та легкість подальшого розвитку.

Розроблена інформаційна система базується на сучасній тришаровій клієнт-серверній моделі, що поділяє всю систему на презентаційний рівень, серверний рівень і рівень зберігання даних. Така структурована модель дозволяє чітко визначити відповідальність кожного рівня, забезпечуючи простоту підтримки та ефективність роботи.

Першим рівнем системи є презентаційний рівень (Frontend). Цей рівень відповідає за інтерфейс користувача та забезпечує взаємодію з основним функціоналом системи. Він розроблений на основі JavaScript бібліотеки React, яка дозволяє будувати ефективні, адаптивні та зручні користувацькі інтерфейси. React надає можливість створювати незалежні компоненти, які можна повторно використовувати у різних частинах системи, що значно спрощує розробку та підтримку інтерфейсу. Завдяки бібліотеці TanStack Query забезпечується комфортна робота з даними: запити до сервера асинхронні, а інформація зберігається у кеші для швидкого доступу. Інтерфейс містить зручні форми введення, таблиці з аналітикою, графіки витрат та історії обслуговувань, які сприяють швидкому отриманню та аналізу необхідної інформації.

Другий рівень архітектури – серверний рівень (Backend) – реалізує основну бізнес-логіку системи та взаємодіє з базою даних. Серверна частина побудована на Node.js та Express.js, що дозволяє ефективно обробляти велику кількість одночасних запитів користувачів. Цей рівень забезпечує безпеку системи через JWT-автентифікацію та шифрування критичних даних, таких як паролі

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

користувачів. Використання Sequelize ORM дозволяє легко працювати з даними, забезпечуючи швидкі й безпечні операції з базою PostgreSQL. Крім того, серверний рівень відповідає за автоматизацію деяких процесів, зокрема за перевірку планових технічних обслуговувань і надсилання нагадувань користувачам. Вся серверна логіка розділена на модулі: модуль керування транспортними засобами, модуль обліку ремонтів, модуль планування робіт, модуль витрат та доходів, модуль керування користувачами. Це дозволяє легко модифікувати і масштабувати функціонал без втрати цілісності архітектури.

Третім важливим компонентом системи є рівень зберігання даних (Database). На цьому рівні застосовується реляційна система управління базами даних PostgreSQL, яка забезпечує високу надійність, цілісність та швидкість обробки даних. Структура бази даних розроблена таким чином, щоб чітко визначити таблиці та зв'язки між ними, що гарантує ефективний доступ до інформації. Основні таблиці включають дані про транспортні засоби, проведені обслуговування, витрати та доходи, інформацію про користувачів, а також заплановані роботи. Це дозволяє системі швидко та ефективно надавати всю необхідну інформацію для прийняття управлінських рішень.

Взаємодія між зазначеними рівнями реалізована через стандартизовані REST API, що дозволяє забезпечити чітке розмежування відповідальності між компонентами та забезпечує простоту інтеграції нових функцій. Система здатна обмінюватися даними в реальному часі, що є важливим для користувачів, яким необхідно миттєво реагувати на зміни стану транспортних засобів.

Для забезпечення стабільної та зручної роботи застосовуються контейнери Docker, які гарантують однаковість середовища на всіх етапах життєвого циклу системи: від розробки до розгортання. Завдяки цьому система стабільно функціонує в будь-яких умовах експлуатації, включаючи можливість роботи в умовах обмежених ресурсів.

Таким чином, реалізована архітектура програмного забезпечення забезпечує високу модульність, зручність подальшого розвитку та підтримки, а також надійність і стабільність роботи у різних умовах її експлуатації. Для демонстрації

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		45

взаємодії основних компонентів системи на кресленнику ІС13.200БАК.005 Д4 наведено діаграму розгортання, яка ілюструє архітектуру програмного забезпечення інформаційної системи. Вона відображає зв'язки між клієнтською частиною (React з TanStack Query), серверною частиною (Node.js, Express.js, Sequelize), базою даних (PostgreSQL). Взаємодія між усіма компонентами реалізована у межах чітко визначеної інфраструктури розгортання.

5.6 Настанова користувача

Однією з важливих складових будь-якого сучасного програмного забезпечення є його зручність для кінцевого користувача. Розроблена інформаційна система створена з урахуванням практичних потреб користувача, тому передбачає просту й логічну послідовність дій, які забезпечують повний контроль над обліком транспортних засобів, їхнім обслуговуванням, фінансовими витратами та доходами. Нижче наведено покрокову настанову щодо використання системи, яка охоплює усі основні дії, які може виконувати користувач у межах функціональних можливостей застосунку.

Після запуску системи користувач потрапляє на екран автентифікації, де передбачено два сценарії – авторизація та реєстрація. Користувач, який вже має обліковий запис, вводить адресу електронної пошти та пароль, як зображено на рисунку 5.1 і натискає кнопку входу.

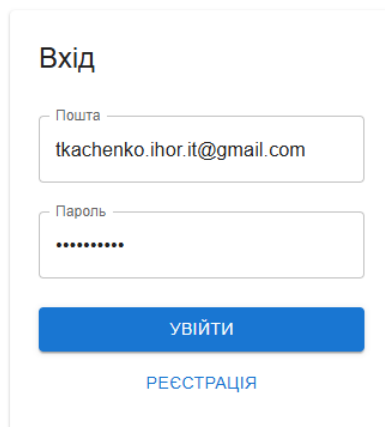


Рисунок 5.1 – Форма авторизації

Новий користувач має можливість пройти реєстрацію натиснувши на відповідну кнопку, де він заповнює форму для створення облікового запису. Цей процес представлено на рисунку 5.2.

Реєстрація

Ім'я
Ігор

Пошта
tkachenko.igor.it@gmail.com

Пароль
.....

Підтвердіть пароль
.....

ЗАРЕЄСТРУВАТИСЯ

УВІЙТИ

Рисунок 5.2 – Форма реєстрації

Після успішної авторизації користувач переходить на головну сторінку системи, яка зображена на рисунку 5.3. Тут відображається навігаційне меню, яке дає змогу переходити між основними вкладками системи: «Транспортні засоби», «Обслуговування», «Витрати», «Доходи», «Нагадування», «Звіти» та «Інтервали обслуговування». На цій сторінці відображається загальна інформація про стан обліку, яка включає кількість проведених обслуговувань, кількість активних транспортних засобів та загальну суму витрат і доходів. Крім того, у відповідних блоках на головному екрані відображаються останні дії користувача, що охоплюють ключові аспекти роботи з транспортними засобами. Зокрема, користувач має змогу переглянути останні фінансові транзакції, до яких належать як витрати, так і доходи, що були зафіксовані по всіх наявних у системі транспортних засобах. Окремо виводиться перелік останніх виконаних ремонтних робіт, що дозволяє користувачу оперативно простежити історію технічного обслуговування. Також формується список найближчих запланованих обслуговувань, відповідно до заданих інтервалів часу або пробігу.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		47

Меню

- Головна
- Транспортні засоби
- Обслуговування
- Витрати
- Доходи
- Нагадування
- Звіти
- Налаштування

ІС для підтримки процесів обслуговування транспортного засобу

1 троп

13

Проведених ремонтів

6

Активних транспортних засобів

- 145 350 €

Загальна сума витрат

+ 253 300 €

Загальна сума доходів

Історія витрат та доходів

Дата	ТЗ	Категорія	Сума
09.06.2025	Volkswagen Transporter	Пасажи́рські перевезення	+ 6 700 €
08.06.2025	Mercedes-Benz Vito	Пасажи́рські перевезення	+ 7 500 €
05.06.2025	MAN TGS	Сервісні роботи	- 18 000 €
05.06.2025	MAN TGS	Вантажні перевезення	+ 18 000 €
30.05.2025	Skoda Octavia	Партнерські доходи	+ 5 200 €

Останні ремонтні роботи

Дата	ТЗ	Тип обслуговування	Вартість
05.06.2025	MAN TGS	Заміна гальмівних дисків	18 000 €
25.05.2025	Skoda Octavia	Заміна ременя ГРМ	7 500 €
15.04.2025	Mercedes-Benz Vito	Перевірка ходової частини	2 200 €
30.03.2025	MAN TGS	Технічне обслуговування	15 000 €
10.03.2025	Toyota Corolla	Заміна фільтрів	1 800 €

Нагадування

ТЗ	Тип обслуговування	Дата	Пробіг
Toyota Corolla	Заміна масла	-	119 900 км Прострочено на 100 км
Volkswagen Passat B7	Перевірка двигуна	10.06.2025 Прострочено на 5 днів	-

Рисунок 5.3 – Головна сторінка системи

У вкладці «Транспортні засоби» користувач може переглядати наявний список техніки, а також додавати нову. Для створення нового транспортного засобу необхідно натиснути кнопку додавання, після чого відкривається модальне вікно зображене на рисунку 5.4, де користувачу необхідно заповнити такі поля: тип транспортного засобу, його марка, модель, рік випуску та пробіг. Крім того, передбачена можливість вказати додаткові дані, які не є обов’язковими – номерний знак, VIN-код і тип палива.

Додати транспортний засіб

Тип ТЗ *
Автомобіль

Марка *
Volkswagen

Модель *
Passat B7

Рік випуску *
2015

Пробіг *
150000 км

Об’єм двигуна, л
2

Номерний знак
AA1111AA

VIN код
WVWXXN7AN1FE123456

Тип палива
Дизель

Рисунок 5.4 – Модальне вікно для додавання нового транспортного засобу

Після заповнення форми та натискання кнопки підтвердження, запис успішно зберігається та додається до загального списку транспортних засобів, який представлений на рисунку 5.5.

Тип ТЗ	Марка	Модель	Рік випуску	Об'єм двигуна	Пробіг	Дії
Автомобіль	Volkswagen	Passat B7	2015	2	150 000 км	
Номерний знак: AA1111AA VIN код: WVWXXN7AN1FE123456 Тип палива: Дизель						
Автомобіль	Toyota	Corolla	2018	1.6	120 000 км	
Автомобіль	Skoda	Octavia	2016	1.8	170 000 км	
Мікроавтобус	Mercedes-Benz	Vito	2017	2.1	220 000 км	
Мікроавтобус	Volkswagen	Transporter	2019	2	200 000 км	
Вантажівка	MAN	TGS	2014	10.5	320 000 км	

Рисунок 5.5 – Сторінка «Транспортні засоби»

Кожен запис можна в подальшому редагувати або видаляти за допомогою відповідних кнопок, доступних безпосередньо у таблиці обслуговувань. На рисунку 5.6 зображено приклад відкритого модального вікна для редагування інформації про обраний транспортний засіб.

Редагувати транспортний засіб

Тип ТЗ *
Автомобіль

Марка *
Toyota

Модель *
Corolla

Рік випуску *
2018

Пробіг *
120000 км

Об'єм двигуна, л
1,6

Номерний знак
BC2222BC

VIN код
JTDBURHE9JJ654321

Тип палива
Бензин

Рисунок 5.6 – Модальне вікно для редагування транспортного засобу

На рисунку 5.7 зображено модальне вікно з підтвердженням видалення транспортного засобу, яке з'являється після натискання відповідної кнопки у таблиці.

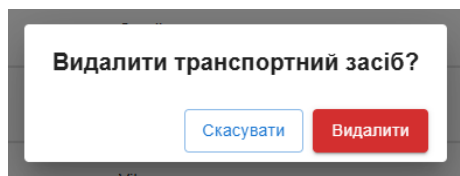


Рисунок 5.7 – Модальне вікно для видалення транспортного засобу

Сторінка «Обслуговування» призначена для детального ведення історії обслуговування кожного авто в системі. На ній користувач має змогу не лише переглядати всі записи, пов'язані з ремонтами та технічним обслуговуванням транспортних засобів, але й керувати ними – додавати нові, редагувати або видаляти наявні. Інтерфейс сторінки представлено на рисунку 5.8.

Дата	ТЗ	Тип обслуговування	Пробіг	Вартість	Сервісний центр	Дії
25.05.2025	Skoda Octavia	Заміна ременя ГРМ	178 500 км	7 500 ₪	СТО Skoda	✎ ✖
15.04.2025	Mercedes-Benz Vito	Перевірка ходової частини	235 500 км	2 200 ₪	СТО BusService	✎ ✖
30.03.2025	MAN TGS	Технічне обслуговування	332 000 км	15 000 ₪	СТО Truck	✎ ✖
10.03.2025	Toyota Corolla	Заміна фільтрів	130 000 км	1 800 ₪	СТО Тойота	✎ ✖
20.02.2025	Volkswagen Passat B7	Заміна гальмівних колодок	160 500 км	3 200 ₪	СТО Пассат	✎ ✖
22.01.2025	Volkswagen Transporter	Ремонт електрики	192 000 км	3 400 ₪	СТО VW	✎ ✖
08.12.2024	Mercedes-Benz Vito	Заміна зчеплення	225 000 км	12 000 ₪	СТО BusService	✎ ✖
18.11.2024	MAN TGS	Заміна шин	325 000 км	42 000 ₪	СТО Truck	✎ ✖
05.11.2024	Toyota Corolla	Перевірка ходової частини	122 000 км	1 500 ₪	СТО Тойота	✎ ✖
12.10.2024	Skoda Octavia	Заміна масла	171 000 км	2 400 ₪	СТО Skoda	✎ ✖
30.09.2024	Volkswagen Transporter	Заміна масла	186 000 км	2 600 ₪	СТО VW	✎ ✖

Рисунок 5.8 – Сторінка «Обслуговування»

Користувач бачить таблицю, в якій відображаються дані про дату виконання, транспортний засіб з яким проводилися роботи, тип виконаних робіт, пробіг, вартість та сервісний центр у якому було проведено обслуговування. У правій

частині кожного рядка доступні іконки для редагування або видалення відповідного запису. Всі операції супроводжуються простими діями через інтерфейс, що робить процес роботи швидким, зрозумілим та зручним.

У верхній частині сторінки розміщено фільтри, що дозволяють переглянути всі сервісні роботи для конкретного транспортного засобу, а також вказати діапазон часу за який потрібно відображати обслуговування. Це забезпечує зручність при роботі зі значним обсягом даних та дозволяє швидко знаходити потрібну інформацію.

Для створення нового запису користувачу необхідно натиснути кнопку «Додати обслуговування», після чого відкривається модальне вікно, де необхідно заповнити всі обов'язкові поля: транспортний засіб, тип обслуговування, дату, пробіг, вартість, а також за бажанням вказати сервісний центр та примітки, як зображено на рисунку 5.9.

Рисунок 5.9 – Модальне вікно для додавання нового обслуговування

Після збереження запис автоматично додається до загального списку обслуговувань. Крім того, з метою зручності та зменшення дублювання дій, система автоматично створює відповідний запис у розділі «Витрати», пов'язаний

Додати витрату

T3 *

Дата *

Сума (€) *

Категорія *

Опис

Рисунок 5.11 – Модальне вікно для додавання нової витрати

Після натискання на кнопку «Додати», нова витрата зберігається у таблиці та відображається в загальному списку фінансових операцій. Кожен створений запис про витрати можна відредагувати або видалити.

Сторінка «Доходи» реалізована за принципом, аналогічним до вкладки з витратами, проте орієнтована на фіксацію усіх фінансових надходжень від транспортних засобів. Інтерфейс вкладки представлено на рисунку 5.12. У верхній частині сторінки доступні фільтри для вибору транспортного засобу, категорії доходу та діапазону часу. Також реалізовано інформаційні блоки, які демонструють загальні суми доходів по кожній категорії, що дає змогу швидко оцінити основні джерела прибутку.

Меню

- Головна
- Транспортні засоби
- Обслуговування
- Витрати
- Доходи
- Нагадування
- Звіти
- Налаштування

Доходи

Всі T3 Всі категорії Дата з Дата по Весь час
+ Додати дохід

Оренда
54 000 €

Пасажирські перевезення
83 800 €

Вантажні перевезення
65 500 €

Спеціалізовані роботи
34 500 €

Партнерські доходи
15 500 €

Інше
0 €

Дата	Сума	Категорія	T3	Опис	Дії
09.06.2025	6 700 €	Пасажирські перевезення	Volkswagen Transporter	Тур вихідного дня	✎ ✖
08.06.2025	7 500 €	Пасажирські перевезення	Mercedes-Benz Vito	Шкільна поїздка	✎ ✖
05.06.2025	18 000 €	Вантажні перевезення	MAN TGS	Міжнародне перевезення	✎ ✖
30.05.2025	5 200 €	Партнерські доходи	Skoda Octavia	Спільний проект	✎ ✖
28.05.2025	6 200 €	Оренда	Mercedes-Benz Vito	Корпоративний захід	✎ ✖
19.05.2025	3 900 €	Оренда	Volkswagen Passat B7	Короткострокова оренда	✎ ✖
11.05.2025	17 500 €	Спеціалізовані роботи	MAN TGS	Транспортування спецобладнання	✎ ✖
30.04.2025	5 900 €	Оренда	Volkswagen Transporter	Корпоративний виїзд	✎ ✖
22.04.2025	5 100 €	Партнерські доходи	Toyota Corolla	Спільна акція з автосалоном	✎ ✖
10.04.2025	6 600 €	Пасажирські перевезення	Mercedes-Benz Vito	Перевезення працівників	✎ ✖
22.03.2025	16 000 €	Вантажні перевезення	MAN TGS	Трансфер техніки	✎ ✖
18.03.2025	6 400 €	Пасажирські перевезення	Volkswagen Transporter	Екскурсія по замках	✎ ✖
14.03.2025	5 200 €	Партнерські доходи	Volkswagen Passat B7	Спільний туристичний маршрут	✎ ✖
02.03.2025	4 500 €	Пасажирські перевезення	Skoda Octavia	Обслуговування заходу	✎ ✖

Рисунок 5.12 – Сторінка «Доходи»

Додавання нового запису здійснюється за допомогою кнопки «Додати дохід», після натискання на яку відкривається модальне вікно, як зображено на рисунку 5.13. У ньому необхідно вказати транспортний засіб, дату, суму, категорію та, за потреби, опис. Після збереження даних запис автоматично з'являється у таблиці доходів. Кожен запис можна редагувати або видалити, використовуючи відповідні кнопки в колонці «Дії».

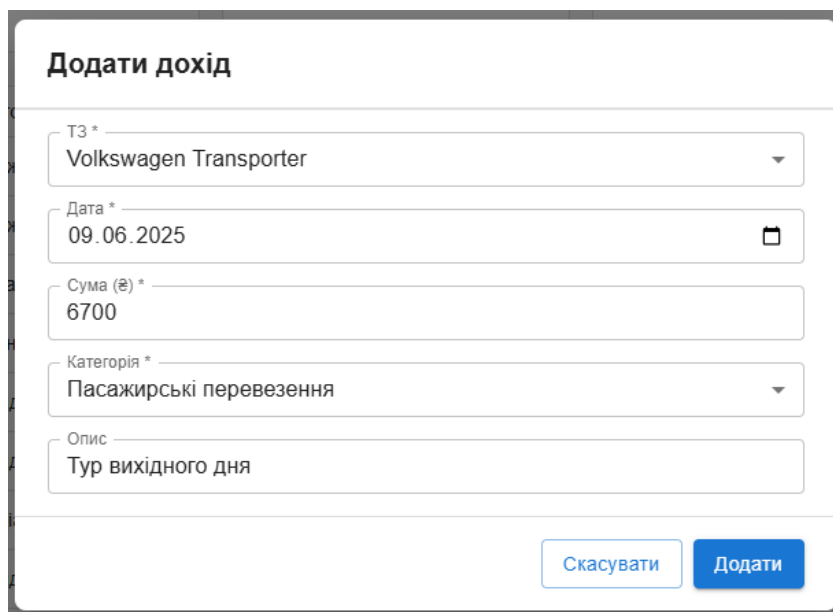


Рисунок 5.13 – Модальне вікно для додавання нового доходу

Однією з ключових функціональних частин системи є сторінка «Нагадування», яка відіграє важливу роль у забезпеченні своєчасного технічного обслуговування транспортних засобів. На цій вкладці користувач може створювати нагадування для майбутніх технічних обслуговувань, які необхідно виконати або за встановленим пробігом, або за визначеною датою. Для зручності використання передбачена можливість фільтрації нагадувань за транспортними засобами. У таблиці, окрім основної інформації, також відображається розрахований залишок пробігу або час до запланованого обслуговування, що дає змогу своєчасно реагувати на майбутні сервісні роботи. Це дозволяє користувачеві оперативно оцінювати пріоритетність кожного з обслуговувань. Інтерфейс вкладки «Нагадування» зображено на рисунку 5.14.

Нагадування				
Тип обслуговування	ТЗ	Дата	Пробіг (Km)	Дії
Перевірка двигуна	Volkswagen Passat B7	10.06.2025 Прострочено на 5 днів	-	✓ ✎ 🗑
Заміна масла	Toyota Corolla	-	119 900 Km Прострочено на 100 км	✓ ✎ 🗑
Перевірка ходової частини	Skoda Octavia	20.07.2025 Залишилось 35 днів	180 000 Km Залишилось 10000 км	✓ ✎ 🗑
Заміна гальмівної рідини	Mercedes-Benz Vito	10.09.2025 Залишилось 87 днів	240 000 Km Залишилось 20000 км	✓ ✎ 🗑
Заміна фільтрів	Volkswagen Transporter	01.08.2025 Залишилось 47 днів	205 000 Km Залишилось 5000 км	✓ ✎ 🗑
Перевірка двигуна	MAN TGS	30.07.2025 Залишилось 45 днів	340 000 Km Залишилось 20000 км	✓ ✎ 🗑
Заміна масла	Mercedes-Benz Vito	-	242 000 Km Залишилось 22000 км	✓ ✎ 🗑
Технічне обслуговування	Volkswagen Passat B7	05.10.2025 Залишилось 112 днів	-	✓ ✎ 🗑

Рисунок 5.14 – Сторінка «Нагадування»

У модальному вікні створення нагадування вводяться всі необхідні параметри: тип обслуговування, пробіг або дата коли потрібно виконати обслуговування, транспортний засіб та примітки, як зображено на рисунку 5.15.

Додати нагадування

Тип обслуговування *
Заміна масла

ТЗ *
Volkswagen Passat B7

Дата
ДД.ММ.ГГГГ

Пробіг (Km)
155000

Примітки
Після пробігу 10 тис. км з моменту останньої заміни

Рисунок 5.15 – Модальне вікно для створення нагадування

Створені нагадування можна змінювати або видаляти, використовуючи відповідні кнопки. Якщо нагадування позначити як виконану, система автоматично переносить її у таблицю обслуговувань і зберігає як виконане технічне обслуговування.

Висновок до розділу 5

У цьому розділі було розглянуто процес розроблення інформаційної системи для обліку технічного обслуговування транспортних засобів, включаючи вибір архітектури програмного забезпечення, деталі реалізації клієнтської та серверної частин, а також особливості зберігання та обробки даних.

На основі ретельного аналізу функціональних і нефункціональних вимог було обрано архітектурний підхід із чітким поділом на клієнтську, серверну частини та рівень бази даних. Такий підхід дозволив забезпечити масштабованість системи, гнучкість у впровадженні нових функціональних модулів, а також незалежність інтерфейсу від логіки обробки даних.

Розроблення клієнтської частини було зосереджене на забезпеченні зручного, адаптивного та інтуїтивно зрозумілого інтерфейсу користувача. Завдяки використанню компонентного підходу та бібліотек інтерфейсних елементів було досягнуто високої гнучкості й повторного використання елементів UI. Особлива увага приділялася створенню зручної навігації, а також ефективному представленню даних, зокрема у вигляді таблиць, графіків і списків.

Серверна частина системи реалізує REST API, яке обробляє запити клієнта, здійснює перевірку даних та взаємодію з базою даних. Через ORM Sequelize було забезпечено прозору роботу з PostgreSQL, що спрощує реалізацію запитів, підвищує безпеку та підтримує цілісність даних. Застосування Docker для розгортання системи дозволяє забезпечити повторюваність середовища та простоту у перенесенні на різні хостинги.

Таким чином, розроблене рішення повністю відповідає вимогам, що були поставлені на початковому етапі проєктування. Архітектурні та технологічні рішення, застосовані у процесі реалізації, дозволили досягти високого рівня модульності, надійності та розширюваності програмного продукту. Це створює підґрунтя для подальшого розвитку системи, її масштабування та впровадження нових функціональних можливостей у майбутньому.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

6.1 Змістовна постановка задачі

Сучасний транспорт, незалежно від того, чи це приватний автомобіль, чи транспортний засіб у складі військового підрозділу, вимагає регулярного технічного обслуговування для забезпечення безпеки та надійності. Протягом експлуатації транспортного засобу проводяться різноманітні види робіт – від заміни мастила та фільтрів до перевірки гальм і інших систем, які впливають на його роботу. При цьому витрати на проведення технічного обслуговування мають бути мінімізовані, адже кожний ремонт, чи планове технічне обслуговування пов'язане з певними фінансовими витратами. Водночас сильне відхилення від оптимальних інтервалів обслуговування може призвести до збільшення ризику виникнення несправностей, що негативно впливає на надійність транспортного засобу.

Задача полягає у визначенні оптимальних інтервалів для проведення технічного обслуговування транспортного засобу з метою зниження загальних витрат на обслуговування та мінімізації ризику виникнення несправностей.

Для кожного виду робіт з обслуговування (наприклад, заміна мастила, заміна фільтрів, перевірка гальм тощо) встановлюються такі параметри:

– прямі витрати: витрати на виконання конкретного виду обслуговування, які включають оплату робіт, закупівлю запчастин та матеріалів;

– коефіцієнт ризику: коефіцієнт, що характеризує, наскільки зростає ймовірність несправності транспортного засобу при відхиленні від рекомендованого інтервалу обслуговування;

– допустимий інтервал пробігу (часу): інтервал пробігу (часу), за яким має проводитись обслуговування для забезпечення надійної роботи транспортного засобу.

Основна мета поставленої задачі – визначити такі значення інтервалів для кожного виду технічного обслуговування, які одночасно мінімізують загальні економічні витрати та знижують ризик виникнення несправностей транспортного

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

засобу. Економічну складову представляє сумарна сума витрат на виконання робіт, що залежить від обраного інтервалу обслуговування. Технічну складову характеризує ризик несправностей, який зростає, коли фактичний інтервал відрізняється від рекомендованого.

Отже, завдання полягає в тому, щоб знайти набір компромісних рішень (Pareto-оптимальних варіантів), де неможливо знизити економічні витрати без збільшення ризику несправностей або зменшити ризик без підвищення витрат. Це дозволить користувачу обрати оптимальний варіант обслуговування, який найкраще відповідає його пріоритетам, забезпечуючи економічну ефективність та надійність технічного стану транспортного засобу.

6.2 Математична постановка задачі

Вхідні дані:

n – кількість видів робіт під час технічного обслуговування (заміна мастила, фільтрів, тощо);

c_i – витрати на проведення відповідного технічного обслуговування (вартість робіт, запчастин тощо), де $i = \overline{1, n}$;

d_i – рекомендований інтервал обслуговування операції i , де $i = \overline{1, n}$;

r_i – коефіцієнт ризику, який визначає, наскільки швидко зростає ризик несправності при відхиленні від мінімально допустимого інтервалу, де $i = \overline{1, n}$;

x_i^{min} та x_i^{max} – нижня та верхня межа інтервалів для проведення технічного обслуговування або заміни деталі, де $i = \overline{1, n}$;

$S_i \in [1, 10]$ – критичність наслідків відмови вузла;

$O_i \in [1, 10]$ – ступінь ризику відмови вузла i після досягнення d_i .

Змінні:

$x = (x_1, x_2, \dots, x_n)$ – вектор рішень, де кожен x_i це обраний пробіг (час) для проведення i -го виду робіт.

Обґрунтування коефіцієнта критичності r_i :

Для кількісної оцінки критичності кожної операції використано FMEA-підхід. Він визначається на основі S_i та O_i :

$$w_i = S_i \cdot O_i \quad (6.1)$$

S_i відображає масштаби наслідків, де повна відмова двигуна ($S = 10$), перегрів гальм ($S = 9$), зростання витрати пального через засмічений повітряний фільтр ($S = 4$).

O_i оцінює ймовірність, що відмова трапиться до регламентного пробігу. Мастило деградує доволі швидко ($O = 5$), колодки стираються із середньою швидкістю ($O = 5$), а строк служби фільтру часто перевищує рекомендований інтервал ($O = 2$).

Нормалізація на $\max w_i$ переводить отримані бали у значення $[0;1]$, тож найкритичніший вузол завжди має $r = 1$:

$$r_i = \frac{w_i}{\max w_i} \in (0,1] \quad (6.2)$$

Обмеження:

$$x_i^{min} \leq x_i \leq x_i^{max} \quad (6.3)$$

$$x_i^{min} \leq d_i \leq x_i^{max} \quad (6.4)$$

$$c_i > 0 \quad (6.5)$$

$$r_i \geq 0 \quad (6.6)$$

Обмеження (6.3) гарантує, що значення обраного пробігу буде лежати у межах допустимого інтервалу. Обмеження (6.4) гарантує, що значення рекомендованого інтервалу буде лежати у межах допустимого інтервалу. Обмеження (6.5) гарантує, що витрати на проведення технічного обслуговування,

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

чи ремонтних робіт завжди будуть більше нуля, а обмеження (6.6) гарантує, що коефіцієнт ризику не може бути від'ємним.

Цільові функції:

Задача має мультицільовий характер, тому ми одночасно мінімізуємо дві функції:

Функція витрат (C_1):

Частота проведення робіт зменшується зі збільшенням інтервалу x_i . Для нормалізації використаємо співвідношення:

$$I(x_i) = \frac{d_i}{x_i}, \quad (6.7)$$

яке задовольняє $I(x_i) = 1$, при $x_i = d_i$ та $I(x_i) < 1$ при $x_i > d_i$. Таким чином, функція сумарних витрат має наступний вигляд:

$$C_1(x) = \sum_{i=1}^n c_i \cdot I(x_i) \quad (6.8)$$

Ця функція відображає, що при використанні мінімального інтервалу витрати на проведення робіт максимальні, а збільшення x_i зменшує частоту проведення робіт i , відповідно, витрати.

Функція ризику (C_2):

Ризик виникнення несправностей не зростає, якщо $x_i \leq d_i$, і збільшується при відхиленні x_i від d_i , тобто:

$$\rho_i(x_i) = \begin{cases} 0, & x_i \leq d_i \\ r_i \left(\frac{x_i - d_i}{d_i} \right)^2, & x_i > d_i \end{cases} \quad (6.9)$$

Функція сумарного ризику виходу з ладу вузла набуває наступного вигляду:

$$C_2(x) = \sum_{i=1}^n \rho_i(x_i) \quad (6.10)$$

Квадратичне зростання відображає, що невелика прострочка виконання технічного обслуговування майже не несе за собою серйозних наслідків, а значне перевищення регламенту стрімко підвищує ймовірність відмови.

Необхідно знайти такий вектор параметрів $x = (x_1, x_2, \dots, x_n)$, для якого обидві цільові функції

$$C_1(x) \Rightarrow \min,$$

$$C_2(x) \Rightarrow \min$$

досягнуть мінімальних значень одночасно. Тобто потрібно знайти набір компромісних рішень, що формує фронт Парето, який задовольняє умову:

$$\nexists x': (C_1(x') \leq C_1(x), (C_2(x') \leq C_2(x))$$

з принаймні однією строгою нерівністю.

Таким чином, задача оптимізації зводиться до пошуку множини рішень $\{x^*\}$, що є не домінованим за Парето-сортуванням для вектор-функції (C_1, C_2) .

6.3 Обґрунтування методу розв'язання

Задача оптимізації інтервалів проведення технічного обслуговування транспортного засобу має мультицільовий характер: з одного боку, необхідно мінімізувати економічні витрати (які зменшуються із збільшенням інтервалів між проведенням робіт), а з іншого – мінімізувати ризик несправностей, який зростає при відхиленні від мінімального інтервалу. Ці цілі є суперечливими, оскільки рідше проведення технічного обслуговування дозволяє знизити витрати, але одночасно підвищує ризик відмови через затримку профілактичних заходів. Таким чином, метою задачі є знаходження такого набору інтервалів обслуговування, який забезпечить допустимий компроміс між економічною доцільністю та технічною надійністю.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61

Для розв'язання такої задачі ефективно застосовувати популяційні алгоритми, які за один запуск генерують ціле покоління кандидатних рішень, здатних наблизити весь спектр компромісів, тобто набір Pareto-оптимальних рішень. З-поміж різних підходів до популяційних алгоритмів можна виділити наступні загальні методи: генетичний алгоритм (NSGA-II), етод диференціальної еволюції (DE), метод оптимізації на основі рою (PSO).

Генетичний алгоритм (NSGA-II) [14]: Цей алгоритм імітує процес природного добору, де кожне рішення кодується як вектор оптимальних інтервалів $x = (x_1, x_2, \dots, x_n)$. За допомогою операторів селекції, схрещування та мутації популяція кандидатів еволюціонує у напрямку покращення цільових функцій. Оцінка рішень відбувається за допомогою двох критеріїв: економічних витрат $C_1(x)$ і ризику несправностей $C_2(x)$. Введення поняття Pareto-домінування дозволяє визначити, які рішення є не домінантними, тобто не можуть бути покращені по одному показнику без погіршення іншого.

Метод диференціальної еволюції (DE) [15]: У цьому підході нові рішення генеруються шляхом лінійних комбінацій існуючих кандидатів, що дозволяє ефективно досліджувати простір неперервних змінних. Основна перевага DE полягає в його швидкій збіжності та простоті налаштування параметрів, однак для багатокритеріальної оптимізації необхідно застосовувати додаткові механізми, що дозволяють вести зовнішній архів не домінантних рішень або використовувати Pareto-ранжування.

Метод оптимізації на основі рою (PSO) [16]: Цей метод імітує колективну поведінку організмів, де кожне рішення рухається в просторі пошуку, враховуючи власний досвід та досвід сусідів. У багатокритеріальній оптимізації PSO модифікується за рахунок ведення архіву Pareto-оптимальних рішень і вибору лідерів, які є орієнтирами для частинок. Основна перевага PSO – висока швидкість збіжності, проте для нашої задачі, де важлива повнота покриття всього фронту компромісів, необхідне додаткове налаштування параметрів для забезпечення різноманіття рішень.

Для поставленої задачі найбільш придатним є генетичний алгоритм. Цей

					IC13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		62

метод дозволяє отримати широкий набір компромісних рішень, які рівномірно охоплюють весь діапазон можливих варіантів балансу між витратами на обслуговування та ризиком несправностей. Крім того, NSGA-II гарантує збереження найкращих рішень протягом усіх поколінь, що забезпечує стабільність результатів при повторних запусках алгоритму. Також метод дозволить отримати повний набір Pareto-оптимальних рішень, з яких кінцевий користувач зможе вибрати оптимальний варіант залежно від власних пріоритетів та умов експлуатації транспортного засобу.

6.4 Опис методу розв'язання

6.4.1. Принцип роботи алгоритму NSGA-II

Розглянемо принцип роботи обраного генетичного алгоритму на основі поставленої задачі. NSGA-II – це вдосконалена версія генетичного алгоритму для багатокритеріальної оптимізації, розроблена спеціально для пошуку набору Pareto-оптимальних рішень. Основні етапи роботи алгоритму:

Етап 1. Ініціалізація.

Задається розмір популяції N та генерується початкова популяція P_0 з N випадкових рішень. Кожне рішення є вектором $x = (x_1, x_2, \dots, x_n)$, де x_i вибирається рівномірно з діапазону $[x_i^{min}, x_i^{max}]$.

Етап 2. Оцінка цільових функцій:

Для кожного рішення x обчислити значення функцій:

$$C_1(x) = \sum_{i=1}^n c_i \cdot I(x_i)$$

$$C_2(x) = \sum_{i=1}^n \rho_i(x_i)$$

Отже, кожне рішення представляється парою показників $(C_1(x), C_2(x))$.

Етап 3. Недомінантне сортування за Pareto-домінуванням:

Рішення популяції сортуються за принципом не домінантності. Рішення, які не домінуються іншими, формують перший фронт F_1 . Решта рішень сортуються по наступних фронтах F_2, F_3, \dots . Далі кожному рішенню присвоюється ранг відповідно до фронту, в якому воно знаходиться (чим нижчий ранг, тим краще).

Етап 4. Оцінка різноманіття (crowding distance):

Обчислюється метрика "crowding distance" для кожного рішення. Вона визначає, наскільки далеко розташовані сусідні рішення в просторі цілей. Рішення з більшим значенням crowding distance надаються перевагою при відборі, оскільки вони сприяють збереженню різноманіття.

Етап 5. Відбір батьків:

За допомогою турнірного відбору (де рішення порівнюються спочатку за рангом, а при однаковому рангу – за crowding distance) формується батьківська вибірка P_{sel} .

Етап 6. Генерація потомків:

Використовуються оператори схрещування та мутації для створення нового покоління Q . Після операції мутації проводиться перевірка, щоб кожне рішення залишалося у межах $[x_i^{min}, x_i^{max}]$.

Етап 7. Оновлення популяції:

Об'єднується поточна популяція P та покоління потомків Q до отримання об'єднаної множини $R = P \cup Q$ розміру $2N$. На R знову виконується недомінантне сортування, після чого з вибірки відбираються N кращих рішень для нового покоління P_{t+1} (при потребі використовується crowding distance для розподілу, якщо кількість рішень у певному фронті перевищує кількість вільних місць).

Етап 8. Критерій зупинки:

Алгоритм повторюється протягом G поколінь або до того моменту, коли покращення Pareto-фронту стає незначним.

Етап 9. Результат:

Після завершення алгоритму виводиться набір оптимальних рішень з фінальної популяції, що є наближенням до Pareto-фронту. Кожне рішення є

вектором інтервалів x і представляє один компромісний план технічного обслуговування: зниження витрат неможливо досягти без збільшення ризику несправностей і навпаки.

6.4.2. Приклад застосування алгоритму

Для ілюстративного прикладу роботи алгоритму NSGA-II сформуємо набір даних, що містить три найбільш поширені види регламентних робіт. Для кожної з них задано рекомендований інтервал обслуговування d_i , вартість виконання c_i , а також коефіцієнти критичності наслідків відмови S_i та ймовірності їх виникнення O_i . На основі цих параметрів згідно з методикою FMEA обчислюється ваговий коефіцієнт ризику $w_i = S_i \cdot O_i$, який нормалізується відносно найвищого значення w_i для отримання коефіцієнта r_i .

У таблиці 6.1 наведено вихідні дані, які використовуються для побудови цільових функцій витрат і ризику у прикладі оптимізації.

Таблиця 6.1 – Вхідні дані для трьох регламентних операцій

Операція	Рекомендований інтервал d_i , км	Вартість c_i , ум.о.	Критичність наслідків S_i	Ймовірність відмови O_i	FMEA-вага $w_i = S_i \cdot O_i$	Нормований коефіцієнт ризику $r_i = \frac{w_i}{\max w_i}$
Заміна моторного мастила	8000	100	10	5	50	1
Заміна гальвоних колодок	9000	120	9	3	27	0.54
Заміна повітряного фільтра двигуна	10000	50	4	2	16	0.16

Для нашого значення x_i задамо межі в яких ми хочемо отримати результати $[x_i^{min}, x_i^{max}] = [8000, 15000]$.

Функція витрат матиме наступний вигляд:

$$C_1(x) = \sum_{i=1}^3 c_i \cdot \frac{d_i}{x_i} = c_1 \cdot \frac{8000}{x_1} + c_2 \cdot \frac{9000}{x_2} + c_3 \cdot \frac{10000}{x_3}$$

Функція ризику виникнення несправностей матиме наступний вигляд:

$$C_2(x) = \sum_{i=1}^3 r_i \cdot \left(\frac{x_i - d_i}{d_i} \right)^2 = r_1 \cdot \left(\frac{x_1 - 8000}{8000} \right)^2 + r_2 \cdot \left(\frac{x_2 - 9000}{9000} \right)^2 + r_3 \cdot \left(\frac{x_3 - 10000}{10000} \right)^2$$

Для демонстрації візьмемо єдиний інтервал x для всіх трьох робіт

1. Рішення А:

Вибираємо $x = (8000, 8000, 8000)$ км.

Складова витрат $C_1(x)$:

$$C_1(A) = 100 \cdot \frac{8000}{8000} + 120 \cdot \frac{9000}{8000} + 50 \cdot \frac{10000}{8000} = 100 + 135 + 62.5 = 297.5 \text{ у.о.}$$

Складова ризику:

$$(\text{усі } x \leq d_i) \Rightarrow C_2 = 0.$$

2. Рішення В:

Вибираємо $x = (10000, 10000, 10000)$ км.

Складова витрат $C_1(x)$:

$$C_1(B) = 100 \cdot \frac{8000}{10000} + 120 \cdot \frac{9000}{10000} + 50 \cdot \frac{10000}{10000} = 238 \text{ у.о.}$$

Складова ризику.

Коефіцієнт ризику від прострочення термінів заміни мастила:

$$\rho_1 = 1 \cdot \left(\frac{10000 - 8000}{8000} \right)^2 = (0.25)^2 = 0.0625$$

Коефіцієнт ризику від прострочення термінів заміни гальмівних колодок:

$$\rho_2 = 0.54 \cdot \left(\frac{10000 - 9000}{9000} \right)^2 = 0.54 \cdot (0.111\bar{1})^2 \approx 0.0067$$

Коефіцієнт ризику від прострочення термінів заміни повітряного фільтра:

$$(x = d_3) \Rightarrow \rho_3 = 0$$

Розрахуємо сумарний ризик виникнення несправностей:

$$C_2 \approx 0.0625 + 0.0067 \approx 0.069$$

									Арк.
									66
Зм.	Лист	№ докум.	Підпис	Дата					

3. Рішення С:

Вибираємо $x = (13000, 13000, 13000)$ км.

Складові витрати $C_1(x)$:

$$C_1(C) = 100 \cdot \frac{8000}{13000} + 120 \cdot \frac{9000}{13000} + 50 \cdot \frac{10000}{13000} = 183.1 \text{ у. о.}$$

Складові ризику.

Коефіцієнт ризику від прострочення термінів заміни мастила:

$$\rho_1 = 1 \cdot \left(\frac{13000 - 8000}{8000} \right)^2 = (0.625)^2 = 0.3906$$

Коефіцієнт ризику від прострочення термінів заміни гальмівних колодок:

$$\rho_2 = 0.54 \cdot \left(\frac{13000 - 9000}{9000} \right)^2 = 0.54 \cdot (0.444\bar{4})^2 \approx 0.1067$$

Коефіцієнт ризику від прострочення термінів заміни повітряного фільтра:

$$\rho_3 = 0.16 \cdot \left(\frac{13000 - 10000}{10000} \right)^2 = 0.16 \cdot (0.3)^2 = 0.0144$$

Розрахуємо сумарний ризик виникнення несправностей:

$$C_2 \approx 0.3906 + 0.1067 + 0.014 \approx 0.5113$$

4. Рішення D:

Вибираємо $x = (15000, 15000, 15000)$ км.

Складові витрати $C_1(x)$:

$$C_1(D) = 100 \cdot \frac{8000}{15000} + 120 \cdot \frac{9000}{15000} + 50 \cdot \frac{10000}{15000} = 158.7 \text{ у. о.}$$

Складові ризику.

Коефіцієнт ризику від прострочення термінів заміни мастила:

$$\rho_1 = 1 \cdot \left(\frac{15000 - 8000}{8000} \right)^2 = (0.875)^2 = 0.7656$$

Коефіцієнт ризику від прострочення термінів заміни гальмівних колодок:

$$\rho_2 = 0.54 \cdot \left(\frac{13000 - 9000}{9000} \right)^2 = 0.54 \cdot (0.666\bar{6})^2 \approx 0.24$$

Коефіцієнт ризику від прострочення термінів заміни повітряного фільтра:

$$\rho_3 = 0.16 \cdot \left(\frac{15000 - 10000}{10000} \right)^2 = 0.16 \cdot (0.5)^2 = 0.04$$

Зм.	Лист	№ докум.	Підпис	Дата

Розрахуємо сумарний ризик виникнення несправностей:

$$C_2 \approx 0.7656 + 0.1067 + 0.014 \approx 1$$

Порівняння знайдених Pareto фронтів наведено у таблиці 6.2.

Таблиця 6.2 – Порівняння Pareto фронтів

Пробіг(км)	Витрати (у.о)	Ризик
8000	297.5	0
10000	238	0.069
13000	183.1	0.5113
15000	158.7	1

Отримані результати демонструють, що рішення А забезпечує мінімальний ризик при максимальних витратах, рішення В та D є компромісними, де В є більш економічно вигідним, проте більш ризикованим відносно D. В свою чергу варіант С є невиправдано ризикованим, хоча й найвигіднішим.

Таким чином, алгоритм NSGA-II дозволяє отримати набір Pareto-оптимальних рішень, що представляє повний спектр компромісів між витратами на технічні обслуговування та ризиком несправностей. Кінцевий користувач може обрати той варіант плану технічного обслуговування, який найкраще відповідає його пріоритетам – для критично важливої безпеки можливо вибрати рішення, близьке до Рішення А, а якщо пріоритетом є економія коштів, тоді доцільніше обрати компроміс, подібне до Рішення В чи D, за умови прийняттого рівня ризику.

Висновок до розділу 6

У розділі було сформульовано та математично обґрунтовано задачу оптимізації інтервалів проведення технічного обслуговування транспортного засобу з урахуванням двох суперечливих критеріїв: економічних витрат та ризику виникнення несправностей. Модель враховує, що при використанні мінімально допустимих інтервалів витрати максимальні, однак ризик виникнення несправностей практично відсутній, тоді як збільшення інтервалів дозволяє

знизити витрати, але призводить до зростання ризику. Такий підхід дає можливість побудувати повний спектр компромісних рішень, що формують Pareto-фронт.

Для розв'язання задачі застосовано популяційний метод на основі генетичного алгоритму NSGA-II. Цей алгоритм здатний ефективно дослідити простір можливих рішень і згенерувати набір оптимальних інтервалів для проведення технічних обслуговувань. Отриманий Pareto-фронт дозволяє обирати варіант, який найкраще відповідає їх пріоритетам – від оптимізації витрат до максимального зниження ризику несправностей.

Таким чином, результати роботи свідчать, що запропонована математична модель коректно описує залежність між інтервалами обслуговування, витратами та ризиком виникнення несправностей, що дозволяє формувати цільові функції для подальшої оптимізації. Застосування NSGA-II забезпечує можливість одночасної мінімізації економічних та технічних критеріїв збереження надійності транспортних засобів. Побудований Pareto-фронт дає змогу користувачу обирати оптимальний баланс між мінімізацією витрат і зменшенням ризику несправностей залежно від пріоритетів та умов експлуатації. Запропонований підхід має практичне значення як для звичайних автовласників, так і для автопарків, сприяючи оптимізації ресурсів, зниженню витрат та підвищенню безпеки експлуатації транспортних засобів.

ВИСНОВКИ

У ході дипломного проекту було розроблено повноцінну інформаційну систему підтримки процесів технічного обслуговування транспортних засобів, яка охоплює широкий спектр задач. Система демонструє високий рівень інтеграції сучасних вебтехнологій і відповідає актуальному рівню розвитку інформаційних технологій.

У рамках роботи повністю досягнуті мета та завдання, визначені у першому розділі. Реалізовано всі ключові функціональні модулі, включно з керування даними про транспортні засоби, обліком витрат і доходів із підтримкою категорій, а також плануванням технічного обслуговування з автоматизованими нагадуваннями. Завдяки злагодженій роботі всіх підсистем система забезпечує повноцінне функціонування та відповідає поставленим вимогам.

Практичне значення розробленої системи полягає у значному підвищенні ефективності управління процесами технічного обслуговування транспортних засобів. Система забезпечує повну автоматизацію процесу обліку витрат, доходів і планування технічних робіт, що дозволяє уникати помилок, оптимізувати витрати і покращити контроль над станом техніки. Це особливо актуально для підприємств з великими автопарками, логістичних компаній, а також у державних структурах, де важливо підтримувати технічний стан транспорту на високому рівні.

Використання системи в реальних умовах продемонструвало її здатність стабільно працювати при значних навантаженнях та зберігати ефективність навіть при великому обсязі даних. Гнучкість архітектури дозволяє легко адаптувати її до різних галузей використання. Перспективним напрямком подальших досліджень є інтеграція додаткових аналітичних інструментів, що дозволять поглиблено аналізувати витрати та оптимізувати процеси обслуговування.

Таким чином, у рамках дипломного проекту було створено ефективну, надійну та універсальну інформаційну систему, яка відповідає всім поставленим завданням та демонструє значний потенціал для подальшого впровадження в реальних умовах експлуатації.

					ІС13.200БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		70

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Fuelio. Vehicle management.* URL: <https://www.fuel.io/>.
2. *App Store Preview. Car Maintenance Reminders.* URL: <https://apps.apple.com/us/app/car-maintenance-reminders/id1617869280>.
3. *Simply Auto. Perfect Companion for your vehicle.* URL: <https://simplyauto.app/>.
4. *Loggy. Welcome to Loggy.* URL: <https://www.loggy.com/>.
5. *React. The library for web and native user interfaces.* URL: <https://react.dev/>.
6. *React Router. A user-obsessed, standards-focused, multi-strategy router you can deploy anywhere..* URL: <https://reactrouter.com/>.
7. *TypeScript. What is TypeScript?.* URL: <https://www.typescriptlang.org/>.
8. *Vite. The Build Tool for the Web.* URL: <https://vite.dev/>.
9. *Mui. Move faster with intuitive React UI tools.* URL: <https://mui.com/>.
10. *NodeJS. Run JavaScript Everywhere.* URL: <https://nodejs.org/en>.
11. *ExpressJS. Fast, unopinionated, minimalist web framework for Node.js.* URL: <https://expressjs.com/>.
12. *Sequelize. Modern TypeScript and Node.js ORM.* URL: <https://sequelize.org/>.
13. *PostgreSQL. The World's Most Advanced Open Source Relational Database.* URL: <https://www.postgresql.org/>.
14. *GeeksforGeeks. Non-Dominated Sorting Genetic Algorithm 2 (NSGA-II).* URL: <https://www.geeksforgeeks.org/deep-learning/non-dominated-sorting-genetic-algorithm-2-nsga-ii/>.
15. Kumar R. *Differential evolution. Cornell University Computational Optimization Open Textbook.* URL: https://optimization.cbe.cornell.edu/index.php?title=Differential_evolution.
16. Particle swarm optimization / D. Schluneker та ін. *Cornell University Computational Optimization Open Textbook.* URL: https://optimization.cbe.cornell.edu/index.php?title=Particle_swarm_optimization.