

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

**ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-
АНАЛІТИЧНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ТЕХНІЧНОГО
ОГЛЯДУ ТРАНСПОРТНИХ ЗАСОБІВ**
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-22-1
Мізяк Д. О.
(прізвище, ініціали)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Інформатика
(повна назва освітньої програми)
Керівник проф. Шафроненко А. Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики Кобилін О. А.
(підпис) (прізвище, ініціали)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Мізяку Дмитру Олександровичу
(прізвище, ім'я, по батькові)1. Тема роботи Проектування та реалізація інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів

затверджена наказом університету від 26 травня 2026 року № 435Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 25 червня 2026 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література; нормативно-правові документи у сфері технічного огляду транспортних засобів; матеріали конференцій; дані мережі Інтернет; технічна документація з технологій ASP.NET Core, SQL Server, Docker та мікросервісної архітектури.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз предметної області технічного огляду транспортних засобів та існуючих програмних рішень для автоматизації процесів обліку і контролю.2. Проектування архітектури інформаційно-аналітичної системи на основі мікросервісного підходу, розроблення структури бази даних та взаємодії компонентів системи.3. Розроблення та реалізація програмного забезпечення, тестування функціональних можливостей системи та оцінка ефективності її використання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) діаграма варіантів використання інформаційної системи; структурна схема архітектури мікросервісної системи; логічна модель бази даних; діаграми послідовності взаємодії компонентів системи; схема взаємодії мікросервісів через API Gateway; інтерфейси користувача (форми авторизації, реєстрації та управління даними); приклад сформованого звіту технічного огляду транспортних засобів.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	14.04.26-16.04.26	
3	Аналіз літератури з проблеми, що вирішується	17.04.26-19.04.26	
4	Аналіз існуючих методів проектування	20.04.26-22.04.26	
5	Формування кроків вирішення проблеми	23.04.26-05.05.26	
6	Програмна реалізація	26.04.26-20.05.26	
7	Аналіз отриманих результатів	21.05.26-04.06.26	
8	Оформлення пояснювальної записки	05.06.26-15.06.26	
9	Перевірка на нормоконтроль	16.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Шафроненко А. Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 101 с., 4 табл., 33 рис., 3 дод., 32 джерела.

ІНФОРМАЦІЙНА СИСТЕМА, МІКРОСЕРВІСНА АРХІТЕКТУРА, REST API, ASP.NET CORE, ТЕХНІЧНИЙ ОГЛЯД, SQL SERVER.

Об'єктом роботи є процес автоматизації технічного огляду транспортних засобів.

Метою роботи є розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів на основі мікросервісної архітектури.

Під час роботи використано методи об'єктно-орієнтованого проєктування, REST API, технології ASP.NET Core, Docker, JWT та засоби веброзробки.

Практична цінність роботи полягає в автоматизації процесів технічного огляду, підвищенні швидкості обробки даних та покращенні контролю діяльності підприємства.

У результаті роботи розроблено інформаційно-аналітичну систему, яка забезпечує реєстрацію клієнтів і транспортних засобів, проведення технічного огляду, формування звітності та аналітичну обробку даних.

Область застосування: станції технічного огляду транспортних засобів та підприємства автомобільного сервісу.

INFORMATION SYSTEM, MICROSERVICE ARCHITECTURE, REST API, ASP.NET CORE, TECHNICAL INSPECTION, SQL SERVER.

The object of the work is the process of automation of vehicle technical inspection.

The purpose of the work is to develop an information and analytical system for automation of vehicle technical inspection based on microservice architecture.

The work uses object-oriented design methods, REST API, ASP.NET Core technologies, Docker, JWT, and web development tools.

The practical value of the work lies in automating technical inspection processes, increasing data processing speed, and improving enterprise activity control.

As a result of the work, an information and analytical system was developed that provides registration of clients and vehicles, technical inspection procedures, reporting, and analytical data processing.

Field of application: vehicle technical inspection stations and automotive service enterprises.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	8
1 Аналіз предметної області та існуючих рішень.....	9
1.1 Аналіз предметної області та постановка проблеми.....	9
1.2 Аналіз існуючих програмних рішень	13
1.3 Аналіз сучасних підходів та технологій розроблення інформаційних систем.....	18
1.4 Постановка задачі	22
2 Проєктування та моделювання інформаційної системи	24
2.1 Аналіз функціональних вимог та моделювання предметної області	24
2.2 Проєктування архітектури інформаційної системи	29
2.3 Проєктування структури бази даних та інформаційних потоків ...	35
2.4 Моделювання аналітичної підсистеми та формування звітовності.....	41
3 Розроблення та реалізація інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів.....	47
3.1 Проєктування архітектури та структури системи	47
3.2 Реалізація функціональних модулів системи.....	59
3.3 Реалізація аналітичної підсистеми та формування звітовності	66
3.4 Тестування та оцінювання результатів розроблення	72
3.5 Перспективи розвитку системи	76
Висновки.....	79
Перелік джерел посилання.....	81
Додаток А Графічні матеріали інформаційної системи.....	84
Додаток Б Інтерфейс користувача інформаційно-аналітичної системи.....	91
Додаток В Приклад звіту сформованого системою	98
Додаток Г Посилання на репозиторій вихідного коду програми	101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

ІАС – інформаційно-аналітична система

ПЗ – програмне забезпечення

СТО – станція технічного обслуговування

ТО – технічний огляд транспортного засобу

API – Application Programming Interface (інтерфейс програмування застосунків)

ASP.NET Core – Active Server Pages .NET Core (платформа для розроблення вебзастосунків)

CRUD – Create, Read, Update, Delete (створення, читання, оновлення та видалення даних)

CI/CD – Continuous Integration / Continuous Delivery (безперервна інтеграція та безперервне розгортання)

Docker – Docker Platform (платформа контейнеризації програмного забезпечення)

ER-model – Entity Relationship Model (модель «сутність–зв’язок»)

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)

HTTPS – HyperText Transfer Protocol Secure (захищений протокол передачі гіпертексту)

JSON – JavaScript Object Notation (текстовий формат обміну даними)

JWT – JSON Web Token (токен авторизації)

KPI – Key Performance Indicators (ключові показники ефективності)

REST – Representational State Transfer (архітектурний стиль побудови вебсервісів)

REST API – Representational State Transfer Application Programming Interface (інтерфейс програмування застосунків на основі REST)

SQL – Structured Query Language (мова структурованих запитів)

SQL Server – Structured Query Language Server (система керування базами даних Microsoft)

UML – Unified Modeling Language (уніфікована мова моделювання)

VIN – Vehicle Identification Number (ідентифікаційний номер транспортного засобу)

ВСТУП

У сучасних умовах розвитку цифрових технологій автоматизація бізнес-процесів є одним із ключових напрямів підвищення ефективності діяльності підприємств. Особливо актуальним це питання є для підприємств автомобільної галузі, зокрема станцій технічного огляду транспортних засобів, де щоденно виконується значний обсяг операцій із реєстрації клієнтів, обробки технічної інформації, проведення перевірок та формування звітної документації. Використання застарілих методів ведення обліку або частково автоматизованих рішень призводить до збільшення часу обробки даних, підвищення ризику виникнення помилок та ускладнення контролю за діяльністю підприємства.

Сучасні інформаційні системи дозволяють значно оптимізувати процеси технічного огляду транспортних засобів шляхом централізованого зберігання даних, автоматизації обробки інформації та використання аналітичних інструментів для підтримки управлінських рішень. Особливого значення набуває застосування мікросервісної архітектури, яка забезпечує гнучкість, масштабованість, незалежність компонентів та спрощує подальший розвиток системи.

На сьогодні існує велика кількість інформаційних систем для автоматизації діяльності підприємств автомобільного сервісу. Проте значна частина таких рішень орієнтована на вузьке коло задач або побудована за монолітним принципом, що ускладнює їх масштабування та подальший розвиток. Крім того, багато існуючих систем не забезпечують достатнього рівня аналітичної обробки даних та підтримки управлінської звітності.

Актуальність роботи полягає у створенні інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів, що забезпечує автоматизацію бізнес-процесів, управління інформаційними потоками та підтримку прийняття управлінських рішень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз предметної області та постановка проблеми

За сучасних умов контроль технічного стану автомобілів виконує важливу роль у підтриманні належного рівня безпеки учасників дорожнього руху. Значення цієї процедури виходить далеко за межі перевірки працездатності окремих елементів транспортного засобу. Не менш важливим завданням технічного огляду є контроль відповідності транспортних засобів екологічним нормам, значення яких зростає внаслідок посилення вимог до охорони довкілля та розвитку цифрових технологій. Постійне зростання автомобільного парку та поява дедалі складніших технічних рішень у конструкції транспортних засобів потребують підвищення якості діагностичних процедур. За таких умов особливого значення набувають нові підходи до організації технічного огляду, серед яких провідне місце займає комплексна автоматизація робочих процесів.

Процес проведення технічного огляду характеризується складною структурою та включає значну кількість взаємопов'язаних операцій. До його складу належать реєстрація транспортного засобу в системі, виконання інструментальних і візуальних перевірок, фіксація отриманих технічних показників, підготовка експертних висновків, а також накопичення та зберігання історії обслуговування кожного об'єкта. Усі перелічені етапи тісно інтегровані між собою та супроводжуються обробкою великих масивів даних. За таких умов виникає потреба у використанні програмних засобів, які дозволяють оперативно опрацьовувати інформацію, зберігати її цілісність та мінімізувати затримки під час виконання робочих операцій. Узагальнену послідовність виконання технічного огляду транспортного засобу наведено на рисунку 1.1.

Узагальнена схема процесу технічного огляду транспортного засобу



Рисунок 1.1 – Узагальнена схема процесу технічного огляду транспортного засобу

Однією з характерних рис розглянутої сфери діяльності є необхідність забезпечення високої достовірності даних. Неточності, що виникають на етапах внесення, опрацювання чи накопичення відомостей, можуть призвести до некоректної оцінки технічного стану транспортного засобу та, як наслідок, до формування хибних висновків. З огляду на це сучасні інформаційні системи повинні підтримувати автоматичну перевірку введених даних, контролювати їх повноту та забезпечувати механізми виявлення й усунення можливих невідповідностей.

Водночас технічний огляд є чітко регламентованим процесом. Переважна більшість перевірок виконується відповідно до чинних нормативних документів, стандартів та затверджених процедур. Це висуває додаткові вимоги до програмного забезпечення, яке повинно забезпечувати формування необхідної документації, ведення електронних журналів та швидкий доступ до інформації, необхідної під час виконання перевірок.

Отже, предметна область технічного огляду транспортних засобів характеризується складною структурою інформаційних взаємозв'язків, значною кількістю учасників процесу та високими вимогами до достовірності й оперативності обробки даних. Сукупність зазначених

факторів підтверджує доцільність використання сучасних інформаційно-аналітичних систем для автоматизації відповідних процедур.

Хоча цифрові технології активно впроваджуються в різних сферах діяльності, значна кількість підприємств технічного контролю досі використовує застарілі підходи до роботи з даними. У багатьох випадках відомості накопичуються в паперовій документації, окремих електронних файлах або локальних програмних продуктах. Такий підхід ускладнює обмін інформацією, створює передумови для дублювання даних та підвищує ймовірність виникнення помилок під час ручного опрацювання відомостей.

Додатковою складністю є необхідність одночасної роботи з великим обсягом різномірної інформації. Під час проведення технічного огляду працівники опрацьовують дані про власників транспортних засобів, результати перевірок, технічні характеристики автомобілів та супровідну документацію. За відсутності єдиної інформаційної системи це призводить до збільшення витрат часу на виконання операцій і підвищує ризик виникнення помилок, пов'язаних із ручною обробкою даних. Основні проблеми, притаманні сучасній організації процесів технічного огляду, узагальнено наведено на рисунку 1.2.

Серйозною проблемою залишається розподілене зберігання інформації між кількома незалежними джерелами. Якщо дані зберігаються у різних джерелах, швидко отримувати достовірні та актуальні відомості про технічний стан транспортних засобів стає значно складніше. Це негативно впливає на швидкість роботи персоналу та ускладнює прийняття обґрунтованих управлінських рішень.

Це, у свою чергу, негативно впливає на якість взаємодії між різними підрозділами підприємства та значно ускладнює виконання аналітичних завдань, пов'язаних з обробкою накопичених даних. У результаті адміністративний персонал і керівництво не мають оперативного доступу до повної та актуальної інформації, необхідної для прийняття обґрунтованих управлінських рішень і своєчасного реагування на проблемні ситуації.

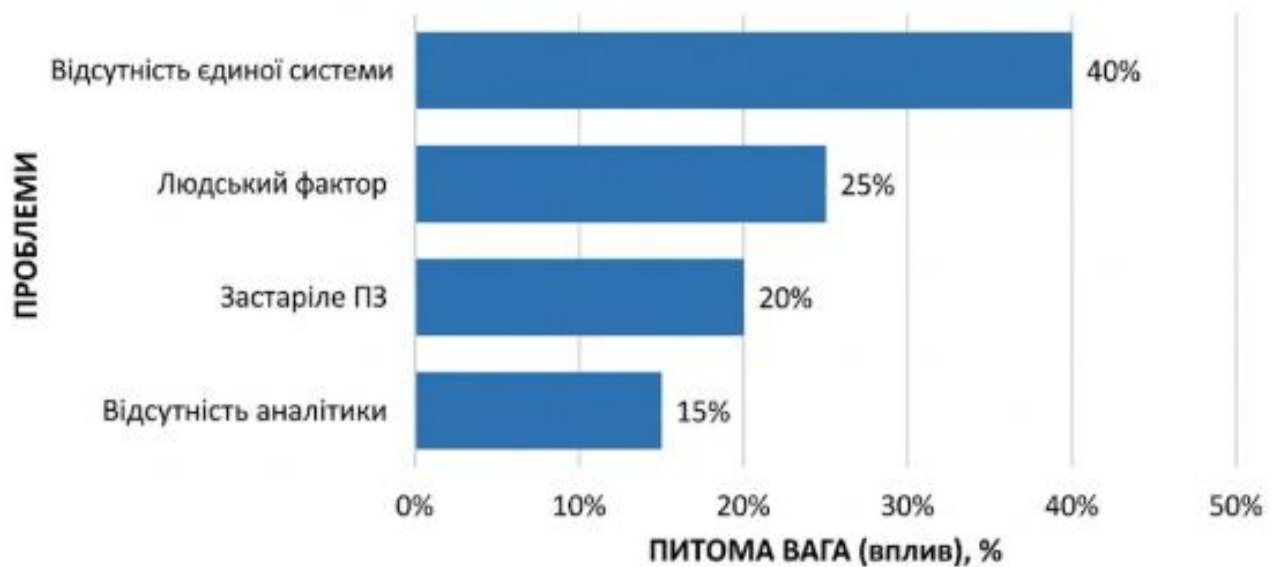


Рисунок 1.2 – Основні проблеми організації процесів технічного огляду

Суттєвою проблемою залишається значна кількість операцій, виконання яких безпосередньо залежить від дій персоналу. За відсутності належної автоматизації значна частина операцій виконується вручну, що підвищує ризик помилок під час введення, перевірки та опрацювання даних. Навіть дрібні помилки під час опрацювання інформації здатні спотворювати результати перевірок та впливати на подальші рішення щодо експлуатації транспортного засобу. У довгостроковій перспективі це може спричиняти як додаткові витрати на виправлення помилок, так і зниження загального рівня безпеки дорожнього руху через допуск до експлуатації транспортних засобів із невиявленими несправностями.

Окремою проблемою є недостатнє використання сучасних аналітичних інструментів. Значні обсяги даних, що формуються під час повсякденної діяльності підприємства, часто використовуються лише для поточного обліку та зберігання. Водночас можливості використання цих даних для пошуку тенденцій, прогнозування ризиків і підтримки прийняття рішень використовуються лише частково. Відсутність інтегрованих механізмів візуалізації даних і автоматизованого формування звітів істотно обмежує можливості оцінювання ключових показників діяльності, аналізу

ефективності роботи персоналу та довгострокового планування розвитку підприємства.

Сучасні підходи до створення програмного забезпечення орієнтовані на забезпечення високого рівня цифровізації, інтеграції та масштабованості інформаційних систем. Водночас значна кількість рішень, що використовуються сьогодні, базується на застарілих архітектурних підходах і не враховує можливостей сучасних хмарних технологій, мікросервісної архітектури та засобів автоматизації.

Проведений аналіз дозволяє стверджувати, що предметна область технічного огляду транспортних засобів досі стикається з низкою невирішених проблем. До них належать недостатній рівень цифровізації процесів, відсутність єдиного інформаційного середовища та обмежені можливості для аналітичної обробки даних. Саме ці фактори формують потребу у створенні сучасної інформаційно-аналітичної системи, здатної забезпечити комплексну автоматизацію процесів технічного огляду, підвищити надійність роботи з даними та надати ефективні інструменти для аналізу отриманих результатів.

1.2 Аналіз існуючих програмних рішень

У сучасних умовах стрімкого розвитку інформаційних технологій автоматизація процесів обліку, операційного контролю та аналітичної обробки даних здійснюється за допомогою широкого спектра програмних продуктів. У сфері технічного огляду транспортних засобів використовуються як вузькоспеціалізовані рішення, орієнтовані на виконання окремих функцій, так і універсальні платформи, які можуть адаптуватися до потреб конкретного підприємства. Аналіз існуючих програмних продуктів дозволяє виділити кілька основних категорій рішень, кожна з яких має власні переваги та обмеження.

До першого класу програмного забезпечення можна віднести системи, призначені для автоматизації діяльності автосервісів і станцій технічного обслуговування. Такі програмні продукти, як правило, орієнтовані на підтримку внутрішніх бізнес-процесів сервісного центру. Типовий функціонал таких рішень включає ведення клієнтської бази, облік транспортних засобів, планування завантаження персоналу, формування замовлень-нарядів, контроль фінансових операцій та організацію взаємодії з клієнтами. Перевагою подібних систем є швидке впровадження та наявність готових механізмів автоматизації типових процесів. Водночас більшість із них розроблена з урахуванням потреб сервісного обслуговування, а не процедур обов'язкового технічного контролю транспортних засобів. Через це в них часто відсутні інструменти для роботи зі стандартизованими діагностичними картами, інтеграції з вимірювальним обладнанням та автоматизованого контролю відповідності транспортного засобу встановленим нормативним вимогам.

Окремий напрям програмного забезпечення представлений CRM-платформами (Customer Relationship Management), основним призначенням яких є управління взаємовідносинами з клієнтами. Такі рішення забезпечують централізоване зберігання контактної інформації, підтримують комунікацію з клієнтами, допомагають планувати звернення та автоматизують окремі маркетингові процеси. Для підприємств, що здійснюють технічний огляд транспортних засобів, CRM-системи можуть бути корисними як інструмент ведення клієнтської бази та організації нагадувань про необхідність проходження чергового огляду. Однак їх функціональність зазвичай не охоплює завдання, пов'язані з безпосереднім проведенням технічного контролю. Зокрема, такі системи не забезпечують детального обліку результатів перевірок, не підтримують формування технічних висновків на основі інструментальних вимірювань та не призначені для ведення повної історії технічного стану транспортних засобів.

До окремого класу програмних продуктів належать державні інформаційні платформи та галузеві реєстри та галузеві реєстри. Їх основне призначення полягає у централізованому накопиченні, зберіганні та обміні інформацією про результати технічного контролю транспортних засобів відповідно до встановлених нормативних вимог. До переваг таких систем належать використання єдиних стандартів обліку даних, висока надійність зберігання інформації та можливість взаємодії між різними організаціями. Водночас для підприємств, які безпосередньо виконують технічний огляд, функціональність подібних систем часто є недостатньою. Зазвичай вони орієнтовані на накопичення та передачу інформації, мають обмежені можливості налаштування інтерфейсу й практично не містять інструментів для внутрішньої аналітики, автоматизованого формування звітності або інтеграції з локальними бізнес-процесами підприємства.

До наступної категорії належать ERP-системи (Enterprise Resource Planning), які призначені для комплексного управління ресурсами організації. Такі програмні платформи дозволяють об'єднати в межах єдиного інформаційного середовища фінансові, кадрові, виробничі та адміністративні процеси. Їх використання забезпечує високий рівень інтеграції даних і централізоване управління діяльністю підприємства. Проте застосування ERP-рішень у сфері технічного огляду транспортних засобів має низку обмежень. Насамперед це пов'язано з високою вартістю впровадження та супроводу, складністю налаштування і надлишковістю функціоналу для більшості пунктів технічного контролю. Крім того, такі системи переважно орієнтовані на управління підприємством у цілому та не враховують специфіку проведення діагностичних процедур і формування технічних висновків.

Окремо слід відзначити універсальні програмні засоби, зокрема електронні таблиці та інші інструменти загального призначення, які нерідко використовуються для ведення обліку на невеликих підприємствах. Їх головною перевагою є простота впровадження та відсутність значних

фінансових витрат. Проте використання таких рішень пов'язане з необхідністю ручного введення й оновлення даних, що збільшує трудомісткість роботи та створює додаткові ризики виникнення помилок. Зі збільшенням кількості інформації ефективність подібного підходу поступово знижується, а забезпечення актуальності даних стає все складнішим.

Окремої уваги заслуговують архітектурні особливості більшості існуючих програмних продуктів. Значна частина таких систем реалізована за монолітним принципом, коли всі функціональні компоненти об'єднані в межах одного програмного застосунку. Незважаючи на відносну простоту початкового впровадження, подібний підхід суттєво ускладнює подальший розвиток системи. Будь-які зміни в одному модулі можуть впливати на роботу інших компонентів, що підвищує складність супроводу та модернізації програмного забезпечення.

Крім того, аналіз існуючих рішень свідчить про недостатній рівень розвитку засобів аналітичної підтримки. У багатьох системах відсутні сучасні інструменти візуалізації даних, механізми гнучкого налаштування звітності та можливості прогностичного аналізу. У результаті накопичені дані використовуються переважно для зберігання та обліку, тоді як їх потенціал для підтримки управлінських рішень залишається реалізованим лише частково.

Для коректної оцінки наявних програмних рішень доцільно застосовувати систему критеріїв, які найбільшою мірою відповідають специфіці автоматизації технічного огляду транспортних засобів. Під час аналізу враховуються такі характеристики, як функціональні можливості, можливості інтеграції, рівень аналітичної підтримки, масштабованість, безпека та зручність використання.

Проведений аналіз показує, що більшість наявних рішень орієнтовані на виконання окремих завдань. CRM-системи забезпечують ефективну взаємодію з клієнтами, проте не підтримують повний цикл обробки технічної інформації. ERP-платформи мають широкі можливості для управління

ресурсами підприємства, але є надто складними для використання у діяльності невеликих і середніх пунктів технічного контролю. Спеціалізовані програмні комплекси, своєю чергою, добре враховують окремі особливості предметної області, однак часто не забезпечують достатнього рівня аналітики та гнучкості подальшого розвитку.

Програмні рішення, орієнтовані на діяльність станцій технічного обслуговування дозволяють автоматизувати окремі виробничі процеси, однак їх можливості часто обмежуються базовим набором функцій. У більшості випадків такі системи не забезпечують достатнього рівня аналітичної підтримки та мають обмежені засоби інтеграції із сучасними інформаційними сервісами.

Важливим недоліком багатьох систем є обмежені можливості подальшого розширення функціоналу. Із зростанням кількості клієнтів, транспортних засобів та обсягу накопичених даних підвищуються вимоги до продуктивності інформаційної системи. Водночас значна частина наявних програмних продуктів створювалася без урахування подальшого розширення функціональних можливостей, що ускладнює їх модернізацію та адаптацію до нових умов експлуатації.

Окремої уваги потребує питання взаємодії програмного забезпечення із зовнішніми інформаційними платформами. Діяльність сучасного підприємства передбачає постійний обмін даними між різними інформаційними системами. Відсутність відкритих програмних інтерфейсів або обмежені можливості взаємодії з зовнішніми сервісами суттєво знижують ефективність використання єдиного інформаційного середовища.

Проведена робота дає підстави стверджувати, що існуючі програмні продукти не забезпечують комплексного вирішення завдань, характерних для сфери технічного огляду транспортних засобів. Жодне з розглянутих рішень повною мірою не поєднує функції обліку, управління процесами, аналітичної обробки інформації та підтримки прийняття рішень. Саме це обумовлює необхідність розроблення нового програмного продукту, орієнтованого на

централізоване управління даними, аналітичну підтримку та можливість подальшого масштабування.

Підсумовуючи результати аналізу, можна зробити висновок, що на сучасному етапі відсутні комплексні програмні рішення, які б повністю відповідали потребам підприємств, що здійснюють технічний огляд транспортних засобів. Виявлені недоліки та функціональні обмеження розглянутих систем підтверджують актуальність створення нової інформаційно-аналітичної системи. Така система повинна базуватися на сучасних архітектурних підходах, підтримувати інтеграцію із зовнішніми сервісами, забезпечувати гнучкість подальшого розвитку та надавати ефективні інструменти для аналітичної обробки накопичених даних.

1.3 Аналіз сучасних підходів та технологій розроблення інформаційних систем

У сучасних умовах проектування програмних систем вибір архітектурного підходу є одним із ключових рішень, що визначає подальший розвиток програмного продукту. Для інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів архітектура має забезпечувати ефективну роботу з великими обсягами даних, оперативну обробку результатів перевірок та можливість безпечного зберігання інформації. Не менш важливими є можливість розширення функціональності системи та її адаптація до майбутніх змін.

Упродовж багатьох років під час створення програмних продуктів переважно використовувалася монолітна модель організації системи. У такій моделі всі функціональні компоненти застосунку, включаючи користувацький інтерфейс, бізнес-логіку та механізми доступу до даних, функціонують у складі одного програмного середовища. На початкових

етапах розробки цей підхід має певні переваги, зокрема простоту реалізації, централізоване управління кодом та відносно нескладне розгортання.

Разом із тим зі збільшенням функціональності програмного продукту недоліки монолітної архітектури стають дедалі помітнішими. Будь-яка зміна окремого модуля часто потребує повторного тестування та розгортання всієї системи. У великих проєктах це суттєво ускладнює подальший розвиток та підтримку системи, сповільнює процес впровадження нових функцій та підвищує ризик виникнення помилок під час оновлення.

Додаткові труднощі виникають під час масштабування. У монолітних системах збільшення навантаження на окремий функціональний компонент нерідко потребує масштабування всього застосунку. У результаті ресурси обчислювальної інфраструктури використовуються нераціонально та збільшення витрат на підтримку інфраструктури. Особливо помітними ці обмеження стають у системах, де різні модулі мають суттєво відмінне навантаження.

Альтернативою монолітному підходу стала мікросервісна архітектура. Її основна ідея полягає у поділі програмної системи на набір незалежних сервісів, кожен з яких відповідає за виконання конкретної функції предметної області. Наприклад, окремі сервіси можуть відповідати за управління користувачами, обробку інформації про транспортні засоби, проведення технічних оглядів, підготовку аналітичної звітності або надсилання повідомлень.

Однією з головних переваг такого підходу є можливість незалежного масштабування окремих компонентів. Якщо певний сервіс працює під підвищеним навантаженням, додаткові ресурси можуть бути виділені саме для нього без необхідності змінювати конфігурацію всієї системи.

Важливою перевагою є технологічна незалежність сервісів, що забезпечує гнучкість розвитку системи та можливість використання найбільш доцільних технологій для окремих компонентів. Не менш важливою є підвищена стійкість до відмов. У разі виникнення проблем у

роботі окремого сервісу інші компоненти системи продовжують функціонувати, що знижує ризик повної зупинки програмного комплексу. Для інформаційно-аналітичних систем це має особливе значення, оскільки дозволяє забезпечити безперервність виконання критично важливих бізнес-процесів.

Додатковою перевагою мікросервісного підходу є можливість незалежного розгортання окремих компонентів системи. Команди розробників можуть вносити зміни та оновлювати конкретні сервіси без необхідності повного перевстановлення програмного комплексу. Це дозволяє скоротити цикл впровадження нових функцій і швидше реагувати на зміни вимог користувачів та бізнес-процесів.

Для забезпечення ефективної взаємодії між окремими сервісами доцільно використовувати архітектурний стиль REST (Representational State Transfer). Взаємодія між сервісами організовується через загальноприйняті вебпротоколи HTTP і HTTPS, що забезпечує сумісність різних модулів незалежно від використовуваних технологій. Завдяки цьому до системи можуть підключатися вебзастосунки, мобільні клієнти та робочі місця інспекторів без необхідності внесення суттєвих змін до архітектури.

Центральну роль у такій архітектурі відіграє API Gateway. Через нього проходить взаємодія між клієнтськими застосунками та внутрішніми сервісами системи. API Gateway виконує маршрутизацію запитів, розподіл навантаження, перевірку доступу та інші службові функції. Завдяки цьому інтеграція окремих компонентів відбувається значно простіше та приховує внутрішню структуру системи від кінцевих користувачів.

Для реалізації серверної частини програмного комплексу обрано платформу .NET та фреймворк ASP.NET Core. Це рішення забезпечує високу продуктивність, підтримку мікросервісної архітектури та надає широкий набір інструментів для створення сучасних вебзастосунків. Використання ASP.NET Core дозволяє реалізувати надійну бізнес-логіку, підтримувати

механізми автентифікації та забезпечувати ефективну взаємодію між окремими сервісами системи.

Для зберігання та обробки інформації використовуються сучасні системи управління базами даних, що забезпечують цілісність даних, надійність зберігання та швидкий доступ до накопиченої інформації. Це особливо важливо для систем технічного огляду, де необхідно працювати з великим обсягом історичних записів та результатів перевірок.

Окрему увагу під час розроблення приділено питанням розгортання та ізоляції середовищ виконання. Для ізоляції середовища виконання обрано платформу Docker, яка дозволяє упаковувати кожен сервіс разом із необхідними залежностями в окремий контейнер. Такий підхід гарантує передбачувану роботу системи незалежно від середовища розгортання незалежно від середовища виконання, що спрощує тестування, впровадження та подальшу експлуатацію.

Для керування контейнерами можуть використовуватися засоби оркестрації, що автоматизують запуск і масштабування сервісів.

Важливою складовою сучасного процесу розроблення є впровадження підходів DevOps та використання механізмів безперервної інтеграції й доставки програмного забезпечення (CI/CD). Автоматизація процесів складання, тестування та розгортання дозволяє своєчасно виявляти помилки та скорочує час між внесенням змін до коду й їх впровадженням у робоче середовище.

Для забезпечення безпеки інформації використовуються сучасні механізми автентифікації та авторизації. Одним із найбільш поширених рішень є застосування JSON Web Token (JWT), що дозволяє організувати захищений доступ до ресурсів системи та забезпечити контроль прав користувачів під час роботи з даними технічного огляду.

Перспективним напрямом розвитку подібних систем є використання хмарних технологій. Хмарні платформи забезпечують гнучке управління обчислювальними ресурсами, високу доступність сервісів та можливість

резервування даних. Для інформаційно-аналітичних систем технічного огляду це створює додаткові можливості щодо підвищення продуктивності, оптимізації витрат на інфраструктуру та забезпечення стабільної роботи програмного забезпечення.

Таким чином, результати проведеного аналізу сучасних технологічних підходів свідчать про те, що поєднання мікросервісної архітектури, платформи .NET, засобів контейнеризації, REST-взаємодії та сучасних механізмів забезпечення безпеки є доцільним для створення інформаційно-аналітичної системи технічного огляду транспортних засобів. Такий підхід дозволяє отримати не лише інструмент для автоматизації операційної діяльності, а й основу для подальшого розвитку, масштабування та аналітичної обробки накопичених даних відповідно до сучасних вимог розробки програмного забезпечення.

1.4 Постановка задачі

Впровадження цифрових технологій у сферу технічного контролю транспортних засобів давно вийшло за межі простої заміни паперових документів електронними. Сучасні вимоги до безпеки дорожнього руху та якості надання послуг передбачають використання комплексних програмних рішень, здатних підтримувати контроль усіх етапів проведення технічного огляду. Ручна обробка інформації, залежність від людського фактора та використання розрізнених джерел даних суттєво знижують ефективність роботи підприємств і ускладнюють прийняття управлінських рішень. У зв'язку з цим створення сучасної інформаційно-аналітичної системи на основі мікросервісного підходу є актуальним та практично значущим завданням.

Об'єктом роботи є процес автоматизації технічного огляду транспортних засобів.

Предметом роботи є методи, моделі та програмні засоби автоматизації технічного контролю транспортних засобів, а також засоби обробки й аналізу відповідної інформації.

Метою роботи є розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів на основі мікросервісної архітектури.

Для досягнення поставленої мети необхідно виконати такі завдання:

- дослідити особливості предметної області та визначити ключові проблеми, що виникають під час організації технічного огляду транспортних засобів;
- провести аналіз сучасних засобів автоматизації відповідних процесів;
- обґрунтувати вибір мікросервісної архітектури та технологічного стеку на базі платформи .NET;
- розробити архітектуру інформаційно-аналітичної системи з виділенням окремих функціональних сервісів та визначенням механізмів їх взаємодії через API Gateway;
- реалізувати серверну частину системи, що забезпечує маршрутизацію запитів, обробку даних та інтеграцію між компонентами;
- створити користувацький інтерфейс для операторів, експертів та адміністративного персоналу;
- реалізувати механізми автоматизованого формування документації, зберігання результатів перевірок та візуалізації статистичних даних;
- виконати тестування системи та оцінити її працездатність в умовах, наближених до реальної експлуатації.

2 ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз функціональних вимог та моделювання предметної області

Процес технічного огляду транспортних засобів є складною багаторівневою процедурою, що поєднує організаційні, технічні та інформаційні аспекти діяльності підприємства. Ефективність роботи станції технічного огляду безпосередньо залежить від швидкості обробки даних, точності фіксації результатів перевірок та зручності взаємодії між персоналом і клієнтами. В умовах постійного зростання кількості транспортних засобів та підвищення вимог до безпеки дорожнього руху використання традиційних методів ведення обліку стає недостатньо ефективним. Саме тому виникає необхідність у впровадженні сучасної інформаційно-аналітичної системи, яка дозволить автоматизувати ключові етапи проведення технічного огляду та забезпечити централізоване управління всіма інформаційними потоками.

Під час проєктування інформаційної системи одним із найважливіших етапів є визначення функціональних вимог. Вони описують основні можливості системи, набір операцій, що мають виконуватися користувачами, а також логіку взаємодії між окремими компонентами програмного забезпечення. Аналіз функціональних вимог дозволяє сформувати цілісне уявлення про майбутню систему та забезпечити відповідність її функціоналу реальним потребам підприємства.

Предметною областю розроблюваної системи є автоматизація діяльності підприємства, що здійснює технічний огляд транспортних засобів. Основними об'єктами обробки інформації виступають клієнти, транспортні засоби, результати технічних перевірок, працівники підприємства та сформована аналітична звітність. Система повинна забезпечувати повний

цикл роботи з даними – від моменту реєстрації транспортного засобу до формування фінальних висновків та статистичних звітів.

У межах роботи було визначено основні категорії користувачів інформаційної системи, кожна з яких має власний набір функціональних можливостей та рівень доступу до інформації.

Адміністратор системи відповідає за управління користувачами, налаштування доступу та підтримку працездатності програмного середовища. До його функцій належить створення облікових записів працівників, контроль ролей користувачів, моніторинг роботи сервісів та управління системними параметрами. Адміністратор має найвищий рівень доступу до інформаційних ресурсів системи.

Оператор технічного огляду здійснює реєстрацію клієнтів та транспортних засобів, формує записи на проходження технічного огляду та виконує первинне внесення інформації до бази даних. Основною задачею оператора є забезпечення швидкої та коректної обробки заявок, а також підтримка актуальності інформації про клієнтів підприємства.

Експерт або інспектор технічного контролю безпосередньо виконує перевірку транспортного засобу та вносить результати діагностики до системи. Для цього передбачено механізми введення параметрів перевірки окремих вузлів автомобіля, фіксації несправностей та автоматичного формування висновку щодо придатності транспортного засобу до експлуатації.

Керівник підприємства використовує систему переважно для перегляду аналітичної інформації та контролю ефективності роботи станції технічного огляду. Система повинна забезпечувати можливість формування статистичних звітів, аналізу навантаження на персонал, оцінки кількості проведених перевірок та контролю фінансових показників діяльності підприємства.

Окремим учасником інформаційної взаємодії є клієнт, який отримує можливість запису на проходження технічного огляду, перегляду результатів

перевірки та отримання повідомлень щодо завершення терміну дії технічного контролю.

Основні функціональні можливості користувачів системи наведено у таблиці 2.1.

Таблиця 2.1 – Основні функціональні можливості користувачів системи

Категорія користувача	Основні функції
Адміністратор	Управління користувачами та налаштування системи
Оператор	Реєстрація клієнтів та транспортних засобів
Експерт	Внесення результатів технічного огляду
Керівник	Перегляд аналітики та формування звітності
Клієнт	Запис на технічний огляд та отримання повідомлень

На основі визначених ролей було сформовано основні функціональні вимоги до інформаційної системи. Однією з ключових вимог є централізоване зберігання інформації про клієнтів та транспортні засоби. Усі дані повинні зберігатися в єдиній структурованій базі даних із можливістю швидкого пошуку, оновлення та аналізу інформації.

Для забезпечення ефективного функціонування інформаційної системи всі вимоги до програмного забезпечення доцільно поділити на функціональні та нефункціональні. Функціональні вимоги визначають перелік операцій, які повинна виконувати система, тоді як нефункціональні вимоги характеризують якість реалізації цих функцій.

До функціональних вимог розроблюваної системи належать ведення обліку клієнтів, реєстрація транспортних засобів, створення заявок на проходження технічного огляду, фіксація результатів перевірок, автоматичне формування висновків, підготовка звітів і надсилання повідомлень

користувачам. Реалізація зазначених можливостей дає змогу спростити виконання щоденних операцій та зменшити навантаження на персонал підприємства.

Окрім функціональних можливостей, під час проєктування враховувалися вимоги до продуктивності, надійності, безпеки та подальшого розвитку системи. Програмне забезпечення повинно стабільно працювати за одночасної роботи кількох користувачів, підтримувати резервне копіювання даних і забезпечувати належний рівень захисту персональної інформації відповідно до сучасних вимог інформаційної безпеки.

Одним із ключових завдань системи є автоматизація процесу реєстрації транспортних засобів та створення заявок на проходження технічного огляду. На етапі введення даних передбачено перевірку їхньої коректності, що дозволяє знизити ймовірність помилок і мінімізувати вплив людського фактора під час роботи операторів.

Важливе місце займає механізм автоматичного формування результатів технічного огляду. Після завершення перевірки система створює електронний висновок, у якому відображаються результати діагностики, виявлені несправності та рішення щодо можливості подальшої експлуатації транспортного засобу.

Окремий функціональний блок пов'язаний з аналітичною обробкою накопичених даних. Для підтримки управлінських рішень передбачено автоматичне формування звітів і візуалізацію статистичної інформації. Аналітичний модуль надає відомості про кількість проведених технічних оглядів, розподіл транспортних засобів за категоріями, рівень завантаження працівників, фінансові показники та динаміку діяльності підприємства за обраний період.

Поряд із функціональними можливостями важливу роль відіграють і нефункціональні характеристики програмного забезпечення. До них належать надійність роботи, швидкодія, можливість масштабування та захист інформації. Оскільки система працює з персональними даними клієнтів і

результатами технічних перевірок, особливу увагу приділено механізмам контролю доступу та захисту інформаційних ресурсів.

Для наочного відображення взаємодії користувачів із програмним забезпеченням доцільно застосовувати UML-моделювання. Діаграма варіантів використання дозволяє показати основні сценарії роботи користувачів та визначити перелік функцій, які реалізуються в системі. Узагальнену UML-діаграму варіантів використання інформаційної системи наведено на рисунку 2.1.

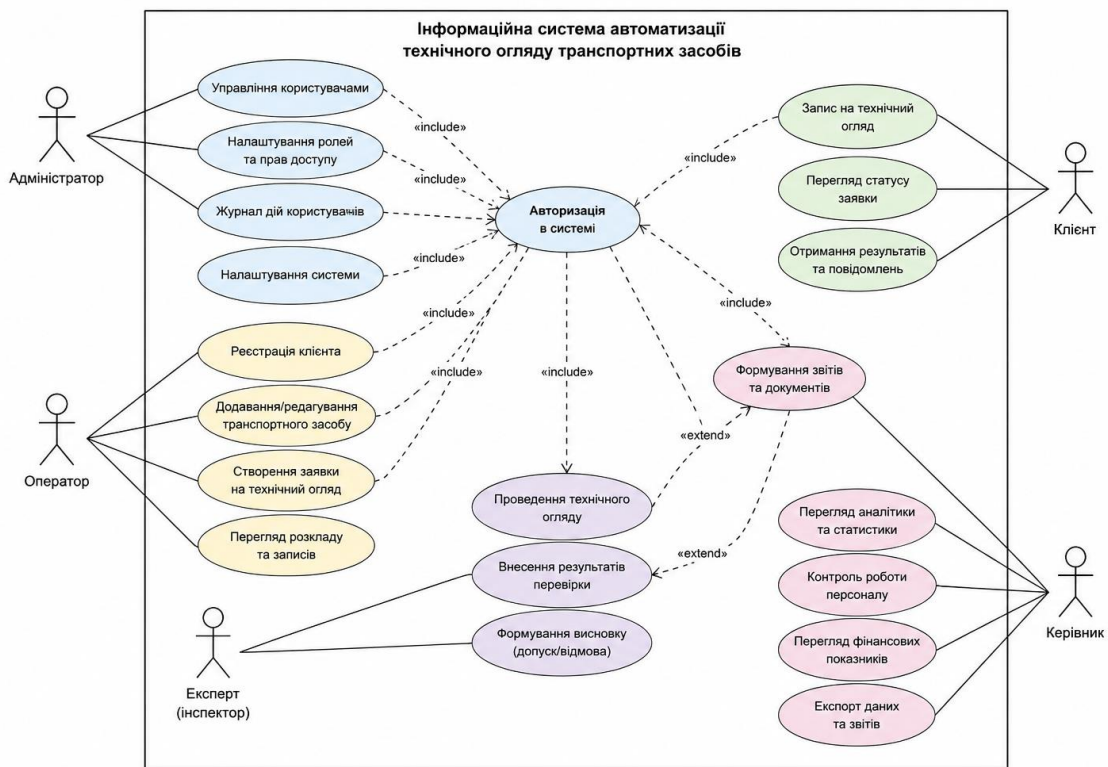


Рисунок 2.1 – UML-діаграма варіантів використання інформаційної системи

Для моделювання внутрішньої логіки роботи системи також використовується діаграма активності, яка дозволяє відобразити послідовність виконання основних операцій під час проходження технічного огляду. Такий підхід забезпечує формалізацію бізнес-процесів та дозволяє виявити критичні точки взаємодії між окремими компонентами системи.

Загальний алгоритм проходження технічного огляду включає декілька послідовних етапів. Спочатку оператор здійснює реєстрацію клієнта та транспортного засобу. Після цього формується заявка на проходження технічного огляду та визначається дата проведення перевірки.

Наступним етапом є проведення технічної діагностики, під час якої експерт фіксує результати перевірки основних вузлів транспортного засобу. Отримані результати зберігаються в базі даних, після чого система автоматично формує підсумковий висновок та генерує необхідну звітну документацію.

Схему алгоритму проходження технічного огляду представлено на рисунку А.1.

Таким чином, проведений аналіз функціональних вимог та моделювання предметної області дозволив визначити основні бізнес-процеси, ролі користувачів та ключові функціональні можливості майбутньої інформаційно-аналітичної системи. Отримані результати формують основу для подальшого проектування архітектури програмного забезпечення, структури мікросервісів та бази даних системи.

2.2 Проектування архітектури інформаційної системи

Архітектура програмного забезпечення є одним із ключових факторів, що визначає ефективність, масштабованість та надійність інформаційної системи. Для систем автоматизації технічного огляду транспортних засобів особливо важливими є швидкість обробки інформації, стабільність роботи при великій кількості запитів та можливість подальшого функціонального розширення без повного перепроєктування програмного комплексу. Саме тому вибір архітектурного підходу є стратегічним етапом розроблення інформаційної системи.

Тривалий час більшість корпоративних програмних рішень будувались за принципом монолітної архітектури. У такій моделі всі компоненти системи – користувацький інтерфейс, серверна логіка, модулі обробки даних та механізми взаємодії з базою даних – функціонують як єдиний програмний застосунок. Монолітний підхід має певні переваги на початкових етапах розроблення, зокрема відносну простоту створення, централізоване управління кодовою базою та спрощене розгортання програмного забезпечення.

Проте в умовах сучасних високонавантажених систем монолітна архітектура демонструє низку критичних недоліків. Насамперед проблема полягає у високому рівні зв'язності між компонентами системи. Будь-яка зміна в одному функціональному модулі може впливати на працездатність усієї системи, що значно ускладнює процес супроводу та модернізації програмного забезпечення. Крім того, монолітні застосунки погано масштабуються, оскільки при збільшенні навантаження необхідно дублювати весь програмний комплекс, навіть якщо перевантаження виникає лише в окремому модулі.

Ще одним суттєвим недоліком монолітної архітектури є складність колективної розробки. У великих програмних проєктах над однією кодовою базою одночасно працює значна кількість розробників, що нерідко призводить до конфліктів під час інтеграції змін, ускладнює тестування та сповільнює впровадження нового функціоналу. Для інформаційно-аналітичної системи технічного огляду транспортних засобів такий підхід є недостатньо гнучким, оскільки програмний продукт повинен постійно розвиватися та доповнюватися новими можливостями без порушення роботи основних сервісів.

Одним із сучасних підходів до побудови програмного забезпечення є мікросервісна архітектура. Її особливість полягає в поділі системи на окремі незалежні компоненти, кожен з яких реалізує власний набір функцій та відповідає за конкретну бізнес-область. Завдяки цьому зменшується

залежність між окремими частинами системи, а розгортання та супровід сервісів можуть виконуватися незалежно один від одного.

Важливою особливістю такого підходу є можливість вибіркового масштабування. У разі збільшення навантаження додаткові ресурси можуть виділятися лише для тих компонентів, які цього потребують. Наприклад, якщо зростає кількість запитів до сервісу авторизації або модуля формування звітності, немає необхідності збільшувати ресурси для всієї системи. Це дозволяє ефективніше використовувати обчислювальні потужності та оптимізувати витрати на інфраструктуру.

Не менш важливою перевагою є підвищення стійкості системи до збоїв. Вихід з ладу окремого сервісу не обов'язково призводить до припинення роботи інших компонентів. Наприклад, навіть у випадку тимчасової недоступності аналітичного модуля сервіси реєстрації транспортних засобів і проведення технічного огляду можуть продовжувати функціонувати без критичних обмежень.

Окремої уваги заслуговує можливість незалежного оновлення сервісів. Кожен мікросервіс має власний цикл розроблення та може модернізуватися окремо від інших компонентів. Такий підхід скорочує час впровадження нових функцій і знижує ризик виникнення помилок під час оновлення програмного забезпечення.

Для реалізації взаємодії між окремими мікросервісами у роботі використовується архітектурний стиль REST (Representational State Transfer). REST API дозволяє організувати обмін даними між сервісами за допомогою стандартних HTTP-запитів. Основними HTTP-методами, що використовуються в системі, є GET, POST, PUT та DELETE.

Метод GET застосовується для отримання інформації із сервісів, POST – для створення нових записів, PUT – для оновлення даних, а DELETE – для видалення інформації з бази даних. Передача інформації між сервісами здійснюється у форматі JSON, який є універсальним та легко інтегрується з різними програмними платформами.

Для забезпечення централізованої маршрутизації запитів у системі використовується API Gateway. API Gateway виступає єдиною точкою входу для всіх клієнтських запитів та приховує внутрішню структуру мікросервісної системи від кінцевого користувача. Основними функціями API Gateway є:

- маршрутизація HTTP-запитів;
- балансування навантаження;
- централізована авторизація;
- перевірка токенів доступу;
- обмеження доступу до внутрішніх сервісів;
- логування запитів.

Використання API Gateway значно спрощує взаємодію клієнтської частини із серверною інфраструктурою та підвищує рівень безпеки інформаційної системи.

Загальну архітектуру мікросервісної системи представлено на рисунку А.2.

У межах розроблюваної системи було виділено декілька основних мікросервісів, кожен із яких реалізує окремий набір функцій.

Сервіс AuthService відповідає за автентифікацію та авторизацію користувачів системи. Основними функціями сервісу є реєстрація користувачів, перевірка облікових даних, генерація JWT-токенів та контроль прав доступу до функціональних модулів системи.

Сервіс UserService використовується для роботи з інформацією про користувачів системи. Він забезпечує створення, редагування та зберігання даних про працівників підприємства, а також підтримує механізми керування ролями користувачів.

Сервіс VehicleService відповідає за обробку інформації про транспортні засоби. До його функцій належать реєстрація автомобілів, зберігання технічних характеристик, пошук транспортних засобів та ведення історії проходження технічного огляду.

Сервіс `InspectionService` реалізує основну бізнес-логіку системи технічного огляду. Саме цей сервіс забезпечує створення заявок на проходження технічного контролю, внесення результатів перевірок, формування висновків та збереження результатів діагностики.

Сервіс `AnalyticsService` відповідає за формування статистичної інформації та аналітичної звітності. Він забезпечує обробку накопичених даних, формування графіків, аналітичних показників та підтримку управлінських рішень.

Сервіс `NotificationService` використовується для автоматичного надсилання повідомлень клієнтам та працівникам системи. Основними функціями є нагадування про завершення терміну дії технічного огляду, інформування про результати перевірки та надсилання службових повідомлень.

Взаємодія між сервісами здійснюється через HTTP/HTTPS-протоколи із використанням REST API. Кожен сервіс має власний набір endpoint-ів, через які виконується обмін інформацією. Для забезпечення безпечної передачі даних використовується протокол HTTPS, що дозволяє шифрувати мережевий трафік та захищати конфіденційну інформацію від несанкціонованого доступу.

Для авторизації користувачів у системі використовується технологія JWT (JSON Web Token). Після успішної автентифікації користувач отримує спеціальний токен доступу, який надалі передається у заголовках HTTP-запитів. API Gateway перевіряє валідність токена та визначає права доступу користувача до конкретного сервісу.

Використання JWT дозволяє реалізувати безстатусну модель авторизації, що особливо важливо для мікросервісних систем із великою кількістю незалежних компонентів. Такий підхід знижує навантаження на сервер авторизації та підвищує швидкість обробки запитів.

Для забезпечення ізоляції середовищ виконання та спрощення розгортання системи використовується технологія контейнеризації Docker.

Кожен мікросервіс розгортається у власному контейнері разом із необхідними бібліотеками та залежностями. Це забезпечує стабільність роботи системи незалежно від конфігурації операційного середовища.

Контейнеризація також значно спрощує масштабування системи та автоматизацію процесів розгортання. За необхідності можна швидко створювати нові екземпляри сервісів для балансування навантаження або перенесення системи на інші сервери.

Схему взаємодії мікросервісів через API Gateway представлено на рисунку А.3.

Для моделювання послідовності обробки запитів у системі доцільно використовувати UML Sequence Diagram. Такий тип діаграми дозволяє відобразити послідовність взаємодії між клієнтським застосунком, API Gateway та окремими мікросервісами під час виконання певної операції, наприклад авторизації користувача або створення заявки на технічний огляд.

Таким чином, використання мікросервісної архітектури, REST API, API Gateway, JWT та Docker забезпечує створення гнучкої, масштабованої та надійної інформаційної системи автоматизації технічного огляду транспортних засобів. Обраний архітектурний підхід дозволяє ефективно розподілити функціональність між незалежними сервісами, забезпечити високу продуктивність системи та створити основу для подальшого розвитку програмного комплексу.

Важливим етапом проєктування інформаційної системи є вибір технологічного стеку, який забезпечуватиме реалізацію функціональних вимог та відповідатиме сучасним вимогам до продуктивності, надійності й масштабованості програмного забезпечення. Для реалізації серверної частини системи було обрано платформу ASP.NET Core, яка є однією з найбільш поширених технологій для розроблення вебзастосунків і мікросервісних систем.

Перевагами ASP.NET Core є висока продуктивність, підтримка кросплатформеності, інтеграція із сучасними засобами розробки та

можливість створення REST API. Завдяки використанню вбудованого механізму залежностей (Dependency Injection) спрощується підтримка програмного коду та забезпечується його модульність.

Для взаємодії з базою даних використовується технологія Entity Framework Core, яка реалізує концепцію об'єктно-реляційного відображення. Використання ORM дозволяє працювати з базою даних через об'єкти предметної області, що значно спрощує процес розроблення та супроводу програмного забезпечення. Крім того, Entity Framework Core підтримує механізм міграцій, який забезпечує контроль змін структури бази даних протягом життєвого циклу системи.

Як система управління базами даних використовується Microsoft SQL Server. Дана СУБД забезпечує високий рівень продуктивності, підтримку складних запитів, механізми резервного копіювання та засоби захисту інформації. Використання SQL Server є доцільним для інформаційно-аналітичних систем, які працюють із великим обсягом структурованих даних.

Для реалізації клієнтської частини системи використовується вебінтерфейс, що забезпечує доступ до функціональних можливостей системи через браузер без необхідності встановлення додаткового програмного забезпечення. Такий підхід спрощує експлуатацію системи та забезпечує доступ до неї з різних пристроїв.

2.3 Проєктування структури бази даних та інформаційних потоків

Одним із ключових етапів розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів є проєктування структури бази даних. Саме база даних забезпечує централізоване зберігання інформації, підтримку цілісності даних, швидкий доступ до інформаційних ресурсів та можливість подальшого аналітичного опрацювання накопичених даних.

У процесі проектування системи особливу увагу було приділено забезпеченню структурованого зберігання інформації про користувачів, клієнтів, транспортні засоби, результати технічного огляду та сформовану звітність. Для реалізації системи було обрано реляційну модель бази даних, оскільки саме такий підхід забезпечує високу надійність зберігання інформації, підтримку зв'язків між сутностями та можливість виконання складних аналітичних запитів.

Використання реляційної бази даних дозволяє реалізувати механізми забезпечення цілісності інформації, уникнути дублювання даних та забезпечити централізоване керування всіма інформаційними ресурсами системи. Крім того, реляційна модель добре інтегрується з сучасними серверними платформами та підтримує ефективну роботу в умовах великого обсягу транзакцій.

Основу структури бази даних складає набір взаємопов'язаних сутностей, кожна з яких відповідає за зберігання певного типу інформації. У межах розроблюваної системи було виділено декілька ключових таблиць, необхідних для забезпечення повного циклу роботи інформаційної системи технічного огляду.

Таблиця Users призначена для зберігання інформації про користувачів системи. У ній містяться дані про логіни, паролі, ролі користувачів, контактну інформацію та параметри доступу до функціональних модулів системи. Основним призначенням цієї таблиці є реалізація механізмів автентифікації та авторизації користувачів.

Таблиця Clients використовується для зберігання інформації про клієнтів підприємства. До її структури входять персональні дані власників транспортних засобів, контактні номери телефонів, адреси електронної пошти та інша службова інформація, необхідна для організації технічного огляду.

Таблиця Vehicles містить інформацію про транспортні засоби, що проходять технічний огляд. У ній зберігаються державний номер автомобіля,

VIN-код, марка, модель, рік випуску, категорія транспортного засобу та інші технічні характеристики.

Таблиця Inspections використовується для збереження інформації про проведення технічного огляду. Вона містить дату перевірки, статус проходження огляду, відповідального інспектора та загальний висновок щодо придатності транспортного засобу до експлуатації.

Таблиця InspectionResults забезпечує збереження результатів перевірки окремих вузлів та параметрів транспортного засобу. Саме ця таблиця містить інформацію про стан гальмівної системи, освітлення, рульового керування, екологічних показників та інших компонентів автомобіля.

Таблиця Reports використовується для збереження сформованої аналітичної та статистичної звітності. У ній можуть міститися як службові звіти для керівництва підприємства, так і автоматично сформовані документи технічного огляду.

Таблиця Notifications призначена для роботи із системою повідомлень та нагадувань. Вона містить інформацію про надіслані повідомлення, їх тип, дату створення та статус доставки користувачам системи.

Основні сутності бази даних та їх призначення наведено у таблиці 2.2.

Під час проектування структури бази даних особливу увагу було приділено організації зв'язків між сутностями. Використання зв'язків дозволяє уникнути дублювання інформації та забезпечити логічну цілісність даних.

Між таблицями Clients та Vehicles реалізовано зв'язок типу «один до багатьох», оскільки один клієнт може володіти декількома транспортними засобами. Аналогічний тип зв'язку використовується між таблицями Vehicles та Inspections, оскільки кожен транспортний засіб може проходити технічний огляд багаторазово.

Таблиця 2.2 – Основні сутності бази даних

Сутність	Призначення
Users	Зберігання даних користувачів системи
Clients	Інформація про клієнтів підприємства
Vehicles	Дані про транспортні засоби
Inspections	Дані про проведення технічного огляду
InspectionResults	Результати перевірки параметрів транспортного засобу
Reports	Формування та зберігання звітності
Notifications	Надсилання повідомлень та нагадувань

Таблиця InspectionResults має зв'язок із таблицею Inspections, що дозволяє зберігати деталізовані результати перевірки для кожного окремого технічного огляду. Таблиця Notifications пов'язується із клієнтами та користувачами системи для реалізації механізмів інформування та автоматичних нагадувань.

Для моделювання структури бази даних використовується ER-модель (Entity Relationship Model), яка дозволяє графічно відобразити сутності системи та зв'язки між ними. ER-модель забезпечує наочне представлення структури інформаційної системи та використовується як основа для подальшої реалізації фізичної бази даних.

ER-модель бази даних інформаційної системи представлено на рисунку А.4.

Під час проєктування структури бази даних важливим завданням є забезпечення відсутності надлишковості інформації та підтримка логічної цілісності даних. Для досягнення цієї мети застосовуються принципи нормалізації баз даних.

Нормалізація являє собою процес організації структури таблиць таким чином, щоб мінімізувати дублювання інформації та усунути можливі аномалії під час вставки, оновлення або видалення даних. У межах

розроблюваної системи структура бази даних проектувалася з урахуванням вимог третьої нормальної форми.

Відповідність першій нормальній формі досягається завдяки тому, що кожне поле таблиці містить лише одне значення. Повторювані групи даних у структурі таблиць відсутні, що забезпечує впорядковане зберігання інформації. Наприклад, відомості про транспортний засіб подаються окремими атрибутами, серед яких державний номер, VIN-код, марка та модель. Такий підхід спрощує подальшу обробку інформації та виключає неоднозначність під час роботи з даними.

Виконання вимог другої нормальної форми передбачає усунення часткових залежностей між атрибутами. Це дозволяє уникнути надлишкового дублювання даних та підтримувати їх узгодженість під час оновлення записів. У результаті зменшується дублювання інформації, а операції оновлення виконуються коректніше без ризику появи суперечливих записів.

Для досягнення третьої нормальної форми було виключено транзитивні залежності між атрибутами. У результаті всі неключові поля залежать безпосередньо від первинного ключа відповідної таблиці. Це означає, що кожен неключовий атрибут залежить лише від первинного ключа відповідної таблиці. Така організація структури даних позитивно впливає на логічну цілісність бази даних та спрощує її подальший супровід.

Використання нормалізованої структури дало змогу зменшити надлишковість збережених даних і забезпечити узгодженість інформації між усіма функціональними компонентами системи. Крім цього, така структура полегшує внесення змін до програмного забезпечення та підвищує загальну надійність його роботи.

Для підвищення швидкодії бази даних передбачено використання індексів для полів, які найчастіше використовуються під час пошуку записів. До таких полів належать VIN-код транспортного засобу, державний номер, ідентифікатор клієнта та дата проведення технічного огляду. Завдяки

індексуванню вдається значно скоротити час виконання запитів навіть за умови постійного збільшення обсягу накопичених даних.

Окрему увагу під час проєктування приділено аналізу маршрутів передавання даних між складовими інформаційної системи. Таке моделювання дозволяє чітко визначити, яким чином дані передаються між користувачами, сервісами, базами даних та аналітичними модулями.

Основними джерелами інформації є оператори, інспектори та адміністратори, які працюють із системою через вебінтерфейс. Після введення необхідних відомостей запити надходять до API Gateway, що відповідає за їх подальше маршрутизування до відповідних мікросервісів. Кожен сервіс виконує власні функції та працює лише з даними, що належать до його предметної області.

Так, під час реєстрації транспортного засобу інформація обробляється сервісом VehicleService, де виконується її перевірка та збереження. Дані, отримані під час проведення технічного огляду, надходять до InspectionService і фіксуються в таблицях Inspections та InspectionResults.

Після накопичення необхідного обсягу інформації аналітичний модуль AnalyticsService формує статистичні показники та узагальнені звіти для керівництва підприємства. Для інформування клієнтів про наближення строків проходження чергового технічного огляду використовується сервіс NotificationService.

Однією з важливих переваг реалізованого підходу є централізоване зберігання даних. Усі зміни, внесені користувачами, одразу відображаються в базі даних, що забезпечує актуальність інформації та знижує ризик її втрати.

Схему руху інформації між основними компонентами програмного забезпечення наведено на рисунку А.5.

Під час проєктування структури бази даних окрема увага приділялася питанням безпеки та продуктивності. Для захисту інформаційних ресурсів використовуються механізми автентифікації та авторизації користувачів, резервне копіювання даних і ведення журналів подій. Додатково

застосовується індексування найбільш важливих полів, що дозволяє зберігати високу швидкодію навіть за значного збільшення обсягу накопиченої інформації.

Таким чином, спроектована структура бази даних забезпечує централізоване зберігання інформації, підтримку цілісності даних та ефективну взаємодію між компонентами інформаційної системи. Реалізована модель інформаційних потоків дозволяє організувати безперервний процес обробки даних та створює основу для подальшого функціонування аналітичної підсистеми.

2.4 Моделювання аналітичної підсистеми та формування звітності

Для ефективного управління діяльністю підприємства недостатньо лише накопичувати дані. Важливо забезпечити їх обробку, узагальнення та представлення у зручній для аналізу формі. Саме ці завдання виконує аналітична підсистема, що є невід'ємною складовою інформаційної системи технічного огляду транспортних засобів. У сфері технічного огляду транспортних засобів аналітична складова має особливе значення, оскільки дозволяє контролювати ефективність роботи підприємства, оцінювати навантаження на персонал, аналізувати динаміку проходження технічних оглядів та формувати прогностичні показники діяльності.

У межах розробленого програмного рішення аналітичні функції винесено в окремий модуль. Він отримує дані від інших сервісів системи, накопичує їх та виконує подальшу обробку з метою формування звітів і статистичних показників. Основним призначенням підсистеми є збір, агрегування, аналіз та візуалізація інформації, що надходить із різних сервісів системи. Аналітичний модуль взаємодіє з базою даних, отримує актуальні показники діяльності підприємства та формує звітність у режимі реального часу.

Значна частина аналітичної інформації пов'язана з результатами проведення технічних оглядів. На основі накопичених даних система формує статистику щодо кількості перевірок, частки позитивних результатів та випадків відмови у допуску транспортних засобів до експлуатації. Система забезпечує автоматичний збір інформації про кількість проведених перевірок, кількість успішно пройдених технічних оглядів, відсоток відмов у допуску транспортних засобів до експлуатації та динаміку змін цих показників у часовому інтервалі.

Формування статистики виконується на основі накопичених даних із таблиць `Inspections` та `InspectionResults`. Після завершення кожного технічного огляду відповідна інформація автоматично передається до аналітичного сервісу, де виконується її подальше агрегування та підготовка до відображення у звітах і графіках.

Важливим напрямом роботи системи є аналітика по клієнтах. Такий тип аналізу дозволяє оцінювати активність клієнтської бази підприємства, визначати кількість постійних клієнтів, аналізувати частоту проходження технічного огляду та контролювати своєчасність повторних звернень. Крім того, система дозволяє визначати найбільш активні категорії клієнтів та формувати статистику за часовими періодами.

Для більш детального аналізу накопичених даних передбачено групування транспортних засобів за визначеними характеристиками. Зокрема, можуть використовуватися відомості про категорію транспортного засобу, марку, тип двигуна або рік випуску. Це дає можливість отримати узагальнену картину структури транспортного потоку та виявити найбільш поширені категорії автомобілів. Система дозволяє виконувати класифікацію автомобілів за типом транспорту, маркою, роком випуску, типом двигуна або категорією транспортного засобу. Такий підхід дозволяє керівництву підприємства оцінювати структуру потоку транспортних засобів та виявляти найбільш поширені категорії автомобілів, що проходять технічний огляд.

Особливого значення набуває аналіз фінансових показників діяльності підприємства. Інформаційна система дозволяє формувати статистику щодо прибутку від проведення технічних оглядів, аналізувати динаміку доходів за певний період часу та визначати середнє навантаження на станцію технічного огляду. На основі накопичених даних можуть формуватися фінансові звіти для оцінки ефективності діяльності підприємства.

Для підтримки управлінських рішень у системі реалізовано механізми розрахунку KPI (Key Performance Indicators) – ключових показників ефективності. KPI дозволяють оцінювати результативність роботи підприємства та ефективність використання ресурсів.

До основних KPI, які можуть розраховуватись у межах системи, належать:

- кількість проведених технічних оглядів за період;
- середній час обробки заявки;
- кількість відмов у проходженні ТО;
- навантаження на одного інспектора;
- середній прибуток за день або місяць;
- кількість повторних звернень клієнтів;
- відсоток успішного проходження технічного контролю.

Використання KPI дозволяє керівництву підприємства оперативно оцінювати поточний стан бізнес-процесів та своєчасно реагувати на негативні тенденції. У разі зниження кількості проведених технічних оглядів або збільшення кількості відмов система може автоматично сигналізувати про необхідність проведення додаткового аналізу.

Аналітична підсистема створена для забезпечення керівництва підприємства актуальною інформацією, необхідною для прийняття обґрунтованих управлінських рішень. Ефективність організації роботи значною мірою залежить від своєчасного отримання достовірних даних про перебіг основних бізнес-процесів. Використання автоматизованих засобів

аналізу дозволяє керівництву підприємства переходити від інтуїтивного прийняття рішень до прийняття рішень на основі об'єктивних даних.

Використання інструментів аналітики дає змогу оцінювати завантаженість персоналу, визначати найбільш затребувані послуги та аналізувати причини виникнення проблем під час проходження технічного огляду. На основі таких даних можуть прийматися рішення щодо оптимізації робочих процесів, перерозподілу ресурсів або впровадження додаткових послуг.

Значну практичну цінність має накопичення та подальше використання історичних даних, отриманих у процесі експлуатації системи. Накопичення статистичної інформації дозволяє виявляти закономірності та тенденції у роботі підприємства. Наприклад, система може демонструвати сезонні коливання кількості клієнтів, зміни структури транспортних засобів, що проходять технічний огляд, а також періоди найбільшого навантаження на персонал.

Отримані результати аналізу можуть використовуватися для планування робочого часу працівників, визначення потреби у додатковому обладнанні або формування графіків роботи інспекторів. Це дозволяє більш ефективно використовувати ресурси підприємства та підвищувати якість обслуговування клієнтів.

Додатково результати аналітичної обробки можуть застосовуватися для оцінювання досягнення стратегічних показників діяльності підприємства. Порівняння фактичних показників із запланованими значеннями дозволяє оцінювати ефективність реалізації управлінських рішень та своєчасно виявляти відхилення від встановлених планів. Такий підхід сприяє підвищенню прозорості управління та забезпечує інформаційну підтримку діяльності керівництва.

Перспективним напрямом розвитку аналітичної підсистеми є впровадження механізмів прогнозування на основі історичних даних. Такі механізми можуть використовуватися для оцінки майбутнього навантаження

на персонал, прогнозування кількості клієнтів та планування ресурсів підприємства. Однією з важливих функцій аналітичної підсистеми є автоматичне формування звітності. Система повинна забезпечувати генерацію звітів без необхідності ручного опрацювання даних оператором або адміністратором. Формування звітів може виконуватись як за запитом користувача, так і автоматично за визначеним часовим інтервалом.

Серед основних типів звітів, які формуються системою, можна виділити:

- статистичні звіти щодо проведення технічних оглядів;
- фінансові звіти;
- звіти про навантаження персоналу;
- звіти щодо категорій транспортних засобів;
- звіти про результати перевірок;
- звіти щодо активності клієнтів.

Підготовлені звіти можуть зберігатися у кількох форматах, що спрощує їх подальше використання, передачу та друк.

Для наочного представлення результатів аналізу застосовуються засоби графічної візуалізації даних. Візуальне представлення статистики дозволяє швидко оцінювати стан роботи підприємства та виявляти тенденції зміни показників.

Значна частина аналітичної інформації відображається за допомогою інтерактивних інформаційних панелей, на яких зібрано основні показники діяльності підприємства. Інформаційна панель забезпечує централізоване представлення інформації для керівництва підприємства та дозволяє здійснювати моніторинг основних показників у режимі реального часу.

Приклад аналітичної панелі системи представлено на рисунку А.6.

Для візуалізації статистичної інформації можуть використовуватись:

- стовпчикові діаграми;
- кругові діаграми;
- лінійні графіки;

- гістограми;
- таблиці показників;
- KPI-віджети.

Наприклад, стовпчикові діаграми можуть використовуватись для порівняння кількості технічних оглядів за різні періоди часу, кругові діаграми – для відображення розподілу транспортних засобів за категоріями, а лінійні графіки – для аналізу динаміки фінансових показників підприємства.

Важливою особливістю системи є автоматичне створення статистичних звітів на основі накопичених даних. Після накопичення інформації аналітичний сервіс виконує автоматичне агрегування даних та формує підсумкові звіти, які можуть бути використані керівництвом для оцінки ефективності роботи підприємства.

Схему формування статистичної звітності представлено на рисунку А.7.

Подальший розвиток аналітичної підсистеми пов'язаний із впровадженням прогностичних механізмів, що базуються на накопичених статистичних даних. Це дозволить оцінювати очікуване навантаження на персонал, прогнозувати кількість клієнтів та планувати використання ресурсів.

Таким чином, реалізована підсистема аналітики забезпечує не лише накопичення та візуалізацію інформації, а й створює інструментальну основу для підтримки управлінських рішень та підвищення ефективності роботи підприємства.

3 РОЗРОБЛЕННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ТЕХНІЧНОГО ОГЛЯДУ ТРАНСПОРТНИХ ЗАСОБІВ

3.1 Проєктування архітектури та структури системи

Одним із ключових етапів розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів є вибір технологічного стеку, який визначає продуктивність, надійність, масштабованість та можливості подальшого розвитку програмного продукту. При виборі інструментальних засобів враховувалися вимоги до реалізації мікросервісної архітектури, забезпечення безпеки даних, підтримки сучасних підходів до розроблення вебзастосунків та можливості інтеграції з іншими інформаційними системами.

Для реалізації серверної частини системи було обрано мову програмування C#. Дана мова є однією з найпоширеніших платформ для розроблення корпоративних інформаційних систем та має розвинену екосистему інструментів і бібліотек.

Основою серверної платформи обрано фреймворк ASP.NET Core. Для роботи з базою даних використано Entity Framework Core.

Як систему управління базами даних обрано Microsoft SQL Server.

Особливу увагу під час розроблення було приділено забезпеченню безпеки інформації. Для реалізації механізму автентифікації та авторизації користувачів використано технологію JSON Web Token (JWT). Даний підхід дозволяє реалізувати безпечний механізм передачі інформації про користувача між клієнтською та серверною частинами системи без необхідності зберігання сесій на сервері. Після успішного входу в систему користувач отримує токен доступу, який використовується для підтвердження його прав під час виконання подальших запитів до API. Такий

механізм добре підходить для розподілених мікросервісних систем та забезпечує високий рівень безпеки.

Для тестування та документування REST API використовується Swagger. Інструмент автоматично формує інтерактивну документацію для створених вебсервісів та дозволяє виконувати тестування запитів безпосередньо через вебінтерфейс. Це значно спрощує процес розроблення та налагодження сервісів, а також дозволяє швидко перевіряти коректність роботи окремих функціональних модулів системи.

З метою забезпечення незалежності програмного забезпечення від середовища виконання було використано технологію контейнеризації Docker.

Таким чином, обраний набір технологій повністю відповідає вимогам до реалізації сучасної інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів. Використання платформи .NET, мови програмування C#, технологій Entity Framework Core, SQL Server, JWT, Swagger та Docker дозволяє створити масштабоване, безпечне та зручне у супроводі програмне рішення, здатне забезпечити ефективну автоматизацію бізнес-процесів підприємства.

Одним із найважливіших етапів розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів стало проєктування її архітектури. Архітектурні рішення безпосередньо впливають на масштабованість, продуктивність, надійність та можливість подальшого розвитку програмного забезпечення. Саме тому на початковому етапі проєктування було проведено аналіз можливих підходів до побудови системи та обґрунтовано вибір мікросервісної архітектури.

Традиційним підходом до створення інформаційних систем є використання монолітної архітектури, у межах якої всі функціональні компоненти програмного продукту об'єднуються в єдиний застосунок. У такій системі модулі авторизації, роботи з користувачами, ведення журналів, формування звітності та інші підсистеми використовують спільний програмний код та єдину точку розгортання.

З огляду на це для реалізації програмного комплексу було обрано мікросервісну архітектуру. Її основна ідея полягає у поділі системи на набір незалежних сервісів, кожен з яких виконує окрему бізнес-функцію. Такий підхід дозволяє ізолювати окремі компоненти, спростити їх супровід та забезпечити можливість незалежного масштабування.

Використання мікросервісної архітектури надало ряд переваг для реалізації системи технічного огляду транспортних засобів:

- незалежна розробка та модернізація окремих функціональних модулів;
- спрощення супроводу програмного забезпечення;
- підвищення відмовостійкості системи;
- можливість незалежного масштабування окремих сервісів;
- спрощення процесу тестування та розгортання;
- підвищення рівня безпеки завдяки ізоляції компонентів;
- можливість подальшого розширення функціоналу без суттєвого впливу на вже реалізовані модулі.

Під час проєктування архітектури було виділено кілька основних сервісів, кожен із яких відповідає за реалізацію певного набору функцій.

Сервіс `AuthService.API` забезпечує автентифікацію та авторизацію користувачів системи. Основними його завданнями є реєстрація нових користувачів, перевірка облікових даних під час входу до системи та формування JWT-токенів доступу. Даний сервіс виступає центральним елементом механізму безпеки та контролю доступу.

Сервіс `UserService.API` призначений для роботи з обліковими записами користувачів. У його функціональні обов'язки входить управління профілями користувачів, призначення ролей, редагування контактної інформації та забезпечення контролю доступу відповідно до ролей користувачів.

Основним компонентом інформаційної системи є `VehicleService.API`. Саме цей сервіс реалізує ключові бізнес-процеси підприємства, пов'язані з проведенням технічного огляду транспортних засобів. У його межах

реалізовано ведення журналу технічних оглядів, облік транспортних засобів та їх власників, імпорту інформації з файлів Excel, роботу з документами та механізм онлайн-запису на проходження технічного огляду.

Для реалізації функцій аналітики та звітності було створено окремий сервіс ReportService.API. Його основним призначенням є формування статистичних даних, обчислення ключових показників ефективності, побудова аналітичних звітів та генерація PDF-документів. Виділення аналітичного функціоналу в окремий сервіс дозволяє уникнути додаткового навантаження на основні бізнес-модулі системи.

Для забезпечення централізованої взаємодії між клієнтською частиною та мікросервісами використовується ApiGateway. Даний компонент виступає єдиною точкою входу до системи та забезпечує маршрутизацію запитів до відповідних сервісів. Крім того, через API Gateway реалізуються механізми контролю доступу, перевірки токенів безпеки та централізованого логування запитів.

Взаємодія між сервісами організована за допомогою REST API через протокол HTTP. Для передачі інформації між сервісами використовується формат JSON, який забезпечує простоту інтеграції та мінімальні накладні витрати під час обміну даними. Автентифікація запитів здійснюється за допомогою JWT-токенів, що дозволяє підтримувати єдиний механізм безпеки для всієї системи.

Загальна структура взаємодії компонентів інформаційно-аналітичної системи наведена на рисунку 3.1.

Таким чином, використання мікросервісної архітектури дозволило створити гнучку, масштабовану та відмовостійку основу для реалізації інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів. Запропонований підхід забезпечує можливість подальшого розвитку програмного комплексу, спрощує його супровід та відповідає сучасним вимогам до побудови корпоративних інформаційних систем.

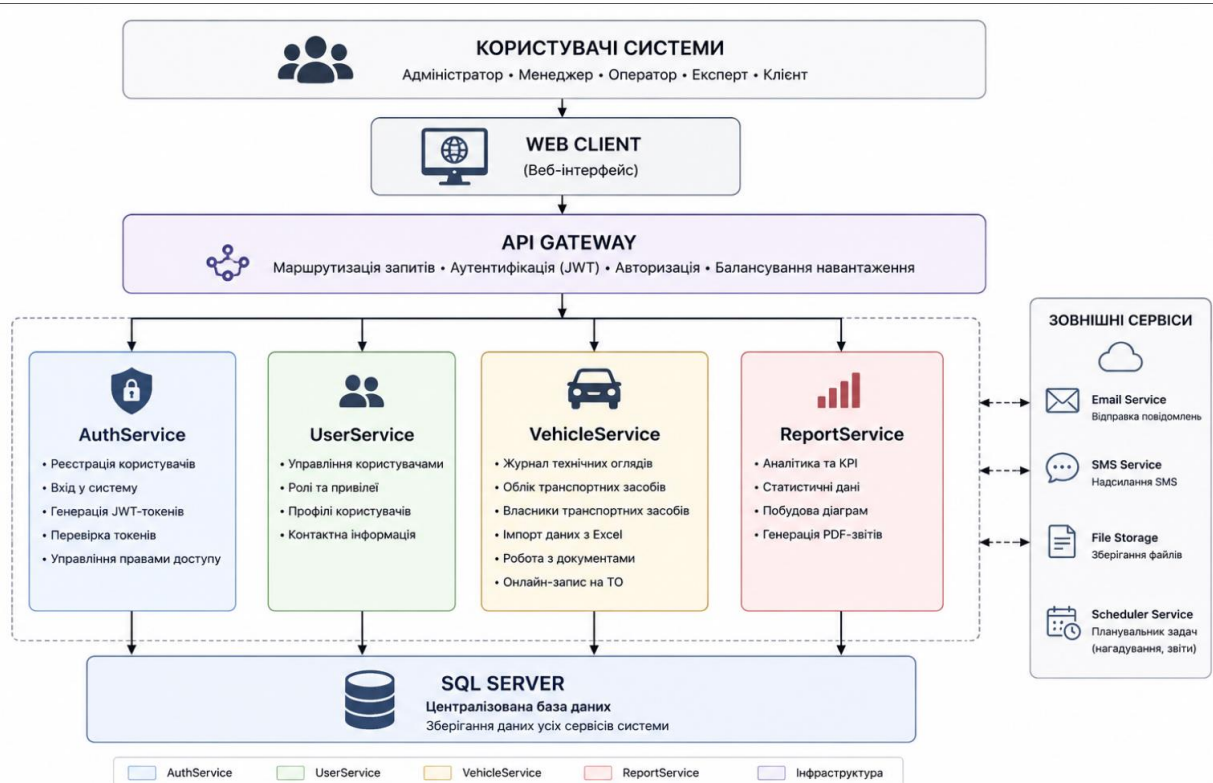


Рисунок 3.1 – Загальна архітектура інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів

Одним із ключових етапів розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів стало проєктування структури бази даних. Саме база даних забезпечує централізоване зберігання інформації, підтримку цілісності даних та взаємодію між окремими функціональними модулями системи. Від якості проєктування моделі даних безпосередньо залежить ефективність роботи програмного забезпечення, швидкість виконання запитів та можливість подальшого масштабування системи.

Для реалізації підсистеми зберігання даних було використано систему керування базами даних Microsoft SQL Server, яка забезпечує високий рівень надійності, підтримку транзакційної обробки даних та ефективну інтеграцію з платформою ASP.NET Core через Entity Framework Core. Використання реляційної моделі даних дозволило формалізувати зв'язки між об'єктами

предметної області та забезпечити контроль цілісності інформації за допомогою первинних і зовнішніх ключів.

У процесі аналізу предметної області було визначено основні інформаційні сутності, які беруть участь у роботі системи. До них належать користувачі системи, власники транспортних засобів, транспортні засоби, результати технічних оглядів, повідомлення про завершення терміну дії обов’язкового технічного контролю та електронні документи. На основі цих сутностей було сформовано логічну модель бази даних, яка представлена на рисунку 3.2.

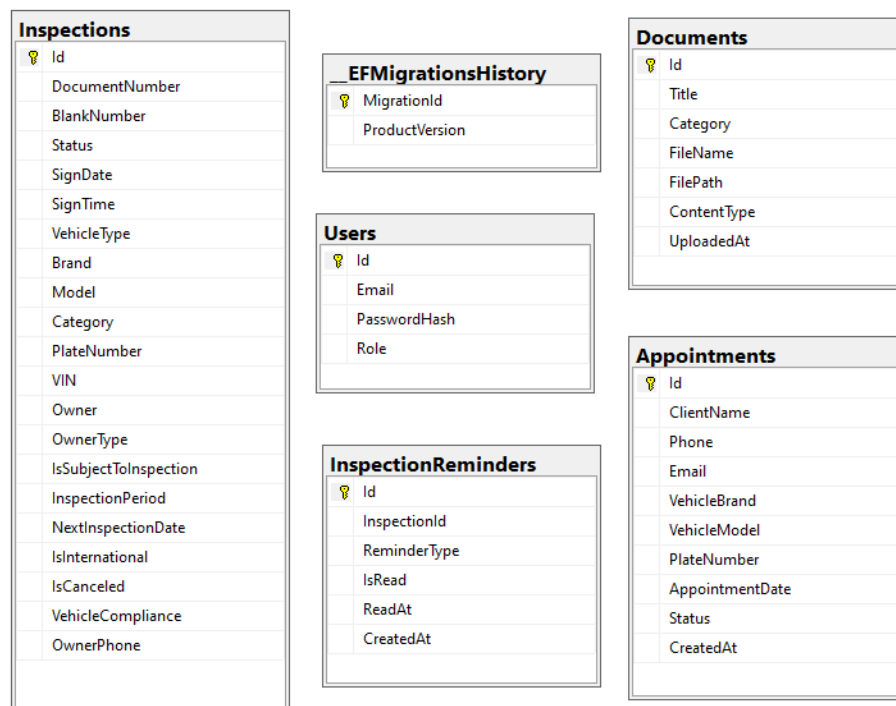


Рисунок 3.2 – ER-діаграма бази даних

Побудована структура передбачає зберігання інформації про користувачів системи незалежно від даних про транспортні засоби та результати перевірок. Такий підхід дозволяє реалізувати рольову модель доступу та забезпечити централізоване керування обліковими записами. Для кожного користувача зберігаються реєстраційні дані, роль у системі та

інформація, необхідна для автентифікації й авторизації. Реалізована модель підтримує роботу адміністраторів, менеджерів, операторів, експертів та клієнтів, що відповідає функціональним вимогам проєкту.

Окремим логічним блоком виступає підсистема обліку власників транспортних засобів та самих транспортних засобів. Такий поділ дозволяє уникнути дублювання інформації та забезпечує коректне зберігання даних у випадках, коли один власник володіє декількома автомобілями. Між відповідними таблицями реалізовано зв'язок типу «один до багатьох», що забезпечує гнучкість структури та відповідає реальним бізнес-процесам пунктів технічного контролю.

Центральним елементом бази даних є сутність, що відповідає за зберігання результатів технічного огляду транспортних засобів. Саме вона акумулює інформацію про проведені перевірки, дату проходження огляду, результати діагностики, строк дії технічного контролю та інші параметри, необхідні для формування історії експлуатації транспортного засобу. Наявність зв'язку між записами технічних оглядів та транспортними засобами дозволяє зберігати повну історію проходження перевірок протягом усього життєвого циклу автомобіля.

Особливу увагу під час проєктування було приділено реалізації механізму автоматичних нагадувань. Для цього в структурі бази даних передбачено окрему сутність повідомлень, яка забезпечує зберігання інформації про створені системою сповіщення, їх статус та стан опрацювання користувачем. Використання даного механізму дозволяє автоматизувати процес інформування клієнтів щодо завершення терміну дії обов'язкового технічного контролю та підвищує рівень сервісного обслуговування.

Для підтримки функціоналу електронного документообігу в структурі бази даних передбачено окрему підсистему зберігання документів. Вона забезпечує облік сформованих діагностичних карт, звітів та інших файлів, що генеруються під час роботи системи. Кожен документ пов'язується з відповідним записом технічного огляду, що дозволяє швидко отримувати

доступ до необхідної інформації та забезпечує простежуваність усіх операцій.

Спроектowana структура бази даних відповідає принципам нормалізації та дозволяє мінімізувати дублювання інформації. Використання зовнішніх ключів забезпечує підтримку референтної цілісності даних, а логічний поділ предметної області на окремі сутності спрощує супровід і подальший розвиток системи. Крім того, така модель добре узгоджується з обраною мікросервісною архітектурою, оскільки дозволяє розділити відповідальність між окремими сервісами без порушення логічних зв'язків між даними.

Таким чином, розроблена структура бази даних забезпечує ефективне зберігання, обробку та аналіз інформації про транспортні засоби, результати технічних оглядів, користувачів та документи. Запропонована модель створює надійну основу для реалізації функціональних можливостей інформаційно-аналітичної системи та забезпечує можливість її подальшого масштабування відповідно до потреб предметної області.

Одним із важливих завдань під час розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів стало забезпечення належного рівня захисту даних та розмежування доступу між різними категоріями користувачів. Оскільки система обробляє персональні дані власників транспортних засобів, результати технічних оглядів та службову інформацію підприємства, необхідним є впровадження механізмів автентифікації, авторизації та контролю доступу.

Для реалізації системи безпеки було використано механізм рольового доступу (Role-Based Access Control, RBAC), який дозволяє надавати користувачам лише ті права, що необхідні для виконання їхніх функціональних обов'язків. Такий підхід забезпечує підвищення рівня інформаційної безпеки та зменшує ризик несанкціонованого доступу до критично важливих даних.

У розробленій системі передбачено п'ять основних ролей користувачів:

- Administrator;

- Manager;
- Operator;
- Expert;
- Client.

Кожна роль має власний набір дозволених операцій та рівень доступу до функціональних модулів системи.

Адміністратор є користувачем із найвищим рівнем привілеїв. Він має доступ до всіх функціональних можливостей системи, включаючи керування користувачами, налаштування ролей, перегляд аналітики, формування звітів та адміністрування системних параметрів.

Менеджер відповідає за організацію роботи підприємства та аналіз результатів діяльності. Для нього передбачено доступ до журналів технічних оглядів, статистичної інформації, аналітичних панелей та звітності, проте відсутня можливість адміністрування користувачів.

Оператор здійснює первинну роботу з клієнтами та транспортними засобами. Його функції включають реєстрацію власників транспортних засобів, створення записів на проходження технічного огляду, редагування даних та ведення журналу перевірок.

Експерт виконує безпосередньо процедури технічного контролю. Для нього передбачено можливість внесення результатів огляду, формування висновків щодо технічного стану транспортних засобів та роботу з відповідною документацією.

Клієнт є кінцевим користувачем системи та має мінімальний набір прав. Йому доступний перегляд власної інформації, запис на проходження технічного огляду та отримання повідомлень щодо завершення терміну дії обов'язкового технічного контролю.

Розподіл функціональних можливостей між ролями наведено в таблиці 3.1.

Таблиця 3.1 – Матриця прав доступу користувачів

Функція	Admin	Manager	Operator	Expert	Client
Авторизація в системі	✓	✓	✓	✓	✓
Перегляд власного профілю	✓	✓	✓	✓	✓
Керування користувачами	✓	–	–	–	–
Призначення ролей	✓	–	–	–	–
Створення записів на ТО	✓	✓	✓	–	✓
Редагування записів ТО	✓	✓	✓	✓	–
Перегляд журналу ТО	✓	✓	✓	✓	–
Внесення результатів огляду	✓	–	–	✓	–
Робота з документами	✓	✓	✓	✓	–
Перегляд аналітики та KPI	✓	✓	–	–	–
Генерація PDF-звітів	✓	✓	–	–	–
Перегляд повідомлень	✓	✓	✓	✓	✓
Керування повідомленнями	✓	✓	✓	–	–

У розробленій інформаційно-аналітичній системі автентифікація користувачів реалізована із застосуванням технології JSON Web Token (JWT). Після успішної перевірки введених облікових даних користувачу формується токен доступу, який містить інформацію про його роль, ідентифікаційні параметри та набір дозволених операцій. Надалі цей токен використовується під час кожного звернення до серверної частини системи, де виконується його перевірка та визначаються права доступу до відповідних функціональних модулів.

Використання JWT дозволило реалізувати механізм автентифікації без необхідності зберігання серверних сесій. Такий підхід особливо ефективний у мікросервісному середовищі, оскільки кожен сервіс може самостійно перевіряти валідність отриманого токена та приймати рішення щодо надання доступу до власних ресурсів без звернення до централізованого сховища сесій.

Під час розроблення програмного комплексу було обрано мікросервісну архітектуру. На відміну від монолітних рішень, де всі функціональні можливості об'єднані в межах одного застосунку, у

запропонованій системі логіка розподілена між окремими сервісами, кожен з яких відповідає за власну предметну область. Така організація дозволяє спростити супровід програмного забезпечення, зменшити залежність між компонентами та забезпечити можливість незалежного оновлення окремих модулів.

Взаємодія між сервісами організована за допомогою REST-підходу. Кожен компонент публікує власний набір API-методів, через які відбувається обмін інформацією. Усі зовнішні звернення надходять до API Gateway, який приймає запит та передає його до відповідного сервісу. Завдяки цьому внутрішня структура системи залишається прихованою від користувача.

Основними сервісами системи є:

- AuthService.API – відповідає за реєстрацію користувачів, авторизацію та генерацію JWT-токенів;
- UserService.API – забезпечує керування обліковими записами та ролями користувачів;
- VehicleService.API – реалізує функціонал роботи з транспортними засобами, журналом технічних оглядів, документами та онлайн-записом;
- ReportService.API – виконує формування аналітичних звітів, статистики та PDF-документів;
- ApiGateway – забезпечує маршрутизацію запитів між клієнтською частиною та сервісами системи.

Загальну схему взаємодії між компонентами наведено на рисунку 3.3.

Процес авторизації починається з надсилання користувачем облікових даних через вебінтерфейс. Після цього запит передається через API Gateway до сервісу автентифікації, де виконується перевірка введених даних. У разі успішної перевірки логіна та пароля формується JWT-токен, який повертається клієнту та використовується для подальших запитів до системи.

Для автентифікації та авторизації запитів використовується механізм JWT, детально розглянутий у підрозділі 3.1.

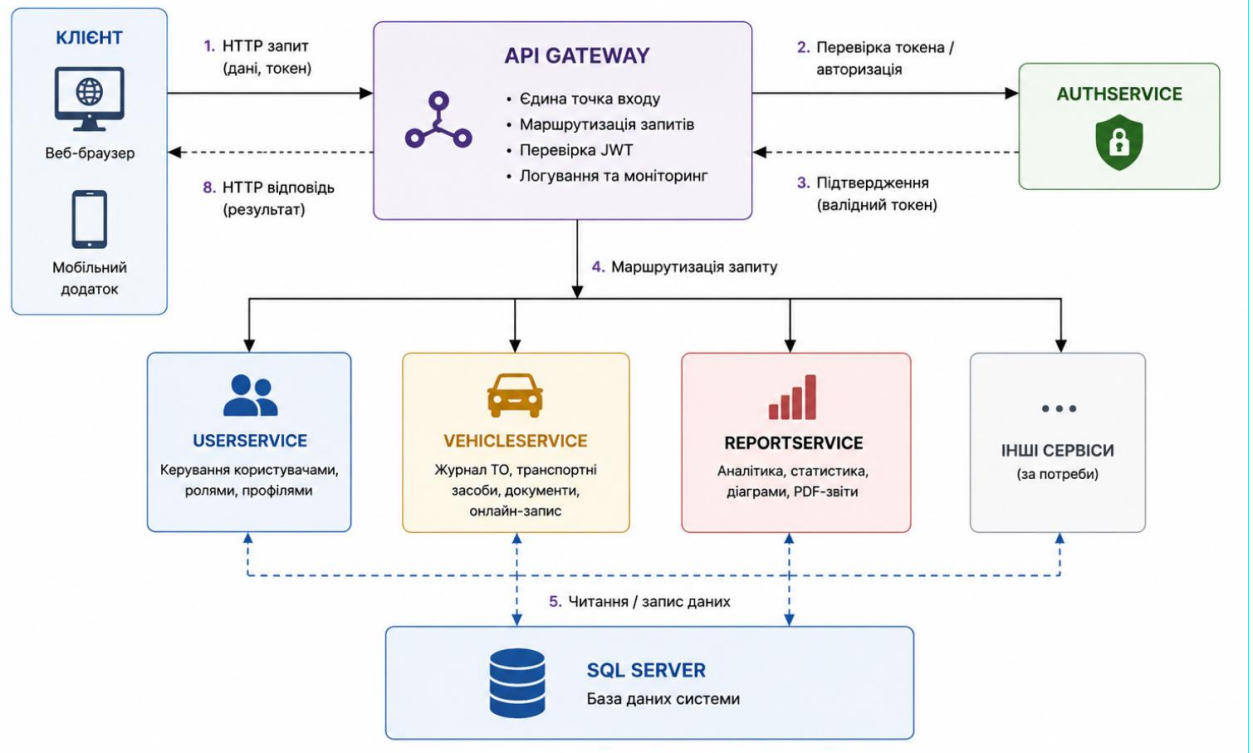


Рисунок 3.3 – Схема взаємодії мікросервісів

Як приклад роботи системи можна розглянути процес внесення нового запису про технічний огляд. Після авторизації користувач заповнює необхідні поля форми та надсилає дані на сервер. Запит спочатку обробляється API Gateway, після чого передається до відповідного сервісу. Далі виконується перевірка отриманих відомостей, їх збереження у базі даних та повернення результату виконання операції.

Подібний механізм використовується і під час формування аналітичної звітності. Необхідні дані надходять до аналітичного сервісу, де здійснюється їх обробка, агрегування та узагальнення. На основі отриманих результатів формуються показники ефективності роботи підприємства, статистичні звіти, графічні візуалізації та PDF-документи, доступні для подальшого перегляду або завантаження.

Використання компонента API Gateway дозволило централізувати процес маршрутизації запитів та ізолювати внутрішні сервіси від прямої взаємодії з клієнтськими застосунками. Крім підвищення рівня безпеки, такий підхід значно спрощує масштабування програмного комплексу,

оскільки окремі сервіси можуть розгортатися, оновлюватися та супроводжуватися незалежно один від одного. У результаті забезпечується стабільна взаємодія між усіма компонентами інформаційної системи та створюється основа для подальшого розширення її функціональних можливостей.

3.2 Реалізація функціональних модулів системи

Важливе місце в структурі розробленої системи займає модуль авторизації та управління користувачами, який відповідає за контроль доступу до функціональних можливостей програмного комплексу. Необхідність реалізації цього модуля обумовлена тим, що система працює з персональними даними клієнтів, інформацією про транспортні засоби, результатами технічних перевірок і сформованою аналітичною звітністю.

Для автентифікації користувачів у системі використовується механізм JWT, описаний у підрозділі 3.1. Після успішної авторизації користувач отримує токен доступу, який використовується під час роботи із захищеними ресурсами системи.

У межах системи реалізовано можливість реєстрації користувачів. Під час реєстрації вказуються основні дані користувача, після чого створюється обліковий запис із відповідною роллю. Для забезпечення безпеки пароль не зберігається у відкритому вигляді, а обробляється з використанням механізмів захисту облікових даних.

Окрему роль у системі відіграє механізм розмежування доступу. У системі передбачено п'ять користувацьких ролей: Admin, Manager, Operator, Expert і Client. Для кожної ролі визначено окремий перелік доступних дій. Наприклад, адміністратор відповідає за керування користувачами та налаштування доступу, менеджер працює зі звітністю та аналітичними матеріалами, оператор здійснює ведення записів технічних оглядів, експерт

вносить результати перевірок, а клієнт використовує функції онлайн-запису та отримання повідомлень.

Використання рольового розподілу прав дало можливість чітко розмежувати доступ до функціональних модулів системи. Такий підхід зменшує ризик випадкового або навмисного втручання у важливі дані та сприяє підвищенню достовірності результатів технічного огляду.

Перевірка працездатності сервісу AuthService.API виконувалася за допомогою Swagger. Через цей інструмент було протестовано основні сценарії взаємодії із сервісом, зокрема реєстрацію нового користувача, авторизацію та отримання токена доступу для подальшої роботи із захищеними маршрутами. Приклад тестування авторизації користувача через Swagger наведено на рисунку 3.4.

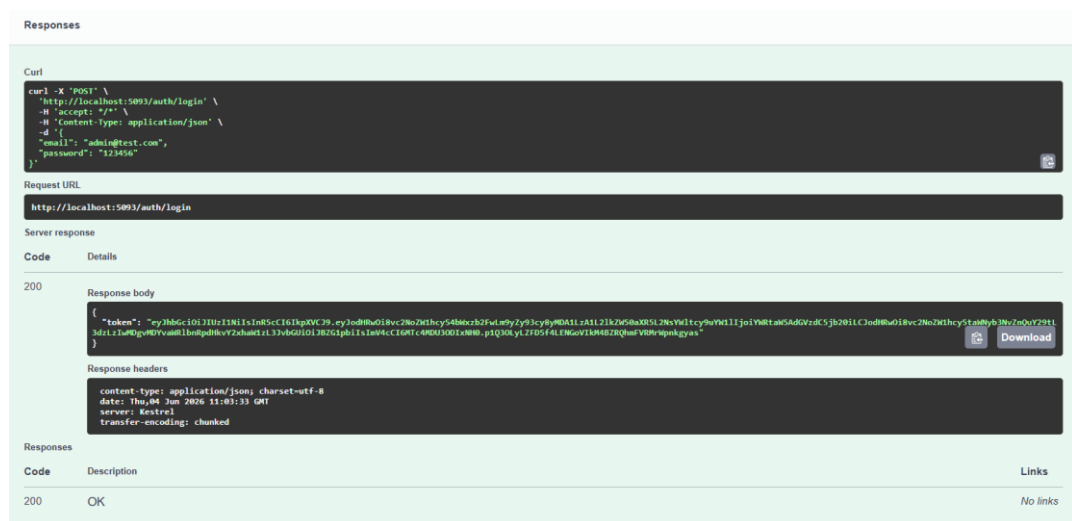


Рисунок 3.4 – Тестування авторизації користувача в Swagger

Після успішного отримання токена він додається до заголовка HTTP-запиту у форматі Bearer Token. Це дозволяє виконувати запити до захищених ресурсів системи відповідно до ролі користувача. Приклад використання JWT-токена для доступу до захищеного API наведено на рисунку 3.5.

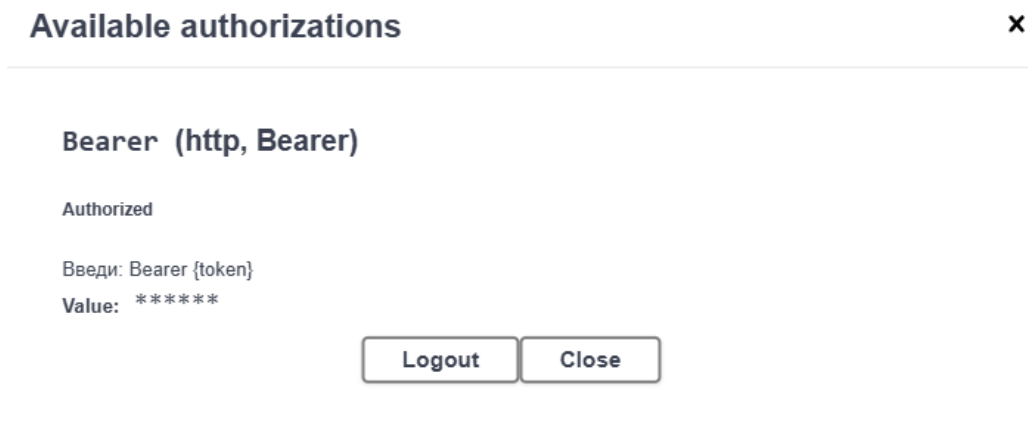


Рисунок 3.5 – Використання JWT-токена для доступу до захищених ресурсів

Крім авторизації, у системі реалізовано модуль управління користувачами. Він дозволяє створювати нові облікові записи, переглядати список користувачів, редагувати їхні дані та призначати ролі. Завдяки цьому адміністратор може централізовано керувати доступом до функціональних можливостей системи без необхідності внесення змін у програмний код.

Отже, впроваджений механізм розмежування прав доступу створює необхідний рівень захисту даних та забезпечує контрольовану роботу користувачів відповідно до їх службових обов’язків. Використання JWT у поєднанні з API Gateway дозволяє ефективно інтегрувати механізми безпеки у мікросервісну архітектуру системи.

Для взаємодії користувачів із системою було реалізовано вебінтерфейси авторизації та реєстрації. Форма авторизації забезпечує введення облікових даних користувача та отримання доступу до функціональних можливостей системи відповідно до призначеної ролі. Інтерфейс авторизації користувача наведено на рисунку Б.1.

Для створення нових облікових записів передбачено форму реєстрації, яка дозволяє ввести необхідні персональні дані та сформувати новий профіль користувача. Після успішної реєстрації обліковий запис зберігається в системі та може використовуватися для подальшої авторизації. Форма реєстрації користувача наведена на рисунку Б.2.

Основна робота користувачів із системою виконується через журнал технічних оглядів транспортних засобів, у якому зосереджено інформацію про результати проведених перевірок. Саме через цей модуль здійснюється зберігання, обробка та подальший аналіз інформації щодо проведених перевірок. У межах одного модуля об'єднано відомості про транспортні засоби, їх власників, результати перевірок та супровідні документи.

Основним призначенням журналу є забезпечення швидкого доступу до інформації про всі проведені технічні огляди. Для кожного запису зберігаються дані про транспортний засіб, його власника, дату проведення огляду, категорію транспортного засобу, результати перевірки та інші службові відомості. Завдяки накопиченню записів формується повна історія проходження технічного контролю для кожного транспортного засобу.

У процесі розроблення було реалізовано механізм створення нових записів журналу. Після введення необхідних даних система виконує перевірку коректності заповнення полів та зберігає інформацію до бази даних. Використання централізованого сховища забезпечує цілісність даних та виключає дублювання інформації.

Для підвищення зручності роботи операторів реалізовано функцію редагування існуючих записів. Вона дозволяє оперативно вносити зміни до інформації у випадках виявлення помилок або необхідності актуалізації відомостей. Усі зміни відображаються безпосередньо в журналі та стають доступними для подальшої роботи інших користувачів системи.

Інтерфейс журналу технічних оглядів наведено на рисунку Б.3.

Оскільки кількість записів у системі може постійно зростати, важливим завданням стало забезпечення швидкого пошуку необхідної інформації. Для цього реалізовано механізми фільтрації та пошуку записів за ключовими параметрами. Користувач може знаходити потрібні записи за даними власника, номерним знаком транспортного засобу, датою проведення огляду або іншими характеристиками. Це значно скорочує час роботи з великими масивами інформації та підвищує ефективність використання системи.

Додатково в журналі реалізовано відображення контактних даних власників транспортних засобів, зокрема номерів телефонів. Наявність цієї інформації дозволяє оперативно зв'язуватися з клієнтами для уточнення даних, повідомлення про результати технічного огляду або нагадування про необхідність повторного проходження контролю. Приклад відображення інформації про власника транспортного засобу наведено на рисунку Б.4.

Для автоматизації перенесення накопичених даних до системи було реалізовано механізм імпорту журналу технічних оглядів з електронних таблиць Excel. Необхідність такого функціоналу обумовлена тим, що значна частина підприємств уже має історичні архіви даних у табличному форматі. Ручне перенесення великої кількості записів потребувало б значних часових витрат та створювало б ризик появи помилок.

Імпорт реалізовано у вигляді окремого сервісу VehicleService.API із використанням бібліотеки EPPlus. Після завантаження файлу система автоматично виконує зчитування даних, перевірку їх коректності та подальше збереження до бази даних. У випадку виявлення помилок користувач отримує відповідне повідомлення про результати обробки файлу.

Приклад імпорту журналу технічних оглядів з Excel наведено на рисунку 3.6.

Імпорт Excel

Оберіть Excel-файл

Выберите файл

Adjusted_Project_Costs_AI.xlsx

Імпортувати

Назад до журналу

Рисунок 3.6 – Імпорт журналу технічних оглядів із файлу Excel

Розроблений модуль журналу технічних оглядів став основою функціонування всієї інформаційно-аналітичної системи. Саме дані, що накопичуються в журналі, використовуються іншими сервісами для формування звітності, аналітичних показників, статистики та системи

автоматичних нагадувань. Завдяки централізованому підходу до зберігання інформації забезпечується висока цілісність даних, спрощується їх обробка та створюються умови для подальшого розвитку функціональних можливостей системи.

Сучасні програмні рішення дедалі частіше орієнтуються на дистанційну взаємодію між підприємством та клієнтами. У межах розробленої інформаційно-аналітичної системи реалізовано модулі онлайн-запису на технічний огляд та роботи з електронними документами. Їх впровадження дозволило зменшити навантаження на персонал, скоротити час обслуговування клієнтів та підвищити загальний рівень автоматизації бізнес-процесів.

За допомогою вебінтерфейсу користувач може самостійно обрати дату та час проходження технічного огляду й подати відповідну заявку. Користувач через веб-інтерфейс може заповнити відповідну форму, вказавши основні дані про себе та транспортний засіб. Після надсилання заявки інформація автоматично зберігається у базі даних та стає доступною для перегляду співробітникам підприємства.

Завдяки використанню централізованої бази даних усі заявки зберігаються в єдиному інформаційному середовищі, що виключає дублювання інформації та дозволяє контролювати завантаженість пункту технічного огляду. Оператори мають можливість переглядати список зареєстрованих заявок, змінювати їх статус та планувати подальшу роботу відповідно до поточного навантаження.

Інтерфейс онлайн-запису на технічний огляд наведено на рисунку Б.5.

Запровадження цього механізму спростило процедуру запису на технічний огляд та зменшило навантаження на працівників пункту технічного контролю. Це особливо актуально при великій кількості клієнтів та необхідності оптимізації процесу прийому транспортних засобів на перевірку.

Окрім роботи із заявками, у системі реалізовано окремий модуль керування документами. Його основне призначення полягає у зберіганні та централізованому доступі до електронних документів, що використовуються під час проведення технічного огляду. До таких документів можуть належати результати перевірок, додаткові матеріали, скановані копії документів клієнтів та інші супровідні файли.

Для реалізації цього функціоналу було розроблено окремий мікросервіс `DocumentService.API`, який забезпечує завантаження, зберігання та подальший доступ до файлів. Використання окремого сервісу дозволяє ізолювати роботу з документами від інших компонентів системи та спрощує подальше масштабування функціоналу.

Користувачі, які мають відповідні права доступу, можуть додавати нові документи безпосередньо через вебінтерфейс. Після завантаження файл реєструється у системі та прив'язується до відповідного запису технічного огляду або транспортного засобу. Це забезпечує швидкий пошук необхідних документів та спрощує ведення електронного архіву.

Інтерфейс роботи з документами наведено на рисунку Б.6.

Важливою перевагою реалізованого підходу є можливість централізованого зберігання файлів та їх інтеграція з іншими функціональними модулями системи. Документи можуть використовуватися під час формування звітності, перегляду історії технічних оглядів або виконання адміністративних процедур.

Для забезпечення коректності роботи було реалізовано перевірку завантажуваних файлів та механізми контролю доступу. Завдяки цьому лише авторизовані користувачі можуть переглядати або додавати документи відповідно до призначених ролей.

Приклад завантаження документа до системи наведено на рисунку Б.7.

Таким чином, реалізовані модулі онлайн-запису та роботи з документами дозволили суттєво розширити функціональні можливості інформаційно-аналітичної системи. Вони забезпечують ефективну взаємодію

з клієнтами, автоматизують процес подання заявок, спрощують ведення електронного архіву та створюють основу для подальшого розвитку цифрових сервісів підприємства.

3.3 Реалізація аналітичної підсистеми та формування звітності

Однією з ключових особливостей розробленої інформаційно-аналітичної системи є наявність аналітичної підсистеми, призначеної для збору, обробки та візуалізації накопичених даних. Її основне завдання полягає у перетворенні великих масивів інформації, що накопичуються під час проведення технічних оглядів, на зручні для сприйняття аналітичні показники та статистичні звіти.

У більшості традиційних систем технічного огляду дані використовуються лише як архівна інформація та практично не застосовуються для оцінювання ефективності роботи підприємства. У розробленій системі реалізовано окремий аналітичний модуль, який дозволяє отримувати узагальнену інформацію про діяльність підприємства у режимі реального часу.

Основою аналітичної підсистеми виступає Dashboard, який об'єднує ключові показники діяльності підприємства на одній сторінці. Такий підхід дозволяє керівникам та відповідальним працівникам оперативно оцінювати поточний стан роботи без необхідності перегляду великої кількості окремих записів.

Загальний вигляд аналітичної панелі наведено на рисунку Б.8.

Для швидкого аналізу стану підприємства реалізовано систему KPI-показників. KPI-картки відображають найважливіші характеристики діяльності системи, зокрема загальну кількість технічних оглядів, кількість зареєстрованих клієнтів, число активних заявок та інші показники, необхідні для оцінювання роботи підприємства.

Використання KPI дозволяє оперативно отримувати узагальнену інформацію без необхідності виконання додаткових запитів до бази даних. Завдяки цьому користувачі можуть швидко реагувати на зміни у роботі підприємства та приймати обґрунтовані управлінські рішення.

Для забезпечення гнучкого аналізу реалізовано механізм фільтрації статистичних даних за часовими періодами. Користувач може переглядати інформацію за конкретний день, місяць, квартал або довільний часовий інтервал. Це дозволяє виконувати порівняльний аналіз показників та оцінювати тенденції зміни діяльності підприємства в часі.

Приклад використання фільтрації статистики за часовим періодом наведено на рисунку 3.7.

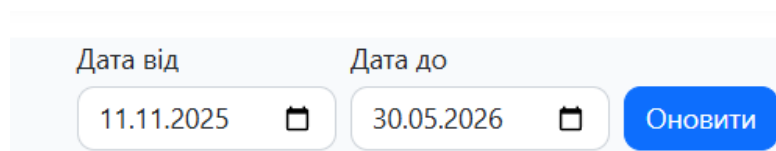
The image shows a user interface for filtering data by date. It consists of two input fields: 'Дата від' (Date from) and 'Дата до' (Date to). The 'Дата від' field contains the date '11.11.2025' and has a small calendar icon to its right. The 'Дата до' field contains the date '30.05.2026' and also has a small calendar icon to its right. To the right of these two fields is a blue button with the text 'Оновити' (Update) in white.

Рисунок 3.7 – Фільтрація аналітичних даних за часовим періодом

Важливим компонентом аналітичної підсистеми є модуль статистичної обробки даних. Він виконує автоматичне узагальнення інформації про технічні огляди, клієнтів та транспортні засоби. На основі накопичених записів система формує статистичні показники, які використовуються для подальшої візуалізації.

Для підвищення наочності представлення інформації реалізовано механізми побудови діаграм. Використання графічних елементів дозволяє значно спростити аналіз даних порівняно з традиційними табличними звітами.

У межах системи реалізовано кругові діаграми, які відображають структуру транспортних засобів за категоріями та типами. Такий спосіб подання інформації дозволяє швидко оцінити розподіл клієнтської бази та визначити найбільш поширені категорії транспортних засобів.

Приклад використання кругових діаграм наведено на рисунку Б.9.

Застосування аналітичної підсистеми дозволяє не лише отримувати поточну статистичну інформацію, а й формувати основу для подальшого розвитку системи бізнес-аналітики. Накопичені дані можуть використовуватися для прогнозування навантаження на пункт технічного огляду, аналізу поведінки клієнтів та оцінювання ефективності роботи окремих підрозділів підприємства.

Таким чином, реалізована аналітична підсистема забезпечує автоматичний збір, обробку та візуалізацію даних, що накопичуються під час експлуатації інформаційної системи. Використання КРІ-показників, статистичних звітів, фільтрів та графічних інструментів дозволяє суттєво підвищити інформативність системи та забезпечити ефективну підтримку процесу прийняття управлінських рішень.

Одним із важливих функціональних модулів розробленої інформаційно-аналітичної системи є підсистема автоматичних нагадувань про завершення терміну дії обов'язкового технічного контролю транспортних засобів. Її впровадження спрямоване на підвищення рівня обслуговування клієнтів, автоматизацію процесу контролю термінів проходження технічного огляду та зменшення ризику пропуску встановлених законодавством строків.

У традиційній практиці працівники підприємства змушені вручну контролювати терміни завершення дії технічних оглядів, переглядаючи журнали або електронні таблиці. Такий підхід потребує значних витрат часу та збільшує ймовірність виникнення помилок через людський фактор. Для усунення зазначених недоліків у системі реалізовано механізм автоматичного аналізу записів технічних оглядів та формування відповідних повідомлень.

Робота підсистеми базується на регулярній перевірці дати завершення терміну дії технічного контролю. Під час виконання алгоритму система порівнює поточну дату із датою завершення чергового огляду та визначає необхідність створення повідомлення.

Перший тип нагадувань формується за тридцять календарних днів до завершення строку дії технічного огляду. Такий підхід дозволяє завчасно повідомити клієнта про необхідність повторного проходження процедури та надати достатній час для планування візиту до пункту технічного контролю.

Другий тип повідомлень створюється безпосередньо у день завершення терміну дії технічного огляду. Його основне призначення полягає у нагадуванні про необхідність негайного проходження повторної перевірки транспортного засобу.

Загальний алгоритм роботи підсистеми нагадувань наведено на рисунку 3.8.

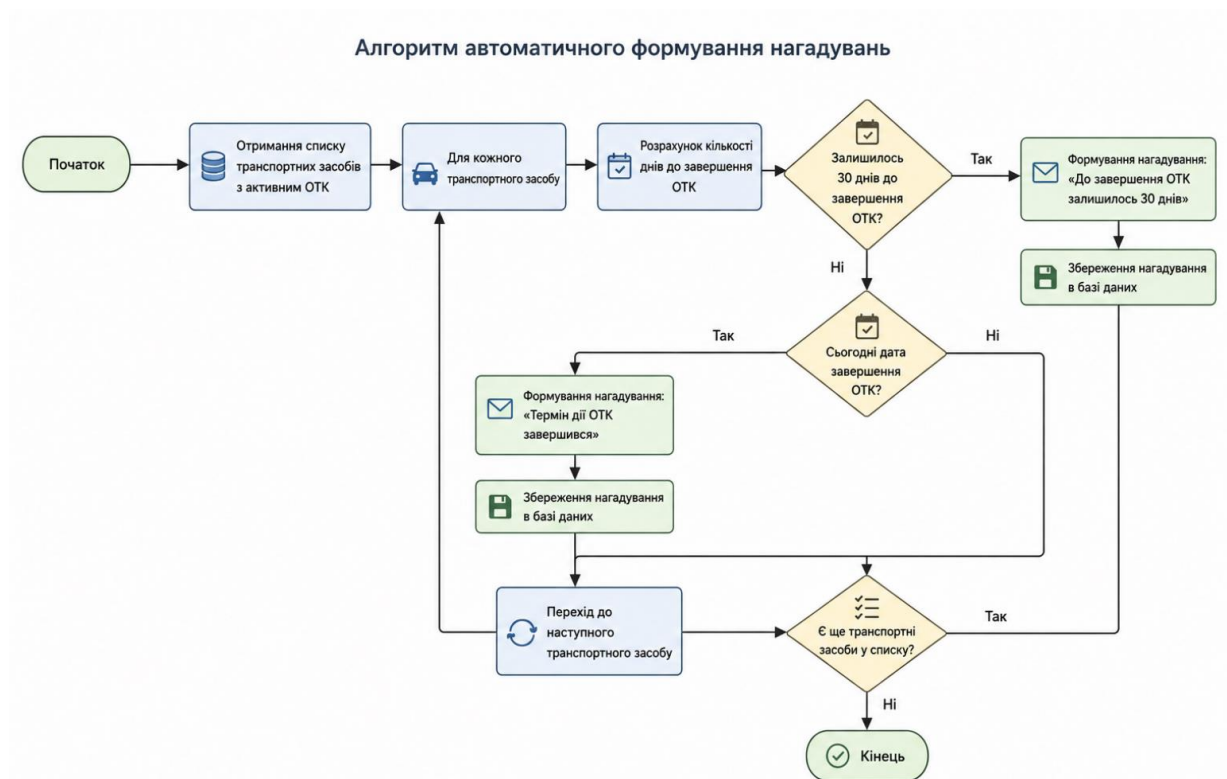


Рисунок 3.8 – Алгоритм автоматичного формування нагадувань

Після створення повідомлення воно автоматично додається до списку активних нагадувань. Активні повідомлення являють собою перелік нагадувань, які потребують подальшого опрацювання працівником підприємства або використання під час комунікації з клієнтом.

Для підвищення зручності роботи операторів кожне повідомлення містить інформацію про власника транспортного засобу, контактний номер телефону, дату завершення технічного огляду та текст сформованого нагадування. Крім того, реалізовано можливість швидкого копіювання тексту повідомлення для подальшого надсилання клієнту за допомогою зовнішніх засобів комунікації.

Інтерфейс перегляду активних нагадувань наведено на рисунку Б.10.

Після обробки повідомлення користувач може змінити його статус на прочитане. У такому випадку запис переміщується до окремого списку оброблених повідомлень. Це дозволяє відокремити актуальні нагадування від уже виконаних та забезпечує більш зручну організацію роботи персоналу.

Водночас у системі реалізовано механізм повернення повідомлень до списку активних. Така функція може бути використана у випадках помилкового закриття повідомлення або необхідності повторного контакту з клієнтом. Після виконання відповідної дії повідомлення автоматично повертається до активного переліку та знову стає доступним для обробки.

Інтерфейс роботи з активними та прочитаними повідомленнями наведено на рисунку 3.9.

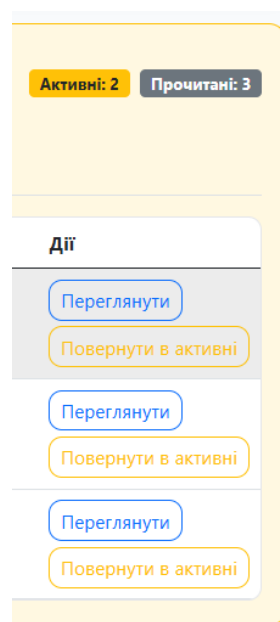


Рисунок 3.9 – Керування статусами нагадувань

Реалізований механізм нагадувань забезпечує автоматичний контроль термінів проходження технічного огляду, зменшує навантаження на працівників підприємства та сприяє підвищенню якості обслуговування клієнтів. Використання автоматизованого формування повідомлень дозволяє мінімізувати ризик пропуску строків повторного проходження технічного контролю та підвищує ефективність роботи всієї інформаційної системи.

Завершальним етапом обробки накопичених даних у розробленій системі є підготовка аналітичної звітності. Для виконання цього завдання використовується сервіс ReportService.API, який формує звіти на основі інформації про проведені технічні огляди, клієнтів та транспортні засоби.

Виділення механізму формування звітів в окремий компонент дало змогу розвантажити інші модулі системи та зосередити всі операції, пов'язані з підготовкою звітної документації, в одному місці. Такий підхід спрощує супровід програмного забезпечення та створює додаткові можливості для подальшого розвитку функцій аналітики.

Під час підготовки звіту сервіс отримує необхідні відомості з бази даних, виконує їх узагальнення та формує підсумковий документ у форматі PDF. Для побудови звітів використовуються дані журналу технічних оглядів, інформація про клієнтів та результати роботи аналітичної підсистеми.

Сформований документ містить не лише числові показники та графічні матеріали, а й текстовий підсумок отриманих результатів. Це дає можливість швидко оцінити поточний стан діяльності підприємства без необхідності детального аналізу кожного окремого показника.

Під час створення файлу система автоматично присвоює йому назву з урахуванням службових параметрів та дати формування звіту. Завдяки цьому спрощується подальше впорядкування документів і пошук потрібних матеріалів у сховищі.

Після завершення формування звіту користувач отримує готовий PDF-документ, який може бути збережений, переданий керівництву або

використаний для подальшого аналізу діяльності підприємства. Приклад сформованого звіту наведено на рисунках В.1–В.3.

Практичне використання автоматизованого формування звітності дозволяє скоротити витрати часу на підготовку документів та зменшити ймовірність помилок, пов'язаних із ручною обробкою інформації. Оскільки всі розрахунки виконуються безпосередньо на основі даних, що містяться у системі, отримані результати залишаються актуальними та узгодженими між собою.

Таким чином, реалізований механізм генерації PDF-звітів забезпечує зручне формування аналітичної документації, яка поєднує статистичні показники, графіки та узагальнені висновки й може використовуватися для підтримки управлінських рішень.

3.4 Тестування та оцінювання результатів розроблення

Після завершення основних етапів реалізації інформаційно-аналітичної системи було проведено функціональне тестування її ключових модулів. Метою тестування було перевірити коректність роботи реалізованих функцій, відповідність фактичної поведінки системи очікуваним результатам, а також виявити можливі помилки у взаємодії між окремими компонентами.

Функціональне тестування охоплювало основні сценарії використання системи, зокрема авторизацію користувачів, роботу з ролями, ведення журналу технічних оглядів, редагування записів, імпорт даних з Excel, онлайн-запис на технічний огляд, роботу з документами, формування аналітики, нагадувань та PDF-звітів.

Перевірка виконувалася шляхом послідовного запуску окремих функцій через вебінтерфейс системи та, за необхідності, через Swagger-інтерфейс відповідних API-сервісів. Для кожної функції визначався

очікуваний результат, після чого він порівнювався з фактично отриманим результатом.

Результати функціонального тестування наведено в таблиці 3.2.

Таблиця 3.2 – Результати функціонального тестування системи

№	Функція	Очікуваний результат	Фактичний результат
1	2	3	4
1	Авторизація користувача	Після введення коректних облікових даних користувач входить до системи	Авторизація виконана успішно, користувач отримує доступ до системи
2	Отримання JWT-токена	Після успішного входу система формує JWT-токен	JWT-токен сформовано та використано для доступу до захищених ресурсів
3	Реєстрація нового користувача	Система створює новий обліковий запис	Нового користувача успішно додано до бази даних
4	Перевірка рольового доступу	Користувач бачить лише функції, доступні його ролі	Розмежування доступу працює відповідно до заданої ролі
5	Створення запису технічного огляду	Новий запис зберігається у журналі ТО	Запис успішно створено та відображено у журналі
6	Редагування запису технічного огляду	Змінені дані оновлюються у базі даних	Дані успішно оновлено та відображено в інтерфейсі

Продовження таблиці 3.2

1	2	3	4
7	Пошук записів у журналі ТО	Система відображає записи відповідно до введених параметрів пошуку	Пошук працює коректно, знайдено відповідні записи
8	Відображення телефону власника	У деталях запису відображається контактний номер клієнта	Телефон власника коректно відображається у системі
9	Імпорт журналу з Excel	Дані з Excel-файлу завантажуються до бази даних	Імпорт виконано успішно, записи додано до журналу
10	Онлайн-запис на ТО	Клієнт може створити заявку на проходження технічного огляду	Заявку успішно створено та збережено у системі
11	Завантаження документа	Файл додається до системи та прив'язується до відповідного запису	Документ успішно завантажено та доступний для перегляду
12	Відображення КРІ-показників	Dashboard показує актуальні статистичні дані	КРІ-картки відображають коректні показники
13	Фільтрація аналітики за періодом	Система оновлює статистику відповідно до вибраного періоду	Фільтр працює коректно, дані оновлюються
14	Формування автоматичних нагадувань	Система створює повідомлення за 30 днів та в день завершення ОТК	Нагадування створюються та відображаються у списку активних

Продовження таблиці 3.2

1	2	3	4
15	Генерація PDF-звіту	Система формує PDF-файл зі статистикою, діаграмами та висновком	PDF-звіт сформовано успішно, файл доступний для завантаження

За результатами тестування встановлено, що основні функціональні модулі системи працюють коректно та відповідають поставленим вимогам. Усі перевірені сценарії завершилися успішно, а отримані результати збігаються з очікуваними. Це свідчить про працездатність розробленого програмного продукту та можливість його подальшого використання для автоматизації процесів технічного огляду транспортних засобів.

Окрему увагу під час тестування було приділено перевірці взаємодії між мікросервісами. Зокрема, було підтверджено коректну роботу авторизації через AuthService.API, обробку даних технічних оглядів у VehicleService.API, формування PDF-звітів через ReportService.API та централізовану маршрутизацію запитів через API Gateway.

Таким чином, функціональне тестування підтвердило, що розроблена інформаційно-аналітична система забезпечує виконання основних бізнес-процесів, підтримує рольову модель доступу, коректно обробляє дані та формує аналітичну звітність.

Після завершення розроблення та тестування інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів було проведено аналіз отриманих результатів. Основною метою аналізу стала оцінка ефективності реалізованого рішення щодо усунення проблем, які були виявлені під час роботи предметної області та аналізу існуючих програмних продуктів.

Використання мікросервісної архітектури забезпечило модульність, масштабованість та зручність супроводу програмного забезпечення. Реалізовані механізми авторизації, управління користувачами, ведення

журналу технічних оглядів, роботи з документами, формування аналітики та генерації звітів функціонують як єдиний інтегрований програмний комплекс, що забезпечує ефективну автоматизацію діяльності підприємств технічного контролю транспортних засобів.

Загалом результати розробки підтверджують, що створена інформаційно-аналітична система успішно вирішує основні проблеми, які були визначені під час аналізу предметної області. Зокрема, забезпечується централізоване зберігання інформації, автоматизація ключових бізнес-процесів, підтримка аналітичної обробки даних, автоматичне інформування клієнтів та формування звітності. Це дозволяє підвищити ефективність роботи підприємств, що здійснюють технічний огляд транспортних засобів, а також створює основу для подальшого розвитку цифрових сервісів у даній галузі.

Таким чином, результати виконаної роботи підтверджують доцільність використання мікросервісної архітектури для побудови інформаційно-аналітичних систем технічного огляду транспортних засобів. Розроблена система відповідає поставленим вимогам і може бути використана у практичній діяльності підприємств відповідного профілю.

3.5 Перспективи розвитку системи

Розроблена інформаційно-аналітична система автоматизації технічного огляду транспортних засобів реалізує основний набір функцій, необхідних для підтримки процесів обліку, контролю, аналітики та звітності. Водночас сучасні тенденції розвитку інформаційних технологій відкривають широкі можливості для подальшого розширення функціональних можливостей системи та підвищення ефективності її використання.

Одним із перспективних напрямків розвитку є впровадження Telegram-бота для автоматизованої взаємодії з клієнтами. Використання месенджерів

дозволить організувати швидке інформування власників транспортних засобів про наближення терміну проходження технічного огляду, зміну статусу заявки або готовність документів. Крім того, Telegram-бот може використовуватися для попереднього запису на технічний огляд, отримання довідкової інформації та перегляду історії звернень клієнта.

Додатковим напрямком розвитку є реалізація SMS-сповіщень. На відміну від месенджерів, SMS-повідомлення можуть бути доставлені навіть користувачам, які не використовують спеціалізовані мобільні застосунки. Інтеграція із сервісами масових розсилок дозволить автоматично надсилати нагадування про завершення терміну дії обов'язкового технічного контролю та інші важливі повідомлення.

Перспективним також є забезпечення інтеграції із зовнішніми державними інформаційними системами та реєстрами. Така інтеграція дозволить автоматизувати перевірку даних транспортних засобів, отримання відомостей про власників та обмін інформацією між різними інформаційними системами. У результаті зменшиться кількість ручних операцій, підвищиться достовірність даних та скоротиться час обробки інформації.

Наступним напрямком розвитку може стати створення мобільного застосунку для клієнтів та співробітників підприємства. Мобільна версія системи забезпечить можливість перегляду інформації про транспортні засоби, отримання нагадувань, запису на технічний огляд та доступу до сформованих документів безпосередньо зі смартфона. Для працівників підприємства мобільний застосунок може використовуватися як додатковий інструмент оперативного доступу до даних під час проведення перевірок.

З точки зору інфраструктури перспективним є подальший розвиток контейнерної архітектури із застосуванням систем оркестрації контейнерів. На даний момент для розгортання мікросервісів використовується Docker, що забезпечує ізоляцію сервісів та спрощує процес розгортання. У майбутньому доцільним є використання Kubernetes, який дозволяє автоматизувати

масштабування сервісів, балансування навантаження, моніторинг стану компонентів та їх автоматичне відновлення у випадку відмов.

Використання Kubernetes особливо актуальне для великих організацій або мереж пунктів технічного огляду, де навантаження на систему може значно змінюватися залежно від кількості користувачів та обсягу оброблюваних даних. Завдяки автоматичному керуванню контейнерами система зможе ефективно використовувати обчислювальні ресурси та забезпечувати високу доступність сервісів.

Таким чином, архітектура розробленої інформаційно-аналітичної системи створює достатній запас масштабованості для подальшого розвитку. Використання мікросервісного підходу дозволяє поступово додавати нові функціональні модулі без суттєвого впливу на вже реалізовані компоненти. Запропоновані напрямки розвитку можуть суттєво підвищити рівень автоматизації, розширити можливості взаємодії з клієнтами та забезпечити подальше вдосконалення системи відповідно до сучасних вимог цифрової трансформації.

ВИСНОВКИ

У кваліфікаційній роботі вирішено задачу розроблення інформаційно-аналітичної системи автоматизації технічного огляду транспортних засобів на основі мікросервісної архітектури. Проведено аналіз сучасного стану автоматизації процесів технічного огляду транспортних засобів та визначено основні недоліки існуючих програмних рішень, серед яких недостатня масштабованість, обмежені аналітичні можливості та складність інтеграції окремих компонентів системи.

У ході роботи досліджено сучасні підходи до побудови інформаційних систем та обґрунтовано доцільність використання мікросервісної архітектури, REST API та контейнеризації Docker для реалізації системи автоматизації технічного огляду транспортних засобів. Використання такого підходу дозволило забезпечити модульність, незалежність сервісів та можливість подальшого масштабування програмного забезпечення.

У межах кваліфікаційної роботи виконано моделювання предметної області та основних бізнес-процесів системи. Спроектовано архітектуру програмного забезпечення, структуру бази даних та механізми взаємодії між мікросервісами через API Gateway. Для моделювання структури системи використано UML-діаграми та ER-модель бази даних.

Розроблена система забезпечує реєстрацію клієнтів і транспортних засобів, формування заявок на проходження технічного огляду, збереження результатів перевірок, автоматичне створення звітності та аналітичну обробку статистичних даних. Також реалізовано механізми автентифікації та авторизації користувачів із використанням JWT-токенів і захищеного обміну даними через HTTPS.

У роботі реалізовано аналітичну підсистему, яка забезпечує формування статистичних показників, аналіз ефективності роботи підприємства та автоматичне створення звітності. Використання дашбордів та засобів візуалізації дозволяє підвищити ефективність моніторингу

діяльності станції технічного огляду та покращити підтримку управлінських рішень.

Практична цінність отриманих результатів полягає у можливості використання розробленої інформаційно-аналітичної системи на станціях технічного огляду транспортних засобів та підприємствах автомобільного сервісу. Впровадження системи дозволяє зменшити вплив людського чинника, підвищити швидкість обробки інформації та покращити контроль за діяльністю підприємства.

Результати роботи апробовано у вигляді тез доповіді під час конференції «Інформатика, управління та штучний інтелект» (ІУШІ-2026) [1].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мізяк Д.О. Інформаційна система як засіб підвищення ефективності підприємства. Інформатика, управління та штучний інтелект: тези тринадцятої міжнародної науково-технічної конференції (Харків – Краматорськ – Хуст, 13–15 травня 2026 р.). Харків: НТУ «ХП», 2026. С. 86.
2. Shafronenko, A. Y., Bodyanskiy, Y. V., & Rudenko, D. A. (2020). Evolutionary Algorithms Based on Cat Swarm Optomozation in Fuzzy Clustering Tasks.
3. Shafronenko, A. Y., Bodyanskiy, Y. V., & Pliss, I. P. (2019, September). The Fast Modification of Evolutionary Bioinspired Cat Swarm Optimization Method. In *2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL)* (pp. 548-552). IEEE.
4. Bodyanskiy, Y. V., Shafronenko, A. Y., & Klymova, I. N. (2021). Online fuzzy clustering of incomplete data using credibilistic approach and similarity measure of special type. *Radio Electronics, Computer Science, Control*, (1), 97-104.
5. Shafronenko, A. Y., Bodyanskiy, Y. V., Shafronenko, Y. O., Brodetskyi, F. A., & Tanianskyi, O. S. (2026). Tunable Squashing Activation Function for Deep Neural Networks. *Radio Electronics, Computer Science, Control*, (1), 49-54.
6. Bodyanskiy, Y. V., Pliss, I. P., & Shafronenko, A. Y. (2022). Кластеризація масивів даних на основі комбінованої оптимізації функцій щільності розподілу та еволюційного методу котячих зграй. *Radio Electronics, Computer Science, Control*, (4), 61-61.
7. Bodyanskiy, Y. V., Pliss, I. P., Shafronenko, A. Y., & Kalynychenko, O. V. (2022). Нечітка довірча кластеризація даних на основі аналізу щільності розподілу даних та їх піків. *Radio Electronics, Computer Science, Control*, (3), 58-58.

8. Bodyanskiy, Y. V., Pliss, I. P., & Shafronenko, A. Y. (2022). Швидка нечітка правдоподібна кластеризація на основі аналізу піків щільності розподілу даних. *Radio Electronics, Computer Science, Control*, (1), 76-76.
9. Bodyanskiy, Y. V., Shafronenko, A. Y., Rudenko, D. A., & Klymova, I. N. (2020). Online Recurrent Method Of Credibilistic Fuzzy Clusterin.
10. Bodyanskiy, Y. V., Shafronenko, A. Y., Rudenko, D. A., & Klymova, I. N. (2020). Online Recurrent Method Of Credibilistic Fuzzy Clusterin.
11. Shafronenko, A. Y., Bodyanskiy, Y. V., & Rudenko, D. A. (2020). Evolutionary Algorithms Based on Cat Swarm Optimization in Fuzzy Clustering Tasks.
12. Bodyanskiy, Y. V., Shafronenko, A. Y., & Klymova, I. N. (2021). Онлайн метод можливісної кластеризації даних на основі еволюційної оптимізації котячих зграй. *Radio Electronics, Computer Science, Control*, (2), 65-70.
13. Bodyanskiy, Y. V., Shafronenko, A. Y., & Klymova, I. N. (2020). Online Credibilistic Fuzzy Clustering of Data with Gaps.
14. Shafronenko, A. Y., Bodyanskiy, Y. V., & Holovin, O. O. (2023). Кластеризація масивів даних на основі модифікованого алгоритму сірого вовка. *Radio Electronics, Computer Science, Control*, (1), 73-73.
15. Bodyanskiy, Y. V., & Shafronenko, A. Y. (2019). Online algorithm for possibilistic fuzzy clustering based on evolutionary cat swarm optimization.
16. Shafronenko, A. U., Volkova, V. V., & Bodianskiy, E. V. (2011). Адаптивна кластеризація даних з пропущеними значеннями. *Radio Electronics, Computer Science, Control*, (2).
17. Newman, S. (2021). *Building microservices: designing fine-grained systems*. " O'Reilly Media, Inc."
18. Richardson, C. (2018). *Microservices patterns: with examples in Java*. Simon and Schuster.
19. Fowler, M. (2012). *Patterns of enterprise application architecture*. Addison-Wesley.

20. Evans, E. (2004). *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional.
21. Kleppmann, M. (2019). *Designing data-intensive applications*.
22. Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*. James Turnbull.
23. Burns, B., Beda, J., Hightower, K., & Evenson, L. (2022). *Kubernetes: up and running: dive into the future of infrastructure*. " O'Reilly Media, Inc."
24. Classon, I. (2024). *Migrating ASP. NET Microservices to ASP. NET Core 8*. Apress.
25. Gorman, B. (2026). *Practical Entity Framework Core 10: Database Access for Enterprise Applications*. Springer Nature.
26. Borglund, N. (2024). *Security and Application Deployment using Docker*.
27. Yildiz, H., & Küpper, A. (2026). *interID--An Ecosystem-agnostic Verifier-as-a-Service with OpenID Connect Bridge*. *arXiv preprint arXiv:2602.14871*.
28. Fernandes, J. P. (2024). *Authentication API-A SSO Authentication and Authorisation Infrastructure for Web*.
29. Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. University of California, Irvine.
30. Agyekum, K. (2023). *Thingsboard API Gateway for the Savonia WaterLAB*.
31. Documentation, A. W. S. (2024). *AWS Documentation*.
32. Sánchez, A. G. (2024). *Azure OpenAI Service for Cloud Native Applications: Designing, Planning, and Implementing Generative AI Solutions*. " O'Reilly Media, Inc."