

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: «Проектування інформаційної системи з контролю стану
комплектуючих автомобіля»

здобувача Франчука Миколи Костянтиновича

(ПІБ)

(Підпис)

Науковий керівник:

к.е.н., доцент

_____ Тішков Б.О.

**Робота допущена до захисту перед
екзаменаційною комісією**

з атестації здобувачів вищої освіти

завідувач кафедри:

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

_____ Помазун О.М.
“ ____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри
_____ Тішков Б.О.

“ ____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачу вищої освіти **Франчуку Миколі Костянтиновичу**

очної (денної) форми навчання

на підготовку кваліфікаційної бакалаврської роботи
на тему: «Проектування інформаційної системи з контролю стану
комплектуючих автомобіля»

Тему затверджено наказом ректора Університету від « 7 » березня 2025 р.
№ 466- ст.

Кваліфікаційна бакалаврська робота виконується на матеріалах отриманих під час навчання, даних з навчальних посібників та інтернет - ресурсів, а також на фактичному матеріалі, зібраному автором за темою дослідження при виконанні курсових робіт.

План кваліфікаційної бакалаврської роботи

Розділ I Характеристика та аналіз предметної області

Розділ II Розробка вимог і моделювання інформаційної системи

Розділ III Проектування та реалізація компонентів системи

Об'єкт дослідження процес контролю технічного стану комплектуючих автомобіля.

Предмет дослідження є інформаційна система для моніторингу, аналізу та візуалізації даних про стан компонентів автомобіля, що базується на використанні технології OBD-II та апаратної платформи Raspberry Pi.

Мета кваліфікаційної бакалаврської роботи полягає в проектуванні та розробці інформаційної системи для автоматизованого контролю стану компонентів автомобіля, що дозволить підвищити безпеку його експлуатації та оптимізувати витрати на обслуговування.

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

У розділі I слід надати загальну характеристику галузі технічного обслуговування та діагностики автомобілів, із акцентом на сучасні методи контролю стану основних комплектуючих транспортного засобу. Необхідно визначити об'єкт дослідження – інформаційну систему моніторингу стану комплектуючих автомобіля – та структуру управління процесом збору та обробки технічних даних. Особливу увагу слід приділити ключовим проблемам, які виникають під час використання традиційних методів діагностики, включаючи низьку оперативність реагування, обмежену достовірність та людський фактор. Доцільно навести схему організаційної структури функціонування інформаційної системи, описати ролі основних користувачів та модулів системи.

У розділі II необхідно провести аналіз потреб користувачів – як водіїв, так і технічного персоналу – на основі чого сформулювати функціональні та нефункціональні вимоги до інформаційної системи. Важливо виконати постановку задачі з урахуванням техніко-економічної доцільності впровадження системи контролю, розробити алгоритм її реалізації та побудувати блок-схему логіки роботи системи. Слід побудувати математичну модель обробки вхідних показників із метою виявлення аномалій та прийняття рішень щодо технічного стану. Окремо необхідно представити інформаційну модель системи у вигляді UML-діаграм: прецедентів, класів, компонентів, послідовностей, а також побудувати матрицю трасування вимог. Також потрібно описати вхідні та вихідні дані, зокрема показники температури, обертів, напруги, швидкості тощо, вказати джерела даних, вимоги до точності, спосіб передачі та збереження.

У розділі III виконати проектування компонентів інформаційної системи: інформаційного, технічного та програмного забезпечення відповідно до вимог методичних рекомендацій. Представити реалізовані основні функції системи: збір технічних показників із використанням емуляції інтерфейсу OBD-II, передача даних на сервер, збереження у базі даних MySQL, обробка та виявлення відхилень від норми, формування звітів і графіків, передача сповіщень на мобільний застосунок, реалізований на Flutter.

**Завдання підготував
науковий керівник**

Тішков Богдан Олександрович

« 10 » березня 2025 р.

**Завдання одержав
здобувач**

Франчук Микола Костянтинівич

« 10 » березня 2025р

Відгук
про кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Франчука Миколи Костянтиновича

на тему
«Проектування інформаційної системи з контролю стану комплектуючих автомобіля»

Кваліфікаційна бакалаврська робота Франчука М.К., присвячена проектуванню та розробці інформаційної системи для автоматизованого контролю стану компонентів автомобіля, що дозволить підвищити безпеку його експлуатації та оптимізувати витрати на обслуговування.

В роботі проведений аналіз системних вимог та обґрунтування методів проектування системи оцінювання знань. На основі узагальнень обґрунтовані пропозиції та проектні рішення для системи моніторингу програмного забезпечення та мережевого обладнання.

Постановка задач виконана у відповідності з вимогами стандартів та методичних вказівок, алгоритм вирішення задачі розроблено з використанням сучасних технологій.

У роботі розглянуті питання інформаційного, технічного, програмного та організаційного забезпечення інформаційної системи.

При виконанні кваліфікаційної бакалаврської роботи студент Франчук М.К. показав високий рівень знань, уміння узагальнювати практичний досвід використання сучасних ІТ-технологій, самостійно впроваджувати в практику проектні рішення.

В цілому, кваліфікаційна бакалаврська робота студента Франчука М.К. відповідає сучасним вимогам, заслуговує позитивної оцінки і може бути допущений до захисту в екзаменаційній комісії з атестації здобувачів вищої освіти.

Науковий керівник

(підпис)

доцент, к.е.н. Тішков Б.О.

“ ___ ” _____ 20__ р.

Рецензія
на кваліфікаційну бакалаврську роботу
здобувача вищої освіти
Франчука Миколи Костянтиновича

тема

**«Проектування інформаційної системи з контролю стану
комплектуючих автомобіля»**

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення: Тема кваліфікаційного бакалаврського проєкту є актуальною, оскільки сучасні автомобілі є складними технічними об'єктами, і забезпечення своєчасного контролю за їхнім станом є ключовим фактором надійності та безпеки експлуатації. Враховуючи, що навіть незначні недоліки можуть призвести до серйозних наслідків, проектування та розробка інформаційної системи для контролю стану комплектуючих є доцільним та важливим завданням, спрямованим на зменшення витрат на ремонт та підвищення безпеки дорожнього руху.

Якість проведеного дослідження: Дослідження виконано на високому рівні, що підтверджується логічною структурою роботи та глибиною опрацювання матеріалу. Автор провів детальний аналіз предметної галузі та існуючих рішень, виявивши їхні ключові обмеження, зокрема залежність від хмарних сервісів та високу вартість. На основі цього було чітко сформульовано функціональні та нефункціональні вимоги до системи. Проектування архітектури виконано з використанням сучасних підходів моделювання UML та SysML.

Позитивні риси кваліфікаційної бакалаврської роботи:

Микола Костянтинович в своїй роботі проявив такі риси:

1. Комплексний підхід та практична реалізація: Робота представляє собою завершене програмне рішення. Було не тільки спроектовано систему, а й реалізовано функціональний прототип, що складається з серверної частини на Python та клієнтського мобільного додатку на Flutter.
2. Автономність та безпека: Ключовою перевагою розробленої системи є її повна автономність. Обробка та зберігання даних відбуваються локально на Raspberry Pi, що не потребує підключення до Інтернету, підвищує конфіденційність та захищає від зовнішніх кіберзагроз.
3. Доступність та масштабованість: Архітектура побудована на відкритих та недорогих програмних і апаратних засобах, що робить рішення доступним для широкого кола користувачів. Модульна


структура програмного забезпечення створює умови для подальшого розширення функціоналу.


4. Функціональність: Система забезпечує повний цикл роботи з даними: збір через OBD-II, зберігання, аналіз, виявлення аномалій, а також зручну візуалізацію поточних та архівних показників у вигляді показників, графіків та звітів.

Зауваження: На поточному етапі аналітичний модуль системи виконує виявлення аномалій шляхом порівняння показників з попередньо заданими нормативними значеннями. Для подальшого вдосконалення варто розглянути можливість інтеграції алгоритмів машинного навчання для реалізації функцій прогнозу аналітики, що дозволить системі передбачати потенційні несправності до їх виникнення.

Практична значимість висновків і рекомендацій: Робота має високу практичну значущість. Створена інформаційна система може бути використаний власниками автомобілів для підвищення безпеки експлуатації та оптимізації витрат на технічне обслуговування. Система дозволяє своєчасно виявляти несправності, що допомагає уникнути дорогих ремонтів та аварійних ситуацій на дорозі. Запропоновані рекомендації, щодо розширення сумісності з різними діагностичними протоколами, та інтеграції алгоритмів машинного навчання, є слушними і відкривають чіткі шляхи для подальшого розвитку проєкту до рівня конкурентоспроможного комерційного продукту.

Кафедри інформаційно вимірювальних технологій КПІ ім. Ігоря Сікорського,
Доцент, кандидат технічних наук


Козир Олег Васильович
(підпис, ПІБ)

Підпис засвідчую: 
(посада, підпис)

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
Франчука Миколи Костянтиновича, виконаної на тему:
«Проектування інформаційної системи з контролю стану комплектуючих
автомобіля»
Київ: кафедра інформаційних систем в економіці, 2025 р.

Кваліфікаційна бакалаврська робота присвячена актуальній проблемі підвищення безпеки та ефективності експлуатації автомобілів шляхом автоматизованого моніторингу їхнього технічного стану. У роботі спроектовано та реалізовано інформаційну систему, що дозволяє збирати дані з діагностичного інтерфейсу OBD-II, локально їх обробляти, аналізувати та візуалізувати для користувача. Робота складається з трьох логічно пов'язаних розділів.

У першому розділі виконано аналіз предметної галузі, розглянуто існуючі комерційні та телематичні системи для діагностики автомобілів, виявлено їхні переваги та недоліки.

Другий розділ присвячений проєктній частині: сформульовано функціональні та нефункціональні вимоги до системи, виконано постановку задачі та розроблено моделі поведінки й архітектури ІС з використанням мов UML та SysML.

У третьому розділі описано реалізацію компонентів системи: інформаційного забезпечення на базі СКБД MySQL, технічного забезпечення з використанням мікрокомп'ютера Raspberry Pi та програмного забезпечення, що включає серверну частину на Python і мобільний застосунок на Flutter. Наведено результати апробації системи, що підтверджують її працездатність.

У висновках підсумовано результати роботи та надано рекомендації щодо практичного застосування розробленої системи автовласниками для підвищення безпеки експлуатації та оптимізації витрат на технічне обслуговування.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 119 сторінок, 18 таблиць, 56 рисунків, список використаних джерел з 20 найменувань, 6 додатків.

«ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З КОНТРОЛЮ СТАНУ КОМПЛЕКТУЮЧИХ АВТОМОБІЛЯ»

Перелік ключових слів: інформаційна система, діагностика автомобіля, автомобіль, поточні показники, датчики, OBD-II, Raspberry Pi, мобільний додаток, Python, Flutter, аналітика, архівні показники.

Об'єкт дослідження: процес контролю технічного стану комплектуючих автомобіля.

Предмет дослідження: інформаційна система для моніторингу, аналізу та візуалізації даних про стан компонентів автомобіля, що базується на використанні технології OBD-II та апаратної платформи Raspberry Pi.

Метою роботи є проєктування та розробка інформаційної системи для автоматизованого контролю стану компонентів автомобіля, що дозволить підвищити безпеку його експлуатації та оптимізувати витрати на обслуговування.

Відповідно до поставленої мети визначені такі завдання: проаналізувати предметну галузь та існуючі програмні рішення для діагностики автомобілів; сформулювати та специфікувати функціональні й нефункціональні вимоги до системи; розробити архітектуру інформаційної системи, включаючи моделі її поведінки та структури; спроектувати інформаційне, технічне та програмне забезпечення; реалізувати ключові компоненти системи: серверну частину на базі Raspberry Pi та мобільний

клієнтський додаток; провести тестування розробленої системи на імітаційному прикладі.

Теоретична значущість полягає в узагальненні підходів до створення автономних систем моніторингу технічного стану транспортних засобів.

Методична цінність роботи полягає в розробці комплексної методики проектування подібних систем, що охоплює моделювання вимог, проектування архітектури, розробку бази даних та програмних модулів.

Практична значущість полягає у створенні функціонального прототипу інформаційної системи, що є доступним, автономним та безпечним рішенням.

Система може бути використана автовласниками для отримання оперативної інформації про стан комплектуючих автомобіля, своєчасного виявлення несправностей та планування технічного обслуговування.

Рік виконання кваліфікаційної бакалаврської роботи – 2025.

Рік захисту роботи – 2025.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП	13
РОЗДІЛ 1	16
ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	16
1.1 Характеристика предметної галузі та об'єкта дослідження.....	16
1.2 Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	23
РОЗДІЛ 2	28
РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	28
2.1 Аналіз і специфікація вимог до інформаційної системи	28
2.2 Постановка та алгоритм розв'язання задачі.....	35
2.2.1 Постановка задачі	35
2.2.2 Алгоритм розв'язання задачі	41
2.3 Моделювання інформаційної системи.....	44
2.3.1 Моделювання поведінки системи	44
2.3.2 Моделювання структури системи	51
2.3.3 Розподіл вимог за компонентами системи	54
РОЗДІЛ 3	57
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ.....	57
3.1 Інформаційне забезпечення	57
3.1.1 Загальна характеристика інформаційного забезпечення.....	57
3.1.2 Організація збору і передавання первинної інформації.....	61

3.1.3 Побудова системи класифікації та кодування	62
3.1.4 Проектування форм первинних документів та відеокадрів ..	64
3.1.5 Структура інформаційних масивів.....	66
3.1.6 Вибір СКБД	70
3.1.7 Інфологічна модель бази (сховища) даних	72
3.1.8 Даталогічна модель бази (сховища)даних	74
3.2 Технічне забезпечення.....	76
3.2.1 Загальні положення та схема автоматизації	76
3.2.2 Структура комплексу технічних засобів	78
3.2.3 Опис автоматизованого робочого місця.....	79
3.2.4 Схема мережі передачі даних	80
3.3 Програмне забезпечення	82
3.3.1 Структура програмного забезпечення	82
3.3.2 Системне програмне забезпечення	84
3.3.3 Прикладне програмне забезпечення	86
3.3.4 Програмна документація.....	87
3.4 Результати реалізації інформаційної системи	89
ВИСНОВКИ.....	101
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	104
ДОДАТКИ.....	107
Додаток А.....	107
Додаток Б	109
Додаток В.....	112
Додаток Г	116
Додаток Д.....	118
Додаток Е	119

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

DTC – diagnostic trouble codes

OBD-II – On-Board Diagnostics II

БД – База Даних

ІЗ – інформаційне забезпечення

ІК – інтерфейс користувача

ІС – інформаційна система

ІС – Інформаційна Система

МБ – Мобільний додаток

СТО – Станція технічного обслуговування

ВСТУП

Автомобільна промисловість є однією з найважливіших та найдинамічніших галузей сучасної економіки, що суттєво впливає на розвиток суміжних галузей, впровадження інноваційних технологій та загальний соціально-економічний розвиток суспільства. Сучасний автомобіль - це складний технічний об'єкт, що складається з безлічі взаємопов'язаних компонентів та систем, які забезпечують його ефективну, безпечну та комфортну експлуатацію. Забезпечення постійного та своєчасного контролю за технічним станом автомобіля є ключовим фактором його надійності, безпеки на дорозі, екологічності та економічної ефективності. Навіть незначний збій у роботі одного з компонентів може призвести до серйозних наслідків, таких як вихід з ладу транспортного засобу, зниження рівня безпеки, збільшення витрат пального та підвищення рівня шкідливих викидів.

У зв'язку з цим, проектування ІС з контролю стану комплектуючих автомобіля набуває особливої актуальності. Така система дозволяє здійснювати безперервний моніторинг роботи ключових вузлів автомобіля, своєчасно виявляти потенційні несправності та надавати користувачеві рекомендації щодо обслуговування. Це сприяє підвищенню безпеки дорожнього руху, зменшенню витрат на ремонти та технічне обслуговування, а також забезпеченню комфорту для водія та пасажирів.

Огляд літературних джерел показує, що існуючі системи моніторингу, такі як Torque, Car Scanner та OBD Auto Doctor, переважно обмежуються відображенням поточних даних та кодів помилок без глибокого аналізу та формування персоналізованих рекомендацій. Професійні діагностичні комплекси, як-от Autel MaxiSys та Launch X431, хоч і мають розширений функціонал, проте є дорогими та орієнтованими на автосервіси. У

той же час, телематичні системи від автовиробників, наприклад OnStar, BMW Connected Drive та Mercedes me, доступні лише для нових автомобілів певних марок і залежать від зовнішніх серверів. Наукові дослідження підкреслюють перспективність застосування алгоритмів штучного інтелекту для прогнозування технічного стану, однак більшість з цих розробок залишаються на рівні прототипів. Таким чином, існує потреба у створенні доступної, автономної та функціональної системи, яка б поєднувала переваги існуючих рішень та усувала їхні недоліки.

Об'єктом дослідження є процес контролю технічного стану комплектуючих автомобіля.

Предметом дослідження є інформаційна система, що забезпечує збір, обробку, аналіз та візуалізацію даних про стан компонентів автомобіля за допомогою використання інтерфейсу OBD-II та мікрокомп'ютера Raspberry Pi.

Метою кваліфікаційної бакалаврської роботи є проектування та розробка інформаційної системи для моніторингу та контролю стану комплектуючих автомобіля, яка забезпечує збір даних у реальному часі, їх аналіз, виявлення аномалій та надання рекомендацій користувачеві.

Для досягнення поставленої мети були визначені наступні завдання:

Проаналізувати предметну галузь, дослідити існуючі рішення для моніторингу стану автомобілів та визначити їхні переваги та недоліки.

Сформулювати функціональні та нефункціональні вимоги до проєктованої інформаційної системи.

Розробити архітектуру інформаційної системи, включаючи моделі поведінки та структури.

Спроекувати інформаційне, технічне та програмне забезпечення системи.

Реалізувати ключові компоненти системи, зокрема сервер на базі Raspberry Pi та клієнтський мобільний додаток.

Провести тестування розробленої системи на імітаційному прикладі для підтвердження її працездатності.

Методи дослідження, що використовувалися в роботі, включають системний аналіз для визначення структури та взаємозв'язків компонентів, об'єктно-орієнтоване моделювання за допомогою UML та SysML для проєктування архітектури, а також методи математичного аналізу для обробки даних та виявлення відхилень.

Теоретична значущість роботи полягає в узагальненні підходів до створення автономних систем моніторингу технічного стану автомобілів та розробці архітектури, що поєднує доступність апаратних засобів з функціональністю аналітичної обробки даних.

Практична значущість роботи полягає у створенні функціонального прототипу інформаційної системи, який може бути використаний власниками автомобілів для підвищення безпеки експлуатації та оптимізації витрат на технічне обслуговування. Розроблена система є доступною, автономною та може бути адаптована для широкого кола транспортних засобів, оснащених інтерфейсом OBD-II.

Структура роботи складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі наведено аналіз предметної області та існуючих рішень. Другий розділ присвячено розробці вимог та моделюванню інформаційної системи. У третьому розділі детально описано проєктування та реалізацію компонентів системи. У висновках підсумовано результати роботи.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Характеристика предметної галузі та об'єкта дослідження

Інформаційна система - це сукупність елементів, які працюють разом для збору, зберігання, обробки та передачі інформації. Інформаційні системи включають різні елементи, такі як програмне забезпечення, обладнання, бази даних та комунікаційні мережі [1].

Основна мета інформаційних систем - забезпечити швидкий та ефективний обмін інформацією між різними користувачами та рівнями системи. Інформаційні системи використовуються в багатьох дисциплінах, включаючи бізнес, науку, технології та освіту.

Інформаційні системи можуть бути розроблені для різних цілей, таких як автоматизація бізнес-процесів, управління ресурсами, обробка даних та інформації, розробка ігор, створення мультимедіа та аналіз даних [1].

Інформаційна система орієнтована на покращення контролю стану автомобіля. Завданням системи є збір, зберігання та аналізі отриманих даних про стан комплектуючих автомобіля.

Автомобільна промисловість є однією з найбільш динамічних та найважливіших галузей сучасної економіки, що впливає на розвиток суміжних галузей, інноваційних технологій та соціально-економічний розвиток суспільства [2]. Автомобіль є складним технічним об'єктом, який поєднує в собі купу різноманітних компонентів та систем, що взаємодіють між собою з метою забезпечення ефективної, безпечної та комфортної експлуатації. Забезпечення своєчасного контролю технічного стану

автомобіля є запорукою його надійності, безпеки руху, екологічності та економічної ефективності використання транспортного засобу [2].

Сучасні автомобілі містять чисельні системи та компоненти, що відповідають за керування його комплектуючими, а саме :двигуном, трансмісією, гальмівною системою, підвіскою, клімат-контролем та системами безпеки. Враховуючи складність конструкції автомобіля, навіть незначний збій у роботі окремого компонента може призвести до серйозних наслідків, включаючи вихід з ладу транспортного засобу, зниження безпеки руху, збільшення витрат пального або підвищення рівня шкідливих викидів в атмосферу [2].

Через це актуальності набуває проєктування ІС з контролю стану комплектуючих автомобіля. Така система дозволить здійснювати постійний моніторинг роботи основних вузлів і комплектуючих автомобіля, вчасно виявляти потенційні несправності, надавати користувачу рекомендації щодо обслуговування транспортного засобу, а також формувати звіти про стан комплектуючих. Проєктування подібної системи є актуальною для підвищення безпеки дорожнього руху, зменшення витрат на ремонт і технічне обслуговування, а також для забезпечення комфорту користувачів.

Автомобіль складається з ряду компонентів, кожен з яких виконує певні функції, необхідні для забезпечення ефективної роботи, комфорту пасажирів, дотримання екологічних норм і стандартів безпеки [2].

До основних систем автомобіля, стан яких потребує контролю, можна віднести:

1. двигун внутрішнього згоряння (або електродвигун) – забезпечує обертальний момент для пересування автомобіля;
2. трансмісія – передає крутний момент від двигуна до коліс;
3. гальмівна система – відповідає за уповільнення та зупинку транспортного засобу;

4. підвіска – забезпечує з'єднання коліс з кузовом автомобіля та комфорт руху;
5. системи живлення – відповідають за подачу пального, мастил, охолоджувальної рідини;
6. системи безпеки – подушки безпеки, ремені, електроніка безпеки;
7. системи клімат-контролю та мультимедіа – для комфорту пасажирів.

Контроль технічного стану автомобіля є важливою складовою безпечної експлуатації автомобіля. Несвоєчасне виявлення несправностей може призвести до серйозних наслідків котрі:

1. виникнення аварійних ситуацій на дорозі;
2. підвищення витрат на ремонт та обслуговування;
3. зниження ефективності роботи окремих комплектуючих та систем автомобіля.

У сучасних автомобілях передбачені системи самодіагностики, однак вони обмежені у своїх можливостях і, як правило, зосереджені на фіксації критичних помилок. Більшість штатних систем не дозволяють водієві здійснювати детальний моніторинг стану комплектуючих або отримувати прогнозовані без відвідування СТО [2]. Це обумовлює необхідність проєктування інформаційної системи з контролю стану компонентів, котра б забезпечувала:

1. збір даних про поточний стан комплектуючих автомобіля;
2. зберігання архівних даних для подальшого аналізу;
3. виявлення відхилень та аномалій у роботі компонентів;
4. формування рекомендацій для користувача;
5. надання сповіщень про виявленні несправності та рекомендації для їх вирішення.

Для реалізації ІС з контролю технічного стану автомобіля важливою складовою є наявність технологій, що дозволяють отримувати доступ до

даних автомобільних систем у стандартизованому форматі. Основним джерелом даних у більшості сучасних автомобілів є діагностичний інтерфейс OBD-II, який дозволяє зчитувати інформацію про стан систем автомобіля за допомогою зовнішніх пристроїв [3].

Інтерфейс OBD-II був стандартизований на законодавчому рівні в США у 1996 році, а згодом впроваджений і в інших країнах [4]. Його основна мета полягає у забезпеченні доступу до інформації про роботу двигуна, системи контролю викидів, паливної системи, а також до діагностичних кодів помилок DTC Серед основних даних, доступних через OBD-II:

1. оберти двигуна;
2. температура охолоджувальної рідини;
3. напруга акумулятора;
4. швидкість руху;
5. споживання пального;
6. коди помилок і їх опис.

OBD-II використовує стандартизовані протоколи передачі даних, серед яких найпоширенішими є SAE J1850, ISO 9141-2, ISO 14230-4 (KWP2000), ISO 15765-4 (CAN). Зчитування даних з автомобіля здійснюється за допомогою спеціальних пристроїв, які підключаються до діагностичного роз'єму [5].

У проєктованій інформаційній системі збір даних з OBD-II здійснюється за допомогою мікрокомп'ютера Raspberry Pi. Raspberry Pi - це доступний, компактний і енергоефективний комп'ютер, який завдяки своїм можливостям широко застосовується в проєктах автоматизації, моніторингу та обробки даних. У цьому проєкті Raspberry Pi виконує роль центрального вузла збору даних, аналізу та взаємодії з користувачем [6].

Raspberry Pi може підключатись до порту OBD-II через сумісний адаптер для отримання поточних даних з автомобіля. Надалі ці дані

обробляються за допомогою програмного модуля аналізу, що працює на самому Raspberry Pi. Завдяки обчислювальним можливостям Raspberry Pi, на ньому можливо реалізувати алгоритми обробки даних, виявлення аномалій, побудови графіків, формування звітів та рекомендацій для користувача.

ІС з контролю стану комплектуючих автомобіля, котра була спроектована в даному проєкті, може складатися з кількох функціональних модулів, що взаємодіють між собою:

1. модуль збору даних - відповідає за підключення до інтерфейсу OBD-II автомобіля та отримання даних у режимі реального часу - цей модуль забезпечує обробку вхідних сигналів, переведення їх у зрозумілий формат та передачу для подальшої обробки;
2. модуль аналізу даних - реалізований на основі базових алгоритмів. виконує обробку зібраних даних, виявлення відхилень від нормальних значень, розрахунків прогнозованих показників, виявлення аномалій та формування рекомендацій для користувача;
3. база даних - локальна база, розташована на Raspberry Pi, що зберігає поточні та історичні дані про роботу систем автомобіля - вона містить інформацію про показники комплектуючих, коди помилок, дати виявлення несправностей, тощо;
4. мобільний додаток - ІК що дозволяє отримувати доступ до даних із Raspberry Pi через Wi-Fi. додаток дозволяє переглядати показники у режимі реального часу, аналізувати історичні дані, отримувати сповіщення та рекомендації, а також генерувати звіти про роботу ІС;
5. Wi-Fi-з'єднання - канал бездротового обміну даними між Raspberry Pi та мобільним додатком. забезпечує зручність

використання системи та мінімізує потребу у додатковому обладнанні.

Взаємодія модулів між собою забезпечує функціонування ІС від збору даних до надання користувачу готової аналітичної інформації та звітності.

ІС з контролю стану комплектуючих автомобіля працює з різними видами даних, що дозволяє формувати повноцінну картину про технічний стан транспортного засобу. Залежно від джерела, призначення та періодичності оновлення, дані, які обробляє система, можна класифікувати на такі категорії:

1. Поточні дані - показники, що надходять у реальному часі від датчиків та електронних блоків управління автомобіля. Це значення температури охолоджувальної рідини, тиску масла, напруги акумулятора, обертів двигуна, швидкості руху, витрати пального та інших параметрів. Поточні дані дозволяють оцінювати стан комплектуючих на даний момент та приймати рішення про необхідність негайного втручання;
2. Архівні дані - показники, що накопичуються в базі даних під час роботи системи. Вони включають значення параметрів за попередні дні, тижні або місяці, що дозволяє аналізувати тенденції, виявляти закономірності та формувати прогнози щодо стану комплектуючих. Зберігання історичних даних є важливим для формування звітів, побудови графіків та оцінки ефективності технічного обслуговування;
3. Аналітичні дані - результати обробки поточних та історичних даних модулем штучного інтелекту. До них належать виявлені аномалії, тенденції змін показників, прогнози щодо можливих відмов, а також рекомендації для користувача. Аналітичні дані є основою для прийняття рішень щодо технічного обслуговування автомобіля;

4. Звіти та рекомендації - сформовані системою документи або повідомлення, що містять узагальнену інформацію про стан комплектуючих автомобіля, виявлені проблеми, поради щодо подальших дій, необхідність технічного обслуговування або ремонту. Звіти можуть формуватися автоматично за певний період часу або за запитом користувача;
5. Помилки - спеціальні діагностичні коди, які генеруються системами автомобіля при виявленні несправностей. Кожен код відповідає певній проблемі. Коди помилок є основою для формування сповіщень користувачеві та ініціювання відповідних рекомендацій щодо ремонту або діагностики.

Ключовим учасниками взаємодії з ІС є – користувачі, власник або водій автомобіля, який отримує доступ до даних через мобільний додаток.

Основні функції, що доступні користувачеві:

1. перегляд поточних показників про стан комплектуючих;
2. ознайомлення з архівними показниками у вигляді графіків;
3. отримання сповіщень про виявлені помилки, аномалії, відхилення від норми;
4. формування запитів на створення нових звітів або отримання додаткових рекомендацій;
5. реагування на повідомлення системи щодо необхідності технічного обслуговування або ремонту автомобіля;
6. користувач також може використовувати мобільний додаток для управління режимами роботи системи та налаштовувати підключення до Raspberry Pi.

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

У сучасних умовах автомобільна галузь активно інтегрує інформаційні технології для забезпечення надійності, безпеки та комфорту експлуатації транспортних засобів. Значна увага приділяється розвитку систем моніторингу технічного стану автомобілів, які дозволяють виявляти несправності, попереджати аварійні ситуації та оптимізувати процес обслуговування автомобіля [2].

До сучасних підходів з моніторингу та контролю за станом транспортних засобів можна віднести:

1. використання вбудованих діагностичних систем;
2. розроблені мобільні застосунки для аналізу даних з автомобіля;
3. використання спеціалізованих діагностичних комплексів для професійного обслуговування.

В дослідженні [7] розглядаються різноманітні аспекти створення інформаційних систем для контролю стану транспортних засобів. Зокрема, дослідники наголошують на важливості використання датчиків та електронних блоків управління для отримання даних про роботу двигуна, трансмісії, гальмівної системи, а також на необхідності ефективного аналізу цих даних для виявлення аномалій.

На даний момент існує великий спектр комерційних рішень для моніторингу технічного стану автомобіля, які можна умовно розділити на дві категорії:

1. Мобільні додатки для роботи з OBD-II адаптерами:
 - а) Torque - один із найпопулярніших додатків, що дозволяє підключитися до автомобіля через Bluetooth-адаптер OBD-II (зазвичай на базі ELM327) і переглядати основні параметри:

оберти двигуна, швидкість, температуру, коди помилок. Однак функціонал обмежується відображенням даних та базовою візуалізацією без глибокого аналізу чи формування рекомендацій [8];

- b) Car Scanner - схожий додаток із розширеним інтерфейсом, підтримкою читання кодів помилок, перегляду живих даних і побудови графіків. Також доступний перегляд описів помилок, проте відсутня функція прогнозування або рекомендацій [9];
- c) OBD Auto Doctor - дозволяє моніторити основні показники автомобіля, зчитувати помилки, але не підтримує інтелектуальний аналіз або накопичення історії даних [10].

2. Системи для професійної діагностики такі як, Autel MaxiSys, Launch X431, Bosch KTS – це комплексні прилади та системи для автосервісів, які дозволяють проводити повну діагностику автомобілів, зчитувати дані з більшості блоків управління.

Проаналізувавши функціонал існуючих рішень можна виявити недоліки, не забезпечують формування рекомендацій для користувача та не дозволяють зберігати архівні дані для подальшого аналізу.

У науковому дослідженні [11] підкреслюється перспективність застосування алгоритмів штучного інтелекту та машинного навчання для обробки даних з автомобіля. Зокрема, досліджуються методи прогнозування технічного стану вузлів автомобіля на основі аналізу змін параметрів у часі, виявлення аномалій за допомогою кластерного аналізу, побудови моделей деградації компонентів на основі історичних даних.

Більшість описаних підходів залишаються на рівні дослідних прототипів і не реалізовані в комерційних продуктах для кінцевих користувачів.

Окремий напрям розвитку систем контролю технічного стану автомобіля пов'язаний із використанням концепцій Інтернету речей (IoT) та телематики. IoT передбачає підключення автомобіля до глобальної мережі Інтернет і обмін даними з хмарними сервісами, іншими транспортними засобами, інфраструктурою. У цьому контексті автомобіль розглядається як частина розумної екосистеми, яка постійно обмінюється інформацією для підвищення ефективності та безпеки дорожнього руху [12].

Такі системи з використанням телеметричних рішень дозволяють здійснювати:

1. віддалений моніторинг стану автомобіля;
2. виявлення аварійних ситуацій і передавання даних до сервісного центру або екстрених служб;
3. інтеграцію з системами управління автопарком для логістики та комерційного використання.

Існуючі вже системи:

1. OnStar - комплексна телематична система, яка забезпечує функції безпеки, віддаленої діагностики, виклику допомоги, моніторингу технічного стану [13];
2. BMW Connected Drive - платформа, що дозволяє власникам bmw отримувати дані про автомобіль, планувати обслуговування, отримувати сповіщення про технічні проблеми [14];
3. Mercedes me - сервіс, що надає користувачу інформацію про технічний стан автомобіля, включаючи залишок пального, тиск у шинах, нагадування про техобслуговування [15].

Проте такі рішення мають кілька суттєвих обмежень, а саме вони орієнтовані переважно на нові автомобілі певних брендів і є недоступними для широкого кола користувачів та не дають змоги отримувати дані безпосередньо без зовнішніх серверів, що обмежує можливість автономної роботи.

Проектована ІС з контролю стану комплектуючих автомобіля має низку важливих переваг порівняно з існуючими рішеннями:

1. доступність: система побудована на відкритих апаратних і програмних засобах а саме: raspberry pi, зв'язок через Wi-Fi, база даних mysql та мови програмування python та dart, що знижує вартість впровадження та дозволяє використовувати її для більшого кола автомобілістів;
2. автономність: немає потреби у підключенні до інтернету або використанні зовнішніх серверів. дані зберігаються локально на raspberry pi, що підвищує безпеку та конфіденційність;
3. функціонал звітності та історії даних: користувач отримує можливість переглядати історичні дані, будувати графіки, формувати звіти, що недоступно в більшості бюджетних рішень;
4. можливість розширення: завдяки відкритій архітектурі, система може бути модифікована та адаптована під різні моделі автомобілів і потреби користувачів.

Велика різноманітність виробників автомобілів і електронних систем призводить до того, що протоколи діагностики, формат даних та інтерфейси взаємодії суттєво відрізняються. Це ускладнює створення універсальних систем моніторингу, які можуть ефективно працювати з автомобілями різних марок і моделей без необхідності індивідуального налаштування.

Із ростом кількості підключених автомобілів з'являються нові загрози, пов'язані з кібербезпекою. Зловмисники можуть отримати доступ до систем автомобіля, порушити їх роботу або викрасти персональні дані користувачів. Тому актуальним є впровадження захищених протоколів передачі даних, аутентифікації, а також локальне зберігання даних без залучення хмарних сервісів, що значно підвищує рівень безпеки.

Сучасні інформаційні системи все частіше розглядаються як частина більшої екосистеми розумного міста або автопарку. Віддалений моніторинг,

автоматичне замовлення запчастин, взаємодія з сервісами технічного обслуговування.

Перспективним є розвиток адаптивних систем, які підлаштовуються під індивідуальні особливості конкретного автомобіля і стилю водіння власника. Створення систем з використанням машинного навчання дозволить підвищити точність виявлення несправностей і скоротити кількість помилкових сповіщень, що підвищить довіру користувачів і ефективність системи [11].

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз і специфікація вимог до інформаційної системи

Процес проєктування ІС з контролю стану комплектуючих автомобіля починається з етапу визначення вимог до її функціонування, структури та взаємодії з користувачем. Ці вимоги охоплюють не лише функціональність, яку має забезпечувати система, а й характеристики, що стосуються її надійності, продуктивності, безпеки та зручності використання. Формування вимог є критично важливим для створення життєздатного технічного рішення, яке відповідатиме реальним потребам користувачів і буде придатним для використання для контролю компонентів транспортного засобу.

Для формування вимог до системи потрібно проаналізувати бізнес цілі та на їх основі сформувані вимоги до ІС з контролю стану комплектуючих автомобіля.

Внутрішні стейкхолдери - розробники, зацікавлені у створенні популярного продукту, котрий буде відповідати всім вимогам ІС. Бізнес цілі розробників включають забезпечення швидкої, стабільної та безперебійної роботи платформи [16].

Керівництво підприємства керує впровадженням системи, розглядає її як інструмент для збільшення безпеки на дорогах за рахунок вчасного ремонту транспортних засобів та збільшення кількості відвідувань їх сервісних центрів для встановлення ІС з контролю стану комплектуючих автомобіля та обслуговування автомобілів котрі потребують ремонту, до

ключових завдань відноситься забезпечення клієнтської бази що дозволить підприємству просунути продукт на ринку для забезпечення збільшення обсягів продажів системи.

Фахівці технічної підтримки відповідальні за забезпечення швидкої та якісної технічної підтримки при виникненні запитання та проблем при роботі з системою.

Зовнішні стейкхолдери, автовиробники та постачальники запчастин зацікавлені в такій системі через збільшення попиту на свої товари. Сервісні центри обслуговувань зацікавлені у збільшенні кількості відвідувачів для надання послуг з обслуговування автомобілів.

Ключовою вимогою користувачів є інтуїтивно зрозумілий інтерфейс з великою кількістю інформації про стан комплектуючих автомобіля для зручного контролю за їхнім станом, за рахунок чого можливе значне зменшення витрат на обслуговування автомобіля.

Для всіх стейкхолдерів розробляються бізнес-вимоги, які враховують потреби всіх зацікавлених в ІС сторін (рис. 2.1) [17].

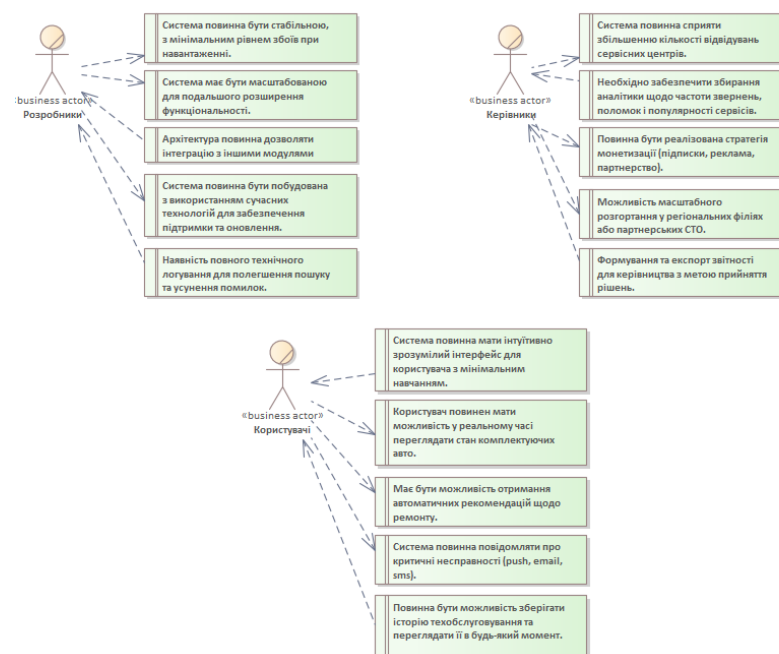


Рисунок 2.1 – Діаграма бізнес вимог

Функціональні вимоги до ІС з контролю стану комплектуючих автомобіля деталізують її основні компоненти та ключові функції, яким система повинна відповідати щоб забезпечити зручної, ефективної роботи для забезпечення контролю стану комплектуючих автомобіля [10].

ІС, що проєктується, покликана отримувати технічні дані з електронних блоків керування автомобілем, використовуючи загальноприйнятий інтерфейс OBD-II. Ці дані включають в себе оберти двигуна, температуру охолоджуючої рідини, швидкість руху, напругу акумулятора, положення дросельної заслінки, витрату пального, а також коди помилок, що виникають у випадку порушення нормального функціонування компонентів автомобіля. Отримання цих параметрів має відбуватися в режимі реального часу, що забезпечує оперативне реагування на зміни в стані компонентів та своєчасне інформування водія про потенційні проблеми.

Всі зібрані дані будуть локально зберігатися на платі комп'ютері Raspberry Pi, використовуючи структуровану базу даних. Це дозволить не лише спостерігати за показниками в реальному часі, а й аналізувати динаміку змін цих параметрів. Такий підхід буде використовуватись для забезпечення аналітичного модуля, що зможе виявляти аномалії, передбачати технічні несправності та генерувати звіти про стан автомобіля на основі накопичених даних. Передбачено, що обробка інформації буде здійснюватися безпосередньо на Raspberry Pi, що забезпечує автономність та незалежність системи від зовнішніх серверів та інтернет-з'єднання.

Взаємодія користувача з системою буде відбуватися через мобільний додаток, встановлений на смартфоні. Додаток надасть доступ до актуальної інформації про стан автомобіля, архівних даних, звітів та діагностичних кодів помилок. Передбачена також функція побудови графіків, які дозволять водієві оперативно оцінювати зміни технічних параметрів з часом. Важливим компонентом є система сповіщень, яка буде надавати

своєчасну інформацію про критичні помилки або відхилення від нормальних показників, а також надаватиме текстові поради щодо подальших дій.

Особливу увагу приділено забезпеченню автономної роботи системи. Вона має функціонувати в повністю локальному середовищі, не потребуючи інтернет-з'єднання та хмарних сервісів. Передача даних між Raspberry Pi та мобільним додатком буде здійснюватися за допомогою бездротового інтерфейсу Wi-Fi, що забезпечує простоту підключення та достатню швидкість обміну даними. Це рішення також сприяє збереженню конфіденційності інформації про технічний стан автомобіля.

Проаналізувавши ці вимоги можна прийти до висновку що проєктована система включає такі компоненти функціональних вимог:

1. Підсистема «Збору Даних» відповідає за автоматичне опитування вбудованих датчиків транспортного засобу через інтерфейс OBD-II, з метою забезпечення системи актуальними параметрами стану комплектуючих. Вона забезпечує безперервний збір технічних даних у режимі реального часу, перевірку їхньої цілісності та передачу до наступних модулів для подальшої обробки.
2. Підсистема «Візуалізація та Інтерфейс Користувача» відповідає за забезпечення зручного доступу користувачів до зібраної та обробленої інформації про стан комплектуючих автомобіля. Вона включає в себе графічне представлення поточних параметрів, відображення попереджень про потенційні несправності, навігацію по функціях системи.
3. Підсистема «Звітності» відповідає за формування деталізованих звітів на основі проаналізованих даних. Система генерує рекомендації щодо необхідного технічного обслуговування. Звіти можуть бути створені автоматично або за запитом користувача.

4. Підсистема «Аналізу Даних» є важливою частиною системи, тому що реалізує аналітичний аналіз отриманих показників. Вона забезпечує виявлення відхилень від базових технічних показників комплектуючих, ідентифікацію потенційних несправностей, зіставлення поточних даних з еталонними значеннями, а також формування рекомендацій щодо необхідності обслуговування та створює сповіщення у разі виявлення аномальних даних.

Для демонстрації функціональних вимог ІС було використано середовище Enterprise Architect, в якому було створено функціональні вимоги (рис. 2.2) та їх специфікацію вимог з використанням функціоналу Specification Manager (рис. 2.3-2.6)

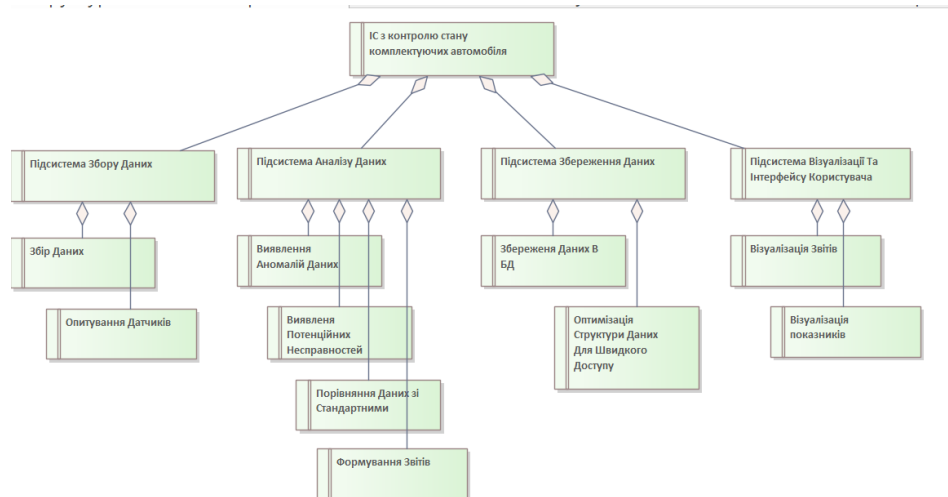


Рисунок 2.2 – Діаграма функціональних вимог

Item	Stereotype	Status	Difficulty	Priority
<input checked="" type="checkbox"/> IC з контролю стану комплектуючих автомобіля	FunctionalRequir...	Proposed	High	Medium
<input checked="" type="checkbox"/> Підсистема Аналізу Даних	FunctionalRequir...	Proposed	High	High
<input checked="" type="checkbox"/> Виявлення Аномалій Даних	FunctionalRequir...	Proposed	Medium	High
<input checked="" type="checkbox"/> Виявлення Потенційних Несправностей	FunctionalRequir...	Proposed	Medium	High
<input checked="" type="checkbox"/> Порівняння Даних зі Стандартними	FunctionalRequir...	Proposed	Low	High
<input checked="" type="checkbox"/> Формування Звітів	FunctionalRequir...	Proposed	Low	Medium

Рисунок 2.3 – Специфікація підсистеми аналізу даних

<input checked="" type="checkbox"/> Підсистема Візуалізації Та Інтерфейсу Користувача	FunctionalRequir...	Proposed	High	Low
<input checked="" type="checkbox"/> Візуалізація Звітів	FunctionalRequir...	Proposed	Low	Medium
<input checked="" type="checkbox"/> Візуалізація показників	FunctionalRequir...	Proposed	Medium	High
<input checked="" type="checkbox"/> Мобільний застосунок	FunctionalRequir...	Proposed	Medium	High

Рисунок 2.4 – Специфікація підсистеми візуалізація та ІК

<input checked="" type="checkbox"/> Підсистема Збереження Даних	FunctionalRequir...	Proposed	High	High
<input checked="" type="checkbox"/> Збереження Даних В БД	FunctionalRequir...	Proposed	Low	Medium
<input checked="" type="checkbox"/> Оптимізація Структури Даних Для Швидкого Доступу	FunctionalRequir...	Proposed	Medium	High

Рисунок 2.5 – Специфікація підсистеми збереження даних

<input checked="" type="checkbox"/> Підсистема Збору Даних	FunctionalRequir...	Proposed	High	High
<input checked="" type="checkbox"/> Збір Даних	FunctionalRequir...	Proposed	High	High
<input checked="" type="checkbox"/> Опитування Датчиків	FunctionalRequir...	Proposed	Medium	High

Рисунок 2.6 – Специфікація підсистеми збору даних

Система повинна бути надійною та спроможною безперервно функціонувати тривалий час без збоїв. Вона має бути стійкою до помилок, таких як нестабільність живлення, втрата Wi-Fi-з'єднання чи помилки при зчитуванні даних з автомобіля. Для цього необхідно передбачити механізми автоматичного відновлення роботи, архівації даних та запису подій, що забезпечить збереження цілісності інформації.

Також критично важливими є характеристики масштабованості та адаптивності. Система має легко адаптуватися до різних моделей автомобілів, мати можливість підключення додаткових датчиків та розширення функціоналу без значної зміни архітектури. Програмна реалізація має бути модульною, щоб спростити впровадження нових алгоритмів аналізу або змін в інтерфейсі взаємодії з користувачем.

Щодо продуктивності, важливо, щоб система могла обробляти інформацію в режимі реального часу або з мінімальною затримкою. Враховуючи обмежені обчислювальні ресурси платформи Raspberry Pi, необхідна оптимізація алгоритмів, ефективна організація зберігання даних

та раціональне управління ресурсами. Пріоритетом є баланс між точністю аналізу, швидкістю обробки та енергоефективністю системи.

Тому для ІС з контролю стану комплектуючих автомобіля було розроблено такі нефункціональні вимоги (рис. 2.7):

1. Безпека – система повинна забезпечити безпечну передачу даних на всіх етапах роботи інформаційної системи;
2. Доступність – Система повинна підтримуватись різними пристроями з різними версіями;
3. Підтримка – Система повинна мати постійну технічну підтримку користувачів та інструкції для її використання;
4. Юзабіліті – Система повинна мати зручний та зрозумілий інтерфейс для користувачів;
5. Швидкодія – Система швидко виконувати аналіз даних для забезпечення максимальної швидкодії.

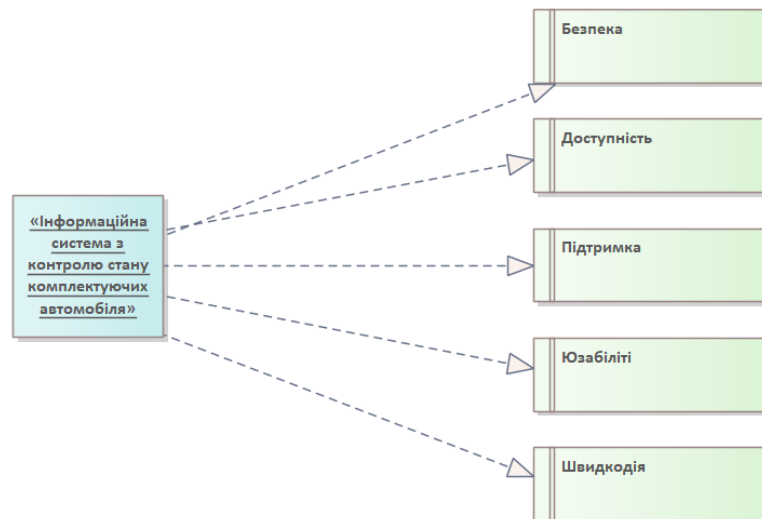


Рисунок 2.7 – Діаграма нефункціональних вимог

2.2 Постановка та алгоритм розв'язання задачі

2.2.1 Постановка задачі

Проектування ІС з контролюючих автомобіля передбачає створення автоматизованої системи, яка буде збирати, зберігати, обробляти та демонструвати інформацію про технічний стан транспортного засобу в реальному часі. Головна мета цієї задачі - збільшити ефективність контролю, зменшити витрати на обслуговування та підвищити безпеку експлуатації автомобіля, шляхом забезпечення користувача своєчасною, змістовною та актуальною інформацією про стан роботи основних компонентів автомобіля.

Задача полягає у забезпеченні постійного моніторингу показників роботи ключових компонентів автомобіля, зокрема двигуна, трансмісії, гальмівної системи та системи живлення. Система дозволяє виявляти відхилення в роботі систем автомобіля, формувати сповіщення про несправності, а також надавати рекомендації щодо обслуговування. Економічна доцільність задачі полягає в зменшенні витрат за ремонт через непередбачувані несправності, уникненні аварійних ситуацій та оптимізації графіку технічного обслуговування транспортного засобу.

Об'єктами управління в рамках цієї задачі є технічні параметри автомобіля, що надходять з діагностичного інтерфейсу OBD-II, а саме: оберти двигуна, температура охолоджуючої рідини, напруга акумулятора, тиск у шинах, положення дросельної заслінки, швидкість автомобіля, діагностичні коди несправностей (DTC), а також похідні показники, які обчислюються на їх основі.

Вихідна інформація, що генерується системою, призначена для користувача ІС - водія чи власника автомобіля. Вона використовується для прийняття рішень щодо подальшої експлуатації транспортного засобу, планування візитів до сервісного центру, а також як джерело даних для технічної підтримки або аналізу технічного стану автомобіля. Відображення інформації відбувається у вигляді текстових сповіщень, попереджень, графіків змін параметрів, рекомендацій, а також звітів, які формуватимуться на вимогу користувача.

ІС працюватиме в режимі реального часу: дані зчитуватимуться та оброблятимуться постійно під час роботи автомобіля, а видача інформації здійснюється з періодичністю кілька разів на секунду (для змінних параметрів) або при певних ситуаціях (виявлення помилки, досягнення критичного значення тощо). Формування узагальнених звітів відбуватиметься на вимогу користувача.

Автоматизована робота ІС припиняється у випадках втрати зв'язку з автомобілем, вимкнення живлення Raspberry Pi або при виникненні критичних помилок програмного забезпечення. При цьому система повинна зберігати попередньо зібрані дані для подальшого аналізу після відновлення роботи.

Взаємодія користувачів з ІС - користувач ініціює запуск системи, підключення до мобільного застосунку та запити на перегляд даних, вся обробка, фільтрація, зберігання та генерація рекомендацій виконуватимуться автоматично.

Інформаційна модель задачі містить логічні зв'язки між джерелами вхідної інформації (датчиками та інтерфейсом OBD-II), центральною обчислювальною частиною (Raspberry Pi), базою даних, аналітичним модулем та користувацьким інтерфейсом. Дані надходять з автомобіля, обробляються, зберігаються, аналізуються та передаються мобільному застосунку для демонстрації.

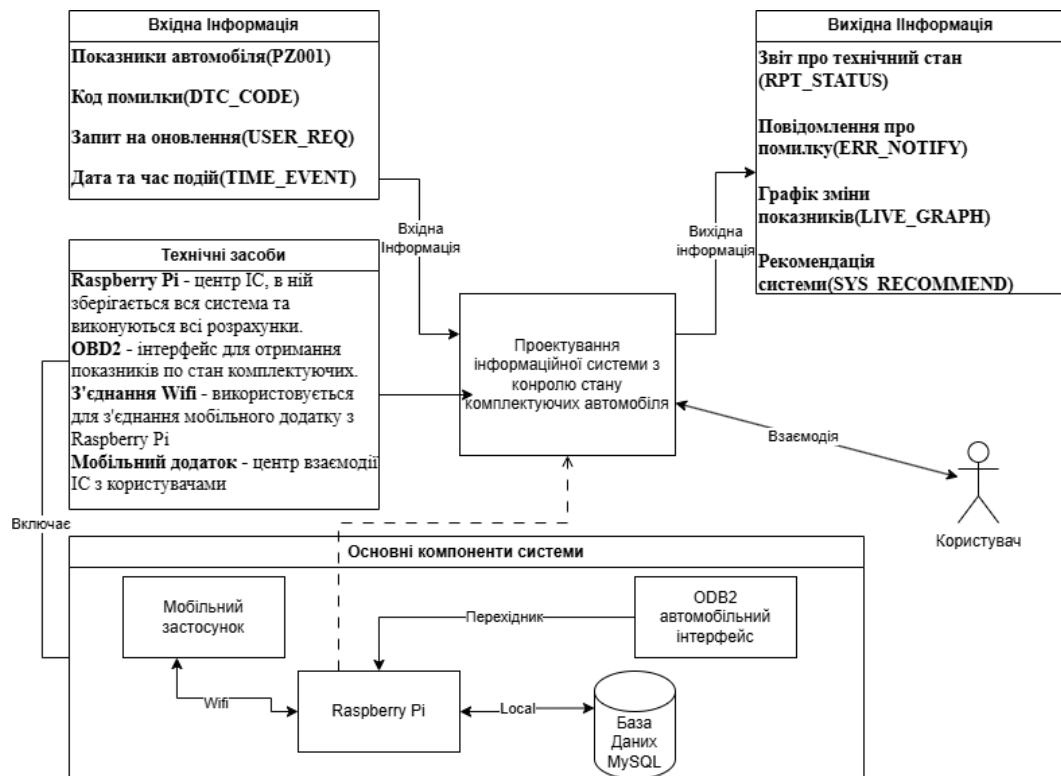


Рисунок 2.8 – Інформаційна модель задачі ІС з контролю стану комплектуючих автомобіля

Вихідна інформація є кінцевим результатом обробки даних ІС. Вона використовується користувачем для особистої оцінки стану комплектуючих, визначення наявності несправностей, планування обслуговування та ухвалення рішень щодо подальшої експлуатації транспортного засобу.

До основних типів вихідної інформації належать:

1. повідомлення про виявлені помилки та аномалії;
2. рекомендації щодо дій у відповідь на критичні значення параметрів;
3. графіки зміни показників у часі;
4. звіти про технічний стан автомобіля.

Таблиця 2.1 – Перелік вихідних повідомлень та їх опис

№	Назва повідомлення	Ідентифікатор	Форма подання	Періодичність	Термін видачі	Користувачі
1	Звіт про технічний стан	RPT_STATUS	екран	За запитом	Протягом 1 с	Користувач,
2	Повідомлення про помилку	ERR_NOTIFY	push-повідомлення	При виявленні	Миттєво	Користувач
3	Графік зміни показників	LIVE_GRAPH	графік	У режимі реального часу	-	Користувач
4	Рекомендація системи	SYS_RECOMMEND	текст	За запитом / подією	Протягом 2 с	Користувач

Джерело: сформовано автором на основі виконаного дослідження

Звіт про технічний стан автомобіля (RPT_STATUS) є агрегованим повідомленням, що містить поточні показники основних параметрів, збережені у базі даних, а також інформацію про виявлені відхилення, якщо такі були виявленні. У складі цього звіту подаються значення температури двигуна, обертів, напруги акумулятора, швидкості тощо. Також міститься перелік виявлених помилок та опис відповідних рекомендацій. Звіт формується за запитом користувача. Призначений для оцінки технічного стану автомобіля та прийняття рішень щодо його подальшої експлуатації.

Повідомлення про помилку (ERR_NOTIFY) генерується автоматично у випадку, коли система виявляє відхилення одного або кількох показників від допустимих меж. Наприклад, перевищення температури двигуна або зниження напруги акумулятора нижче встановленого порогу. Таке повідомлення містить назву параметра, зафіксоване значення, граничне значення, а також опис можливої причини відхилення. Надсилається користувачеві миттєво у вигляді сповіщення в мобільному застосунку.

Графік зміни показників (LIVE_GRAPH) відображає динаміку змін одного технічного показнику у режимі реального часу. Будується на основі

збережених у базі даних значень з часовими мітками. Використовується для візуального аналізу стабільності роботи окремих систем автомобіля та контролю навантаження під час руху.

Рекомендація системи (SYS_RECOMMEND) формується на основі аналізу даних модулем аналізу. Містить опис дій, які рекомендується виконати користувачу: наприклад, звернутись до сервісу, перевірити рівень охолоджувальної рідини або провести діагностику певної системи. Рекомендації зберігаються у звітах та відображаються при перегляді історії показників.

Вхідна інформація - це первинні дані, які надходять до системи від автомобіля про стан його комплектуючих та від користувачів. Вона використовується для формування інформаційної бази та подальшого аналітичного аналізу. Основні джерела вхідної інформації - інтерфейс OBD-II та запити користувача.

Таблиця 2.2 – Перелік вхідних повідомлень з їх описом

№	Назва повідомлення	Ідентифікатор	Форма подання	Частота надходження	Джерело
1	Показники автомобіля	PZ001	числове значення	Щосекундно	OBD-II
2	Код помилки	DTC_CODE	буквено-цифрове	За подією	OBD-II
3	Запит на оновлення	USER_REQ	команда з мобільного застосунку	За запитом	Користувач
4	Дата та час події	TIME_EVENT	datetime	При кожному записі	Raspberry Pi clock

Джерело: сформовано автором на основі виконаного дослідження

Показники автомобіля (PZ001) - це набір значень ключових параметрів, що характеризують роботу основних компонентів автомобіля, таких як система охолодження, електроживлення, трансмісія, гальмівна

система, ходова частина. Показники надходять із бортової системи автомобіля через інтерфейс OBD-II, обробляються у системі та виводяться у мобільному застосунку. До таких параметрів належать, зокрема, температура охолоджувальної рідини, напруга акумулятора, стан датчиків ABS, навантаження двигуна. Ця інформація дозволяє користувачу здійснювати оперативний контроль технічного стану транспортного засобу в реальному часі.

Код помилки (DTC_CODE) - буквено-цифрове повідомлення, яке надходить у разі виявлення електронною системою управління автомобілем несправностей. Кожен код має уніфіковану структуру (наприклад, P0420, P0301), яка дозволяє ідентифікувати характер помилки. Система зчитує ці коди через OBD-II, аналізує їх, зберігає у базі даних і відображає користувачу у звітах або повідомленнях. Використовується для діагностики та інформування про несправності в системах автомобіля.

Запит на оновлення (USER_REQ) є керуючим повідомленням, яке надходить до системи від користувача через мобільний застосунок. Такі запити запускають певні дії: оновлення поточних показників, побудова нового графіка, формування звіту або перевірка наявності помилок. Інформація подається у вигляді команди в текстовому або структурованому форматі.

Дата та час події (TIME_EVENT) - службове повідомлення, яке фіксує момент надходження кожного вхідного параметра. Отримується від системного годинника Raspberry Pi у форматі datetime (рік, місяць, день, година, хвилина, секунда). Використовується для хронологічного маркування усіх записів у базі даних, синхронізації графіків і створення логів подій у системі.

2.2.2 Алгоритм розв'язання задачі

Алгоритм розв'язання завдання з контролю стану комплектуючих автомобіля використовує як поточні, так і архівні дані, що надходять від систем автомобіля через інтерфейс OBD-II. Призначення використаної інформації полягає в забезпеченні системи актуальними технічними параметрами, які в подальшому аналізуються для виявлення відхилень та використовуються для формування рекомендацій та створення звітів. Щоб забезпечити функціонування ІС для реалізації алгоритму роботи можна використати для роботи такі масиви даних(див. таб.2.3)

Таблиця 2.3 – Перелік масивів використовуваної інформації

№	Масив	Ідентифікатор	Максимальна кількість записів
1	Поточні параметри з OBD-II	M_OBD	100 000
2	Коди помилок DTC	M_DTC	10 000
3	Команди користувача	M_USER	5 000
4	Історичні параметри	M_HISTORY	500 000
5	Дані про відхилення та сповіщення	M_ALERTS	20 000

Джерело: сформовано автором на основі виконаного дослідження

Результатом виконання алгоритму є набір діагностичних висновків, повідомлень а також рекомендацій, сформованих на основі аналізу параметрів та виявлених аномалій. Ці результати призначені для виведення на інтерфейс мобільного застосунку, а також для архівації у базі даних з метою подальшого перегляду та аналізу.

Таблиця 2.4 – Перелік масивів результатної інформації

№	Масив	Ідентифікатор	Максимальна кількість записів
1	Звіт про технічний стан автомобіля	R_STATUS	50 000
2	Рекомендації системи	R_RECOMMEND	10 000
3	Графіки змін параметрів	R_GRAPH	100 000
4	Повідомлення про виявлені аномалії	R_ANOMALY	20 000

Джерело: сформовано автором на основі виконаного дослідження

Математична модель задачі полягає в аналізі динамічних технічних параметрів автомобіля, з подальшим виявленням їх виходу за допустимі межі. Оцінювання критичних ситуацій базується на порівнянні з пороговими значеннями, визначеними у технічній документації виробника.

Стан критичності автомобіля - з початку маємо закладені розробником константи показників поведінки автомобіля, в кінцевому результаті відбувається порівняння отриманих даних розрахованих за математичним апаратом з показником автовиробника, розраховується за формулою:

(2.1)

$$f = \frac{E_m \times \left(\frac{S}{1000}\right)}{t_p \times B}$$

де E_m – це середній показник використання мотору в обертах;

S – відстань котру пройшов автомобіль(відстань у кілометрах);

t_p – час роботи паливного насосу(час в хвилинах);

B – кількість палива що використав автомобіль(паливо у літрах) – це загальний показник використаного палива;

F – середній показник нормалі що може бути встановлений на рівні 0,1 з граничним відхиленням 0,09 в обидві сторони.

Якщо показник f має відхилення від середнього значення 0,09 в обидві сторони то система зберігає інформацію в базу даних про неправильне використання автомобіля та надає сповіщення про відхилення та його критичний.

Діаграма станів моделює процес роботи ІС з контролю стану комплектуючих автомобіля(рис 2.9). Взаємодія користувача з системою:

Початок роботи: користувач запускає мобільний додаток;

Отримання запиту від користувача: користувач надсилає запит до системи для отримання показників;

Збір: система отримує показники з датчиків - поточні та від системи; керування баз даних – попередні;

Збереження: система зберігає дані про поточні показники;

Аналіз: система аналізує отримані показники:

1. якщо показники в нормі починає створювати звіт;
2. якщо показники мають відхилення створює звіт та сповіщення про відхилення.

Створення: система створює звіт про показники;

Демонстрація: система демонструє користувачеві показники;

Запитання: Перегляд наступного показника:

1. якщо так то система починає аналізувати інший показник;
2. якщо ні, користувач виходить із додатку.

Кінець: користувач виходить з мобільного додатку.

Діаграма взаємодії – демонструє взаємодію користувача з системою

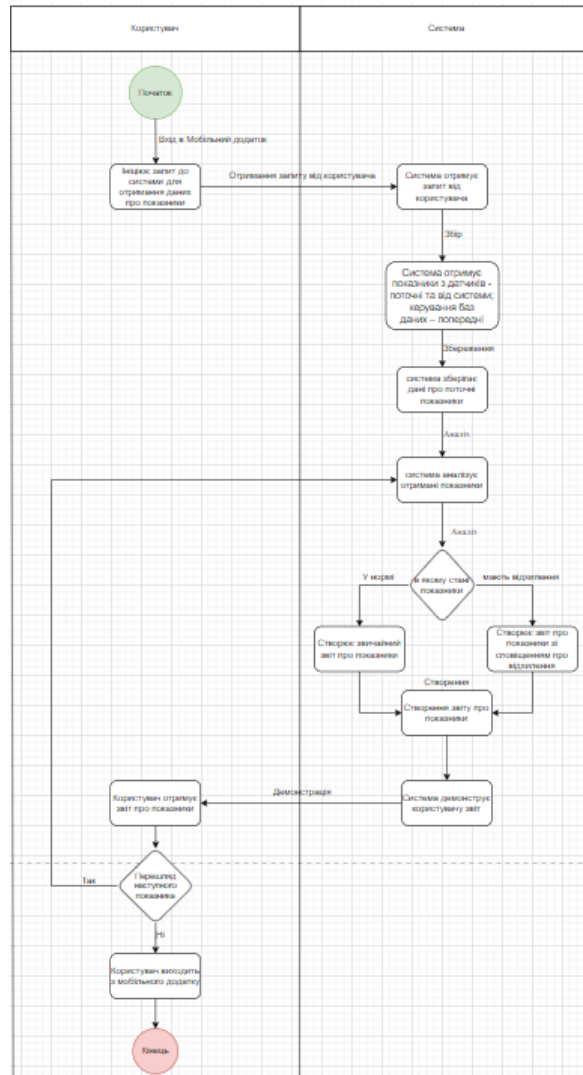


Рисунок 2.9 – UML діаграма станів та дій

2.3 Моделювання інформаційної системи

2.3.1 Моделювання поведінки системи

Розроблена прецедентів для ІС з контролю стану комплектуючих автомобіля відображає взаємодію користувачів та внутрішніх компонентів

системи з її функціональними можливостями. В ролі акторів виступають користувач, модуль збору та модуль аналізу даних.

Діаграма охоплює такі функціональні блоки як: збір даних про комплектуючі, збереження даних, аналіз даних з виявленням недоліків, створення сповіщення, створення звітів та візуалізація інформації.

В межах функціоналу збору інформації, модуль збору даних з'єднаний з системою автомобіля через інтерфейс OBD-II, отримує дані про показники всіх підсистем транспортного засобу та передає отримані показники до модулю аналізу даних а також, записує отримані показники до бази даних.

Модуль аналізу даних, отримує дані від модулю збору даних та виконує їх порівняння з базовим показниками котрі були отримані від виробника, при виявленні відхилень від цих даних система створює сповіщення про помилку в роботі комплектуючих автомобіля після чого формує звіт про помилку в якому описує яка підсистема автомобіля та в який період часу працювала некоректно.

Користувацький інтерфейс отримує поточні показники від модуля збору даних та архівні показники від бази даних для демонстрації стану комплектуючих автомобіля з можливістю побудови графіків, отримання сповіщень та перегляду звітів про стану комплектуючих автомобіля.

Мобільний додаток має такий функціонал: демонстрація поточних та архівних показників комплектуючих автомобіля за проміжок часу, будувати графік показників, переглядати звіти з інформацією про недоліки та проблеми в роботі комплектуючих автомобіля та отримувати сповіщення у разі виявлення таких проблем у поточний час.

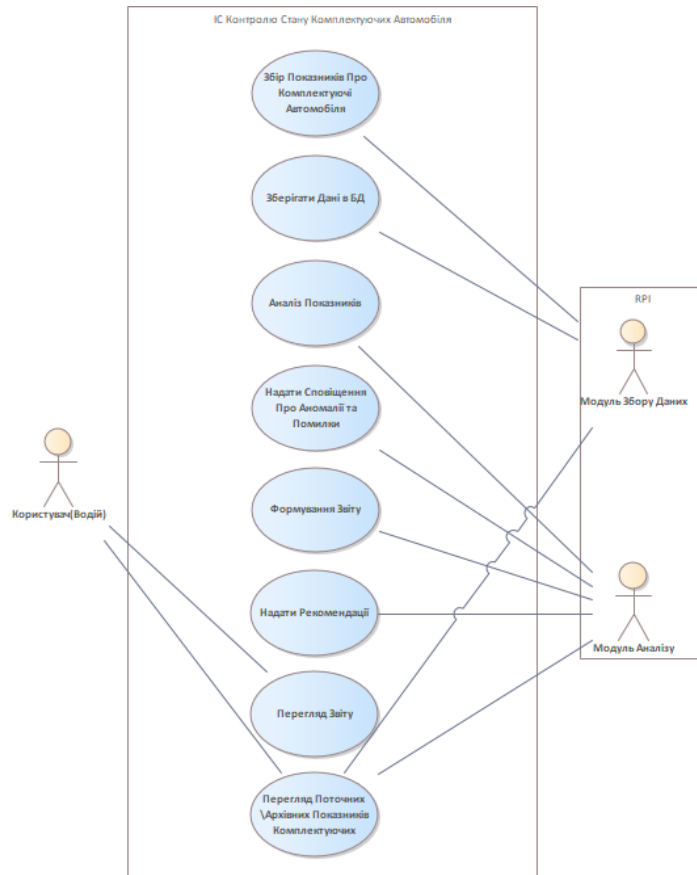


Рисунок 2.10 – Діаграма прецедентів

Також була заповнена матриця відповідності вимог і прецедентів (рис. 2.11).

Source	Use case: Аналіз Показників	Use case: Зберігати Дані в БД	Use case: Збір Показників Про Комплексуєчих Автомобіля	Use case: Надати Рекомендації	Use case: Надати Сповіщення Про Аномалії та Помилки	Use case: Перегляд (Поточних/Архівних) Показників	Use case: Перегляд Звіту	Use case: Формування Звіту
Функціональні вимоги: Доступ до Бази Даних	↑							
Функціональні вимоги: Доступ до Систем Автомобіля		↑						
Функціональні вимоги: Функція Аналізу Даних	↑		↑	↑	↑			↑
Функціональні вимоги: Функція Збереження Даних		↑						
Функціональні вимоги: Функція Надання Звітності				↑				↑
Функціональні вимоги: Функція перегляду даних					↑	↑		

Рисунок 2.11 – Матриця відповідності вимог

Діаграми послідовності - використовуються для моделювання логіки сценаріїв використання ІС, демонструючи інформацію, що передається між об'єктами в системі під час використання ІС. Було створено 8 діаграм послідовності для відображення внутрішньої роботи всіх функціональних можливостей (рис. 2.10) ІС з контролю стану комплектуючих автомобіля (рис. 2.12 - 2.19).

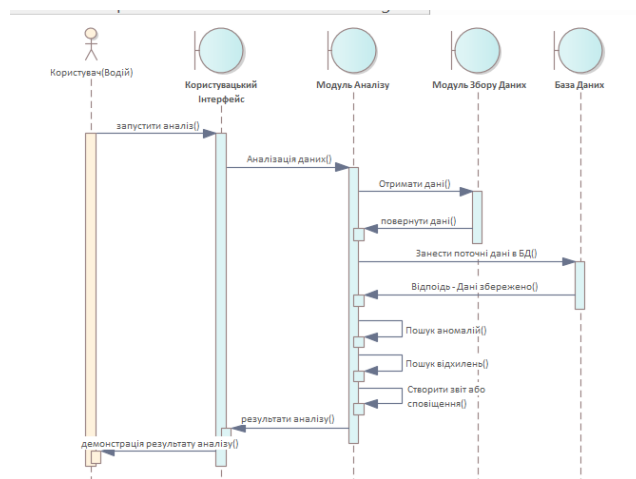


Рисунок 2.12 – Діаграма послідовності процесу «Аналіз показників»

Ця діаграма послідовності (рис. 2.12) візуалізує кроки взаємодії, що відбуваються при виконанні прецеденту "Аналіз Показників". Вона демонструє, як користувач системи, взаємодіє з програмними компонентами для аналізу даних про стан компонентів автомобіля. Учасниками цієї взаємодії є користувач, користувацький інтерфейс, модуль аналізу, модуль збору даних, та база даних.

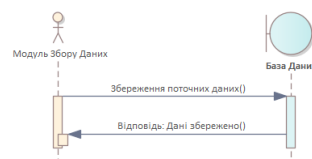


Рисунок 2.13 – Діаграма послідовності процесу «Збереження даних в базу даних»

Ця діаграма послідовності (рис. 2.13) демонструє взаємодію між компонентами, модуль збору даних та базою даних під час процесу збереження інформації.

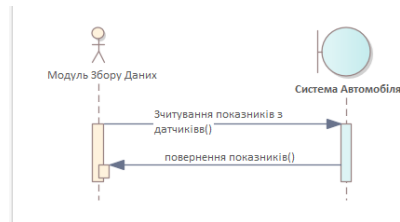


Рисунок 2.14 – Діаграма послідовності процесу «Збір Даних Про Комплектуючі»

Діаграма послідовності (рис. 2.14) демонструє процес отримання даних системою від автомобіля про стан його комплектуючих.

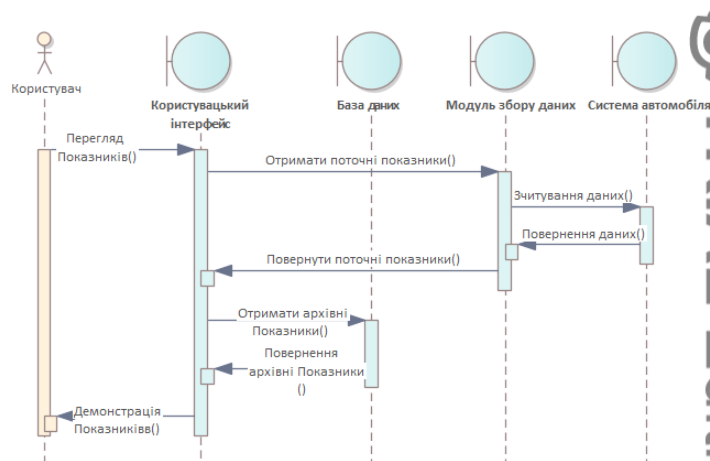


Рисунок 2.15 – Діаграма послідовності процесу «Перегляд поточних/архівних показників»

Ця діаграма (рис. 2.15) відображає процес, за допомогою якого користувач системи може отримати та переглянути як поточні, так і архівні дані про стан комплектуючих автомобіля. Учасниками є користувач,

користувацький інтерфейс, модуль збору даних, система автомобіля (для поточних даних) та база даних (для архівних даних).

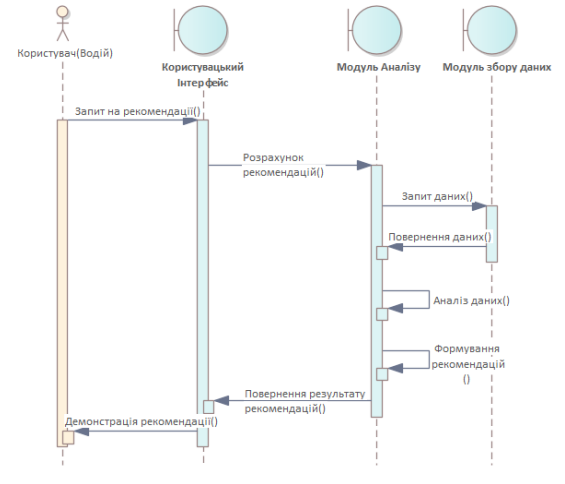


Рисунок 2.16 – Діаграма послідовності процесу «Надати рекомендації»

Ця діаграма послідовності (рис. 2.16) демонструє, як користувач отримує рекомендації щодо стану комплектуючих автомобіля. Учасниками є користувач, користувацький інтерфейс, модуль аналізу та модуль збору даних.

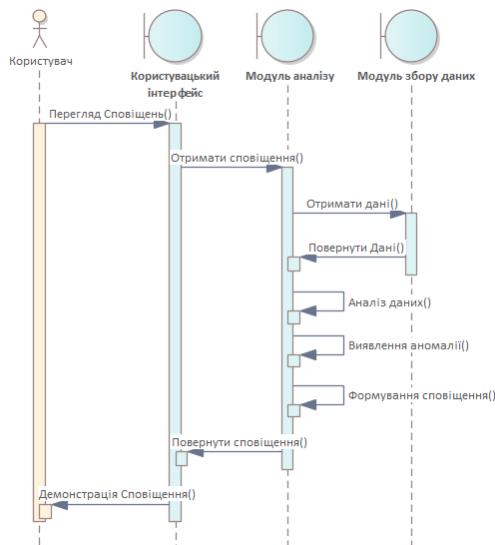


Рисунок 2.17 – Діаграма послідовності процесу «Надати сповіщення про аномалії»

Ця діаграма(див. рис. 3.8.) відображає процес інформування користувача про виявлені несправності у стані комплектуючих автомобіля. Вона демонструє взаємодію між користувачем, користувацьким інтерфейсом, модулем аналізу та модулем збору даних.

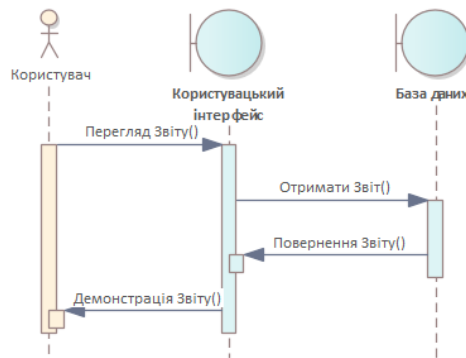


Рисунок 2.18 – Діаграма послідовності процесу «Перегляд звіту»

Ця діаграма (рис. 2.18) відображає процес, за допомогою якого користувач може переглянути звіт про стан комплектуючих автомобіля. Учасниками цієї взаємодії є користувач, користувацький інтерфейс та база даних.

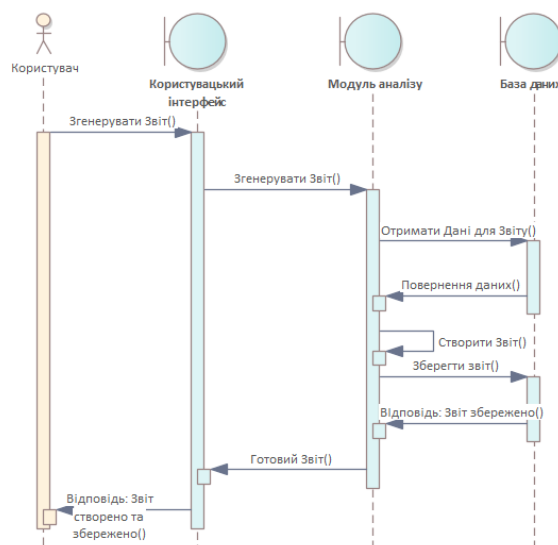


Рисунок 2.19– Діаграма послідовності процесу «Формування Звіту»

Ця діаграма послідовності (рис. 2.19) демонструє процес створення та збереження звіту до бази даних. Учасниками взаємодії є користувач, користувацький інтерфейс, модуль аналізу та база даних.

2.3.2 Моделювання структури системи

Діаграму класів було змодельовано у середовищі enterprise architect (рис. 2.20) вона відображає структуру логіки основних компонентів та сутностей ІС з контролю стану комплектуючих автомобіля. Класи були змодельовані та згруповані на діаграмі відповідно до патерну boundary-control-entity, щоб продемонструвати взаємозв'язки в ІС [18].

Класи-сутності, що моделюють основні дані ІС:

1. параметри датчиків - зберігає інформацію про вимірювання, отримані від компонентів автомобіля;
2. результат аналізу - моделює результати перевірки комплектуючих;
3. користувач - описує користувача системи;
4. модуль збору даних - відповідає за збір показників з датчиків: атрибути;
5. модуль аналізу - реалізує функціональність обробки даних.

Граничні класи що реалізують взаємодію з користувачем:

1. інтерфейс аналізу - надає змогу формувати звіти та рекомендації, операції що виконує клас: надати рекомендації(), сформувати звіт();
2. інтерфейс підключення - відповідає за створення нових з'єднань з джерелами даних, операції що виконує клас: перевірити з'єднання(), створити нове з'єднання();

3. інтерфейс рекомендацій - інтерфейс генерації та надання рекомендацій, операції що виконує клас: надати сповіщення(), формувати звіт();
4. інтерфейс перегляду показників - дозволяє переглядати результати аналізу, операції що виконує клас: перегляд архівних показників(), передача даних(), перегляд поточних показників().

Класи управління – контролери:

1. контролер аналізу - головний контролер логіки аналізу даних, операції що виконує контролер: згенерувати звіт(), формувати рекомендації(), створити сповіщення(), запустити аналіз(), створення графіку();
2. контролер даних - відповідає за обробку вхідних даних, операції що виконує контролер: перевірка даних(), порівняння з нормальними();
3. контролер експорту -забезпечує вивантаження звітів, операції що виконує контролер: згенерувати звіт(), створити звіт(), записати до бази даних().

Також на діаграмі продемонстровано основні зв'язки між класами що демонструє їх взаємодію (рис. 2.20).

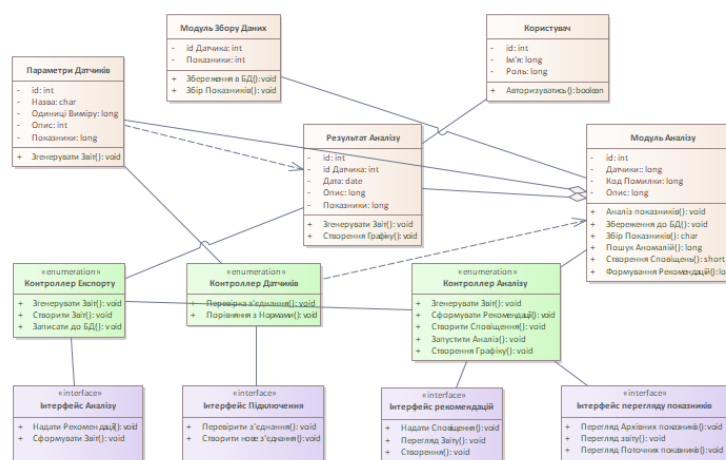


Рисунок 2.20 – Діаграма класів ІС з контролю стану комплектуючих автомобіля

Для ІС з контролю стану комплектуючих автомобіля, була змодельована діаграма визначення блоків (рис. 2.21) котра візуалізує основні структурні компоненти системи та їхню ієрархію.

Центральним елементом діаграми є блок ІС з контролю комплектуючих автомобіля. Цей блок представляє всю систему як єдине ціле і показує її складові частини з яких він складається.

Цими частинами є: програмне забезпечення, технічне забезпечення та база даних. Кожен з цих великих блоків, у свою чергу розділяється, на менші компоненти підсистем.

Для взаємодії з користувачами, головний блок системи має визначені порти: Wi-Fiport, fileport, sensorport та uipor.

Блок програмне забезпечення об'єднує всі програмні компоненти системи а саму: модуль збору даних, модуль аналізу та інтерфейс користувача (мобільний додаток).

Блок технічне забезпечення описує апаратну частину ІС, на якій вона функціонує. Та складається з: система автомобіля, датчики, raspberrу рі, ODB2 connector та Wi-Fi модуль.

Блок база даних відповідає за збереження всієї інформації інформаційної системи. Та складається з таких частин: сховище архівних даних, сховище інформації про аномалії даних, сховище помилок системи та сховище звітів.

Таким чином, ця діаграма визначення блоків демонструє системний погляд на архітектуру ІС з контролю стану комплектуючих автомобіля, розкриваючи її підсистеми котрі в свою чергу розкриваються на компоненти, а також ієрархічну залежність підсистем та компонентів між собою.

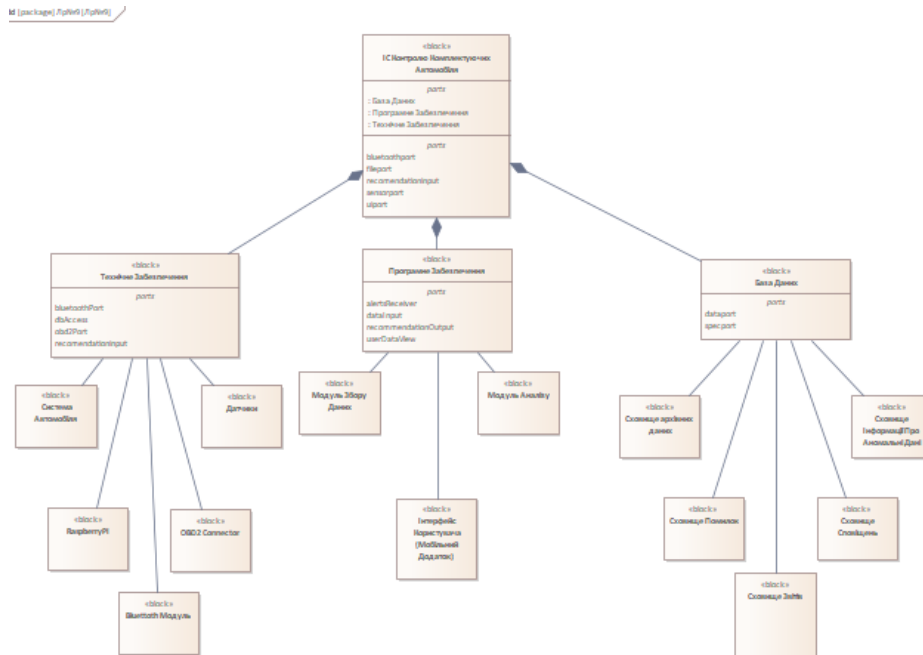


Рисунок 2.21 – Діаграма визначення блоків

2.3.3 Розподіл вимог за компонентами системи

Діаграму трасування було змодельовано у середовищі enterprise architect, на цій діаграмі присутні: вимоги, прецеденти, варіанти тестування системи та елементи користувацького інтерфейсу та відповідні зв'язки між ними [19].

На діаграмі (рис. 2.22), зліва розташовані функціональні вимоги, функції: перегляду даних, надання звітності, аналізу даних, збереження даних та доступу до систем автомобіля і бази даних.

У центральній частині діаграми – сценарії використання у вигляді овалів, що показують конкретні дії користувача в системі. Вони пов'язані з класами діаграми класів які реалізують відповідну логіку. Ці класи містять як контролери, так і прикладні об'єкти, які взаємодіють з інтерфейсами.

З правої сторони розташовані тестові випадки, які завершують ланцюг трасування. Вони демонструють, як кожен сценарій використання тестується в рамках модульного тестування.

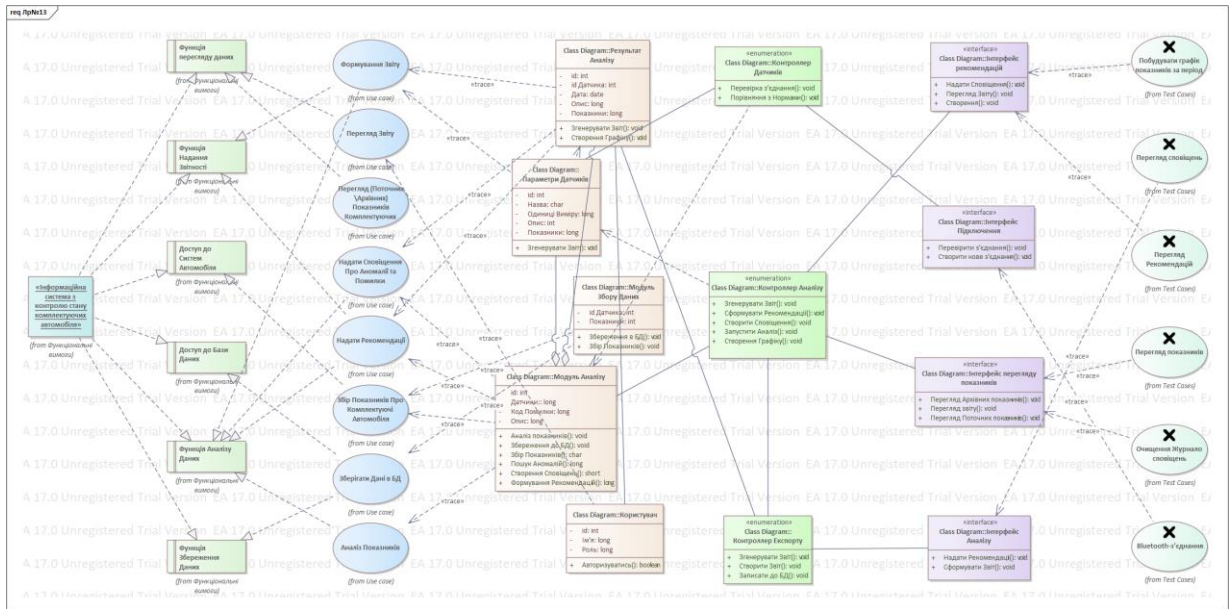


Рисунок 2.22 - Діаграма трасування

Матриці взаємозв'язків, демонструють зв'язки між різними елементами матриці трасування для більш детального пояснення їх взаємодії (рис. 2.23 – 2.24):

Target +	Use case::Аналіз Показників	Use case::Зберігати Дані в БД	Use case::Збір Показників Прс	Use case::Надати Рекомендаці	Use case::Надати Сповідчення	Use case::Перегляд (Поточних	Use case::Перегляд Звіту	Use case::Формування Звіту
+ Source								
Class Diagram::Користувач							↑	
Class Diagram::Модуль Аналізу	↑		↑	↑	↑			
Class Diagram::Модуль Збору Даних		↑	↑					
Class Diagram::Параметри Датчиків								↑
Class Diagram::Результат Аналізу				↑	↑			↑

Рисунок 2.23 - Зв'язок між класами та прецедентами

Ця матриця зв'язків (рис. 2.23), демонструє зв'язки трасування між елементами клас з діаграми класів та прецедентами з діаграми прецедентів (рис. 2.10). Такий тип діаграми дозволяє відстежити, які елементи класів реалізують або підтримують конкретні сценарії використання системи.

Source	Target +	Use case: Аналіз Показників	Use case: Зберегти Дані в БД	Use case: Збір Показників Про Комплекси	Use case: Надати Рекомендації	Use case: Надати Слов'яння Про Анализи	Use case: Перегляд (Потокових) Архівів	Use case: Перегляд Звіту	Use case: Формування Звіту
Функціональні вимоги: Доступ до Бази Даних		↑							
Функціональні вимоги: Доступ до Систем Ав...			↑						
Функціональні вимоги: Функція Аналізу Даних	←			↑	↑				↑
Функціональні вимоги: Функція Збереження ...		↑							
Функціональні вимоги: Функція Надання Звіт...				↑					↑
Функціональні вимоги: Функція перегляду да...						↑	↑		

Рисунок 2.24 - Зв'язок між вимогами та прецедентами

На матриці зв'язків (рис. 2.24) зображено зв'язки між функціональними вимогами та елементами діаграмами прецедентів. Така діаграма дозволяє перевірити, чи кожна вимога має відповідну реалізацію у вигляді конкретного сценарію використання, тим самим забезпечуючи повноту функціонального покриття системи.

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Інформаційне забезпечення

Інформаційне забезпечення (ІЗ) - це сукупність структурованих даних, нормативно-довідкових матеріалів, правил організації, зберігання, обробки, передавання та використання інформації, яка забезпечує функціонування інформаційної системи з контролю стану комплектуючих автомобіля. ІЗ є фундаментальним компонентом, що визначає ефективність роботи системи, її здатність до масштабування, адаптації та інтерпретації результатів аналізу [20].

Інформаційне забезпечення проєктованої системи передбачає збір, зберігання, аналіз та передачу даних від електронних підсистем автомобіля через інтерфейс OBD-II, їх обробку модулем аналізу (на базі Raspberry Pi) і подальшу демонстрацію користувачеві через мобільний застосунок.

3.1.1 Загальна характеристика інформаційного забезпечення

ІЗ системи має клієнт-серверну структуру. Клієнтська частина це мобільний додаток реалізований з допомогою фреймворка Flutter котрий відповідає за взаємодію з користувачем та отримання команд від користувача. Серверна частина складається з бази даних під керуванням СКБД MySQL, де централізовано зберігаються дані, модуля збору даних та модуля аналізу котрий обробляє всі дані.

До складу ІЗ входять такі інформаційні об'єкти:

1. довідкова інформація про компоненти та параметри автомобілів;
2. поточні ("живі") дані, отримані з датчиків через OBD-II ;
3. історичні дані показників для аналізу трендів\$
4. журнал діагностичних кодів помилок, аномалій та системних сповіщень;
5. згенеровані звіти та рекомендації для користувача.

Організація інформаційного забезпечення побудована на принципах централізованого зберігання, нормалізованої структури бази даних за для: мінімізації дублювання інформації, забезпечення логічних зв'язків між усіма сутностями та розділення на поточні й архівні дані. У структурі бази даних визначено окремі таблиці для поточних та архівних показників, довідників компонентів та параметрів, повідомлень про помилки, звітів та рекомендацій.

Уся інформація зберігається на флеш-накопичувачі Raspberry Pi у вигляді реляційної бази даних, реалізованої за допомогою MySQL. Це рішення забезпечує гнучкість, автономність, швидкий доступ до даних, підтримку транзакцій та індексів, а також можливість роботи в умовах обмеженого інтернет-з'єднання. На клієнтському мобільному пристрою використовується флеш-пам'ять для кешування даних та зберігання локальних налаштувань додатку.

Переваги використання MySQL для ІС з контролю стану комплектуючих автомобіля:

1. MySQL є безкоштовною та широко розповсюдженою СКБД, що знижує витрати на розгортання та підтримку;
2. висока продуктивність та надійність - система оптимізована для швидких операцій читання та запису, що є критичним для обробки потокових даних з датчиків;

3. MySQL має чудову підтримку в усіх популярних мовах та фреймворках для розробки серверної частини API, таких як python та php, що значно спрощує інтеграцію з мобільними додатками.

Контроль достовірності та цілісності інформації здійснюється на кількох рівнях. На рівні бази даних застосовуються обмеження NOT NULL, FOREIGN KEY, а також перевірки типів даних. На рівні серверного API - проводиться перевірка відповідності даних, що надходять до ІС від систем автомобіля та мобільного додатку, перед їх збереженням у базу даних.

Достовірність даних з датчиків залежить від автомобільної системи OBD-II . Система зчитує дані що надходять через цей інтерфейс, та виконує перевірку на аномалії, порівнюючи отримані значення з нормальними показниками.

Загальна схема ІЗ демонструє інформаційний потік в ІС з контролю стану комплектуючих автомобіля (рис. 3.1 - 3.2), та містить такі компоненти як:

1. датчики;
2. ODB2 адаптер;
3. Raspberry Pi;
4. база даних;
5. API;
6. Wi-Fi модуль;
7. мобільний додаток.

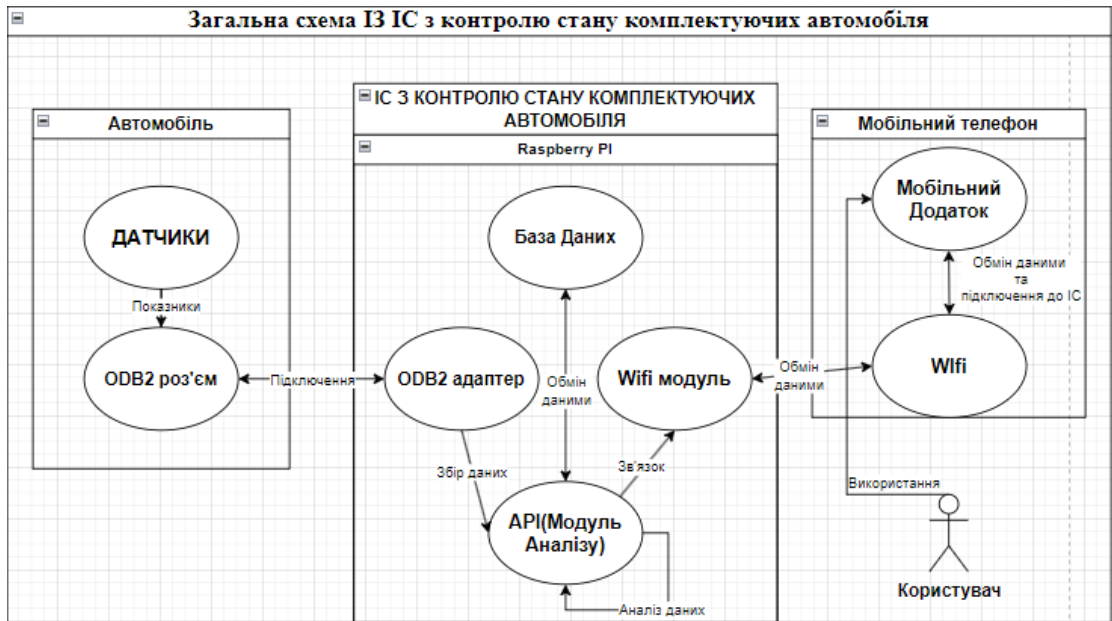


Рисунок 3.1 – Схема ІЗ для ІС(А)

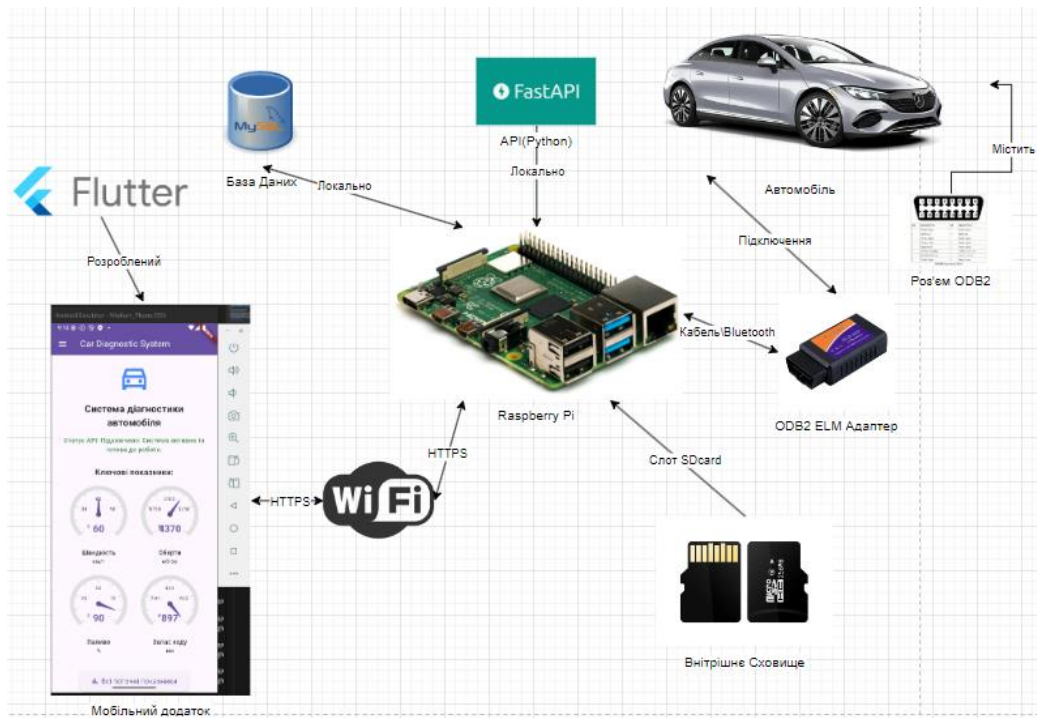


Рисунок 3.2 – Схема ІЗ для ІС(Б)

3.1.2 Організація збору і передавання первинної інформації

Система функціонує в режимі реального часу, безперервно отримуючи дані від електронних блоків управління транспортного засобу, що передають дані до інтерфейсу OBD-II . Ці дані виступають як вхідні повідомлення, які зчитуються апаратно-програмним комплексом, що складається з адаптера OBD-II та пристроєм обробки даних - Raspberry Pi.

Тому основним джерелом первинної інформації є електронні блоки управління автомобіля, які через інтерфейс OBD-II надають доступ до набору показників. Серед яких: температура охолоджуючої рідини, напруга акумулятора, частота обертання двигуна, тиск масла, положення дросельної заслінки, а також коди діагностичних помилок. Передача даних між автомобілем та Raspberry Pi здійснюється по Bluetooth або через USB-кабель, в залежності від конфігурації адаптера ODB2.

Інформація, що надходить від систем автомобіля, подається у вигляді цифрових структурованих повідомлень, які обробляються модулем аналізу. Цей модуль проводить первинну перевірку отриманих значень де перевіряється допустимість значень, відповідність формату, наявність усіх необхідних параметрів. Після цього дані заносяться до бази даних MySQL, яка розгорнута локально на самій Raspberry Pi. Збереження відбувається в таблицях поточних або архівних даних в залежності від типу повідомлення та його частоти надходження.

Передача даних від автомобіля до ІС є циклічною. Більшість технічних параметрів оновлюються з періодичністю в 1 секунду або 5 секунд, залежно від налаштувань системи Повідомлення подій, як-от виявлення коду помилки, надходять асинхронно - лише в момент виникнення відповідної ситуації в системах автомобіля.

Для тестування комунікації з користувачем ІС пропонується мобільний застосунок котрий буде виступати як зовнішній користувацький інтерфейс та одночасно як джерело вхідної інформації керуючих запитів. Запити користувача, наприклад, на побудову графіку або створення звіту, надсилаються через Wi-Fi канал зв'язку з використанням локальної мережі що забезпечує безпеку передачі даних. У результаті таких запитів відбувається запуск певних процесів таких як аналізу або оновлення інформації в базі.

Носіями інформації в ІС є вбудована карта пам'яті Raspberry Pi, а також оперативна пам'ять пристрою. Уся інформація передається в структурованому, цифровому форматі.

Взаємодія між джерелами, каналами передачі, приймачами та системою обробки відбувається з мінімальною затримкою та високим рівнем достовірності, що є критично важливим для стабільної роботи всієї ІС.

3.1.3 Побудова системи класифікації та кодування

В процесі проєктування ІС з контролю стану комплектуючих автомобіля важливою задачею є впровадження чіткої системи класифікації та кодування об'єктів і параметрів. Така система забезпечує уніфікованість, однозначність інтерпретації даних та ефективність обробки інформації в усіх компонентах ІС.

Класифікація здійснюється на основі функціонального поділу автомобіля на логічні компоненти, таких як: двигун, акумулятор, трансмісія, паливна система та інші вузли автомобіля. Кожен з цих компонентів має власний унікальний ідентифікатор у довіднику компонентів системи, що

дозволяє системі легко визначати належність параметрів до конкретної підсистеми транспортного засобу (таблиця А.1).

Кодування параметрів, які збираються від кожного з компонентів, реалізоване за допомогою символічних кодів, що мають фіксовану структуру. Наприклад, код параметра ENG_TEMP означає «температура двигуна», де перша частина (ENG) відповідає компоненту, а друга (TEMP) – конкретному параметру. Така структура дозволяє не лише легко ідентифікувати тип параметра, а й швидко фільтрувати та групувати дані у процесі аналізу (таблиця А.2).

Метод кодування базується на принципі поєднання скорочених латинських позначень компонентів і скорочень назв параметрів, з обмеженням загальної довжини коду до 10 символів. Такий підхід гарантує компактність та зручність зберігання й обробки коду у СКБД:

1. RPM - оберти двигуна;
2. ENG_TEMP - температура охолоджуючої рідини двигуна;
3. BATT_VOLTAGE - напруга акумулятора;
4. VEHICLE_SPEED - швидкість автомобіля.

Класифікаційна система також включає вторинні характеристики, як-от одиниці виміру показників та їх допустимі значення. Всі ці характеристики зберігаються у довіднику параметрів із прив'язкою до основного класифікатора компонентів, що дозволяє забезпечити повноцінний інформаційний контекст для кожного параметра.

Крім того, було спроектовано окремі класифікатори для системних повідомлень та службових документів, які систематизують типи реакцій і результати аналізу. Типи повідомлень, такі як помилки, сповіщення, попередження та аномалії, класифікуються за допомогою унікальних символічних кодів (таблиця А.3). Ці коди забезпечують уніфіковану обробку подій у всіх підсистемах.

Усі результати, сформовані системою такі як звіти, графіки, рекомендації та сповіщення, мають стандартизовані коди документів, що дозволяють автоматизувати логіку обробки, зберігання та виводу інформації. (таблиця А.4).

Спроектвана система класифікації є ієрархічною та нормалізованою. В ній передбачено мінімізацію дублювання інформації, чітке розмежування ідентифікаційних кодів та збереження логічної цілісності бази даних. Усі зв'язки між параметрами, компонентами та записами з даними реалізовані через зовнішні ключі в структурі бази даних.

3.1.4 Проектування форм первинних документів та відеокадрів

Основним видом первинного документа в ІС є звіт про технічний стан автомобіля, що формується за запитом користувача. Він містить поточні значення параметрів, результати аналізу на пошук відхилень, виявлені помилки, текстові пояснення, а також рекомендовані дії. Звіт має ідентифікатор RPT_STATUS, формується у текстовому форматі, видається користувачеві в інтерфейсі мобільного застосунку, а також зберігається в базі даних для формування історії технічного стану. Періодичність створення звіту - за запитом, з максимально допустимим часом затримки до 2 секунд (таблиця Б.1).

Ще одним типом документа є машинограма повідомлення про помилку, що має ідентифікатор ERR_NOTIFY. Вона формується автоматично при виявленні коду помилки або несправності (наприклад, P0171 - система надто бідна). Такий документ надсилається у вигляді push-сповіщення на мобільний пристрій користувача та паралельно фіксується у журналі подій. Мета повідомлення - оперативне інформування користувача

про технічні недоліки, а також збереження події для подальшого аналізу. Формат подання - коротке текстове повідомлення з критичністю, описом помилки, компонентом та кодом (таблиця Б.2).

У системі також спроєктовано форму графічного представлення параметрів - так динамічні графіки, які мають ідентифікатор LIVE_GRAPH. Ці графіки будуються на основі архівних та поточних значень параметрів, зчитаних із датчиків. Вони дозволяють користувачу відслідковувати динаміку зміни параметрів комплектуючих автомобіля протягом обраного проміжку часу. Формат подання - інтерактивний графік, що демонструється у мобільному застосунку. Побудова графіка виконується за запитом користувача, із затримкою до 2 секунд, джерелом даних є таблиця архівних показників з бази даних MySQL (таблиця Б.3).

Рекомендації системи - SYS_RECOMMEND є одним з документів котрі формуються системою на основі модуля аналізу при виявленні аномальних показників при перевищенні критичних порогів. У рекомендації наводиться текстовий опис дії, яку бажано виконати (наприклад, «перевірте рівень охолоджувальної рідини»), вказується, до якого параметра вона стосується, та зазначається рівень критичності котрий поділяється на такі рівні: Low, Medium та High. Рекомендації можуть бути збережені, переглянуті користувачем в інтерфейсі додатка або передані у вигляді звіту (таблиця Б.4).

Всі документи ІС мають уніфіковану структуру, часову мітку, пов'язані параметри, назву компонента, ідентифікатор повідомлення та, за необхідності, детальний опис або графічне подання.

3.1.5 Структура інформаційних масивів

В межах функціонування ІС з контролю стану комплектуючих автомобіля критично важливим є чітке визначення структури інформаційних масивів, які використовуються системою для зберігання, обробки та демонстрації даних. Під час проєктування ІС було визначено такі інформаційні масиви:

Масив `historical_data`(архівні показники) - містить архівні показники параметрів комплектуючих автомобіля що використовуються для подальшого аналізу, побудови графіків та створення звітів.

Ідентифікатор масиву: `historical_data`

Носій: MySQL

Максимальний обсяг: 10 000 000 записів

Довжина запису: ~256 байт

Метод організації: Реляційна таблиця

Ключі упорядкування: `timestamp`, `parameter_id`

Індексний масив: `IDX_param_time`

Таблиця 3.1 - Структури масиву `historical_data`

Найменування	Ідентифікатор	Позначення	Формат	Бізнес-правила	Зв'язки	Ключ	Умова	Обов'язк.	Індекс
ІД запису	<code>id</code>	<code>ID_hist</code>	BIGINT	автоінкремент	-	PK	> 0	так	ІНД
Параметр	<code>parameter_id</code>	<code>p</code>	INT	зовн. ключ	<code>parameters(id)</code>	FK	> 0	так	ІД
Значення	<code>value</code>	<code>v</code>	VARCHAR(255)	Не. Порож.	-	-	≠ null	так	-
Час запису	<code>timestamp</code>	<code>t</code>	DATETIME	Хроно. Прив.	-	-	≠ null	так	ІД

Джерело: сформовано автором на основі виконаного дослідження

Масив live_data(поточні показники) - містить поточні показники параметрів, що постійно оновлюються у реальному часі для оперативної демонстрації стану комплектуючих у мобільному застосунку.

Ідентифікатор: live_data

Носій: MySQL

Обсяг: ~1000 записів

Довжина: ~128 байт

Метод: Реляційна таблиця

Упорядкування: parameter_id

Індекс: PK_param_id

Таблиця 3.2 - Структури масиву live_data

Найменування	Ідентифікатор	Позначення	Формат	Бізнес-правила	Зв'язки	Ключ	Умова	Обов'язк.	Індекс
Параметр	parameter_id	p	INT	унікальний ключ	parameters(id)	PK	> 0	так	ІНД
Значення	value	v	VARCHAR(255)	актуальне значення	-	-	≠ null	так	-
Час оновлення	timestamp	t	DATETIME	автооновлення по зміні	-	-	≠ null	так	ІНД

Джерело: сформовано автором на основі виконаного дослідження

Масив error_logs(помилки) - виконує функцію журналу помилок, в якому фіксуються всі виявлені технічні несправності, аномалії та сповіщення разом із часом виникнення і описом.

Ідентифікатор: error_logs

Носій: MySQL

Обсяг: 100 000 записів

Довжина: ~512 байт

Метод: Реляційна таблиця

Упорядкування: timestamp, type

Індекс: IDX_error_type_time

Таблиця 3.3 - Структури масиву error_logs

Назва поля	Ідентифікатор	Позначення	Формат	Бізнес-правила	Зв'язки	Ключ	Умова	Обов'язк.	Індекс
ID запису	id	ID_err	INT	автоінкремент	-	PK	> 0	так	ІНД
Код помилки	code	c	VARCHAR(50)	може бути порожнім	-	-	-	ні	ІДД
Тип повідомлення	type	t	VARCHAR(100)	ERR, WARN, NOTIFY	-	-	≠ null	так	-
Опис	description	d	TEXT	короткий опис	-	-	≠ null	так	-
Компонент	component_name	comp	VARCHAR(255)	пов'язаний з помилкою	-	-	-	ні	-
Час	timestamp	time	DATETIME	дата фіксації	-	-	≠ null	так	-
Нове	is_new	new	BOOLEAN	TRUE/FALSE	-	-	≠ null	так	-

Джерело: сформовано автором на основі виконаного дослідження

Масив reports(звіти) - призначений для збереження сформованих звітів про технічний стан комплектуючих автомобіля, які генеруються за запитом користувача.

Ідентифікатор: reports

Обсяг: 50 000 записів

Довжина: ~1024 байти

Метод: Реляційна таблиця

Упорядкування: timestamp

Індекс: IDX_report_time

Таблиця 3.4 - Структури масиву reports

Поле	Ідентифікатор	Позначення	Формат	Бізнес-правила	Зв'язки	Ключ	Умова	Обов'язк.	Індекс
ID	id	ID_rpt	INT	автоінкремент	-	PK	> 0	так	ІНД
Назва	title	ttl	VARCHAR(255)	заголовок звіту	-	-	≠ null	так	-
Вміст	content	cnt	TEXT	текстовий зміст звіту	-	-	≠ null	так	-
Час створення	timestamp	time	DATETIME	фіксація моменту створення	-	-	≠ null	так	-

Джерело: сформовано автором на основі виконаного дослідження

Масив recommendations - використовується для збереження системних рекомендацій щодо обслуговування чи перевірки певних компонентів на основі їх критичності, виявленої системою аналізу.

Ідентифікатор: recommendations

Обсяг: 10 000 записів

Довжина: ~512 байт

Метод: Реляційна таблиця

Упорядкування: timestamp

Індекс: IDX_recommend_time

Таблиця 3.5 - Структури масиву reports

Поле	Ідентифікатор	Позначення	Формат	Бізнес-правила	Зв'язки	Ключ	Умова	Обов'язк.	Індекс
ID	id	ID_rec	INT	автоінкремент	-	PK	> 0	так	ІНД
Рекомендація	content	cnt	TEXT	текст поради	-	-	≠ null	так	-
Параметр	related_param_code	param	VARCHAR(50)	посилання на параметр	-	-	-	ні	-
Критичність	criticality	crit	ENUM	Low, Medium, High	-	-	≠ null	так	-
Час	timestamp	time	DATETIME	момент створення	-	-	≠ null	так	-

Джерело: сформовано автором на основі виконаного дослідження

3.1.6 Вибір СКБД

У процесі проектування ІС з контролю стану комплектуючих автомобіля важливою складовою є вибір системи керування базами даних, яка забезпечуватиме надійне зберігання, обробку та доступ до даних у

реальному часі. Враховуючи специфіку системи, яка передбачає часті читання, регулярне оновлення інформації від датчиків та аналітичну обробку архівних даних, ключовими вимогами до СКБД є:

1. підтримка високої продуктивності при великій кількості записів;
2. зручна інтеграція з серверною частиною;
3. можливість реалізації складних запитів SQL;
4. підтримка транзакцій, зовнішніх ключів та цілісності даних;
5. відкритість, безкоштовність та підтримка широким співтовариством;
6. сумісність з платформами Linux та ARM-Архітектури.

З урахуванням наведених критеріїв, у проєкті слід використовувати СКБД MySQL, яка відповідає усім вимогам. Вона є однією з найпопулярніших у світі реляційних систем керування базами даних з відкритим кодом. MySQL дозволяє створювати оптимізовані індекси, використовувати складні агрегатні функції для аналітики, реалізовувати тригери, обмеження, зовнішні ключі та забезпечувати масштабованість у майбутньому.

Додатковими перевагами MySQL є її інтеграція з Python через бібліотеки `mysql-connector-python` або `SQLAlchemy`, а також простий процес встановлення на серверні платформи з обмеженими ресурсами що є критично важливим для проєктованої системи.

Можливі обмеження при використанні MySQL полягають у:

1. обмеженій підтримці NoSQL-підходів;
2. складніша реалізації масштабування у горизонтальному напрямі в порівнянні з новими СКБД;
3. потребі у грамотному проєктуванні індексів при роботі з великими масивами архівних даних, щоб уникнути зниження продуктивності при аналітичних запитах.

Незважаючи на ці обмеження, MySQL залишається збалансованим вибором для системи, що поєднує режим реального часу, обробку архівних даних, високу надійність та доступність інструментів адміністрування. Ця СКБД забезпечує усі необхідні функції для реалізації проєктованої ІС.

3.1.7 Інфологічна модель бази (сховища) даних

Інфологічна модель бази даних ІС з контролю стану комплектуючих автомобіля спроектована на основі аналізу предметної області та функціональних вимог до системи. В основі проєктування інфологічної моделі було використано принцип нормалізації до третьої нормальної форми, що дозволяє мінімізувати дублювання даних та забезпечити логічну цілісність бази.

Моделювання виконано за допомогою CASE-засобу dbdiagram.io, який дозволяє візуально створити сутності та зв'язки між ними. Модель зосереджена на ключових об'єктах системи - компонентах автомобіля, параметрах, архівних даних, поточних показниках, помилках, звітах та рекомендаціях. Кожна сутність відображає відповідну таблицю, що використовується в процесі збору, аналізу та демонстрації інформації (рис. 3.3).

Основна структура інфологічної моделі має такі логічні сутності:

1. `component` – описує логічну одиницю автомобіля (двигун, акумулятор тощо).
2. `parameter` – параметри, пов'язані з компонентами (наприклад, температура, напруга).

3. reading – концептуальна сутність, що представляє зафіксоване значення певного параметра в конкретний момент часу, об'єднуючи ідею поточних та архівних даних.
4. error log – журнал помилок, аномалій та системних сповіщень.
5. report – згенеровані системою звіти про технічний стан автомобіля.
6. recommendation – автоматичні поради та вказівки користувачу, сформовані на основі аналізу даних.

Всі сутності зв'язані між собою через атрибути зовнішніх ключів. Також передбачено унікальні коди (наприклад, ENG_TEMP, RPM, ERR_NOTIFY), які дозволяють легко класифікувати, індексувати та групувати інформацію.

У рамках цієї IC OLAP не проєктувався. Спроектowana IC орієнтована на операційну обробку даних, основною метою якої є збір, аналіз, збереження та демонстрація даних у реальному часі. У зв'язку з цим, для зберігання даних було спроектовано класичну нормалізовану реляційну базу даних, а не спеціалізоване сховище даних.

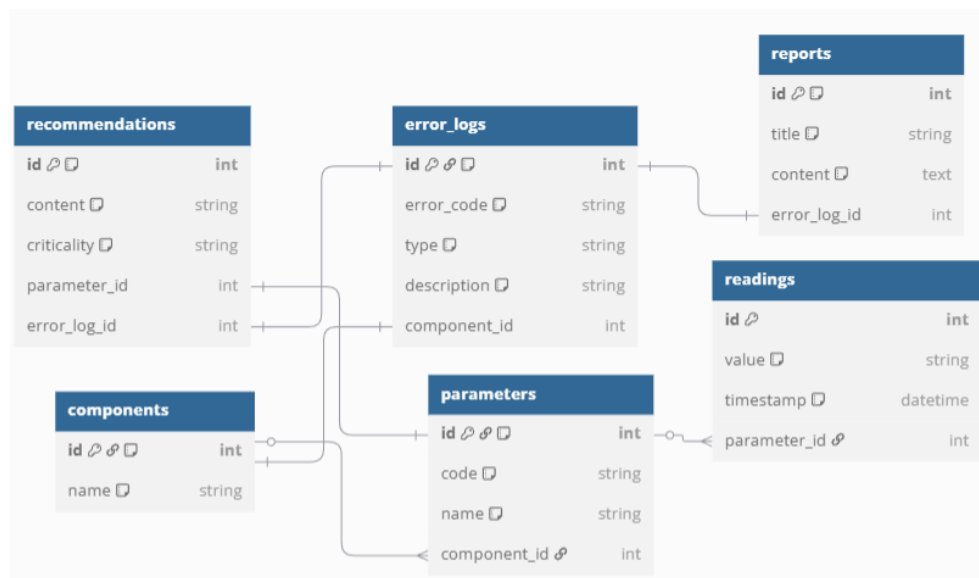


Рисунок 3.3 – Інфологічна модель бази даних

3.1.8 Даталогічна модель бази (сховища)даних

У проєктованій ІС з контролю стану комплектуючих автомобіля було обрано MySQL, що дозволяє реалізувати гнучку, масштабовану та продуктивну модель даних із підтримкою транзакцій, зовнішніх ключів та високою швидкістю доступу до інформації.

Даталогічна модель інформаційної системи реалізована у вигляді реляційної структури, що складається з взаємопов'язаних таблиць, зв'язки між якими забезпечуються за допомогою первинних та зовнішніх ключів. Для проєктування моделі було створено ER-діаграму реалізації бази даних із використанням інструменту MySQL Workbench. Діаграма відображає основні сутності системи - recommendations, error_logs, reports, components, historical_data, live_data, parameters та відповідні зв'язки між ними (рис. 3.4).

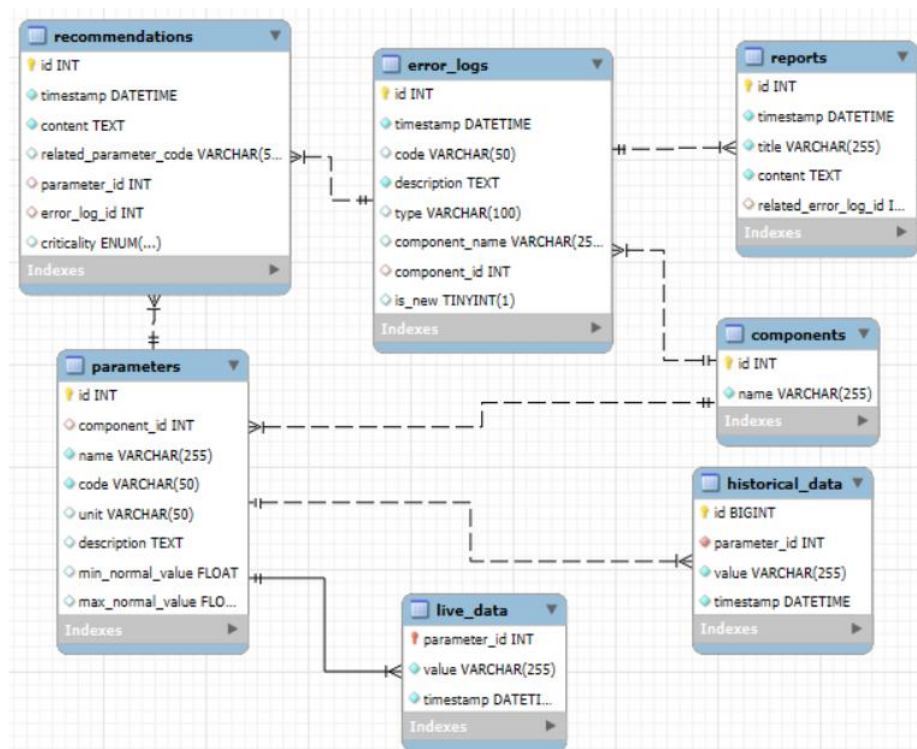


Рисунок 3.4 – Даталогічна модель бази даних

Характеристики спроектованої даталогічної моделі бази даних:

1. усі таблиці (components, parameters, live_data, historical_data, error_logs, reports, recommendations) мають первинні ключі що забезпечують унікальність кожного запису;
2. для зберігання даних про параметри реалізовано чітке розділення: таблиця components є базовим довідником вузлів автомобіля, а parameters містить детальний опис кожного показника, що з ним пов'язаний;
3. потоки даних розділені для оптимізації доступу: таблиця live_data зберігає лише останні актуальні значення для швидкого відображення, тоді як historical_data накопичує всі показники для подальшого аналізу та побудови графіків;
4. результати аналізу зберігаються в окремих логічних таблицях: error_logs фіксує всі виявлені помилки та аномалії, recommendations містить поради для користувача, а reports – узагальнені звіти про стан системи;
5. цілісність даних забезпечується за допомогою реляційних зв'язків (зовнішніх ключів), наприклад, кожне значення в таблицях live_data та historical_data обов'язково пов'язане з відповідним записом у таблиці parameters.

3.2 Технічне забезпечення

3.2.1 Загальні положення та схема автоматизації

Архітектура побудована на відкритих та поширених апаратних засобах, що знижує вартість впровадження. Система спроектована для повної автономної роботи, не потребуючи підключення до мережі Інтернет або зовнішніх серверів, що підвищує її безпеку та конфіденційність (рис. 3.5).

Комплекс технічних засобів включає такі компоненти:

1. У ролі сервера виступає одноплатний мікрокомп'ютер Raspberry Pi 4B. Цей пристрій виконує всі основні функції: на ньому розгорнуто операційну систему, СКБД MySQL, працює модуль аналізу даних, і він же створює Wi-Fi мережу для зв'язку з мобільним застосунком. Його вибір обґрунтований низькою вартістю, компактністю, енергоефективністю та достатньою продуктивністю для обробки даних в реальному часі;
2. Для зчитування даних з комплектуючих автомобіля використовується зовнішній адаптер OBD-II, який підключається до стандартного діагностичного роз'єму автомобіля. Цей пристрій відповідає за зчитування даних з електронного блоку управління транспортного засобу та їх передачу на Raspberry Pi;
3. Вся системна інформація, включаючи операційну систему, програмне забезпечення та базу даних, зберігається на карті пам'яті формату microSD об'ємом від 32 ГБ, яка встановлена в Raspberry Pi;

4. Автоматизованим робочим місцем користувача є мобільний додаток, розроблений з використанням фреймворку Flutter. Він встановлюється на персональний смартфон власника та слугує основним засобом для взаємодії з ІС. Додаток підключається до локальної Wi-Fi мережі, створеної Raspberry Pi, та дозволяє користувачеві переглядати поточні та архівні показники, отримувати сповіщення про помилки й аномалії, а також генерувати звіти про технічний стан автомобіля.

Засобами комунікації виступають - вбудований у Raspberry Pi Wi-Fi модуль використовується для створення локальної точки доступу, до якої підключається мобільний пристрій користувача для використання мобільного застосунку та обміну даними, а також підключення Bluetooth або USB використовується для з'єднання між OBD-II адаптером та Raspberry Pi, в залежності від моделі адаптера.

Система не передбачає використання спеціалізованих периферійних технічних засобів; їхню функцію виконує мобільний пристрій користувача. Також, оскільки ІС розрахована на індивідуальне використання власником автомобіля, наявність окремого обслуговуючого персоналу не потрібна.

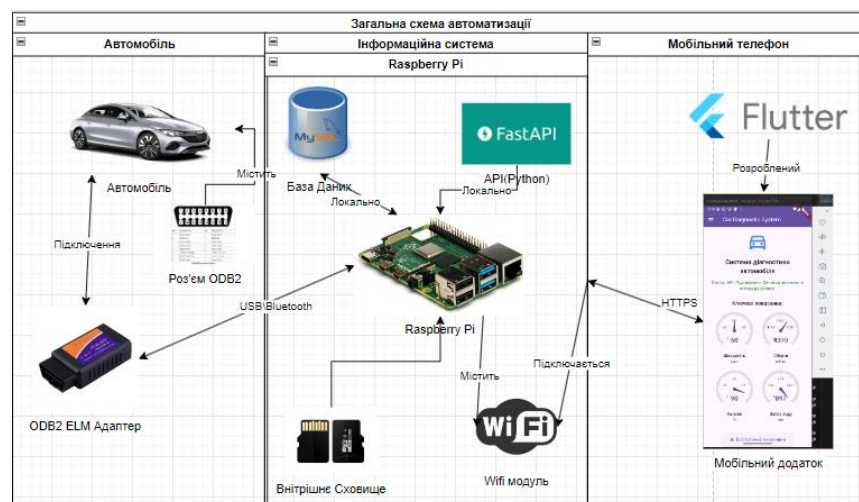


Рисунок 3.5 – Загальна схема автоматизації

3.2.2 Структура комплексу технічних засобів

Вибір структури комплексу технічних засобів базується на ключових вимогах до системи: автономності, доступності та безпеки. Використання мікрокомп'ютера Raspberry Pi як центрального вузла дозволяє створити автономне рішення, що не залежить від зовнішніх серверів та інтернет-з'єднання. Це забезпечує високий рівень конфіденційності даних та знижує вартість впровадження завдяки використанню доступних компонентів.

В якості сервера та основної ЕОМ було обрано одноплатний мікрокомп'ютер Raspberry Pi. Цей вибір обґрунтований його низькою вартістю, компактними розмірами, енергоефективністю та достатньою продуктивністю для обробки поточкових даних з автомобіля в реальному часі. До складу КТЗ також входить адаптер OBD-II для зчитування даних з діагностичного порту автомобіля та карта пам'яті microSD яка слугує основним носієм інформації для операційної системи, програмного забезпечення та бази даних.

Технічні рішення по обміну даними включають - використання Bluetooth або USB для з'єднання між OBD-II адаптером та Raspberry Pi, а також вбудованого Wi-Fi модуля Raspberry Pi для створення локальної точки доступу. Ця локальна мережа є єдиною обчислювальною мережею об'єкта, що забезпечує зв'язок з мобільним додатком користувача.

Розміщення КТЗ здійснюється безпосередньо в салоні автомобіля: адаптер підключається до діагностичного роз'єму, а Raspberry Pi розташовується поруч і живиться від бортової мережі. Захист даних реалізовано шляхом створення ізольованої локальної мережі Wi-Fi, захищеної паролем, що використовується для обміну даними між мобільним додатком та API.

Система не вимагає спеціалізованих периферійних технічних засобів. Функцію основного засобу відображення та введення інформації виконує персональний мобільний пристрій користувача. Оскільки ІС розрахована на індивідуальне використання власником автомобіля, залучення окремого технічного персоналу для її експлуатації не передбачається.

3.2.3 Опис автоматизованого робочого місця

У проєктованій ІС функцію автоматизованого робочого місця виконує персональний мобільний пристрій користувача, на якому встановлено клієнтський мобільний додаток. Таке рішення обґрунтоване тим, що система призначена для індивідуального використання власником автомобіля, а смартфон є найзручнішим та завжди доступним засобом для взаємодії. Мінімально допустимі системні вимоги до технічних засобів АРМ для коректної роботи мобільного додатку, розробленого з використанням фреймворку Flutter, є наступними:

1. операційна система - android 8.0 (oreo) або вище / ios 12.0 або вище;
2. оперативна пам'ять - від 3 гб для стабільної роботи інтерфейсу та обробки даних;
3. процесор -4-ядерний процесор з тактовою частотою від 1.8 гГц;
4. вільне місце - від 100 мб вільного місця для встановлення додатку та зберігання кешованих даних;
5. мережевий інтерфейс - наявність вбудованого модуля wi-fi.

Обґрунтування вибору периферійних технічних засобів АРМ полягає в тому, що система не потребує додаткового обладнання. Усі функції отримання команд, контролю та демонстрації інформації виконуються за

допомогою стандартних компонентів смартфона: сенсорного екрану та вбудованих динаміків для звукових сповіщень.

Розрахунок кількості технічних засобів: архітектура системи передбачає один АРМ на один апаратно-програмний комплекс, тому що система розрахована на одного користувача. Підключення АРМ до обчислювального вузла відбувається через бездротову мережу Wi-Fi, яку створює Raspberry Pi. Це забезпечує пряме з'єднання в локальному, ізольованому середовищі, тому використання додаткового мережевого устаткування не є необхідним.

3.2.4 Схеми мережі передачі даних

Проектована ІС з контролю стану комплектуючих автомобіля використовує однорангову локальну мережу, створену Raspberry Pi як центральним вузлом зв'язку, який одночасно виконує функції сервера, точки доступу Wi-Fi та вузла збору і обробки даних. Обмін інформацією між модулями системи відбувається в межах цієї локальної мережі, без необхідності підключення до інтернету, що значно підвищує рівень інформаційної безпеки та автономності роботи.

Каналом зв'язку між мобільним додатком користувача та сервером виступає бездротовий зв'язок Wi-Fi, який забезпечує стабільну швидкість передачі даних, достатню для демонстрації показників у реальному часі, обміну запитами до API та перегляду архівних показників та їх графіків. Передача інформації від діагностичного адаптера до Raspberry Pi відбувається через Bluetooth або USB-інтерфейс - в залежності від типу адаптера. У разі використання Bluetooth-з'єднання забезпечується радіус дії до 10 метрів, чого цілком достатньо для розміщення пристроїв у салоні

автомобіля. USB-з'єднання застосовується у разі потреби підвищеної стабільності та швидкості передачі сигналів.

Система не використовує широкосмугові зовнішні канали зв'язку, що захищає від залежності якості інтернет-покриття і захищає інформацію від несанкціонованого доступу. Всі дані зберігаються локально на карті пам'яті microSD, встановленій у Raspberry Pi, що містить операційну систему, базу даних, програмну логіку і лог-файли.

Розмежування каналів передачі інформації полягає у поділі локальної бездротової мережі, створеної Raspberry Pi, та каналу Bluetooth/USB, який з'єднує адаптер з сервером. Загальна кількість мережного обладнання обмежується Raspberry Pi, OBD-II -адаптером та мобільним пристроєм користувача, що робить систему простою, портативною та енергоефективною.

До основного мережевого обладнання відноситься вбудований Wi-Fi-модуль Raspberry Pi, який функціонує в режимі точки доступу та автоматично видає динамічні IP-адреси клієнтам за протоколом DHCP.

Для потреб безпеки та стабільності роботи система може бути забезпечена захистом від статичної напруги, підтримкою оновлення прошивки безпосередньо через локальну мережу, а також можливістю резервного копіювання даних з SD-карти. Тип IP-адрес є динамічним, генерується Raspberry Pi локально і не виходить за межі внутрішнього сегменту мережі.

Щодо активності користувачів, система орієнтована на одного одночасного користувача, який отримує доступ до даних з мобільного пристрою. Дані передаються у вигляді JSON-запитів, що мінімізує навантаження на мережу та дозволяє системі працювати без затримок. Передача даних залежить від запитів користувачів, тобто передача відбувається при ініціації користувачем або при фіксації змін у параметрах.

Обладнання, яке використовується в системі, має високий рівень надійності. Raspberry Pi демонструє середній термін безвідмовної роботи в середньому два роки, OBD-II адаптер підтримує до 100 тисяч з'єднань, а карти пам'яті класу High Endurance витримують до 10 тисяч циклів перезапису.

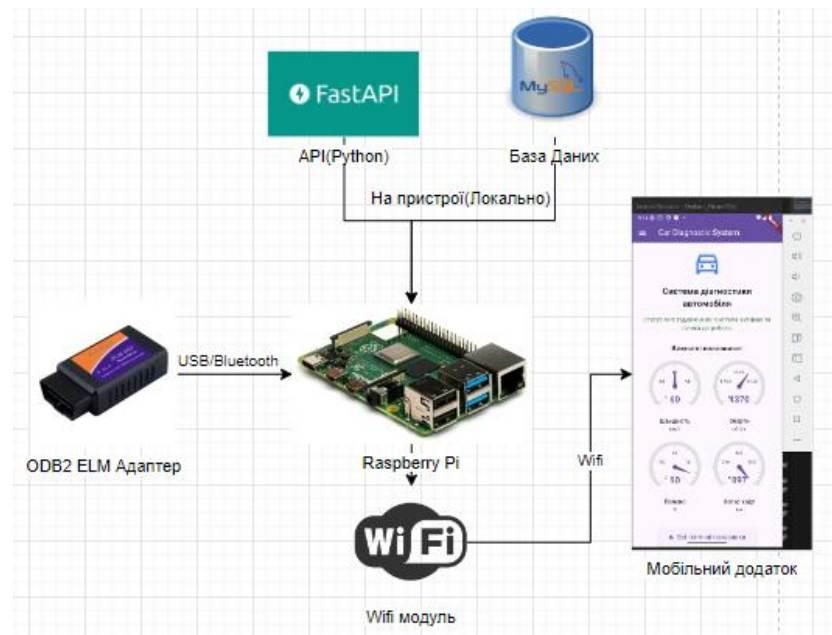


Рисунок 3.5 – Схема мережі передачі даних

3.3 Програмне забезпечення

3.3.1 Структура програмного забезпечення

Структура програмного забезпечення проєктованої ІС з контролю стану комплектуючих автомобіля є багаторівневою і охоплює як базові програмні засоби, так і прикладні компоненти, що забезпечують виконання функціональних вимог. В основі побудови структури ПЗ лежить принцип

модульності, що дозволяє забезпечити гнучкість, масштабованість та простоту супроводу системи (рис. 3.6).

На верхньому рівні архітектури програмного забезпечення виокремлюються три основні підсистеми: системне програмне забезпечення, прикладне програмне забезпечення та програмна документація. Взаємодія між цими підсистемами реалізована через стандартизовані інтерфейси, що дозволяє забезпечити сумісність, повторне використання коду та зниження витрат на адаптацію системи до нових середовищ.

Системне програмне забезпечення базується на операційній системі Raspberry Pi OS, яка встановлюється на одноплатному комп'ютері Raspberry Pi 4B. У складі базового ПЗ також функціонують такі компоненти, як сервер бази даних MySQL, інтерпретатор Python, сервер API на базі веб-фреймворку FastAPI, а також драйвери для взаємодії з OBD-II адаптером. Такий набір засобів забезпечує стабільну платформу для виконання прикладних програм, реалізуючи низькорівневу взаємодію з апаратною частиною, управління ресурсами та обробку даних у реальному часі.

Прикладне програмне забезпечення включає два основні компоненти. Перший - це серверна частина, реалізована мовою Python, яка здійснює обробку, збереження та аналітику отриманих даних. У її складі працюють сервіси збору даних, аналізу аномалій, формування звітів та генерації рекомендацій. Другий - це мобільний застосунок, створений на базі фреймворку Flutter, який виконує функції візуалізації даних, надання сповіщень користувачу та управління параметрами системи.

Третім структурним компонентом є програмна документація, що створюється як у процесі розробки, так і після впровадження ПЗ. Вона охоплює такі елементи, як опис архітектури, специфікації API, інструкції для користувача, формуляри, приклади використання, а також діаграми, створені в Enterprise Architect. Така документація дозволить новим

розробникам швидко ознайомитися з проєктом, а також є необхідною складовою для подальшого супроводу та вдосконалення системи.

Структура програмного забезпечення		
Системне програмне забезпечення	Прикладне програмне забезпечення	
	Серверна частина(Backend)	Мобільний застосунок(Frontend)
Операційна система: Raspberry Pi OS Компоненти системного ПЗ: Сервер бази даних MySQL; Інтерпритатор мови програмування Python; Сервер API на базі веб-фреймворку FastAPI; Драйвери для взаємодії з OBD-II адаптером	Мова програмування: Python Веб-фреймворк: FastAPI Серверна частина забезпечує: Прийом та обробку даних з OBD-II адаптера; Збереження та оновлення даних у MySQL бази даних; Виявлення аномалій, помилок та створення рекомендацій;	Розробка у середовищі Flutter (Dart); Підключення до серверного API через HTTPS; Основні функції: Авторизація/ідентифікація користувача; Відображення поточних показників з датчиків у вигляді графіків і циферблатів; Перегляд архівних даних;
Програмна документація	Формування REST API для мобільного застосунку; Підтримку WebSocket для передачі актуальних показників у реальному часі; Сховище: локально на Raspberry Pi, карта пам'яті microSD; Бібліотеки: pymysql, sqlalchemy, uvicorn, pydantic, obd, numpy, scikit-learn (для простого аналізу).	Отримання повідомлень про помилки, сповіщення та аномалії; Ознайомлення з рекомендаціями та перегляд сформованих звітів; Фільтрація даних за критичністю, джерелом, періодом часу тощо; UI-бібліотеки: fl_chart, http, provider, shared_preferences.
Програмна документація, що створюється як у процесі розробки, так і після впровадження ПЗ. Вона охоплює такі елементи, як опис архітектури, специфікації API, інструкції для користувача, формуляри, приклади використання, а також діаграми, створені в Enterprise Architect. Така документація дозволить новим розробникам швидко ознайомитися з проєктом, а також є необхідною складовою для подальшого супроводу та вдосконалення системи.		

Рисунок 3.6 – Структура програмного забезпечення

3.3.2 Системне програмне забезпечення

У проєктованій ІС з контролю стану комплектуючих автомобіля використовується спеціалізована версія операційної системи на базі ядра Linux - Raspberry Pi OS, котра забезпечує надійне та ефективне середовище для розгортання сервера та обробки даних. Її вибір обумовлений високою стабільністю, безкоштовністю, сумісністю з апаратним забезпеченням Raspberry Pi та широкою підтримкою спільноти.

До складу системного програмного забезпечення входить система керування базами даних MySQL, яка забезпечує зберігання структурованих

даних, у тому числі поточних і архівних показників, журналів повідомлень, рекомендацій та інших інформаційних масивів. MySQL є надійним та продуктивним рішенням, що добре масштабується та підтримується засобами Python.

Інтерпретатор мови Python з використанням веб-фреймворку FastAPI є основою серверної частини системи. FastAPI обрано через його продуктивність, зручність реалізації RESTful API, можливість обробки запитів у реальному часі та розширення за допомогою WebSocket.

Для розробки та візуалізації моделі бази даних використовувались такі інструменти як dbdiagram.io та MySQL Workbench, що дозволяють створювати інфологічну та даталогічну модель.

Серед інструментального ПЗ, що підтримує розробку та тестування, використовувалися:

1. текстові редактори та IDE (Visual Studio Code);
2. емулятор терміналу для Linux;
3. утиліти для моніторингу навантаження на систему;
4. бібліотеки Python для роботи з OBD-II інтерфейсом (python-OBD), обробки даних (NumPy, pandas).

Безпека даних та захист інформації забезпечується ізоляцією системи в локальній мережі та з'єднання між клієнтом та сервером, а також збереженням логів на пристрої без доступу до зовнішнього інтернету.

Завдяки використанню відкритих та кросплатформних рішень, обрана структура системного програмного забезпечення дозволяє знизити витрати, забезпечити гнучкість, адаптивність до змін, та відповідність сучасним вимогам до розробки вбудованих систем.

3.3.3 Прикладне програмне забезпечення

У межах проєктування ІС з контролю стану комплектуючих автомобіля прикладне програмне забезпечення реалізується як сукупність взаємодіючих модулів, орієнтованих на автоматизацію збирання, обробки, візуалізації та зберігання технічної інформації комплектуючих автомобіля та складається з двох розроблених компонентів, а саме серверної частини та мобільного застосунку.

Серверна частина побудована з використанням Python та веб-фреймворку FastAPI, який дозволяє ефективно реалізувати REST API для взаємодії з мобільним застосунком. Вона відповідає за обробку вхідних даних від OBD-II адаптера, збереження цих даних у MySQL, виконання базових аналітичних функцій та виконання запитів клієнта. Також на стороні серверу реалізовано механізм сповіщень про критичні події.

Мобільний застосунок, розроблений на Flutter, реалізує інтерфейс користувача для перегляду поточних і архівних показників, графіків, повідомлень, рекомендацій та параметрів налаштувань. Він забезпечує зручний доступ до інформації в реальному часі завдяки прямому підключенню до локальної Wi-Fi мережі Raspberry Pi. Мобільна частина охоплює методо-орієнтоване ПЗ, дозволяючи користувачеві інтуїтивно керувати процесами взаємодії із системою, отримувати сповіщення та візуально контролювати стан автомобіля.

До середовищ що були використані при розробці прикладного ПЗ відносяться Visual Studio Code та Android Studio, що відповідають вимогам до інструментів категорії RAD – вони забезпечують швидке проєктування інтерфейсів, автостворення коду, підтримку емуляторів і вбудованих інструментів для налагодження.

Розроблене прикладне програмне забезпечення системи забезпечує повний цикл: збирання, обробки, зберігання, аналітики та демонстрації даних, будучи гнучким, масштабованим і адаптованим до реального використання в умовах локального середовища.

3.3.4 Програмна документація

У процесі розробки ІС з контролю стану комплектуючих автомобіля мною було сформовано комплекс програмної документації, що забезпечує можливість її впровадження, тестування та подальшої експлуатації. Документація охоплює опис структури системи, особливості реалізації компонентів, специфікацію середовища розгортання та мінімальні технічні вимоги. Вона є невід'ємною частиною інженерного проєкту та відповідає чинним вимогам до оформлення програмних засобів.

Формуляр інформаційної системи (таблиця 3.6) узагальнює ключові технічні характеристики програмного комплексу. У ньому вказано найменування системи, мови програмування, середовище розгортання, тип використовуваної бази даних, а також склад функціональних компонентів: серверна частина, мобільний застосунок, база даних, API та засоби інтеграції з OBD-II адаптером.

Таблиця 3.6 – Формуляр ІС

Найменування	ІС з контролю стану комплектуючих автомобіля
Мова розробки	Мова програмування Dart та фреймворк Flutter для мобільного клієнта, мова програмування Python з FastAPI
База даних	MySQL

Продовження таблиці 3.6

Середовище розгортання	Raspberry Pi 4B, з встановленою операційною системою Raspberry Pi OS
Компоненти ПЗ	Сервер API для аналітики та функцій системи, база даних MySQL для збереження та мобільний застосунок для взаємодії користувача з системою

Джерело: сформовано автором на основі виконаного дослідження

У ході розробки було визначено призначення системи - реалізація автоматизованого процесу моніторингу технічного стану транспортного засобу. Основний функціонал системи охоплює збір телеметричних даних з електронного блоку управління автомобіля через інтерфейс OBD-II, їх локальну обробку на сервері, зберігання історичних значень параметрів у базі даних, формування повідомлень, графіків, звітів і рекомендацій, а також представлення обробленої інформації у зручному для користувача форматі через мобільний інтерфейс.

Встановлення програмного забезпечення та його коректне функціонування можливе лише за умови дотримання певних вимог до технічної конфігурації середовища. Для цього були визначені як мінімальні, так і оптимальні вимоги до обчислювального пристрою, що виконує роль локального сервера. В якості платформи було обрано мікрокомп'ютер Raspberry Pi, який відповідає критеріям енергоефективності, компактності та має достатню обчислювальну потужність для виконання поставлених завдань.

Було підготовлено інструкції для налаштування програмного середовища. Ще наведено порядок розгортання серверної частини проєкту, інсталяції залежностей за допомогою менеджера `pip`, ініціалізації бази даних у MySQL, запуску FastAPI-сервера (додаток Д).

Для перевірки працездатності розробленої системи було реалізовано тестовий сценарій, у якому імітується передача діагностичних даних з адаптера на сервер. У результаті запуску цього сценарію відбувається

оновлення значень у таблицях `live_data` та `historical_data`, формування відповідного повідомлення (у разі виходу параметра за межі допустимого діапазону), а також генерація короткої рекомендації для користувача.

3.4 Результати реалізації інформаційної системи

У результаті реалізації ІС з контролю стану комплектуючих автомобіля було запропоновано та реалізовано ряд оригінальних технічних і програмних рішень, що формують основу імітаційної інформаційної системи для контролю стану комплектуючих автомобіля.

Розроблена система орієнтована на індивідуальне використання власниками транспортних засобів і забезпечує локальний аналіз даних, зібраних із діагностичного інтерфейсу OBD-II без потреби у підключенні до хмарних сервісів або зовнішніх серверів.

Головне меню мобільного застосунку складається з таких сторінок (рис. 3.7):

1. головна сторінка;
2. сторінка поточні показники;
3. сторінка архівні показники;
4. сторінка налаштування з'єднання;
5. сторінка аналіз та рекомендації;
6. сторінка помилки та сповіщення.

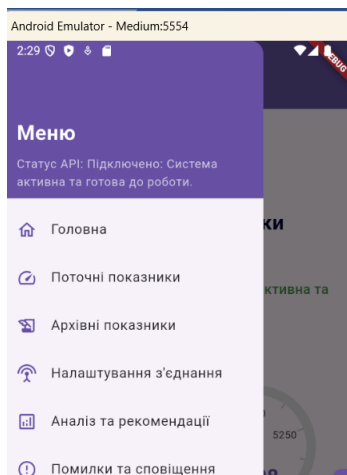


Рисунок 3.7 – Меню мобільного застосунку

На головній сторінці мобільного застосунку відображаються ключові показники автомобіля, такі як швидкість, оберти двигуна, рівень палива та запас ходу. Дані оновлюються в режимі реального часу, забезпечуючи користувачеві оперативну інформацію про стан транспортного засобу (рис. 3.8).



Рисунок 3.8 – Головна сторінка мобільного застосунку

Сторінка "Поточні показники" надає деталізовану інформацію про різні параметри автомобіля, включаючи оберти двигуна, температуру охолоджуючої рідини, навантаження на двигун, напругу акумулятора, рівень палива, швидкість автомобіля, тиск масла та положення дросельної заслінки (рис. 3.9).

Сторінка підтримує функцію оновлення та авто-оновлення даних кожні 0.5 секунди, для забезпечення користувача актуальною інформацією про стан комплектуючих автомобіля. Коли користувач вмикає цю функцію то змінюється логотип (рис. 3.10) та система формує сповіщення «Автооновлення увімкнено» (рис. 3.11).

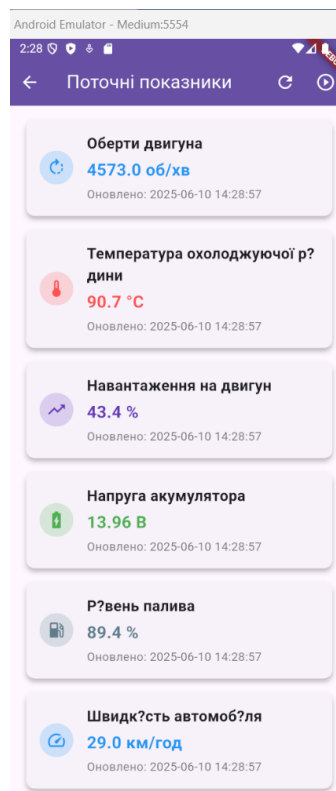


Рисунок 3.9 – Сторінка поточні показники

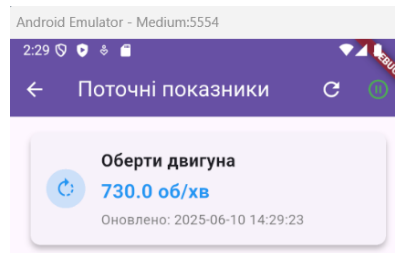


Рисунок 3.10– Увімкнена функція авто-оновлення

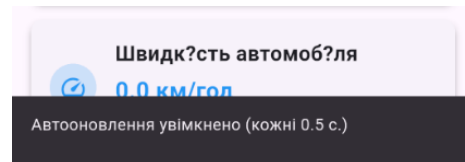


Рисунок 3.11 – Повідомлення про активацію авто-оновлення

На сторінці Архівні показники користувач має можливість переглядати архівні дані (рис. 3.12). Для цього потрібно вибрати компонент автомобіля (рис. 3.13) та його параметр (рис. 3.14). Після вибору компонента та параметра, а також зазначення початкової та кінцевої дати (рис. 3.15), система буде відповідний графік системи автомобіля з вибраним параметром за обраний проміжок часу (рис. 3.16).

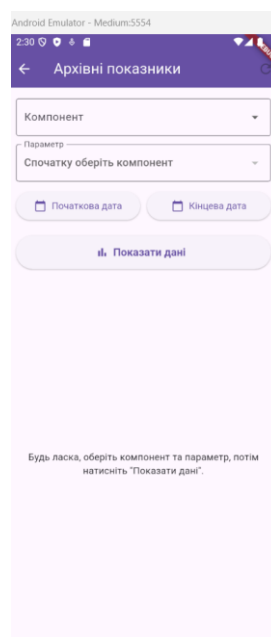


Рисунок 3.12 – Сторінка архівні показники

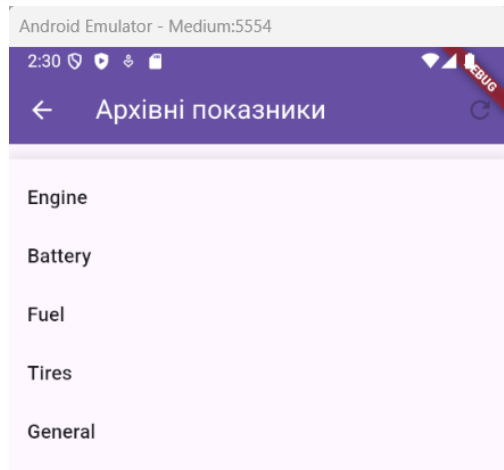


Рисунок 3.13 – Вибір компонентів автомобіля

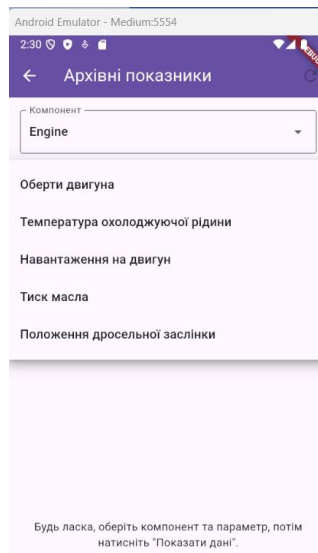


Рисунок 3.14 – Вибір параметру компонента

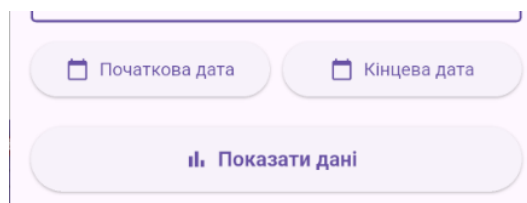


Рисунок 3.15 – Вибір часового проміжку



Рисунок 3.16 – Графік комплектуючої «Engine» з параметром «Оберти двигуна»

На сторінці "Налаштування з'єднання" користувач може перевірити поточний статус підключення до серверу API (рис. 3.17) системи та за потреби змінити IP-адресу серверу. При коректному підключенні до серверу система надає повідомлення про успішне підключення (рис. 3.19). На сторінці присутній функціонал перевірки поточного з'єднання при використанні якого система перевірить коректність підключення, якщо підключення коректне система виведе повідомлення «Підключено. Система активна та готова до роботи» (рис. 3.18).

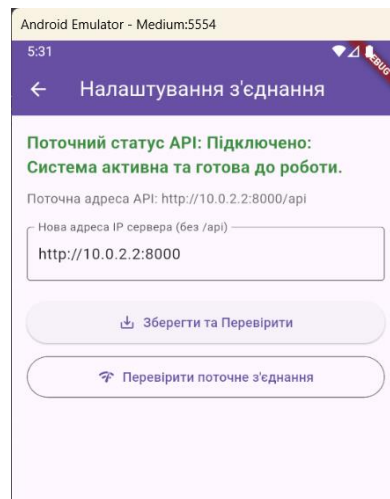


Рисунок 3.17 – Сторінка налаштування з'єднання

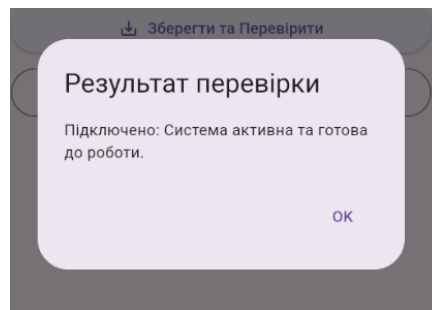


Рисунок 3.18 – Перевірка поточного з'єднання

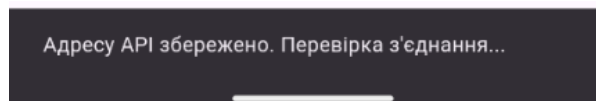


Рисунок 3.19 – Сповіднення про вдале підключення до серверу

Сторінка "Аналіз та рекомендації" дозволяє генерувати нові звіти про стан систем автомобіля та переглядати існуючі (рис. 3.20). Користувач отримує рекомендації щодо обслуговування або перевірки компонентів, що ґрунтуються на аналізі виявлених аномалій та відхилень від норми (рис. 3.20). Рекомендації містять інформацію про рівень критичності такий як (Low, Medium, High) та комплектуючі з їхніми параметрами котрі мають недоліки. Згенеровані звіти можна переглянути та скопіювати в текстовому вигляді (рис. 3.22). Звіт має інформацію про стан комплектуючих автомобіля (додаток Г).

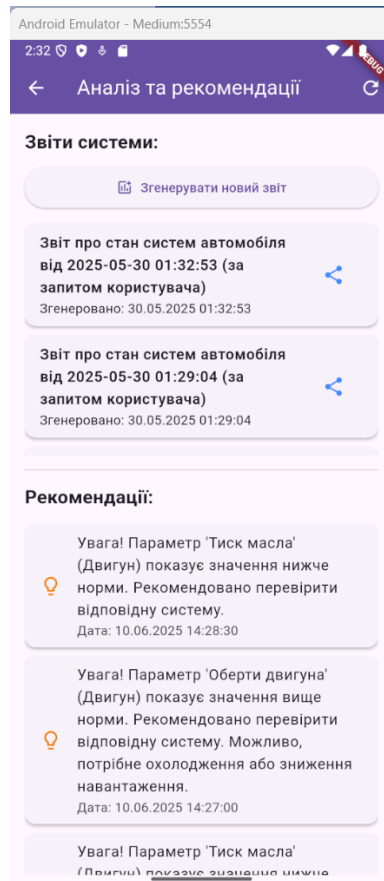


Рисунок 3.20 – Сторінка аналіз та рекомендації

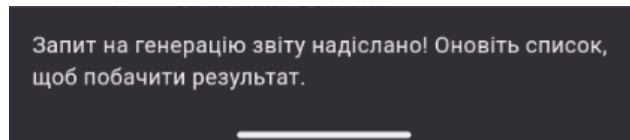


Рисунок 3.21 – Відповідь системи при генеруванні нового звіту

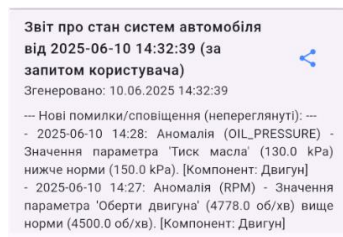


Рисунок 3.22 – Вигляд звіту про стан комплектуючих

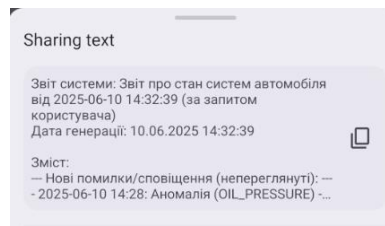


Рисунок 3.23 – Копіювання вмісту звіту

Сторінка «Помилки та сповіщення» відображає всі виявлені помилки та аномалії (рис. 3.24). Системою передбачено такий функціонал для цієї сторінки - користувач може фільтрувати лише нові сповіщення, позначати їх як вирішені (рис. 3.27), увімкнути функцію авто-оновлення помилок (рис. 3.25), відображати певну кількість помилок (рис. 3.24), переглянути деталізовану інформацію про кожну помилку (рис. 3.30) та позначати обрану помилку вирішеною (рис. 3.31).

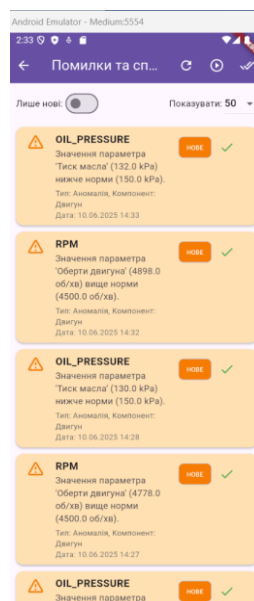


Рисунок 3.24– Сторінка помилки та сповіщення



Рисунок 3.25 – Функціонал авто-оновлення та позначення прочитані для всіх помилок

Автооновлення увімкнено (кожні 10 с.)

Рисунок 3.26 – Відповідь системи при увімкненні авто-оновлення

Всі помилки позначено як вирішені.

Рисунок 3.28 – Відповідь системи при натисненні прочитати всі



Рисунок 3.27 – Вигляд записів вирішених помилок



Рисунок 3.29 – Результат увімкнення «Лише нові»

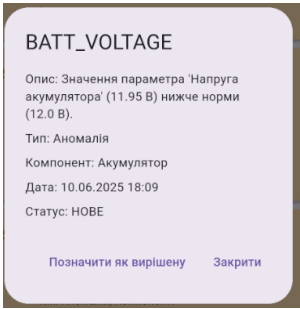


Рисунок 3.30 – Деталізована інформація про помилку



Рисунок 3.31 – Вигляд тільки однієї вирішеної помилки

В результаті проектування та реалізації компонентів ІС було створено функціональний прототип мобільного додатку для апаратно-програмного комплексу для моніторингу стану комплектуючих автомобіля. Апробація на контрольному прикладі, що імітує реальні умови експлуатації за допомогою імітації ODB-II, підтвердила коректність роботи ключових модулів: збору, аналізу та збереження даних. Система продемонструвала здатність виявляти аномалії у вхідних показниках та генерувати відповідні сповіщення й рекомендації, що було основною метою розробки.

ВИСНОВКИ

У ході виконання кваліфікаційної бакалаврської роботи було успішно спроектовано та реалізовано ІС з контролю стану комплектуючих автомобіля. Розроблена система забезпечує комплексний моніторинг технічних параметрів транспортного засобу, що сприяє підвищенню безпеки його експлуатації та оптимізації витрат на обслуговування.

Відповідно до поставлених завдань були отримані такі результати:

Проведено детальний аналіз предметної галузі та існуючих на ринку рішень для діагностики автомобілів. Встановлено, що більшість доступних систем мають обмеження, такі як залежність від хмарних сервісів, висока вартість або недостатній функціонал для глибокого аналізу даних. Це обґрунтувало актуальність розробки автономної, доступної та функціональної ІС.

Сформульовано та специфіковано функціональні й нефункціональні вимоги до системи. Було визначено ключові можливості: збір даних через OBD-II, їх локальне зберігання та аналіз, візуалізація поточних та архівних показників, формування звітів та рекомендацій, а також надання сповіщень користувачеві.

Розроблено комплексну архітектуру ІС з використанням сучасних підходів моделювання. За допомогою діаграм UML та SysML було змодельовано поведінку системи та її статичну структуру, що забезпечило логічну цілісність та прозорість проектних рішень.

Інформаційне забезпечення: розроблено інфологічну та даталогічну моделі реляційної бази даних під управлінням MySQL, що забезпечує структуроване та надійне зберігання даних.

Технічне забезпечення: обґрунтовано вибір апаратної платформи на базі мікрокомп'ютера Raspberry Pi, адаптера OBD-II та персонального смартфона як автоматизованого робочого місця користувача.

Програмне забезпечення: визначено структуру ПЗ, що складається з серверної частини на Python з використанням фреймворку FastAPI та мобільного додатку на Flutter.

Особистий внесок автора полягає у розробці архітектури системи, проектуванні бази даних, реалізації програмних модулів серверної частини для збору, обробки та аналізу даних, а також у створенні клієнтського мобільного додатку з інтуїтивно зрозумілим інтерфейсом. Було реалізовано функціонал для відображення даних у реальному часі, побудови інтерактивних графіків на основі архівних показників, генерації звітів та системи сповіщень про виявлені аномалії.

Проведена апробація на контрольному прикладі з імітацією даних від OBD-II підтвердила працездатність системи. ІС продемонструвала здатність коректно збирати, зберігати та аналізувати показники, виявляти відхилення від нормативних значень та формувати відповідні рекомендації для користувача.

Перевагами розробленої системи є її повна автономність, що підвищує безпеку та конфіденційність даних, а також доступність, оскільки вона побудована на відкритих та недорогих програмних і апаратних засобах.

Рекомендації щодо подальшого вдосконалення:

Інтеграція алгоритмів машинного навчання для більш точного прогнозування несправностей та виявлення прихованих закономірностей у даних.

Розширення підтримки різних діагностичних протоколів для забезпечення сумісності з більшою кількістю моделей автомобілів.

Подальший розвиток користувацького інтерфейсу, додавання кращого інтерфейсі та розширення аналітичних можливостей на основі відгуків користувачів.

Таким чином, у рамках кваліфікаційної бакалаврської роботи було створено завершене програмно-апаратне рішення, яке відповідає сучасним вимогам до інформаційних систем та має значний практичний потенціал для широкого кола автовласників.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційні системи і технології в економіці: навч. посіб. - М.А. Сендзюка, М.Б. Вітер К. :КНЕУ, 2011. – 422с.
2. Автомобільна промисловість в Україні: перспективи розвитку [Електронний ресурс]. Режим доступу до ресурсу: <https://novaton.ua/news/388?srsltid=AfmBOop-iF3LVaAx5FRuURyAbqTytQCKU6OeXTD5nd162WnSmvwyEpYq> (дата звернення: 10.05.2025).
3. OBD on UDS - SAE J1979-2 [Електронний ресурс]. Режим доступу до ресурсу: <https://www.vector.com/gb/en/products/solutions/diagnostic-standards/obdonuds/> (дата звернення: 10.05.2025).
4. ISO 15031-5:2015. Road vehicles - Communication between vehicle and external equipment for emissions-related diagnostics - Part 5: Emissions-related diagnostic services [Електронний ресурс]. Режим доступу до ресурсу: <https://www.iso.org/standard/66368.html> (дата звернення: 10.05.2025).
5. Using obd-2 technology for vehicle diagnostic and using it in the information system [Електронний ресурс]. Режим доступу до ресурсу: https://www.researchgate.net/publication/372436756_USING_OBD-2_TECHNOLOGY_FOR_VEHICLE_DIAGNOSTIC_AND_USING_IT_IN_THE_INFORMATION_SYSTEM (дата звернення: 10.05.2025).
6. Raspberry Pi [Електронний ресурс]. Режим доступу до ресурсу: <https://www.raspberrypi.com/> (дата звернення: 10.05.2025)

7. Vehicle and Driver Monitoring System Using On-Board and Remote Sensors [Електронний ресурс]. Режим доступу до ресурсу: https://www.researchgate.net/publication/367009497_Vehicle_and_Driver_Monitoring_System_Using_On-Board_and_Remote_Sensors (дата звернення: 10.05.2025)
8. Torque Pro – система діагностики стану автомобіля [Електронний ресурс]. Режим доступу до ресурсу: <https://torque-bhp.com/> (дата звернення: 10.05.2025)
9. CarMD – система діагностики стану автомобіля [Електронний ресурс]. Режим доступу до ресурсу: <https://carmd.com/> (дата звернення: 10.05.2025)
10. OBD Auto Doctor – система діагностики стану автомобіля [Електронний ресурс]. Режим доступу до ресурсу: <https://www.obdautodoctor.com/> (дата звернення: 10.05.2025)
11. Applications of Machine Learning Techniques for Fault Diagnosis of UAVs - [Електронний ресурс]. Режим доступу до ресурсу: https://www.researchgate.net/publication/369452621_Applications_of_Machine_Learning_Techniques_for_Fault_Diagnosis_of_UAVs (дата звернення: 10.05.2025)
12. Smart vehicle management by using sensors and an IoT based black box - [Електронний ресурс]. Режим доступу до ресурсу: https://www.researchgate.net/publication/364335124_Smart_vehicle_management_by_using_sensors_and_an_IoT_based_black_box (дата звернення: 10.05.2025)
13. OnStar - [Електронний ресурс]. Режим доступу до ресурсу: (дата звернення: <https://www.onstar.com/> 10.05.2025)
14. BMW Connected Drive - [Електронний ресурс]. Режим доступу до ресурсу: <https://bmw.dp.ua/bmw-connecteddrive> (дата звернення: 10.05.2025)

15. Mercedes me - [Електронний ресурс]. Режим доступу до ресурсу: <https://www.mercedes-benz.ua/services/digital-extras-overview/> (дата звернення: 10.05.2025)
16. Внутрішні стейкхолдери - [Електронний ресурс]. Режим доступу до ресурсу: <https://brainrain.com.ua/uk/hto-taki-steykholderi-ua/> (дата звернення: 10.05.2025)
17. Бізнес вимоги - [Електронний ресурс]. Режим доступу до ресурсу: <https://www.ba.in.ua/2023/11/24/shho-take-dokument-biznes-vymog-business-requirements-document/> (дата звернення: 10.05.2025)
18. EA User Guide (pdf) (Інструкція використання Enterprise Architect) [Електронний ресурс]. Режим доступу до ресурсу: <https://sparxsystems.com/bin/EAUserGuide.pdf> (дата звернення: 03.05.2025)
19. Example Traceability Diagram (Інструкція використання Enterprise Architect) [Електронний ресурс]. Режим доступу до ресурсу: https://sparxsystems.com/enterprise_architect_user_guide/17.1/the_application_desktop/create_traceability_diagrams.html (дата звернення: 03.05.2025)
20. Інформаційне забезпечення [Електронний ресурс]. Режим доступу до ресурсу: <https://studfile.net/preview/7789985/page:6/> (дата звернення: 03.05.2025)

ДОДАТКИ

Додаток А

Таблиці класифікаторів інформаційної системи

Таблиця А.1 - Класифікатор комплектуючих автомобіля

№	Назва компонента	Код	Довжина	Метод кодування
1	Двигун	ENG	3	Абревіатура
2	Акумулятор	BATT	4	Абревіатура
3	Паливна система	FUEL	4	Абревіатура
4	Трансмсія	TRANSM	6	Абревіатура
5	Загальні	GEN	3	Абревіатура

Джерело: сформовано автором на основі виконаного дослідження

Таблиця А.2 - Класифікатор показників

№	Назва параметра	Код	Компонент	Формат	Довжина
1	Температура охолоджуючої рідини	ENG_TEMP	ENG	float (°C)	8
2	Оберти двигуна	RPM	ENG	int	3
3	Напруга акумулятора	BATT_VOLTAGE	BATT	float (V)	13
4	Рівень палива	FUEL_LEVEL	FUEL	float (%)	11
5	Положення дросельної заслінки	THROTTLE_POS	ENG	float (%)	13

Джерело: сформовано автором на основі виконаного дослідження

Таблиця А.3 - Класифікатор типів повідомлень

№	Тип повідомлення	Код	Опис повідомлення
1	Помилка	ERR	Виявлено технічну несправність
2	Сповіщення	NOTIFY	Інформаційне повідомлення
3	Попередження	WARN	Потенційна небезпека
4	Аномалія	ANOM	Відхилення від нормальної поведінки

Джерело: сформовано автором на основі виконаного дослідження

Таблиця А.4 - Коди звітів та документів

№	Назва документа	Код	Формат	Призначення
1	Звіт про технічний стан	RPT_STATUS	текст	Основний технічний звіт
2	Рекомендація системи	SYS_RECOMMEND	текст	Система пропонує дії користувачу
3	Графік параметрів	LIVE_GRAPH	графік	Візуалізація зміни показників
4	Повідомлення про помилку	ERR_NOTIFY	push	Оповіщення про критичну несправність

Джерело: сформовано автором на основі виконаного дослідження

Додаток Б

Таблиці структури документів інформаційної системи

Таблиця Б.1 - Структура звіту про технічний стан (RPT_STATUS)

№	Назва поля	Ідентифікатор	Тип даних	Опис	Обов'язкове
1	Дата та час створення	timestamp	datetime	Час створення звіту	Так
2	Заголовок	title	string	Назва документа (типово: «Звіт про стан автомобіля»)	Так
3	Значення параметрів	parameter_values	JSON/table	Поточні значення всіх показників	Так
4	Виявлені аномалії	anomalies	string/list	Коди або назви відхилень	Ні
5	Рекомендації	recommendations	text	Текстові поради системи	Ні
6	Графіки параметрів	charts	image/graph	Графіки зміни показників (історія, порівняння)	Ні

Джерело: сформовано автором на основі виконаного дослідження

Таблиця Б.2 - Структура повідомлення про помилку (ERR_NOTIFY)

№	Назва поля	Ідентифікатор	Тип даних	Опис	Обов'язкове
1	Дата події	timestamp	datetime	Час фіксації помилки	Так
2	Код помилки	code	string	DTC або внутрішній код	Так
3	Тип повідомлення	type	string	Наприклад, ERR, WARN, ANOM	Так

4	Компонент	component	string	Назва або ідентифікатор компонента	Ні
5	Опис	description	text	Опис помилки	Так
6	Рівень критичності	criticality	ENUM	Low, Medium, High	Ні

Джерело: сформовано автором на основі виконаного дослідження

Таблиця Б.3 - Структура графіка параметра (LIVE_GRAPH)

№	Назва поля	Ідентифікатор	Тип даних	Опис	Обов'язкове
1	Код параметра	parameter_code	string	Ідентифікатор показника (наприклад, ENG_TEMP)	Так
2	Дані	data_series	array/JSON	Масив значень з часом	Так
3	Тип графіка	chart_type	string	Наприклад: line, bar, multi-line	Так
4	Період побудови	time_range	datetime[]	Проміжок часу побудови графіка	Так
5	Одиниця виміру	unit	string	Наприклад, °C, В, %	Ні

Джерело: сформовано автором на основі виконаного дослідження

Таблиця Б.4 - Структура рекомендації (SYS_RECOMMEND)

№	Назва поля	Ідентифікатор	Тип даних	Опис	Обов'язкове
1	Дата рекомендації	timestamp	datetime	Дата формування рекомендації	Так
2	Текст рекомендації	content	text	Повна порада або інструкція	Так

3	Параметр	related_param	string	Код параметра, що спричинив рекомендацію	Ні
4	Критичність	criticality	ENUM	Low, Medium, High	Так

Джерело: сформовано автором на основі виконаного дослідження

Додаток В**Структура бази даних та тестові запити**

```
CREATE DATABASE IF NOT EXISTS car_diag_db CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

```
USE car_diag_db;
```

```
CREATE TABLE IF NOT EXISTS components (  
  
    id INT AUTO_INCREMENT PRIMARY KEY,  
  
    name VARCHAR(255) NOT NULL UNIQUE COMMENT 'Назва компонента'  
  
) COMMENT 'Довідник компонентів автомобіля';
```

```
CREATE TABLE IF NOT EXISTS parameters (  
  
    id INT AUTO_INCREMENT PRIMARY KEY,  
  
    component_id INT,  
  
    name VARCHAR(255) NOT NULL COMMENT 'Назва параметра',  
  
    code VARCHAR(50) NOT NULL UNIQUE COMMENT 'Унікальний код параметра',  
  
    unit VARCHAR(50) COMMENT 'Одиниця виміру',  
  
    description TEXT COMMENT 'Детальний опис параметра',  
  
    min_normal_value FLOAT COMMENT 'Мінімальне нормальне значення',  
  
    max_normal_value FLOAT COMMENT 'Максимальне нормальне значення',  
  
    FOREIGN KEY (component_id) REFERENCES components(id) ON DELETE SET NULL  
  
) COMMENT 'Довідник параметрів';
```

```
CREATE TABLE IF NOT EXISTS live_data (  
  
    parameter_id INT PRIMARY KEY,
```

```

value VARCHAR(255) NOT NULL COMMENT 'Поточне значення параметра',

timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

FOREIGN KEY (parameter_id) REFERENCES parameters(id) ON DELETE CASCADE
) COMMENT 'Поточні (живі) дані з датчиків';

CREATE TABLE IF NOT EXISTS historical_data (

id BIGINT AUTO_INCREMENT PRIMARY KEY,

parameter_id INT NOT NULL,

value VARCHAR(255) NOT NULL,

timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY (parameter_id) REFERENCES parameters(id) ON DELETE CASCADE,

INDEX idx_historical_data_timestamp (timestamp) -- Індекс для прискорення вибірки за часом
) COMMENT 'Архівні дані з датчиків';

CREATE TABLE IF NOT EXISTS error_logs (

id INT AUTO_INCREMENT PRIMARY KEY,

timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,

code VARCHAR(50) COMMENT 'Код помилки (наприклад, OBD2 код P0300)',

description TEXT NOT NULL,

type VARCHAR(100) COMMENT 'Тип: "Помилка", "Сповіщення", "Аномалія"',

component_name VARCHAR(255) COMMENT 'Компонент, з яким пов'язана помилка (для сумісності)',

component_id INT NULL COMMENT 'ID компонента, з яким пов'язана помилка',

is_new BOOLEAN DEFAULT TRUE,

FOREIGN KEY (component_id) REFERENCES components(id) ON DELETE SET NULL

```

) COMMENT 'Журнал помилок, аномалій та сповіщень';

CREATE TABLE IF NOT EXISTS reports (

id INT AUTO_INCREMENT PRIMARY KEY,

timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,

title VARCHAR(255) NOT NULL DEFAULT 'Звіт про стан систем',

content TEXT NOT NULL,

related_error_log_id INT NULL COMMENT 'ID головної помилки, до якої відноситься звіт (необов"язково)',

FOREIGN KEY (related_error_log_id) REFERENCES error_logs(id) ON DELETE SET NULL

) COMMENT 'Згенеровані звіти про стан автомобіля';

CREATE TABLE IF NOT EXISTS recommendations (

id INT AUTO_INCREMENT PRIMARY KEY,

timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,

content TEXT NOT NULL,

related_parameter_code VARCHAR(50) COMMENT 'Код параметра (для сумісності)',

parameter_id INT NULL COMMENT 'ID параметра, на основі якого надано рекомендацію',

error_log_id INT NULL COMMENT 'ID помилки, що викликала цю рекомендацію',

criticality ENUM('Low', 'Medium', 'High') DEFAULT 'Low',

FOREIGN KEY (parameter_id) REFERENCES parameters(id) ON DELETE SET NULL,

FOREIGN KEY (error_log_id) REFERENCES error_logs(id) ON DELETE SET NULL

) COMMENT 'Рекомендації щодо обслуговування або перевірки';

INSERT INTO components (name) VALUES

('Двигун'), ('Акумулятор'), ('Трансмiсія'), ('Паливна система'), ('Загальнi')

ON DUPLICATE KEY UPDATE name=name;

INSERT INTO parameters (component_id, name, code, unit, description, min_normal_value, max_normal_value) VALUES

((SELECT id FROM components WHERE name='Двигун'), 'Оберти двигуна', 'RPM', 'об/хв', 'Частота обертання колінчастого валу двигуна', 600, 4500),

((SELECT id FROM components WHERE name='Двигун'), 'Температура охолоджуючої рідини', 'ENG_TEMP', '°C', 'Температура охолоджуючої рідини двигуна', 75, 105),

((SELECT id FROM components WHERE name='Двигун'), 'Навантаження на двигун', 'ENG_LOAD', '%', 'Розрахункове навантаження на двигун', 0, 90),

((SELECT id FROM components WHERE name='Акумулятор'), 'Напруга акумулятора', 'BATT_VOLTAGE', 'V', 'Напруга на клеммах акумулятора', 12.0, 14.8),

((SELECT id FROM components WHERE name='Паливна система'), 'Рiвень палива', 'FUEL_LEVEL', '%', 'Залишок палива у баку', 5, 100),

((SELECT id FROM components WHERE name='Загальнi'), 'Швидкiсть автомобiля', 'VEHICLE_SPEED', 'км/год', 'Поточна швидкiсть руху', 0, 250),

((SELECT id FROM components WHERE name='Двигун'), 'Тиск масла', 'OIL_PRESSURE', 'кПа', 'Тиск масла в системi змащення двигуна', 150, 450),

((SELECT id FROM components WHERE name='Двигун'), 'Положення дросельної заслінки', 'THROTTLE_POS', '%', 'Вiдсоткове вiдкриття дросельної заслінки', 0, 100)

ON DUPLICATE KEY UPDATE name=VALUES(name);

INSERT INTO error_logs (code, description, type, component_name, is_new) VALUES

('P0171', 'Система надто бiдна (Банк 1)', 'Помилка', 'Паливна система', TRUE);

INSERT INTO recommendations (content, related_parameter_code, criticality) VALUES

('Перевiрте рiвень охолоджуючої рiдини.', 'ENG_TEMP', 'Medium');

Додаток Г**Приклад звіту про стан систем автомобіля**

Звіт системи: Звіт про стан систем автомобіля від 2025-06-10 14:32:39 (за запитом користувача)

Дата генерації: 10.06.2025 14:32:39

Зміст:

--- Нові помилки/сповіщення (непереглянуті): ---

- 2025-06-10 14:28: Аномалія (OIL_PRESSURE) - Значення параметра 'Тиск масла' (130.0 kPa) нижче норми (150.0 kPa). [Компонент: Двигун]

- 2025-06-10 14:27: Аномалія (RPM) - Значення параметра 'Оберти двигуна' (4778.0 об/хв) вище норми (4500.0 об/хв). [Компонент: Двигун]

- 2025-06-10 14:23: Аномалія (OIL_PRESSURE) - Значення параметра 'Тиск масла' (148.0 kPa) нижче норми (150.0 kPa). [Компонент: Двигун]

- 2025-06-10 14:21: Аномалія (RPM) - Значення параметра 'Оберти двигуна' (4998.0 об/хв) вище норми (4500.0 об/хв). [Компонент: Двигун]

- 2025-06-10 14:21: Аномалія (ENG_LOAD) - Значення параметра 'Навантаження на двигун' (91.7 %) вище норми (90.0 %). [Компонент: Двигун]

--- Останні зареєстровані події: ---

- 2025-06-10 14:28: Аномалія (OIL_PRESSURE) - Значення параметра 'Тиск масла' (130.0 kPa) нижче норми (150.0 kPa). [Компонент: Двигун]

- 2025-06-10 14:27: Аномалія (RPM) - Значення параметра 'Оберти двигуна' (4778.0 об/хв) вище норми (4500.0 об/хв). [Компонент: Двигун]

- 2025-06-10 14:23: Аномалія (OIL_PRESSURE) - Значення параметра 'Тиск масла' (148.0 kPa) нижче норми (150.0 kPa). [Компонент: Двигун]

- 2025-06-10 14:21: Аномалія (RPM) - Значення параметра 'Оберти двигуна' (4998.0 об/хв) вище норми (4500.0 об/хв). [Компонент: Двигун]

- 2025-06-10 14:21: Аномалія (ENG_TEMP) - Значення параметра 'Температура охолоджуючої рдини' (20.5 °C) нижче норми (75.0 °C). [Компонент: Двигун]

--- Поточні значення ключових параметрів: ---

- Оберти двигуна (RPM): 753 об/хв (станом на 14:32:37)

- Температура охолоджуючої рдини (ENG_TEMP): 95.1 °C (станом на 14:32:37)

- Напруга акумулятора (BATT_VOLTAGE): 14.1 В (станом на 14:32:37)

- Швидкість автомобіля (VEHICLE_SPEED): 0.7 км/год (станом на 14:32:37)
- Рівень палива (FUEL_LEVEL): 89.2 % (станом на 14:32:37)
- Тиск масла (OIL_PRESSURE): 105 kPa (станом на 14:32:37)
- Навантаження на двигун (ENG_LOAD): 10.1 % (станом на 14:32:37)
- Положення дросельної заслінки (THROTTLE_POS): 2.4 % (станом на 14:32:37)
- Параметр CURRENT_GEAR: не знайдено в системі
- Параметр ODOMETER: не знайдено в системі

--- Середні значення за сьогодні (з 00:00): ---

- Оберти двигуна (RPM): 2287.96 об/хв
- Температура охолоджуючої рідини (ENG_TEMP): 90.20 °C
- Навантаження на двигун (ENG_LOAD): 23.37 %
- Напруга акумулятора (BATT_VOLTAGE): 13.89 В
- Рівень палива (FUEL_LEVEL): 82.63 %
- Швидкість автомобіля (VEHICLE_SPEED): 17.72 км/год
- Тиск масла (OIL_PRESSURE): 195.48 kPa
- Положення дросельної заслінки (THROTTLE_POS): 20.55 %

--- Останні рекомендації: ---

- 2025-06-10 14:28: [Критичність: Medium] Увага! Параметр 'Тиск масла' (Двигун) показує значення нижче норми. Рекомендовано перевірити відповідну систему. (Параметр: OIL_PRESSURE)

- 2025-06-10 14:27: [Критичність: High] Увага! Параметр 'Оберти двигуна' (Двигун) показує значення вище норми. Рекомендовано перевірити відповідну систему. Можливо, потрібне охолодження або зниження навантаження. (Параметр: RPM)

- 2025-06-10 14:23: [Критичність: Medium] Увага! Параметр 'Тиск масла' (Двигун) показує значення нижче норми. Рекомендовано перевірити відповідну систему. (Параметр: OIL_PRESSURE)

Додаток Д

Інструкції для налаштування програмного середовища

```
# Встановлення залежностей Python
pip install fastapi uvicorn mysql-connector-python "pydantic[email]"
pip install requests

# Налаштування змінних середовища для підключення до БД
DB_USER = os.getenv("DB_USER", "your_mysql_user")
DB_PASSWORD = os.getenv("DB_PASSWORD", "your_mysql_password")




# Запуск Android емулятора та застосунку Flutter
flutter emulators --launch medium_phone
flutter run -d emulator-5554

# Запуск API-сервера FastAPI
python -m uvicorn main_api:app --reload --port 8000


# Запуск генератора даних (імітація роботи авто)
python car_simulator.py
```

Додаток Е

Звіт подібності

Дата звіту: 6/13/2025
 Дата редагування: —


Звіт не був оцінений

Звіт подібності

метадані

Назва організації
Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок:
Проектування інформаційної системи з контролю стану комплектуючих автомобіля

Автор: Науковий керівник / Експерт
Франчук Микола Костянтинович Тішков Б.О.

підрозділ:
кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

3.01%
3.01%

КП 1

0.78%
0.78%

КП 2

0.65%
0.65%

КЦ

25

Довжина фрази для коефіцієнта подібності 2

13454






Кількість слів

106110

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		0
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		29

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА (URL (НАЗВА БАЗИ))	Копія тексту
		кількість ідентичних слів (фрагментів)
1	https://ir.kneu.edu.ua/server/api/core/bitstreams/1b35cb3a-b293-4677-9674-83983838c914/content	43 0.32 %
2	https://ir.kneu.edu.ua/bitstreams/958c3965-af04-49ca-8d39-b0213a87d87a/download	31 0.23 %
3	https://ir.kneu.edu.ua/bitstreams/a5c7419f-c089-4b0c-8e70-3923cf3566a2/download	31 0.23 %
4	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	24 0.18 %
5	https://ir.kneu.edu.ua/bitstreams/066dbca0-6afa-4cf4-a181-c3e88a2aab6f/download	23 0.17 %

Рисунок Г.1 – Звіт подібності