

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему АВТОМАТИЗОВАНА СИСТЕМА МОНІТОРИНГУ
ТРОЛЕЙБУСНОГО ПАРКУ

Виконав: студент 2 курсу, групи КНТ-513м
спеціальності _____

123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні системи та мережі

ФІЛППЕНКОВ Д. Ю.

(ПРИЗВИЩЕ та ініціали)

Керівник ТЯГУНОВА М.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЗІНА Г.Л.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти магістерський
 Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
 Освітня програма (спеціалізація) «Комп'ютерні системи та мережі»
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Зав. кафедри Кудерметов Р.К.

 “ _____ ” _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ФІЛІППЕНКОВА Данііла Юрійовича
(ПРИЗВИЩЕ, ім'я, по батькові)

- Тема проєкту (роботи) Автоматизована система моніторингу
тролейбусного парку
 керівник проєкту (роботи) к. т. н., доцент, ТЯГУНОВА Марія Юріївна
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)
 затверджені наказом вищого навчального закладу від “18” жовтня 2024 № 149
- Строк подання студентом проєкту (роботи) 10 грудня 2024 року
- Вихідні дані до проєкту (роботи) мова програмування C#; середовище
розробки Rider; веб-фреймворки ASP.NET Core і React; персональний
комп'ютер із процесором Intel Core i7 на базі операційної системи
Windows 10.
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____
 - Аналіз предметної області.
 - Розробка програмного забезпечення;
 - Реалізація системи;
 - Керівництво програміста;
 - Керівництво оператора.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ТЯГУНОВА М. Ю., доцент		
нормоконтроль	ЩЕРБАК Н.В., викл.		

7. Дата видачі завдання 01.10.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз предметної області, визначення вимог до системи моніторингу тролейбусного парку	10.10.2024 р.	
2	Розробка структури системи	15.10.2024 р.	
3	Розробка алгоритму роботи системи	20.10.2024 р.	
4	Реалізація системи	05.10.2024 р.	
5	Дослідження системи автоматизованого моніторингу тролейбусного парку	20.11.2024 р.	
6	Оформлення отриманих результатів у ПЗ	25.11.2024 р.	
7	Оформлення графічного матеріалу	01.12.2024 р.	
8	Оформлення допоміжного матеріалу	10.12.2024 р.	

Студент _____ **Данііл ФІЛІПШЕНКОВ**
(підпис) (ім'я, ПРИЗВИЩЕ)

Керівник проєкту (роботи) _____ **Марія ТЯГУНОВА**
(підпис) (ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 84 с., 1 табл., 26 рис., 15 джерел.

АВТОМАТИЗАЦІЯ, УПРАВЛІННЯ, ТРАНСПОРТ, ТРОЛЕЙБУСИ,
ASP.NET CORE, REACT, MYSQL.

Об'єкт дослідження – процес автоматизації управління міським транспортом.

Предмет дослідження – методи розробки автоматизованих систем для моніторингу та управління транспортними засобами.

Метою даної роботи є підвищення рівня якості обслуговування тролейбусного парку за рахунок автоматизованого контролю та моніторингу технічного стану тролейбусів.

У першій частині проведено аналіз предметної області та визначено вимоги і ключові характеристики тролейбуса і тролейбусного парку.

У другій частині описано загальну структуру розроблюваної системи. Обґрунтовано вибір архітектурного підходу та патерну проєктування для забезпечення модульності та масштабованості системи.

У третій частині представлено реалізацію основних алгоритмів системи, описано структуру класів і бази даних та реалізацію графічного інтерфейсу користувача та обґрунтовано інноваційність підходу до створення системи

У четвертій частині надано технічний опис програми, API серверної частини, структуру звернень до функціоналу, а також повідомлення програмісту для підтримки та розширення функціональності системи.

У п'ятій частині описано призначення програми, умови її використання, а також сценарії та керівництво для роботи операторів.

ABSTRACT

Explanatory note to the master's thesis: 84 pages, 1 tables, 26 figures, 15 sources.

AUTOMATION, MANAGEMENT, TRANSPORT, TROLLEYBUSES, ASP.NET CORE, REACT, MYSQL.

The object of the research is the process of automation in urban transport management.

The subject of the research is methods for developing automated systems for monitoring and managing transport vehicles.

The purpose of the work is improving the service quality of a trolleybus fleet through automated monitoring and control of the technical condition of trolleybuses.

In the first part, the subject area is analyzed, and the requirements and key characteristics of trolleybuses and trolleybus depots are defined.

In the second part, the general structure of the developed system is described. The choice of the architectural approach and design patterns is justified to ensure the system's modularity and scalability.

In the third part, the implementation of the main system algorithms is presented. The structure of classes and the database is described, along with the implementation of the user interface. The innovative approach to creating the system is substantiated.

In the fourth part, a technical description of the software is provided, including the server-side API, the structure of functional calls, and developer notes for maintaining and extending the system's functionality.

In the fifth part, the purpose of the application, its usage conditions, and scenarios, along with guidance for operators, are described.

ЗМІСТ

Перелік скорочень та умовних позначок	8
1 Аналіз предметної області	11
1.1 Визначення і технічний склад тролейбусів.....	11
1.2 Тролейбусні парки.....	12
1.3 Типові проблеми і незручності в обслуговуванні тролейбусів.....	14
1.4 Вимоги до розроблюваної системи	16
1.5 Вибір інструментарію розробки програмного забезпечення	20
2 Проєктування програмного забезпечення	23
2.1 Загальна структура розроблюваної системи.....	23
2.2 Структура серверної частини	25
2.3 Вибір підходу до розробки системи.....	28
2.4 Вибір патерну проєктування.....	30
3 Реалізація системи.....	31
3.1 Реалізація основних алгоритмів системи.....	31
3.2 Опис структури класів системи	45
3.3 Структура бази даних.....	51
3.4 Реалізація графічного інтерфейсу	53
3.5 Інноваційність підходу реалізації системи	54
4 Керівництво програміста	56
4.1 Призначення та умови застосування програми	57
4.2 Характеристики програми	57
4.3 Звертання до програми.....	63
4.4 API-інтерфейс серверної частини.....	64
4.5 Повідомлення програмісту	67
5 Керівництво оператора	68
5.1 Призначення програми.....	68
5.2 Умови використання програми	69

5.3 Сценарії використання програми	69
Висновки.....	82
Перелік джерел посилання	83

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
Backend	– серверна частина додатку;
Frontend	– клієнтська частина додатку;
HTTP	– HyperText Transfer Protocol;
JSON	– JavaScript Object Notation;
JWT	– JSON Web Token;
ООП	– об'єктно орієнтовне програмування;
ТЗ	– технічне завдання;

ВСТУП

Комфорт проживання у тому чи іншому місті значною мірою залежить від транспортної інфраструктури цього міста. Якість і розгалуженість доріг, наявність достатньої кількості зупинок, а також кількість і технічний стан транспорту - визначають ефективність переміщення мешканців і впливають на загальний рівень їх життя.

Тролейбуси відіграють важливу роль у забезпеченні швидкого, екологічно чистого та зручного транспортного сполучення для мешканців міст. Однією з основних переваг троллейбусів є екологічність. Тролейбуси є екологічними, бо працюють за допомогою електроенергії. Така особливість допомагає зменшувати викиди шкідливих речовин, що в свою чергу сприяє боротьбі з забрудненням довкілля і робить троллейбуси важливим інструментом у стратегії сталого розвитку міст. Варто також зазначити, що сучасна і комплексна троллейбусна система може сприяти зменшенню автомобільного трафіку, зниженню кількості заторів і покращенню якості повітря в місті, а розгалужена троллейбусна мережа значно полегшує щоденні переміщення мешканців.

У 21 столітті основним інструментом покращення будь-якої системи є насамперед її автоматизація. Автоматизація в сфері громадського транспорту відкриває нові перспективи для підвищення ефективності та надійності транспортних систем.

Автоматизована система троллейбусного парку - це інтегрований набір програмних та апаратних засобів, які спрямовані на автоматизацію контролю роботи троллейбусів. Система може включати в себе елементи моніторингу технічного стану транспорту, автоматизоване планування маршрутів, контроль за роботою водіїв, а також віддалений моніторинг та діагностику.

Переваги впровадження автоматизованої системи троллейбусного парку є очевидними. Вона дозволяє оперативно реагувати на технічні несправності,

а також забезпечує контроль за витратою електроенергії, що може призвести до економії ресурсів та зниження вартості експлуатації.

Створення автоматизованих систем для тролейбусних парків міст допомагає не лише поліпшити комфорт та доступність громадського транспорту, але й сприяє загальній меті зробити міста більш сталими та придатними для життя. Саме тому створення таких систем сьогодні в пріоритеті. Тому обрана тема магістерської роботи є актуальною.

Метою даної роботи є підвищення рівня якості обслуговування тролейбусного парку за рахунок автоматизованого контролю та моніторингу технічного стану тролейбусів.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз предметної області;
- визначити вимоги до системи моніторингу тролейбусного парку;
- спроектувати систему моніторингу відповідно визначених вимог;
- реалізувати спроектовану систему;
- протестувати систему і проаналізувати отримані результати.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення і технічний склад тролейбусів

Тролейбус є складним комплексом електричних, пневматичних, електронних і механічних компонентів, які разом забезпечують його надійну роботу як міського електротранспорту [1]. Основними компонентами тролейбусної системи є електродвигун, контактна мережа, акумуляторна батарея та система управління.

Електродвигун є ключовим елементом тролейбуса, відповідальним за приведення транспортного засобу в рух [2]. Цей компонент працює на електричній енергії, отриманій від системи живлення. У сучасних тролейбусах зазвичай застосовуються асинхронні електродвигуни змінного струму через їх високу енергоефективність, надійність і тривалий термін служби. Живлення двигуна здійснюється через контактну мережу, яка складається з системи проводів або струмопроводників, розташованих вздовж маршруту [3]. На даху тролейбуса розташований пантограф – механізм, що забезпечує постійний контакт із контактними лініями для передачі електроенергії до транспортного засобу.

Для забезпечення автономного руху [4], наприклад, у зонах без контактної мережі або під час аварійних ситуацій, у тролейбусах встановлюються акумуляторні батареї [5]. Ці батареї складаються з блоків, які у свою чергу включають електрохімічні комірки, здатні зберігати електричний заряд. Це дозволяє тролейбусу долати короткі відстані без підключення до контактної мережі, що значно розширює його можливості.

Система управління є невід'ємною частиною тролейбуса, яка забезпечує ефективну взаємодію між електродвигуном, акумуляторами та іншими компонентами [6]. Вона включає електронні контролери, що регулюють подачу електроенергії до двигуна, датчики, які відстежують робочий стан систем, та інші пристрої для контролю заряду й розряду

аккумуляторних батарей. Завдяки цій системі забезпечується оптимальний режим руху, безперебійна робота та енергоефективність.

Також важливою складовою тролейбусної системи є гальмівна система, яка поєднує електродинамічні, пневматичні й стоянкові гальма. Електродинамічне гальмування дозволяє не тільки ефективно уповільнювати тролейбус, але й здійснювати рекуперацію енергії, повертаючи її до аккумуляторів або контактної мережі, що суттєво знижує споживання електроенергії.

Таким чином, технічний склад тролейбусів включає сучасні та високотехнологічні компоненти, які забезпечують ефективність, екологічність і надійність цього виду транспорту.

1.2 Тролейбусні парки

Тролейбусні парки відіграють ключову роль у забезпеченні безперебійної роботи міського електротранспорту. Це спеціалізовані комплекси, де тролейбуси зберігаються, проходять технічне обслуговування, ремонт, а також підготовку до щоденної експлуатації. Тролейбусний парк складається з низки функціональних зон та обладнання, які забезпечують його ефективну роботу.

Зберігання тролейбусів у парку здійснюється на відкритих або критих майданчиках. Майданчики повинні бути добре освітленими та оснащеними системами відеоспостереження для забезпечення безпеки техніки [7]. У деяких сучасних тролейбусних парках передбачені автоматизовані системи паркування, що дозволяють максимально раціонально використовувати простір.

Технічне обслуговування є одним із ключових завдань тролейбусного парку. Регулярна перевірка технічного стану тролейбусів включає огляд

електродвигуна, контактної мережі, акумуляторів, гальмівної системи, підвіски та інших компонентів. Своєчасна діагностика дозволяє виявити потенційні проблеми на ранніх стадіях, що мінімізує ризик виникнення несправностей під час експлуатації. Окрім того, проводиться очистка кузова, а також перевірка систем вентиляції й опалення для забезпечення комфорту пасажирів.

Ремонтні роботи виконуються у спеціалізованих зонах парку, обладнаних підйомниками, інструментами та діагностичним обладнанням. Тут можуть виконуватися як поточні ремонти, такі як заміна деталей або усунення несправностей, так і капітальні, які включають повну заміну вузлів або модернізацію тролейбуса. У багатьох парках передбачені окремі відділення для роботи з електронними компонентами, такими як контролери, датчики та блоки управління.

Особливу увагу в сучасних тролейбусних парках приділяють моніторингу та контролю рухомого складу. Завдяки інтеграції систем GPS-трекінгу й диспетчерських пунктів забезпечується відстеження місцезнаходження тролейбусів у реальному часі, контроль за дотриманням графіків руху та швидке реагування на позаштатні ситуації. Системи моніторингу також дозволяють збирати дані про технічний стан тролейбусів, що сприяє підвищенню якості обслуговування.

Тролейбусні парки також відіграють важливу роль у підвищенні кваліфікації персоналу. Тут проводяться тренінги для водіїв, механіків і технічних фахівців, що включають як практичні заняття, так і теоретичну підготовку. Усе це сприяє підвищенню безпеки та надійності експлуатації тролейбусів.

Таким чином, тролейбусні парки є невід'ємною складовою міської транспортної інфраструктури. Їх належна організація та функціонування забезпечують високу якість і надійність перевезень, а також сприяють розвитку тролейбусних систем у напрямку підвищення їх екологічності й енергоефективності.

1.3 Типові проблеми і незручності в обслуговуванні тролейбусів

Обслуговування тролейбусів у сучасних умовах є складним завданням, яке ускладнюється низкою організаційних, технічних і кадрових проблем. Застарілі методи управління, ручна обробка інформації, відсутність централізованих систем і нестача автоматизації створюють значні виклики для персоналу, відповідального за підтримку транспортної інфраструктури [8][9]. Нижче детально розглянуто основні труднощі, з якими стикаються служби технічного обслуговування тролейбусів.

1.3.1 Проблеми з оформленням технічної документації

Оформлення технічних актів є важливою частиною обслуговування тролейбусів, однак цей процес супроводжується багатьма труднощами. Перш за все, складання технічних актів вимагає заповнення великої кількості полів вручну. Інженери змушені самостійно вносити дані про стан вузлів, виконані роботи та необхідні ремонтні дії. Через це збільшується час на обробку інформації, а також зростає ризик людських помилок.

Крім того, багато датчиків і компонентів тролейбуса, які потребують регулярного огляду, розташовані у важкодоступних місцях. Це змушує персонал витратити додатковий час на фізичний доступ до цих елементів, що значно ускладнює процес діагностики.

Ще однією суттєвою проблемою є те, що перевірки стану тролейбусів обов'язково вимагають фізичної присутності інженера. Відсутність дистанційного моніторингу не дозволяє оперативно оцінювати стан транспортного засобу, особливо якщо він знаходиться в дорозі чи на віддаленій локації.

Архівування технічних актів також викликає труднощі. Більшість документації зберігається у паперовому вигляді, що унеможливорює швидкий доступ до потрібної інформації. Щоб знайти необхідний технічний акт або

роздрукувати його, працівники повинні звертатися до бухгалтерського відділу або архіву.

Особливу увагу заслуговує суб'єктивність оцінки технічного стану. Через відсутність сучасних автоматизованих систем інженери змушені покладатися на власний досвід та інтуїцію. Такий підхід значно підвищує ризик недооцінки проблем або пропуску ранніх ознак поломок.

1.3.2 Складність доступу до інформації про технічний стан

Оперативний доступ до інформації про стан тролейбусів є критично важливим для ефективного управління парком. Однак нині цей процес є надто ускладненим. Інженери часто змушені звертатися до спеціальних відділів або спілкуватися з відповідальними особами, щоб отримати необхідні дані. Це не лише забирає час, але й перешкоджає прийняттю оперативних рішень.

Відсутність централізованої системи моніторингу значно ускладнює роботу. Дані про технічний стан тролейбусів зазвичай зберігаються у фрагментованому вигляді в різних підрозділах, таких як диспетчерська служба чи бухгалтерія. Це означає, що для отримання повної картини необхідно звертатися до кількох джерел, що потребує додаткових ресурсів.

Навіть за наявності сучасних датчиків їхні показники не завжди інтегровані в централізовану систему. Наприклад, інформація про стан акумуляторів або гальмівної системи може бути доступною лише через ручний огляд. Це робить неможливим моніторинг у реальному часі, що критично важливо для запобігання аварій.

1.3.3 Труднощі у виявленні несправностей

Процес діагностики технічних несправностей залишається значною мірою неавтоматизованим. Інженери змушені вручну оглядати основні вузли тролейбуса, що займає багато часу і залежить від їхнього досвіду. Через це ранні ознаки несправностей часто залишаються непоміченими, а самі поломки виявляються лише на пізніх стадіях.

Ручний аналіз даних також створює додаткові ризики. Наприклад, інженер може неправильно інтерпретувати показники або пропустити важливі деталі, які свідчать про необхідність термінового ремонту. Відсутність автоматизованих інструментів для аналізу значно знижує ефективність діагностики.

1.3.4 Недостатній рівень автоматизації обслуговування

Багато процесів в обслуговуванні тролейбусів все ще виконуються вручну. Наприклад, складання технічних актів, планування ремонтних робіт і аналіз технічного стану здебільшого залежать від людської праці. Це не лише уповільнює роботу, але й створює додаткові ризики через людський фактор.

Відсутність сучасного програмного забезпечення для управління технічним обслуговуванням також є суттєвою проблемою. Тролейбусні парки рідко використовують спеціалізовані програми, які могли б інтегрувати всі дані в єдину систему. Це унеможлиблює ефективне планування та координацію роботи між різними підрозділами.

Окрім цього, доступ до інформації часто обмежується лише внутрішньою локальною мережею, що не дозволяє переглядати дані про технічний стан у реальному часі через онлайн-додатки.

1.4 Вимоги до розроблюваної системи

Створення автоматизованої системи моніторингу тролейбусного парку є важливим кроком для підвищення ефективності управління міським транспортом. У попередніх розділах було детально проаналізовано особливості роботи тролейбусного парку та сучасні виклики, з якими стикаються транспортні підприємства, включно з необхідністю оперативного моніторингу стану тролейбусів, автоматизації технічного контролю та зручного доступу до історичних даних про ремонти й несправності.

У цьому розділі сформульовано основні вимоги до розроблюваної системи, яка має забезпечити інтерактивний контроль технічного стану транспортних засобів у режимі реального часу, підтримку управління технічними актами різних категорій, а також інтеграцію даних із сенсорів тролейбусів.

Вимоги до системи ґрунтуються на потребах різних груп користувачів: адміністраторів, технічних працівників із правами запису й читання, а також користувачів із правами тільки читання. Крім того, окрему увагу приділено нефункціональним аспектам, зокрема безпеці, зручності використання та адаптивності інтерфейсу.

Даний розділ окреслює як загальні вимоги до архітектури системи, так і специфічні функціональні особливості, що забезпечать її ефективну роботу відповідно до сучасних стандартів програмного забезпечення.

1.4.1 Загальні вимоги

Метою розробки є створення інтерактивної системи моніторингу та контролю стану тролейбусів для транспортного підприємства м. Дніпро.

Система має забезпечити доступ до даних у режимі реального часу, автоматизувати створення технічних актів і надавати можливість ручного введення інформації.

Система складається з трьох основних компонентів:

- користувацький інтерфейс (frontend), який забезпечує зручну взаємодію з системою;
- серверна частина (backend), яка обробляє запити, виконує бізнес-логіку та забезпечує обмін даними між користувачами та базою даних;
- база даних, яка зберігає всю необхідну інформацію, включно з даними тролейбусів, технічними актами, історією ремонтів тощо.

1.4.2 Типи користувачів та їхні права доступу

Система підтримує три типи користувачів із різними рівнями доступу.

Адміністратор:

- керує обліковими записами користувачів (створення, редагування, видалення);
- видаляє технічні акти;
- може налаштовувати параметри системи;
- має доступ до всіх функцій системи, включно з функціями інших користувачів.

Користувач із правами запису і читання:

- створює, експортує у PDF, редагує та переглядає технічні акти;
- переглядає дані про стан тролейбусів у режимі реального часу;
- додає інформацію про порушення чи аварії вручну, а також може позначати їх як відремонтовані;
- переглядає історію пошкоджень і ремонтів.

Користувач із правами тільки читання:

- має доступ до інформації про тролейбуси у режимі перегляду;
- переглядає або експортує у PDF технічні акти, але не може їх редагувати чи створювати.

1.4.3 Функціональні вимоги

Система повинна відповідати наступним функціональним вимогам:

- онлайн-моніторинг стану тролейбусів. Система приймає дані з сенсорів у реальному часі дані батарей, зокрема напругу і температуру на комірках акумуляторів. Користувачі мають можливість переглядати ці дані у зручному інтерфейсі, а також керувати технічними актами і пошкодженнями/аваріями тролейбусів. Адміністратор може налаштовувати граничні значення для параметрів, які надходять з сенсорів, а також управляти користувачами системи;

- робота з технічними актами. Система повинна підтримувати п'ять типів технічних актів: поточний акт; акт 1-ої категорії, 1-ої групи; акт 1-ої категорії, 2-ої групи; акт 2-ої категорії; акт 3-ої категорії. Технічні акти повинні автоматично заповнюватися даними з сенсорів (за наявності),

містити можливість ручного введення інформації, мати функцію експорту у PDF-формат для друку чи збереження;

- робота з порушеннями і ремонтами. Користувачі можуть додавати інформацію про несправності чи аварії вручну, позначати несправності як усунені, переглядати історію пошкоджень і ремонтів кожного тролейбуса;

- система повинна працювати з трьома типами тролейбусів: з суперконденсатором, з коротким ходом, з довгим ходом.

- інтерфейс повинен забезпечувати візуалізацію даних за допомогою таблиць, графіків (наприклад, графік напруги комірок акумуляторних батарей), а також інтерактивних індикаторів.

1.4.4 Нефункціональні вимоги

Система повинна відповідати наступним нефункціональним вимогам:

- захист доступу до системи через систему автентифікації;
- запобігання brute force атакам;
- захист API інтерфейсу від несанкціонованого доступу та типових веб-вразливостей;

- зберігання програмного коду у приватному репозиторії;

- інтуїтивно зрозумілий та інтерактивний інтерфейс із підтримкою адаптивності для різних пристроїв;

- підтримка багатомовності з базовою мовою інтерфейсу — українською;

- інтерфейс повинен забезпечувати високу швидкість роботи та продуктивність, оскільки користувачам важливо мінімізувати час очікування;

- передача замовнику готового рішення з повною документацією, доступами та інструкціями.

1.4.5 Вхідні і вихідні дані

Вхідними даними програми є інформація, що надходить від сенсорів тролейбусів, яка включає технічні параметри, такі як стан батареї, дані про несправності та системні повідомлення. Додатково, користувачі мають можливість вводити дані вручну, наприклад, реєструвати порушення,

описувати аварії або надавати уточнення до технічних актів, що дозволяє враховувати аспекти, які неможливо автоматично фіксувати за допомогою сенсорів.

Вихідними даними програми є інтерактивні графіки, які відображають поточний технічний стан тролейбусів і дозволяють здійснювати моніторинг у реальному часі. Крім того, система генерує технічні акти, доступні для перегляду, завантаження або друку, що забезпечує зручність у роботі з документацією. Також програма надає структуровану інформацію про стан тролейбусів та історію їхніх несправностей, що дозволяє отримувати повний огляд про технічний стан транспортних засобів у зрозумілому форматі.

1.5 Вибір інструментарію розробки програмного забезпечення

Під час визначення інструментарію для розробки програмного забезпечення основними критеріями були швидкість розробки, зручність інтеграції сторонніх модулів, а також використання сучасних технологій, що активно застосовуються у професійній розробці.

Згідно з концепцією побудови сучасних вебсервісів, систему було вирішено створити з розподілом на дві основні частини: клієнтську (Frontend) та серверну (Backend). Для розробки серверної частини була обрана мова програмування C# із використанням фреймворку ASP.NET Core, а для роботи з даними – система керування базами даних MySQL. Усі переваги, недоліки та особливості обраного інструментарію буде детально розглянуто в наступних підрозділах.

1.5.1 Мова програмування C# (C Sharp)

C# – це мова програмування з об'єктно-орієнтованим підходом і суворою системою типізації, розроблена компанією Microsoft для платформи

.NET. Її синтаксис схожий на мови C++ та Java, що спрощує адаптацію для досвідчених розробників.

Мова підтримує кілька парадигм програмування, включаючи об'єктно-орієнтовану та функціональну, що робить її гнучким інструментом для вирішення різноманітних завдань.

Серед ключових переваг C#:

- підтримка багатой екосистеми бібліотек та NuGet-пакетів для швидкої розробки;

- висока продуктивність і безпечна робота з пам'яттю;

- інтеграція з платформою .NET Core, що дозволяє створювати потужні, кросплатформенні додатки.

Застосування C# в комбінації з ASP.NET Core забезпечує розробку продуктивних та масштабованих серверних компонентів.

1.5.2 Фреймворк ASP.NET Core

ASP.NET Core – це сучасний, відкритий фреймворк для створення вебдодатків та API, який підтримує високу продуктивність і легкість масштабування. Завдяки активній підтримці Microsoft і спільноти розробників, ASP.NET Core регулярно отримує оновлення, що включають нові функції та оптимізації.

Основні переваги ASP.NET Core:

- можливість розробки вебсервісів із високою швидкодією;

- підтримка кросплатформенності: додатки можна запускати на Windows, macOS та Linux;

- інтеграція з іншими інструментами .NET Core, включаючи Entity Framework Core, для зручної роботи з базами даних.

Вибір ASP.NET Core був зумовлений його стабільністю, високою продуктивністю та широкими можливостями для розробки сучасних серверних додатків.

1.5.3 MySQL

Для роботи з даними було обрано MySQL – потужну реляційну систему керування базами даних із відкритим вихідним кодом. MySQL є одним із найпоширеніших рішень для зберігання та обробки даних завдяки своїй надійності, високій продуктивності та гнучкості.

Ключові особливості MySQL:

- підтримка реляційної моделі даних із чітко структурованими таблицями;
- можливість обробки великого обсягу даних та одночасного доступу багатьох користувачів;
- висока продуктивність завдяки оптимізованій роботі з SQL-запитами;
- доступність як у безкоштовній, так і в комерційній версіях, залежно від потреб проєкту.

1.5.4 Entity Framework Core

Для взаємодії з базою даних було обрано Entity Framework Core – ORM-фреймворк, що дозволяє працювати з базами даних як із колекціями об'єктів. Це рішення значно спрощує доступ до даних та автоматизує виконання SQL-запитів.

Entity Framework Core забезпечує:

- автоматичне створення запитів до бази даних;
- підтримку різних типів баз даних, включаючи MySQL, завдяки адаптерам;
- легкість у роботі з об'єктно-реляційним перетворенням даних, що мінімізує необхідність написання SQL-запитів вручну.

Завдяки використанню Entity Framework Core та MySQL у цьому проєкті вдалося забезпечити ефективне управління даними з мінімальними витратами часу на налаштування взаємодії між сервером та базою даних.

1.5.5 Інтегроване середовище розробки Rider

Для розробки було обране інтегроване середовище JetBrains Rider, яке є одним із найбільш зручних інструментів для роботи з мовою програмування C#.

Основні переваги Rider:

- висока швидкодія та стабільність роботи навіть у великих проєктах;
- кросплатформеність: однаковий інтерфейс і функціонал на Windows, macOS і Linux;
- інтеграція з ReSharper, що забезпечує розширені можливості для рефакторингу, перевірки коду та автоматизації завдань розробника.

Rider став оптимальним вибором завдяки своїй продуктивності та широкому набору інструментів для .NET-розробки.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Загальна структура розроблюваної системи

Створена система моніторингу складається з трьох основних частин, які комунікують між собою за допомогою HTTPS протоколу: веб-інтерфейсу, веб-серверу та множини тролейбусів тролейбусного парку з встановленими контролерами і датчиками.

Узагальнена схема системи відображена на рис. 2.1

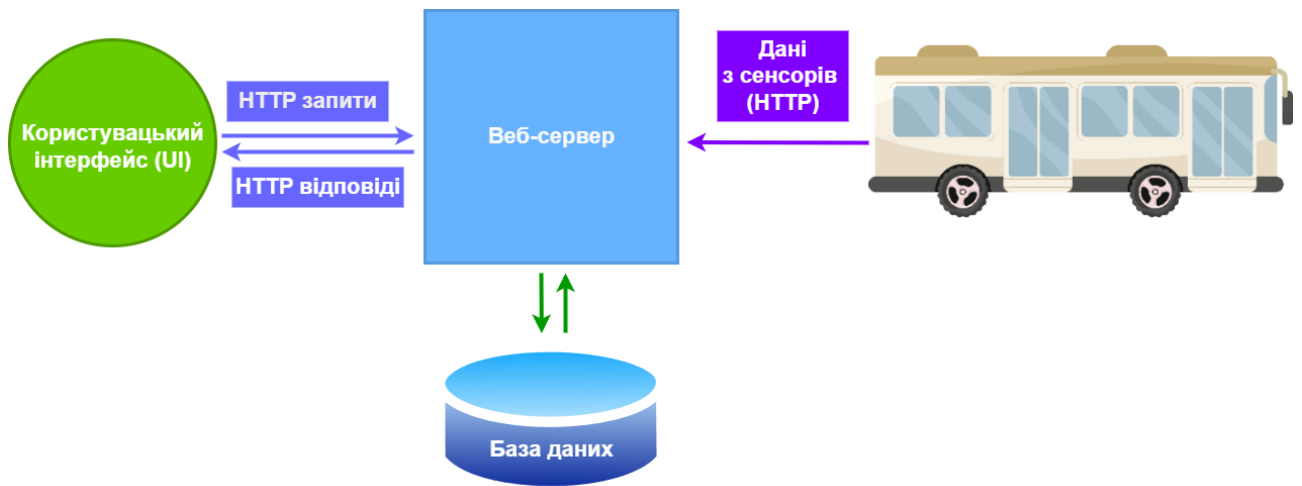


Рисунок 2.1 – Схема моніторингової системи

Веб-інтерфейс представлений у вигляді звичайного сайту в мережі Інтернет, який у сутності є терміналом взаємодії з системою моніторингу. Основною перевагою такого веб-інтерфейсу є доступність: оператори тролейбусів можуть отримувати до нього доступ з будь-якого пристрою, що має доступ до інтернету (телефон, планшет, стаціонарний комп'ютер, тощо). Розмітка користувацького інтерфейсу була побудована з використанням динамічних елементів, що дало змогу зробити веб-інтерфейс адаптивним, тобто таким, що відображається однаково гарно на дисплеях будь-якого розміру.

Веб-сервер - ключова частина системи моніторингу. Саме тут міститься вся бізнес-логіка системи. Веб-сервер працює як REST API сервіс, тобто може приймати HTTP запити і відповідати на них. Веб-сервер отримує дані з датчиків, а також інформацію, введену користувачем за допомогою веб-інтерфейсу. Вся інформація, що надходить, обробляється і структурується у зручний для сервера формат і потім може бути отримана користувачем системи через веб-інтерфейс. Веб-сервер зберігає отриману інформацію в базу даних.

Для обробки і передачі даних з боку тролейбусу, як вже було зазначено у минулому розділі, використовується, встановлений на кожному з тролейбусів парку, сучасний контролер Teltonika TRB145. Варто зазначити,

що створена моніторингова система абсолютно не залежить від модему, встановленому у тролейбусі. Дані отримуються сервером за допомогою HTTP запиту, який відправляє контролер. Для сервера немає різниці за допомогою якого модему відправлялися дані, тобто якщо технічні спеціалісти тролейбусного парку вирішили замінити модеми на тролейбусах – кінцевий користувач системи абсолютно ніяк про це не дізнається і не доведеться вносити жодні зміни до сирцевого коду системи. Операторам треба просто буде налаштувати нові контролери на відправку даних на сервер

Для розробки автоматизованої системи моніторингу були використані найсучасніші технології веброзробки. Користувацький інтерфейс створений за допомогою фреймворку React і мови програмування Typescript. Серверна частина написана з використанням фреймворку ASP.NET Core та мови програмування C#. Для роботи з базою даних була використана СКБД MySQL.

2.2 Структура серверної частини

Серверна частина є ключовим елементом системи моніторингу тролейбусного парку. Вона об'єднує всі компоненти системи, забезпечує обробку бізнес-логіки та підтримує збереження і доступ до даних. Кожна функціональна частина системи реалізована через підсистеми, які тісно взаємодіють між собою, забезпечуючи комплексне виконання завдань.

Структура серверної частини відображена на рис 2.2.

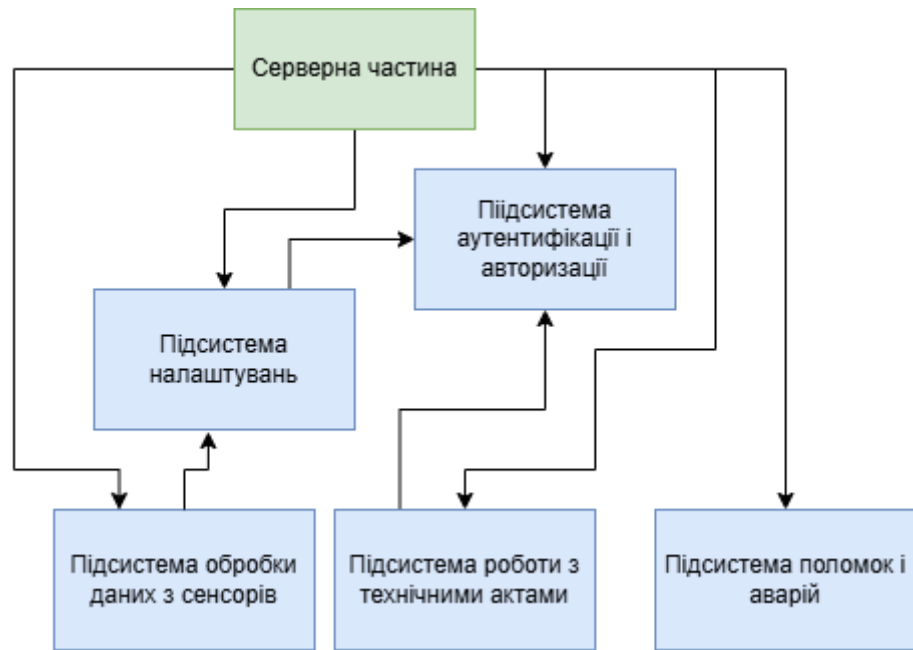


Рисунок 2.2 – Структура серверної частини

Основою роботи всієї системи є підсистема аутентифікації та авторизації, реалізована на основі технології JWT (JSON Web Token). Ця підсистема відповідає за створення та валідацію токенів, які надають доступ до системи лише авторизованим користувачам. Ролі користувачів визначаються під час авторизації, що забезпечує диференційований доступ до різних функцій системи. Усі інші підсистеми використовують функціонал аутентифікації для обмеження доступу до даних і операцій, таких як створення технічних актів, реєстрація аварій або зміна системних налаштувань.

Підсистема налаштувань відповідає за конфігурацію системи та управління параметрами. Її функціонал доступний виключно адміністраторам через підсистему аутентифікації. У ній реалізовано облік користувачів (додавання, редагування та видалення облікових записів), а також налаштування критичних параметрів, таких як допустимі межі сенсорних даних, зокрема напруги акумуляторів або температури. Ці параметри використовуються іншими підсистемами, наприклад, для обробки даних з сенсорів. Крім того, налаштування дозволяють адаптувати систему до особливостей обладнання або змін у вимогах експлуатації.

Підсистема роботи з технічними актами дозволяє користувачам вручну створювати та редагувати акти, що документують технічний стан транспортних засобів. Кожен акт містить інформацію про тип операції (ремонт, обслуговування або перевірка), деталі виконаних робіт, а також дані про час і відповідальну особу. Доступ до цієї функції мають лише авторизовані користувачі, що контролюється через підсистему аутентифікації. Інформація зберігається у базі даних у структурованому вигляді для подальшого аналізу або створення звітів. Експорт актів у PDF реалізовано на рівні фронтенду, що забезпечує зручність для користувачів та знижує навантаження на сервер.

Підсистема обробки поломок і аварій реалізує механізм реєстрації інцидентів. Вона дозволяє авторизованим користувачам додавати інформацію про аварії, вказуючи деталі, такі як час, місце, тип інциденту та стан обладнання. Ці дані зберігаються у базі даних для аналізу, створення звітів або організації ремонтних робіт. Автоматичні функції у цій підсистемі відсутні, що знижує ризик помилок і забезпечує високу точність інформації.

Підсистема отримання і обробки даних з тролейбусів відповідає за взаємодію із сенсорним обладнанням транспортних засобів. Вона отримує дані через HTTP-запити від модемів Teltonika TRB145, після чого здійснює первинну обробку і перевірку даних. Ці дані структуруються, перевіряються на відповідність критичним параметрам, що налаштовуються через підсистему налаштувань, і зберігаються у базі для подальшого використання. Гнучкість підсистеми дозволяє інтегрувати різні типи обладнання без змін у серверному коді.

Користувацький інтерфейс тісно пов'язаний із серверною частиною. У ньому реалізовано відповідні екрани для взаємодії з кожною підсистемою. Наприклад, користувачі можуть створювати технічні акти, переглядати дані про аварії або редагувати системні налаштування. Однією з ключових функцій є інтерактивний dashboard, який відображає дані в реальному часі,

такі як технічний стан тролейбусів, показники сенсорів, інформацію про аварії та поломки.

Крім того, у користувацькому інтерфейсі реалізовано функціонал побудови графіків і діагностичних таблиць, які дозволяють аналізувати дані, зібрані з тролейбусів, та виявляти аномалії або тренди. Функція експорту даних у PDF для технічних актів також реалізована на фронтенді, що дає змогу користувачам швидко генерувати документацію для звітності чи архівування.

Завдяки інтеграції всіх підсистем серверна частина забезпечує стабільність і високу ефективність роботи системи, а також створює зручний інструмент для моніторингу та управління тролейбусним парком.

2.3 Вибір підходу до розробки системи

Для розробки автоматизованої системи тролейбусного парку було обрано підхід Domain-Driven Design (DDD). Цей підхід дозволяє ефективно організувати процес розробки складних програмних систем, орієнтуючись на потреби бізнесу та доменну область. Основна ідея DDD полягає у тому, щоб створити програмну систему, яка точно відображає процеси реального світу, що робить її зрозумілою для всіх учасників проєкту – розробників, аналітиків та замовників.

Вибір DDD обумовлений специфікою проєкту, де доменна область – управління міським транспортом – має багаторівневу структуру з багатьма взаємопов'язаними аспектами. До таких аспектів належать реєстрація технічних актів, обробка аварій, моніторинг стану транспортних засобів у реальному часі, управління користувачами та забезпечення збереження історичних даних.

Основні принципи реалізації DDD у системі:

- фокус на доменній моделі;
- використання обмежених контекстів (Bounded Contexts);
- чіткий поділ шарів;
- залучення експертів доменної області;
- реалізація доменних сервісів.

Фокус на доменній моделі позначає, що система побудована навколо доменних об'єктів, які відповідають ключовим елементам предметної області, таким як "тролейбус", "технічний акт", "аварія". Кожен із цих об'єктів має свої атрибути та поведінку, що забезпечує точне моделювання бізнес-логіки.

Використання обмежених контекстів, кожен з яких відповідає за конкретну підсистему (наприклад, моніторинг тролейбусів у реальному часі, обробка технічних актів та історії несправностей, реєстрація аварій, управління користувачами), дозволило зменшити складність системи, уникнути конфліктів між різними аспектами бізнес-логіки та забезпечити незалежність підсистем.

DDD передбачає багат шарову архітектуру, тому структура автоматизованої системи моніторингу тролейбусного парку поділена на шари. Шар домену: включає доменні моделі, сутності, об'єкти значення та доменні сервіси, що реалізують бізнес-логіку. Шар додатку: забезпечує координацію дій між доменним шаром та інфраструктурними компонентами. Шар інфраструктури: реалізує технічні деталі, такі як доступ до бази даних, обробка запитів API та інтеграція з іншими системами.

Під час розробки активно залучалися експерти, які детально описували вимоги та особливості управління транспортом. Це забезпечило відповідність системи реальним потребам.

Для функціоналу, який не можна віднести до конкретної сутності, були створені доменні сервіси. Наприклад, обробка складних зв'язків між технічними актами та станом транспортних засобів.

Застосування підходу DDD дозволило створити систему, яка:

- чітко відображає процеси управління транспортом;

- має модульну структуру, що спрощує підтримку та розширення функціоналу;
- залишає бізнес-логіку незалежною від технічних деталей, що підвищує гнучкість системи;
- легко сприймається як розробниками, так і представниками бізнесу завдяки зрозумілій моделі домену.

Таким чином, обраний підхід дозволив забезпечити ефективність розробки та створення масштабованої системи, яка відповідає вимогам сучасної транспортної інфраструктури.

2.4 Вибір патерну проєктування

Для серверної частини автоматизованої системи моніторингу тролейбусного парку було обрано патерн проєктування "Медіатор". Цей патерн є одним з поведінкових патернів, що дозволяє зменшити зв'язність між різними компонентами системи, впроваджуючи посередника для комунікації. Він допомагає уникнути безпосередніх взаємодій між об'єктами, замінюючи їх на взаємодію через медіатора, що спрощує подальше розширення і підтримку системи.

У контексті ASP.NET Core патерн "Медіатор" може бути ефективно використаний для організації взаємодії між різними частинами серверної частини системи. Запити обробляються через відповідні обробники, кожен з яких відповідає за певний тип запиту. Патерн дозволяє створити центрального медіатора, який виступає посередником між обробниками запитів. Коли запит надходить, медіатор визначає, який обробник повинен обробити цей запит, передає йому необхідні дані та отримує результат виконання. Це знижує рівень залежностей між обробниками та спрощує архітектуру системи.

Для реалізації цього патерну в ASP.NET Core використовується бібліотека MediatR, яка надає підтримку медіаторного підходу. Вона дозволяє створювати та реєструвати обробники запитів у додатку, завдяки чому запити надходять до медіатора, який обирає відповідного обробника, передає йому дані та повертає результат користувачу.

Застосування патерну "Медіатор" в серверній частині системи моніторингу тролейбусного парку дозволяє зробити код більш структурованим, знижує залежності між компонентами системи і спрощує її подальше розширення та підтримку. Це забезпечує гнучкість і дає змогу змінювати логіку обробки запитів без необхідності змінювати клієнтський код.

3 РЕАЛІЗАЦІЯ СИСТЕМИ

У цьому розділі розглянуто ключові аспекти реалізації бізнес-логіки програми та представлено структуру її класів.

3.1 Реалізація основних алгоритмів системи

Як було зазначено у попередніх розділах, система моніторингу тролейбусного парку складається з двох основних компонентів: клієнтської частини (користувацького інтерфейсу, frontend) та серверної частини (вебсервера, backend), які взаємодіють між собою за допомогою HTTP-запитів. Далі детально розглянуто процес обробки запитів у системі.

Запит формується на стороні клієнта за допомогою відповідних функцій, реалізованих у фронтенді. Наприклад, коли користувач натискає

кнопку "Додати", клієнтська частина створює HTTP POST-запит із необхідними даними про інцидент і надсилає його на сервер для обробки (рис. 3.1).

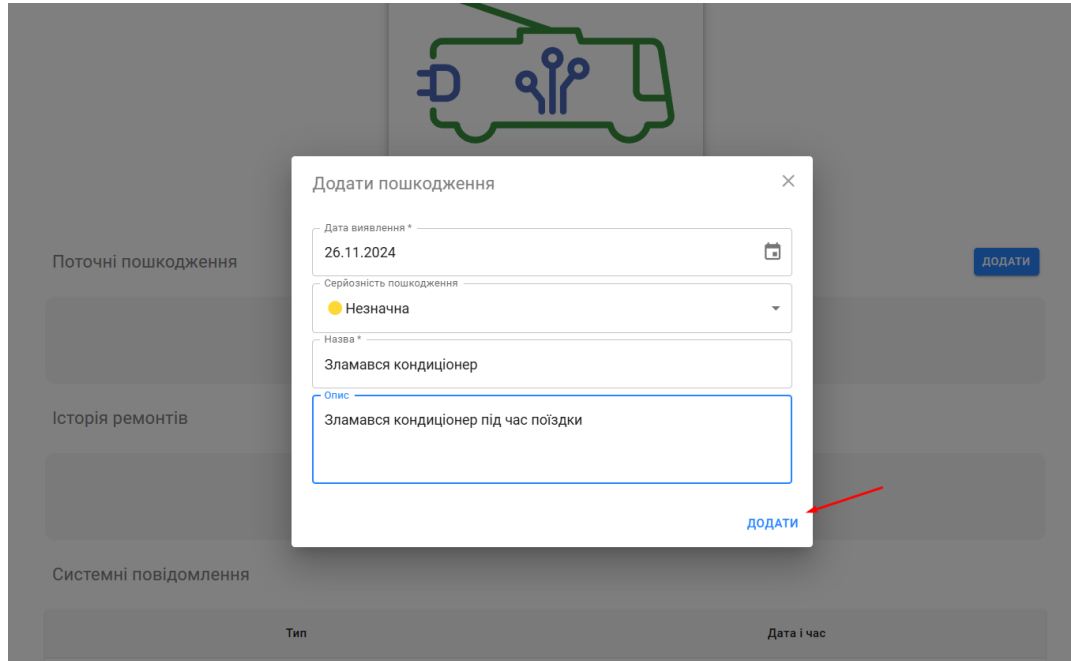


Рисунок 3.1 – Користувач натискає на кнопку “Додати”

Після надсилання запит проходить через middleware серверної частини. У цьому каналі обробки виконується логування, автентифікація та перевірка валідності запиту. Якщо перевірки успішні, запит перенаправляється у відповідний контролер, до якого прив’язана функція-обробник (рис. 3.2).

Розглянемо, наприклад, запит на реєстрацію поломки. Він потрапляє у контролер `FaultsController`, де функція-обробник звертається до бази даних через `ORM Entity Framework`. У результаті в базі створюється новий запис про інцидент, який містить інформацію про тип поломки, час її реєстрації та місце події. Сервер формує відповідь, яка повертається клієнту. Після отримання відповіді користувацький інтерфейс оновлюється, наприклад, додаючи нову поломку до інтерактивної таблиці.

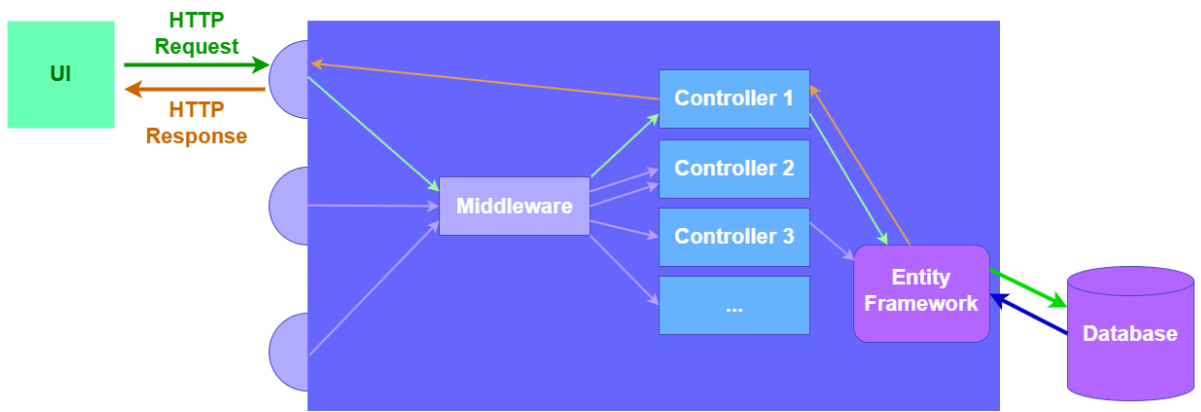


Рисунок 3.2 – Детальна схема роботи серверної частини

Запити та відповіді в системі стандартизовані за допомогою OpenAPI специфікації. Серверна частина автоматично генерує JSON-схему API за допомогою Swagger. Ця схема включає всі доступні ендпоінти, їхні параметри та типи відповідей. На основі цієї схеми клієнтська частина за допомогою інструменту NSwag автоматично генерує JavaScript-клієнти. Ці клієнти забезпечують коректну взаємодію з сервером, гарантуючи відповідність запитів та відповідей між компонентами системи. Такий підхід дозволяє уникнути несумісностей та спрощує розробку, оскільки будь-які зміни в API автоматично відображаються на фронтенді.

Таким чином, інтеграція всіх компонентів системи через стандартизовані алгоритми забезпечує стабільну та зручну роботу системи моніторингу, зокрема функції реєстрації подій, обробки даних із сенсорів та візуалізації інформації для кінцевих користувачів.

3.1.1 Аутентифікація та реєстрація

Аутентифікація та реєстрація в навчальній платформі реалізовані на основі сучасних практик безпеки, забезпечуючи надійність обміну даними між клієнтом і сервером. У цій платформі для ідентифікації та авторизації користувачів використовується стандарт JSON Web Token (JWT), який дозволяє безпечно передавати інформацію у форматі токенів. Особливість реалізації системи полягає в тому, що реєстрація нових користувачів

здійснюється виключно адміністратором через панель управління, що знижує ризики несанкціонованого доступу.

JWT-токен складається з трьох основних частин: заголовка, який вказує тип токена (наприклад, JWT) і алгоритм підпису (наприклад, HMAC SHA256); набору даних (payload), що містить інформацію про користувача, як-от його ідентифікатор, роль чи інші атрибути; і підпису, який генерується на основі секретного ключа для забезпечення цілісності даних. Кожна частина токена закодована у форматі Base64URL, що дозволяє передавати його у запитах.

У процесі реєстрації, адміністратор через спеціальний інтерфейс вводить дані нового користувача (логін, ім'я, email та пароль) та визначає його права (“запис” чи “запис і читання”). Ці дані надсилаються серверу через HTTP POST-запит. Сервер обробляє запит у кілька етапів:

- перевірка формату даних;
- перевірка наявності користувача;
- створення нового запису;
- повідомлення адміністратора.

Під час надходження запиту система перевіряє, чи відповідають дані заданим критеріям (наприклад, чи коректний формат email). Далі система виконує пошук у базі даних, щоб переконатися, що користувача з таким логіном ще не існує. Якщо перевірка успішна, у базі даних створюється новий запис із переданими даними. У відповідь на запит сервер надсилає адміністратору підтвердження успішного створення користувача або повідомляє про помилку у випадку дублювання email.

Процес реєстрації зображено на діаграмі прецедентів (рисунок 3.3), яка показує основні кроки взаємодії адміністратора з платформою.

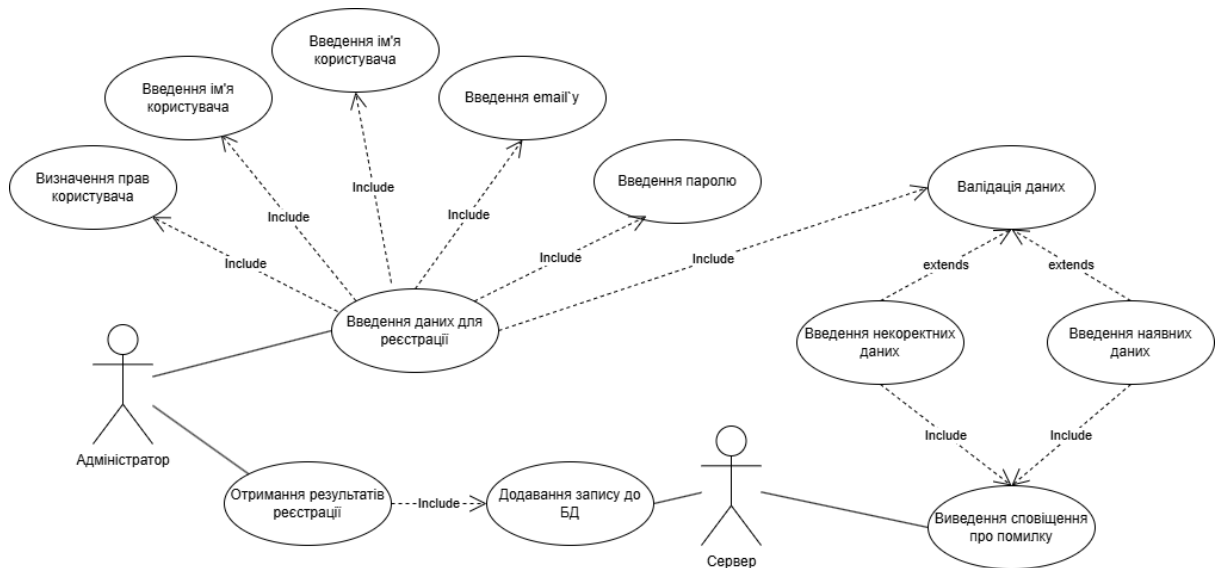


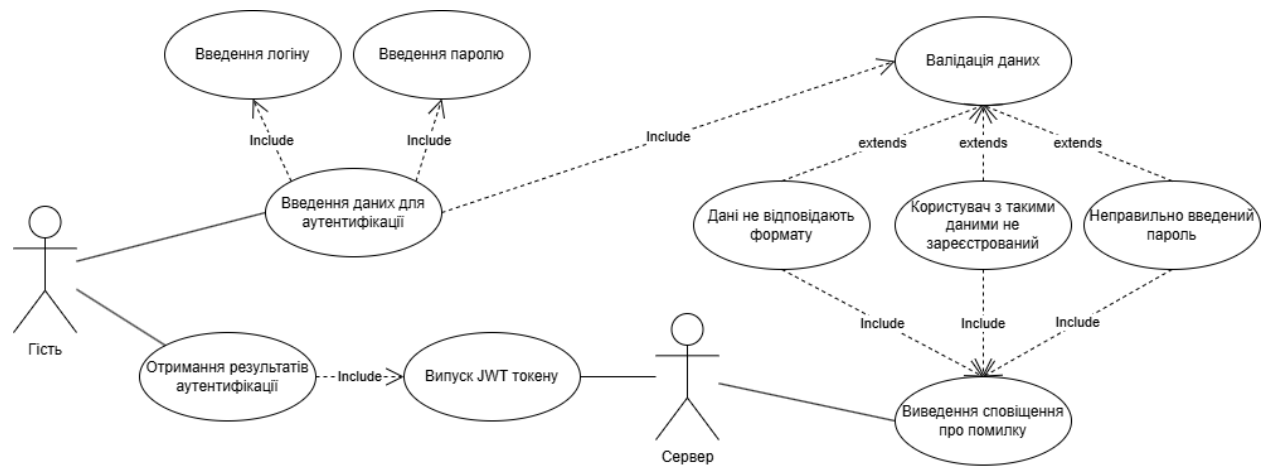
Рисунок 3.3 – Діаграма прецедентів для реєстрації користувача у системі

Аутентифікація користувачів організована за принципом перевірки облікових даних і подальшого використання токенів для доступу до ресурсів. Користувач вводить свої дані (email і пароль), які надсилаються серверу через HTTP POST-запит. Сервер виконує такі дії:

- перевірка даних;
- пошук у базі даних;
- генерація токена;
- відповідь клієнту.

Перед обробкою запиту сервер перевіряє, чи коректно введені дані (наприклад, довжина пароля та його формат).

На діаграмі прецедентів для аутентифікації (рис. 3.4) відображено всі етапи цього процесу, включно з взаємодією між користувачем, клієнтом та сервером. Система звіряє логін і пароль із записами у базі. Якщо облікові дані не збігаються, сервер повертає повідомлення про помилку. У разі успішного збігу даних сервер генерує JWT-токен, який містить інформацію про користувача, його роль та час дії токена. Сервер надсилає токен клієнту, який зберігає його для подальшого використання в заголовках HTTP-запитів.



Рисунк 3.4 – Діаграма прецедентів для алгоритму аутентифікації

JWT-токени є ключовим елементом безпеки системи. Вони використовуються для підтвердження автентичності кожного запиту. Якщо запит надходить без токена або токен є недійсним (наприклад, через закінчення терміну дії), сервер автоматично відхиляє його. Це дозволяє захистити конфіденційні дані користувачів і запобігти несанкціонованому доступу до системи.

Унікальна особливість цієї платформи полягає в тому, що лише авторизовані користувачі можуть отримати доступ до функціоналу системи, а реєстрація нових користувачів є централізованою і здійснюється адміністратором. Це мінімізує ризики, пов'язані зі спробами створення фальшивих акаунтів, і гарантує, що всі користувачі є перевіреними.

Таким чином, інтеграція алгоритмів аутентифікації та реєстрації через JWT забезпечує високу надійність платформи, її безпеку та зручність для користувачів і адміністраторів.

3.1.2 Прийом і обробка даних з датчиків

Працівники троллейбусних парків повинні постійно відстежувати технічний стан троллейбусів. Відстеження таких параметрів акумулятора, як напруга на комірках, перепад напруги на блоці, перепад напруги між всіма блоками, температура блоків та загальний перепад температури - має велике значення з точки зору безпеки, ефективності та тривалості служби

аккумуляторної системи і електродвигуна. Моніторинг напруги комірок і перепаду напруги допомагає забезпечити рівномірний розподіл заряду між комірками, що збільшує ефективність роботи батареї. Високий перепад напруги може вказувати на несправності в акумуляторі, які можуть зменшити продуктивність. Регулярне відстеження температур блоків і перепаду температури допомагає уникати перегріву, який зазвичай призводить до зниження тривалості служби акумуляторів. Збереження оптимальних умов сприяє тривалому та ефективному використанню акумуляторних батарей. Тому моніторинг саме цих параметрів є ключовим елементом для забезпечення безпеки, тривалості служби та ефективності акумуляторних батарей у тролейбусних системах.

Без існування автоматизованої системи моніторингу в користуванні тролейбусного парку всі ці параметри необхідно відстежувати «ручним» способом. Для того, щоб отримати дані з датчиків всіх тролейбусах парку необхідно обстежити кожний з них (приклад панелі з даними з датчиків акумулятора реального тролейбусу приведено на рис. 3.5). Це забирає велику кількість часу. Також суттєвим недоліком такого підходу є неможливість отримати дані, коли ти знаходишся не поряд з тролейбусом. І звісно варто не забувати про втрату даних, у випадку коли технічний експерт по деякій причині не зберіг їх (не записав у технічний акт наприклад).

Для побудови моніторингової системи тролейбусного парку необхідно збирати дані технічних параметрів тролейбусів у режимі реального часу. Для збору таких даних використовуються датчики. У рамках цієї роботи була необхідність збирати дані, щодо температури блоків акумуляторних батарей і напруги на комірках їх комірок. Ці параметри є ключовими.

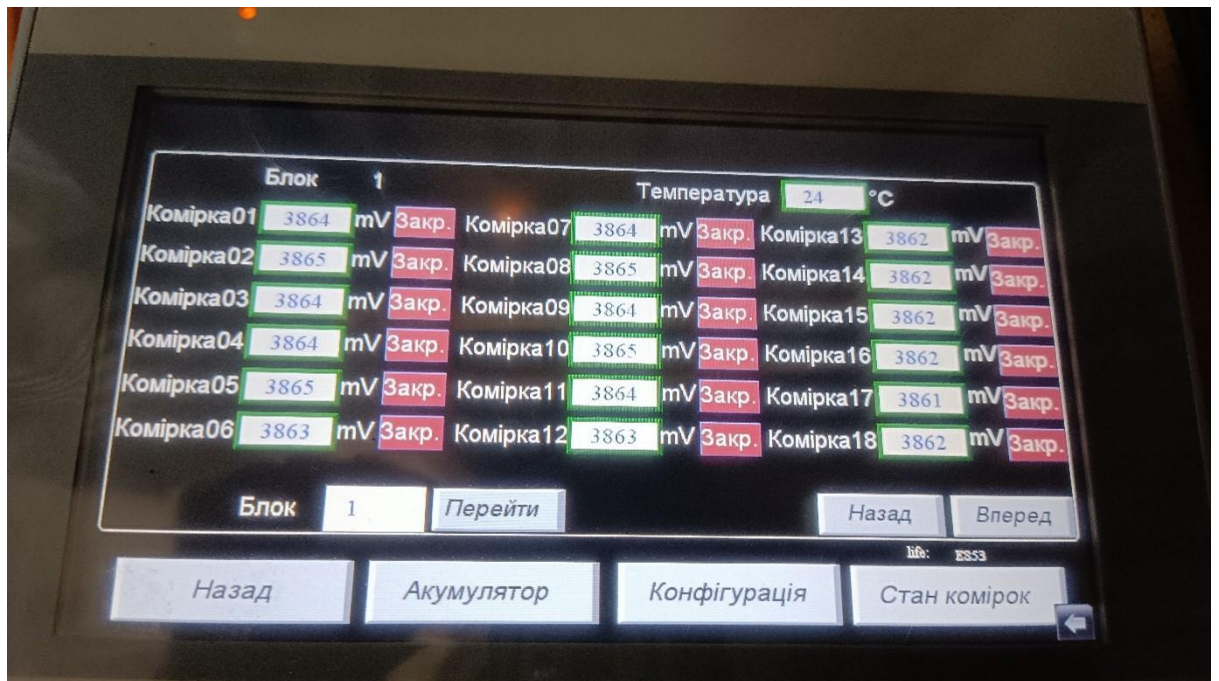


Рисунок 3.5 – Панель датчику акумулятора тролейбусу

Датчики температури комірок акумулятора є критично важливими для забезпечення безпеки та ефективності роботи акумуляторних батарей. Вони дозволяють відслідковувати температуру кожної комірки, що допомагає запобігати перегріву та можливим аварійним ситуаціям. Датчики напруги комірок акумулятора, в свою чергу, забезпечують моніторинг напруги на кожній комірці, що є необхідним для контролю стану заряду та виявлення будь-яких аномалій у роботі акумулятора.

В якості датчиків температури блоків акумулятора були обрані датчики Dallas DS18B20 [10]. Ця модель датчиків належить до типу цифрових термометрів і може вимірювати температуру в діапазоні від -55°C до $+125^{\circ}\text{C}$ з точністю $\pm 0.5^{\circ}\text{C}$. Перевагами датчиків Dallas DS18B20 є маленький розмір, низьке енергоспоживання, проста інтеграція та інтерфейс підключення 1-Wire, що дозволяє підключати кілька датчиків на одну лінію.

Для моніторингу напруги комірок акумуляторів тролейбусів були використані аналогові датчики напруги LEM LV 25-P [11] з діапазоном вимірювання від 0 до 1000В. Основною перевагою цих датчиків є висока точність, можлива завдяки використанню технології гальванічної ізоляції.

Датчики мають аналогові виходи, що дозволяють підключення до контролера або модему через аналоговий входи. Також варто зазначити, що датчики LEM LV 25-P є дуже надійними та здатні працювати у важких умовах експлуатації.

Для передачі отриманих з датчиків даних був обраний і встановлений на кожному тролейбусі тролейбусного парку модем Teltonika TRB145 [12]. Цей модем є компактним і потужним пристроєм для обробки і передачі даних.

Модель TRB145 використовує мобільну 4G LTE мережу для зв'язку з зовнішнім світом, яка забезпечує високу швидкість передачі даних. Модем також оснащений вбудованим GPS, що дозволяє відслідковувати місцезнаходження тролейбуса в режимі реального часу. TRB145 має широкий діапазон робочих температур (-40°C до 75°C), що дозволяє використовувати його в різних кліматичних умовах.

Модем Teltonika TRB145 дуже компактний, а тому зручний для встановлення. Варто також зазначити, що контролер має Ethernet-порт, який надає можливість технічним експертам підключитися до локальної мережі модему для налаштування та діагностики несправностей. Високий рівень безпеки модему можливий завдяки підтримці VPN, IPsec, та Firewall.

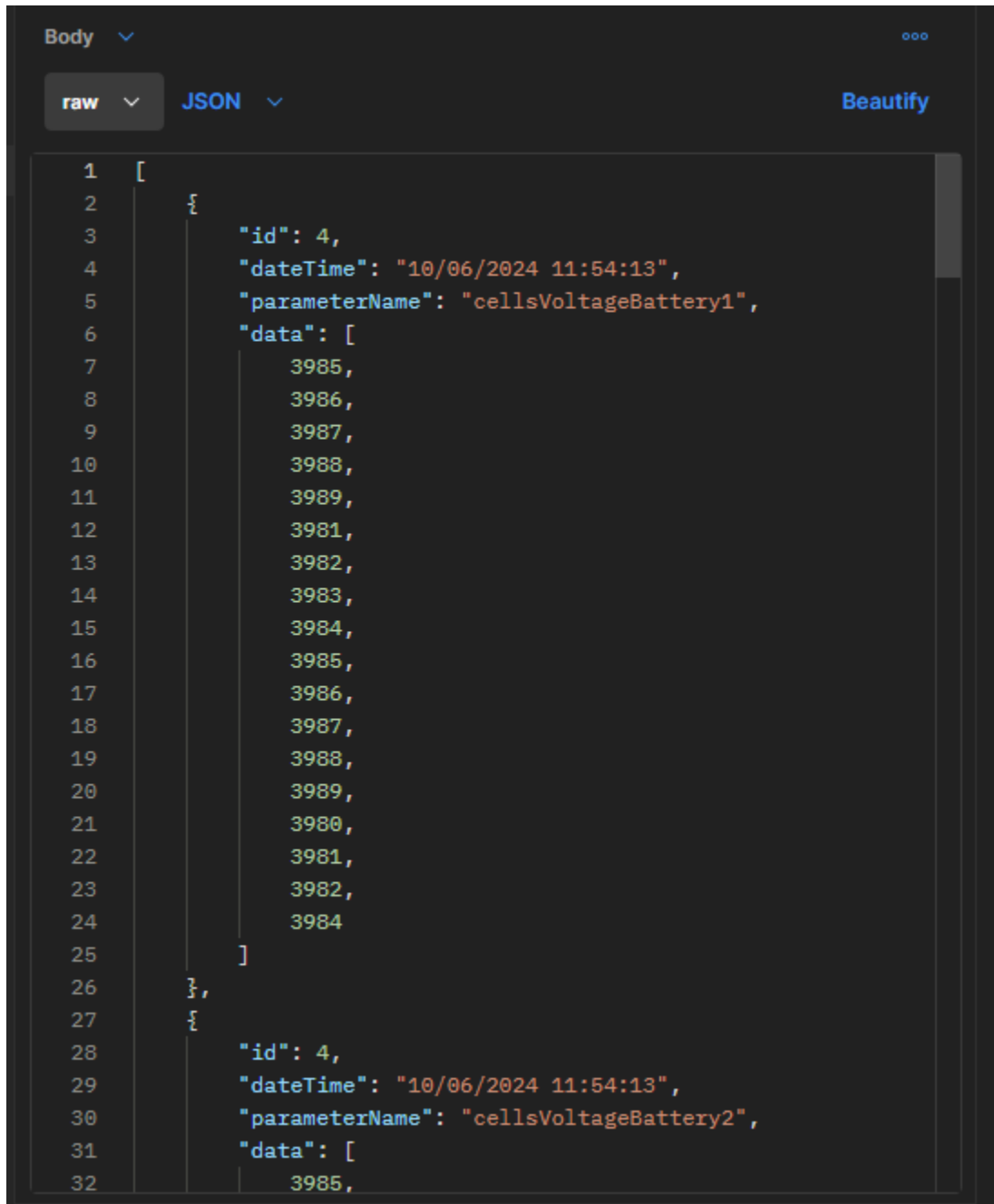
Модем підключається до джерела живлення тролейбуса (зазвичай 12V) та до датчиків температури і напруги акумуляторних комірок. Для підключення використовується аналоговий вхід модему (можна підключити також використовуючи стандартний інтерфейс RS485).

Конфігурація модему відбувається за допомогою програмного забезпечення Teltonika RMS (Remote Management System) [13] або веб-інтерфейсу. Для коректної роботи необхідно налаштувати параметри підключення до мобільної мережі (APN, користувач, пароль) та параметри передачі даних (часовий інтервал, формат даних, IP-адреса сервера).

Для передачі даних на сервер використовується формат JSON. Формат даних має наступну структуру:

- id — унікальний ідентифікатор тролейбуса;
- dateTime — час отримання даних;
- parameterName — назва параметра (наприклад, напруга чи температура);
- data — масив значень.

Приклад реальних даних відображено на рис. 3.6.



```
Body
raw JSON Beautify
1  [
2    {
3      "id": 4,
4      "dateTime": "10/06/2024 11:54:13",
5      "parameterName": "cellsVoltageBattery1",
6      "data": [
7        3985,
8        3986,
9        3987,
10       3988,
11       3989,
12       3981,
13       3982,
14       3983,
15       3984,
16       3985,
17       3986,
18       3987,
19       3988,
20       3989,
21       3980,
22       3981,
23       3982,
24       3984
25     ]
26   },
27   {
28     "id": 4,
29     "dateTime": "10/06/2024 11:54:13",
30     "parameterName": "cellsVoltageBattery2",
31     "data": [
32       3985,
```

Рисунок 3.6 – Приклад даних, що надходять з датчиків тролейбусів

Дані передаються на захищений серверний ендпоінт через HTTPS. Запити ідентифікуються за допомогою токена, що забезпечує додатковий рівень безпеки. Сервер обробляє дані та зберігає їх у відповідних таблицях бази даних для подальшого використання.

Схема передачі даних з датчиків до серверу:

- модем збирає дані з датчиків;
- модем передає дані через HTTP на захищений ендпоінт сервера;
- сервер обробляє та зберігає дані, забезпечуючи доступ до них через інтерфейс моніторингової системи.

Узагальнена схема отримання та передачі даних з датчиків відображена на рис. 3.7.

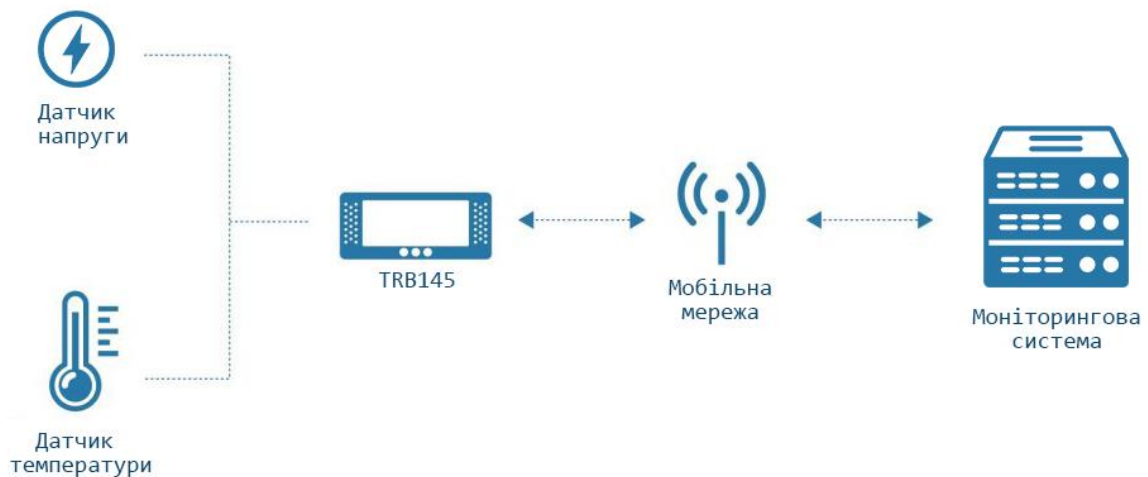


Рисунок 3.7 – Схема передачі даних з датчиків тролейбуса до моніторингової системи

3.1.3 Оновлення стану тролейбуса в режимі реального часу

Оновлення стану тролейбуса в реальному часі є ключовим компонентом моніторингової системи, що дозволяє оперативно реагувати на будь-які зміни в технічному стані тролейбусів. Алгоритм складається з наступних кроків:

- відкриття сторінки з дашбордом;

- відправка запиту на сервер;
- обробка запиту сервером;
- аналіз отриманих даних.;
- візуалізація стану тролейбусу на дашборді.

Користувач входить у систему та відкриває сторінку з інтерактивним дашбордом, на якому відображаються актуальні дані про всі тролейбуси в парку.

Після завантаження сторінки клієнтський додаток автоматично відправляє запит до серверу для отримання останніх даних щодо обраного тролейбусу. Запит містить ідентифікатор тролейбусу та токен автентифікації.

Сервер приймає запит, перевіряє валідність токена, знаходить відповідний тролейбус у базі даних та підвантажує останні отримані технічні дані (наприклад, параметри напруги та температури).

Отримані дані про напругу комірок порівнюються з граничними значеннями, які зберігаються в налаштуваннях системи. У разі перевищення межі або якщо значення напруги занадто низьке, в дашборді поруч із відповідним параметром з'являється червоний індикатор. Аналізується дата останнього успішного отримання даних із тролейбуса. Якщо дані не оновлювались понад тиждень, система вважає, що зв'язок із тролейбусом втрачено, і відображає індикатор зв'язку. Дані про зареєстровані аварії тролейбуса перевіряються на наявність критичних несправностей, які залишаються невідремонтованими. Якщо є невідремонтовані аварії то іконка тролейбусу забарвлюється у відповідний колір.

Якщо всі параметри в нормі, іконка тролейбусу на дашборді має зелений колір. Якщо присутні некритичні несправності, напруга виходить за допустимі межі, або втрачено зв'язок із тролейбусом, іконка стає жовтого кольору. Якщо тролейбус має критичні невідремонтовані аварії, іконка змінює колір на червоний, сигналізуючи про необхідність термінового втручання.

Цей алгоритм гарантує, що стан кожного тролейбуса оперативно оновлюється та відображається в зручній формі для користувачів, забезпечуючи високу швидкість реагування на потенційні проблеми.

3.1.4 Робота з технічними актами

Технічні акти в системі є важливим елементом для відображення стану тролейбусів та обліку проведених робіт. Кожен технічний акт фіксує поточний стан тролейбуса та визначає необхідність подальших дій. Усі технічні акти є незмінними після їх створення, що забезпечує достовірність і правильність обліку.

Існує п'ять основних типів технічних актів:

- поточний технічний акт;
- технічний акт 1-ої категорії, 1-ої групи;
- технічний акт 1-ої категорії, 2-ої групи;
- технічний акт 2-ої категорії;
- технічний акт 3-ої категорії.

Алгоритм роботи з технічними актами починається з створення акту. Користувачі, які мають відповідні права, ініціюють створення нового технічного акта в системі, де вказують тип акту, дату створення, а також опис технічного стану тролейбуса. Для поточного технічного акту і технічного акту 1-ої категорії, 1-ої групи можливо підвантаження актуальних даних з датчиків, що автоматично оновлює відображення технічного стану тролейбуса без необхідності додаткових вручну введених даних. Це дозволяє забезпечити точність та оперативність обробки інформації.

Технічний акт є незмінним після його збереження в системі, що дозволяє запобігти помилкам або зміні історії обслуговування тролейбуса. Усі дані фіксуються на основі реальних параметрів тролейбуса на момент створення акту, і в подальшому їх неможливо редагувати.

3.1.5 Роботи з аваріями тролейбусів

Аварії, що виникають у тролейбусах, мають безпосередній вплив на їх технічний стан і можливість подальшої експлуатації. Для кожного

тролейбуса система моніторингу фіксує інформацію про аварії, які можуть бути класифіковані за рівнем критичності: критичні або незначні. В залежності від типу аварії, інтерактивна іконка троллейбусу на дашборді змінює колір, що дозволяє операторам швидко оцінити стан транспорту.

Якщо троллейбус має невідремонтовані критичні аварії, це свідчить про серйозні несправності, які роблять його небезпечним для подальшої експлуатації. У такому випадку, інтерактивна іконка троллейбусу на дашборді буде забарвлена в червоний колір. Це є сигналом для персоналу, що даний троллейбус потребує негайного ремонту, і його подальша експлуатація неприпустима, оскільки може призвести до аварії чи пошкодження обладнання.

Якщо троллейбус має невідремонтовані незначні аварії, це означає, що на ньому є проблеми, але вони не загрожують безпеці та можуть бути усунуті в рамках планового обслуговування. В цьому випадку, інтерактивна іконка троллейбусу на дашборді буде забарвлена в жовтий колір. Такий індикатор вказує на те, що необхідно звернути увагу на троллейбус, однак він все ще може бути використаний, якщо не виникає жодних небезпечних ситуацій.

Алгоритм роботи з аваріями:

- створення аварії;
- перегляд аварій;
- ремонт аварій;
- історія ремонтів.

В системі можна створювати аварії вручну, коли технічний персонал виявляє несправність, що потребує ремонту. Для цього необхідно вказати тип аварії, її опис, а також пріоритет ремонту (критична чи незначна).

Користувачі можуть переглядати поточні аварії, що виникли в троллейбусі. Інформація про аварії доступна для всіх відповідальних осіб і відображається на дашборді. Це дозволяє швидко оцінити стан транспорту і прийняти рішення про подальшу експлуатацію або ремонт.

Після усунення несправності відповідальна особа оновлює статус аварії в системі. Вказується дата ремонту, виконані роботи, а також особа, яка виконувала ремонт. Якщо аварія була критичною, після ремонту іконка тролейбусу змінює колір з червоного на звичайний, що свідчить про виправлення ситуації.

Система зберігає історію всіх ремонтів, що проводились на тролейбусах. Це дає можливість переглядати хронологію аварій та ремонту, аналізувати частоту виникнення проблем і визначати, які саме несправності часто повторюються. Це також допомагає у плануванні профілактичного обслуговування і мінімізації ризиків у майбутньому.

Завдяки цьому алгоритму, оператори можуть швидко реагувати на технічні проблеми і ефективно управляти станом тролейбусів, що забезпечує безпеку і надійність їх експлуатації.

3.2 Опис структури класів системи

Структура класів веб серверу автоматизованої системи моніторингу тролейбусного парку включає різноманітні компоненти, які реалізують бізнес-логіку додатку та сприяють зручності розробки і підтримки коду.

Основу серверної частини, створеної на базі ASP.NET Core, складають класи Program і Startup.

Клас Program є головною точкою входу в додаток. Він містить метод Main, який запускає програму. У цьому класі виконується базова конфігурація додатку, підключення служб і запуск веб сервера для обробки HTTP-запитів.

Клас Startup відповідає за налаштування додатку та його середовища. Він включає два ключових методи:

1. `ConfigureServices` — використовується для підключення служб, які будуть доступні в додатку. Тут реєструються всі необхідні залежності, наприклад, бази даних, системи аутентифікації, кешування тощо.

2. `Configure` — забезпечує налаштування обробки HTTP-запитів через `middleware`. У цьому методі визначається послідовність компонентів, які будуть обробляти запити.

Структуру класів можна розділити на кілька функціональних груп:

- контролери;
- API-класи;
- класи конфігурації;

Ця структура дозволяє зберігати код упорядкованим, підвищуючи модульність, масштабованість і простоту обслуговування системи.

3.2.1 Контролери

Контролери є ключовими функціональними компонентами серверної частини автоматизованої системи моніторингу тролейбусного парку. Вони реалізують основні функції, які забезпечують взаємодію клієнтської частини із сервером.

Усі контролери мінімізують кількість бізнес-логіки, дотримуючись принципів патерна "Медіатор". Їхнє головне завдання — передавати запити до медіатора та повертати отримані відповіді клієнтам.

До складу системи входять 7 контролерів.

`TrolleybusesController` — відповідає за обробку запитів, пов'язаних із тролейбусами:

- `GetTrolleybuses` — отримує список тролейбусів;
- `GetTrolleybus` — отримує інформацію про конкретний тролейбус за ідентифікатором.

`SettingsController` — керує налаштуваннями системи:

- `GetSettings` — отримує поточні налаштування;
- `PutSettings` — оновлює налаштування системи.

AuthController — забезпечує функції аутентифікації та управління користувачами:

- LogIn — виконує аутентифікацію користувача;
 - GetCurrentUser — отримує дані про поточного користувача;
- SensorsController — обробляє дані, що надходять від датчиків:
- AcceptDataFromSensors — отримує дані з тролейбусних сенсорів.

TechActsController — керує технічними актами:

- GetTechActs — повертає список технічних актів;
- GetTechAct — отримує дані про певний технічний акт;
- CreateTechAct — створює новий технічний акт;
- DeleteTechAct — видаляє існуючий технічний акт.

FaultsController — відповідає за реєстрацію та обробку несправностей:

- AddTrolleybusFault — реєструє несправність тролейбуса;
- RepairTrolleybusFault — відзначає ремонт несправності;
- GetFaults — отримує список зареєстрованих несправностей.

UsersController — дозволяє керувати обліковими записами користувачів:

- CreateUser — створює нового користувача;
- EditUser — змінює дані про існуючого користувача;
- DeleteUser — видаляє користувача;
- GetUsers — отримує список усіх користувачів системи.

VoltageGroupsController — обробляє дані, пов'язані з групами напруги:

– GetVoltageGroup — отримує інформацію про конкретну групу напруги;

– SearchVoltageGroup — здійснює пошук групи напруги за критеріями;

- GetVoltageGroups — повертає список усіх груп напруги.

Ця організація контролерів допомагає структурувати серверний код, забезпечуючи чітке розділення відповідальностей і модульність у розробці системи.

3.2.2 API-класи

API-класи потрібні для опису форматів запитів і відповідей, які використовуються для комунікації між вебсервером і клієнтським інтерфейсом. Ці класи використовуються як у серверній частині, так і на стороні користувачького інтерфейсу.

У системі наявні наступні API-класи:

- `TrolleybusDetailsDto` — описує детальну інформацію про тролейбус, включаючи його ідентифікатор, тип, стан, кількість несправностей та рівень заряду;

- `TrolleybusDto` — містить базову інформацію про тролейбус, зокрема його ідентифікатор, бортовий номер та тип;

- `EditManualTrolleybusFaultDto` — використовується для передачі даних про ручне додавання несправності, включаючи ідентифікатор тролейбуса, опис несправності, її назву, дату виявлення та ступінь серйозності;

- `TrolleybusFaultDto` — відображає інформацію про несправність тролейбуса, зокрема її джерело, серйозність, дати виявлення та ремонту, а також дані користувача, який її зареєстрував;

- `AuthResponseDto` — відповідає за передачу токена аутентифікації після успішного входу;

- `LogInDto` — містить дані для аутентифікації користувача: ім'я користувача та пароль;

- `SensorDataDto` — описує дані, отримані від датчиків, зокрема час, параметр та список значень;

- `SettingsDto` — містить налаштування для різних типів тролейбусів, таких як діапазони напруги;

– `TechActDto` — відображає технічний акт із великою кількістю параметрів, пов'язаних із перевіркою стану тролейбуса та його підсистем. Містить базові властивості, результати перевірок і додаткову інформацію залежно від типу акта;

– `TechActInfoDto` — стисла інформація про технічний акт, включаючи ідентифікатор, тип, користувача, що виконав перевірку, та дату;

– `ElementTechStateDto` — містить технічний стан окремого елемента тролейбуса, включаючи інформацію про шум, вібрації та робочий струм;

– `PressureDto` — представляє інформацію про позитивний і негативний тиск у системі;

– `ThermalExaminationDto` — описує температурні показники для теплового обстеження окремих компонентів;

– `VoltageGroupDto` — включає дані про групу напруги тролейбуса, її блоки, падіння напруги та температури;

– `UserDto` — відображає дані користувача, включаючи його ідентифікатор, ім'я, електронну пошту та роль у системі;

– `CreateUserDto`, `EditUserDto`, `DeleteUserDto` — забезпечують передачу даних для створення, редагування та видалення користувачів;

– `VoltageGroupSearchResultDto` — містить результати пошуку групи напруг, включаючи попередню, поточну та наступну групи;

– `VoltageGroupShortDto` — стисло описує групу напруги, зокрема ідентифікатор, рівень заряду та час отримання.

3.2.3 Класи-конфігурації

Класи конфігурації використовуються для зручного управління параметрами налаштувань проєкту. Всі вони реалізують відповідні інтерфейси, що забезпечує чітку структуру і спрощує роботу з конфігурацією. Управління конфігураціями здійснюється через можливості фреймворку ASP.NET Core.

Основним інтерфейсом конфігурації є `IAppConfig`. До кожного конфігу можна отримати доступ через `IOptionsMonitor`. В системі наявні 5 класів конфігурації.

`DatabaseConfig` містить параметри для налаштування підключення до бази даних:

- `ConnectionString` – рядок підключення до бази;
- `Retries` – кількість повторних спроб з'єднання;
- `Timeout` – максимальний час очікування відповіді від бази даних.

`JwtConfig` — визначає параметри роботи з JWT-токенами:

- `Issuer` – ідентифікатор видавника токенів;
- `Audience` – цільова аудиторія токена;
- `Authority` – адреса авторизаційного сервісу;
- `Key` – ключ для шифрування токенів.

`AuthConfig` містить параметри, пов'язані з доступом до ресурсів:

- `AllowAnonymous` – список ресурсів, до яких дозволено анонімний доступ.

`SensorsConfig` зберігає параметри, що стосуються обміну даними з датчиками:

- `Token` – токен для авторизації датчиків.

`VoltageRangeSettingsConfig` визначає діапазон допустимих значень напруги за замовчуванням:

- `Min` – мінімально дозволене значення;
- `Max` – максимально дозволене значення.

Класи конфігурації є ключовим компонентом структури системи, що дозволяє централізовано зберігати налаштування та ефективно їх використовувати.

3.3 Структура бази даних

Для забезпечення збереження і обробки даних моніторингової системи використовується база даних під управлінням системи керування базами даних (СКБД) MySQL. Програмний доступ до бази організований за допомогою Entity Framework Core 7, що дозволяє реалізовувати механізм міграцій, спрощуючи процес управління схемою бази даних.

Основою бази даних є ряд пов'язаних між собою моделей, які забезпечують структуру і організацію інформації. Нижче наведено короткий опис ключових класів:

- Trolleybus — центральний клас бази даних, що зберігає основну інформацію про тролейбуси: їхні бортові номери, тип, поточний стан, список технічних актів, зареєстровані несправності, групи напруги, а також історію аварійних сигналів;

- VoltageGroup — клас для зберігання даних про групи напруги, які періодично зчитуються із сенсорів тролейбусів. Група включає в себе блоки напруги, дату отримання даних та джерело інформації;

- TbsDniproUser — користувачі системи моніторингу, які виконують функції адміністраторів або технічного персоналу. Клас містить інформацію про імена, електронні адреси та дані для аутентифікації;

- TechAct — технічний акт, який описує виконані перевірки або обслуговування тролейбуса. Акти класифікуються за категоріями і групами, зберігають дату проведення, коментарі, а також інформацію про відповідального користувача;

- Category1Group1TechAct, Category1Group2TechAct, Category2TechAct, Category3TechAct — підкатегорії технічних актів, які відображають спеціалізовані перевірки, наприклад, стан компресорів, мікроклімату, з'єднань або вузлів. Ці класи фіксують конкретні параметри, такі як температурні показники, тиск, рівень охолоджувальної рідини тощо;

– `CurrentStateTechAct` — технічний акт для реєстрації поточного стану тролейбуса. Містить інформацію про доступність систем і сенсорів, а також ключові параметри, такі як напруга та температура в групах напруги;

– `TbsDniproSettings` — налаштування для різних типів тролейбусів: з коротким, середнім і довгим пробігом. Містить параметри діапазонів напруги та інші важливі конфігурації;

– `TrolleybusFault` — несправності тролейбуса, що реєструються в системі. Клас включає дані про виявлені та усунені несправності, їх джерело і ступінь серйозності;

– `ManualTrolleybusFault` — клас для несправностей, які було додано вручну.

Для роботи з базою використовується механізм міграцій `Entity Framework Core`. Це дозволяє автоматизовано створювати, оновлювати або відкочувати структуру бази до певної версії. При розробці додатку не було необхідності створювати SQL-скрипти вручну — всі необхідні таблиці створюються автоматично на основі визначених моделей.

Для спрощення доступу до бази даних використовується єдиний клас репозиторію — `TbsDniproRepository`, який забезпечує управління всіма основними сутностями, такими як тролейбуси, технічні акти, групи напруги та несправності.

Розроблена структура бази даних дозволяє зберігати повну інформацію про стан тролейбусного парку, забезпечувати зручний доступ до даних і гнучко реагувати на нові вимоги, додаючи або змінюючи структуру за допомогою міграцій.

3.4 Реалізація графічного інтерфейсу

Основні рішення щодо розробки графічного інтерфейсу ґрунтуються на принципах зручності, сучасності та адаптивності для забезпечення найкращого досвіду користувача. Інтерфейс розроблений таким чином, щоб бути простим і зрозумілим для користувачів різного рівня підготовки, при цьому забезпечуючи необхідний функціонал для ефективної роботи з системою.

Інтерфейс реалізовано з використанням сучасних технологій, зокрема, Material UI (MUI), що дозволяє створювати естетично приємні та зручні для користувача компоненти. Це забезпечує інтеграцію з популярними інтерфейсними елементами, такими як кнопки, індикатори, таблиці, форми та інші компоненти, що відповідають сучасним вимогам.

Адаптивність інтерфейсу є важливою характеристикою, що гарантує коректне відображення на різних пристроях, від комп'ютерів до мобільних телефонів. Всі елементи адаптуються до розміру екрану, що робить систему доступною для користувачів незалежно від того, чи працюють вони на настільному ПК або мобільному пристрої.

Окрема увага приділена інтерактивним елементам, таким як відображення стану тролейбусів. Зображення тролейбуса змінює колір відповідно до його технічного стану, що дає користувачам чітке візуальне уявлення про стан кожного транспортного засобу. Крім того, система має індикатори для відображення стану напруги в акумуляторах та доступності зв'язку, що дозволяє користувачеві швидко оцінити, чи є які-небудь проблеми у системі.

Графічний інтерфейс також включає адміністраторську панель для налаштування і керування різними аспектами системи. Адміністратор має доступ до налаштувань тролейбусів, моніторингу стану парку та інших функцій, що забезпечують ефективну роботу системи.

Всі таблиці в інтерфейсі мають функції пагінації, фільтрації та сортування, що дозволяє користувачам швидко знаходити необхідну інформацію серед великих обсягів даних. Для зручності користувачів передбачена можливість експорту даних в PDF одним натисканням кнопки, що спрощує процес створення звітів або передачі інформації.

Таким чином, графічний інтерфейс системи розроблений з орієнтацією на зручність користувачів і забезпечує ефективний доступ до всіх необхідних функцій при мінімальних зусиллях з боку кінцевого користувача.

3.5 Інноваційність підходу реалізації системи

3.5.1 Використання сучасних технологій у транспортній автоматизації

У рамках даної роботи впроваджено сучасні технології розробки програмного забезпечення, зокрема ASP.NET Core, React і MySQL, які відповідають актуальним вимогам у сфері автоматизації міського транспорту. Інтеграція цих технологій дозволила створити масштабовану, продуктивну та надійну систему для управління тролейбусним парком. Завдяки високій гнучкості технологій система ефективно працює з великими обсягами даних, включаючи показники сенсорів, історію технічного стану транспортних засобів і технічні акти.

3.5.2 Використання архітектури, орієнтованої на домен

Архітектура системи побудована на принципах Domain-Driven Design (DDD), що дозволяє моделювати складні бізнес-процеси у сфері моніторингу й управління транспортними засобами. Цей підхід забезпечує деталізацію доменної моделі, орієнтованої на потреби користувачів, зокрема в таких аспектах, як управління технічними актами, реєстрація несправностей та

доступ користувачів. Використання DDD сприяє створенню системи, яка відображає реальні виклики й потреби міського транспорту, забезпечуючи ефективність її функціонування.

3.5.3 Інтеграція даних у реальному часі

Ключовою особливістю є інтерактивна робота з даними в реальному часі, що реалізовано завдяки технологіям React. Це дозволяє відобразити актуальну інформацію на панелі моніторингу без затримок, що є критично важливим для прийняття оперативних рішень щодо технічного стану транспорту. Такий підхід значно підвищує ефективність управління міським транспортом.

3.5.4 Безпека та централізована реєстрація

Для аутентифікації користувачів використовується технологія JWT, яка гарантує безпеку та цілісність даних. Реєстрація нових користувачів здійснюється централізовано через адміністративну панель, що дозволяє уникнути ризиків, пов'язаних із несанкціонованим доступом, і забезпечує контрольоване використання системи.

3.5.5 Гнучкість і розширюваність системи

Система розроблена з модульним підходом, що дозволяє інтегрувати нові функції, наприклад автоматичну генерацію технічних актів або прогнозування обслуговування. Це забезпечує її довготривалу актуальність та адаптивність до змін у транспортній інфраструктурі.

3.5.6 Унікальність розробки для української транспортної галузі

В Україні автоматизовані системи для управління та моніторингу громадського транспорту, особливо тролейбусного, є рідкістю. Більшість сучасних рішень зосереджені на легковому або вантажному транспорті. Натомість запропонована система враховує специфіку функціонування тролейбусного транспорту, включаючи роботу з технічними даними в реальному часі та управління технічними несправностями.

Ця розробка заповнює прогалину в автоматизації комунального транспорту, пропонуючи рішення, яке поєднує сучасні веб-технології та

інноваційний архітектурний підхід. Унікальність системи полягає у врахуванні реальних потреб українських тролейбусних парків, які потребують модернізації для покращення ефективності експлуатації.

Сучасні дослідження підтверджують, що більшість ініціатив у сфері громадського транспорту України спрямовані на оновлення рухомого складу, зокрема заміну старих тролейбусів на нові з автономним ходом або електробуси. Автоматизація управління маршрутами та моніторингом технічного стану транспорту, натомість, залишається недостатньо розвиненою. Експерти наголошують, що інтеграція реального часу даних може значно підвищити ефективність роботи комунального транспорту, зокрема тролейбусів [14][15].

Таким чином, створена система є прикладом інтеграції сучасних технологій у громадський транспорт, підвищуючи його безпеку, ефективність і відповідність вимогам сучасної транспортної інфраструктури України.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

У цьому розділі детально розглядаються мета та умови використання розробленої системи моніторингу, її ключові характеристики, особливості звернень до системи, вхідні та вихідні дані, а також повідомлення, що генерує система.

4.1 Призначення та умови застосування програми

Система моніторингу розроблена для забезпечення ефективного управління тролейбусним парком та оптимізації роботи технічного персоналу. Вона надає зручний інструмент для контролю технічного стану тролейбусів, реєстрації несправностей та ведення технічних актів. Завдяки інтерактивним функціям і зручному інтерфейсу система є корисною для адміністраторів, інженерів та працівників, відповідальних за обслуговування транспорту.

4.2 Характеристики програми

У цьому підрозділі окремо буде розглянуто структуру проєктів серверної частини та користувацького інтерфейсу.

4.2.1 Веб-сервер

Застосунок веб серверу являє собою сукупність C# проєктів згрупованих в один великий проєкт (solution). Структура проєкту веб серверу відображена на рис. 4.1. Архітектура розроблена таким чином, щоб забезпечити модульність, чітке розділення відповідальностей і можливість масштабування. Система складається з 15 проєктів.

Проєкт TbsDnipro.Database відповідає за роботу з базою даних. Основними завданнями цього проєкту є: зберігання контексту бази даних для інтеграції з Entity Framework Core та управління схемою бази даних за допомогою міграцій. Також наявний окремий проєкт TbsDnipro.Database.Migrations, що використовується для створення та виконання міграцій, що полегшує розгортання та оновлення бази даних.

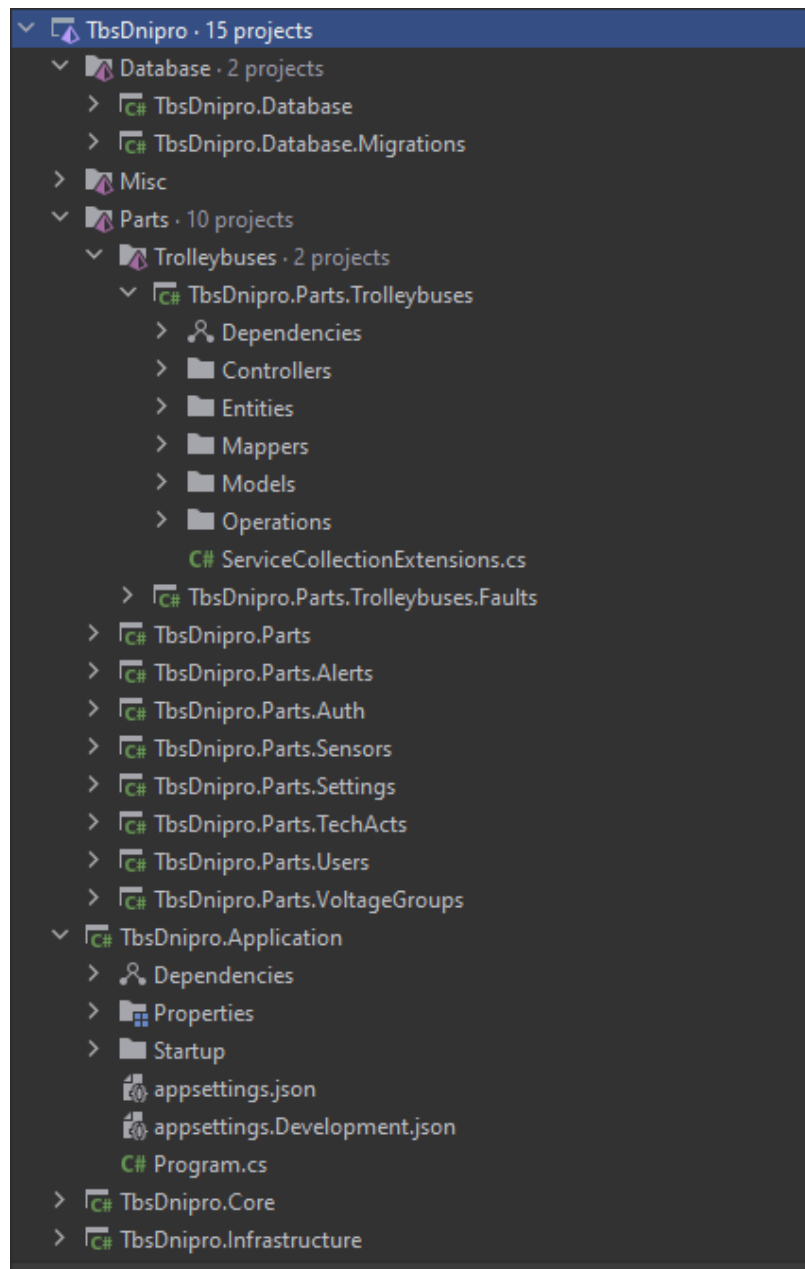


Рисунок 4.1 - Структура проєкту веб-серверу

Основним простором системи є `TbsDnipro.Parts`. Він містить окремі модулі для реалізації різних функцій системи.

- `TbsDnipro.Parts.Trolleybuses`;
- `TbsDnipro.Parts.Alerts`;
- `TbsDnipro.Parts.Auth`;
- `TbsDnipro.Parts.Sensors`;
- `TbsDnipro.Parts.Settings`;
- `TbsDnipro.Parts.TechActs`;

- TbsDnipro.Parts.Users;
- TbsDnipro.Parts.VoltageGroups.

TbsDnipro.Parts.Trolleybuses реалізує функціонал, пов'язаний із тролейбусами, включаючи: контролери для управління ендпоінтами; сутності та моделі даних; мапери для перетворення моделей; операції з бізнес-логікою (команди та запити); додатковий підмодуль TbsDnipro.Parts.Trolleybuses.Faults, який обробляє дані про несправності тролейбусів.

TbsDnipro.Parts.Alerts відповідає за обробку аварійних сигналів, включаючи їх збереження та оповіщення користувачів.

TbsDnipro.Parts.Auth реалізує аутентифікацію та авторизацію користувачів, зокрема інтеграцію з JWT (JSON Web Tokens).

TbsDnipro.Parts.Sensors обробляє дані, що надходять із сенсорів тролейбусів.

TbsDnipro.Parts.Settings зберігає конфігурації системи, включаючи параметри роботи тролейбусів.

TbsDnipro.Parts.TechActs відповідає за створення, зберігання та перегляд технічних актів.

TbsDnipro.Parts.Users реалізує управління користувачами системи, включаючи адміністраторів і технічний персонал.

TbsDnipro.Parts.VoltageGroups обробляє дані про групи напруги тролейбусів.

Кожен із модулів групи Parts має наступну внутрішню структуру:

- Controllers;
- Mappers;
- Models;
- Operations;

Controllers містить ASP.NET Core контролери, що містять ендпоінти. Вони викликають відповідні операції (команди та запити) за допомогою бібліотеки MediatR.

Mappers містить налаштування AutoMapper, що забезпечує автоматичне перетворення між різними моделями даних.

Models містить моделі, що використовуються для передачі даних між сервером і клієнтом.

Operations містить реалізація команд та запитів, які забезпечують бізнес-логіку ендпоінтів.

Проект TbsDnipro.Application містить головний вхідний пункт застосунку (Program.cs), налаштування середовища, запуск вебсервісу та конфігурацію залежностей через Startup.cs. Також тут зберігаються файли конфігурації, такі як appsettings.json.

Проект TbsDnipro.Core. Містить загальні утиліти, інтерфейси та базові класи, які використовуються у всій системі.

Проект TbsDnipro.Infrastructure. Реалізує інфраструктурні аспекти застосунку, такі як інтеграція з зовнішніми сервісами, логування, кешування та робота з чергами повідомлень.

Така модульна структура забезпечує гнучкість і масштабованість системи. Кожен проєкт виконує чітко визначені завдання, що спрощує підтримку, тестування та розвиток. Завдяки цьому вебсервер здатний адаптуватися до змін та легко інтегрувати новий функціонал.

4.2.2 Користувацький інтерфейс

Користувацький інтерфейс (UI) системи розроблений на основі React і має чітко структуровану архітектуру, що забезпечує зручність у підтримці, масштабуванні та розширенні функціональності. (рис. 4.2) Проєкт організовано у вигляді каталогу з папками та файлами.

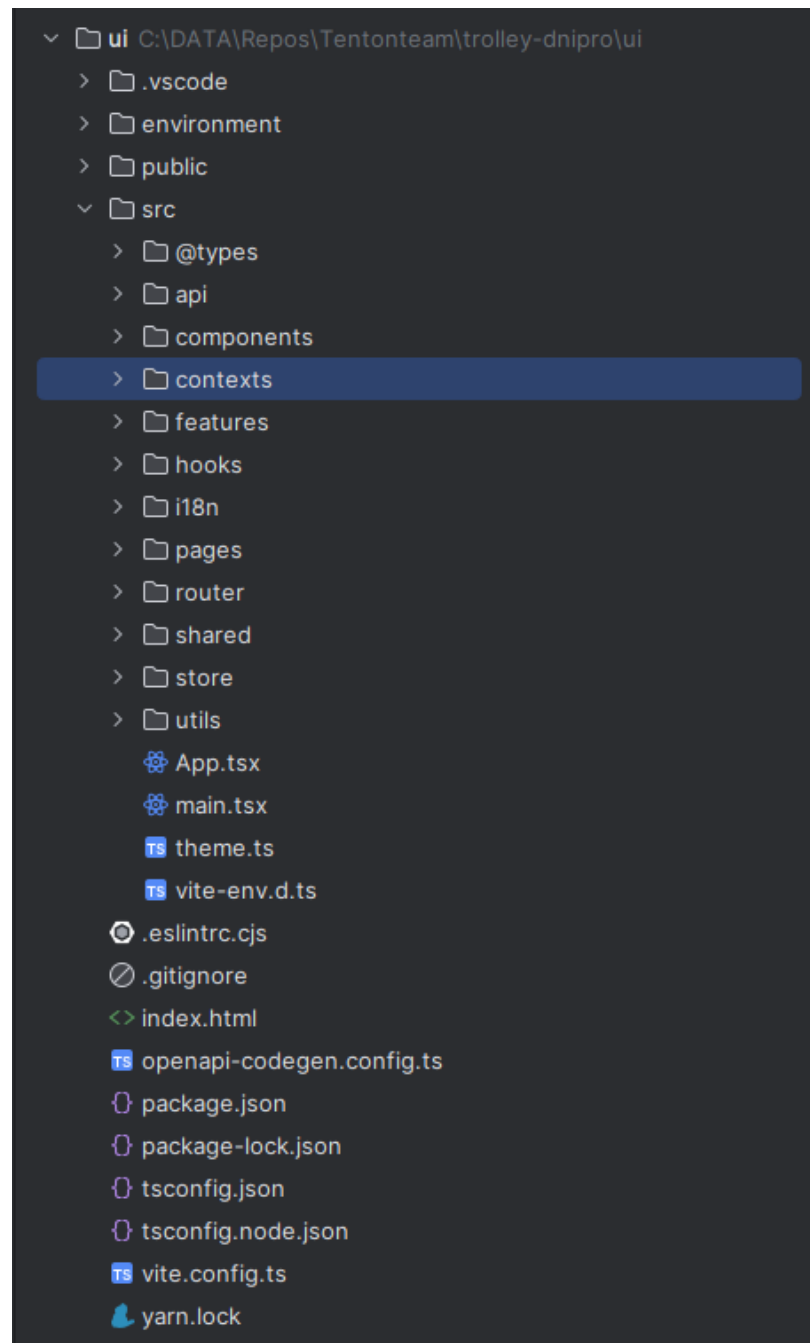


Рисунок 4.2 - Структура проєкту користувацького інтерфейсу

Папка “src” – це основна директорія, де знаходиться весь код користувацького інтерфейсу. Вона містить такі ключові підкаталоги:

– “types” — зберігає глобальні та користувацькі TypeScript-типи, які використовуються у всьому застосунку. Це дозволяє стандартизувати та типізувати дані.

– “api” — містить функції та модулі для взаємодії з бекендом через HTTP-запити. Зазвичай у цій папці зберігаються клієнти для API або специфічні запити до серверу.

– “components” — каталог із загальними компонентами, які використовуються в різних частинах застосунку. Наприклад: кнопки, таблиці, форми, модальні вікна тощо.

– “contexts” — містить контексти, створені за допомогою React Context API, які забезпечують глобальне управління станом і передавання даних між компонентами.

– “features” — реалізує функціональні модулі, які можуть включати компоненти, логіку та стилі, об’єднані в рамках певної бізнес-логіки.

– “hooks” — містить користувацькі React-хуки (custom hooks), що інкапсулюють загальну логіку роботи з API, обробки стану чи інших процесів, спрощуючи повторне використання коду.

– “i18n” — папка для підтримки інтернаціоналізації. Тут зберігаються переклади текстів, що використовуються в застосунку, у різних мовах.

– “pages” — зберігає компоненти, що відповідають окремим сторінкам застосунку, наприклад: сторінка логіну, дашборд, сторінка з графіками тощо. Кожна сторінка є контейнером, який організовує вкладені компоненти.

– “router” — містить конфігурацію маршрутизації за допомогою React Router. Визначає основні маршрути застосунку та компоненти, що відображаються для кожного маршруту.

– “shared” — каталог із загальними ресурсами, такими як стилі, теми чи спільні утиліти, які використовуються у всьому застосунку.

– “store” — реалізує централізоване управління станом застосунку (наприклад, Redux або інша бібліотека для state management).

– “utils” — зберігає допоміжні функції та утиліти, які спрощують виконання повторюваних завдань у коді (наприклад, форматування даних, обробка часу тощо).

Основні файли в корені “src”:

– “App.tsx” — головний компонент застосунку, який об’єднує всі маршрути, глобальні налаштування та контексти.

– “main.tsx” — точка входу застосунку. У цьому файлі підключається React до DOM через ReactDOM.render.

– “theme.ts” — визначає загальні налаштування теми для застосунку, такі як кольори, шрифти, стилі компонентів тощо.

– “vite-env.d.ts” — налаштування для Vite (інструменту для швидкої збірки React-застосунків).

Інші важливі каталоги проєкту:

– “environment” — містить конфігураційні файли для різних середовищ (наприклад, розробка, тестування, продакшн).

– “public” — директорія зі статичними ресурсами, такими як іконки, зображення, шрифти тощо.

Структура користувацького інтерфейсу забезпечує зручну навігацію по проєкту для розробників і чіткий поділ відповідальності між функціональними частинами. Завдяки використанню контекстів, хуків та компонентного підходу, цей застосунок є гнучким і легко розширюваним.

4.3 Звертання до програми

Вхідними даними в програмі є інформація, що надходить від різних сенсорів тролейбусів, зокрема дані про технічний стан, аварійні ситуації, показники напруги, а також дані, що вводяться користувачами, наприклад, про ремонтні роботи чи статуси несправностей. Користувачі також можуть вводити або оновлювати дані про технічні акти, користувачів системи та налаштування тролейбусів, що зберігається в базі даних. Вхідними даними є також запити користувачів через веб-інтерфейс або API, наприклад, для отримання інформації про конкретний тролейбус або його технічний стан.

Ці дані надсилаються через HTTP-запити у форматі JSON, що дозволяє системі обробляти інформацію та генерувати відповідні дії. Наприклад, при отриманні даних про несправність тролейбуса на сервер надходить JSON-об'єкт із детальною інформацією про тип несправності, її місцезнаходження та час реєстрації. Запити для оновлення даних про технічні акти або зміни в налаштуваннях тролейбусів також мають форму JSON-об'єктів.

Вихідними даними є відповіді на запити користувачів, зокрема інформація про технічний стан тролейбусів, статистика аварій та ремонту, згенеровані технічні акти, а також дані про статуси та налаштування системи. Вихідні дані також формуються у вигляді JSON-об'єктів, що містять результат виконання операцій або запитувану інформацію. Наприклад, при запиті про технічний стан тролейбуса система повертає JSON-об'єкт з інформацією про всі параметри тролейбуса, включаючи його напругу, стан сенсорів, останні аварії та інформацію про попередній ремонт.

4.4 API-інтерфейс серверної частини

Ендпоінти API-інтерфейсу серверної частини автоматизованої системи тролейбусного парку наведені у таблиці 4.1.

Таблиця 4.1 - Ендпоінти API інтерфейсу серверної частини

Маршрут	Http метод	Параметри запиту або тіла запиту	Опис
/api/trolleybuses	GET	–	Отримання масиву тролейбусів, що зареєстровані в системі

Продовження таблиці 4.1

Маршрут	Http метод	Параметри запиту або тіла запиту	Опис
/api/trolleybuses/{id}	GET	id (параметр запиту) – унікальний ідентифікатор тролейбусу	Отримання сутності тролейбусу за id
/api/faults	POST	тіло запиту повинно містити модель пошкодження	Реєстрація пошкодження в тролейбусу у системі
/api/faults/{id}	PATCH	id (параметр запиту) – унікальний ідентифікатор пошкодження repairedAt (параметр запиту) – дата ремонту	Ремонт пошкодження тролейбуса
/api/sensors	POST	тіло запиту повинно містити дані з сенсорів	Реєстрація даних з сенсорів в системі
/api/auth/user	GET	–	Отримання поточного користувача за його токеном
/api/auth/login	POST	тіло запиту повинно містити логін і пароль користувача	Аутентифікація і авторизація користувача у системі
/api/settings	POST	тіло запиту повинно містити модель налаштувань	Зміна налаштувань системи
/api/settings	GET	–	Отримання налаштувань системи

Продовження таблиці 4.1

Маршрут	Http метод	Параметри запиту або тіла запиту	Опис
/api/faults	GET	trolleybusId (параметер запиту) — унікальний ідентифікатор тролейбусу page (параметер запиту) — номер сторінки size (параметер запиту) — розмір сторінки isRepaired (параметер запиту) — фільтр, що визначає які пошкодження необхідно отримати: відремонтовані чи невідремонтовані	Отримання пошкоджень конкретного тролейбусу
/api/users	POST	тіло запиту повинно містити модель користувача	Реєстрація користувача в системі
/api/users	PATCH	Тіло запиту повинно містити модель користувача та його ідентифікатор	Зміна даних користувача
/api/users/{id}	DELETE	id (параметр запиту) — унікальний ідентифікатор користувача	Видалення користувача
/api/users	GET	page (параметер запиту) — номер сторінки size (параметер запиту) — розмір сторінки	Отримання користувачів системи
/api/techacts	POST	тіло запиту повинно містити модель технічного акту	Реєстрація технічного акту тролейбусу у системі

Продовження таблиці 4.1

Маршрут	Http метод	Параметри запиту або тіла запиту	Опис
/api/techacts/{id}	GET	id (параметр запиту) – унікальний ідентифікатор технічного акту	Отримання технічного акту
/api/techacts	GET	trolleybusId (параметр запиту) – унікальний ідентифікатор тролейбусу page (параметр запиту) – номер сторінки size (параметр запиту) – розмір сторінки date (параметр запиту) – фільтр дати технічного актів type (параметр запиту) – фільтр типу технічного акту	Отримання технічних актів системи

4.6 Повідомлення програмісту

Під час розробки та експлуатації системи можуть виникати нестандартні ситуації або помилки, що потребують уваги програміста. В таких випадках система генерує відповідні повідомлення, які допомагають виявити причину проблеми та вжити необхідних заходів для її виправлення. Всі виключні та нестандартні ситуації фіксуються в логах як серверної, так і клієнтської частини програми. Повідомлення про помилки або проблеми можна побачити в консолі, де виводяться деталі про тип помилки, її причину та можливі рекомендації щодо виправлення. Це дозволяє швидко

ідентифікувати помилки та вирішувати їх. Коли на серверній частині виникає помилка, вона записується в консоль за допомогою механізму логування. Повідомлення містить опис помилки, трасування стеку (stack trace) та інші корисні деталі. Якщо ж помилка виникає на клієнтській частині, наприклад, у JavaScript або при взаємодії з API, повідомлення також з'являється в консолі браузера, включаючи деталі про тип помилки і місце її виникнення. Повідомлення про помилки відіграють важливу роль у процесі підтримки та розвитку системи, дозволяючи швидко виявляти та усувати неполадки.

5 КЕРІВНИЦТВО ОПЕРАТОРА

У цьому розділі розглядаються призначення програми, умови та сценарії її використання.

5.1 Призначення програми

Програма призначена для автоматизації управління тролейбусним парком, забезпечуючи моніторинг технічного стану транспортних засобів, реєстрацію несправностей, створення технічних актів, опрацювання даних із сенсорів, а також управління користувачами й налаштуваннями системи. Вона спрямована на підвищення ефективності роботи парку, зменшення витрат і покращення обслуговування пасажирів.

5.2 Умови використання програми

Для ефективного використання автоматизованої системи управління тролейбусним парком необхідно забезпечити відповідні умови експлуатації як для апаратної частини, так і для користувачів.

Система потребує серверного обладнання, здатного підтримувати роботу веб-сервера та бази даних, з характеристиками, що відповідають зазначеним у технічній документації. Для роботи клієнтської частини необхідно мати комп'ютер або мобільний пристрій із сучасним веб-браузером, що підтримує JavaScript та роботу з API.

Оператор системи повинен володіти базовими навичками роботи з комп'ютером або мобільним пристроєм, мати доступ до мережі Інтернет та бути ознайомленим із функціональними можливостями системи через відповідне керівництво користувача. Інтерфейс системи наразі підтримує українську та англійську мови, тому оператор повинен володіти однією з них для комфортної роботи.

Для забезпечення повної функціональності системи важливо регулярно проводити технічне обслуговування обладнання, оновлення програмного забезпечення та резервне копіювання бази даних.

5.3 Сценарії використання програми

У цьому підрозділі будуть детально розглянуті сценарії використання автоматизованої системи тролейбусного парку операторами.

5.3.1 Аутентифікація

Як вже було зазначено у минулих розділах доступ до програми можуть отримати тільки зареєстровані користувачі. Саме тому перший екран,

що зустрічає операторів - це екран аутентифікації (рис. 5.1). Цей екран являє собою форму, де користувач повинен зазначити свої логін і пароль і натиснути кнопку “Увійти”. Якщо користувач введе неправильні дані, то він побачить відповідну помилку на екрані. Варто також зазначити, що якщо користувач неаутентифікований і він спробує перейти на будь яку з сторінок (вказавши відповідний шлях у браузері), то він буде переадресований на сторінку аутентифікації.

Рисунок 5.1 – Сторінка аутентифікації моніторингової системи

5.3.2 Перегляд технічного стану троллейбусного парку

Після вдалої аутентифікації користувача зустрічає головна сторінка системи (рис. 5.2). На цій сторінки видно всі троллейбуси парку і їх поточний стан в режимі реального часу. Троллейбуси займаються 3 різні вкладки, які відповідають їх типам: “з суперконденсатором”, “з коротким ходом”, “з довгим ходом”. Кожен троллейбус представлений іконкою, забарвленою відповідно до технічного стану, індикатором поточної напруги (якщо напруга вийшла за межі - колір червоний), а також індикатором зв'язку. Щоб переглянути більш детальну інформацію про конкретний троллейбус необхідно натиснути на відповідну іконку - після цього користувач буде перенаправлений на сторінку з деталізацією стану троллейбусу. Варто також зазначити, що у верхньому правому куті є дві кнопки: кнопка переходу до

адмін панелі (доступна тільки для адміністраторів) і кнопка “виходу” з системи.

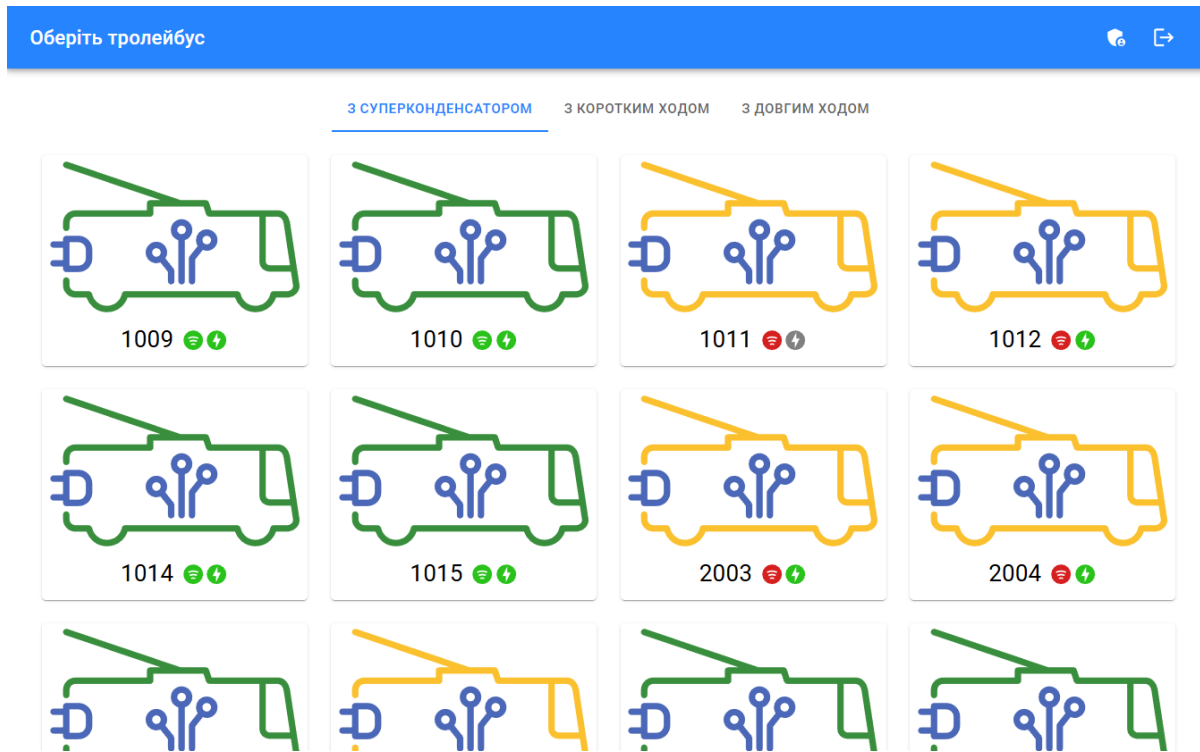


Рисунок 5.2 – Головна сторінка системи

5.3.3 Перегляд даних з датчиків

Натиснувши на один із тролейбусів і перейшовши на сторінку з деталями перший віджет, що користувач побачить перед собою буде віджетом з останніми отриманими даними з сенсорів (рис. 5.3). Велика таблиця з даними містить напругу кожної ячейки кожного блоку акумулятору, а кожен блок містить такі параметри як “Температура” та “Перепад напруги”. Також у верхній частині таблиці розміщена агрегована інформація із усіх блоків. Ця інформація включає напругу поточного заряду, перед температур між блоками і перепад напруги між блоками. Система також передбачає перегляд даних з датчиків за датою. Дату можна обрати у спеціальному селекторі, що знаходиться зверху таблиці.

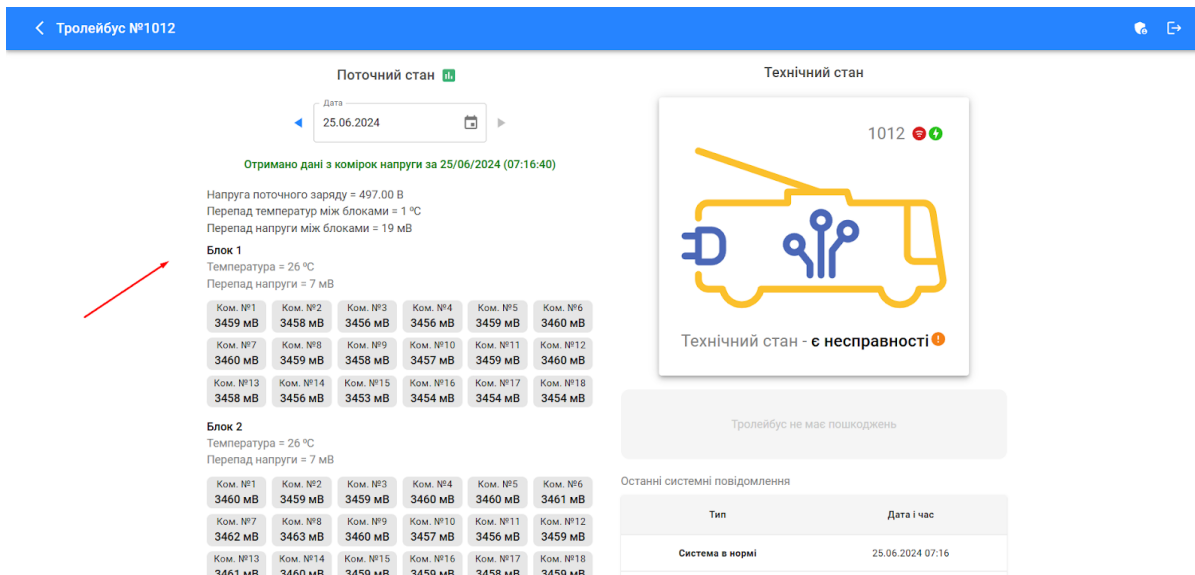


Рисунок 5.3 – Таблиця з даними з сенсори

Варто зазначити, що користувач має змогу подивитися на графік напруги поточного заряду акумулятору. Для цього йому необхідно натиснути зелену кнопку з правої частини від назви віджету “Поточний стан”. Користувач буде перенаправлений на сторінку з графіком, де він може обрати період (за тиждень або за місяць). Сторінка з графіком відображена на рис. 5.4.

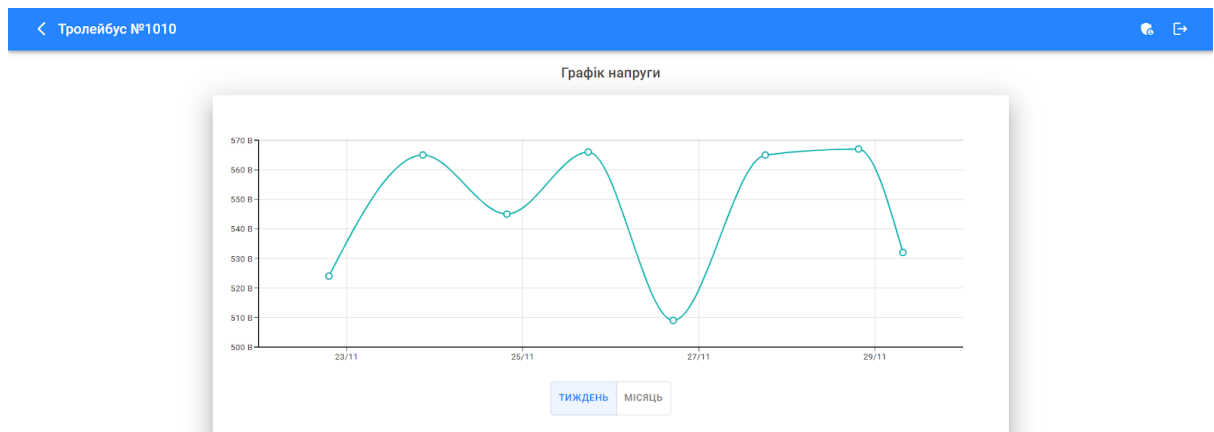


Рисунок 5.4 – Графік напруги поточного заряду

5.3.4 Робота з технічними актами тролейбусів

Перехід на сторінку з технічними актами можливий зі сторінки з детальною інформацією про вибраний тролейбус. Віджет з технічними актами відображає 5 останніх зареєстрованих в системі актів у таблиці. Під таблицею розміщена кнопка, що перенаправляє користувача на сторінку з технічними актами (рис. 5.5).

Блок 3
Температура = 23 °C
Перепад напруги = 8 мВ

Ком. №7 3455 мВ	Ком. №8 3455 мВ	Ком. №9 3454 мВ	Ком. №10 3441 мВ	Ком. №11 3451 мВ	Ком. №12 3450 мВ
Ком. №13 3452 мВ	Ком. №14 3450 мВ	Ком. №15 3449 мВ	Ком. №16 3453 мВ	Ком. №17 3451 мВ	Ком. №18 3448 мВ

Блок 4
Температура = 23 °C
Перепад напруги = 9 мВ

Ком. №1 3456 мВ	Ком. №2 3454 мВ	Ком. №3 3453 мВ	Ком. №4 3456 мВ	Ком. №5 3456 мВ	Ком. №6 3458 мВ
Ком. №7 3453 мВ	Ком. №8 3453 мВ	Ком. №9 3450 мВ	Ком. №10 3453 мВ	Ком. №11 3455 мВ	Ком. №12 3454 мВ
Ком. №13 3449 мВ	Ком. №14 3450 мВ	Ком. №15 3450 мВ	Ком. №16 3451 мВ	Ком. №17 3451 мВ	Ком. №18 3450 мВ

Блок 5
Температура = 23 °C
Перепад напруги = 8 мВ

Ком. №1 3455 мВ	Ком. №2 3454 мВ	Ком. №3 3453 мВ	Ком. №4 3451 мВ	Ком. №5 3450 мВ	Ком. №6 3450 мВ
--------------------	--------------------	--------------------	--------------------	--------------------	--------------------

Дата	Тип
29.11.2024 11:47	Система в нормі
29.11.2024 11:42	Система в нормі
29.11.2024 11:37	Система в нормі
29.11.2024 11:32	Система в нормі
29.11.2024 11:27	Система в нормі

Технічні акти (5 останніх)

Дата	Тип
05.07.2024	Поточний

ПЕРЕЙТИ ДО СТОРІНКИ ТЕХ. СТАНУ

ПЕРЕЙТИ ДО СТОРІНКИ ТЕХ. АКТИВ

Рисунок 5.5 – Віджет з технічними актами на сторінці тролейбуса

На сторінці з технічними актами розміщена відповідна таблиця з відсортованими за датою створення актами (рис. 5.6). Можливість визначати дату і тип технічних актів робить процес пошуку необхідного документу дуже швидким. Для кожного технічного акту в таблиці відображений його тип, дата створення і ім'я користувача, який його створив. Користувач також може обирати скільки технічних актів будуть відображені у таблиці - це стало можливим завдяки механізму пагінації. Якщо натиснути іконку з трьома крапками в останній колонці таблиці, то користувач може експортувати акт у PDF або видалити (доступно тільки для адміністраторів).

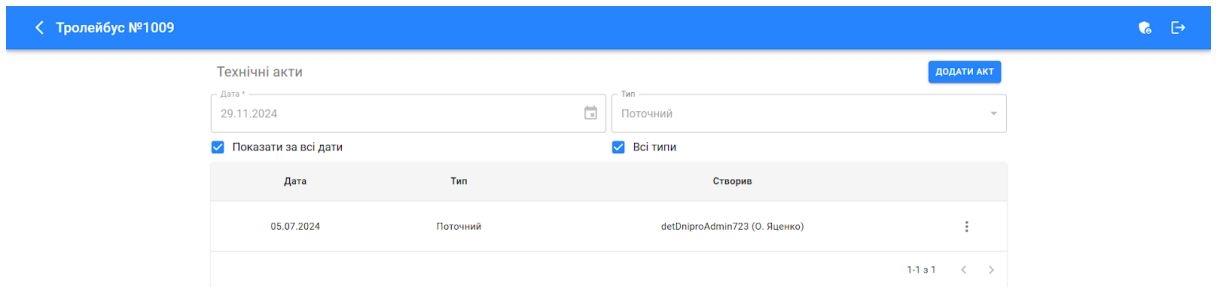


Рисунок 5.6 – Сторінка з технічними актами

Щоб створити новий технічний акт користувач повинен натиснути кнопку “Додати акт” на сторінці з технічними актами. Після того, як користувач натисне цю кнопку, він зможе обрати тип технічного акту і система переадресує його до сторінки створення технічних актів (рис. 5.7). Щоб створити технічний акт заповнити його форму і натиснути кнопку “Створити”. Процес заповнення дуже швидкий і за наявного досвіду може зайняти 20-30 секунд. На формі присутня валідація і, якщо користувач введе будь які неправильні чи невідповідні дані, то форма відреагує відповідним чином: поле з невалідними даними буде виділене червоним кольором і знизу з’явиться напис, що вкаже користувачеві на причину. Також варто зазначити, що деякі технічні акти (наприклад, “Поточний”) очікують від користувача даних з сенсорів тролейбуса і система надає можливість заповнити ці дані автоматично, що робить неможливим помилки викликані людським фактором (рис. 5.8).

← Створення технічного акту

Акт поточного стану Тролейбус №1009

Дата огляду*
29.11.2024

Чи працює підключення до систем керування?

Доступність вузлів

Чи працює головний екран?

Чи працює кондиціонер?

Чи працюють датчики?

Блок 1

Блок 2

Блок 3

Блок 4

Блок 5

Блок 6

Блок 7

Блок 8

Рисунок 5.7 – Сторінка створення технічного акту

Чи наявні пошкодження блоку мікроклімату?

Напруга на комірках, мВ

ОТРИМАТИ ДАНІ ЗА 29/11

Блок 1

1 Температура, °C*

1 ком.	2 ком.	3 ком.	4 ком.	5 ком.	6 ком.
7 ком.	8 ком.	9 ком.	10 ком.	11 ком.	12 ком.
13 ком.	14 ком.	15 ком.	16 ком.	17 ком.	18 ком.

Блок 2

1 Температура, °C*

1 ком.	2 ком.	3 ком.	4 ком.	5 ком.	6 ком.
7 ком.	8 ком.	9 ком.	10 ком.	11 ком.	12 ком.
13 ком.	14 ком.	15 ком.	16 ком.	17 ком.	18 ком.

Блок 3

1 Температура, °C*

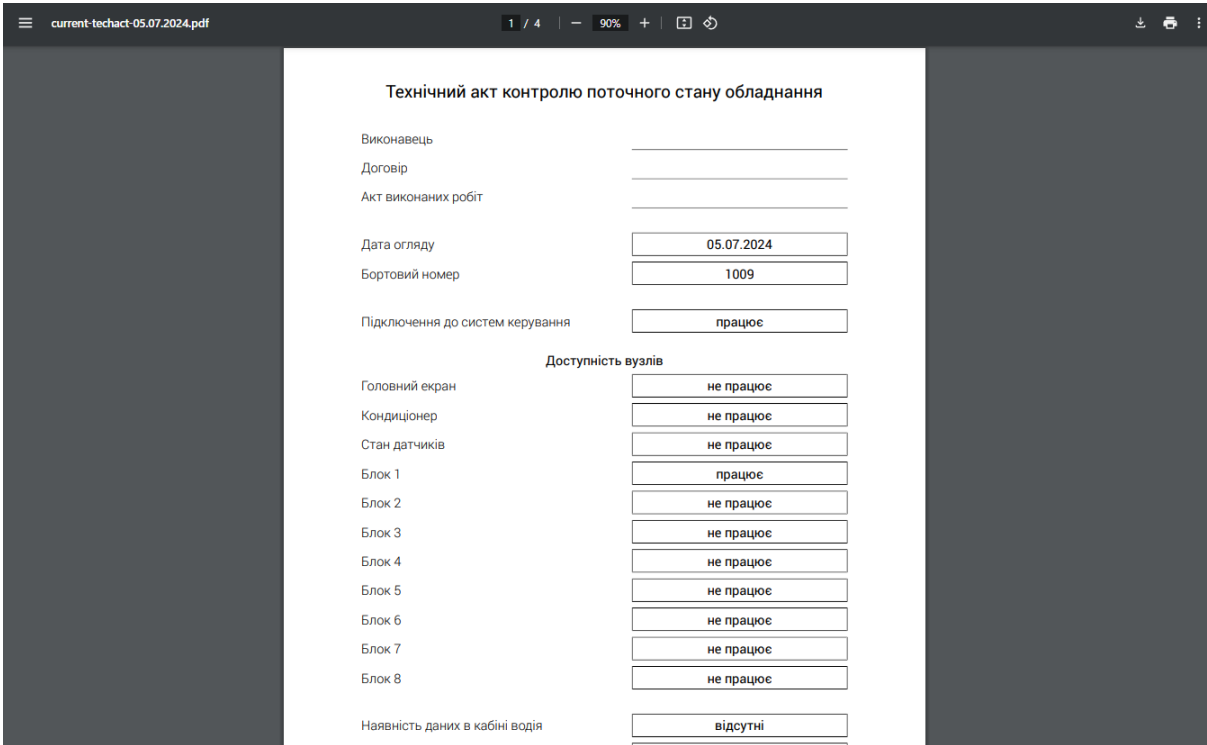
Рисунок 5.8 – Демонстрація можливості автоматично завантажити дані з сенсорів тролейбуса

Якщо на сторінці з технічними актами користувач натисне безпосередньо на будь-який з актів, то система переадресує його до сторінки, де він зможе переглянути вміст обраного документу (рис. 5.9). Всі дані на сторінці перегляду доступні тільки для читання, а у верхньому правому кутку розташована кнопка PDF експорту.

Технічний акт контролю поточного стану обладнання	
Виконавець	
Договір	
Акт виконаних робіт	
Дата огляду	05.07.2024
Бортовий номер	1009
Підключення до систем керування	працює
Доступність вузлів	
Головний екран	не працює
Кондиціонер	не працює
Стан датчиків	не працює
Блок 1	працює
Блок 2	не працює
Блок 3	не працює
Блок 4	не працює
Блок 5	не працює
Блок 6	не працює
Блок 7	не працює
Блок 8	не працює
Наявність палива в кабіні воєні	

Рисунок 5.9 – Сторінка перегляду вмісту технічного акту

Функція PDF експорту доступна користувачеві або з сторінки з технічними актами або з сторінки перегляду акту. Натиснувши на кнопку PDF експорту користувач перенаправляється на окрему вкладку з готовим документом (рис. 5.10). Документ можна як переглянути так і завантажити.



Технічний акт контролю поточного стану обладнання

Виконавець _____

Договір _____

Акт виконаних робіт _____

Дата огляду

Бортовий номер

Підключення до систем керування

Доступність вузлів

Головний екран	<input type="text" value="не працює"/>
Кондиціонер	<input type="text" value="не працює"/>
Стан датчиків	<input type="text" value="не працює"/>
Блок 1	<input type="text" value="працює"/>
Блок 2	<input type="text" value="не працює"/>
Блок 3	<input type="text" value="не працює"/>
Блок 4	<input type="text" value="не працює"/>
Блок 5	<input type="text" value="не працює"/>
Блок 6	<input type="text" value="не працює"/>
Блок 7	<input type="text" value="не працює"/>
Блок 8	<input type="text" value="не працює"/>
Наявність даних в кабіні водія	<input type="text" value="відсутні"/>

Рисунок 5.10 – Згенерований PDF варіант технічного акту на окремій сторінці

5.3.5 Робота з аваріями тролейбусів

Щоб перейти до сторінки аварій тролейбуса необхідно на сторінці з деталями обраного тролейбуса знайти віджет технічного стану і натиснути кнопку “Перейти до сторінки тех. стану” (рис. 5.11). Варто зазначити, що віджет технічного стану містить таблицю з 5-ма останніми невідремнтованими пошкодженнями/аваріями.

На сторінці з аваріями відображені три віджети: інтерактивна іконка тролейбусу, що відображає його поточний стан; таблиця з поточними пошкодженнями і аваріями, а також таблиця з ремонтами (рис. 5.12). Обидві таблиці відсортовані по даті і мають механізм пагінації. У таблицях містяться відповідні сутності: аварії і ремонти. Натиснувши на елемент таблиці можна отримати більш детальну інформацію про аварію чи ремонт.

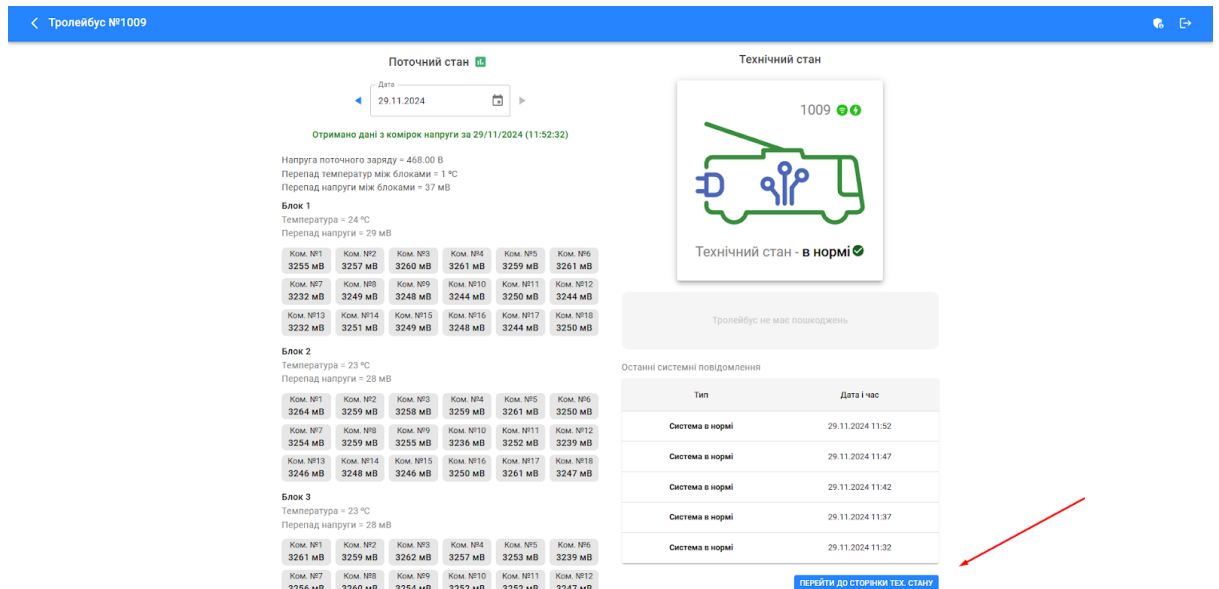


Рисунок 5.11 – Віджет технічного стану тролейбусу

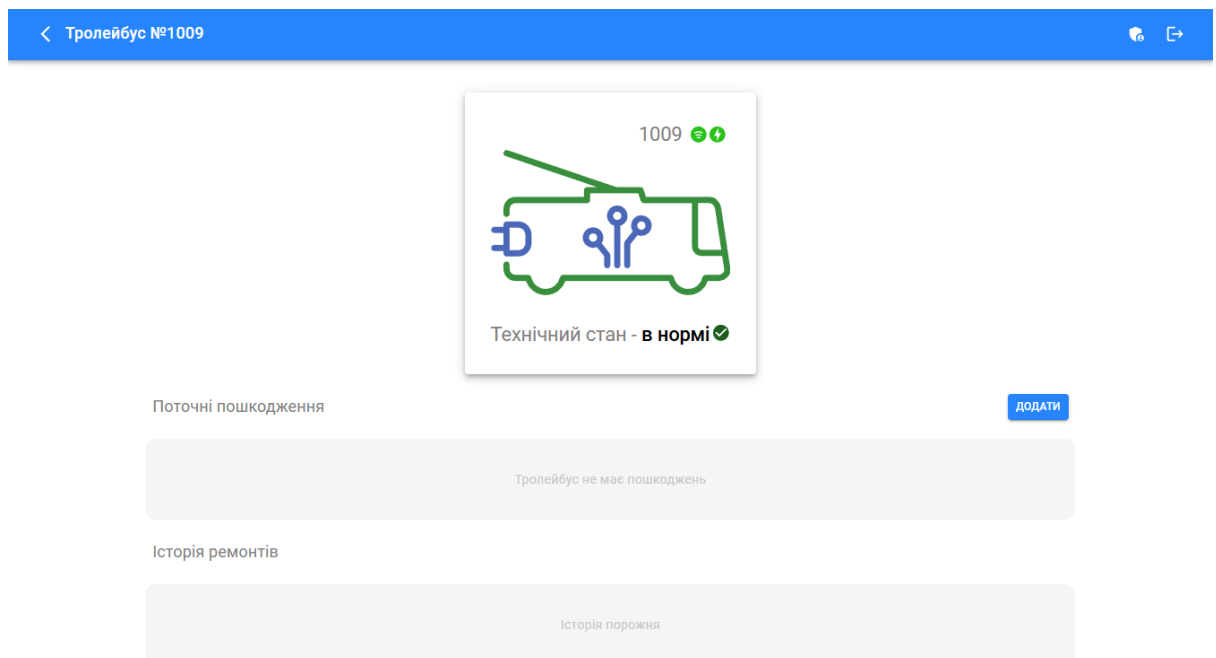
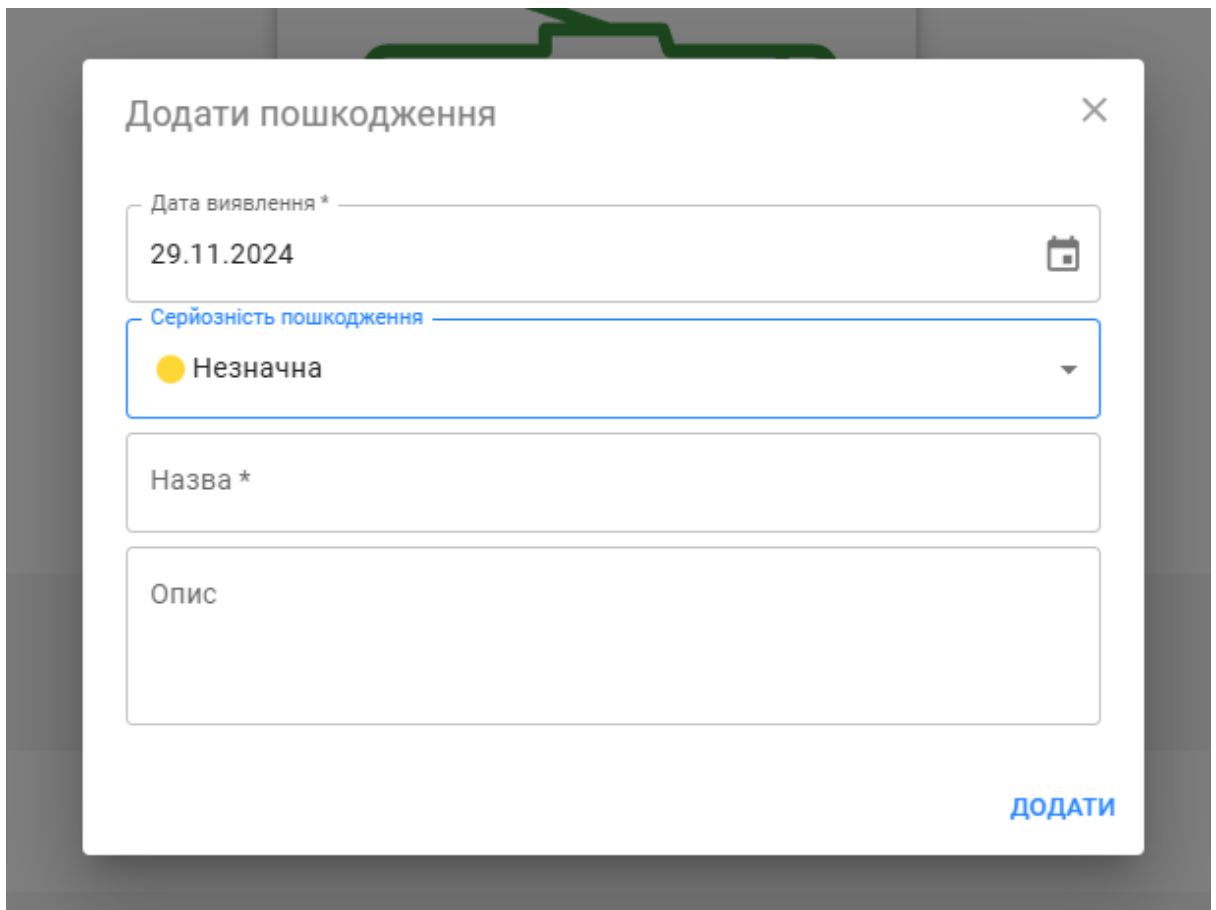


Рисунок 5.12 – Сторінка технічного стану тролейбусу

Робота з аваріями і пошкодженнями передбачає їх додавання у систему та ремонт. Щоб додати аварію необхідно натиснути відповідну кнопку у верхній правій частині віджету з пошкодженнями. Після натискання кнопки відкриється спеціальний діалог, де оператору необхідно зазначити інформацію щодо аварії, зокрема дату її виявлення, серйозність (незначна або критична), назву та опис (рис. 5.13). Додані аварії містяться у відповідній

таблиці до поки не будуть відремонтовані. Щоб відремонтувати аварію необхідно перейти до деталей аварії (необхідно натиснути на відповідну аварію у таблиці), вказати дату ремонту у відповідному полі і натиснути “Відремонтувати”. Історія ремонтів відображена у таблиці з ремонтами.



Додати пошкодження

Дата виявлення *

29.11.2024

Серйозність пошкодження

Незначна

Назва *

Опис

ДОДАТИ

Рис. 5.13 – Діалогове вікно реєстрації аварії тролейбуса

5.3.6 Зміна налаштувань системи

Як вже зазначалось у минулих підрозділах: такі параметри як гранична напруга заряду акумулятору можна програмно регулювати. Це можна зробити в налаштуваннях системи. Налаштування знаходяться в адмін панелі. Щоб увійти до адмін панелі необхідно натиснути відповідну кнопку, що знаходиться у правій частині хедеру веб сайту. Перейшовши до адмін панелі необхідно у меню, що знаходиться у лівій частині сторінки, обрати пункт “Налаштування”. У налаштуваннях є три секції відповідно до типів

тролейбусів. Регулювання границь допустимої напруги поточного заряду відбувається за допомогою спеціального веб-контролю (рис. 5.14). Після змін налаштувань необхідно натиснути кнопку “Зберегти”.

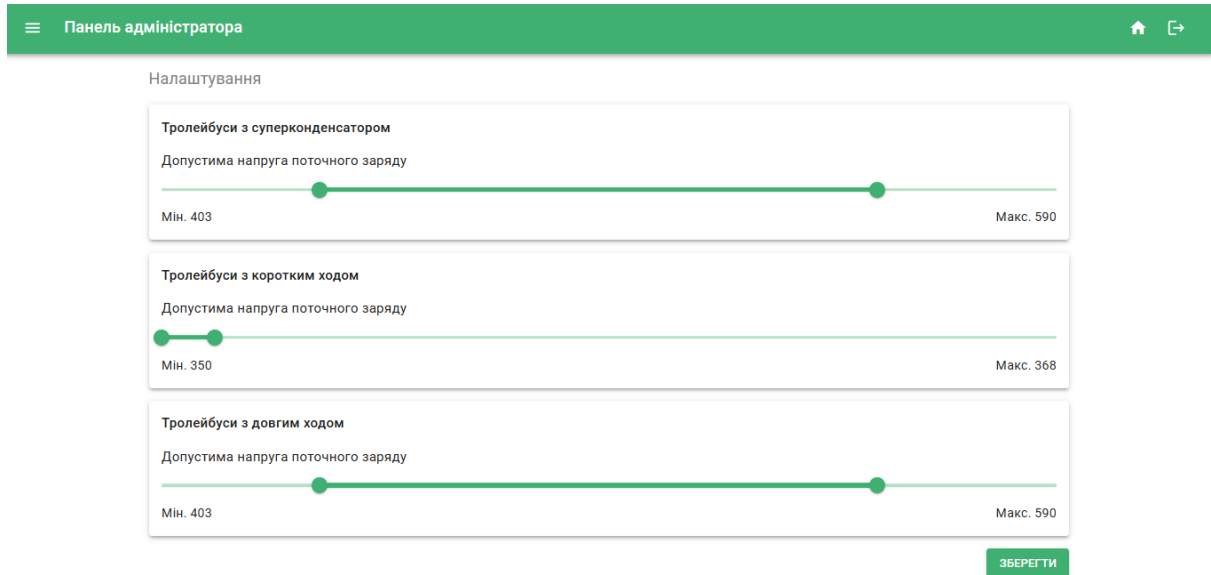


Рисунок 5.14 – Сторінка налаштувань системи

5.3.7 Робота з користувачами системи

Тут же в адмін панелі адміністратор може створювати і видаляти користувачів системи, а також змінювати їх і свої дані. Щоб перейти до сторінки роботи з користувачами необхідно у меню адмін панелі обрати пункт “Користувачі”. Сторінка для роботи з користувачами містить таблицю із зареєстрованими користувачами, а також інформацію профілю адміністратора (рис. 5.15). Щоб додати користувача адміністратор повинен натиснути кнопку “Додати”, що знаходиться у верхній правій частині над таблицею. Адміністратору необхідно вказати ім’я, прізвище, логін, пароль, електронну пошту та права доступу для нового користувача. Натиснувши на іконку з трьома точками в останній колонці таблиці навпроти користувача, адміністратор може змінювати його дані і права доступу, а також може його видалити з системи. Також варто зазначити, що адміністратор може змінити свої дані і пароль таким самим чином.

Панель адміністратора

Адміністратор

Логін	Ім'я	Email	
detDniproAdmin723	Олександр Яценко	yahteng@gmail.com	⋮

Користувачі системи ДОДАТИ

Логін	Ім'я користувача	Email	Права доступу	
SRSDET	Фазил Алекберов	srs.det@ukr.net	читання	⋮
Gladush	Олексій Гладуш	gladushalekseiy@gmail.com	читання	⋮
Bezugliy	Володимир Безуглий	rabochaya152@gmail.com	читання	⋮
depo1ingener	Олександр Ковба	depo1.ingener@gmail.com	читання	⋮
depo2ingener	Олександр Дворниченко	alecsandrdvornichenko@gmail.com	читання	⋮

1-5 з 8 < >

Рисунок 5.15 – Сторінка для роботи з користувачами в адмін панелі

ВИСНОВКИ

У рамках дипломної роботи було розроблено систему моніторингу, яка є першим кроком до оптимізації роботи тролейбусного парку. Система має значний потенціал для підвищення ефективності управління транспортною інфраструктурою.

Основна перевага системи полягає у можливості оперативного моніторингу технічного стану тролейбусів, зокрема електродвигунів, що дозволяє вчасно виявляти та усувати проблеми. Це сприяє зменшенню ризику серйозних несправностей, зниженню витрат на ремонт і підвищенню надійності транспортних засобів. Реєстрація несправностей та створення технічних актів забезпечують систематизацію даних про технічне обслуговування, що сприяє покращенню загального стану тролейбусів і підвищенню їх доступності для експлуатації.

Система має модульну архітектуру, що дозволяє поступово розширювати її функціонал відповідно до потреб користувачів. У перспективі можна додати функції маршрутизації та відстеження місцезнаходження транспортних засобів, що дозволить оптимізувати графіки руху та зменшити час очікування пасажирів. Подальший розвиток може також включати впровадження системи моніторингу витрат електроенергії, аналіз ефективності роботи водіїв, а також реалізацію інструментів дистанційного управління.

Результати цієї роботи створюють основу для впровадження сучасних технологій у сфері міського транспорту. Запропонована система сприятиме підвищенню продуктивності праці технічного персоналу, зниженню експлуатаційних витрат, а також покращенню обслуговування пасажирів. Таким чином, її впровадження може стати важливим етапом у модернізації роботи тролейбусного парку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 4905: 2008 Колісні транспортні засоби. ТРОЛЕЙБУСИ ПАСАЖИРСЬКІ. Загальні технічні вимоги – Київ: ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ, 2008. – 36 с.
2. Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design / [E. Mehrdad, Y. Gao, A. Emadi and others.]. – United States of America: CRC Press, 2004. – 424 p
3. Система електроживлення тролейбуса з високовольтним акумулятором. Мартюхін І. А. / Національний технічний університет України «Київський політехнічний інститут ім. І. Сікорського». – Київ, 2020
4. Тролейбус з автономним ходом [Електронний ресурс] – Режим доступу:
https://tvoemisto.tv/exclusive/u_lvovi_rozrobyly_pershyy_trolleybus_z_avtonomny_m_hodom_navishcho_vin_mistu_ta_kudy_poide_125910.html
5. X. Zhu, "A Dual-Active-Bridge Converter for Novel Trolleybus Powertrain System," 2018 Asian Conference on Energy, Power and Transportation Electrification (ACEPT), 2018, pp. 1–6, doi: 10.1109/ACEPT.2018.8610670.
6. Рухомий склад міського електричного транспорту. Механічна частина : навч. посібник / В. Х. Далека, М. В. Хворост, В. І. Скуріхін, Д. І. Скуріхін. ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018
7. Programmable Digital Thermometer DS18B20 [Електронний ресурс] – Режим доступу: <https://www.analog.com/media/en/technical-documentation/data-sheets/DS18B20>
8. Керівництво з експлуатації. Тролейбус E183D1-01, Львів – 2006
9. Tiahunova M., Kurychek H., Filippenkov D. The automated system of the trolleybus park as part of the sustainable city infrastructure. 5th International Conference on Sustainable Futures: Environmental, Technological, Social and

Economic Matters (ICSF 2024). 21-24 May, Kryvyi Rih, Ukraine. (Scopus) - прийнято до друку

10. Філіппенков Д. Ю., Тягунова М. Ю. Розробка автоматизованої системи тролейбусного парку / Комп'ютерно-інтегровані технології, автоматизація та робототехніка - 2024: матеріали І-ої Всеукраїнської конференції, Харків, 16-17 травня 2024.: тези доповідей / [редкол. І.Ш. Невлюдов (відповідальний редактор)].-Харків: [електронний друк], 2024. – С. 72-73.

11. Voltage transducer LEM LV 25-P [Електронний ресурс] – Режим доступу: <https://coefs.charlotte.edu/rcox3/2012/11/LV25-P-SP2>

12. Teltonika networks [Електронний ресурс] – Режим доступу: <https://teltonika-networks.com/products/gateways/trb145>

13. Remote management system. Teltonika [Електронний ресурс] – Режим доступу: <https://rms.teltonika-networks.com/documentation>

14. Держзвнiшiнформ "В Україні готують глобальне оновлення громадського транспорту" [Електронний ресурс] – Режим доступу: <https://dzi.gov.ua/press-centre/news/v-ukrayini-gotuyut-globalne-onovlennya-gromadskogo-transportu/>

15. Ліга Нет "Світ переходить на електробуси: в чому їх недоліки і чи потрібні вони Україні?" [Електронний ресурс] – Режим доступу: <https://tech.liga.net/ua/technology/article/mir-perehodit-na-elektrobusy-v-chem-ih-nedostatki-i-nujny-li-oni-ukraine>