

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

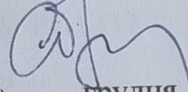
«Метод автоматизації процесів обслуговування користувачів мережі
електрозарядних станцій ECOSTATION»

Виконав: студент 2-ого курсу, групи ЗАКІТ-22м
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології



Іван НЕДОЛЯ

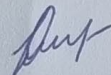
Керівник: Зав. кафедри АІТ, д.т.н., проф.



Олег БІСКАЛО

« 4 » грудня 2023 р.

Опонент: К.т.н, доц., проф. каф. КН



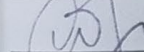
Олег КОЛЕСНИЦЬКИЙ

« 7 » грудня 2023 р.

Допущено до захисту

Зав. кафедри АІТ

д.т.н., проф.



Олег БІСКАЛО

« 11 » грудня 2023 р.

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти _____ II-й (магістерський)
Галузь знань – _____ 15 Автоматизація та приладобудування
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології
Освітня програма – Інформаційні системи і інтернет речей

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО

« 20 » вересня 2023 року

ЗАВДАННЯ

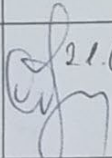
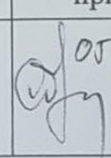
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Недолі Івану Руслановичу

1. Тема роботи: Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION
- Керівник роботи: Зав. кафедри АІТ, д.т.н., проф. Бісікало О.В.,
Затверджені наказом ВНТУ від « 18 » вересня 2023 року № 247
2. Термін подання студентом роботи: до « 05 » грудня 2023 року
3. Вихідні дані до роботи: архітектура системи; База даних MySQL підключена до серверної частини Node JS, платформа React Native в середовищі розробки WebStorm, конфігурація розумного маршрутизатора MikroTik, конфігурація контролеру зарядної станції VK Module Socket-2W
4. Зміст текстової частини: вступ; аналіз проблематики електрозарядних станцій; вибір оптимальних інформаційних технологій; розробка методу автоматизації процесів; висновки; перелік використаних джерел.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Акт про впровадження результатів магістерської кваліфікаційної роботи; Блок схем взаємозв'язків апаратної частини зі серверною; Діаграма Use Case методу комунікації та обміну даних інформаційної системи; Блок-схема алгоритму роботи методу комунікації; Діаграма Use Case модулю реєстрації та автентифікації; Діаграма Use Case інформаційної системи; Блок схема роботи програмного забезпечення інформаційної системи.

6. Консультанти розділів роботи

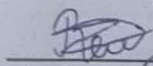
Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділи 1,2,3	Олег БІСІКАЛО	 21.09.2023	 05.11.2023

7. Дата видачі завдання « 21 » вересня 2023р.

КАЛЕНДАРНИЙ ПЛАН

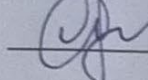
№ з/п	Назва етапів магістерська дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз проблематики електростанцій	22.09.2023 - 13.10.2023	випирав
2	Вибір оптимальних інформаційних технологій	14.10.2023 - 21.10.2023	випирав
3	Розробка методу автоматизації процесів обслуговування користувачів	22.10.2023 - 03.11.2023	випирав
4	Оформлення пояснювальної записки, графічного матеріалу і презентації	04.11.2023 - 26.11.2023	випирав
5	Попередній захист роботи	20.12.2023	випирав
6	Остаточний захист роботи	20.11.2023 - 18.12.2023	випирав

Студент



Іван НЕДОЛЯ

Керівник роботи



Олег БІСІКАЛО

АНОТАЦІЯ

УДК 004.7:621.3

Недоля І.Р. Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION. Вінниця: ВНТУ, 2023. ___ с.

На укр. мові. Бібліогр.: 42 назви; рис.: ____.

Магістерська робота присвячена розробці та впровадженню інформаційної системи для оптимізації процесів обслуговування та взаємодії користувачів мережі зарядних станцій для електромобілів. В контексті зростаючої популярності електрокарів та необхідності ефективної інфраструктури зарядних станцій, дослідження зосереджується на виявленні шляхів автоматизації цих процесів. У рамках роботи було проведено аналіз потреб користувачів та вивчено різні підходи до реалізації системи автоматизації мережі зарядних станцій EcoStation.

Як результат, була розроблена архітектура системи, яка включає модулі для ефективного обслуговування користувачів, управління зарядними процесами та забезпечення стабільної взаємодії. Окремо досліджено застосування передових технологій, зокрема мобільних та веб-додатків, для зручності користувачів.

Ця магістерська робота спрямована на вирішення актуальних завдань у сфері електромобільного транспорту та забезпечення потреби в стабільній та ефективній роботі інфраструктури зарядних станцій. Робота вносить важливий внесок у розвиток електромобільної інфраструктури та сприяє сталому розвитку транспортної галузі.

Ключові слова: автоматизація, електромобілі, зарядні станції, інформаційна система, оптимізація процесів.

ANOTATION

Nedolya I.R. Method of automation of customer service processes for the network of electric charging stations ECOSTATION. Vinnytsia: VNTU, 2023. ___ p.

The master's thesis is dedicated to the development and implementation of an information system for optimizing the service processes and user interaction of the electric vehicle charging station network. In the context of the growing popularity of electric cars and the need for an efficient charging station infrastructure, the research focuses on identifying ways to automate these processes. The work involved an analysis of user needs and the study of various approaches to implementing the automation system for the EcoStation charging station network.

As a result, a system architecture was developed that includes modules for efficient customer service, charge process management, and stable interaction. The application of advanced technologies, including mobile and web applications, for user convenience was also investigated.

This master's thesis aims to address current tasks in the field of electric vehicle transport and to meet the need for stable and efficient operation of the charging station infrastructure. The work makes an important contribution to the development of electric vehicle infrastructure and promotes the sustainable development of the transportation sector.

Key words: automation, electric vehicles, charging stations, information system, process optimization.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРОБЛЕМАТИКИ ЕЛЕКТРОЗАРЯДНИХ СТАНЦІЙ	10
1.1 Аналіз проблематики електрозарядних станцій	10
1.2 Технології та протоколи електрозарядних систем.....	12
1.3 Переваги та недоліки електромобілей, зарядних станцій	16
1.4 Огляд існуючих інформаційних систем електрозарядних станцій.....	18
1.5 Висновки	23
2 ВИБІР ОПТИМАЛЬНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	24
2.1 Аналіз можливостей мови програмування JavaScript	24
2.2 Аналіз фреймворку для розробки мобільних додатків React Native	25
2.3 Аналіз можливостей програмної платформи Node JS	29
2.4 Аналіз можливостей розумних маршрутизаторів MikroTik	32
2.5 Аналіз можливостей ethernet-контролерів VK Module	35
2.6 Висновки.....	37
3 РОЗРОБЛЕННЯ МЕТОДУ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ОБСЛУГОВУВАННЯ КОРИСТУВАЧІВ МЕРЕЖІ ECOSTATION	38
3.1 Розроблення архітектури інформаційної системи.....	38
3.2 Розроблення алгоритму методу автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION.....	44
3.3 Реалізація мобільного додатку та веб-порталу інформаційної системи ECOSTATION	47
3.3 Програмна реалізація методу автоматизація процесів обслуговування користувачів	51
3.4 Висновки.....	68

ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
ДОДАТКИ	75
Додаток А (обов'язковий) Технічне завдання.....	76
Додаток Б (обов'язковий) Ілюстративна частина	79
Додаток В (обов'язковий) Фрагмент лістингу програми	84
Додаток Г (обов'язковий) Протокол перевірки плагіату.....	93
Додаток Д (додатковий) Акт про впровадження МКР	94

ВСТУП

Актуальність теми. Сьогодні відбувається інтенсивне просування в напрямку електрифікації транспортного сектору. Електрокари стають все частіше обраною альтернативою та створюють вимоги до формування відповідної інфраструктури, зокрема потребують встановлення зарядних станцій. У 2018 році кількість проданих електромобілів перевищила два мільйони, і прогнозується, що до 2025 року буде реалізовано 10 мільйонів таких авто. Передбачається, що до 2040 року 57% всіх легкових автомобілів та понад 30% загального автомобільного парку будуть електричними. Очікуваний світовий споживчий попит на енергію для електромобілів може досягнути 280 мільярдів кВт/год у 2030 році, порівняно з 20 мільярдами у 2020 році. Однак, лише створення зарядних станцій не вистачає; ключовим є забезпечення їх ефективної та стабільної роботи, так, щоб водії могли легко та швидко заряджати свої авто. Тому актуальним є розробка інформаційної системи для оптимізації роботи користувачів мережі зарядних станцій.

Мета і задачі дослідження. Основною метою є створення методу автоматизації та відповідної інформаційної системи, яка спростить процеси обслуговування та взаємодії користувачів мережі зарядних станцій EcoStation. Для реалізації цієї мети необхідно вирішити наступні завдання:

- провести дослідження проблематики зарядних станцій та взаємодії користувачів з ними;
- створити метод автоматизації процесів обслуговування користувачів мережі електрзарядних станцій EcoStation;
- визначити оптимальні технології і платформи для розробки програмного забезпечення та інтеграції його з існуючою серверною архітектурою;
- реалізувати мобільний та веб-додаток для клієнтської частини.

Об'єкт дослідження представляє собою процеси обслуговування та взаємодії користувачів мережі електрзарядних станцій EcoStation.

Предмет дослідження охоплює методи, моделі та засоби автоматизації процесів обслуговування і взаємодії користувачів мережі електростанцій EcoStation.

Науково практичним результатом роботи є розроблення універсально методу комунікації електростанції з серверною частиною, автомобілем та апаратною частиною зарядки. Завдяки цьому дана розробка дозволяю використовувати її в любых проектах, в різних типах зарядних пристроїв до електромобілей з різними зарядними інтерфейсами, програмним та апаратним забезпеченням.

Методи дослідження розробки було використання платформ Node JS та React Native, на базі цих платформ були налаштовані API-запити для зв'язку з серверної-апаратної частини та виведення результатів роботи у клієнтських додатках.

Практична цінність даної розробки є те що вона вже реалізовано на підприємстві ТОВ «ЕкоТаксі-Вінниця» на протязі трьох років, проявила себе стабільністю, можливістю швидкого реагування при збоях та простому налаштуванню в нових пристроях.

Апробація результатів та публікація. В результаті виконання бакалаврської дипломної роботи, що отримала продовження розробки та дослідження як магістерська кваліфікаційна робота, було представлено доповідь на тему «Інформаційна система для оптимізації роботи користувачів мережі електростанцій EcoStation» на LI Науково-технічній конференції факультету інформаційних технологій та автоматизації Вінницького національного технічного університету (2022) із публікацією тез [1]. Планується також доповідь та публікація тез на Всеукраїнській науково-практичній інтернет-конференції [2] «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)».

1 АНАЛІЗ ПРОБЛЕМАТИКИ ЕЛЕКТРОЗАРЯДНИХ СТАНЦІЙ

1.1 Аналіз проблематики електрозарядних станцій

З посиленням проблем кліматичних змін і зростанням цін на нафту, інтерес та наукові дослідження в області електромобілів стають все актуальнішими. Електромобілі, завдяки своїм акумуляторам, здатні зберігати або постачати електроенергію в період нерухомості [3]. Таким чином, електромережа та парк автомобілів можуть взаємодіяти як комплементарні системи для управління енергією та потужністю. В ізольованих системах, як-от міських, впровадження електромобілів може стати вигідним не лише з екологічного погляду, зіставляючи з автомобілями на бензиновому паливі, але і з економічного боку, враховуючи місцеві енергетичні потреби.

Оскільки обмежений діапазон пробігу залишається одним з основних перешкод для масового впровадження електромобілів, створення інфраструктури зарядних станцій є критично важливим завданням для майбутнього. Для відповіді на зростання числа електромобілів ефективно застосовуються різні методи заряджання з різним функціоналом, які взаємно компенсують один одного. Існують численні технології заряджання електромобілів, включаючи стаціонарні, мобільні зарядні станції та інноваційні бездротові системи зарядки (рис. 1.1) [4].

Електромобілі завжди функціонують на енергії, збереженій в акумуляторах. Для руху вони використовують електродвигуни, які перетворюють збережену в батареях енергію. Термінологія і технології в області електромобілів постійно розвиваються. Більшість базових концепцій з'явилися навіть до розвитку двигунів внутрішнього згоряння [5].

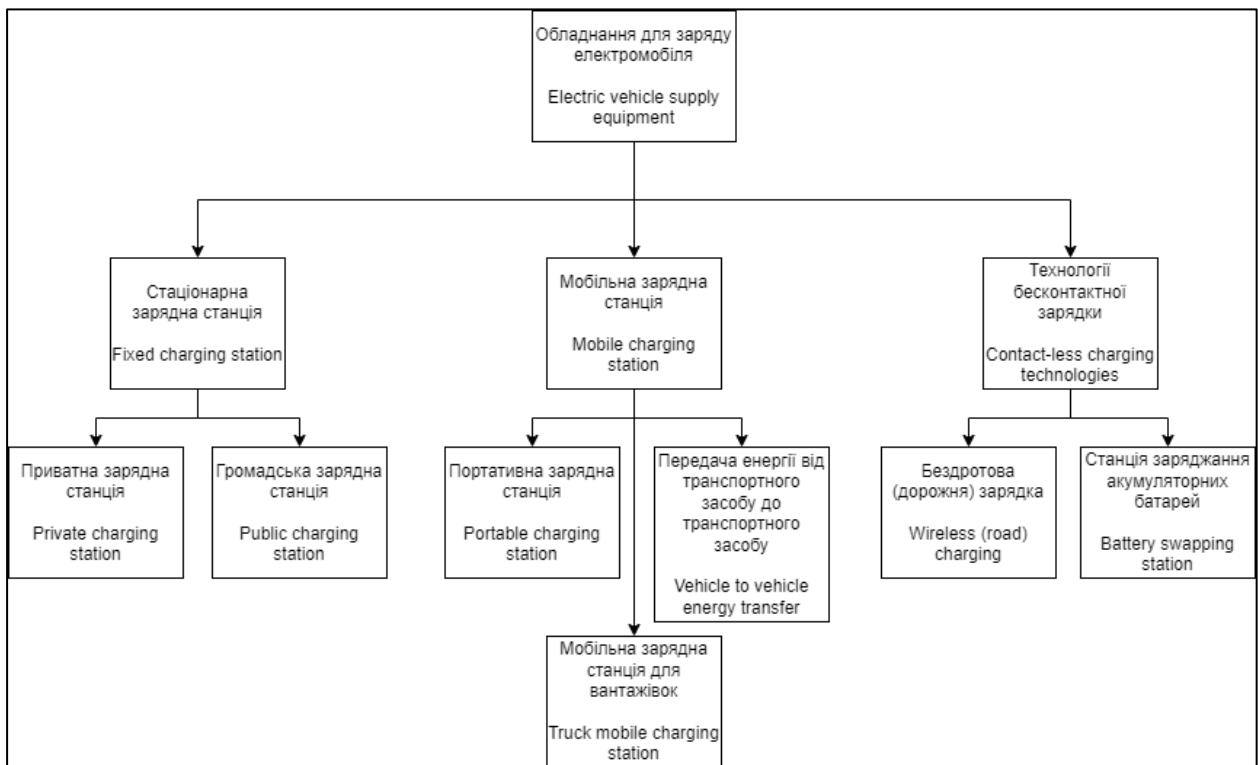


Рисунок 1.1 – Класифікація способів зарядання електромобілів

На початку 20-го століття електромобілі переважали на дорогах порівняно з автомобілями на бензині. Прогрес в технологіях акумуляторів, легкі конструкції, автоматизації електромереж та інші фактори підвищують інтерес до електромобілів серед споживачів, корпорацій та урядових організацій, підтримуючи тривалий перехід до більш ефективних транспортних систем [6].

В електромобілі відсутнє додаткове джерело живлення, тому після розрядки акумулятора, його потрібно зарядити. Зарядка відбувається через спеціальне зарядне устаткування, яке синхронізується з автомобілем і контролює процес зарядки для безпеки. На рис. 1.2 представлено схему потоку енергії для зарядки від електромережі через зарядне устаткування та автомобіль.

Електроніка зарядного пристрою конвертує змінний струм в постійний для зберігання в батареї автомобіля. Для подальшого руху електромобіля використовується двигун на змінний струм, який конвертується з постійного струму, збереженого в акумуляторі, за допомогою інвертора.

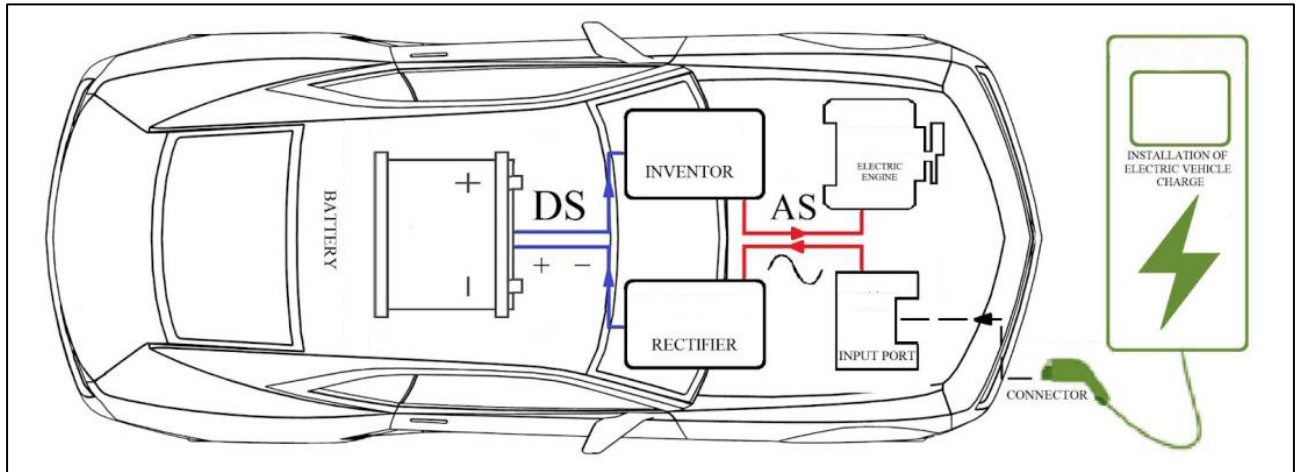


Рисунок 1.2 - Схема зарядного обладнання для електромобіля

1.2 Технології та протоколи електрозарядних систем

У процесі створення мережі електрозарядних пристроїв застосовуються різноманітні технології та методики зарядки електромобілів. Враховуючи різні автовиробники, кожен із яких використовує свої типи роз'ємів та стандарти для підзарядки, необхідно адаптувати мережі зарядних станцій до цієї різноманітності, щоб задовольнити потреби всіх типів електромобілів [7].

Кожен стандарт також має бути сумісний з особливостями електромереж у різних частинах світу, що додає складності у створенні універсальних зарядних станцій.

В контексті заряджання, критичними є не лише зарядні станції, але й зарядні кабелі та вбудоване зарядне устаткування. У цьому розділі зосередимо увагу на важливості вибору правильного зарядного кабелю і, зокрема, його роз'єму [8].

Роз'єми можна класифікувати відповідно до регіону, де вони в основному використовуються, але це узагальнення. Зарядні станції для змінного струму зазвичай не мають вбудованих кабелів, тому водії мають власні кабелі, які відповідають їхнім автомобілям [9].

А зарядні станції постійного струму мають вбудовані кабелі через вимоги безпеки та специфікації струму. Таким чином, правильний вибір роз'єму є важливим (рис 1.3).








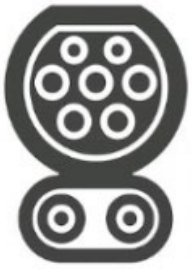
Current type and plug name	Region			
	Japon	China	America	Europe
AC				
Plug name	Type 1 - J1772	GB/T	Type 1 - J1772	Type 2
DC				
Plug name	CHAdeMO	GB/T	CCS - Type 1	CCS - Type 2

Рисунок 1.3 – Класифікація типів пістолетів та типів зарядних станцій

Комунікація між автомобілем та зарядною станцією включає передачу даних про статус зарядки, потреби автомобіля, параметри заряджання та інше. Саме тут на передній план виходять протоколи заряджання – вони є критично важливими для інтеграції зарядних систем у повсякденне життя споживачів та підтримки електромобілей[10].

Протоколи заряджання зародилися з потреби у стандартизації. Ранні дні електромобілів були хаотичними з точки зору зарядних технологій – кожен виробник пропонував своє унікальне рішення, що ускладнювало сумісність та доступність. Ця фрагментація створювала бар'єри для споживачів, ускладнюючи

використання електромобілів як повноцінних транспортних засобів. Така ситуація вимагала уніфікації, і тут на сцену вийшли протоколи заряджання.

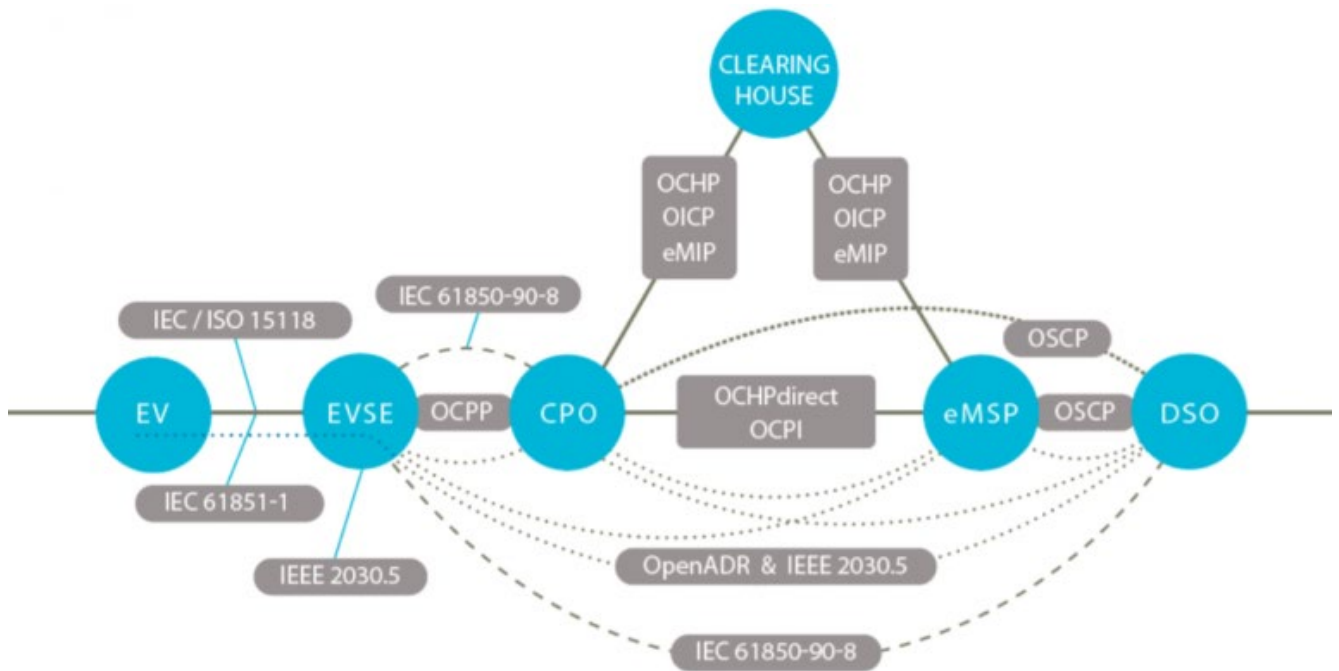


Рисунок 1.4 – Блок-схема протоколів, технічних звітів та зв'язок між ними.

Це може включати в себе бездротові технології та спеціальні протоколи для забезпечення правильної та безпечної зарядки. Протоколи заряджання для електрозарядних пристроїв описані нижче:

1. OCPP – Open Charge Point Protocol Розроблений Open Charge Alliance, цей протокол є фундаментальним у стандартизації комунікації між зарядними пунктами для електромобілів та їх керуючими системами. Відкритість OCPP дозволяє легко адаптувати мережі заряджання, мінімізуючи необхідність заміни обладнання. Протокол підтримує обмін даними про транзакції, управління зарядними станціями та їх технічне обслуговування, а також забезпечує сумісність і доступність послуг заряджання.

2. Open ADR - Open Automated Demand Response Standard Цей стандарт спрямований на автоматизацію відповідей на коливання попиту на енергію. Він дозволяє енергетичним системам і пристроям адаптуватися до потреб мережі за

допомогою гнучких тарифних планів та стимулів, а також аварійних сигналів для збалансування попиту і стабільного постачання енергії.

3. OSCP - Open Smart Charging Protocol Цей протокол передбачає інтелегентне управління енергетичними потужностями, дозволяючи мережам уникати перевантажень завдяки розподілу енергетичних бюджетів між системами. Протокол базується на принципі, де системи можуть заявляти свої потреби у центральній системі, що дозволяє оптимізувати споживання енергії та запобігати її марнотратству.

4. IEC 61850 Технічний звіт IEC 61850-90-8 визначає об'єктну модель для інтеграції електромобілів у енергетичні системи. Цей документ розглядає електромобілі як особливий вид розподіленого енергоресурсу, інтегрованого в загальну енергетичну мережу відповідно до стандартів IEC 61850.

5. EV Roaming Технологія роумінгу для електромобілів спрощує процес заряджання для водіїв, надаючи їм можливість користуватися різними зарядними станціями без обмежень. Це включає універсальний доступ до зарядної інфраструктури та забезпечує прозорість розрахунків за кожне заряджання.

6. OCPI – Open Charge Point Interface Protocol OCPI сприяє глобальному обміну даними між різними операторами зарядних станцій та роумінг-платформами. Він надає інформацію про доступність та вартість зарядних станцій, сприяючи прозорості та зручності для користувачів електромобілів.

7. OCHP – Open Clearing House Protocol Основна функція OCHP - обмін даними про дозволи на зарядку, транзакції та деталі зарядних станцій через e-clearing.net. Протокол включає два рівні комунікації: безпосередній обмін даними між учасниками ринку та комунікацію через інформаційну службу.

8. eMIP – eMobility Interoperation Protocol Розроблений GIREVE, eMIP спрямований на полегшення роумінгу заряджання електромобілів. Протокол має дві основні функції: забезпечення дозволу на заряджання та надання доступу до обширної бази даних про зарядні станції через API.

1.3 Переваги та недоліки електромобілей, зарядних станцій

Переваги володінням електромобілем:

1. Висока енергоефективність. Електрокари відзначаються високою ефективністю використання електроенергії, що на 33% ефективніше, порівняно з традиційними автомобілями, які використовують паливо. Електрокари отримують енергію з мережі та зберігають її в батареї, звідки вона перетворюється на механічну енергію для руху автомобіля. Ефективність електродвигунів значно перевищує ефективність двигунів внутрішнього згорання, що дозволяє значно зекономити енергію[11].

2. Екологічність. Електрокари не виробляють шкідливих викидів під час їзди, на відміну від автомобілів на бензині або дизелі, які викидають вихлопні гази. Це робить електрокари екологічно чистішими та більш зеленими, що сприяє збереженню навколишнього середовища та зменшенню впливу на клімат.

3. Зручність заряджання вдома. Можливість заряджання автомобіля вдома вночі або в будь-який зручний час виключає необхідність відвідування зарядних станцій, що дозволяє уникнути черг та економити час. Це також дозволяє підтримувати автомобіль у готовності до виїзду в будь-який момент.

4. Мінімізація обслуговування. Електрокари мають менше рухомих частин, що зменшує ризик поломок та зносу, а також знижує витрати на обслуговування та ремонт. Відсутність потреби в заміні масла, фільтрів та інших регулярних процедур забезпечує додаткову економію часу та грошей.

5. Розвиток інфраструктури. З підвищенням популярності електрокарів спостерігається і розвиток інфраструктури, що включає збільшення кількості зарядних станцій та введення державних ініціатив для підтримки електромобілей. Це сприяє створенню сприятливих умов для власників електрокарів та стимулює розвиток ринку електромобілів.

6. Електрокари сприяють зменшенню викидів CO₂ та інших парникових газів, що є важливим у контексті глобального потепління[12].

7. Постійне удосконалення технологій батарей та зниження їхньої вартості сприяють зростанню доступності та привабливості електромобілів для споживачів.

8. Збільшення кількості моделей електрокарів на ринку дозволяє споживачам вибирати з більшого розмаїття автомобілів, що відповідають їхнім потребам та бажанням.

Недоліки володінням електромобілей:

1. Обмежена дальність ходу. Дальність ходу є критичним фактором при виборі електромобіля. Залежно від розміру батареї, погодних умов, віку батареї та стилю водіння, дальність може варіюватися. Важливо планувати маршрут, враховуючи доступність зарядних станцій, особливо на довгих дистанціях[13].

2. Обмежена інфраструктура підзарядки. Наявність зарядних станцій залежить від місцевості. В деяких районах їх достатньо, у інших може бути важко знайти доступну станцію. Розвиток інфраструктури є важливим для зменшення "запасу тривожності" серед водіїв електромобілів.

3. Час на "заправку". Заряджання електромобіля займає більше часу, ніж заправка традиційного автомобіля. Швидкі зарядні станції можуть зменшити цей час до 30 хвилин, але вони все одно вимагають більше планування та часу.

4. Вартість ремонту або заміни батареї. Батареї є однією з найдорожчих частин електромобіля, і їх ремонт або заміна може бути коштовними. Виробники надають гарантії на батареї, але після їх закінчення власник автомобіля буде відповідальний за витрати на ремонт.

5. Вплив на навколишнє середовище та утилізація. Виробництво електромобілів та їхні батареї має свій вплив на навколишнє середовище через використання рідкісних матеріалів та енергії. Утилізація батарей також є викликом, хоча існують методики їх повторного використання та переробки[14].

1.4 Огляд існуючих інформаційних систем електрзарядних станцій.

У контексті стрімкого розвитку електромобілей, ключовим аспектом є ефективне управління електрзарядними станціями. Проект компанії Advantech відображає сучасні тенденції в цій галузі, демонструючи як інтеграція передових технологій та стандартів, таких як OCPP (Open Charge Point Protocol), сприяє оптимізації роботи зарядних станцій. Компанія, розробила власну мережу електрзарядних станцій на базі власних розробок та компонентів [15].

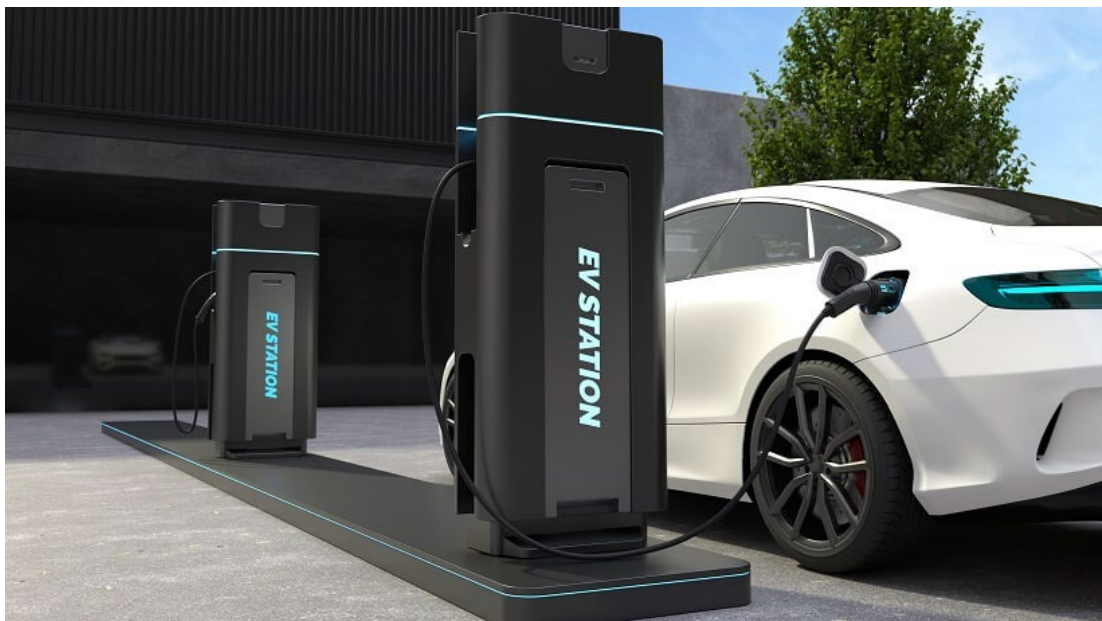


Рисунок 1.5 – Зображення зарядних станцій компанії Advantech

Основним завданням проекту було забезпечити простий, зручний та надійний процес зарядки для користувачів електромобілів. Для цього використовувалася обладнання Advantech, зокрема, некерований Ethernet-комутатор і ультрашвидкісний 5G маршрутизатор для збору та передачі даних [16]. Система передбачала, що інформація про оплату, введена користувачем, передається в віддалений дата-центр через маршрутизатор, а потім — до контролера зарядної точки, який активує струм для зарядки електромобіля [17]. Більш детально, складові мережі електрзарядних станцій компанії Advantech описано нижче:

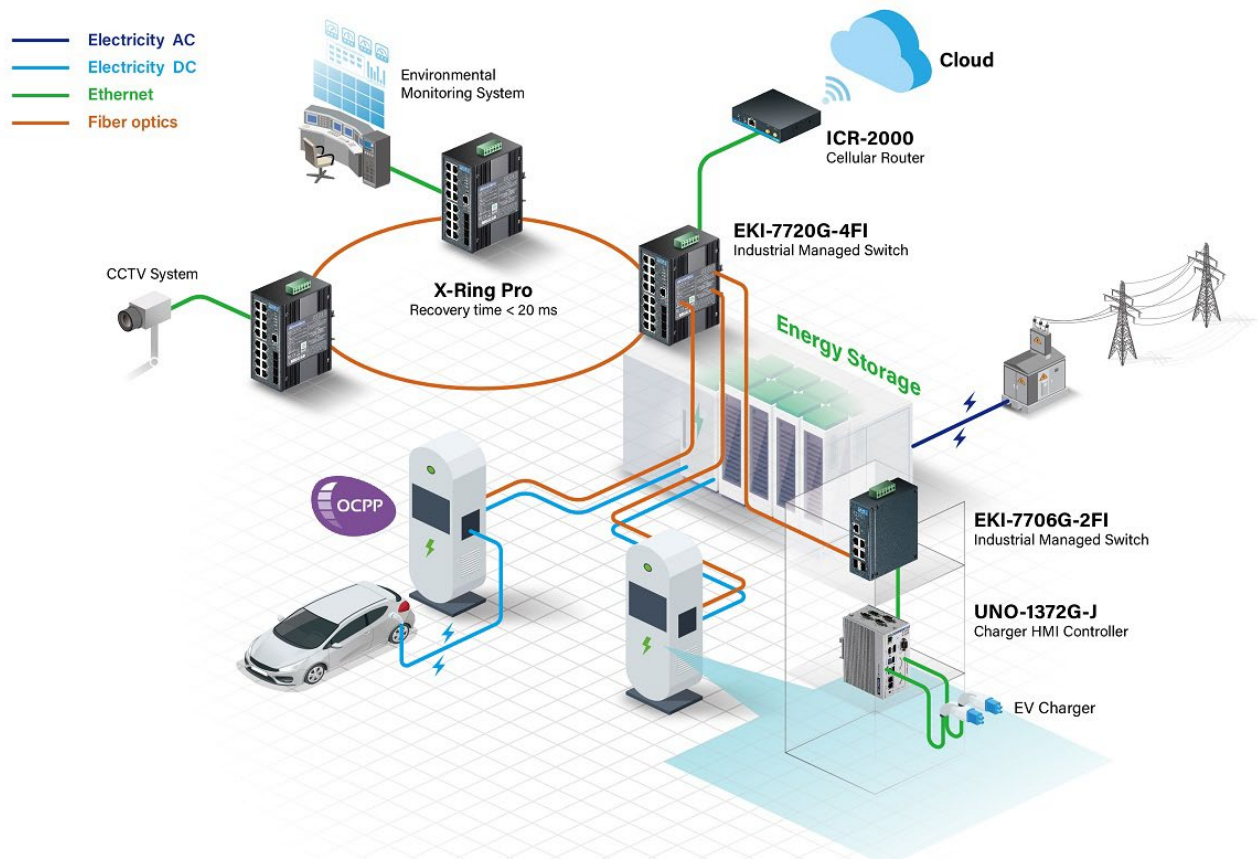


Рисунок 1.6 – Блок-схема архітектури мережі електрзарядних станцій Advantech.

1. Відеоспостереження (CCTV System): Пристрій відео-нагляду підключений до мережі, що гарантує безпеку на зарядній станції.
2. X-Ring Pro: Високопродуктивний пристрій мережевого комутації, забезпечує швидке відновлення з'єднання.
3. Система моніторингу навколишнього середовища: Цей компонент дозволяє відстежувати зовнішні умови, що можуть впливати на роботу зарядних пристроїв.
4. ICR-2000 Cellular Router: Підключає всю систему до хмарних сервісів для дистанційного управління та моніторингу.
5. Хмарні сервіси (Cloud): Хмарна інфраструктура для збору, обробки та аналізу даних від усіх компонентів системи.

6. EKI-7720G-4FI Industrial Managed Switch: Пристрій керування мережею, що забезпечує зв'язок між різними частинами системи та інтеграцію з системами зберігання енергії.

7. Система зберігання енергії (Energy Storage): Блок батарей, який може зберігати електроенергію для забезпечення стабільної роботи зарядних пристроїв.

8. EKI-7706G-2FI Industrial Managed Switch: Ще один мережевий комутатор, що підключений до системи зберігання енергії, забезпечує ефективний розподіл енергії.

9. UNO-1372G-J Charger HMI Controller: Контролер для управління людським інтерфейсом зарядних пристроїв, що дозволяє операторам легко маніпулювати налаштуваннями та статусом зарядки.

10. Електрозарядний пристрій (EV Charger): Основні пристрої, які використовуються для підзарядки електромобілів, забезпечуючи передачу енергії від системи зберігання до автомобілів.

Лінії на схемі, позначені різними кольорами, представляють різні типи зв'язків і живлення: змінний (Electricity AC) та постійний (Electricity DC) струм, мережевий зв'язок (Ethernet) та оптичні волокна (Fiber optics), що використовуються для високошвидкісного передавання даних [18].

Загалом, ця блок-схема показує інтегровану систему зі складними зв'язками між різними технологічними компонентами, кожен з яких відіграє важливу роль у забезпеченні надійної та ефективної роботи зарядної станції для електромобілів [19].

Також, хотілось би оглянути компонування самої електрозарядної станції. На рисунку 1.7, представлено елементи системи для електрозарядної станції з використанням сучасних технологій, таких як хмарні обчислення та зв'язок 5G.

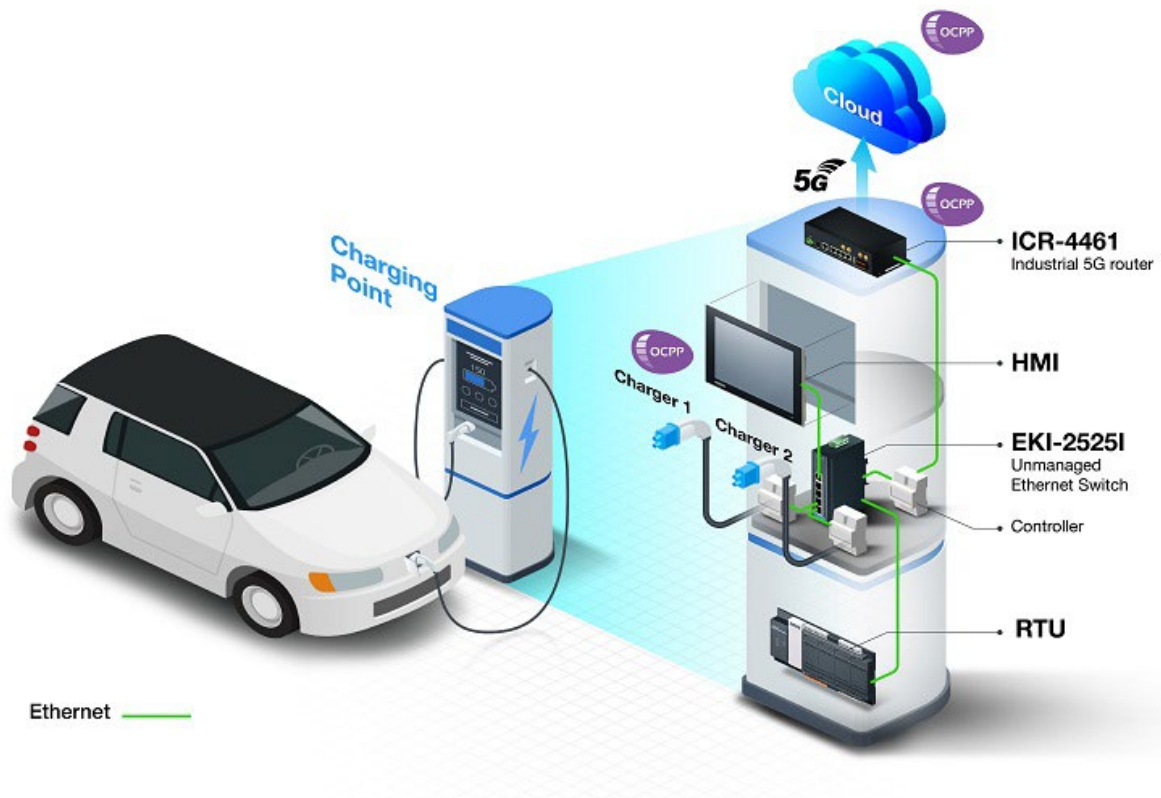


Рисунок 1.7 – Блок-схема архітектури зарядної станції мережі електрозарядних станцій Advantech

Наводимо детальний опис кожного компонента на цьому зображенні:

1. **Charging Point:** Місце, де електромобіль підключається для зарядки.
2. **Ethernet:** Це зображення підключення Ethernet, яке індикують зеленою лінією, можливо, вказуючи на мережеве з'єднання між компонентами системи.
3. **Cloud with 5G Symbol:** Хмарна інфраструктура з підключенням через мережу 5G, що вказує на високошвидкісний доступ до даних і управління системою в реальному часі.
4. **ICR-4461 Industrial 5G router:** Промисловий маршрутизатор 5G, що забезпечує підключення компонентів зарядної станції до хмарних сервісів через мережу 5G. Це серце зв'язкової інфраструктури, який забезпечує швидкий та надійний доступ до хмарних сервісів через 5G мережу. Він приймає дані з контролера та HMI, а також з RTU, і передає цю інформацію до хмари для обробки та зберігання.

5. НМІ (Human-Machine Interface): Інтерфейс людина-машина, який використовується для відображення інформації про статус зарядних пристроїв та взаємодії операторів зі станцією. Дозволяє операторам станції візуалізувати та контролювати процеси зарядки, включаючи статуси зарядних станцій, поточні навантаження, інформацію про помилки та інші важливі показники. НМІ взаємодіє з контролером для впровадження команд користувача.

6. ЕКІ-2525І Unmanaged Ethernet Switch: Некерований Ethernet-комутатор, що забезпечує мережеві з'єднання між зарядними пристроями та іншими компонентами системи. Підключається до НМІ, RTU, та інших компонентів через Ethernet для передачі даних у мережі без необхідності додаткового управління. Це забезпечує просту взаємодію між пристроями та їхнє з'єднання з маршрутизатором 5G.

7. Controller: Контролер системи, який управляє роботою зарядних пристроїв, процесами моніторингу та іншими автоматизованими задачами. Це обчислювальний мозок системи, який управляє різними зарядними пристроями, виконує алгоритми зарядки, моніторингу та діагностики. Він також обробляє дані з RTU та відправляє команди до зарядних пристроїв, регулюючи зарядні процеси.

8. RTU (Remote Terminal Unit): Пристрій дистанційного термінального управління, який служить для збору даних з зарядних пристроїв та передачі їх в хмару для обробки та аналізу.

На зображенні також присутні два зарядні пристрої (означені як "Charger 1" та "Charger 2"), які символізують різні точки зарядки, доступні для електромобілів. Кожен зарядний пристрій має індикатори стану, підключені до системи управління та моніторингу [20].

Загалом, це зображення показує, як інтегровані технології працюють разом для створення сучасної, ефективної та інтелектуальної інфраструктури для електрозарядних станцій, здатної управляти зарядними процесами та забезпечувати високий рівень обслуговування та безпеки.

Проект підкреслює важливість стійкості обладнання до різних погодних умов та екстремальних температур. Станції були обладнані для витримування умов з

високим рівнем пилу, опадів, тривалого сонячного світла та екстремальних температур.

Використання протоколу ОСРР забезпечувало стабільний та надійний обмін даними, а також інтеграцію з системами управління на задньому плані, віддаленим моніторингом станцій та системами управління енергією. Важливою складовою системи було використання мережевого протоколу TCP/IP для моніторингу в реальному часі, що суттєво підвищувало ефективність управління.

Загалом, проект Advantech демонструє важливість інтеграції передових технологій у сфері управління електрзарядними станціями та відіграє ключову роль у розвитку інфраструктури електромобілей.

В результаті огляду реалізації проекту Advantech, було прийнято рішення що проект потребує багато апаратних рішень для мережі зарядної станції, а також залежить від багатьох зовнішніх факторів та сам метод ОСРР не є універсальним протоколом під кожний тип інтерфейсу зарядки, тому потрібно розглянути або ставити свій метод автоматизації інформаційної системи мережі електрзарядних станцій.

1.5 Висновки

У цьому розділі було проведено аналіз проблематики предметної області, а саме проблеми розвитку інфраструктури електромобілей. Розглянуто основні можливості, технології, переваги і недоліки існуючих інформаційних систем для автоматизації роботи користувачів з електрзарядними станціями, прийнято рішень щодо винаходу нового методу автоматизації.

2 ВИБІР ОПТИМАЛЬНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

2.1 Аналіз можливостей мови програмування JavaScript

JavaScript відомий як основна мова скриптів для веб-браузерів, що є ключовою для розробки сучасних веб-застосунків. Розробники активно використовують цю мову для створення складних додатків, однак під час розробки доступні не так багато інструментів підтримки.

JavaScript було оголошено компанією Netscape у 1995 році як "легку для використання мову об'єктних сценаріїв, розроблену для реалізації енергійних онлайн-додатків, які інтегрують об'єкти та ресурси як на клієнтській, так і на серверній стороні" [21].

З тих пір JavaScript став нормою для створення сценаріїв на стороні клієнта у веб-браузерах, проте багато інших застосунків також інтегрують JavaScript двигун. Його універсальність змусила розробників створювати об'ємні програми мовою, що була спочатку призначена для скриптів, а не для масштабного програмування [22].

Особливості JavaScript включають, зокрема, його популярність, як показано в опитуванні Stack Overflow, та його здатність адаптуватися завдяки розвитку технологій браузерів та завдяки переходу на сервер з використанням Node.js та інших фреймворків. З JavaScript можна робити наступне:

- Він був розроблений для маніпуляцій з DOM, дозволяючи веб-сайтам стати динамічними.
- Функції в JS є об'єктами, мають властивості та методи і можуть передаватися як аргументи.
- Може обробляти дату та час.
- Може валідувати форми, які створені за допомогою HTML.
- Не вимагає компілятора.

Області застосування JavaScript:

- Веб-розробка: JavaScript виник як засіб для додавання інтерактивності до статичних веб-сайтів. З AngularJS цей процес стає ще простішим.

- Веб-додатки: Сучасні браузеры дозволяють створювати високоякісні веб-додатки, і JavaScript відіграє важливу роль у цьому. Наприклад, детальні картографічні додатки, такі як Google Maps, підтримуються JavaScript та API, що доповнюють його потужність. Технології як Electron та React теж є корисними у цьому контексті [23].
- Серверні застосунки: З допомогою Node.js JavaScript може функціонувати на серверній стороні, розширюючи свої можливості.
- Машинне навчання: Бібліотека JavaScript ml5.js може бути використана для інтеграції машинного навчання в веб-розробці.
- Мобільні застосунки: JavaScript також дозволяє створювати мобільні додатки, використовуючи технології як React Native для розробки під різні операційні системи [24].

Із-за універсальності та інтуїтивно простої мови для програмування JavaScript, було обрано для реалізації методу автоматизації інформаційної системи мережі електрозарядних пристроїв. Так-як мова підходить для розробки як клієнтської так і серверно-апаратної частини проекту [25].

2.2 Аналіз фреймворку для розробки мобільних додатків React Native

React Native - це фреймворк, розроблений командою інженерів з Facebook, і базується він на ReactJS. Він дозволяє написати код лише один раз, використовуючи популярну мову програмування JavaScript або TypeScript, і створити мобільний додаток як для платформи Android, так і для iOS [26].

Компанія Facebook активно використовує React Native у своїх продуктах, включаючи мобільний додаток Facebook (наприклад, функціональність Marketplace була розроблена з використанням React Native) і Instagram. Тому ви можете бути впевнені в якості фреймворку та майбутній підтримці з боку Facebook[27].

Однією з основних проблем для розробників є потреба створювати різні версії додатків для різних мобільних платформ. Однак можливим рішенням цієї проблеми є створення гібридного додатку (див. рисунок 2.1). Гібридний додаток ґрунтується

на веб-додатку, який можна створити за допомогою будь-яких front-end бібліотек, таких як ReactJS, Angular або JQuery [28].

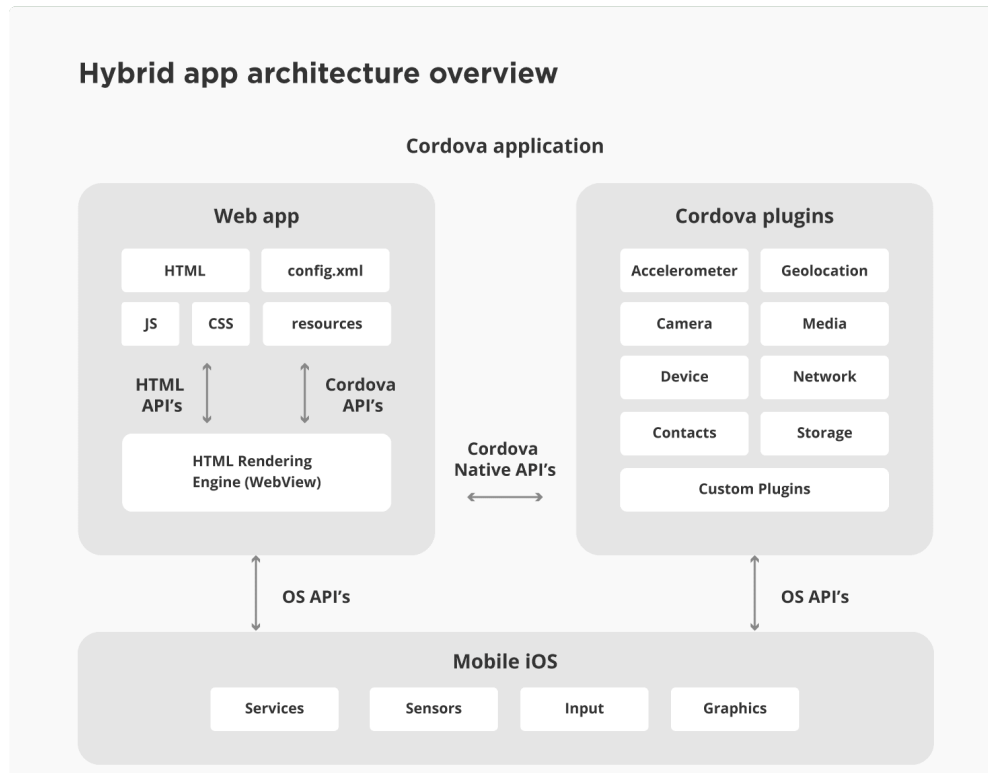


Рисунок 2.1 – Блок-схема архітектури гібридного додатку

Наступним етапом є створення мобільного додатку за допомогою фреймворку Apache Cordova, який включає в себе готовий додаток у WebView (це нативний компонент на платформах iOS і Android, який відображає веб-сайти). Cordova також надає можливість гібридним програмам використовувати різні нативні функції, такі як Bluetooth, GPS і роботу з акумулятором.

Цей підхід дозволяє створити єдину програму, яка працює на різних платформах, і використовується один і той же код, без необхідності писати рідний код для кожної платформи. Більше того, такий додаток може використовуватися і як веб-додаток [29].

Проте гібридний підхід має свої недоліки:

- Потрібно більше часу на створення досвіду користувача, сумісного з рідними додатками, що іноді може вимагати імітації рідних вражень.

- Продуктивність програми залежить від движка веб-рендерингу, що може стати обмежуючим фактором.
- Гібридні додатки мають потенційні проблеми, що властиві веб-додаткам, через різні движки рендерингу на кожній платформі.

Останнім часом з'явилися кроссплатформні технології, які дозволяють писати код однією мовою та використовувати компоненти, надані різними технологіями, для створення додатків на різних платформах (див. рисунок 2.2). Це дозволяє писати код один раз і використовувати його для створення додатків на бажаних платформах. Хоча конкретні фреймворки можуть відрізнятися за деталями реалізації, вони використовують рідний код для кожної платформи.

Завдяки цим кроссплатформні рішенням додатки можуть досягати значно вищої продуктивності, подібно до нативних додатків. Вони також забезпечують користувачам досвід, який аналогічний або схожий на той, що отримують у рідних додатках для конкретних платформ

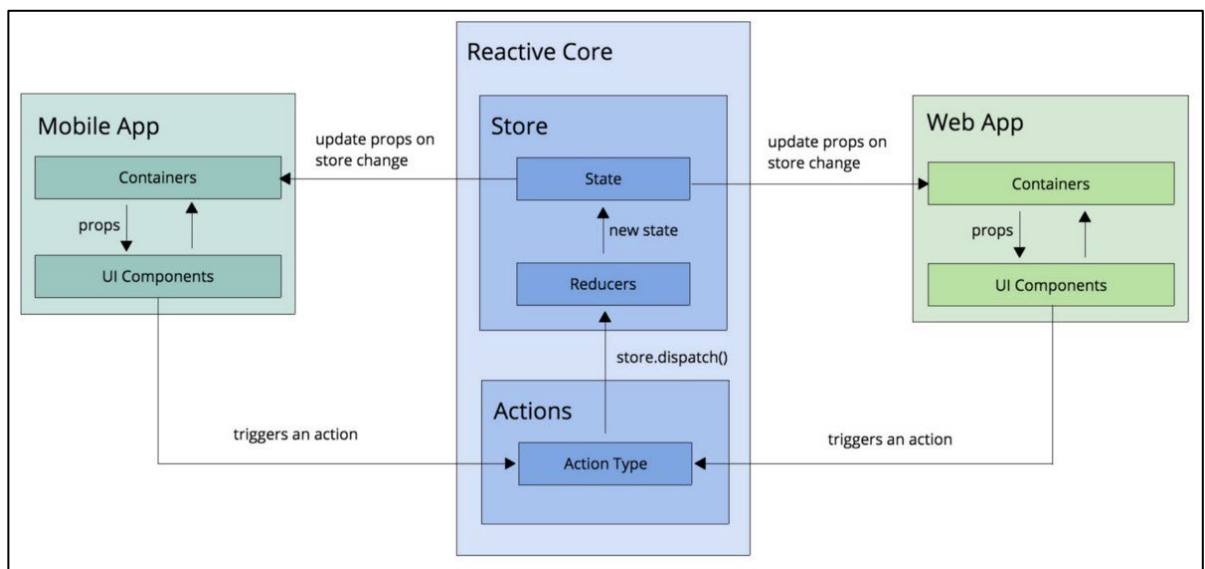


Рисунок 2.2 – Блок-схема архітектури кроссплатформного додатку React Native

React Native має низку значущих переваг:

1. JavaScript: React Native базується на JavaScript, одній з найпопулярніших мов програмування на сьогодні. Це відкриває доступ до широкого спектру зовнішніх бібліотек, які можна використовувати в React Native, такі як `day.js` і `lodash`. Більше того, багато бібліотек, спроектованих для ReactJS, можуть бути використані і з React Native, такі як `Formik`, `redux`, `victory.js` та `AWS Amplify`.

2. Інтеграція з існуючими додатками: React Native дозволяє інтегрувати його з існуючими мобільними додатками, що дозволяє поступово розширювати функціональність додатку за допомогою React Native. Це особливо корисно на етапі проектування, коли можна визначити, які частини додатку варто реалізувати за допомогою React Native і які краще реалізувати на рідній мові програмування для оптимальної продуктивності.

3. "Learn once, write everywhere": Розробники, які вже знайомі з ReactJS, можуть швидко перейти до розробки на React Native. Знання та досвід, набуті при роботі з веб-версією, можна перенести на React Native майже без змін.

4. Крос-платформеність: React Native дозволяє писати код, який може бути використаний на різних платформах. Також, для вирішення платформозалежних завдань, можна написати фрагменти коду, які будуть виконуватися тільки на певній платформі.

5. Нативна інтеграція: Можливість взаємодії з нативним кодом платформи, що дозволяє використовувати нативні API та інтегруватися з існуючими рішеннями.

6. Розширюваність: Архітектура React Native орієнтована на можливість додавання підтримки для нових платформ. Це дозволяє розширювати його функціональність та використовувати його для розробки не тільки мобільних, але і веб-додатків.

7. Expo: Не можна не згадати про фреймворк Expo, який спрощує розробку на React Native, надаючи багато корисних компонентів та інструментів для створення стандартних мобільних додатків і підтримує різні пристрої та функції, такі як камера, геолокація і вхід через соціальні мережі.

React Native володіє низкою переваг, що робить його привабливим вибором для розробки мобільних додатків на різних платформах. Тому його обрано для реалізації мобільного додатку для нашого проекту, тому як він нативно розробляється одразу під дві платформи Android та iOS. Також додаток на цій платформі дуже просто інтегрується з серверною частиною, тому що також використовує мову програмування Java Script [30].

2.3 Аналіз можливостей програмної платформи Node JS

Node.js - це кросплатформове середовище виконання з відкритим вихідним кодом, яке призначене для виконання коду JavaScript поза контекстом веб-браузера. Важливо пам'ятати, що Node.js не є ні фреймворком, ні мовою програмування. Це виключно середовище виконання.

Node.js широко використовується для створення різних веб-додатків, мобільних додатків та внутрішніх послуг, таких як API. Великі компанії, такі як PayPal, Uber, Netflix і Walmart, активно використовують Node.js у своєму виробництві [31]. Його потужність полягає в здатності обробляти багато одночасних запитів, що робить його ідеальним для створення швидких та ефективних веб-додатків.

Node.js має декілька особливостей, які роблять його популярним і корисним для розробки back-end сервісів:

- Реалізація на JavaScript: Однією з основних особливостей Node.js є використання JavaScript як мови програмування для створення back-end. Це дозволяє розробникам використовувати одну мову (JavaScript) як для клієнтської, так і для серверної частини додатків, що спрощує розробку та зменшує витрати часу на вивчення різних мов програмування.

- Простота використання: Node.js дозволяє легко створювати прототипи та розвивати програми, завдяки швидкій інсталяції та простому використанню.

- Висока продуктивність і масштабованість: Через асинхронну та неблокуючу архітектуру Node.js може обробляти багато одночасних запитів, що робить його відмінним вибором для створення швидких і високопродуктивних серверних додатків[32].

- Чистий та послідовний код: Через асинхронну природу Node.js, код може бути написаний у вигляді послідовних операцій, що робить його більш зрозумілим і легким для обслуговування.

- Велика екосистема бібліотек: Node.js має широку екосистему бібліотек з відкритим вихідним кодом, які допомагають розробникам швидше створювати функціональність своїх додатків без необхідності вдаватися до написання коду з нуля.

- Асинхронний або неблокуючий характер: Node.js використовує асинхронну модель виконання, що дозволяє обробляти запити без блокування інших операцій. Це робить його ідеальним для обробки багатьох одночасних запитів та роботи з вводом/виводом.

Використання Node.js має безліч переваг:

- Проста масштабованість: Node.js легко масштабувати, як у горизонтальному (за допомогою додавання нових серверів), так і у вертикальному напрямках (за допомогою збільшення ресурсів на окремому сервері). Це дає можливість створювати високопродуктивні програми та забезпечувати їх зростання під збільшенням навантаження.

- Веб-програми в реальному часі: Node.js ідеально підходить для створення веб-додатків в реальному часі, таких як чати і ігри, завдяки швидкій синхронізації та асинхронній обробці подій. Це дозволяє обробляти багато одночасних підключень без зайвого навантаження HTTP.

- Швидкий набір: Node.js працює на движку V8, розробленому Google, що робить його дуже швидким у виконанні коду JavaScript. Асинхронна природа Node.js дозволяє ефективно обробляти ввід/вивід операції, такі як робота з базою даних і мережевими з'єднаннями.

- Простота вивчення і кодування: Завдяки використанню JavaScript, Node.js легкий у вивченні та використанні, особливо для розробників front-end, які вже мають досвід з цією мовою.
- Хешування модулів: Node.js дозволяє хешувати модулі у пам'яті, що допомагає уникнути повторного виконання коду та підвищує продуктивність.
- Потокова передача даних: Node.js дозволяє обробляти HTTP-запити та відповіді як потоки даних, що робить обробку великих файлів та мультимедійних даних ефективною та швидкою.
- Зручний хостинг: Існують хостингові платформи, такі як Heroku і PaaS (Platform as a Service), які спеціалізуються на розгортанні програм Node.js, що робить розміщення додатків простим та зручним.
- Корпоративна підтримка: Багато великих компаній, таких як Walmart, PayPal, Microsoft і Yahoo, використовують Node.js для своїх програм, оскільки він об'єднує front-end та back-end розробку під єдиним капотом і дозволяє використовувати знання JavaScript для різних аспектів розробки.

Node.js став популярним рішенням для створення ефективних та швидких серверних додатків завдяки своїм перевагам та зручності в розробці [33].

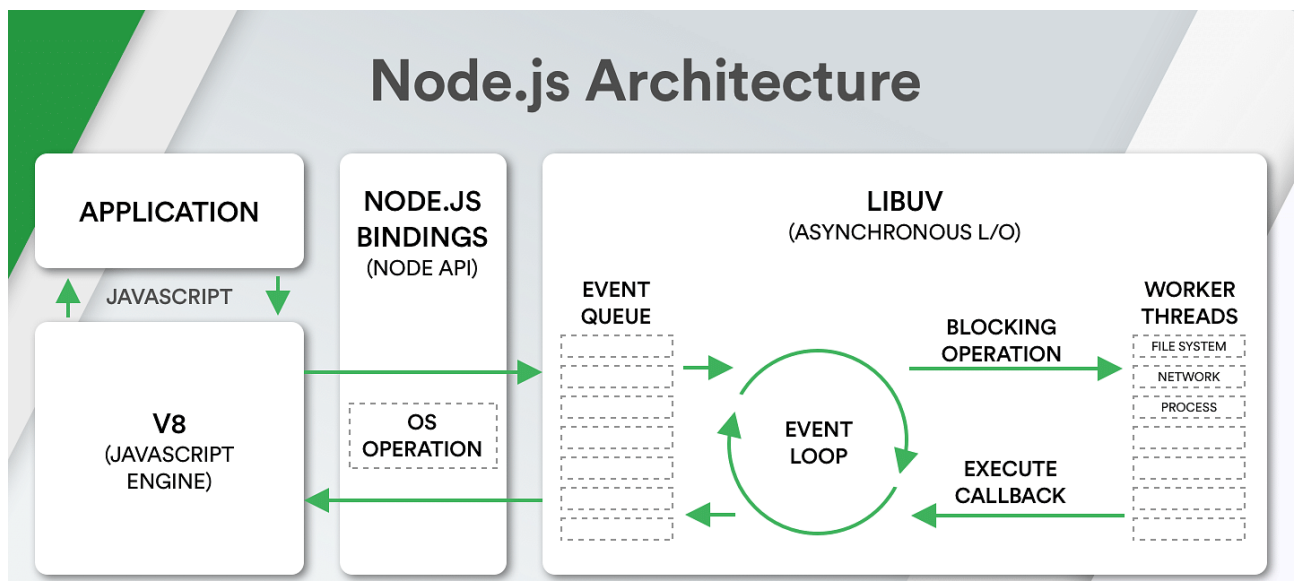


Рисунок 2.3 – Блок-схема архітектури кросплатформерного додатку Node JS

Тому було обрано використовувати NodeJS як платформу для серверної частини, його особливість в обробці поточкових даних є дуже актуальним для такого виду проекту з постійним трафіком з різних приладів, а також ця платформа дає велику зручність в налаштуванні API-запитів та роботою з нею [34].

2.4 Аналіз можливостей розумних маршрутизаторів MikroTik

MikroTik - латвійська компанія, спеціалізація якої - розробка та виробництво мережевого обладнання. Продуктовий асортимент MikroTik включає в себе провідні та безпроводні маршрутизатори, мережеві комутатори, а також точки доступу, відомі як RouterBOARD. Це обладнання розроблено для різних застосувань та відноситься до напівпрофесійного сегменту [35].

Порівнюючи MikroTik з домашніми маршрутизаторами від таких виробників, як D-Link, TP-Link, Asus, відзначається велика кількість функцій та можливостей виробів MikroTik. Це робить їх привабливими для малого та середнього бізнесу, а також деяких великих мережевих провайдерів.

Незважаючи на високий функціонал, MikroTik пропонує свої продукти за доступними цінами, що робить їх конкурентоздатними. Проте, варто враховувати, що порівнювати MikroTik з професійним обладнанням Cisco чи Juniper було б некоректно, оскільки ці виробники належать до абсолютно різних класів.

З точки зору безпеки та надійності програмного забезпечення та апаратної частини, MikroTik, безумовно, має переваги над домашніми роутерами, проте в порівнянні з Cisco та Juniper вони можуть бути менш надійними для організації великих та складних мереж.

Отже, MikroTik - це оптимальний вибір для багатьох компаній, особливо малих та середніх, які цінують баланс між функціональністю та доступністю в мережевому обладнанні [36].

Порівнюючи обладнання MikroTik та Cisco, слід враховувати, що ці два виробника належать до різних класів. Обидва компанії вкладають багато

можливостей у своє обладнання, забезпечуючи широкий функціонал, такий як VPN, налаштування міжмережових екранів, безшовний роумінг та гнучкість налаштувань.

Однак, Cisco, безперечно, відома своєю високою надійністю та цілодобовою підтримкою, якої MikroTik не має. Незважаючи на це, MikroTik приваблює малий і середній бізнес завдяки своїй більш доступній ціні. Прикладом може бути порівняння бюджетних маршрутизаторів: MikroTik RB2011UiAS-IN коштує 2500 грн, тоді як аналогічне обладнання Cisco Catalyst 2960C вартує 11000 грн, що в 4,5 рази дорожче, при практично однакових характеристиках.

Малі та середні компанії часто вибирають MikroTik через більш доступну ціну, особливо враховуючи, що потрібно закупити не один обладнання для комп'ютерної мережі. Проте, важливо враховувати, що MikroTik може бути менш надійним у великих та складних мережах [37].

Крім того, виробник Ubiquiti також конкурує з MikroTik, зокрема в галузі Wi-Fi. Точки доступу Ubiquiti зазвичай мають потужні радіомодулі і якісно працюють у безшовному роумінгу завдяки технології MIMO. Якщо потрібно забезпечити стабільне покриття Wi-Fi, то точки доступу Ubiquiti можуть бути кращим вибором.

З іншого боку, якщо важливо побудувати безпечну та відмовостійку мережу, MikroTik має більш гнучкі налаштування та можливості, що дозволяють забезпечити захист від атак.

MikroTik RouterOS - це спеціалізована операційна система, створена для розробки багатофункціональних маршрутизаторів, фаєрволів, бриджів, базових станцій, VPN-серверів, веб-серверів та інших мережових пристроїв. Вона може працювати як на апаратних платформах MikroTik RouterBoard, які зазвичай мають процесори PowerPC і Atheros, так і на обладнанні, заснованому на архітектурі x86, таких як звичайні персональні комп'ютери.

Операційна система MikroTik RouterOS підтримує наступні характеристики та обладнання:

- Архітектура i386.
- Підтримка SMP (Symmetric Multiprocessing) - можливість використовувати декілька ядер та процесорів.
- Мінімальна кількість оперативної пам'яті 32 Мб (максимальна - 2 Гб).
- Підтримка накопичувачів IDE, SATA, USB обсягом від 64 Мб.
- Мережеві карти, підтримувані ядром Linux v2.6 kernel (PCI, PCI-X).

Важливо відзначити, що хоча операційна система MikroTik RouterOS базується на ядрі Linux, вона не є відкритим проектом OpenSource, і її внутрішній код не доступний для загальної громадськості. Отже, доступ до командного рядка Linux є обмеженим.

Встановити RouterOS можна різними способами: на персональний комп'ютер за допомогою установочного CD-диска або ISO-файлу, який доступний на сайті MikroTik, або через мережу за допомогою утиліти Netinstall, також доступної на сайті виробника.

Після встановлення MikroTik RouterOS можна налаштовувати та керувати різними способами, включаючи:

- MAC-адресне обмеження доступу до пристрою.
- Використання WinBox - утиліти для Windows OS.
- Використання веб-інтерфейсу через WebBox.
- Використання розширеного веб-інтерфейсу Webfig для конфігурації.
- Взаємодія через командний рядок (консоль) з підтримкою скриптів та робота по протоколам Telnet і SSH.
- Використання API для створення власних додатків для налаштування та моніторингу мережі.

MikroTik RouterOS також підтримує збереження, відновлення та імпорт всіх налаштувань конфігурації в різних форматах, включаючи бінарний файл та текстовий формат, зрозумілий для користувача.

Загалом, MikroTik RouterOS є потужним інструментом для створення та управління мережами, який має велику кількість функцій для роботи з різними мережевими протоколами і може бути використаний на різних обладнаннях, тому для нашого проекту був обраний сам маршрутизатор цієї фірми за можливість налаштування власного конфігу з особливістю постійного, захищеного трафіку [38].

2.5 Аналіз можливостей ethernet-контролерів VK Module

В сучасному світі інформаційних технологій, де автоматизація та віддалене керування стали нормою, модулі управління та моніторингу по мережі стали ключовими компонентами для забезпечення ефективності та надійності різних систем та процесів. Однією з важливих компаній, яка пропонує такі рішення, є "VK Module", а їх контролер "Socket 2W" стає символом нового підходу до автоматизації [39].

Компанія "VK Module" впроваджує інновації у сферу індустріальної автоматизації та контролю. Їх продукти спрощують життя інженерам і менеджерам, дозволяючи віддалено керувати та моніторити різні процеси у реальному часі. Один з їхніх контролерів, "Socket 2W", створений з метою надати користувачам зручний інструмент для відстеження стану замкнуто/розімкнуто на дискретних входах, вимірювання напруги на аналоговому вході та керування реле, а все це через мережу Ethernet [40].

Функціональність контролера "Socket 2W"

Цей контролер був розроблений для віддаленого керування різними системами, пристроями та інтеграцією з серверно-клієнтської частиною через мережу Ethernet. Він має вбудовані функції та дозволяю налаштувати власні сценарії, конфігурацію роботи, якщо їх можливо виконати апаратною частиною:

- Відстеження та передача стану замкнуто/розімкнуто на двох дискретних входах. Цей контролер має можливість моніторингу стану двох дискретних входів,

що може бути корисним для відслідковування стану різних сенсорів або пристроїв, які мають два можливих стани (замкнено або розімкнено).

- Вимірювання напруги в діапазоні від 0 до 1,0 Вольта на одному аналоговому вході. Ця можливість дозволяє контролеру вимірювати аналоговий сигнал в діапазоні від 0 до 1,0 Вольта. Це може бути використано для моніторингу аналогових датчиків або пристроїв, які генерують аналоговий сигнал.

- Керування двома реле 240В 10А. Контролер може керувати двома реле, які підходять для вимикання або увімкнення пристроїв з напругою 240 Вольт і струмом до 10 Ампер.

- Підключення до локальної мережі за технологією WiFi. Цей контролер може підключатися до локальної мережі через Wi-Fi, що дозволяє бездротово здійснювати комунікацію та керування ним через мережу.

- Налаштування виконується Web-браузером. Контролер може бути налаштований за допомогою веб-браузера, що робить процес конфігурації зручним та доступним.

- Скидання контролера у заводські налаштування. Існує можливість скидання контролера до заводських налаштувань, що корисно у випадку, коли потрібно відновити початкові налаштування.

- Управління модулем та моніторинг стану входів через різні протоколи. Контролер підтримує різні протоколи для управління та моніторингу, такі як TCP, UDP, HTTP і ICMP. Це дозволяє вам взаємодіяти з контролером з використанням різних програмних інтерфейсів та протоколів залежно від вашої потреби.

Загалом, контролер "КонтролерSocket-2W" має широкий спектр можливостей і може бути корисним для великої кількості застосувань, включаючи відстеження, вимірювання, керування та моніторинг різних пристроїв та процесів у мережі Ethernet з використанням Wi-Fi [41].

Контролери управління та моніторингу по мережі, такі як "Socket 2W" від компанії "VK Module", є важливими складовими в сучасному світі автоматизації та віддаленого керування. Вони допомагають підвищити продуктивність, надійність та

зручність управління різними системами та пристроями. Інновації в цій галузі продовжують змінювати спосіб, яким ми контролюємо та моніторимо процеси, і компанія "VK Module" відіграє важливу роль у цьому процесі, надаючи нам потужні та зручні інструменти [42].

Даний контролер був обраний для розробку рішення щодо методу автоматизації роботи інформаційної системи із-за універсальності налаштування з любым приладом який працює з інтернетом або просто керується струмом 12В або має свій протокол роботи, завдяки цьому модулю проект буде мати можливість розробити універсальне рішення управління, будь яким зарядним пристроєм.

2.6 Висновки

У цьому розділі було проведено аналіз різних аспектів важливих технологій та платформ, які є ключовими для предметної області дослідження. Були розглянуті такі питання, як можливості мови програмування JavaScript, фреймворку React Native для мобільних додатків, програмної платформи Node.js, розумних маршрутизаторів MikroTik і ethernet-контролерів VK Module.

Цей аналіз надав можливість краще зрозуміти переваги і недоліки цих технологій і платформ, а також їхню застосовність у відповідних сферах. Результати аналізу вказують на те, що вибір конкретної технології або платформи може бути залежним від конкретних вимог і завдань проекту, і вони можуть бути корисними для подальшого розвитку і дослідження у галузі електрозарядних пристроїв.

3 РОЗРОБЛЕННЯ МЕТОДУ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ОБСЛУГОВУВАННЯ КОРИСТУВАЧІВ МЕРЕЖІ ECOSTATION

3.1 Розроблення архітектури інформаційної системи

Сучасний світ характеризується стрімким розвитком технологій та їх впливом на всі сфери людської діяльності. Особливо це стосується сфери транспорту, де активно розвиваються електромобілі та інфраструктура зарядних станцій. Мережа ECOSTATION є однією з ключових елементів цієї інфраструктури, що забезпечує зарядку електромобілів. Щоб відповідати зростаючим вимогам користувачів та ефективно управляти ресурсами, необхідно розробити ефективний метод автоматизації процесів обслуговування.

Однією з ключових складових успішного впровадження такого методу є розроблення архітектури інформаційної системи. Цей процес вимагає глибокого аналізу потреб користувачів, а також особливостей функціонування зарядних станцій. Архітектура інформаційної системи повинна бути гнучкою, масштабованою та забезпечувати високий рівень безпеки та надійності. Вона має інтегрувати різні технічні компоненти та програмні рішення, створюючи єдиний механізм для ефективного управління процесами зарядки.

У цьому розділі ми зосередимося на ключових етапах розробки архітектури інформаційної системи. Наступним кроком буде представлення детальної блок-схеми, яка ілюструє структуру та взаємозв'язки між різними компонентами системи. Ця схема допоможе візуалізувати основні елементи системи, включаючи модулі управління даними, інтерфейси користувача та інтеграцію з зовнішніми сервісами. Опис кожного компонента та його функціоналу надасть глибше розуміння того, як архітектура сприятиме ефективності процесів обслуговування користувачів мережі ECOSTATION. Ця блок-схема стане основою для подальшого аналізу та дискусії щодо реалізації запропонованого методу автоматизації.

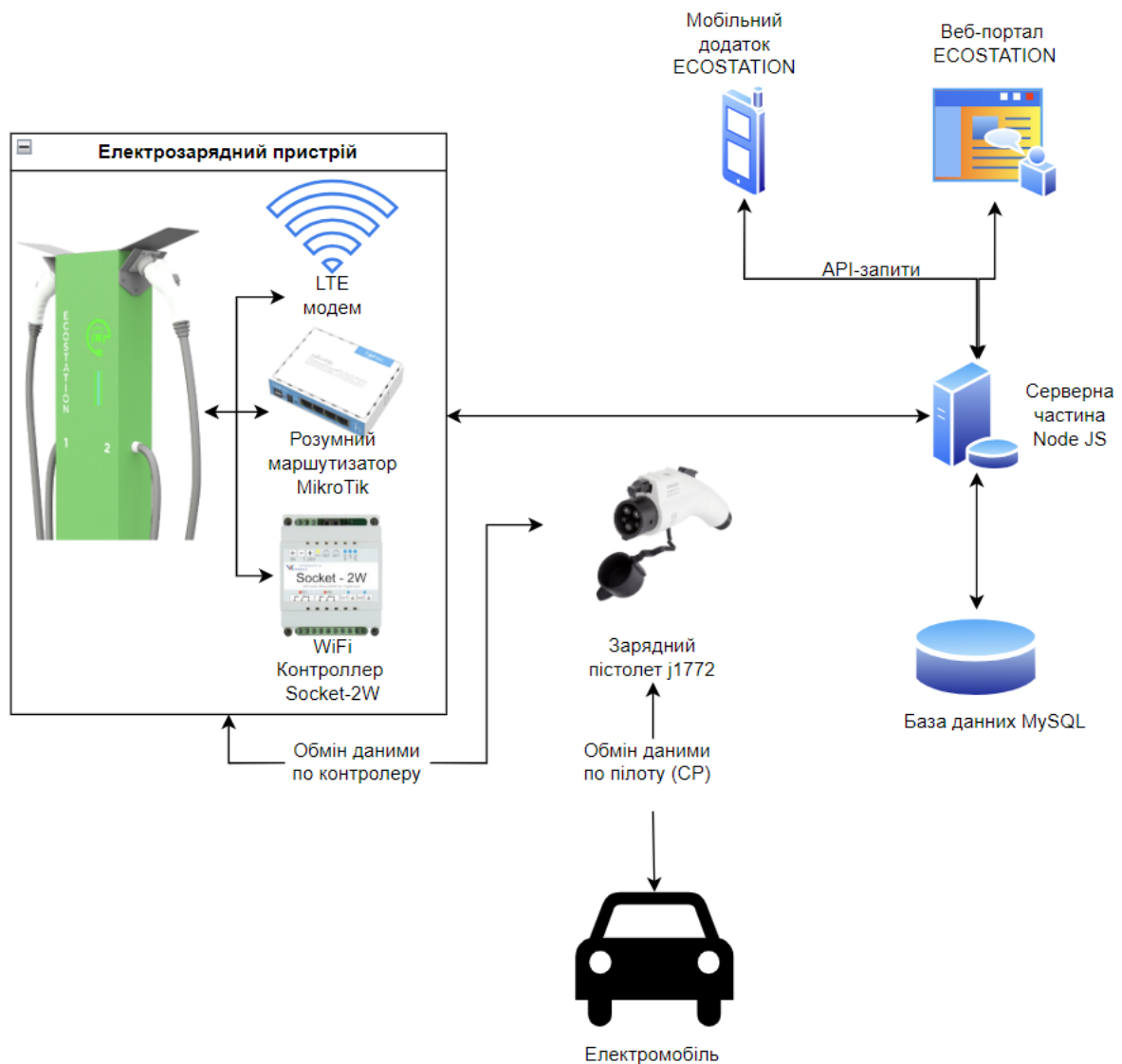


Рисунок 3.1 – Блок-схема архітектури інформаційної системи

На представленій блок-схемі відображено архітектуру інформаційної системи для процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION. Система складається з декількох ключових компонентів, які взаємодіють між собою:

- Електрозарядний пристрій: Оснащений LTE модемом для підключення до мережі та маршрутизатор MikroTik для маршрутизації даних між серверною частиною та апаратної частини електрозарядок. Також пристрій містить WiFi

контролер Socket-2W, який дозволяє керувати процесами зарядки та взаємодіяти з автомобілем через інтерфейс зарядки.

- Зарядний пістолет J1772: Використовується для фізичного підключення до електромобіля та передачі даних про процес зарядки (CP - Control Pilot).
- Електромобіль: Транспортний засіб, який підключається до зарядного пристрою для отримання заряду.
- Серверна частина Node JS: Центральний елемент обробки даних, який керує логікою взаємодії між компонентами системи, базою-даних та апаратною частиною самого зарядного пристрою.
- База даних MySQL: Сховище даних, яке зберігає інформацію про процеси зарядки, користувачів, транзакції та інші важливі дані.
- Мобільний додаток ECOSTATION: Дозволяє користувачам взаємодіяти з системою, включаючи управління процесами зарядки, перегляд історії зарядок та інші функції.
- Веб-портал ECOSTATION: Інтерфейс, через який користувачі та адміністратори системи можуть отримувати доступ до сервісів ECOSTATION через браузер. Включає функції, аналогічні мобільному додатку та систему адміністрування системи електростанцій.
- API-запити: Механізм взаємодії між мобільним додатком, веб-порталом, серверною частиною, апаратною частиною електростанцій, що дозволяє реалізувати запити користувачів та обмін даними в реальному часі.

Ця архітектура демонструє інтеграцію апаратних та програмних компонентів, які разом створюють повноцінну систему для ефективного управління процесами зарядки в мережі ECOSTATION. Кожен компонент відіграє важливу роль у забезпеченні безперебійної роботи системи та високого рівня обслуговування користувачів.

Програмне забезпечення проекту "Автоматизація процесів обслуговування користувачів мережі ECOSTATION" складається з декількох ключових компонентів, що детальніше проілюстровано на рисунці 3.2, всі компоненти налаштовані під

кожний модуль системи, щоб спільно функціонувати для надання ідеального результату:

1. Серверна частина (Backend) на Node.js: Ця частина є основою для обробки запитів, взаємодії з базою даних та інтеграції з іншими компонентами системи. Вона обробляє бізнес-логіку, авторизацію користувачів, управління сесіями зарядки та інші задачі, пов'язані з управлінням даними.

2. База даних MySQL: Зберігає всі необхідні дані, такі як інформація про користувачів, зарядні сесії, транзакції, стан зарядних станцій тощо. Забезпечує швидкий доступ та високу надійність зберігання даних.

3. Мобільний додаток ECOSTATION: Дозволяє користувачам віддалено управляти процесом зарядки, переглядати історію своїх зарядок, отримувати сповіщення про статус зарядки та доступність зарядних станцій. Додаток може включати функціонал геолокації для виявлення найближчих станцій та навігації до них, а також забезпечує інтеграцію з платіжними системами для здійснення оплат.

4. Веб-портал ECOSTATION: Веб-версія мобільного додатку, яка дозволяє користувачам через браузер отримувати ті самі функції, що й у мобільному додатку. Це включає управління зарядними сесіями, перегляд історії транзакцій та інформації про станції, а також керування особистими налаштуваннями профілю.

5. API для обміну даними: Інтерфейс програмування додатків, який дозволяє мобільному додатку та веб-порталу спілкуватися з серверною частиною, передаючи запити користувачів та отримуючи відповіді. API забезпечує безпеку та ефективність обміну даними.

6. Зарядний модуль: розробка апаратного оснащення зарядної станції та оснащення його програмним забезпеченням який дозволив би виконувати функції зарядки, обміну даних та постійного моніторингу стану зарядного пристрою та автомобіля, а саме метод комунікації та автоматизації роботи користувачів з зарядною станцією.

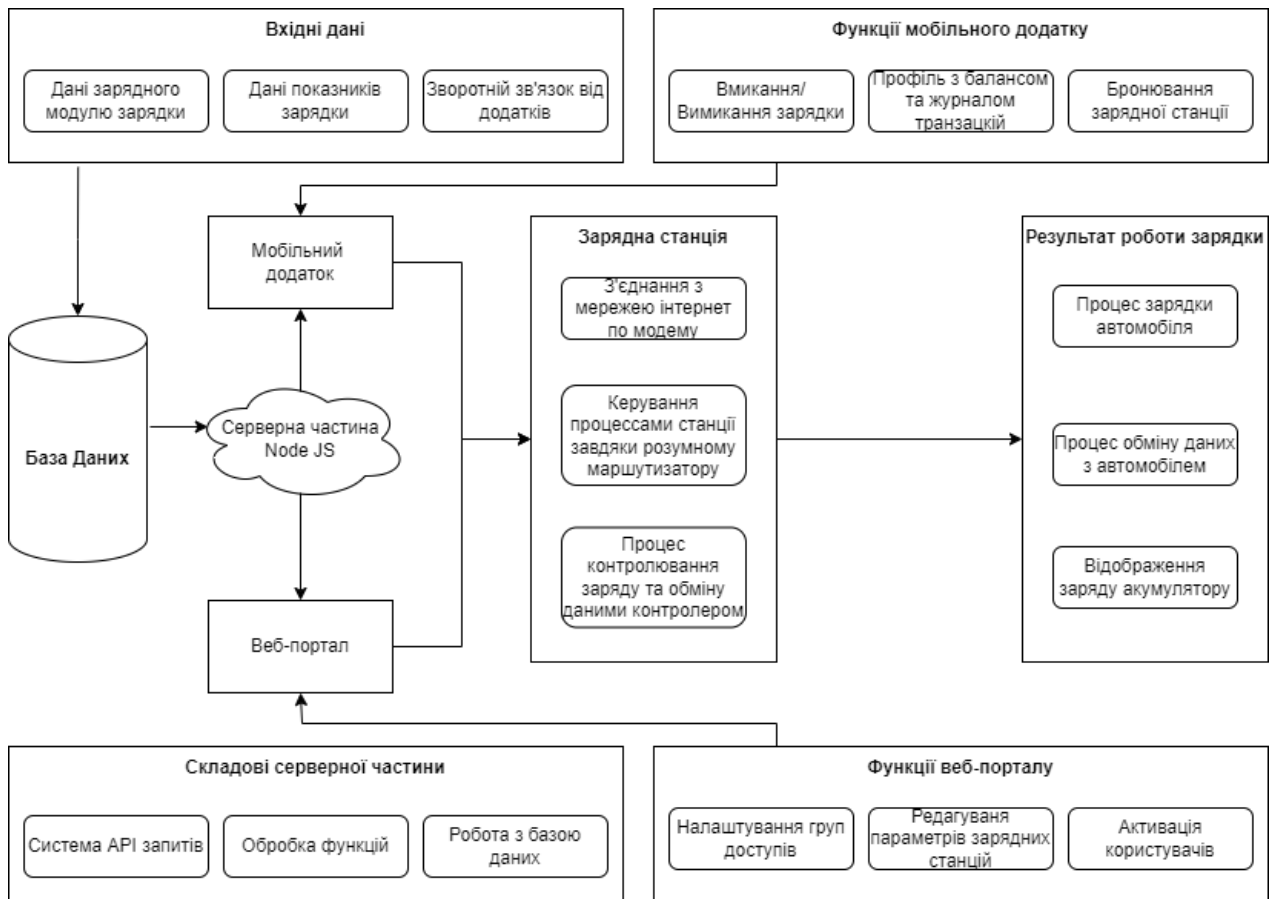


Рисунок 3.2 – Блок схема роботи програмного забезпечення інформаційної системи

Діаграма Use Case інформаційної системи для автоматизації роботи користувачів мережі електрозарядними станціями EcoStation буде мати вигляд, як зображено на рисунку 3.3.

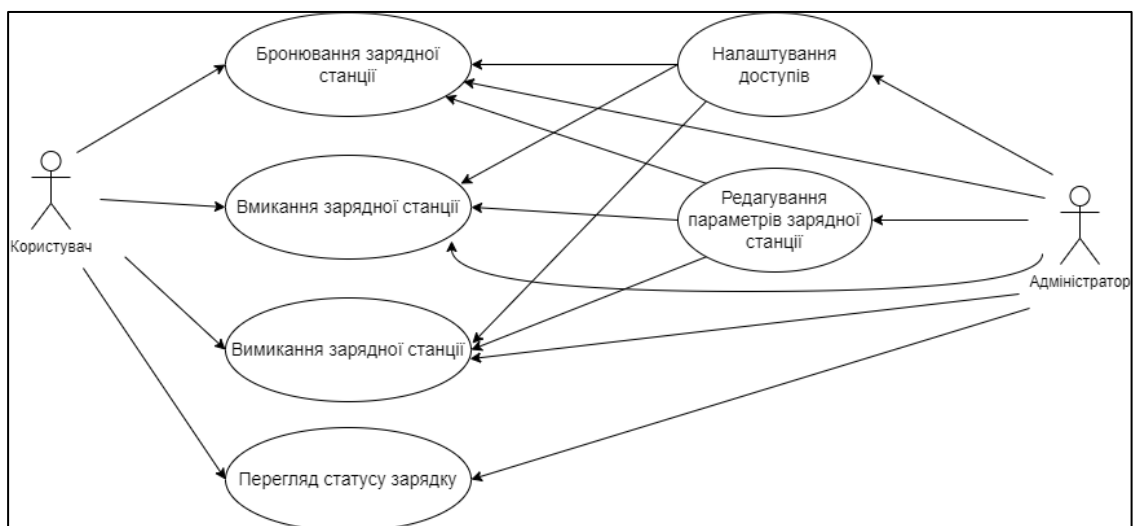


Рисунок 3.3 – Діаграма Use Case інформаційної системи

Основним модулем мобільного додатку є модуль реєстрації та аутентифікації користувача. Діаграма Use Case цього модулю зображено на рисунку 3.4.



Рисунок 3.4 – Діаграма Use Case модулю реєстрації та автентифікації

і

Також важливою частиною є розробка метода комунікація зарядної станції між своїми компонентами та серверною частиною тому нижче, на рисунку 3.5. На use case діаграмі зображено комунікацію зарядної станції, включаючи взаємодію між контролером, маршрутизатором та серверною частиною. Лінії з точками між компонентами символізують взаємодію та передачу даних. На кожній лінії вказано основну функцію, що відбувається під час цієї взаємодії: "Передача даних" між контролером та маршрутизатором та "Синхронізація" між маршрутизатором та серверною частиною.

Ця діаграма служить для візуалізації та розуміння процесів та взаємодій, які відбуваються в системі управління зарядною станцією.



Рисунок 3.5 – Діаграма Use Case методу комунікації та обміну даних інформаційної системи

3.2 Розроблення алгоритму методу автоматизації процесів обслуговування користувачів мережі електрзарядних станцій ECOSTATION

Розробимо алгоритм роботи методу автоматизації процесів обслуговування користувачів, де продемонструємо його можливості та взаємодію з основними компонентами системи, На зображеній блок-схемі, який проілюстровано на рисунку 3.6, представлено розробку алгоритму методу комунікації інформаційної системи автоматизації роботи користувачів мережі електрзарядної станції.

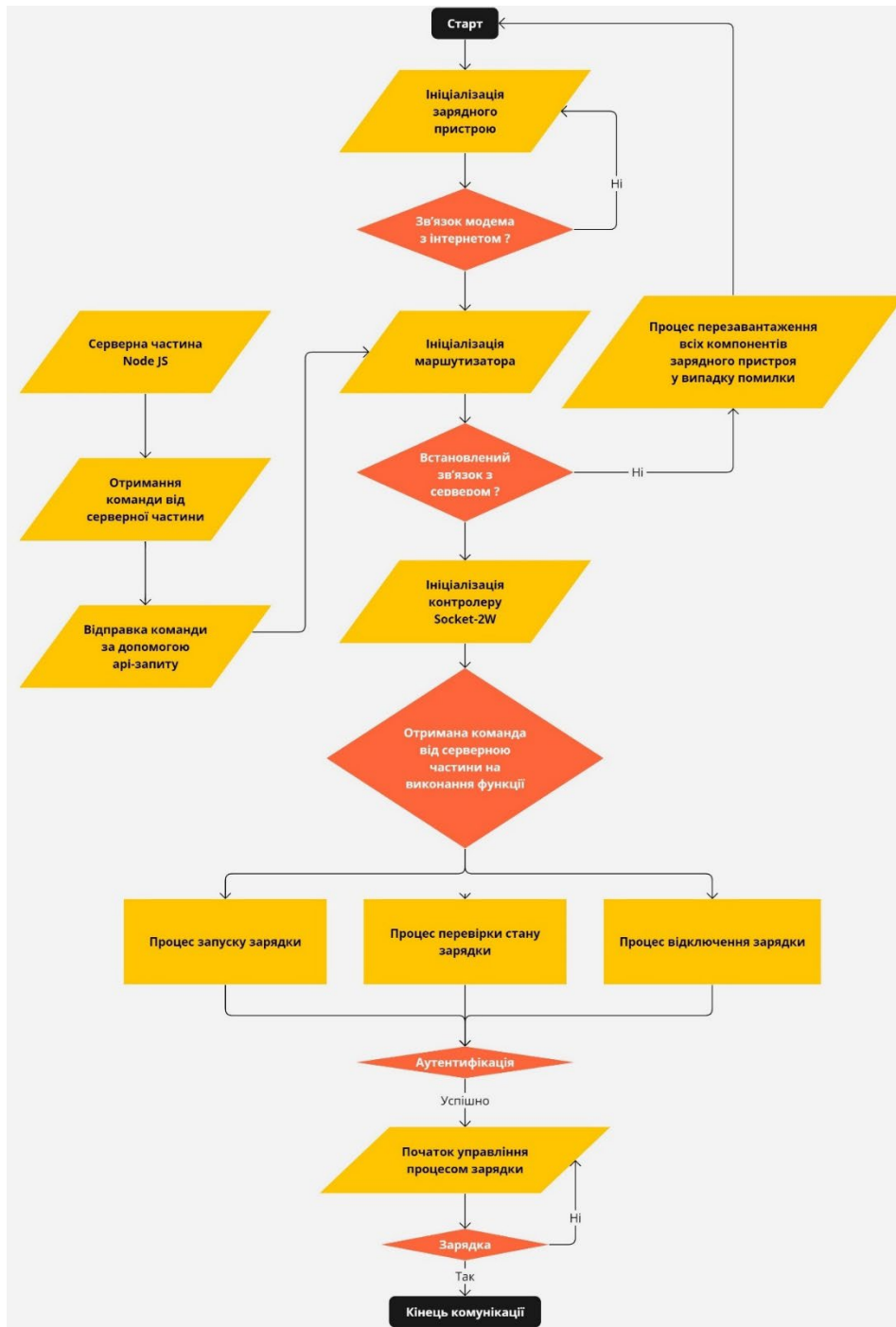


Рисунок 3.6 – Блок-схема алгоритму роботи методу комунікації

Діаграма починається зі "Старту" і містить серію операцій, рішень та процесів, які демонструють потік роботи системи. Опис процесів на діаграмі:

- Ініціалізація зарядного пристрою: Перевірка стану зарядного пристрою перед початком використання.

- Зв'язок з Інтернетом: Встановлення з'єднання з Інтернетом для комунікації з серверною частиною.
- Ініціалізація маршрутизатора: Налаштування маршрутизатора для правильної маршрутизації даних.
- Встановлення зв'язку з сервером: Перевірка з'єднання з серверною частиною.
- Ініціалізація контролеру Socket-2W: Підготовка контролера для управління процесом зарядки.
- Отримання команди від серверної частини: Прийом команд від сервера для управління зарядним пристроєм.
- Відправка команди до зарядного пристрою: Команди від сервера пересилаються до зарядного пристрою для виконання.
- Процес зарядки запускається/ініціюється/відхиляється: На основі отриманих команд відбувається або запуск зарядки, її перезапуск або відхилення.
- Аутентифікація: Перевірка ідентичності користувача, який ініціює зарядку.
- Початок управління процесом зарядки: Реальне управління зарядним процесом, після успішної аутентифікації.
- Зарядка завершена?: Перевірка статусу зарядки.
- Кінець комунікації: Закінчення процесу зарядки та комунікаційного обміну даними.

Ця діаграма була створена для візуалізації та деталізації кожного кроку в алгоритмі комунікації між зарядним пристроєм та сервером, що дозволяє розробникам і аналітикам глибше розуміти процеси та взаємодії в системі. Вона також служить для виявлення потенційних точок відмови та оптимізації потоку даних.

3.3 Реалізація мобільного додатку та веб-порталу інформаційної системи ECOSTATION

В результаті виконання бакалаврської дипломної роботи було розроблено клієнтські додатки для взаємодії з інформаційною системою мережі електрзарядних пристроїв ECOSTATION. А саме, було виконано:

Розробка мобільного застосунку ECOSTATION:

- Розробка застосунку була здійснена на платформі React Native з використанням JavaScript, включаючи інтеграцію API для з'єднання з серверною частиною та обробки клієнтських запитів;
- Була створена інтегрована система реєстрації та аутентифікації, яка об'єднувала застосунок і веб-сайт;
- Розроблено спеціалізований модуль користувацького кабінету;
- Реалізовано основний модуль для відображення та взаємодії з зарядними станціями;
- Здійснено дизайн мобільного застосунку та його наповнення контентом;
- Застосунок було опубліковано в Google Play Store.

Розробка веб-сайту ECOSTATION:

- Створено веб-сайт на основі JavaScript із використанням API для взаємодії з серверною та клієнтською частинами;
- Реалізовано спільну систему реєстрації та аутентифікації для застосунку та веб-сайту;
- Розроблено спеціалізований модуль користувацького кабінету;
- Створено панель управління правами доступу, систему груп і ролей;
- Забезпечено можливість редагування інформації про клієнтів та станції;

- Надано функціонал для повного керування електрзарядними станціями;
- Розроблено модуль для звітності про роботу мережі електрзарядних станцій та взаємодію з клієнтами.

В результаті поставлених задач, було виконано програмну реалізацію клієнтської частини, яка є важливою частиною загалом інформаційної системи та взаємодіє з серверною частиною, що грає дуже важливу роль у проекті, тому нижче буде приведено коротко про розробку:

Розробка мобільного додатку ECOSTATION стала значним кроком у автоматизації роботи користувачів у проекті мережі електрзарядних станцій. Використовуючи середовище JetBrains WebStorm, було створено проект на мові програмування JavaScript, інтегруючи його з React Native. Цей вибір технологій забезпечив гнучкість та ефективність у процесі розробки.

Основою для успішної реалізації проекту стало глибоке розуміння архітектури додатку. Ретельно продумана структура коду дозволила розробникам зосередитись на ключових аспектах додатку, таких як інтеграція API для взаємодії з серверною частиною, а також розробка інтуїтивно зрозумілих UI/UX елементів. Основний код додатку був розміщений у файлі App.js, який слугував відправною точкою для інших важливих компонентів.

Було приділено особливу увагу системі реєстрації та аутентифікації, що є критичною для забезпечення безпеки та простоти використання додатку. На UML діаграмах було чітко показано взаємодію між різними модулями, що сприяло зрозумілості та ефективності розробки.

Окрім архітектурних і технічних аспектів, велику увагу було приділено дизайну інтерфейсу. Також створено чіткий, зручний і візуально привабливий інтерфейс, який забезпечував легку навігацію по додатку. Особливу цінність представляв модуль Home.js, який включав мапу з маркерами зарядних станцій, що в значній мірі полегшував їх пошук і використання.

На завершальному етапі, після ретельного тестування та оптимізації, додаток ECOSTATION був успішно опублікований у Google Play Store. Це не тільки відкрило широкі можливості для користувачів, але й стало яскравим прикладом якісної та продуманої розробки в сфері мобільних технологій.

Результат реалізацію мобільного додатку представлений на рисунка 3.7 – 3.9.

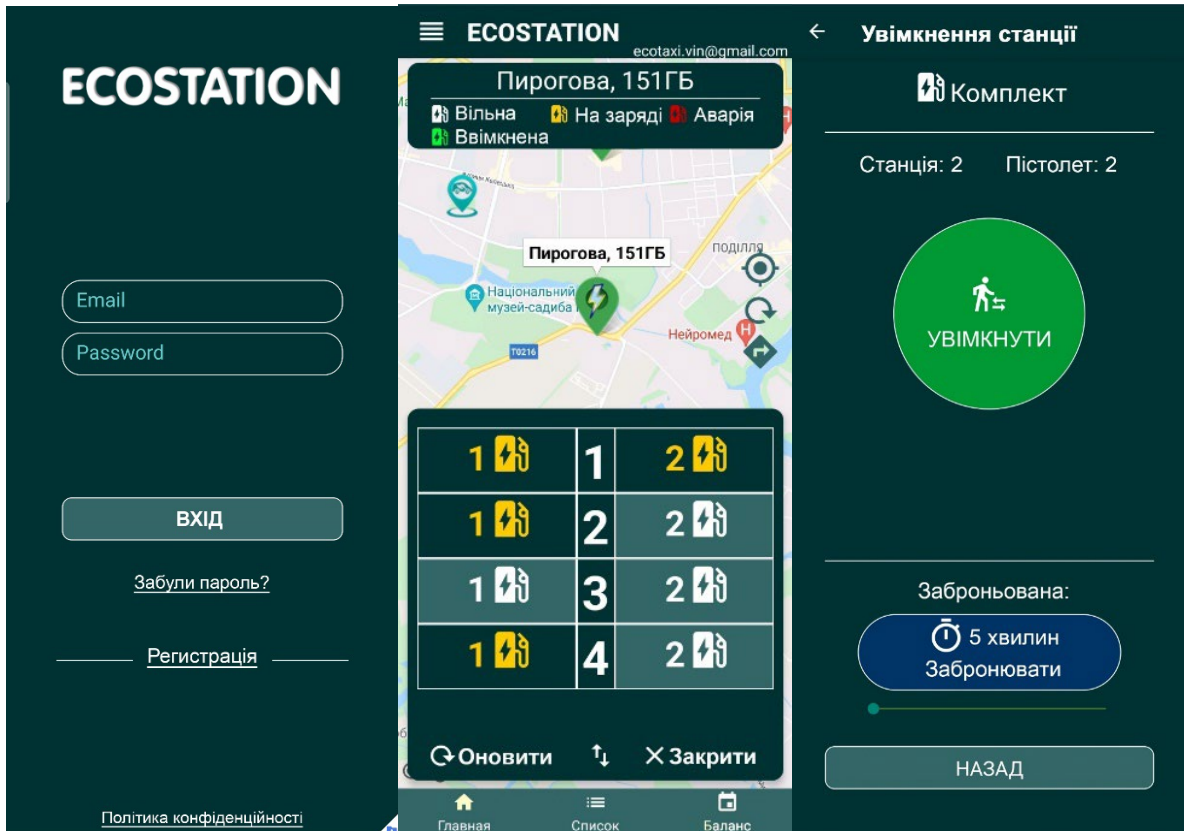


Рисунок 3.7 – 3.9 - Результат реалізацію мобільного додатку

Розробка веб-порталу ECOSTATION стала важливим етапом у створенні цілісної системи автоматизації роботи з мережею електрзарядних станцій. Використовуючи мову програмування JavaScript, команда розробників забезпечила взаємодію веб-порталу з серверною частиною за допомогою API-запитів, що дозволило створити динамічний та ефективний інструмент для керування системою зарядних станцій.

В основі веб-порталу лежала продумана структура коду, в якій ключову роль відігравали класи, відповідальні за різні аспекти управління системою. Клас Clients.ejs був розроблений для відображення та редагування інформації про

користувачів, включаючи їх особисті дані, баланс та статус аккаунту. Це забезпечило адміністраторам зручний інструмент для керування реєстрацією та активацією аккаунтів користувачів.

Клас Stations.ejs взяв на себе відповідальність за управління даними електрозарядних станцій. Він дозволив адміністраторам веб-порталу переглядати та редагувати інформацію про станції, управляти їх статусом та переглядати історію помилок та логів. Це стало особливо важливою функцією, яка забезпечила оперативне управління зарядними станціями і їх статусами.

Окрім цього, були розроблені додаткові модулі для редагування даних станцій та користувачів, а також для створення нових аккаунтів. Ці модулі забезпечили повний контроль над системою і дозволили адміністраторам налаштовувати доступ до різних груп станцій і користувачів.

В цілому, розробка веб-порталу ECOSTATION стала ключовим моментом у створенні ефективної та зручної системи управління мережею електрозарядних станцій. Кожен аспект порталу був ретельно продуманий і реалізований з метою забезпечення максимальної функціональності та зручності використання, що в кінцевому результаті сприяло підвищенню ефективності управління системою зарядних станцій.

Реалізація веб-порталу ECOSTATION було проілюстровано на рисунках 3.10 -3.11

Зарядки які:		Список станцій :										Користувач : vinnitsa		
		ID	Дата	На'н	Ф/хв	Тип	GPS	Адреса станції	Група	кВт/год	Статус	Виконати	Група : vinnitsa	
Заряджені		1-(153)	15.01.2022	1.2 АЗС Greenwave 2628	0.7	low	49.537014, 25.603634	Тернопіль, вул. Мителіничка	vinnitsa	6.9	Не Доступна!		Профіль	Вийти
На заряді		2-(152)	15.01.2022	1.1 АЗС Greenwave 2627	0.7	low	49.537014, 25.603634	Тернопіль, вул. Мителіничка	vinnitsa	6.9	Не Доступна!			
Недоступні		3-(151)	01.10.2021	1.1 Зулінського 46а, 2721	0.7	low	49.267870, 26.504442	Вінниця, Зулінського 46а	vinnitsa	6.9	Выключена	✓ ON		
Повні:		4-(150)	28.05.2021	1.2 Злуки 2, 2732	0.7	low	49.561882, 26.613365	Тернопіль, Злуки 2, заправка Автогасерсїс	vinnitsa	6.9	Не Доступна!			
Журнал подій		5-(149)	28.05.2021	1.1 Злуки 2, 2731	0.7	low	49.561882, 26.613365	Тернопіль, Злуки 2, заправка Автогасерсїс	vinnitsa	6.9	Не Доступна!			
Історія клієнтів		6-(148)	16.06.2021	2.2 Типографія 2722	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	Выключена	✓ ON		
Рух грошей		7-(147)	16.06.2021	2.1 Типографія 2721	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	Выключена	✓ ON		
Історія транзакцій		8-(146)	04.02.2022	1.2 Червонола 159/2 2764	0.7	low	49.402131, 27.056280	Хмельницький, Червонола 159/2	vinnitsa	7	Не Доступна!			
Лог помилок		9-(145)	04.02.2022	1.1 Червонола 159/2 2763	0.7	low	49.402131, 27.056280	Хмельницький, Червонола 159/2	vinnitsa	7	Не Доступна!			
		10-(144)	30.12.2020	4.2 Електросіль 2782	0.7	low	49.212022, 28.425395	Львівська, 151ГБ	vinnitsa	7	Выключена	✓ ON		
		11-(143)	18.12.2020	4.1 Електросіль 2781	0.7	low	49.212022, 28.425395	Львівська, 151ГБ	vinnitsa	7	Выключена	✓ ON		
		12-(142)	04.02.2022	1.2 Червонола 2771	0.7	low	49.402567, 27.058113	Запорожжя	vinnitsa	7	Не Доступна!			
		13-(141)	04.02.2022	1.1 Червонола 2770	0.7	low	49.402567, 27.058113	Запорожжя	vinnitsa	7	Не Доступна!			
		14-(140)	08.05.2021	1.2 Заправка Green wave 2728	0.7	low	49.561961, 25.540179	Тернопіль, П'яльська, заправка Green wave	vinnitsa	7	Не Доступна!			
		15-(139)	08.05.2021	1.1 Заправка Green wave 2727	0.7	low	49.561961, 25.540179	Тернопіль, П'яльська, заправка Green wave	vinnitsa	7	Не Доступна!			
		16-(138)	06.05.2021	1.2 Заправка Автогасерсїс 2793	0.7	low	49.545845, 25.639472	Тернопіль, Заправка Автогасерсїс	vinnitsa	7	Не Доступна!			
		17-(137)	06.05.2021	1.1 Заправка Автогасерсїс 2792	0.7	low	49.545845, 25.639472	Тернопіль, Заправка Автогасерсїс	vinnitsa	7	Не Доступна!			
		18-(136)	03.02.2021	1.2 Стадіон 2746	0.7	low	49.420319, 26.964412	Хмельницький, Прокурівська 8311	vinnitsa	7	Выключена	✓ ON		
		19-(135)	03.02.2021	1.1 Стадіон 2745	0.7	low	49.420319, 26.964412	Хмельницький, Прокурівська 8311	vinnitsa	7	Выключена	✓ ON		
		20-(134)	12.04.2021	1.2 Типографія 2859	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	Выключена	✓ ON		
		21-(133)	12.04.2021	1.1 Типографія 2858	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	6.9	Выключена	✓ ON		
		22-(132)	13.11.2020	1.2 Зоопарк 2707	0.7	low	49.269907, 28.477305	Вінниця, Зоопарк	vinnitsa	7	Выключена	✓ ON		
		23-(131)	13.11.2020	1.1 Зоопарк 2706	0.7	low	49.269907, 28.477305	Вінниця, Зоопарк	vinnitsa	7	На Заряді	✓ OFF		
		24-(130)	26.01.2022	2.2 Довіська, Тамонан 2770	0.7	low	49.239982, 28.394393	Довіська, Тамонан	vinnitsa	7	Выключена	✓ ON		
		25-(129)	26.01.2022	2.1 Довіська, Тамонан 2770	0.7	low	49.239982, 28.394393	Довіська, Тамонан	vinnitsa	7	Выключена	✓ ON		
		26-(128)	02.02.2021	4.3 Довіська, Тамонан 2775	5	high	49.239982, 28.394393	Довіська, Тамонан	vinnitsa	20	Заброньована			
		27-(127)	16.12.2020	1.2 Панася Мирного 41 2734	0.7	low	49.456925, 27.005328	Хмельницький, Панася Мирного 41	vinnitsa	7	Выключена	✓ ON		
		28-(126)	26.10.2020	1.1 Панася Мирного 41 2733	0.7	low	49.456925, 27.005328	Хмельницький, Панася Мирного 41	vinnitsa	7	Выключена	✓ ON		
		29-(125)	23.11.2021	1.2 Пляж 2771	0.7	low	49.435282, 26.981834	Хмельницький, Пляж	vinnitsa	7	Выключена	✓ ON		
		30-(124)	23.11.2021	1.1 Пляж 2770	0.7	low	49.435282, 26.981834	Хмельницький, Пляж	vinnitsa	7	На Заряді	✓ OFF		
		31-(123)	11.12.2020	1.2 Прибузька 11 2752	0.7	low	49.422570, 27.010219	Хмельницький, Прибузька 11	vinnitsa	7	Выключена	✓ ON		
		32-(122)	11.12.2020	1.1 Прибузька 11 2751	0.7	low	49.422570, 27.010219	Хмельницький, Прибузька 11	vinnitsa	7	Выключена	✓ ON		
		33-(121)	21.10.2020	1.2 Колодязька 2747	0.7	low	48.438438, 26.860453	Хмельницький, Колодязька	vinnitsa	7	На Заряді	✓ OFF		

Рисунок 3.10 - Результат роботи модулю stations

Редагування електрозарядок	
Имя:	<input type="text" value="station-eco"/> *
Адрес:	<input type="text" value="Хмельницьке Шоссе 107 б"/> *
gps:	<input type="text" value="49.23075 28.3990203"/> *
ip:	<input type="text" value="10.55.52.10"/> *
port:	<input type="text" value="2581"/> *
Доступность:	<input type="text" value="/ajax_inputs"/> *
ON:	<input type="text" value="?R0=1"/> *
OFF:	<input type="text" value="?R0=0"/> *
Логин:пароль:	<input type="text" value="admin:admin123"/> *
Тариф грн./мин.:	<input type="text" value="1"/> *
Power кВт:	<input type="text" value="7.4"/> *
Группа:	<input type="text" value="vinnitsa"/> ▼
Вид станции:	<input type="text" value="low"/> ▼
<input type="button" value="Сохранить"/>	
<input type="button" value="К списку станций"/>	

Рисунок 3.11 – Результат коду модулю редагування station_edit.ejs

3.3 Програмна реалізація методу автоматизація процесів обслуговування користувачів

Програмну реалізацію автоматизації процесів обслуговування користувачів мереж електрозарядок ECOSTATION, базується на базі серверною частини та написаний на мові Java Script , а саме на платформі Node JS, на якій також розміщений веб-портал. Обробка всіх процесів та функцій виконується на серверній частині та розміщена по модулям згідно з функціоналу, в основному же класі routes.js описаний основний функціонал, підключенні основні бібліотеки.

На рисунку 3.12 представлена діаграма основних класів серверною частини.

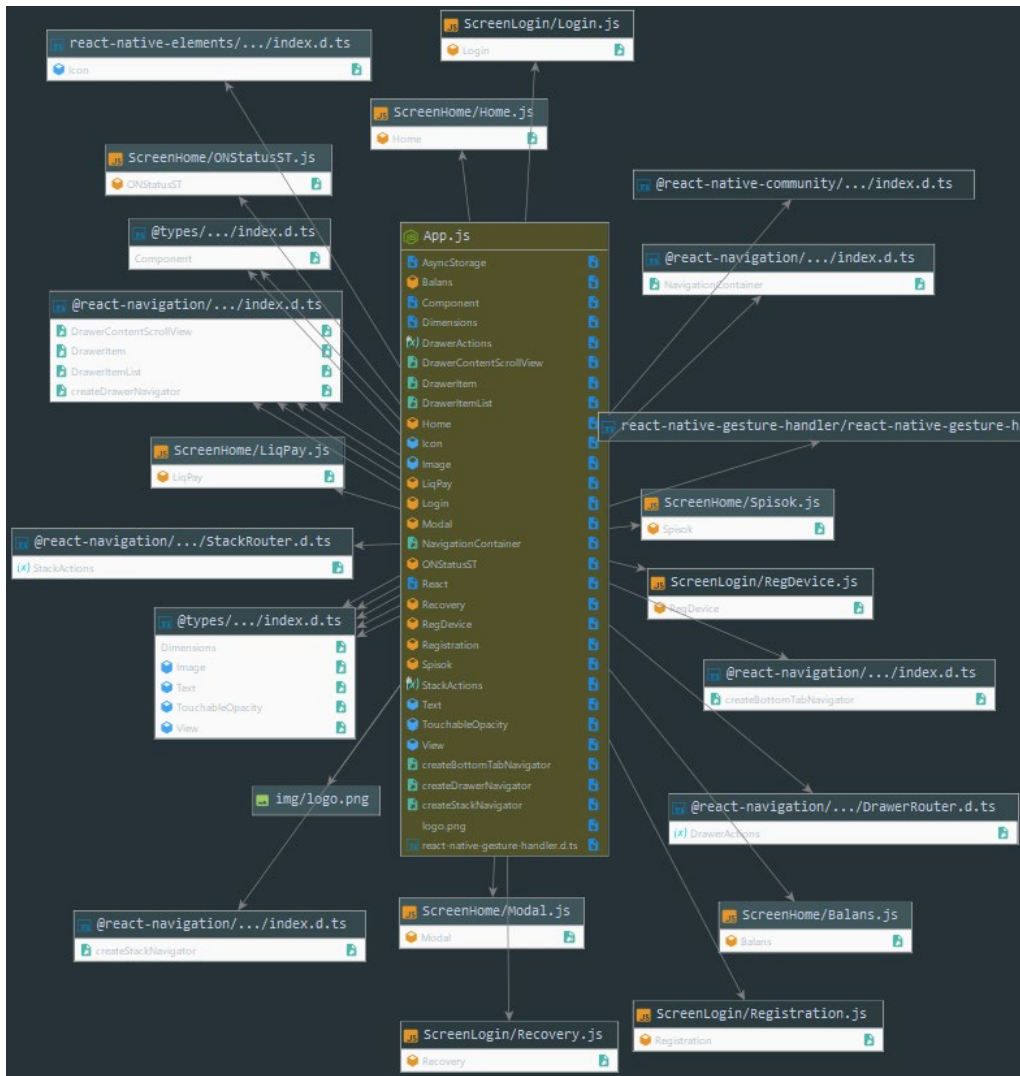


Рисунок 3.12 – Діаграма класів серверною частиною інформаційної системи

Далі розробку потрібно розпочати з налаштуванням апаратної частини електрозарядних пристроїв з серверними машинами для правильності роботи системи. На рисунку 3.13, показана блок-схема взаємозв'язків апаратної частини зі серверною.

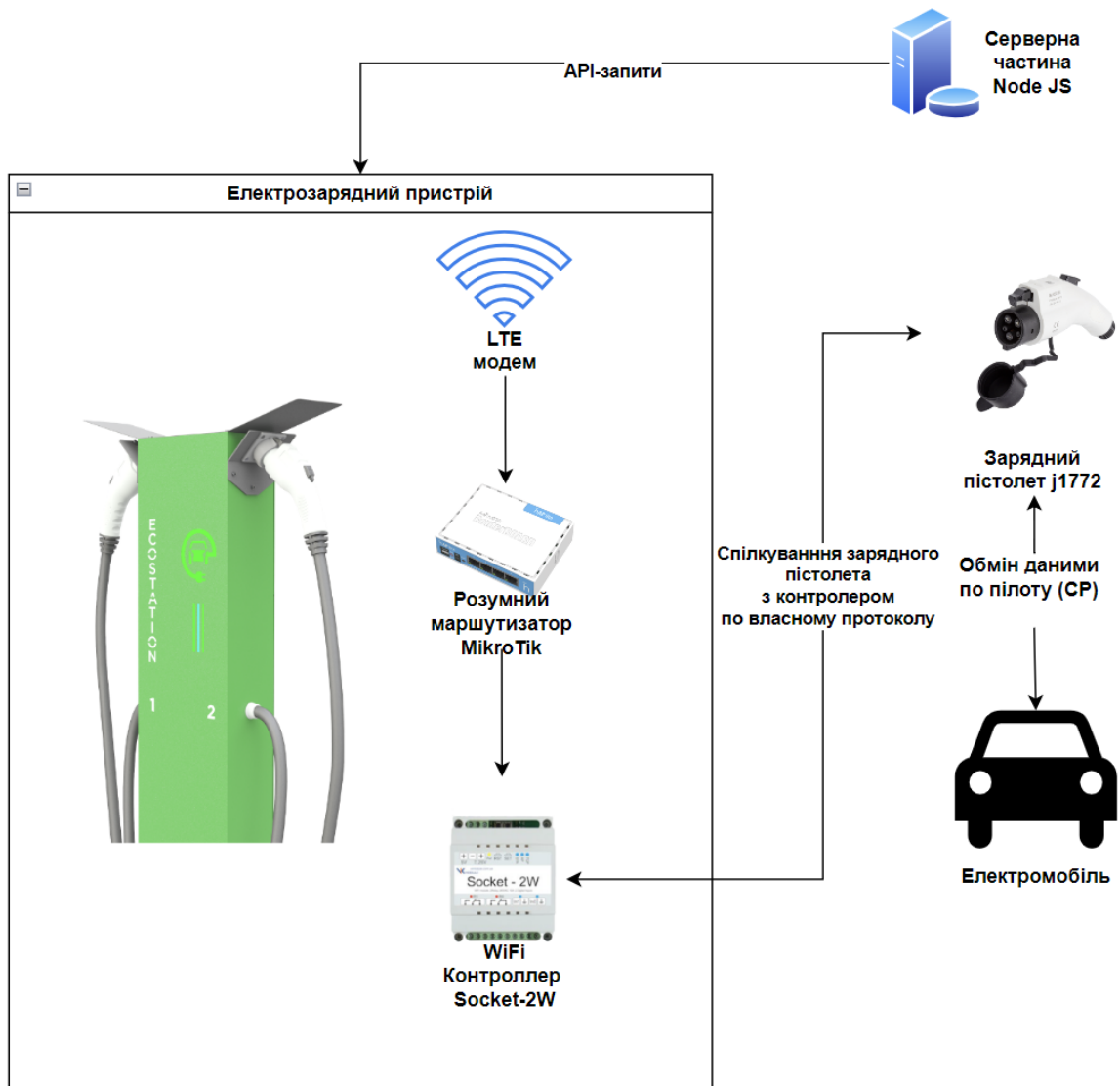


Рисунок 3.13 – Блок схема взаємозв'язків апаратної частини зі серверною

Для початку функціонування, розберемо електрозарядний пристрій, для його функціонування потрібно налаштувати розумний маршрутизатор MikroTik, налаштувати на ньому DHCP-сервер, прописати в ARP-таблицю всі MAC-адреси та контролерів зарядного пристрою та прив'язати IP-адреси, також налаштувати модем та доступи щодо приймання API-запитів з певних адресів та портів у фаєрволі маршрутизатора. На базі RouterOS та програмного забезпечення WinBox ми маємо можливість виконати налаштування всіх цих параметрів. Нижче на рисунці 3.14-3.15, представлено налаштування параметрів маршрутизатора для роботи з серверною частиною.

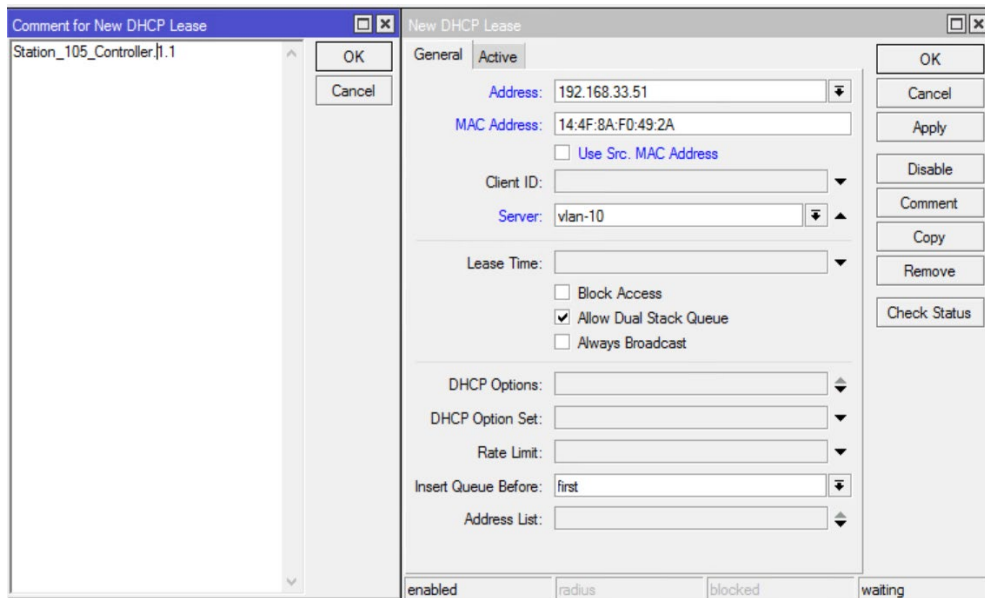


Рисунок 3.14 – Додавання нового контролера в список DHCP-серверу та ARP-таблиці

#	Action	Chain	Src. Address	Dst. Address	Proto.	Src. Port	Dst. Port	In. Interface	Out. Int...	In. Inter...	Out. Int...	Src. Ad...	Dst. Ad...	Bytes	Packets
0	drop	forward	69.69.69.0/24											9.4 MB	46 299
1	drop	input			17 (u...	53				WAN				1100.9 KB	17 772
2	drop	forward			17 (u...	53				WAN				68.0 MB	527 841
3	drop	forward		69.69.69.0/24										7.9 KB	72
4	drop	forward	192.168.51.0/24											0 B	0
5	drop	forward		192.168.51.0/24										9.0 MB	125 340
6	drop	forward	192.168.2.80	192.168.0.0/16										0 B	0
7	drop	forward	192.168.2.80	110.101.0.0/16										0 B	0
8	drop	forward	192.168.3.100	192.168.0.0/16										41.6 KB	654
9	drop	forward	192.168.50.11	192.168.0.0/16										284.9 MB	5 296 008
10	drop	input	192.168.200.0/22		1 (c...									499.2 MB	1 302 598
11	drop	input	192.168.90.0/24		1 (c...									0 B	0
12	drop	input	8.8.4.4		1 (c...			Internet1						4859.5 KB	88 860
13	drop	input	8.8.4.4		1 (c...									0 B	0
14	acc...	input	192.168.0.0/16		1 (c...									213.0 MB	1 633 327
15	acc...	forward	192.168.0.0/16		1 (c...									23.5 GiB	297 830 ...
16	acc...	input	10.101.0.0/23		1 (c...									0 B	0
17	acc...	input	10.0.104.0/24		1 (c...			wiz_sky_net_ovpn						14.4 KB	245

Рисунок 3.15 – Налаштування фаєрволу маршрутизатора Mikrotik

Після налаштування зв'язків між серверною частиною, розумним маршрутизатором та контролером, потрібно виконати початкові налаштування контролера Socket-2W для його роботи, а саме виконати такий перелік дій:

- Потрібно обрати та вписати назву пристрою для майбутньої швидкої навігації серед пристроїв на маршрутизаторі;

- Контролер потрібно підключити до WiFi-мережі маршрутизатора, вписавши SSID мережі, логін та пароль, також потрібно прописати мережеві дані по типу статичного IP-адресу, шлюзу, підмаска, DNS-сервера, мережевий протокол, сокет та інших мережевих налаштувань;
- Також потрібно налаштувати входи та виходи контролера, налаштувати реле та затримку по спрацюванню.

The screenshot displays the configuration page for the Socket-2W device. The settings are as follows:

- Имя устройства:** Socket-2W
- Имя WiFi сети:** VKmodule
- Пароль:** 0123456787
- Имя WiFi сети AP:** Socket-2W
- Пароль AP:** (empty)
- IP Address:** 192.168.0.191
- Gateway:** 192.168.0.1
- Subnet:** 255.255.255.0
- DNS Address:** 192.168.0.1
- IP Address AP:** 192.168.0.191
- Socket Port:** 9761
- Protocol:** TCP
- Socket:** Server
- Server IP Addr:** 192.168.0.192
- Новый пароль:** (masked with dots)
- Подтверждение:** (masked with dots)
- Связь кнопка -> реле:** Откл
- Связь двух устройств кнопка -> реле:** Откл
- Сохранение состояний реле:**
 - Реле 0: Откл
 - Реле 1: Откл
- Входы:**
 - Вход 0: Цифровой
 - Вкл
 - Антидребезг: 60 (0-255мс)
 - Вход 1: Вкл
 - Антидребезг: 60 (0-255мс)

A "Сохранить" (Save) button is located at the bottom of the form.

Рисунок 3.16 – Початкова сторінка налаштування контролера Socket-2W

Далі встановивши та відкривши програму «Socket-2W Control and Testing», яка працює по TCP/IP і призначена для перевірки та керування контролером, а також для демонстрації протоколу обміну даними. Працює під управлінням Windows. Пакети команд, що відсилаються, та отримані відповіді відображаються у вікні програми, що дозволяє використовувати програму для розробки/налагодження власного ПЗ по контролеру Socket-2W (рис.3.17).

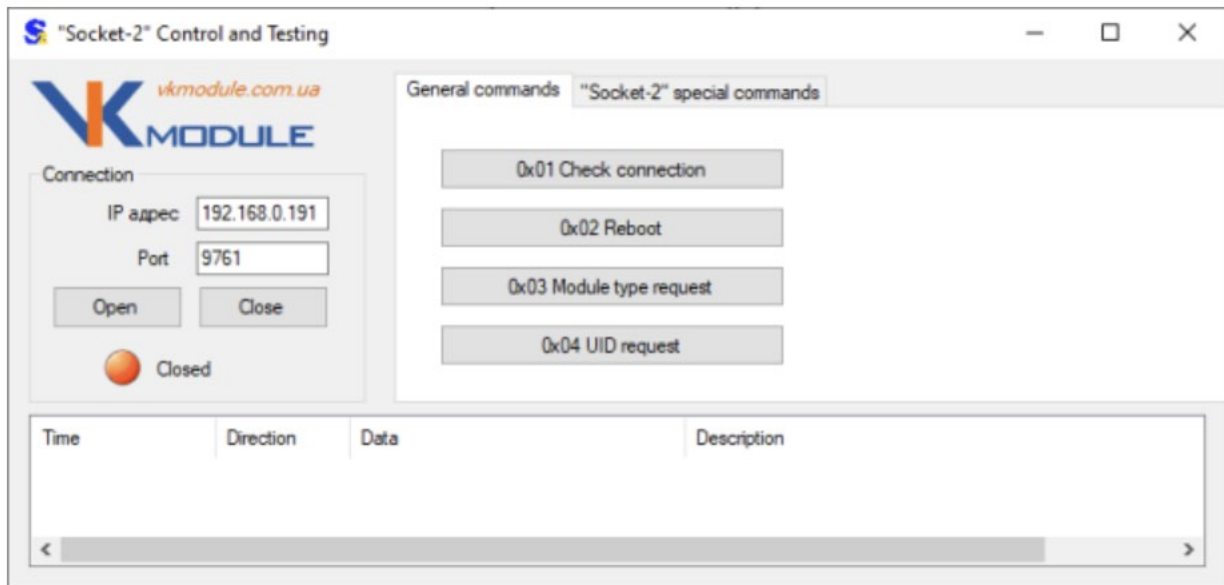


Рисунок 3.17 - Програма «Socket-2W Control and Testing»

Нижня частина вікна містить список команд та отриманих відповідей. Верхня ліва частина вікна дозволяє встановити налаштування для з'єднання з контролером. Верхня права частина вікна містить дві закладки з групами команд. Закладка «General commands» дозволяє виконати спільні для кожного Ethernet-контролер команди. Закладка «Socket-2W special commands» виконує команди, які відносяться лише до контролера Socket-2W (рис. 3.18). Нам потрібно завантажити конфігурацію контролера, розробленою для проекту мережі електростанцій та завдяки програмі переконатись в роботі контролера.

Далі відправивши API-запит перевіряємо статус його виконання і чи все правильно працює, та отримаємо позитивний результат про інформацію від користувача який залогінився в інформаційну та перевіряє статус зарядки в даний момент, результат продемонстровано на рисунку 3.18

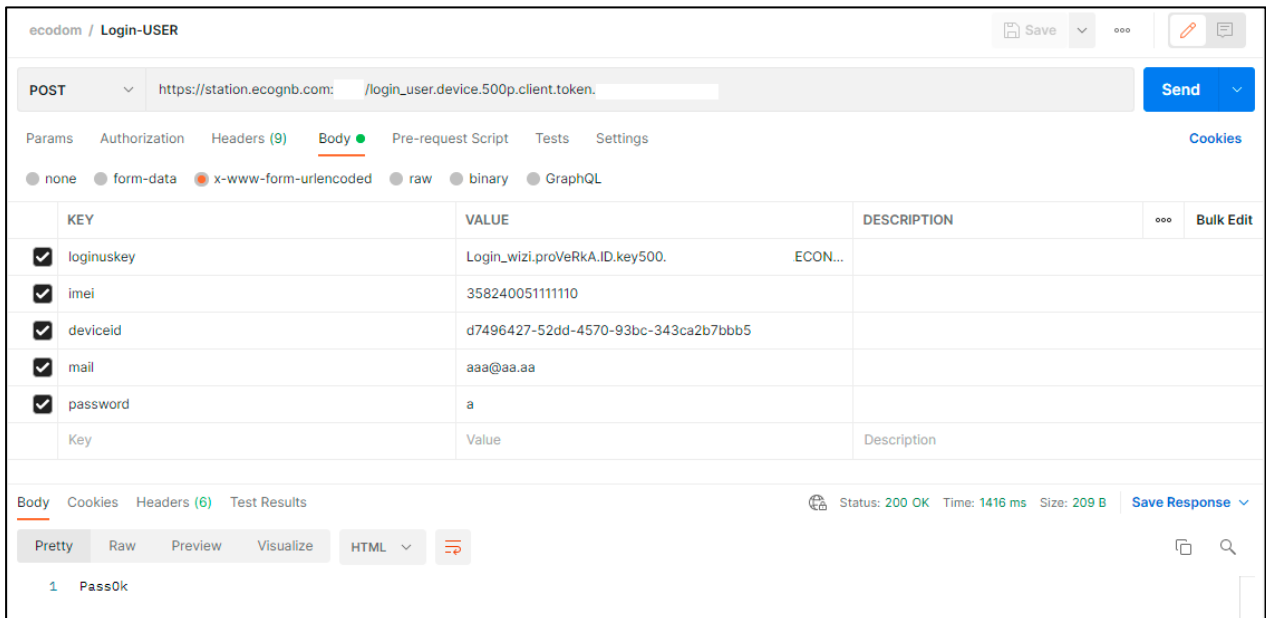


Рисунок 3.18 – Тестовий API-запит

Далі потрібно створити основні функції комунікації серверною частиною та апаратною частиною електрозарядного пристрою, складова програмного забезпечення серверною частиною та апаратною частиною роби метод універсальним та дає змогу використовувати його як універсальний протокол для зв'язку з різними типами інтерфейсів зарядних пістолетів та роз'ємів. Зв'язок між сервером та апаратною частиною підтримується завдяки API-запитам. Нижче буде наведено код та пояснення коду щодо основних функцій методу:

– Модуль `StatusStUserZaryad.js` – Модуль, який відповідає за отримання користувачем статусу зарядки та взагалі даних про стан зарядки в даний момент завдяки API-запитам.

```

//*****
2 usages
function stations_request(id, status, url, groupuser) {
  return new Promise( (resolve) => {
    request.get({ // ЗАПРОС К СТАНЦИИ НА ОТКЛИК
      url: url,

    }, (err, response, body) => {
      if (err) {
        zaprosOFF_station(id, status); // ОШИБКА ЗАПРОСА - ЗАПРАВКА НЕДОСТУПНА
        resolve ("ERROR");
      }else {
        parseString(body, function (err, result) {
          on_off_station(id, status, result.inputs.LED[0], groupuser); // возврат Данных ОК
          resolve ("OK");
        });
      }
    });
  });
}
//***** ON *****
1 usage
function on_off_station(id, status, switch_st, groupuser){ // ЗАПРАВКА ON - OFF
  //console.log(data.inputs.switch);

  if (switch_st === "on"){status_update_station(id, status, groupuser);} // ЗАРАДКА
  if (switch_st === "off"){status_update_station(id, status, "off"+groupuser);
  insert_hystory_client(groupuser, id, groupuser);
  } // АКТИВНА

  console.log("portal: "+ id + " запрвка - " + switch_st); // ЗДЕСЬ БУДЕМ ПИСАТЬ В ЛОГ
}
//***** OFF *****

```

Рисунок 3.19 – Код модулю StatusStUserZaryad.js

Демонстрацію роботи цього запиту можливо побачити у вигляді статусу який постійно відслідковується та зміни можна побачити в мобільному додатку, а також веб-порталі а також на самому корпусу зарядки в результаті чого світова ідентифікація може змінюватись в залежності від статусу. Результати ідентифікація продемонстровані на рисунках 3.20-3.23.

Список станцій :

ID	Дата	Ім'я	№/кв.	Тип	GPS	Адреса станції	Група	кВт/год	Статус	Виконати
1-(153)	15.01.2022	1.2 АЗС Greenwave 2628	0.7	low	49.537014, 25.603634	Тернопіль, вул. Микілинецька	vinnitsa	6.9	Не Доступна!	
2-(152)	15.01.2022	1.1 АЗС Greenwave 2627	0.7	low	49.537014, 25.603634	Тернопіль, вул. Микілинецька	vinnitsa	6.9	Не Доступна!	
3-(151)	01.10.2021	1.1 Зупинського 46а, 2721	0.7	low	49.267870, 28.504442	Виниця, Зупинського 46а	vinnitsa	6.9	Виключена	✓ ON
4-(150)	28.05.2021	1.2 Злуки 2, 2732	0.7	low	49.561882, 25.613365	Тернопіль, Злуки 2, заправка Автотехсервіс	vinnitsa	6.9	Не Доступна!	
5-(149)	28.05.2021	1.1 Злуки 2, 2731	0.7	low	49.561882, 25.613365	Тернопіль, Злуки 2, заправка Автотехсервіс	vinnitsa	6.9	Не Доступна!	
6-(148)	16.06.2021	2.2 Типографія 2722	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	Не Доступна!	
7-(147)	16.06.2021	2.1 Типографія 2721	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	Виключена	✓ ON
8-(146)	04.02.2022	1.2 Чорновола 159/2 2764	0.7	low	49.402131, 27.056280	Хмельницький, Чорновола 159/2	vinnitsa	7	Не Доступна!	
9-(145)	04.02.2022	1.1 Чорновола 159/2 2763	0.7	low	49.402131, 27.056280	Хмельницький, Чорновола 159/2	vinnitsa	7	Не Доступна!	
10-(144)	30.12.2020	4.2 Електросеть 2782	0.7	low	49.212022, 28.425396	Пирогова, 151ГБ	vinnitsa	7	Виключена	✓ ON
11-(143)	18.12.2020	4.1 Електросеть 2781	0.7	low	49.212022, 28.425396	Пирогова, 151ГБ	vinnitsa	7	Виключена	✓ ON
12-(142)	04.02.2022	1.2 Чорновола 2771	0.7	low	49.402557, 27.058113	Запална	vinnitsa	7	Не Доступна!	
13-(141)	04.02.2022	1.1 Чорновола 2770	0.7	low	49.402557, 27.058113	Запална	vinnitsa	7	Не Доступна!	

Користувач: vinnitsa
Група: vinnitsa

Профіль Вихід

Відключені клієнти

Створити клієнта

Увімкнена
На заряді
Недоступна!

Вимкнена
Бронь
Аварія!

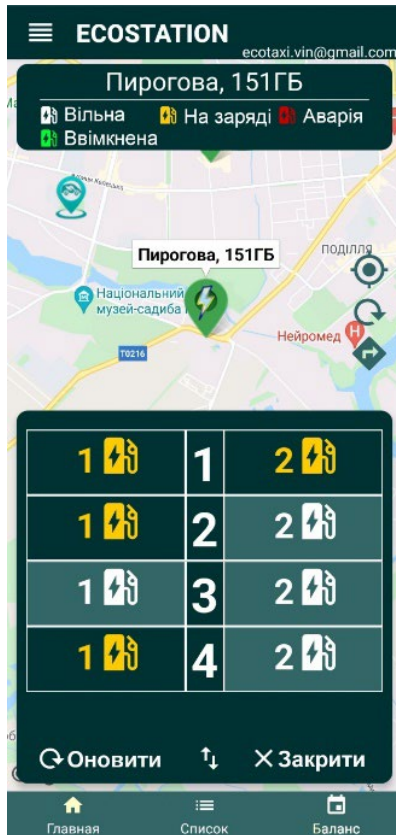


Рисунок 3.20 – 3.23 – Результат роботи запиту перевірки статусу, який демонструється в мобільному та веб-додатку, а також світове відображення на самій зарядці.

– Модуль Stations_Use.js – Модуль який відповідає за зміну статусу щодо включення або вимкнення електрозарядного пристрої. Також перевірки груп, запису в лог та журнал транзакцій.

```

const OnOffST = (id, imei, swst, erroroff) : Promise<unknown> => //STATIONS_-ON-OFF
new Promise<(executor: (resolve, reject) : void => {
  let details : {akeyecode: string, erroroff: any, imei: any, nstation: any, swst: any} = {
    akeyeco: '*****',
    imei: imei,
    swst: swst, //on-off
    nstation: id, //ID станції
    erroroff: erroroff,
  };

  let formBody : any[] = [];
  for (let property in details) {
    let encodedKey : string = encodeURIComponent(property);
    let encodedValue : string = encodeURIComponent(details[property]);
    formBody.push(encodedKey + '=' + encodedValue);
  }
  formBody = formBody.join('&');

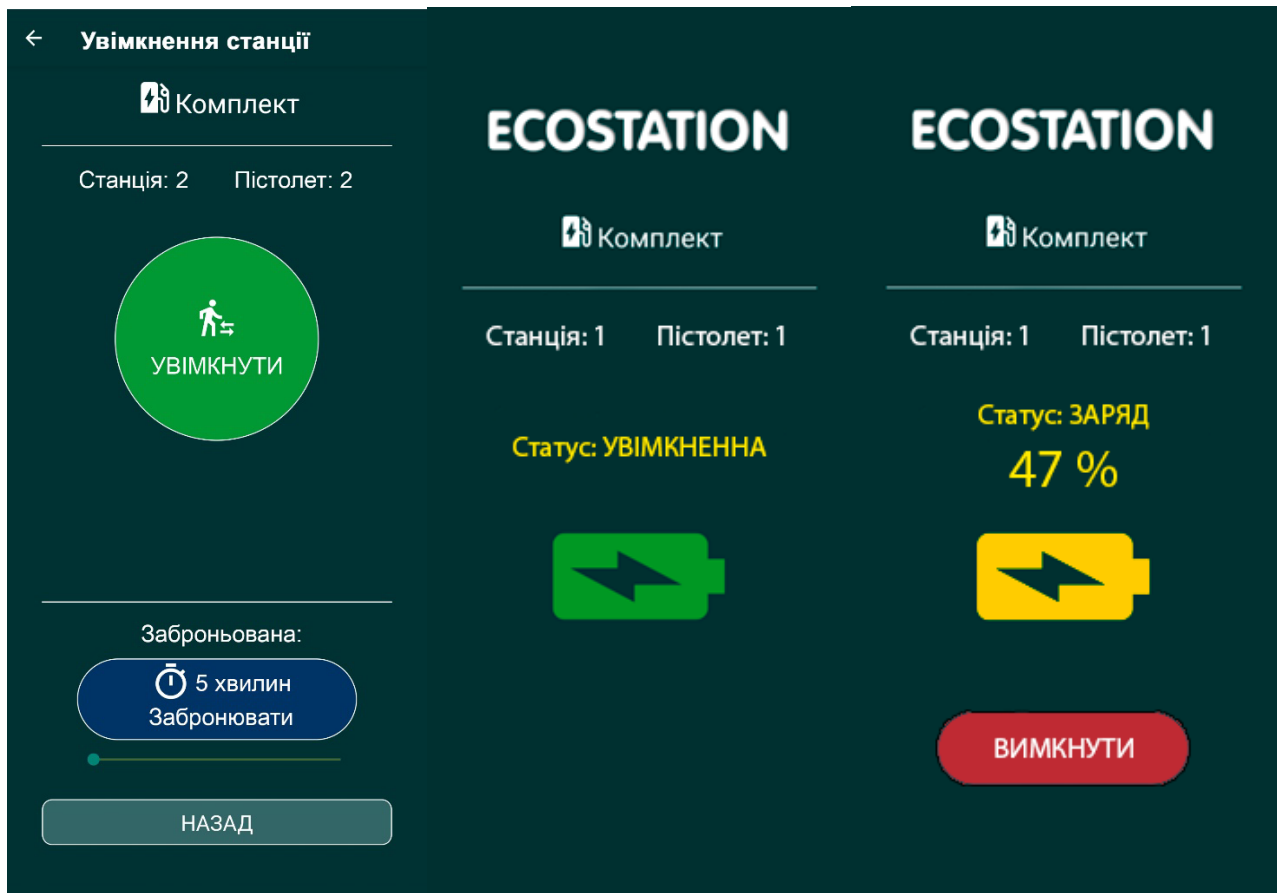
  fetch( {input: 'https://station.ecognb.com:****/*****',
    method: 'POST',
    headers: {
      Authorization: 'Bearer token',
      'Content-Type': 'application/x-www-form-urlencoded',
    },
    body: formBody,
  }) Promise<Response>
  .then(response : Response => response.text()) Promise<string>
  .then(result : string => {
    resolve(result);
  })
  .catch(error => {
    console.error(error);
    resolve( {value: 'ERROR'} );
  });
});

app.get("/stati_-----/:id", isLoggedIn, async function(req, res){
  const num_start = req.query.num;
  const id = req.params.id;
  const sw = req.query.sw;
  const power = '';
  let nday=dateFormat(new Date(), "yyyy-mm-dd HH:MM:ss");
  console.log(sw);
  const data_station = await station_sql(id); //получає дані з бази
  let status = data_station[0].status;
  if(data_station[0].groupid === req.user.groupid || req.user.groupid === "admin") { //перевіряє групу
    if (status !== 0){
      if(sw=="on"){
        let url = data_station[0].logpass + data_station[0].ip + ":" + data_station[0].port +
          data_station[0].checkst + data_station[0].onst;
        await stations_request(id, 4, url, req.user.groupid);
      }
      if(sw=="off"){
        let url = data_station[0].logpass + data_station[0].ip + ":" + data_station[0].port +
          data_station[0].checkst + data_station[0].offst;
        await stations_request(id, 1, url, req.user.groupid);
      }
      insert_on_off_bron(nday, req.user.groupid, req.user.groupid, id, sw, power); //ВІДПРАД.
      res.redirect("/stations_us?num="+num_start);
    }else{res.redirect("/stations_us?num="+num_start);}
  }else {res.redirect('/')};
});

```

Рисунок 3.24,3.25– Код модулю Stations_Use.js

Запити по включенню, або відключенню процесу зарядки здійснюються за допомогою мобільного додатку та веб-порталу, обравши операцію та підтвердивши її в додатках, запускається транзакція яка вже дає команду на контролер щодо зміни статусу на включений або вимкнений. Демонстрація результатів роботи модулю у вигляді візуалізацію у додатків ілюстровано на рисунках 3.26 – 3.30.



ON Головна Клієнти Заправки Звіти

Список станцій :

- ID -	Дата	Ім'я	№/хв.	Тип	GPS	Адреса станції	Група	кВт/год	Статус	Виконати
1-(153)	15.01.2022	▲ log ● hstr 1.2 АЗС Greenwave 2628	0.7	low	49.537014, 25.603634	Тернопіль, вул. Микулинецька	vinnitsa	6.9	● Не Доступна!	
2-(152)	15.01.2022	▲ log ● hstr 1.1 АЗС Greenwave 2627	0.7	low	49.537014, 25.603634	Тернопіль, вул. Микулинецька	vinnitsa	6.9	● Не Доступна!	
3-(151)	01.10.2021	▲ log ● hstr 1.1 Зулинського 46в, 2721	0.7	low	49.267870, 28.504442	Винниця, Зулинського 46в	vinnitsa	6.9	● Виякличена	✓ ON
4-(150)	28.05.2021	▲ log ● hstr 1.2 Злуки 2, 2732	0.7	low	49.561882, 25.613365	Тернопіль, Злуки 2, заправка Автотехсервіс	vinnitsa	6.9	● Не Доступна!	
5-(149)	28.05.2021	▲ log ● hstr 1.1 Злуки 2, 2731	0.7	low	49.561882, 25.613365	Тернопіль, Злуки 2, заправка Автотехсервіс	vinnitsa	6.9	● Не Доступна!	
6-(148)	16.06.2021	▲ log ● hstr 2.2 Типографія 2722	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	● Не Доступна!	
7-(147)	16.06.2021	▲ log ● hstr 2.1 Типографія 2721	0.7	low	49.442790, 26.993168	Хмельницький, Проспект миру, 59	vinnitsa	7	● Виякличена	✓ ON
8-(146)	04.02.2022	▲ log ● hstr 1.2 Чорновола 159/2 2764	0.7	low	49.402131, 27.056280	Хмельницький, Чорновола 159/2	vinnitsa	7	● Не Доступна!	

Рисунки 3.26 – 3.30 – Результати візуалізації роботи модулю Stations_Use.js в мобільному додатку та веб порталі

– Модуль `zaprosoff_station.js` – модуль який відповідає за функцію моніторингу відключення станції у випадку аварії зарядки, запис в журнал та лог та також, відключення зарядки.

```

}
//***** OFF *****
1 usage
function zaprosOFF_station(id, status_old) { // ЗАПРАВКА НЕ ДОСТУПНА!
  pool.query("SELECT status FROM stations WHERE id=?", [id], function (err, data) {
    if (err) return console.log(err);
    if (data[0].status !== 0) {
      status_update_station(id, 0);
    } // ЗАПРАВКА НЕ ДОСТУПНА
  });
  console.log(id + " " + status_old + " - OFF"); // ЗДЕСЬ БУДЕМ ПИСАТЬ В ЛОГ
}
}

```

Рисунок 3.31 – Код модулю zaprosOFF_station.js

– Даний модуль певний час виконує запит щодо перевірки статусу зарядки, перевіряє доступність до з'єднання з системою зарядного пристрою, а саме доступність маршрутизатору, контролера. Якщо виникає ситуація що зарядна станція недоступна зміню статус зарядки на аварія, а також намагається певний проміжок часу провести перепідключення до зарядної станції, якщо виключення можливо тимчасове, наприклад як: відключення електроенергії, збій в інтернет-мережі. Результат роботи модулю проілюстровано на рисунках 3.32, 3.33.

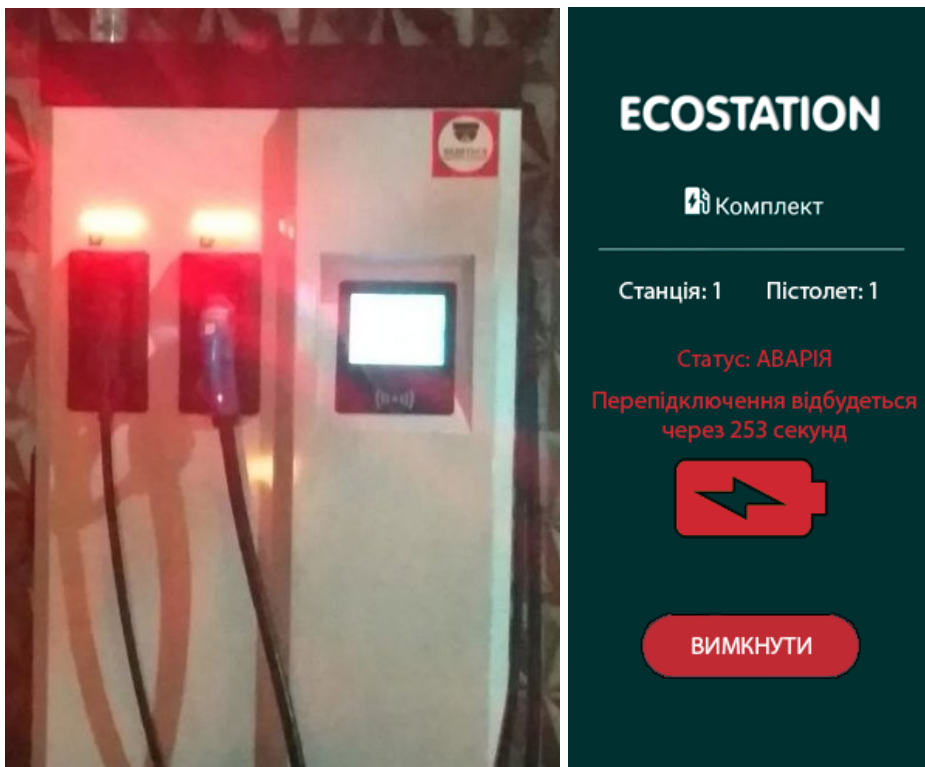


Рисунок 3.32, 3.33 – Відображення статусу «Аварія» на зарядці та в додатку

Модуль `time_station.js` – Відповідає за моніторинг балансу під час процесу зарядки автомобіля, якщо з певним часом у користувача закінчились гроші на балансі, модуль відповідає за від'єднання автомобіля від зарядки, а саме перериває процес заряду.

```

async function time_bablo(date1, date2, idst, idclient) {

  if (idst !== ""){
    const data_q_history = " idstation, SUM(time) AS st_time, SUM(money) AS st_money ";

    return new Promise((resolve) => {
      pool.query('SELECT ' + [data_q_history] + ' FROM history_client WHERE idstation = "' + [idst] + '" ' +
        'AND DATE(date) BETWEEN "' + [date1] + '" AND "' + [date2] + '" ',
        (error, result) => {
          if (error) throw error;
          resolve(result);
        });
    });
  }else{
    const data_q_history = " idclient, SUM(time) AS cl_time, SUM(money) AS cl_money ";

    return new Promise((resolve) => {
      pool.query('SELECT ' + [data_q_history] + ' FROM history_client WHERE idclient = "' + [idclient] + '" ' +
        'AND DATE(date) BETWEEN "' + [date1] + '" AND "' + [date2] + '" ',
        (error, result) => {
          if (error) throw error;
          resolve(result);
          console.log(result);
        });
    });
  }
}

//***** // ЗАПРОС НА СТАНЦИИ TURBO

```

Рисунок 3.34 – Код модулю `time_station.js`

```

async function time_TURBO(date1, date2, idst, idclient) {

  let data_steve = [];

  return new Promise ((resolve) => {

    pool_steve.query('SELECT * FROM transaction_start WHERE event_timestamp BETWEEN "' +
      [date1] + ' 00:00:00" AND "' + [date2] + ' 23:59:59" ',
      (error, result) => {
        if (error) throw error;

        let ii=0;
        for (let i = 0; i < result.length; i++) {
          //for (const item of result) {

            pool_steve.query('SELECT stop_timestamp FROM transaction_stop WHERE transaction_pk = "' +
              [result[i].transaction_pk] + '" ',
              (error, res) => {
                if (error) throw error;
                pool_steve.query('SELECT ocpp_tag_pk FROM ocpp_tag WHERE id_tag = "' +
                  [result[i].id_tag] + '" ',
                  (error, res_tag) => {
                    if (error) throw error;
                    pool_steve.query('SELECT first_name FROM user WHERE ocpp_tag_pk = "' +
                      [res_tag[0].ocpp_tag_pk] + '" ',
                      (error, res_name) => {
                        if (error) throw error;
                        pool_steve.query('SELECT * FROM transaction_start WHERE event_timestamp BETWEEN "' +
                          [dateFormat(result[i].start_timestamp, "yyyy-mm-dd")] + ' 00:00:00" AND "' +
                          [dateFormat(result[i].start_timestamp, "yyyy-mm-dd")] + ' 23:59:59" AND id_tag = "' +
                          [result[i].id_tag] + '" ',
                          (error, res_count) => {

```

Рисунок 3.35 – Код модулю `time_station.js`

В результаті роботи модулю, в мобільному додатку є сторінка «Баланс» в якому користувач може переглянути свої останні сеанси зарядки, побачити час та скільки було використано коштів з його баланс, який ще баланс зберігається.

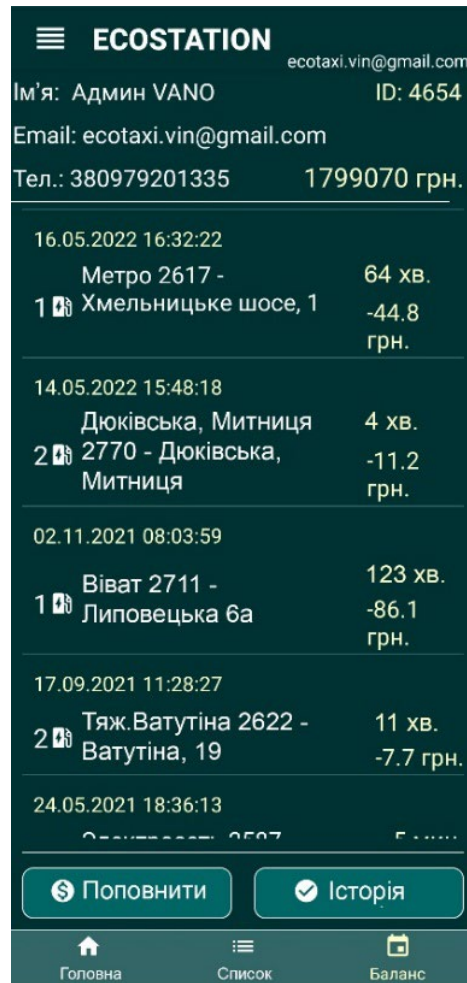


Рисунок 3.36 – Вікно «Баланс» у мобільному додатку, який працює завдяки модулю `time_station.js`

Розробка методу автоматизації роботи інформаційної системи електрозарядної мережі включає в себе впровадження технічних та програмних рішень, спрямованих на підвищення ефективності взаємодії між користувачами та зарядними станціями. Розглянемо переваги та недоліки такого методу.

Переваги:

- Підвищення ефективності обслуговування: Автоматизація дозволяє зменшити час на обслуговування користувачів, а також збільшити кількість зарядок, що можуть бути оброблені одночасно.
- Зменшення помилок: Ручне введення даних часто призводить до помилок, тоді як автоматизована система може значно знизити їх кількість завдяки точному програмному контролю.
- Підвищення задоволеності користувачів: Зручний інтерфейс і швидка відповідь системи на запити користувачів збільшують їх загальне задоволення послугами.
- Доступність та мобільність: Мобільний додаток та веб-інтерфейс дозволяють користувачам управляти процесом зарядки в будь-який час з будь-якого місця.
- Аналітика та звітність: Система збору та аналізу даних дозволяє вести детальну статистику використання станцій, що може бути використано для оптимізації роботи мережі.
- Універсальність: головною розробкою та новизною проекту є його адаптивність під любий тип інтерфейсу зарядного пристрої, його універсальність досягається за рахунок методу комунікації API-запитів, а також апаратної частини яка дає можливість працювати з безліч пристроями.

Недоліки:

- Висока початкова вартість: Розробка, тестування та впровадження комплексної автоматизованої системи можуть бути коштовними.
- Технічне обслуговування: Автоматизовані системи потребують регулярного оновлення та технічного обслуговування, що також може потребувати додаткових витрат.

- Комплексність впровадження: Інтеграція нової системи з існуючим обладнанням та програмним забезпеченням може бути складною.
- Залежність від технологій: Збої в програмному забезпеченні або апаратні несправності можуть призвести до простою всієї системи.
- Опір змінам: Персонал та користувачі можуть проявляти опір при переході на нову систему, якщо не будуть належно проінструктовані.

Розроблений метод є спробою гармонізувати потреби користувачів та технічні можливості сучасних електрзарядних станцій, щоб створити сталу та ефективну інфраструктуру для власників електромобілів. Важливо забезпечити ретельне планування, адекватний бюджет та якісну реалізацію, щоб мінімізувати недоліки та максимізувати переваги введення такої системи.

Розробка було виконана на замовлення для підприємства ТОВ «ЕкоТаксі-Вінниця», а саме інформаційна система для автоматизації роботи користувачів та процесів, комунікація між електрзарядною станцією та серверно-апаратною частиною. Проект був реалізовано та втілено у структуру підприємства та пропрацював рік, під час цього року було розроблено саме метод автоматизацію роботи користувачів з інформаційною системи.

В результаті роботи інформаційної системи та методу, можна зробити оцінку виконаній роботі та зробити звіт за рік роботи системи в реаліях підприємства, який проілюстровано на рисункові 3.37, згідно цих даних можна отримати такі показники:

Звіт по станціям з 01.01.2023 по 13.12.2023

- ID -	Адреса	Ім'я	Спожито е-гії, кВт	Кількість сеансів	Тип	Час заряду	Сума, грн
77	Київська, 1366	1.1 Комплект інвест 2581	10266,06	587	low	1466.58 годин	61596 грн.
107	Київська, 1366	1.2 Комплект інвест 2667	6653,99	380	low	950.57 годин	39923 грн.
108	Київська, 1366	2.1 Комплект інвест 2668	8203,86	469	low	1171.98 годин	49223 грн.
109	Київська, 1366	2.2 Комплект інвест 2669	2654,54	152	low	379.22 годин	15927 грн.
81	600-річчя, 17	1.1 Мегамоп 2591	19636,40	1122	low	2805.20 годин	117818 грн.
80	600-річчя, 17	1.2 Мегамоп 2590	18313,61	1046	low	2616.23 годин	109881 грн.
93	600-річчя, 17	2.1 Мегамоп 2592	17790,71	1017	low	2541.53 годин	106744 грн.
94	600-річчя, 17	2.2 Мегамоп 2593	20822,41	1190	low	2974.63 годин	124934 грн.
84	Стрілецька, 2	1.1 ПивЗавод 2594	15585,29	891	low	2226.47 годин	93511 грн.
85	Стрілецька, 2	1.2 ПивЗавод 2595	18912,95	1081	low	2701.85 годин	113477 грн.
90	Дюківська, Таможня	1.1 Дюківська, Таможня 2599	7702,45	440	low	1100.35 годин	46214 грн.
91	Дюківська, Таможня	1.2 Дюківська, Таможня 2600	0,00	0	low	0.00 годин	0 грн.
129	Дюківська, Таможня	2.1 Дюківська, Таможня 2770	8837,71	505	low	1262.53 годин	53026 грн.
130	Дюківська, Таможня	2.2 Дюківська, Таможня 2771	12294,10	703	low	1756.30 годин	73764 грн.
92	Дюківська, Таможня	3.1 Дюківська, Таможня 2600	0,00	0	high	0.00 годин	0 грн.
128	Дюківська, Таможня	4.1 Дюківська, Таможня 2775	0,00	0	high	0.00 годин	0 грн.
95	Салтикова щедрина, 112	1.1 Екоенерджіпарк 2605	10389,05	594	low	1484.15 годин	62334 грн.
96	Салтикова щедрина, 112	1.2 Екоенерджіпарк 2606	6837,81	391	low	976.83 годин	41026 грн.
98	Чехова, 23	1.1 Грош 2612	12525,80	716	low	1789.40 годин	75154 грн.
97	Чехова, 23	1.2 Грош 2611	12979,96	742	low	1854.28 годин	77879 грн.
99	Хмельницьке шосе, 1	1.1 Метро 2617	3197,80	183	low	456.80 годин	19185 грн.
100	Хмельницьке шосе, 1	1.2 Метро 2618	2535,75	145	low	362.25 годин	15214 грн.
105	Хмельницьке шосе, 1	2.1 Метро 2625	0,00	0	low	0.00 годин	0 грн.
106	Хмельницьке шосе, 1	2.2 Метро 2626	5790,61	331	low	827.23 годин	34743 грн.
101	Ватутина, 19	1.1 Тяж.Ватутина 2619	6020,14	344	low	860.02 годин	36120 грн.
102	Ватутина, 19	1.2 Тяж.Ватутина 2620	5535,04	316	low	790.72 годин	33210 грн.
103	Ватутина, 19	2.1 Тяж.Ватутина 2621	5536,86	316	low	790.98 годин	33221 грн.
104	Ватутина, 19	2.2 Тяж.Ватутина 2622	6203,75	355	low	886.25 годин	37222 грн.
110	Ватутина, 18а	1.1 Шансон 2743	16014,39	915	low	2287.77 годин	96086 грн.
111	Ватутина, 18а	1.2 Шансон 2744	9377,55	536	low	1339.65 годин	56265 грн.
112	Шепеля 1	1.1 Новая почта 1 2682	18309,06	1046	low	2615.58 годин	109854 грн.

Рисунок 3.37 – Звіт по зарядним станціям за період 2023 року

- Список зарядних станцій – їх назва, адреса та внутрішній ID;
- Спожита електроенергія – кількість спожитої електроенергії електрозарядним пристроєм для процесу заряджання електромобіля;
- Кількість сеансів – кількість сеансів під час яких автомобіль заряджався певний проміжок часу, за який можна стверджувати про вдалий процес заряджання;
- Тип – тип електрозарядної станції, а саме постійного струму або перемінного струму, або як кажуть простіше - швидка та звичайна зарядка;
- Час заряду – час перебування автомобілем на зарядці під час процесу зарядки;
- Сума – сума заробітку за зарядку станцію, в результаті якої потреби зарядки вже включені, а саме закупівля електроенергії.

В результаті, наведених даних можна зробити висновок, що за 2023 рік, цей список зарядних станцій отримав прибуток, не враховуючи податкові обов'язки - 1733551,00 грн, було спожито 288927,45 кВт активної електроенергії, на електростанціях було зафіксовано 16510 сеансів заряду та проведено 41275,35 годин. За цими даними можна дійти висновку, що проект був доцільно реалізований на підприємстві та вже показує практичні результати.

3.4 Висновки

В даному розділі було розроблено архітектору інформаційної системи, представлено блок-схеми архітектури та структури та use case діаграми інформаційної системи. Також було розроблено алгоритм методу автоматизації процесів роботи користувачів інформаційної системи електрзарядних пристроїв та побудовано блок-схему алгоритму, далі було реалізовано метод автоматизацію процесів обслуговування користувачів, описано його створення з апаратною частини до створення програмно. Продемонстровано роботу інформаційної системи в цілому

ВИСНОВКИ

В ході виконання магістерської дипломної роботи було проведено розгляді світу електрзарядних станцій. На початку, у першому розділі, ми зосередилися на всебічному аналізі електрзарядних станцій, їх ролі у сучасному світі електромобілів. Ми дослідили різні технології та протоколи, що використовуються у цих системах, а також оцінили переваги та недоліки електромобілів та зарядних станцій. Це дало нам можливість зрозуміти поточний ландшафт та виклики, з якими стикаються користувачі та розробники. Крім того, ми провели огляд існуючих інформаційних систем, які вже функціонують на ринку, щоб виявити їхні сильні та слабкі сторони та на базі їх наробок, виконати свою реалізацію.

У другому розділі ми перейшли до вибору оптимальних інформаційних технологій для нашого проекту. Тут ми глибоко занурилися в аналіз JavaScript як мови програмування, вивчили можливості фреймворку React Native для розробки мобільних додатків та розглянули можливості програмної платформи Node JS. Також ми вивчили потенціал розумних маршрутизаторів MikroTik та Ethernet-контролерів VK Module, оцінюючи, як ці технології можуть сприяти нашому проекту.

Завершальний третій розділ був присвячений розробці методу автоматизації процесів обслуговування користувачів мережі EcoStation. Ми створили комплексну архітектуру інформаційної системи, розробили алгоритм автоматизації та продемонструвати реалізацію мобільного додатку та веб-порталу, як частини інформаційної системи. Це дозволило нам не лише створити ефективну систему, але й продемонструвати її практичну застосовність у реальному світі.

Дана розробка була розроблена та реалізовано по замовленню підприємства ТОВ «ЕкоТаксі-Вінниця», даний проект проіснував та існує на підприємстві більше року та продемонстрував себе, як потужною інформаційної системою яка здатна виконувати поставлені перед нею задачею, стабільно працювати та при необхідності додавати новий функціонал у систему. За результатами річної звітності, було проведено аналіз щодо рентабельності системи та підприємство дійшло до висновків

що розробка є економічна доцільною та навіть може приносити заробіток з тих напрямків, в які не входили поставлені задачі підприємством.

Загалом, це дослідження не лише принесло нові знання у сферу електрозарядних станцій, але й запропонувало практичні рішення для поліпшення взаємодії між користувачами та цими системами."

За результатами даної роботи була зроблена доповідь на тему «Інформаційна система для автоматизації роботи користувачів мережі електрозарядних станцій EcoStation» на LI Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації Вінницького національного технічного університету (2022) з публікацією тез, планується також доповідь та публікація тез на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)»

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Недоля І.Р. Жуков С.О., Інформаційна система для автоматизації роботи користувачів мережі електрозарядних станцій EcoStation. *Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації*: зб. Матеріалів конференції. Вінниця, 2022. [Електронний ресурс]. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15787>
2. Недоля І.Р., Бісікало О.В., Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION. Молодь в науці: дослідження, проблеми, перспективи (МН-2024)» <https://conferences.vntu.edu.ua/index.php/mn/mn2024>.
3. Larry Ericson, Solar Powered Charging Infrastructure for Electric Vehicles, Larry Ericson, USA, Washington: CRC PRESS, 2019. – 168 с. Thompson, G., & Bhatt, A. (2018). Electric Vehicle Charging Stations: Market Analysis and Trends. *Renewable Energy Focus*, 29, 36-42.
4. Johnson, M. L. (2019). Challenges and Solutions in Electric Vehicle Charging Infrastructure. *Transport Research Part D*, 71, 22-35.
5. Smith, P., & Chen, L. (2020). The Evolving Landscape of EV Charging Technology. *Journal of Power Sources*, 450, 227642.
6. Electric Vehicle Charging Network Development. [Електронний ресурс]. URL: <https://www.evchargingnetwork.com/analysis>
7. Davis, J., & Kumar, R. (2018). A Comparative Analysis of EV Charging Protocols. *IEEE Transactions on Sustainable Energy*, 9(4), 1806-1819.
8. Patel, H. S., & Gupta, N. K. (2021). Modern Technologies in EV Charging Systems: A Review. *Electric Power Systems Research*, 191, 106821.
9. Zhang, W., & Li, X. (2019). The Development and Implementation of Charging Communication Protocols for Electric Vehicles. *Energy Procedia*, 158, 4134-4139.
10. Overview of Electric Vehicle Charging Technologies. [Електронний ресурс]. URL: <https://www.electric-vehicles-technology.com/charging-protocols>

11. Greene, S., & Hopkinson, J. (2020). Benefits and Drawbacks of Electric Vehicles: An Environmental Perspective. *Environmental Science & Technology*, 54(8), 4634-4642.
12. Morris, D. E., & Sharma, A. (2019). Electric Cars: Pros, Cons, and the Future of the Automobile Industry. *Journal of Environmental Management*, 244, 374-383.
13. Fischer, A., & Baumgartner, R. J. (2021). Assessing the Ecological Impact of Electric Vehicle Charging Stations. *Journal of Cleaner Production*, 279, 123456.
14. Electric Vehicles: Pros and Cons. [Електронний ресурс]. URL: <https://www.electricvehicleprosandcons.org>
15. Brooks, A., & Simmons, G. (2018). Analysis of Smart Charging Systems for Electric Vehicles. *Journal of Power and Energy Engineering*, 6(5), 11-25.
16. Advantech, огляд та рішення електрозарядного пристрою [Електронний ресурс]. URL: <https://www.advantech.com/en/resources/case-study/ev-charging-data-management>
17. Liu, Y., & Wang, Z. (2020). Smart Charging Systems for Electric Vehicles: A Survey. *Energy Reports*, 6, 210-225.
18. Kumar, P., & Singh, R. K. (2019). Review of Information Systems for Electric Vehicle Charging Networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(12), 4423-4437.
19. Current EV Charging Station Information Systems. [Електронний ресурс]. URL: <https://www.evchargingsolutions.com/information-systems>
20. Advantech огляд рішення, та структури електрозарядної мережі <https://www.advantech.com/en-us/resources/case-study/ev-charging-solution-network-management>
21. JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native, and Electron 1st Edition, Kindle Edition by Adam D. Scott Format: Kindle Edition, 2019, p.46 – 52
22. Miller, M. J. JavaScript: Comprehensive Guide. New York: Springer, 2021, pp. 45-67.

23. Brown, K. L. Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript. Addison-Wesley, 2019, pp. 32-58.
24. Flanagan, D. JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, 2020, pp. 71-95.
25. JavaScript, Навчальний посібник [Електронний ресурс]. URL: <https://www.javascript.com/resources>
26. McKenzie, G. React Native Development. Packt Publishing, 2018, pp. 20-40.
27. Hartington, M. Mastering React Native. Wiley, 2020, pp. 60-85.
28. Newman, R. Building Mobile Apps with React Native. Manning Publications, 2019, pp. 45-70.
29. React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications 2nd ed. Edition by Akshat Paul, Abhishek Nalwaya, 2020, p. 25 - 36
30. React Native, Навчальний посібник, [Електронний ресурс]. URL: <https://reactnative.dev/docs/getting-started>
31. Cantelon, M., & Harter, M. Node.js in Action, Second Edition. Manning Publications, 2021, pp. 35-60.
32. Hughes-Croucher, T., & Wilson, M. Up and Running with Node.js. O'Reilly Media, 2019, pp. 50-75.
33. Wandschneider, K. Learning Node.js Development. Packt Publishing, 2020, pp. 40-65.
34. Node JS, Навчальний посібник[Електронний ресурс]. URL: <https://nodejs.org/en/docs>
35. Smith, J. MikroTik RouterOS Guide. Independent, 2020, pp. 15-38.
36. Turner, L. The Comprehensive Guide to MikroTik RouterOS Configuration. IT Guru Publications, 2018, pp. 22-47.
37. Davis, R. MikroTik RouterOS: Advanced Setup Guide. Tech Press, 2019, pp. 30-55.
38. Розумний маршрутизатор MikroTik, Навчальний посібник [Електронний ресурс]. URL: <https://www.mikrotik.com/documentation>

39. VK Module, технічна документація та навчальний посібник [Електронний ресурс]. URL: https://vkmodule.com.ua/Ethernet/Socket_2W.html
40. Green, A. Ethernet Controllers: A Comprehensive Guide. TechMedia, 2021, pp. 67-89.
41. Foster, B. Networking with Ethernet Controllers. Network World Publishing, 2018, pp. 12-34.
42. Patel, S. VK Module and Ethernet Technology. Electronic Solutions Press, 2020, pp. 45-68.

ДОДАТКИ

Додаток А
(обов'язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ПОГОДЖЕНО

Керівник підприємства
ТОВ "ЕКОТАКСІ-ВІННИЦЯ"

(підпис) (Ім'я, ПРІЗВИЩЕ)
« 11 » жовтня 2023 р.

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСКАЛО

« 12 » жовтня 2023 року

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
МЕТОД АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ОБСЛУГОВУВАННЯ КОРИСТУВАЧІВ
МЕРЕЖІ ЕЛЕКТРОЗАРЯДНИХ СТАНЦІЙ ECOSTATION
08-31.МКР.010.02.000 ТЗ

Керівник: Зав. кафедри АІТ, д.т.н., проф

Олег БІСКАЛО

« 12 » жовтня 2023 р.

Розробив студент гр. ЗАКІТ-22м

Іван НЕДОЛЯ

« 12 » жовтня 2023 р.

1. Назва та галузь застосування.

Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION. Інформаційні системи та технології.

2. Підстава для проведення робіт.

Підставою для виконання роботи є наказ № 247 по ВНТУ від «18» вересня 2023р., та індивідуальне завдання на МКР, затверджене протоколом № 1 засідання кафедри АПТ від «30» серпня 2023р.

3. Мета та призначення роботи.

Метою дослідження є розробка методу автоматизації процесів інформаційної системи обслуговування користувачів мережі електрозарядних пристроїв.

4. Джерела розробки:

1) Недоля І.Р., Бісікало О.В., Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION. Молодь в науці: дослідження, проблеми, перспективи (МН-2024)» <https://conferences.vntu.edu.ua/index.php/mn/mn2024>.

2) Недоля І.Р. Жуков С.О., Інформаційна система для автоматизації роботи користувачів мережі електрозарядних станцій EcoStation. Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації: зб. Матеріалів конференції. Вінниця, 2022. [Електронний ресурс]. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15787>

5. Показники призначення

Основні технічні вимоги до розробки:

5.1. Вимоги до програмної платформи:

- Microsoft SQL server, з мінімальним простором на диску у 10 гб, 4 гб оперативною пам'яті та чотири обчислюваних ядра;
- Node JS server;

5.2. Вимоги до апаратної частини:

- Router MikroTik, мінімально 10 мбіт/с пропускна здатність, 512 мбайт оперативної пам'яті та два обчислюваних ядра ;
- Контролер VK Module Socket 2W, WiFi з мінімальною пропускною здатністю в 10 мбіт/с;
- Інтернет-модем, підтримуючий 2G, 3G, LTE мережі, з мінімальною пропускною здатністю у 10 мбіт/с.

Вихідні дані для проведення робіт:

База даних MySQL підключена до серверної частини Node JS, платформа React Native в середовищі розробки WebStorm, конфігурація розумного маршрутизатора MikroTik, конфігурація контролеру зарядної станції VK Module Socket-2W

Методи дослідження:

Створення методу автоматизації інформаційної системи з використанням платформ React Native та Node JS, завдяки комунікації API-запитами, мовою програмування JavaScript.

Результати розробки : Розроблена інформаційна система по автоматизації роботи користувачів мережі електрозарядних пристроїв ECOSTATION та універсальний метод комунікації та автоматизації процесів інформаційної системи.

6. Стадії розробки:

- | | | | |
|---|-------|---|-------|
| а) Аналіз предметної області | _____ | — | _____ |
| б) Вибір оптимальних інформаційних технологій | _____ | — | _____ |
| в) Розробка методу автоматизації процесів обслуговування користувачів | _____ | — | _____ |
| г) Оформлення матеріалів до захисту МКР | _____ | — | _____ |
| г) Попередній захист роботи | _____ | — | _____ |
| д) Остаточний захист роботи | _____ | — | _____ |

8. Порядок контролю та приймання

Рубіжний контроль провести до « 1 » листопада 2023 р.

Попередній захист МКР провести « 20 » листопада 2023 р.

Захист МКР провести до « 18 » грудня 2023 р.

Розробив студент групи ЗАКІТ-22м _____ Іван НЕДОЛЯ

Додаток Б
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

**Метод автоматизації процесів обслуговування користувачів мережі
електрозарядних станцій ECOSTATION**

Вінниця 2023

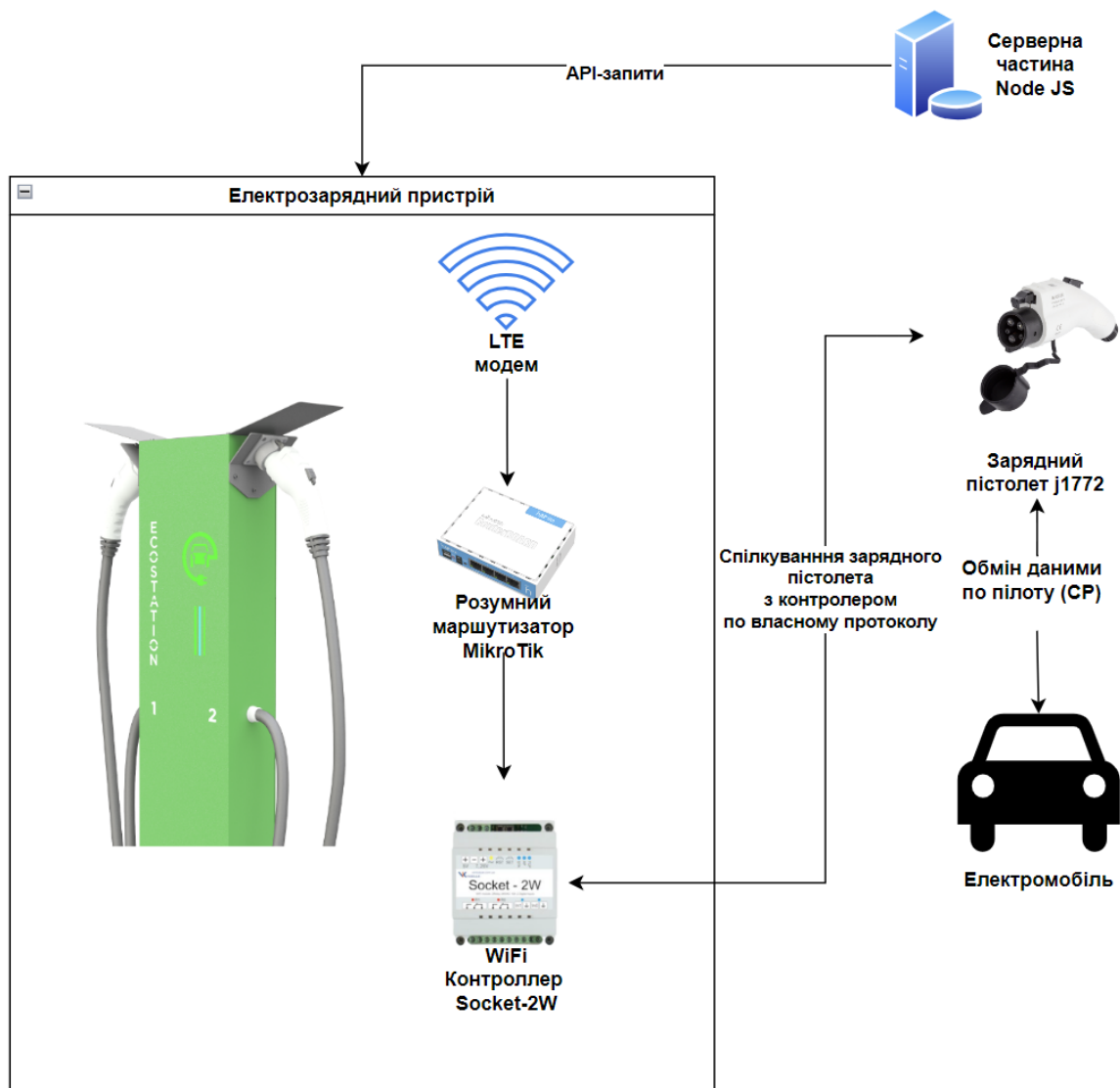


Рисунок Б.1 – Блок схема взаємозв'язків апаратної частини зі серверною



Рисунок Б.2 – Діаграма Use Case методу комунікації та обміну даних інформаційної СИСТЕМИ

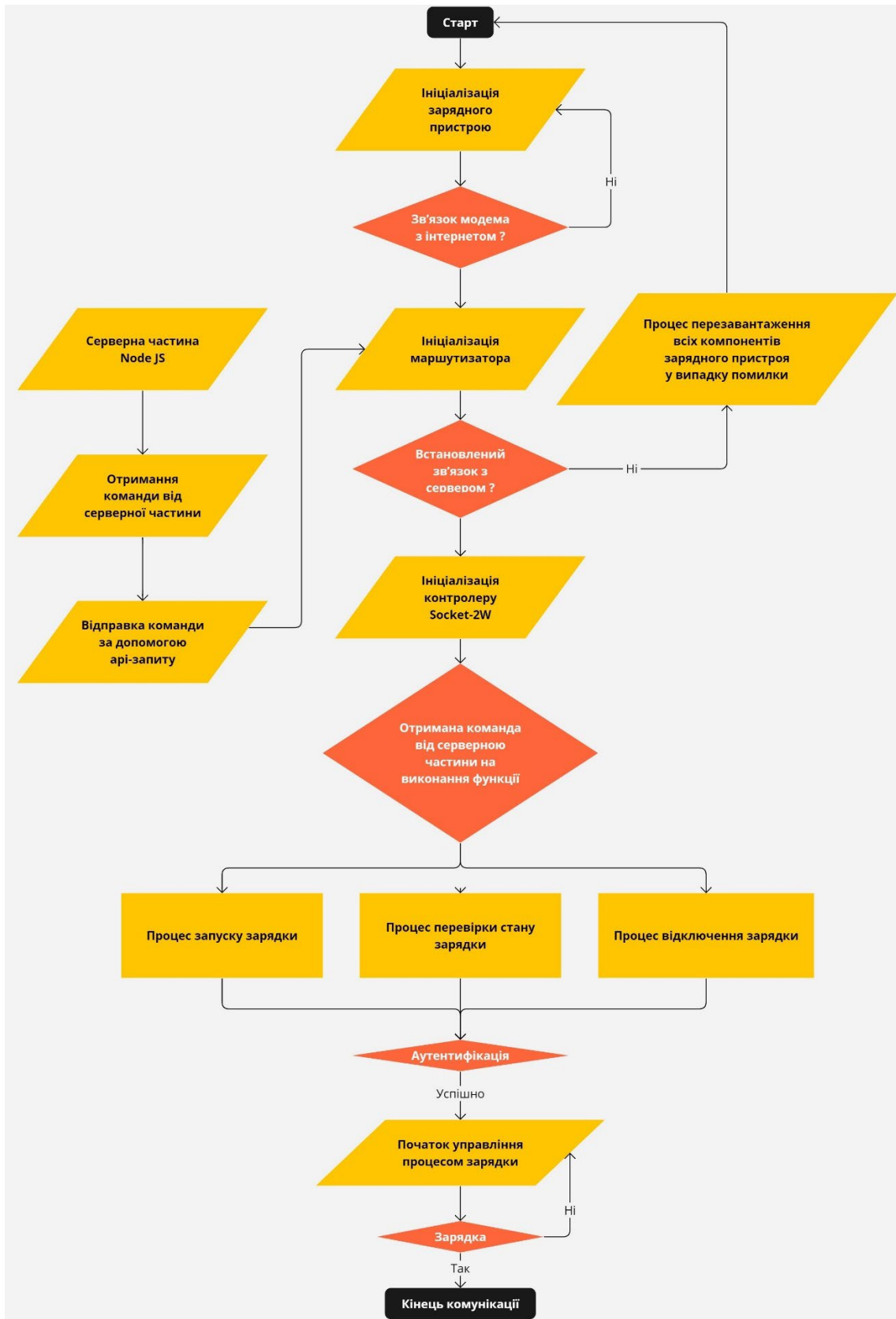


Рисунок Б.3 – Блок-схема алгоритму роботи методу комунікації



Рисунок Б.4 – Діаграма Use Case модулю реєстрації та автентифікації

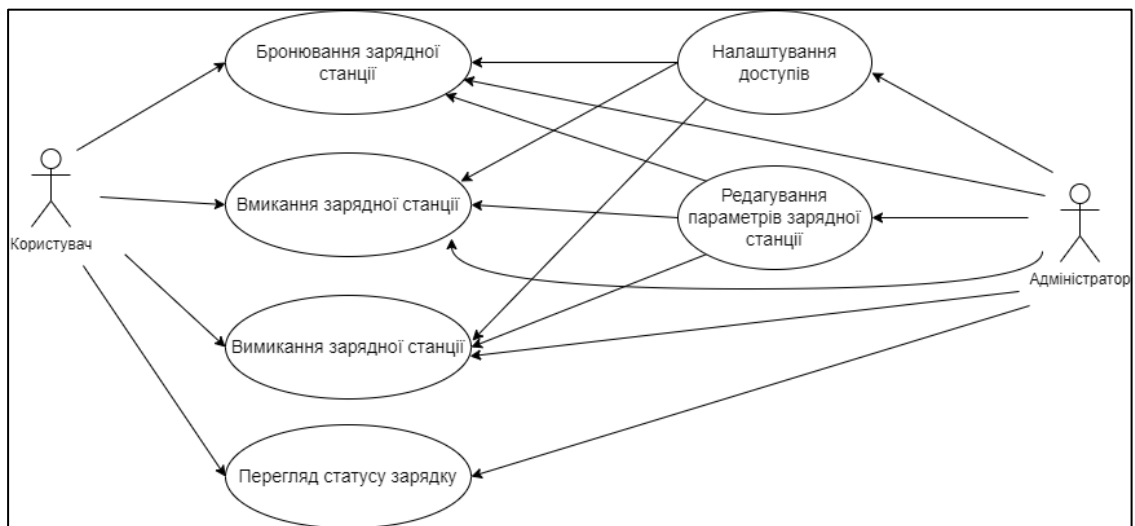


Рисунок Б.5 – Діаграма Use Case інформаційної системи

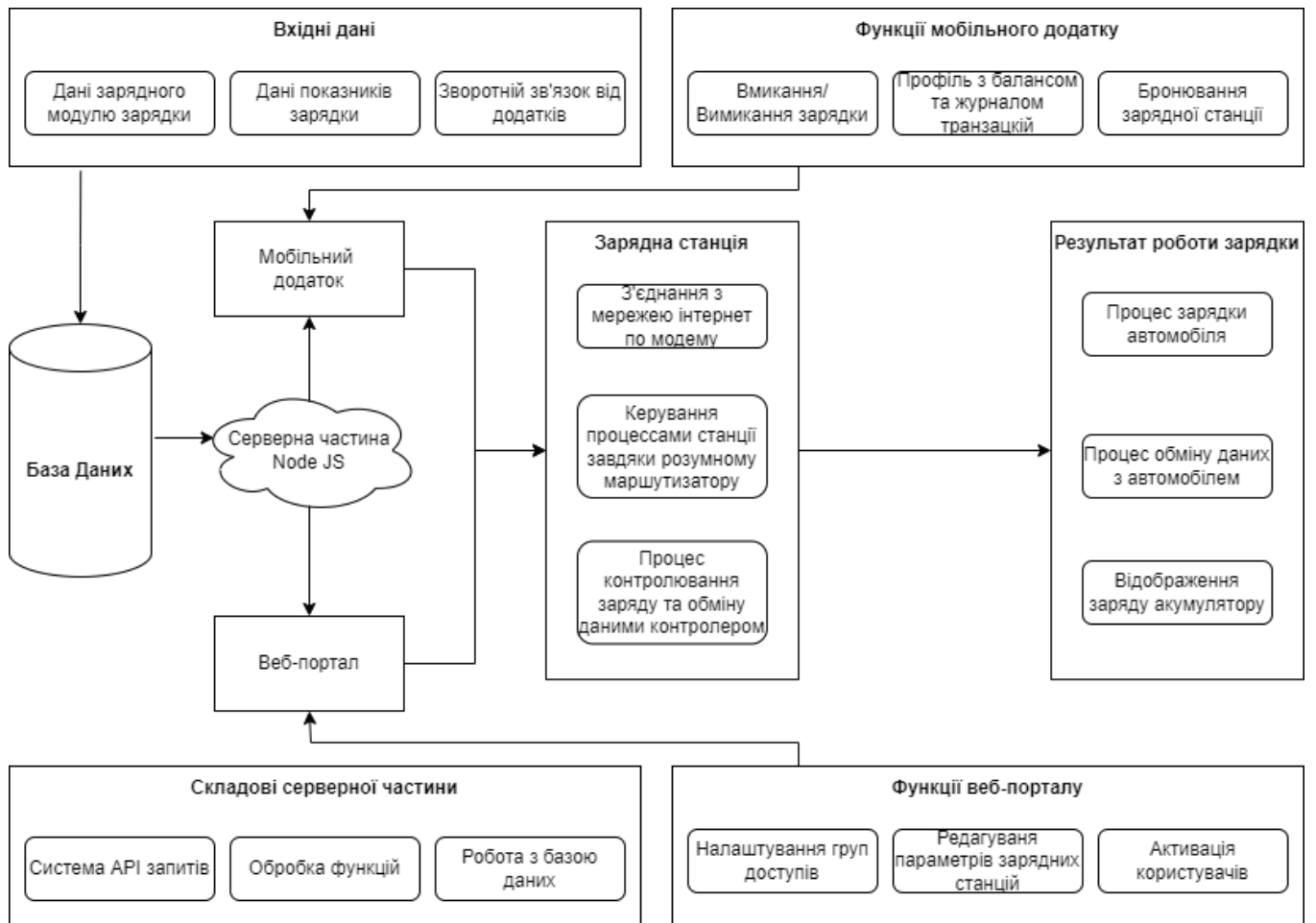


Рисунок Б.6 – Блок схема роботи програмного забезпечення інформаційної системи

Додаток В
(обов'язковий)

Фрагмент лістингу програми

Код файлу додатку **StatusStUserZaryad.js**

```
function stations_request(id, status, url, groupuser) {
    return new Promise( (resolve) => {
        request.get({ // ЗАПРОС К СТАНЦИИ НА ОТКЛИК
            url: url,

        }, (err, response, body) => {
            if (err) {
                zaprosOFF_station(id, status); // ОШИБКА ЗАПРОСА -
                ЗАПРАВКА НЕДОСТУПНА
                resolve ("ERROR");
            }else {
                parseString(body, function (err, result) {
                    on_off_station(id, status, result.inputs.LED[0],
groupuser); // возврат Данных ОК
                    resolve ("OK");
                });
            }
        });
    });
}

//*****
*****
```

ON

```

function on_off_station(id, status, switch_st, groupuser){ //
ЗАПРАВКА ON - OFF
    //console.log(data.inputs.switch);

    if (switch_st === "on"){status_update_station(id, status, groupuser);}
// ЗАРАДКА
    if (switch_st === "off"){status_update_station(id, status,
"off"+groupuser);
        insert_hystory_client(groupuser, id, groupuser);
    } // АКТИВНА

    console.log("portal: "+ id + " заправка - " + switch_st); // ЗДЕСЬ БУДЕМ
ПИСАТЬ В ЛОГ
    }
    //***** OFF
*****

function zaprosOFF_station(id, status_old) { // ЗАПРАВКА
НЕ ДОСТУПНА!
    pool.query("SELECT status FROM stations WHERE id=?", [id], function
(err, data) {
        if (err) return console.log(err);
        if (data[0].status !== 0) {
            status_update_station(id, 0);
        }// ЗАПРАВКА НЕ ДОСТУПНА
    });

    console.log(id + " " + status_old + " - OFF"); // ЗДЕСЬ БУДЕМ ПИСАТЬ В
ЛОГ
}

```

```
//-----
```

```
-
function insert_hystory_client(idclient, idstation, groupid){ // ДОБАВЛЯЕМ
ЗАПИСЬ В ИСТОРИЮ КЛИЕНТА!
```

```
pool.query('SELECT * FROM jurnal WHERE usid = ? AND statusonoff =
"on" ORDER BY id DESC LIMIT 1',
```

```
[idclient],(error, result) => {
```

```
if (error) throw error;
```

```
if (result.length !== 0) {
```

```
let nday = dateFormat(new Date(), "yyyy-mm-dd
```

```
HH:MM:ss");
```

```
let d1 = moment(new Date());
```

```
let d2 = moment(Date.parse(result[0].date));
```

```
let time = d1.diff(d2, 'minutes'); // 44700
```

```
//console.log(a.diff(b, 'hours')); // 745
```

```
//console.log(a.diff(b, 'days')); // 31
```

```
//console.log(a.diff(b, 'weeks')); // 4
```

```
pool.query("INSERT INTO history_client (date,
idclient, groupid, idstation, time) VALUES (?, ?, ?, ?, ?)",
```

```
[nday, idclient, groupid, idstation, time],
```

```
function (err, data) {
```

```
if (err) return console.log(err);
```

```
console.log("Add hystory_client:" +
idclient + " " + idstation + " " + time + "МИН. - " + nday);
```

```
});
```

```
}
```

```

    });
}
//*****ЖУРНАЛ---
-----
function insert_on_off_bron(nday, imei, groupid, nstation, statust, power){ //
ДОБАВЛЯЕМ ЗАПИСЬ В ЖУРНАЛ!
    console.log(nday, imei, groupid, nstation, statust, power);
    pool.query("INSERT INTO jurnal (date, usid, groupid, stationid, statusonoff,
power) VALUES (?, ?, ?, ?, ?, ?)",
        [nday, imei, groupid, nstation, statust, power], function (err, data) {
            if (err) return console.log(err);
            //console.log(groupid);
            console.log("Add jurnal imei:" + nstation + " " + imei + " " +
statust + " " + nday);
        });
}
//*****
*****
////////////////////////////////////
//*****
*****

```

Код файлу додатку **Stations_Use.js**

```

const OnOffST = (id, imei, swst, erroroff) => new Promise((resolve, reject) => {
    let details = {
        akeycsec: '*****', imei: imei, swst: swst, // on-off
        nstation: id, // ID станції
        erroroff: erroroff,
    };
});

```

```

let formBody = [];
for (let property in details) {
  let encodedKey = encodeURIComponent(property);
  let encodedValue = encodeURIComponent(details[property]);
  formBody.push(encodedKey + '=' + encodedValue);
}
formBody = formBody.join('&');

fetch(input: 'https://station.ecogmb.com:****/*****', {
  method: 'POST',
  headers: {
    Authorization: 'Bearer token',
    'Content-Type': 'application/x-www-form-urlencoded',
  },
  body: formBody,
})
.then(response => response.text())
.then(result => {
  resolve(result);
})
.catch(error => {
  console.error(error);
  resolve({ value: 'ERROR' });
});
});

// STATIONS ON-OFF
app.get('/sttat/:id', isLoggedIn, async function(req, res) {
  const num_start = req.query.num;
  const id = req.params.id;
  const sw = req.query.sw;
  let power = "";
  const dateFromNow = (new Date(), "yyyy-mm-dd HH:mm:ss");
  console.log(sw);
  const data_station = await station_sql('id'); //получаем данные о запрашке с базы
  let status = data_station[0].status;
  if (data_station[0].groupid === req.user.groupid || req.user.groupid === "admin") { //
Проверяем группу
  if (status !== '0') {
    if (sw === "ON") {

```



```

    let url = data_station[0].logpass + data_station[0].ip + ":" + data_station[0].port
+
    data_station[0].checkst + data_station[0].onst;
    await stations_request(id, 4, url, req.user.groupid);
    if (sw === "OFF") {
        let url = data_station[0].logpass + data_station[0].ip + ":" +
data_station[0].port +
        data_station[0].checkst + data_station[0].offst;
        await stations_request(id, 1, url, req.user.groupid);
    }
    insert_on_off('brnOn', req.user.groupid, req.user.groupid, id, sw, power);
//ЖУРНАЛ.
    res.redirect('/c/stations_us?num=' + num_start);
  } else {
    res.redirect('/c/stations_us?num=' + num_start);
  }
} else {
  res.redirect('/');
}
});
});

```

Код файла добавки **zaprosoff_station.js**

```

// ***** OFF ЗАПРАВКА НЕ ДОСТУПНА!
function zaprosoff_station(id, status_old) {
  pool.query("SELECT status FROM stations WHERE id=?", [id], function (err, data) {
    if (err) return console.log(err);
    if (data[0].status !== 0) {
      status_update_station(id, 0);
    } // ЗАПРАВКА НЕ ДОСТУПНА
  });

  console.log(id + " " + status_old + " - OFF"); // ЗДЕСЬ БУДЕМ ПИСАТЬ В ЛОГ
}

```

Код файла добавки **time_station.js**

```

async function time_bablo(date1, date2, idst, idClient) {
  if (idst !== "") {

```

```
const data_q_history = " idstation, SUM(time) AS st_time, SUM(money) AS
st_money ";
```

```
return new Promise((resolve) => {
  pool.query('SELECT ' + [data_q_history] + ' FROM history_client WHERE
idstation = "' + [idst] + '"' +
  ' AND DATE(date) BETWEEN "' + [date1] + '" AND "' + [date2] + '"',
  (error, result) => {
    if (error) throw error;
    resolve(result);
  });
});
} else {
  const data_q_history = " idClient, SUM(time) AS cl_time, SUM(money) AS
cl_money ";
```

```
return new Promise((resolve) => {
  pool.query('SELECT ' + [data_q_history] + ' FROM history_client WHERE
idClient = "' + [idClient] + '"' +
  ' AND DATE(date) BETWEEN "' + [date1] + '" AND "' + [date2] + '"',
  (error, result) => {
    if (error) throw error;
    resolve(result);
    console.log(result);
  });
});
}
}
```

```
// ***** ЗАПРОС НА СТАНЦИИ TURBO
```

```

async function time_TURBO(date1, date2, idst, idClient) {
  let data_steve = [];

  return new Promise ((resolve) => {
    pool_steve.query('SELECT * FROM transaction_start WHERE event_timestamp
BETWEEN "' +
      [date1] + ' 00:00:00" AND "' + [date2] + ' 23:59:59"',
      (error, result) => {
        if (error) throw error;

        let i=0;
        for (let i = 0; i < result.length; i++) {
          //for (const item of result) {
            pool_steve.query('SELECT stop_timestamp FROM transaction_stop WHERE
transaction_pk = "' +
              [result[i].transaction_pk] + '"',
              (error, res) => {
                if (error) throw error;

                pool_steve.query('SELECT ocpp_tag_pk FROM ocpp_tag WHERE id_tag =
"' +
                  [result[i].id_tag] + '"',
                  (error, res_tag) => {
                    if (error) throw error;

                    pool_steve.query('SELECT first_name FROM user WHERE ocpp_tag_pk
= "' +
                      [res_tag[0].ocpp_tag_pk] + '"',
                      (error, res_name) => {
                        if (error) throw error;

                        pool_steve.query('SELECT * FROM transaction_start WHERE
event_timestamp BETWEEN "' +
                          dateFormat(result[i].start_timestamp, "yyyy-mm-dd") + ' 00:00:00"
AND "' +
                            dateFormat(result[i].start_timestamp, "yyyy-mm-dd") + ' 23:59:59"
AND id_tag = "' +
                              [result[i].id_tag] + '"',
                              (error, res_count) => {
                                if (error) throw error;
                                // ...

```

```
    });  
  });  
});  
}  
});  
}
```

Додаток Г
(обов'язковий)

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: « Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION»

Тип роботи: Магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ КСУ, АПТ
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 98.6 % Схожість 1.4 %

Аналіз звіту подібності (відмітити потрібне)

Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку Роман МАСЛІЙ
(підпис) (ім'я, прізвище)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Іван НЕДОЛЯ
(підпис) (ім'я, прізвище)

Керівник роботи Олег БІСІКАЛО
(підпис) (ім'я, прізвище)

Додаток Д
(додатковий)

Акт про впровадження магістерської кваліфікаційної роботи

Товариство з обмеженою відповідальністю «ЕкоТаксі-Вінниця»
Код ЄДРПОУ: 43314436
Індивідуальний податковий номер платника податку на додану вартість: 433144302288
Розрахунковий рахунок:
IBAN: UA77305299000026007006400677
ЖИТОМИРСЬКЕ РУ АТ КБ "ПРИВАТБАНК" МФО: 305299

Юридична адреса: м.Вінниця, вул.Пирогова, 151 Г-Б

Тел. 380677579970
Пошта. Ecotaxi.vin@gmail.com

ЗАТВЕРДЖУЮ
Директор ТОВ ЕкоТаксі-Вінниця
Сидорук В.П.
«08» грудня 2023 р.

АКТ

про впровадження результатів магістерської кваліфікаційної роботи
Недолі Івана Руслановича

«Метод автоматизації процесів обслуговування користувачів мережі електрозарядних станцій ECOSTATION»

Комісія у складі голови IT-відділу - Луговського Олександра Сергійовича та керівника підприємства - директора Сидорук Віри Павлівни розглянула матеріали магістерської кваліфікаційної роботи (МКР) Недолі І.Р. і встановила, що при розробці методу автоматизації процесів обслуговування користувачів електрозарядних станцій перспективним є використання розробленого алгоритму.

Комісія вважає, що розроблене Неделею І.Р. програмне забезпечення є корисним для використання при побудові та оптимізації процесів обслуговування в мережі електрозарядних станцій ECOSTATION, що підтверджує актуальність та практичну цінність роботи.

Окрім того, комісія підтверджує, що у ТОВ "ЕкоТаксі" Вінниця впроваджено основні результати магістерської кваліфікаційної роботи Недолі Івана Руслановича, яка була виконана на замовлення підприємства. Апробацію отриманого програмного забезпечення у ТОВ "ЕкоТаксі" проведено з 27 листопада по 8 грудня 2023 року, суттєвих недоліків не виявлено.

Цей акт не є підставою для фінансових розрахунків між Неделею І.Р. та ТОВ "ЕкоТаксі" Вінниця.

Члени комісії:

Голова IT-відділу

Луговський О.С.

Директор

Сидорук В.П.



Рисунок Д.1 - Акт про впровадження результатів магістерської кваліфікаційної роботи