

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ДОДАТКУ ДЛЯ ЗАРЯДКИ
ЕЛЕКТРОКАРІВ З ВИКОРИСТАННЯМ VUE JS
DEVELOPMENT OF AN APPLICATION FOR
CHARGING ELECTRIC CARS BASED ON VUE JS

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ПЗ-31

Коваль Павло Сергійович


(підпис)

Керівник:

к.т.н., доцент

Повстяна Юлія Славомирівна


(підпис)

Кваліфікаційну роботу

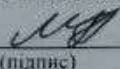
допущено до захисту

« 8 » червне 2023р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

« 02 » 02 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Ковалю Павлу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка додатку для зарядки електрокарів з використанням Vue.js

Керівник роботи: к.т.н., доцент Повстяна Юлія Славомирівна

затверджені наказом закладу вищої освіти від «23» грудня 2022 р. № 961-01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «8» червня 2023 р.

3. Вихідні дані до роботи: програмне забезпечення, вимоги до розробки програмного забезпечення, функціональні та не функціональні вимоги до програмного засобу, Технології програмування клієнтської частини – HTML, SCSS, JavaScript, Vue, Vuetify, Vuex

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів розробки, опис функціонального наповнення об'єкта проєктування, розробка дизайну, розробка додатку.

5. Перелік графічного матеріалу: 27 рис.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Повстяна Ю.С.		
Специфікація вимог до розробленої системи	Повстяна Ю.С.		
Розробка об'єкта проєктування	Повстяна Ю.С.		
Нормоконтроль	Повстяна Ю.С.		
Гарант ОП	Ліщина Н.М.		
Показник запозичень тексту		822%	
Академічна доброчесність	Яциук А.А.		

7. Дата видачі завдання «2» лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ 3/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	05.02.2023 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2023 р.	вик.
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	10.03.2023 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкта проєктування та проєктування бази даних	17.04.2023 р.	вик.
5	Практична реалізація об'єкта проєктування та розробка бази даних	18.05.2023 р.	вик.
6	Тестування та налагодження об'єкта проєктування	01.06.2023 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	08.06.2023 р.	вик.

Здобувач вищої освіти

(підпис)

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

(прізвище, ініціали)

Рецензія
на кваліфікаційну роботу

Здобувач вищої освіти:
Коваль Павло Сергійович

Тема: Розробка додатку для зарядки електрокарів з використанням Vue.js / Development of an Application for Charging Electric Cars based on Vue.js

Коротка характеристика кваліфікаційної роботи:

Робота складається з трьох розділів, протягом яких дипломатом здійснюється огляд поставленої перед ним проблеми, пошук рішень та їх реалізації. Для реалізації програмного забезпечення було наступні програмні засоби: Vue.js, Node.js, Websockets, Vuetify.

Самостійні розробки і пропозиції автора:

Включають впровадження різноманітних функціональних можливостей у програмне забезпечення для зарядки, таких як вибір типу роз'єму, інтерфейс з декількома мовами тощо. Автор також пропонує вдосконалення системи безпеки за допомогою використання протоколу OSCRP для захисту даних користувачів. Ці самостійні розробки автора спрямовані на поліпшення функціональності, зручності та безпеки ПЗ для зарядки електромобілів.

Практичне значення роботи:

Полягає в розробці програмного забезпечення для зарядки електромобілів, що відповідає сучасним потребам та технологіям. Результати цієї роботи дозволяють користувачам відстежувати стан зарядки своїх електромобілів в реальному часі, забезпечуючи точну інформацію про рівень заряду, час та потужність. Розроблене програмне забезпечення покращує зручність використання зарядних станцій.

Недоліки:

У роботі не виявлено новації – вона більш походить на альтернативу існуючим варіантам вирішення проблеми, проте слід помітити, що складна структура роботи добре висвітлює вміння дипломанта.

Загальний висновок:

Рівень якості виконаної роботи є задовільним. Усі поставлені завдання успішно й повноцінно виконані. Кваліфікаційна робота виконана на якісному рівні та рекомендується до захисту та заслуговує на високу оцінку.

Рецензент: к.т.н Ткачук А.А.

(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи,
(підпис)

«05» грудня 2025 р.

АНОТАЦІЇ

Коваль П.С. Розробка додатку для зарядки електрокарів з використанням Vue.js. Рукопис.

Кваліфікаційна робота за спеціальністю 121 Інженерія програмного забезпечення. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота присвячена розробці системи для зарядки електроавтомобілів. Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі досліджено сучасний стан проблеми із електроавтомобілями. У другому розділі визначено вимоги для розроблювальної програми. У третьому розділі продемонстровану практичну реалізацію об'єкта проектування.

Ключові слова: *додаток, зарядка електрокарів, JavaScript, Vue.js.*

ANNOTATION

Koval P.S. Development of an Application for Charging Electric Cars based on Vue.js. Bachelor thesis. Manuscript.

Bachelor thesis in the specialty 121 Software Engineering. Lutsk National Technical University. Lutsk, 2023.

The thesis is devoted to the development of an application for charging electric cars using Vue.js. The bachelor's qualification work consists of an introduction, three sections, conclusions, a list of used sources and appendices.

The first chapter examines the current state of the problem with electric cars. The second section defines the requirements for the development program. The third section demonstrates the practical implementation of the design object.

Keywords: *application, charging electric cars, JavaScript, Vue.js*

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ДОДАТКІВ ДЛЯ ЗАРЯДКИ ЕЛЕКТРОКАРІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	8
1.1 Аналіз сучасного стану проблеми	8
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	18
Висновки до розділу 1	18
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	19
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	19
2.1.1 Функціональні та нефункціональні вимоги	21
2.1.2 Моделі елементів програмного додатку у нотації UML	22
2.1.3 Створення UX/UI-дизайну	25
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	29
Висновки до розділу 2	33
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ ДЛЯ ЗАРЯДКИ ЕЛЕКТРОКАРІВ З ВИКОРИСТАННЯМ VUE.JS	34
3.1 Практична реалізація об'єкта проектування	34
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	43
3.3 Розробка документації для програмного продукту	46
Висновки до розділу 3	49
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52
ДОДАТКИ	54

ВСТУП

Актуальність теми цього дослідження полягає у зростаючій важливості електромобілів як екологічного варіанту транспортування. Оскільки впровадження таких автомобілів продовжує зростати, потреба в ефективних і зручних рішеннях для зарядки стає першорядною. Вивчаючи існуючі рішення та визначаючи області для вдосконалення, ця робота має на меті зробити внесок у створенні нового досвіду користувача.

На даному етапі електромобілі набули значної популярності як екологічно чистий вид транспорту, що сприяє зменшенню викидів шкідливих газів. Однак наявність і доступність надійної зарядної інфраструктури залишається ключовим питанням для власників електромобілів. Щоб вирішити цю проблему, було розроблено програму, яка допомагає користувачам зручно заряджати авто.

У цьому дослідженні було проведено комплексний аналіз існуючих рішень у сфері зарядки електромобілів. Мета полягала в тому, щоб виявити будь-які прогалини або обмеження в поточних знаннях і вивчити можливості для вдосконалення. Завдяки ретельному вивченню доступної літератури і тематичних досліджень було отримано цінну інформацію щодо сучасного стану зарядки електромобілів.

Кілька популярних додатків для заряджання були оцінені, щоб зрозуміти їхні особливості, функціональні можливості та взаємодію з користувачем. Ці програми, розроблені провідними компаніями галузі, надають користувачам можливість шукати зарядні станції поблизу, переглядати їх наявність і навіть бронювати.

Реалізація цього дослідження служить багатьом цілям. По-перше, це надає цінну інформацію для розробників і професіоналів галузі, які займаються розробкою та вдосконаленням додатків. Розуміючи поточний стан галузі, з якими стикаються користувачі, розробники можуть приймати обґрунтовані рішення та впроваджувати вдосконалення, які відповідають конкретним потребам користувачів. По-друге, це дослідження робить внесок у базу знань у цю сферу.

Висвітлюючи останні інновації та передовий досвід, він забезпечує основу для майбутніх досліджень і розробок.

Розробка надійних і зручних додатків для зарядки має вирішальне значення для сприяння широкому впровадженню електромобілів. Вирішуючи проблеми, пов'язані з зарядною інфраструктурою, ці додатки відіграють важливу роль у сприянні переходу до більш екологічного майбутнього.

Мета дослідження є розробка програмного забезпечення для електричних зарядних станцій, що дозволяє користувачам швидко та зручно заряджати свої електричні автомобілі. Основним завданням є розробка інтуїтивно зрозумілого та простого інтерфейсу, який забезпечує зручний доступ до функцій заряджання та оплати, а також впровадження розширених можливостей моніторингу та діагностики зарядних станцій для підвищення їх ефективності та надійності.

Об'єкт дослідження є додаток, який дозволяє користувачам швидко та зручно заряджати свої електромобілі. Дослідження включає розробку програмного забезпечення, використання сучасних технологій та бібліотек для досягнення найкращої продуктивності та забезпечення користувачам максимального комфорту під час використання додатку. Ця тема має важливість у зв'язку з широким поширенням електромобілів та необхідністю розвитку інфраструктури для їх зарядки. Розробка додатку для зарядки електромобілів є актуальною проблемою в сучасному світі, і вимагає дослідження нових технологій та розробки продуктивних програмних засобів.

Предмет дослідження є програмна реалізація системи зарядки електромобілів на базі Vue.js та використання таких технологій, як WebSocket та i18n. Основними аспектами дослідження є проєктування, розробка та тестування програмного забезпечення, а також вивчення можливостей використання відповідних технологій для розробки масштабованих та ефективних систем зарядки електромобілів.

РОЗДІЛ 1

АНАЛІЗ ДОДАТКІВ ДЛЯ ЗАРЯДКИ ЕЛЕКТРОКАРІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

На сьогоднішній день все більше виникає потреба у станціях для зарядки електроавтомобілів та стає все більше користувачів автомобілями такого типу. Тому потрібно задовільнити потребу людей у доступності і зручності підзарядки власних авто. Коли ці потреби будуть задовільнені, то більше людей користуватимуться такими авто.

Актуальним є також екологічність авто. В Європі електромобілі викидають майже втричі менше CO₂, ніж бензинові або дизельні автомобілі. Це згідно з новим онлайн-інструментом, який дозволяє громадськості порівнювати викиди за життєвий цикл електромобіля з транспортними засобами, що працюють на викопному паливі.

Для розробки додатку для зарядки електроавтомобілів потрібно провести аналіз серед аналогів, щоб уникнути недоліків, яких допустили конкуренти. Для аналізу існуючих аналогів було обрано наступні системи:

- Electrify America;
- Kempower;
- UGV Chargers;
- EVgo.

Electrify America – це мережа станцій швидкої зарядки для електромобілів, розроблена компанією Volkswagen у рамках світової угоди з урядом США щодо скандалу з дизельними викидами. Компанія має партнерські стосунки з такими мережами автозаправних станцій, як Love's Travel Stops, і встановила свої зарядні станції у вибраних місцях Love's. Дана мережа налічує понад 788 точок заряджання та понад 3531 окремих зарядних пристроїв станом на вересень 2022 року [2].

Мета Electrify America – сприяти більш широкому впровадженню автомобілів з нульовим викидом (ZEV), пропонуючи швидку та зручну мережу заряджання. Це буде стимулювати сучасні електромобілі, а також швидке зростання моделей електромобілів, яке очікується від багатьох автомобільних компаній від сьогодні до початку 2020-х років [2].

Переваги Electrify America [3]:

- велика мережа. Компанія має одну з найбільших мереж швидкісних зарядних станцій для електромобілів у США з понад 600 зарядними станціями, розташованими в 45 штатах і окрузі Колумбія. Це полегшує водіям електромобілів пошук зарядної станції, коли вона їм потрібна;

- швидкість заряджання. Зарядні станції здатні видавати до 350 кВт потужності, що може забезпечити значну кількість заряду за короткий проміжок часу. Це означає, що водії електромобілів можуть витратити менше часу на зарядку та більше часу на дорогу;

- зручне розташування. Багато зарядних станцій Electrify America розташовані у зручних місцях, таких як: торговельні центри, ресторани та автомагістралі, що дозволяє водіям легко заряджати свої транспортні засоби під час виконання завдань або подорожі;

- додаток і варіанти оплати. Система пропонує зручний додаток, який дозволяє водіям знаходити та резервувати зарядні станції та навіть оплачувати сеанси заряджання через додаток. Це робить процес зарядки більш зручним і безперебійним.

Недоліками системи є [3]:

- висока вартість: тарифи зарядки Electrify America можуть бути дорожчими, ніж в інших мережах зарядки, особливо для сеансів швидкої зарядки. Це може зробити дорожчим володіння та експлуатацію електромобіля;

- обмежене покриття в деяких регіонах. Незважаючи на те, що компанія має велику мережу зарядних станцій, у деяких регіонах їх покриття обмежене, особливо в сільській місцевості чи віддалених районах;

– технічні проблеми: надходили повідомлення про технічні проблеми із зарядними станціями Electrify America. Прикладом повідомлення про несправність: станції не працюють належним чином або сеанс заряджання переривається. На рисунках 1.1 – 1.3, зображено інструкцію, підключення конектору та панель стану зарядки.



Рисунок 1.1 – Інструкція станції Electrify America [3]



Рисунок 1.2 – Підключення конектору до системи [3]



Рисунок 1.3 – Панель стану зарядки авто Electrify America [3]

Наступною системою для порівняння – Kempower. Це програма зарядки електромобілів, розроблена KenGen, провідною компанією з виробництва електроенергії в Кенії [4]. Компанія зазнала величезного зростання. До кінця 2023 року намагаються відкрити власні підприємства в США, Північній Кароліні, а також вивчають варіанти розширення виробництва в Європі.

Можливості системи Kempower [4]:

- контролювати стан заряду в мобільному додатку;
- оплата картою або телефоном;
- заряджати для кількох видів вилок.

Переваги Kempower:

- доступна ціна. Тарифи Kempower відносно доступні. Це робить для кенійців доступнішим володіння та керування електромобілями;
- висока швидкість заряджання. Зарядні станції Kempower здатні видавати до 50 кВт потужності, що може забезпечити значну енергії за короткий час. Це

означає, що водії електромобілів можуть витрачати менше часу на зарядку та більше часу на дорогу (рис. 1.4);

- стійкі джерела енергії. В рамках своєї прихильності до сталого розвитку компанія Kempower працює над тим, щоб її зарядні станції використовували відновлювані джерела енергії, такі як вітер і сонце. Це допомагає зменшити загальний вуглецевий слід електромобілів і підтримувати чистіше довкілля;

- урядова підтримка. Уряд Кенії висловив підтримку програмі Kempower і працює над створенням політики та стимулів, які заохочують впровадження електромобілів.



Рисунок 1.4 – Головна панель стану зарядки авто Kempower [4]

На рисунку 1.5, зображено графік потужності зарядної станції.

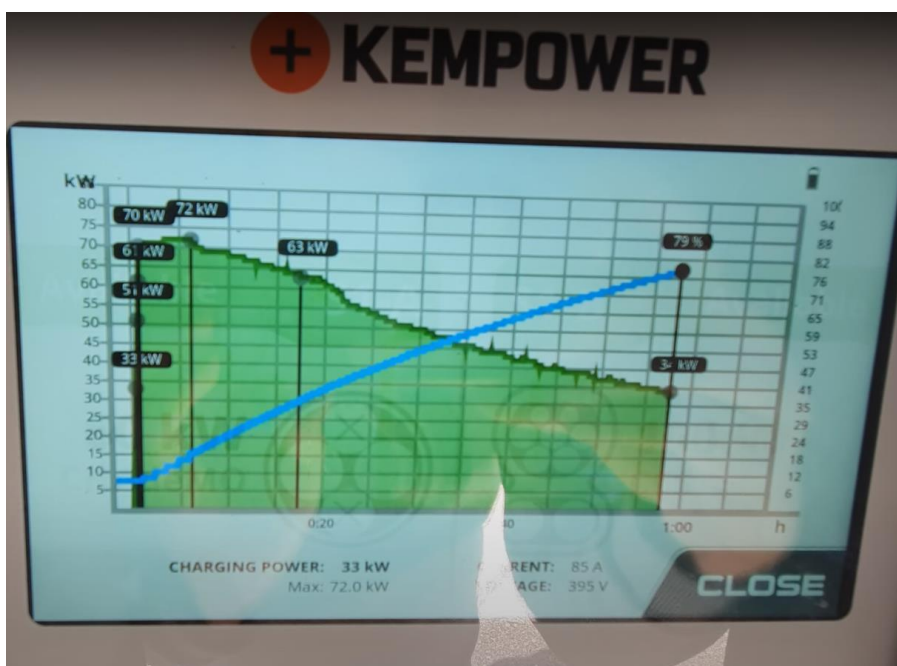


Рисунок 1.5 – Графіку потужності зарядки Kempower [4]

Недоліки:

- обмежене покриття: хоча компанія досягла значного прогресу в розширенні своєї мережі зарядних станцій, у Кенії все ще є багато районів, де програма не охоплює. Через це водіям електромобілів може бути складно подорожувати на великі відстані, не турбуючись про те, що заряд заряду закінчиться;
- технічні проблеми: надходили повідомлення про технічні проблеми із зарядними станціями Kempower. Це може засмучувати водіїв електромобілів, які покладаються на зарядну мережу для пересування;
- обмежена доступність моделей електромобілів: наразі в Кенії доступні обмежені варіанти моделей електромобілів, через що кенійцям може бути важко купувати електромобілі навіть за наявності зарядних станцій;
- недостатня обізнаність: серед населення Кенії все ще недостатньо обізнаності та розуміння електромобілів та їхніх переваг. Це може ускладнити для Kempower залучення великої кількості користувачів і просування впровадження електромобілів.

Ще однією системою для порівняння було обрано – UGV Chargers. Це український виробник зарядних станцій для електромобілів (рис. 1.6). Фахівці компанії проєктують і виготовляють зарядні станції для електромобілів, розробляють програмне забезпечення та мобільний додаток (рис. 1.7). UGV Chargers розвиває зарядну інфраструктуру для електрокарів, а також розвиває власну мережу зарядних станцій. На швидкісних (експрес) зарядних станціях UGV Chargers, процес зарядки займе у вас від 40 до 60 хвилин, поки акумулятор електромобіля не зарядиться на 80% [5].

The screenshot displays the UGV Chargers mobile application interface for a specific charging station. At the top, the station is identified as '#028 - A3C 'CORAL''. Below this, key information is presented in two columns: 'Робочий час' (Operating hours) from 00:00 to 24:00, 'Мережа' (Network) as UGV (CORAL), 'Номер телефону' (Phone number) 0800401090, and 'Адреса' (Address) in Kyiv Oblast, Boryspil, Gliboc'ka St. 113 A. A small image of the charging station is shown below the contact info. A table lists the available charging options with columns for 'Порт' (Port), 'Тип' (Type), 'Потужність' (Power), and 'Послуга' (Service). The first option is selected, showing a green checkmark, port 1, J1772 type, 7 kW power, and a service fee of 7.99 UAH per kWh. Below the table, there are three buttons for selecting the charging amount: 'кВт-г' (kWh), 'кВт-г' (kWh), and 'Безлімітна' (Unlimited). A green notification box states that the amount will be debited from the user's account based on the selected rate and that a recalculation will occur if the charging is interrupted. At the bottom, the total cost is shown as 'Разом: 0.00 UAH', followed by two buttons: 'Увійти' (Login) and 'Зарядитись як гість' (Charge as guest).

Порт	Тип	Потужність	Послуга
1	J1772	7 кВт	7,99 грн/кВт-ч

Разом: 0.00 UAH

Увійти

Зарядитись як гість

Рисунок 1.6 – Початок зарядки UGV Chargers [5]

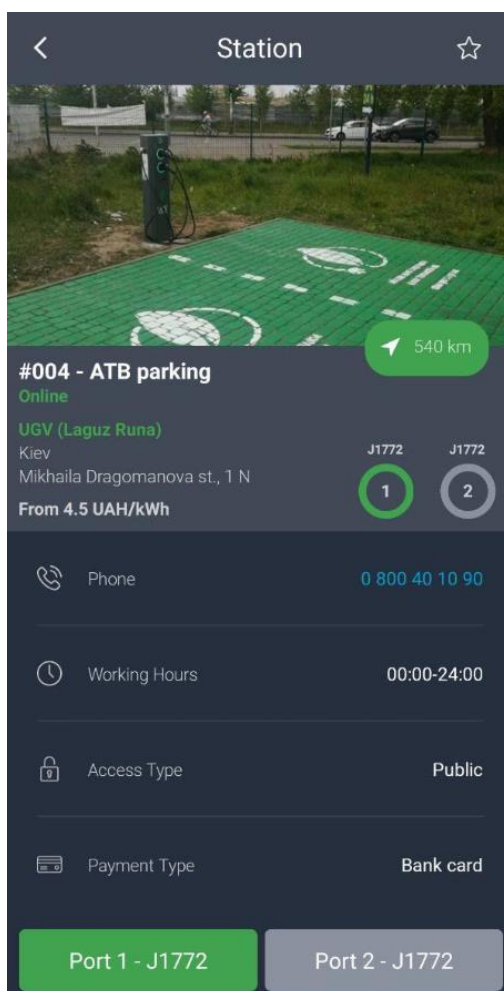


Рисунок 1.7 – Перегляд станцій через мобільний додаток UGV Chargers [5]

Однією із ключових застосувань цієї системи є забезпечення надійної та доступної мережі зарядних станцій для водіїв електромобілів. Маючи понад 800 пунктів по всій території США, зарядні станції EVgo стратегічно розташовані в таких зручних місцях, як торговельні центри, ресторани та готелі, що дозволяє водіям легко заряджати акумулятори під час роботи [5].

Загалом застосування EVgo полягає в тому, щоб забезпечити зручну, надійну та стійку мережу зарядних станцій для електромобілів, яка підтримує зростання та впровадження електромобілів.

Переваги:

– висока швидкість зарядки: EVgo пропонує одні з найшвидших швидкостей зарядки для електромобілів, причому багато їхніх станцій здатні

забезпечити до 350 кВт потужності. Це означає, що водії можуть швидко зарядити батарею під час руху;

- велика мережа: EVgo має велику мережу зарядних станцій по всій території США з понад 800 точками в 67 містах. Це полегшує водіям пошук зарядної станції, коли вона їм потрібна;

- зручне розташування: багато зарядних станцій EVgo розташовані у зручних місцях, наприклад у торгових центрах, ресторанах і готелях, що дозволяє водіям легко заряджати свої транспортні засоби під час виконання завдань або подорожі;

- додаток і варіанти оплати: EVgo має зручний додаток, який дозволяє водіям знаходити та резервувати зарядні станції та навіть оплачувати сеанси зарядки через додаток. Це робить процес зарядки більш зручним і безперебійним.

Недоліки:

- висока вартість: тарифи зарядки EVgo можуть бути дорожчими, ніж в інших мережах зарядки, особливо для сеансів швидкої зарядки. Це може зробити дорожчим володіння та експлуатацію електромобіля;

- обмежене покриття в деяких регіонах. Незважаючи на те, що EVgo має велику мережу зарядних станцій, можуть бути деякі райони, де їх покриття обмежене, особливо в більш сільських або віддалених районах;

- відсутність безкоштовного заряджання: на відміну від деяких інших мереж заряджання, EVgo не пропонує жодних варіантів безкоштовного заряджання. Це може бути недоліком для водіїв, які прагнуть мінімізувати витрати на зарядку;

- потрібне членство. Незважаючи на те, що EVgo пропонує варіанти оплати за використання, водіям, які хочуть скористатися знижками, потрібно буде підписатися на членство, що може бути додатковим коштом.

На рисунках 1.8, 1.9, зображено панель зарядки та інформацію про стан зарядки станції EVgo.



Рисунок 1.8 – Панель зарядки EVgo [6]



Рисунок 1.9 – Панель інформації про зарядку EVgo [6]

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою даної роботи є створення додатку для системи зарядки електрокарів. Клієнти матимуть можливість переглядати: стан зарядки, потужність, вибраний роз'єм. А також інструкція з експлуатації системи, вибір оплати: через телефон, банківську карточку із можливістю відправки чеку на електронну пошту. Після реєстрації користувач матиме власний обліковий запис із його даними.

Для досягнення кінцевої мети необхідно виконати наступний список завдань:

- проведення аналізу актуальності додатку;
- вибір технології та засобів реалізації для створення проєкту;
- написання технічного завдання;
- опис функціоналу;
- опис можливостей користувачів;
- створення інтерфейсу додатку;
- налаштування середовища розробки та технологій;
- написання коду для проєкту;
- тестування додатку;
- розробка документації для програмного продукту.

Висновки до розділу 1

Було розглянуто аналоги систем, додатків для зарядки електрокарів. Кожна з них має свої особливості. Враховуючи плюси і мінуси аналогів, можна покращити власну систему, а саме: дружелюбний інтерфейс, доступність, захищеність та інструкція по експлуатації. Поставлено завдання на кваліфікаційну роботу бакалавра.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проєктування програмного забезпечення

Для проведення аналізу та визначення вимог до розроблюваного ПЗ необхідно:

- визначити предметну область і цілі ПЗ;
- проаналізувати потреби та вимоги користувачів та інших зацікавлених сторін;
- визначити функціональні та нефункціональні вимоги до ПЗ;
- вказати обмеження системи, такі як апаратне забезпечення, програмне забезпечення;
- створити випадки використання, діаграми послідовності та інші необхідні UML моделі, які описують поведінку ПЗ та взаємодію з користувачами;
- документація в стислій формі.

Керуючись ціллю та предметною областю додатку, можна виявити такі вимоги:

- загальний опис: розроблена система є додатком для зарядки електромобілів. Програма повинна мати можливість відображати стан з'єднання та зарядки як для лівого, так і для правого роз'ємів, а також повинна мати можливість виявляти та сповіщати користувачів, якщо виникають проблеми з підключенням;
- управління доступом: система повинна мати як публічний, так і приватний контроль доступу. Публічний доступ має бути доступним для всіх користувачів і мати можливість приймати оплату банківською картою. Приватний доступ має бути доступний лише авторизованим користувачам і має включати додаткові функції, недоступні у відкритому доступі;
- керування користувачами: система повинна мати функцію керування користувачами для оперування обліковими записами користувачів, паролями та

рівнями доступу. Функціональність керування користувачами повинна містити можливість додавати, редагувати та видаляти користувачів, а також призначати рівні доступу користувачам;

- обробка платежів: система повинна містити функцію обробки платежів для прийому від користувачів. Функціональність обробки платежів повинна бути інтегрована з публічним контролем доступу та повинна підтримувати декілька методів оплати (банківська карта або карта місцевого АЗС);

- статус роз'ємів: система повинна мати можливість відображати стан лівого та правого роз'ємів. Статус має оновлюватися в режимі реального часу та вказувати, чи доступні з'єднувачі для використання, чи є проблеми з підключенням;

- стан зарядки: система повинна мати можливість відображати стан зарядки для кожного підключеного електромобіля. Статус заряджання має оновлюватися в режимі реального часу та вказувати поточний рівень заряду, приблизний час, що залишився для заряджання, а також будь-які проблеми чи помилки, які можуть виникнути під час процесу заряджання;

- безпека: система повинна включати функції безпеки для захисту даних користувача та запобігання несанкціонованому доступу. Функції безпеки повинні включати шифрування даних, автентифікацію користувача та контроль доступу;

- інтерфейс користувача: система повинна мати інтуїтивно зрозумілий UI, простий у використанні та навігації. Інтерфейс має бути розроблений із сучасним і адаптивним макетом і повинен містити такі функції, як пошук, функція фільтрації, кнопка стоп/старт та оновлення в реальному часі;

- продуктивність: система повинна бути розроблена для роботи з великою кількістю користувачів і зарядних станцій, а також повинна мати можливість обробляти запити та оновлювати статуси в режимі реального часу. Система повинна бути масштабованою та мати можливість працювати з додатковими зарядними станціями та користувачами за потреби;

– сумісність: система повинна бути сумісна з низкою веб-браузерів і пристроїв, включаючи настільні та мобільні пристрої. Система має бути розроблена таким чином, щоб бути швидкою та доступною, а також повинна забезпечувати узгоджену взаємодію з користувачем на всіх пристроях і платформах.

Загалом розроблена система має бути надійною, масштабованою та зручною для користувача. Вона має надавати низку функцій і функціональних можливостей для керування зарядними станціями та заряджанням електромобілів, а також повинен легко обслуговувати велику кількість користувачів і зарядних станцій.

2.1.1 Функціональні та нефункціональні вимоги

Також варто розглянути функціональні та нефункціональні вимоги.

Функціональні вимоги:

- система повинна дозволяти користувачам вибирати з двох типів роз'ємів (лівий і правий);
- відображати статус підключення (підключено або не підключено);
- дозволяти користувачам оплачувати зарядку банківською картою або картою місцевої АЗС;
- відображати стан, час і потужність зарядки;
- дозволяти користувачам підключатися до бездротової мережі Інтернет;
- дозволяти користувачам вибрати одну з трьох мов (українська, англійська та німецька).

Нефункціональні вимоги:

- система має бути зручною та простою у використанні;
- безпечною та захищати особисту та фінансову інформацію користувачів;
- повинна бути надійною та надавати точну інформацію про стан зарядки;
- система має бути масштабованою та здатною працювати з кількома користувачами та зарядними станціями;

- повинна бути сумісна з різними типами електромобілів і зарядними станціями.

Вимоги до інтерфейсу користувача:

- інтерфейс користувача повинен включати два типи роз'ємів (лівий і правий);
- повинен відображати стан зарядки, час і потужність;
- інтерфейс повинен дозволяти користувачам вибрати мову (українська, англійська або німецька);
- надавати користувачам доступ до меню налаштувань для підключення до бездротового Інтернету.

Вище описані вимоги, спрямовані на створення зручної, безпечної та надійної програми для зарядки електромобілів. Система повинна дозволяти користувачам вибирати роз'єм, оплачувати зарядку та контролювати стан зарядки. Крім того, користувачі зможуть підключитися до бездротового Інтернету та вибрати бажану мову.

2.1.2 Моделі елементів програмного додатку у нотації UML

Для того, що скласти подібні моделі потрібно визначити основні компоненти системи та їхні взаємозв'язки, функції та операції, які виконуватиме кожен компонент [7]. Також визначити інтерфейси та структури даних, які будуть використовуватися для обміну інформацією між компонентами.

Першою моделлю представлено діаграму послідовності, рисунок 2.1, для параметрів мови та підключення до Інтернету. Ця діаграма є лише представленням деяких основних функцій програми і не є вичерпними.

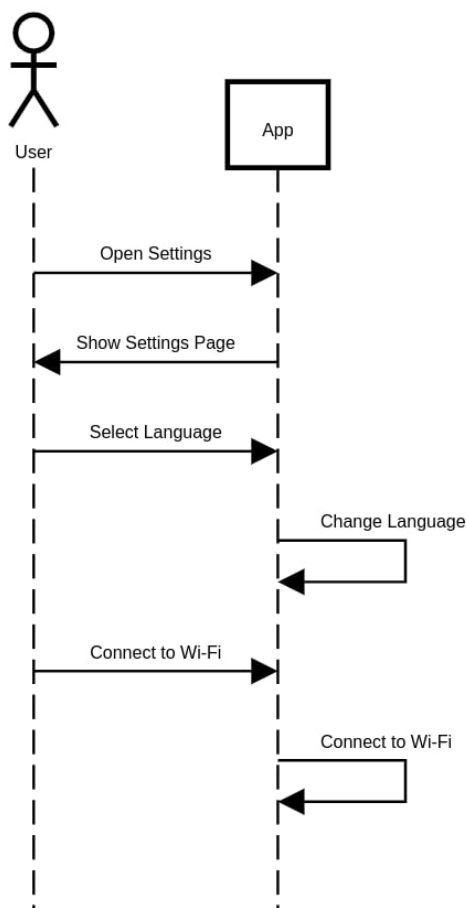


Рисунок 2.1 – Діаграма послідовності для параметрів мови та підключення до Інтернету

Джерело: розроблено автором

На рисунку 2.2, представлено діаграму, де чітко зображено послідовність кроків, які виконує користувач, електромобіль і зарядна станція, включаючи підтвердження вибраного конектора, підтвердження та початок сеансу заряджання, а також отримання статусу заряджання та резюме.

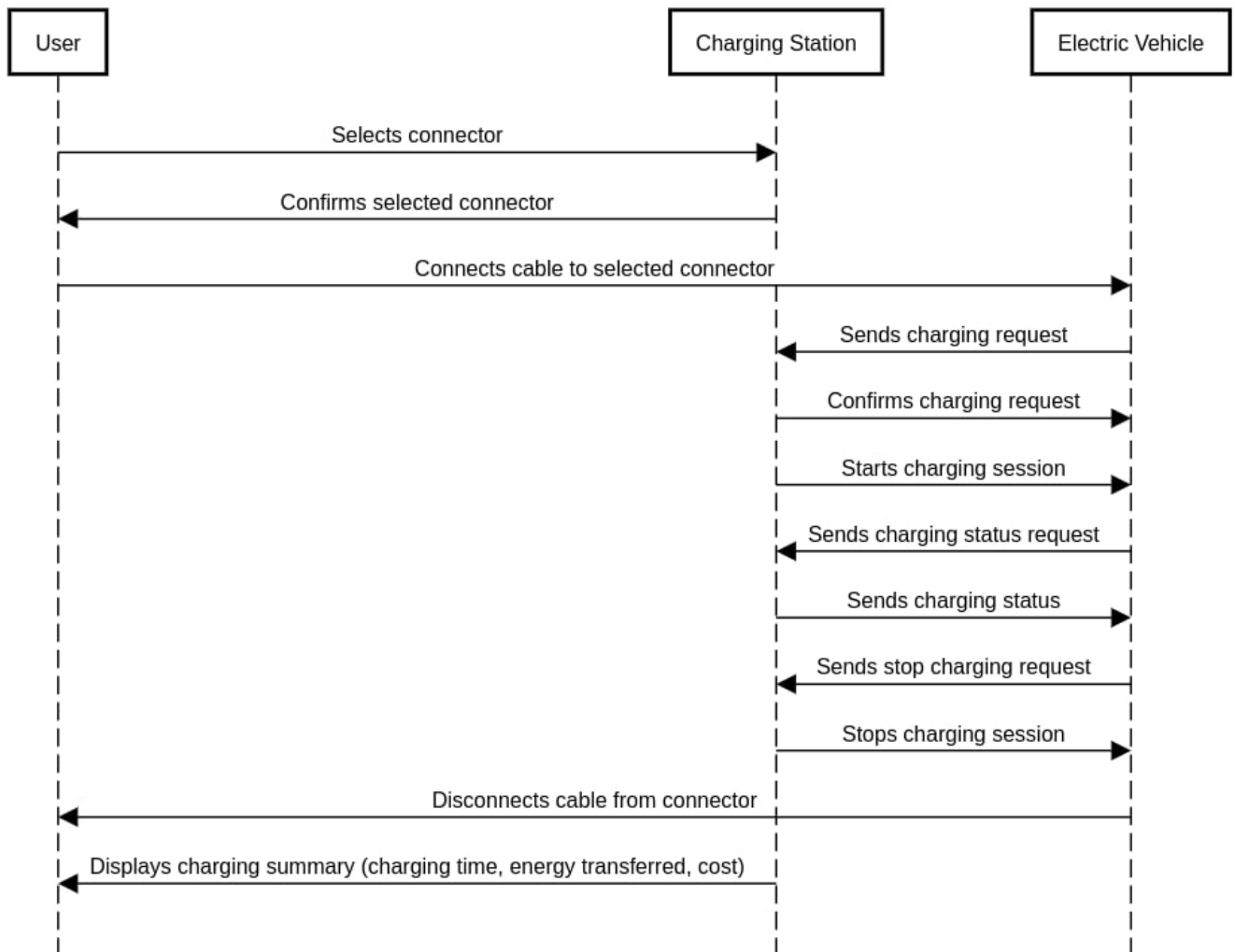


Рисунок 2.2 – Діаграма послідовності для вибору роз'єму та стану заряджання

Джерело: розроблено автором

На рисунку 2.3 зображено користувача (User), де вибирає спосіб оплати та варіант із зарядної станції (Charging Station). Після підтвердження платежу на зарядній станції відображається платіжний інтерфейс NFC. Користувач авторизує платіж у своєму банку (Bank) і підтверджує платіж, торкаючись до NFC пристрою на зарядній станції. Зарядна станція надсилає платіжну інформацію на АЗС (Gas Station), яка надсилає підтвердження оплати назад на зарядну станцію. Нарешті, зарядна станція показує користувачеві стан зарядки.

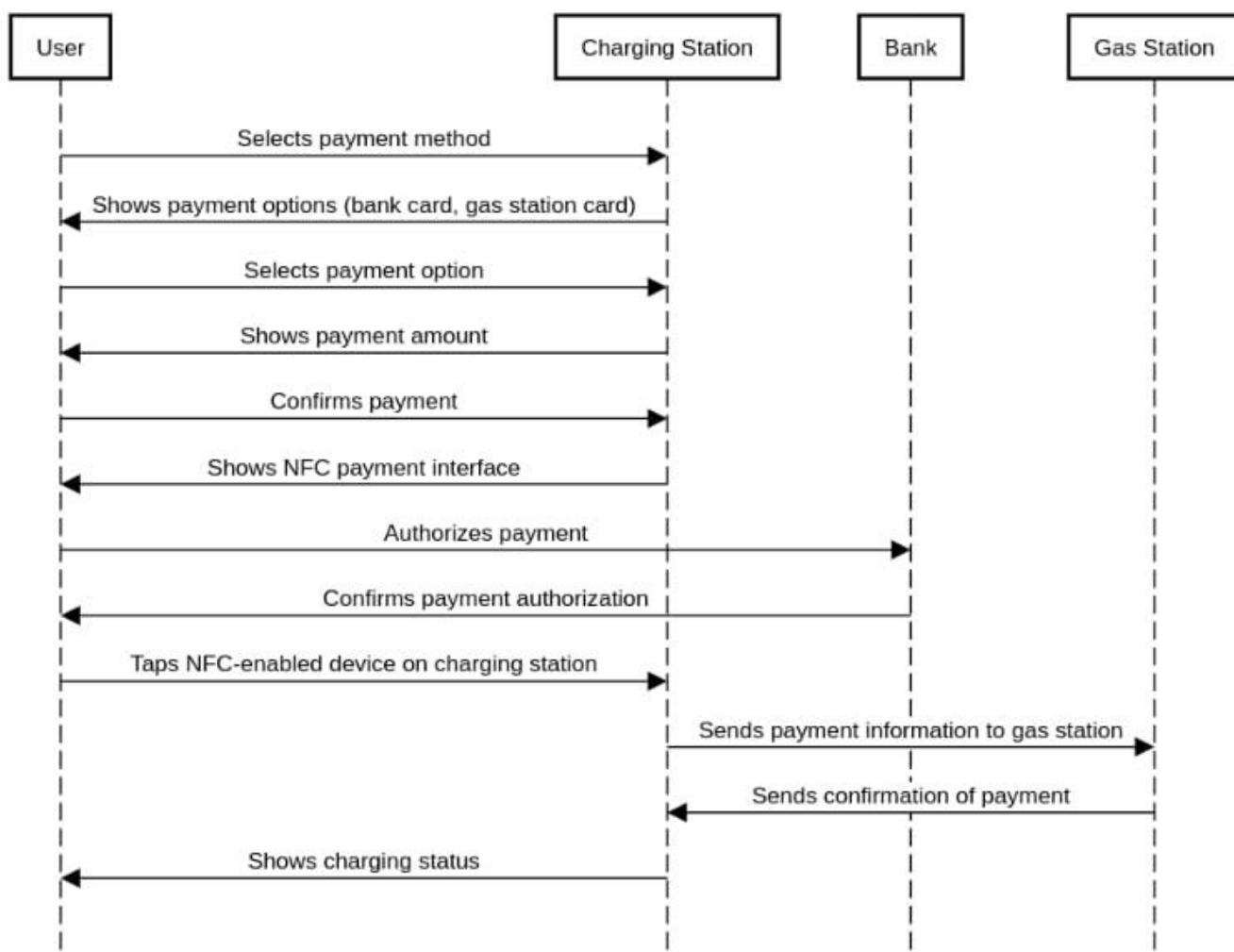


Рисунок 2.3 – Діаграма послідовності обробки платежів

Джерело: розроблено автором

2.1.3 Створення UX/UI-дизайну

Для створення дизайну потрібна Figma. Це чудовий інструмент для створення дизайну UI/UX. Також це потужний інструмент для спільної роботи, який дозволяє дизайнерам створювати та ділитися своїми роботами в режимі реального часу зі своєю командою чи клієнтами. Figma має широкий набір функцій і функціональних можливостей, які спрощують створення та ітерацію дизайнів, включаючи інструменти векторного редагування, функції прототипування і надійну бібліотеку компонентів інтерфейсу користувача.

Однією з найбільших переваг Figma є те, що це хмарний інструмент, який означає, що ви можете отримати доступ до своїх проєктів з будь-якого місця та

співпрацювати з іншими в режимі реального часу. Це робить його чудовим варіантом для команд, які розподілені в різних місцях або працюють віддалено. Figma також пропонує широкий спектр інтеграцій з іншими інструментами, такими як системи проєктування, інструменти створення прототипів та інструменти управління проєктами, що дозволяє легко включити Figma у ваш існуючий робочий процес проєктування.

На рисунку 2.4, зображено перший варіант головної панелі додатку. На верхній панелі зображено два конектора, один з них вибраний і показується стан: потужність, час, відсоток зарядки. Знизу кнопка стоп/старт для зарядки та налаштування.

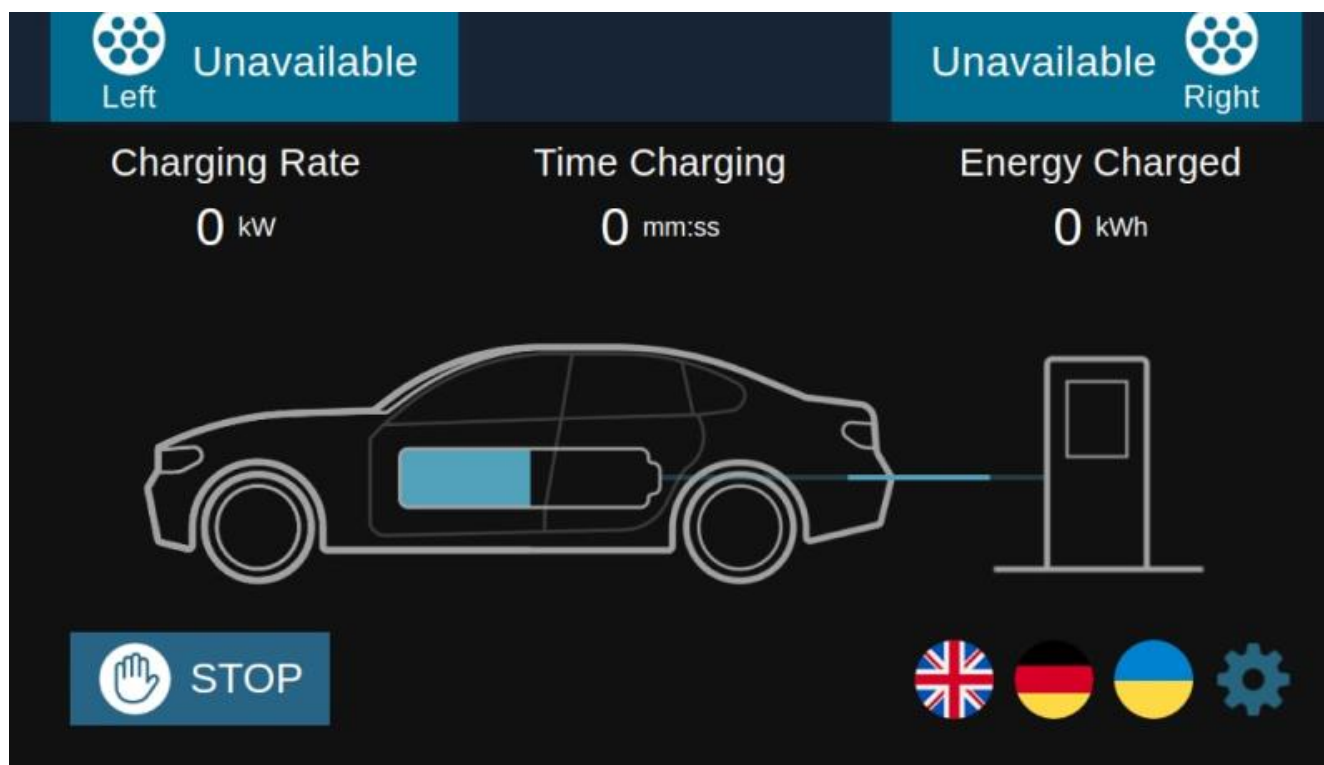


Рисунок 2.4 – Головна сторінка першої версії

На рисунку 2.5 зображено налаштування мережі першої версії програми, де можна підключатись до Wi-Fi, з'являється меню із списком доступних мереж. Є варіант підключення до дротової мережі (рис. 2.6).

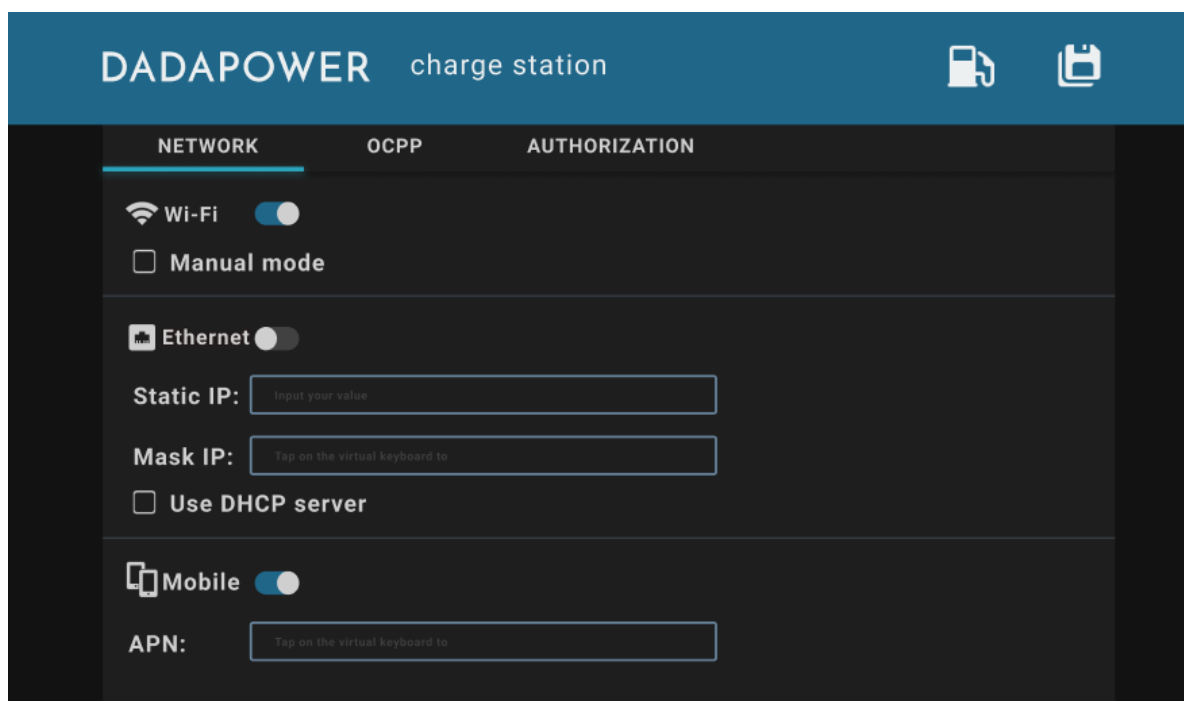


Рисунок 2.5 – Налаштування програми

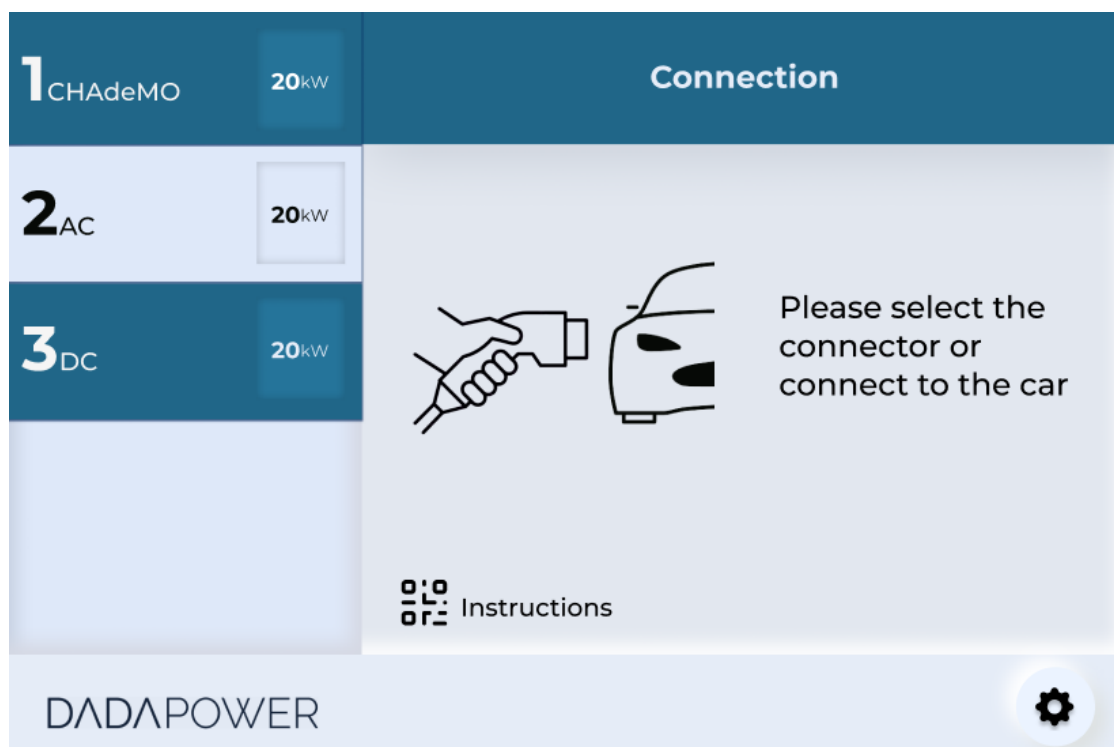


Рисунок 2.6 – Вибір конектора

На рисунку 2.7, продемонстровано вивід помилок у разі, якщо підключення хибне.

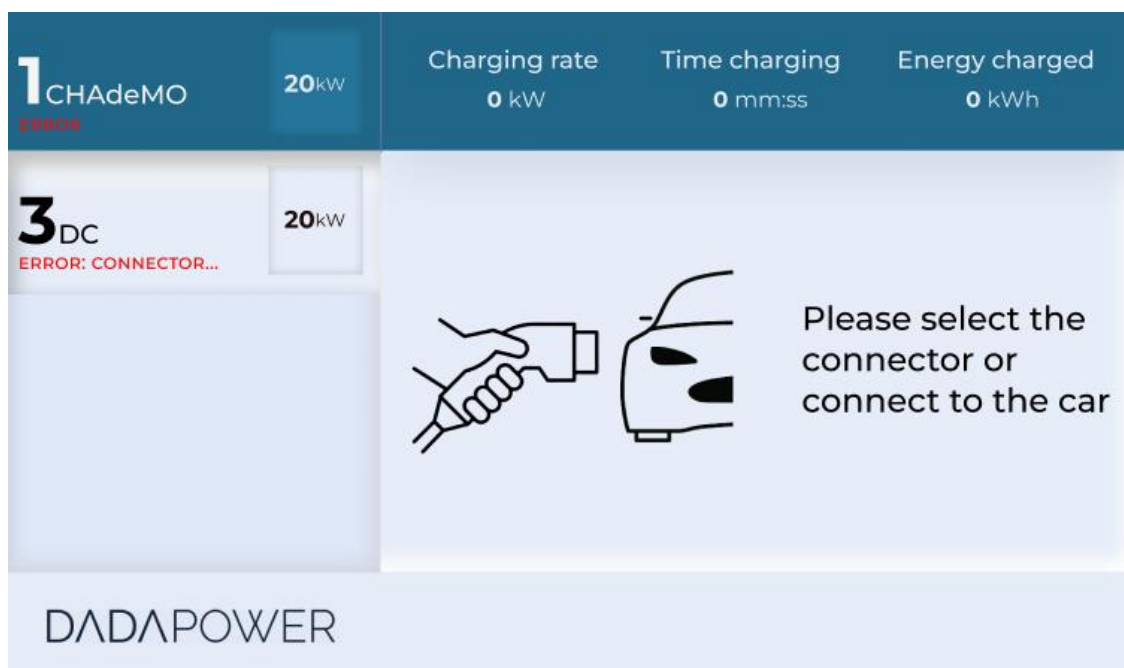


Рисунок 2.7 – Помилка із підключенням

Додаток надає користувачам можливість заряджати свої електромобілі, отримавши доступ до зарядної станції через додаток. Додаток має кілька функцій, які дозволяють користувачам легко та ефективно заряджати свої транспортні засоби.

Сторінка привітання надає користувачам можливість вибрати роз'єм і почати процес заряджання. Після того, як користувач вибере потрібний роз'єм, він перейде на сторінку, де він зможе побачити статус свого процесу заряджання, включаючи час, що залишився, і кількість використаної енергії.

Додаток також містить кнопку зупинки/запуску, яка дозволяє користувачам призупиняти або перезапускати процес заряджання. Це корисно в ситуаціях, коли користувачеві потрібно тимчасово зупинити процес заряджання або потрібно перезапустити його після перерви.

Додаток також містить сторінку налаштувань, де користувачі можуть підключатися до бездротової мережі Інтернет і вибирати між трьома різними мовами: українською, німецькою та англійською.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Перед вибором засобів та методів слід враховувати наступні чинники:

- продуктивність: програма повинна мати можливість обробляти велику кількість одночасних користувачів, а також забезпечувати швидкий час відповіді на взаємодію користувача;
- масштабованість: додаток має бути розроблено таким чином, щоб його можна було легко збільшити або зменшити за потреби залежно від кількості користувачів;
- безпека: програма має бути розроблена з урахуванням безпеки, включаючи такі заходи, як шифрування конфіденційних даних, безпечна автентифікація користувачів і захист від типових атак веб-додатків;
- взаємодія з користувачем. Додаток має бути розроблено таким чином, щоб забезпечувати плавну та інтуїтивно зрозумілу взаємодію з користувачем, із зрозумілими та простими у використанні інтерфейсами та корисними повідомленнями зворотного зв'язку;
- сумісність: програма має бути сумісною з широким спектром пристроїв і веб-браузерів і повинна добре працювати як на настільних, так і на мобільних пристроях.

Вибраним фреймворком для розробки став Vue.js. Причиною даного вибору було з його переваг, а саме:

- реактивне зв'язування даних: Vue.js надає функцію реактивного зв'язування даних, яка дозволяє легко підтримувати інтерфейс користувача в

актуальному стані зі змінами даних. Ця функція допомагає зменшити кількість коду, необхідного для оновлення DOM, і забезпечує швидший рендеринг програми;

- компонентна архітектура: Vue.js базується на компонентній архітектурі, що полегшує розробку складних програм, розділяючи їх на менші компоненти, які можна багаторазово використовувати. Ця функція допомагає покращити зручність обслуговування, багаторазове використання та масштабованість програми;

- простий синтаксис: Vue.js має простий синтаксис, який легко вивчити;

- продуктивність: Vue.js розроблений як легкий і швидкий, що робить його хорошим вибором для розробки високопродуктивних програм. Має невеликий розмір і високу швидкість візуалізації, що забезпечує кращий досвід користувача та швидший час завантаження;

- підтримка: Vue.js має велику та активну спільноту розробників, яка надає підтримку, ресурси та оновлення. Ця функція допомагає забезпечити стабільність і надійність фреймворку та полегшує пошук рішень будь-яких проблем, які можуть виникнути під час розробки (рис. 2.8).



Рисунок 2.8 – Логотип Vue [8]

Наступним вибором для розробки інтерфейсу став Vuetify (рис. 2.9) – це популярна бібліотека компонентів UI для Vue.js, яка надає набір повторно використовуваних компонентів інтерфейсу користувача [9]. Він пропонує широкий спектр готових компонентів і дозволяє легко налаштувати за допомогою системи оформлення тем.

Причиною вибору Vuetify як бібліотеки інтерфейсу користувача для додатка для зарядки електромобілів є простота використання, гнучкість, і функціональність. Vuetify постачається з широким набором компонентів, які можна легко використовувати для створення складних елементів інтерфейсу користувача, таких як форми, таблиці, картки. Це забезпечує швидший процес розробки, а також більш послідовний і згуртований дизайн усієї програми.

Крім того, Vuetify надає комплексну систему оформлення тем, яка дозволяє налаштовувати зовнішній вигляд програми. Це важливо, оскільки програма для заряджання електромобілів має бути чіткої теми, яку користувачі зможуть легко асоціювати з брендом. З Vuetify, легко створювати спеціальні теми, які можна легко застосувати до всієї програми.

Загалом, поєднання Vue.js і Vuetify забезпечує потужну та гнучку платформу для створення високоякісних користувацьких інтерфейсів для додатків для зарядки електромобілів.



Рисунок 2.9 – Логотип Vuetify [9]

Наступною технологією є WebSockets. Вона включає в себе такі особливості:

- передача даних у реальному часі: WebSockets забезпечує зв'язок у реальному часі між клієнтом і сервером [10]. Це забезпечує миттєве оновлення та зменшує потребу в постійному опитуванні;

- низька затримка: WebSockets забезпечує низьку затримку, високошвидкісний зв'язок між клієнтом і сервером, що робить його придатним для програм, які вимагають оновлення в реальному часі;

- дуплексний зв'язок: WebSockets підтримують дуплексний зв'язок, тобто клієнт і сервер можуть надсилати й отримувати дані одночасно;

- архітектура, керована подіями: WebSockets використовують керовану подіями архітектуру, яка забезпечує ефективний зв'язок і зменшує накладні витрати, пов'язані з протоколами на основі HTTP.

WebSockets – чудовий вибір для цього додатка, оскільки забезпечує двосторонній зв'язок у реальному часі між клієнтом (додатком) і сервером (зарядною станцією). Це означає, що додаток може отримувати оновлення в режимі реального часу про стан зарядної станції, наприклад, поточну швидкість заряджання, розрахунковий час до повного заряджання та будь-які помилки чи попередження, які можуть виникнути під час процесу заряджання.

Крім того, WebSockets може допомогти зменшити затримку та збільшити швидкість передачі даних, оскільки вони дозволяють двосторонній зв'язок із сервером без необхідності клієнту постійно запитувати сервер щодо оновлень.

Загалом використання WebSockets у програмі може значно покращити взаємодію з користувачем, надаючи оновлення в реальному часі та відгуки, а також може допомогти забезпечити плавний та ефективний процес заряджання електромобілів.

Також потрібна бібліотека для інтернаціоналізації додатку. Для цього дуже підходить `i18n`. Бібліотека `i18n` (інтернаціоналізація) була обрана для цієї програми для підтримки кількох мов. Зважаючи на вимогу наявності трьох мов у додатку (української, німецької та англійської), дуже важливо мати інструмент, який може ефективно виконувати переклади. Бібліотека забезпечує ефективний спосіб керування перекладами у `Vue.js`.

Ця бібліотека дозволяє розробникам легко визначати ключі перекладу та відповідні переклади для різних мов. Вона також надає такі функції, як

перемикання локалізації, динамічне оновлення перекладу та форматування повідомлень.

Використання бібліотеки `i18n` гарантує, що програма легко адаптується до різних мов і може використовуватися будь-ким. Це спрощує процес перекладу та підтримки перекладів для програми, роблячи його ефективнішим і знижуючи ризик помилок у процесі перекладу. Крім того, бібліотека `i18n` добре задокументована та має хорошу підтримку спільноти, що робить її надійним вибором для інтернаціоналізації в програмах `Vue.js`.

Висновки до розділу 2

Розроблена система є додатком для зарядки електромобілів, створеним за допомогою `Vue.js`, `Vuetify`, `Websockets`, `i18n`. Метою цієї програми є надання власникам електромобілів зручної та ефективної платформи для використання електрозарядних станцій. Розроблена специфікація окреслює вимоги до розробленої системи та описує методи та прийоми, що використовуються в процесі розробки.

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ ДЛЯ ЗАРЯДКИ ЕЛЕКТОКАРІВ З ВИКОРИСТАННЯМ VUE.JS

3.1 Практична реалізація об'єкта проєктування

У практичній реалізації об'єкта описано фрагменти коду програми за даною темою кваліфікаційної роботи. На рисунку 3.1, зображено стартову панель із вибором роз'ємів з правої сторони розташований qrcode з інструкціями. На панелі знизу розташовано налаштування та мови, які підтримує програма. Вибравши бажану мову, користувач бачить миттєвий переклад програми.

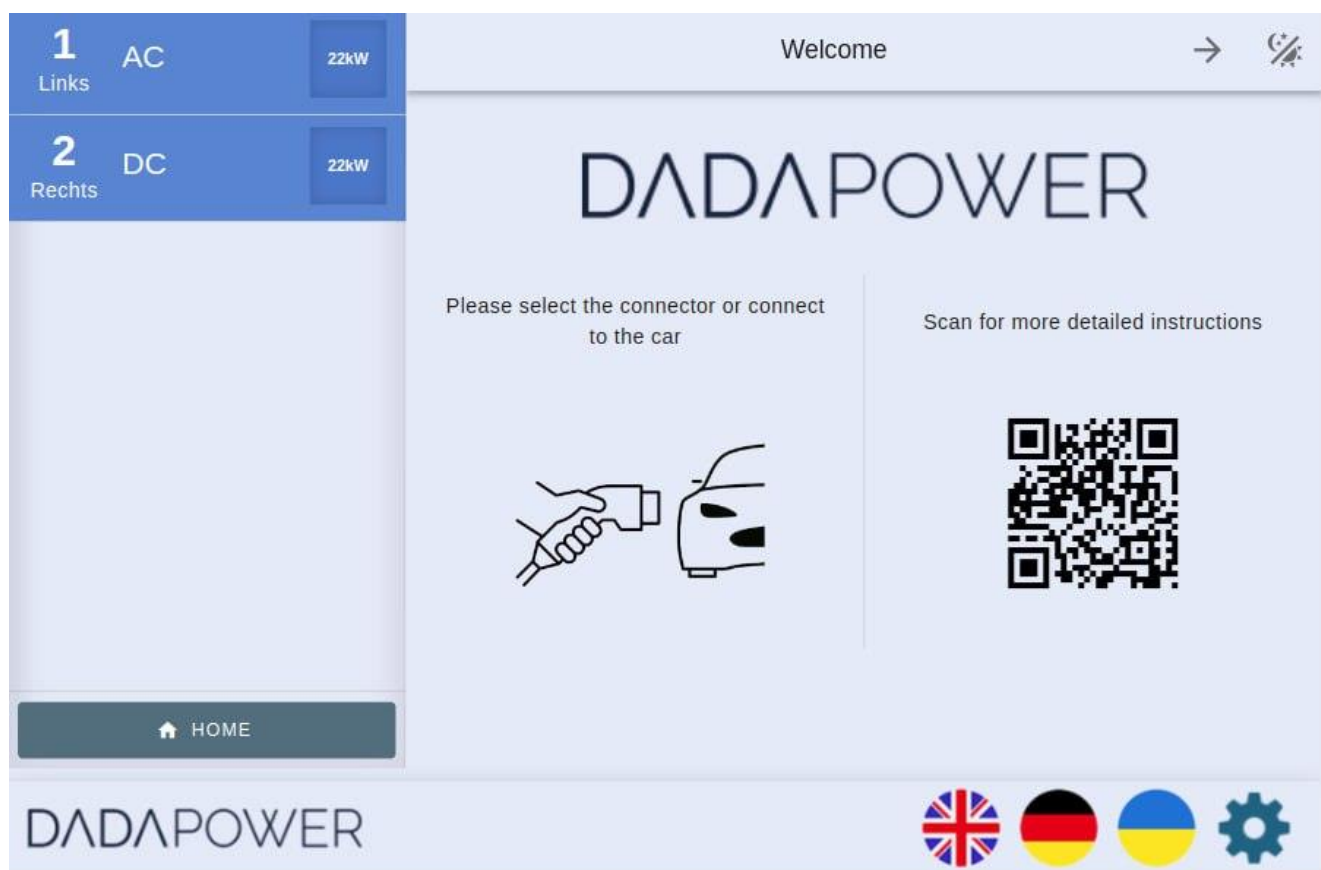


Рисунок 3.1 – Початкова сторінка програми

Потрібно обрати роз'єм та під'єднати до авто. Це можна зробити натиснувши на лівій стороні екрану, де відображено список роз'ємів (рис. 3.2).



Рисунок 3.2 – Список доступних роз'ємів

Код у лістингу 3.1, включає зв'язок через сокет і використовує Vuex для керування станом. Компонент отримує об'єкт «connector» і необов'язковий рядок «sideName», як властивості. Ці атрибути, представляють інформацію про роз'єм. Компонент визначає обчислені властивості, які базуються на методах отримання Vuex. Ці обчислені властивості отримують свої значення від стану компонента та інших даних. Наприклад, «onlyOneConnector» перевіряє, чи є лише один конектор у сховищі Vuex, «activeConnector» отримує активний поле із сховища, «progressStatus» отримує опис стану конектора тощо.

Компонент також імпортує функції «mapGetters» і «mapActions» з Vuex. Ці функції дозволяють компоненту отримувати доступ до гетерів і дій Vuex, відповідно, і відображати їх на обчислені властивості та методи компонента. Компонент включає дію «setActiveConnector» від Vuex як метод компонента. Це вказує на те, що компонент може ініціювати дію «setActiveConnector», яка оновить активний роз'єм у сховищі Vuex. Метою цього компонента, здається, є відображення інформації та керування станом, пов'язаним із роз'ємом, використання Vuex для керування станом і потенційне використання зв'язку через сокет для оновлень у реальному часі.

Лістинг 3.1 – Скрипт компоненти роз'єму

```
import { sendMessage } from "@/utils/socket";
import { mapGetters, mapActions } from "vuex";

export default {
  name: "ConnectorItem",
```

Продовження лістинга 3.1

```

props: {
connector: {
  type: Object,
  required: true,
},
sideName: String,
},
computed: {
  ...mapGetters(["connectors"]),
  onlyOneConnector() {
    return this.connectors.length === 1;
  },
  activeConnector() {
    return this.$store.getters.activeConnector;
  },
  progressStatus() {
    return this.connector.statusDescription;
  },
  isActiveConnector() {
    return this.connector.id === this.activeConnector.id;
  },
  connectorEvseId() {
    return this.connector.evseid;
  },
  errorState() {
    if (this.connector.error_description === "") return "";
    return (
      this.connector.error_code + ": " + this.connector.error_description
    );
  },
},
methods: {
  ...mapActions(["setActiveConnector"]),
},
};

```

кінець лістингу 3.1

Коли користувач обрав певний тип роз'єму буде показано екран (рис. 3.3) із проханням під'єднати до авто та ознайомитись з цінами в оператора.



Рисунок 3.3 – Повідомлення про під'єднання роз'єму

Після фізичного під'єднання буде показано, що авто було під'єднано або виникла помилка (рис. 3.4). Також на цьому етапі, користувач може використати карту з підтримкою NFC. Після чого, користувач може надіслати собі на пошту чек.

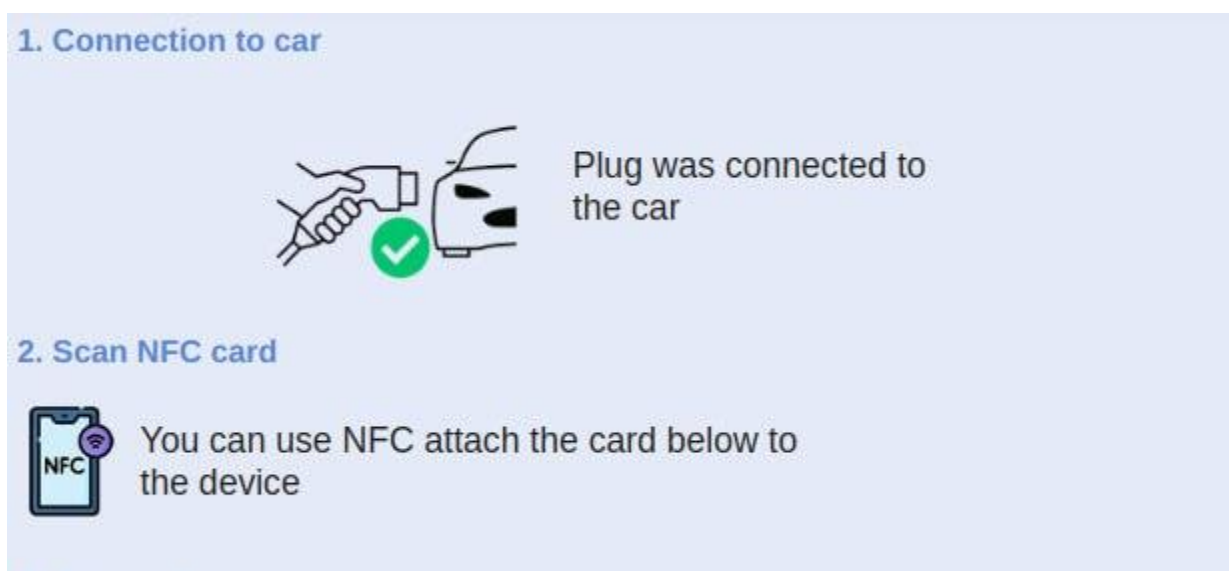


Рисунок 3.4 – Успішне підключення

Важливим функціоналом даної програми – це налаштування. Завдяки їм, користувач може налаштувати під власні потреби. З головної сторінки можна зайти в налаштування. Першим ділом буде продемонстровано інформацію про конкретну станцію (рис. 3.5).

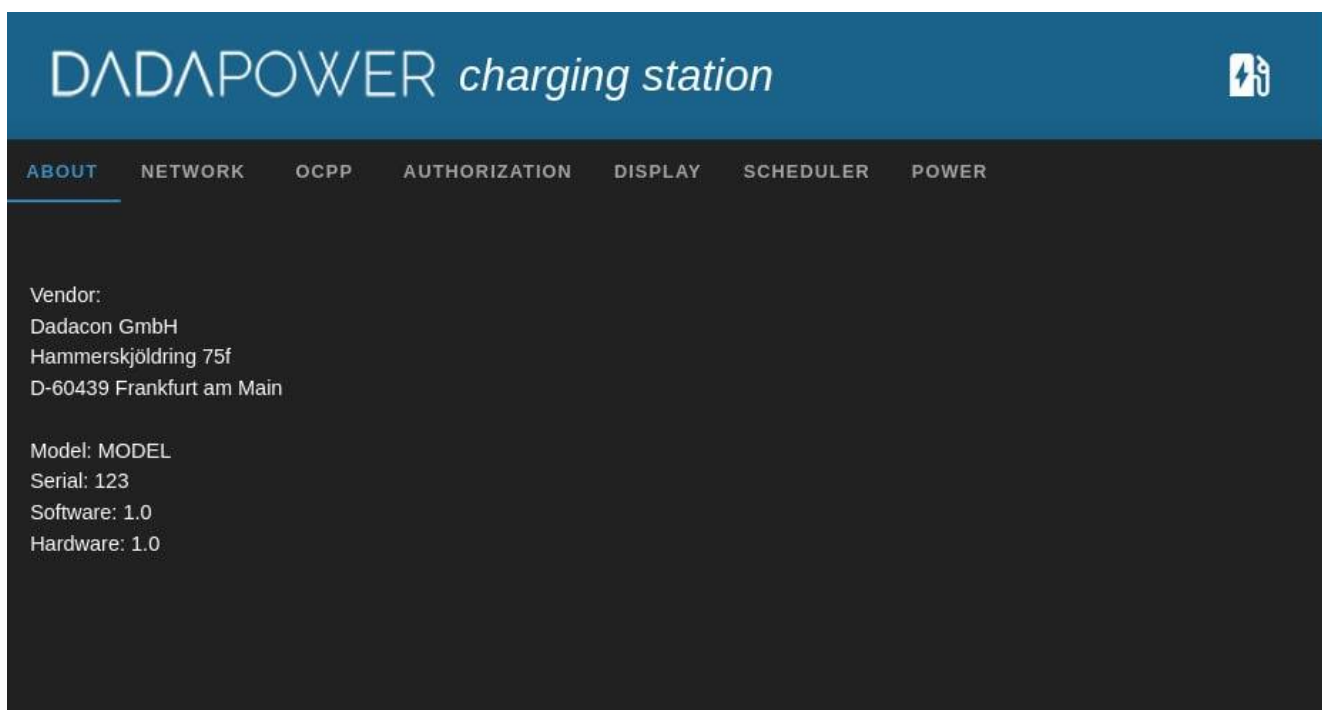


Рисунок 3.5 – Інформація про станцію

Наступною вкладкою є мережа. Через неї можна підключитись до Інтернету через точку доступу Wi-Fi або налаштувати вручну, ввівши дані, які на рисунку 3.6.

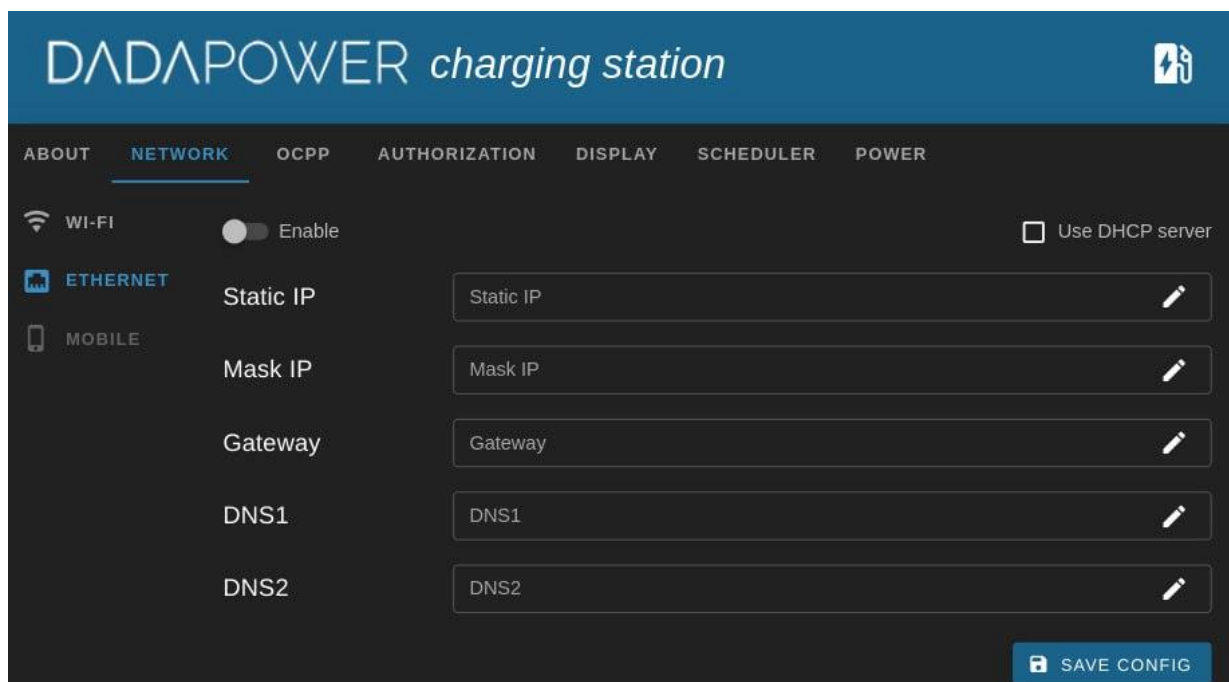
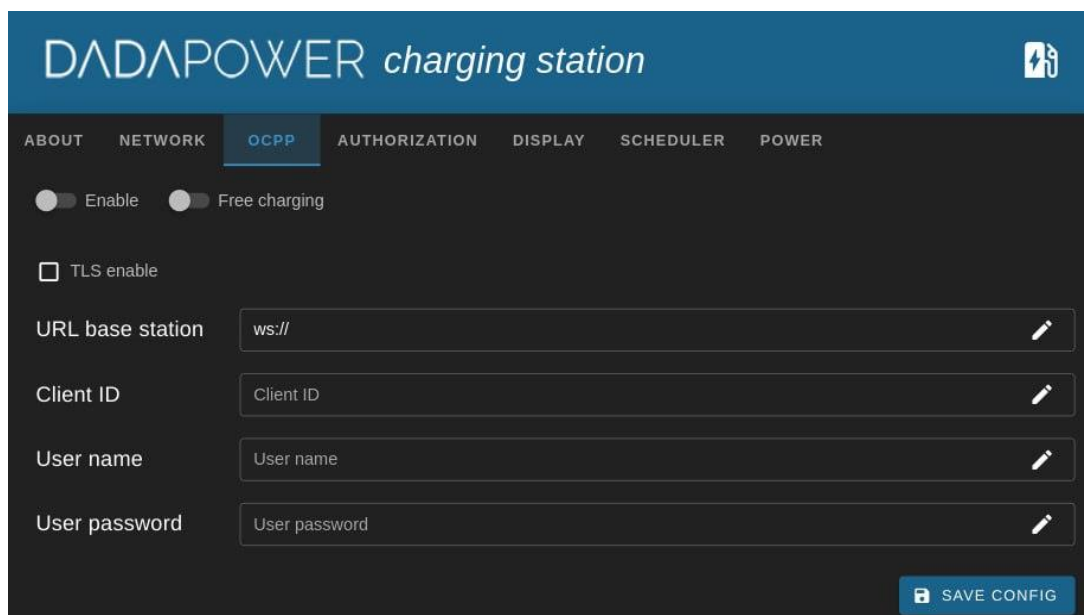


Рисунок 3.6 – Вкладка Network

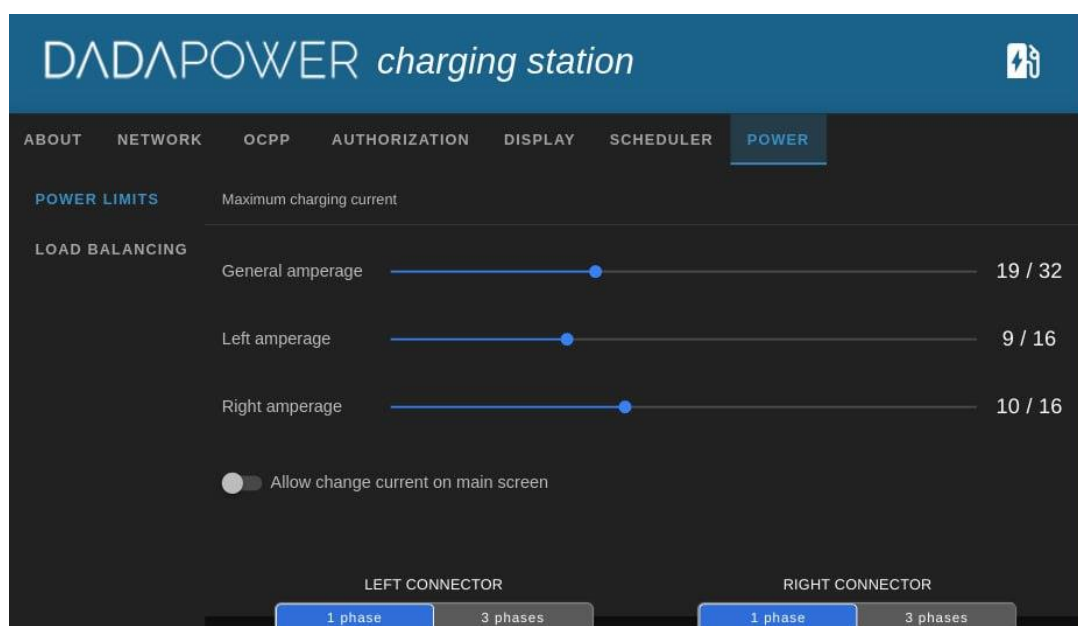
Також можна налаштування безпечне підключення, яке робиться на вкладці OCPP (рис. 3.7).



The screenshot shows the 'DADAPOWER charging station' web interface. The 'OCPP' tab is selected in the top navigation bar. Below the navigation bar, there are two toggle switches: 'Enable' (turned on) and 'Free charging' (turned off). Below these is a checkbox for 'TLS enable' which is unchecked. There are five input fields with edit icons: 'URL base station' (containing 'ws://'), 'Client ID' (containing 'Client ID'), 'User name' (containing 'User name'), and 'User password' (containing 'User password'). A 'SAVE CONFIG' button is located at the bottom right.

Рисунок 3.7 – Вкладка OCPP

Важливим є налаштування потужності, де користувач вибирає певну потужність під авто, що зображено на рисунку 3.8. Налаштовуються ампери та кількість фаз на конектор.



The screenshot shows the 'DADAPOWER charging station' web interface with the 'POWER' tab selected. Under 'POWER LIMITS', there is a label 'Maximum charging current'. Under 'LOAD BALANCING', there are three sliders: 'General amperage' (set to 19 / 32), 'Left amperage' (set to 9 / 16), and 'Right amperage' (set to 10 / 16). Below the sliders is a toggle switch for 'Allow change current on main screen' which is turned off. At the bottom, there are two sections: 'LEFT CONNECTOR' and 'RIGHT CONNECTOR'. Each section has two buttons: '1 phase' (highlighted in blue) and '3 phases' (grey).

Рисунок 3.8 – Вкладка Power

На лістингу 3.2, продемонстровано фрагмент коду, який містить кілька методів, які використовуються для обробки оновлень сервера, застосування конфігурації сервера та оновлення даних про обмеження потужності.

Метод «subscribeForServerUpdates» відстежує зміни, пов'язані з обмеженнями потужності та застосовує нову конфігурацію, якщо вона відрізняється від поточної. Метод призначає оновлені обмеження потужності з корисного навантаження об'єкту «powerLimits» компонента та викликає метод «applyServerConfig» для застосування змін.

Метод «applyServerConfig» відповідає за застосування конфігурації сервера до даних компонента. Якщо є лише один роз'єм («onlyOneConnector» має значення true), він викликає метод «applyConfigForOneConnector». В іншому випадку він викликає метод «applyConfigForConnectors».

Метод «applyConfigForConnectors» застосовує конфігурацію сервера до окремих з'єднувачів. Він оновлює значення «amperageSliders» і «connectorsPhases» на основі обмежень потужності, отриманих від сервера.

Метод «applyConfigForOneConnector» застосовує конфігурацію сервера, коли є лише один роз'єм. Він оновлює значення повзунка сили струму та фаз для одного роз'єму.

Метод «onAmperageSliderMove(slider, index)» викликається під час переміщення повзунка сили струму. Якщо індекс дорівнює 0, він оновлює властивість «overall_max_current» для «powerLimit». В іншому випадку він знаходить відповідний роз'єм у «powerLimit».

Метод «updatePowerLimitsData» відповідає за надсилання оновлених даних про обмеження потужності на сервер. Він готує дані шляхом виклику методу «configureSendData» і надсилає їх за допомогою функції «sendSocketMessage».

Метод «configureSendData» налаштовує дані для надсилання на сервер на основі поточних обмежень потужності та кількості конекторів.

Лістинг 3.2 – Методи компоненти регулювання потужності зарядки

```

subscribeForServerUpdates() {
  this.$store.subscribe(({ type, payload }) => {
    if (type !== 'setPowerLimits' || isEqual(cloneDeep(this.powerLimits),
cloneDeep(payload))) return;

    Object.assign(this.powerLimits, payload);
    this.applyServerConfig();
  });
},

applyServerConfig() {
  if (this.onlyOneConnector) this.applyConfigForOneConnector();
  else this.applyConfigForConnectors();
},

applyConfigForConnectors() {
  this.amperageSliders[0].value = this.powerLimits.overall_max_current;
  this.amperageSliders[0].max = this.powerLimits.overall_max_current_limit;
  this.amperageSliders[0].min = this.powerLimits.overall_min_current_limit;

  this.powerLimits.connectors.forEach(c => {
    this.amperageSliders[c.id].value = c.max_current;
    this.amperageSliders[c.id].max = c.max_current_limit;
    this.amperageSliders[c.id].min = c.min_current_limit;

    this.connectorsPhases[parseInt(c.id) - 1].value = c.use_three_phase;
  });
},

applyConfigForOneConnector() {
  const [c] = this.powerLimits.connectors;

  this.amperageSliders[0].value = c.max_current;
  this.amperageSliders[0].max = c.max_current_limit;
  this.amperageSliders[0].min = c.min_current_limit;
  this.amperageSliders.splice(1, this.amperageSliders.length);

  this.connectorsPhases[0].value = c.use_three_phase;
  this.connectorsPhases.splice(1, this.connectorsPhases.length);
},

onConnectorPhaseToggle(value, toggle, index) {

```

Продовження лістинга 3.2

```

toggle.value = value;
const idx = this.powerLimits.connectors.findIndex(c => c.id === index + 1);

if (idx < 0) throw new Error('No connector index found to update');

this.powerLimits.connectors[idx].use_three_phase = value;
},

onAmperageSliderMove(slider, index) {
  if (index === 0) {
    this.powerLimits.overall_max_current = slider.value;
    return;
  }
  const idx = this.powerLimits.connectors.findIndex(c => c.id === index);

  if (idx < 0) throw new Error('No connector index found to update');

  this.powerLimits.connectors[idx].max_current = slider.value;
},

updatePowerLimitsData() {
  const data = this.configureSendData();
  sendSocketMessage({ type: 'ChargePointPowerLimits', data });
},
configureSendData() {
  if (this.onlyOneConnector) return {
    ...this.powerLimits,
    connectors: [
      {
        ...this.powerLimits.connectors[0],
        max_current: this.powerLimits.overall_max_current,
        max_current_limit: this.powerLimits.overall_max_current_limit,
        min_current_limit: this.powerLimits.overall_min_current_limit,
      },
    ],
  };

  return this.powerLimits;
}},

```

кінець лістингу 3.2

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Під час розробки програми зарядної станції для електромобілів було проведено ретельні процеси тестування та налагодження, щоб переконатися в її функціональності, надійності та зручності для користувача. Для виявлення та усунення потенційних проблем і помилок використовувалися різні методології та техніки тестування. Етап тестування охоплював різні аспекти інформаційної та комп'ютерної системи, включаючи програмне забезпечення, апаратне забезпечення та загальну інтеграцію системи.

Модульне тестування було проведено для перевірки функціональності окремих програмних компонентів, таких як інтерфейс користувача, модулі обробки даних і протоколи зв'язку. Це включало написання тестів і їх виконання для перевірки очікуваної поведінки кожного компонента. Мета полягала в тому, щоб кожен модуль точно та ефективно виконував покладені на нього завдання.

Через модульний тест по типам конекторів потрібно було переконатися, що всі доступні типи роз'ємів правильно відображаються в інтерфейсі користувача. В результаті меню вибору роз'єму відображало всі очікувані типи роз'ємів: тип 1, тип 2, CHAdeMO, CCS і Tesla Supercharger. Кожен тип роз'єму був правильно позначений і відповідав відповідному порту зарядки.

Далі потрібно перевірити відображення статусу підключення на основі стану заряджання. В результаті, коли зарядну станцію було підключено до тестового автомобіля, статус підключення точно відображався як «Підключено» в інтерфейсі користувача. Після відключення статус змінився на «Не підключено» в режимі реального часу. Після чого потрібно було переконатись, що під час сеансу заряджання точне відображення стану, часу, що залишився, і рівня потужності.

Також було проведено тестування платіжної транзакції з використанням даних тестової картки. Було успішно оброблено із зашифрованим зв'язком між програмою та платіжним шлюзом. Методи автентифікації, такі як ім'я користувача/пароль і двофакторна автентифікація, працювали належним чином, забезпечуючи захист особистої та фінансової інформації користувачів.

Ці змодельовані результати модульного тестування свідчать про те, що програма зарядної станції пройшла тести, проведені для різних функцій, включаючи типи роз'ємів, статус підключення, стан заряджання та дисплей, заходи безпеки та налаштування бездротового підключення до Інтернету. Ці позитивні результати демонструють дотримання програми визначеними вимогами та її загальну функціональність і надійність.

Тестування інтеграції було проведено для оцінки взаємодії та сумісності між різними програмними модулями та зовнішніми компонентами, такими як апаратне забезпечення зарядної станції, внутрішній сервер і система баз даних. Це включало моделювання реальних сценаріїв та оцінку відповіді системи та цілісності даних. За допомогою тестування інтеграції було виявлено та виправлено будь-які проблеми, пов'язані з протоколами зв'язку, синхронізацією даних або сумісністю.

Було проведено наскрізне тестування, щоб підтвердити загальну функціональність і продуктивність програми зарядної станції для електромобілів. Це включало тестування всього процесу користувача, від початку сеансу заряджання до моніторингу ходу заряджання та доступу до різних налаштувань. Змодельовавши реалістичні користувацькі сценарії, було оцінено швидкість реагування, надійність і точність програми в наданні інформації про стягнення плати в реальному часі.

Було проведено тестування на сумісність, щоб переконатися, що програма бездоганно працює на платформі і браузері (рис. 3.9). Метою було забезпечити взаємодію з користувачем і функціональність на кількох платформах.

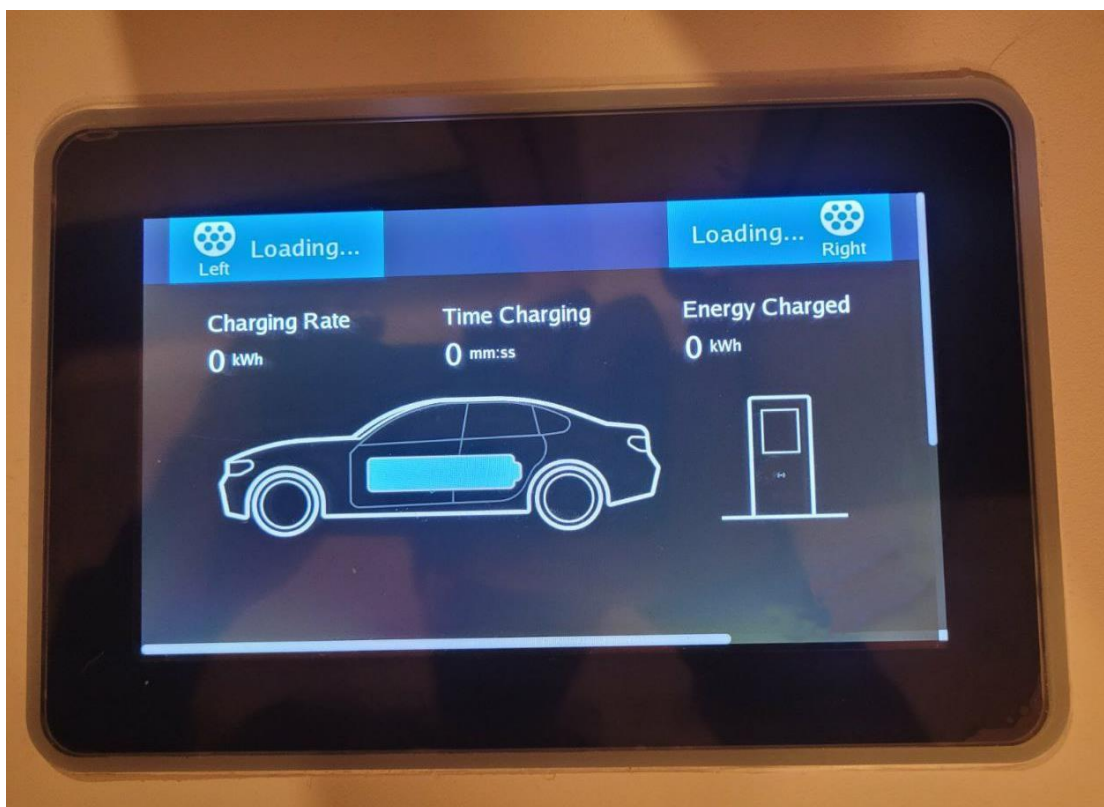


Рисунок 3.9 – Проведення тестування інтерфейсу першої версії

Проводилися опитування серед користувачів та аналіз відгуків, щодо першої версії. Початковий дизайн, дещо не зрозумілий для користувачів, тому потрібно було внести зміни, щодо відображення інтерфейсу.

Крім того, було проведено стрес-тестування та тестування навантаження, щоб оцінити продуктивність програми за сценаріїв інтенсивного використання. Це передбачало піддавання системи значній кількості одночасних запитів на стягнення плати для оцінки її масштабованості та швидкості реагування. За допомогою цього тестування, було виявлено та оптимізовано будь-які вузькі місця або проблеми з продуктивністю.

Протягом усього етапу тестування було впроваджено комплексні журнали та механізми відстеження помилок для фіксації та аналізу будь-яких виявлених проблем. Це сприяло ефективному налагодженню та вирішенню виявлених проблем.

3.3 Розробка документації для програмного продукту

Відомості про зарядну інфраструктуру та джерело живлення для зарядної станції.

Інфраструктура зарядки:

- зарядні станції: кілька зарядних станцій із можливостями рівня 2, кожна з яких оснащена роз'ємом типу 2 для сумісності з більшістю електромобілів. Система має відображати статус підключення, щоб вказати, чи успішно підключено автомобіль до зарядної станції чи ні. Ця інформація гарантує користувачам і допомагає усунути будь-які проблеми з підключенням. Статус підключення можна відобразити за допомогою візуальних індикаторів, таких як значок або відповідний текст. Дисплей стану підключення має надавати чітку інформацію, наприклад «Підключено» або «Не підключено». Якщо транспортний засіб успішно підключено, користувач може продовжити процес заряджання. Якщо з'єднання не встановлено, користувачеві може знадобитися змінити положення або перевірити зарядний кабель на наявність проблем. Крім того, система може надавати вказівки або інструкції щодо того, як правильно підключити автомобіль для тих, хто вперше користується системою;

- роз'єми. Система повинна дозволяти користувачам вибирати з різних типів на основі зарядного порту свого електромобіля. Доступні типи роз'ємів можуть включати тип 1 (J1772), тип 2 (Mennekes), CHAdeMO, CCS (комбінована система зарядки), Tesla Supercharger та інші [11]. Кожен тип має свій унікальний дизайн і характеристики. Коли користувач прибуває на зарядну станцію, йому потрібно буде визначити відповідний тип роз'єму для свого автомобіля. Зарядна станція повинна мати чітке маркування або етикетки, що вказують на різні доступні типи роз'ємів. Потім користувач фізично підключає зарядний кабель свого автомобіля до відповідного роз'єму на зарядній станції. Вони призначені для забезпечення безпечного та надійного з'єднання між автомобілем і зарядною станцією;

– рівні зарядної потужності. Зарядні станції рівня 2, здатні видавати потужність 7,2 кВт, щоб забезпечити ефективну зарядку більшості авто. Крім того, станція швидкої зарядки постійного струму рівня 3 потужністю 50 кВт дозволяє швидко заряджати сумісні електромобілі. Дисплей рівня потужності показує швидкість, з якою заряджається автомобіль. Він може бути представлений у кіловатах (кВт) або у відсотках від максимальної потужності заряджання. Ця інформація допомагає користувачам зрозуміти швидкість заряджання та відповідно налаштувати.

Джерело живлення:

– підключення до електричної мережі. Спеціальне підключення до трифазної електричної мережі з потужністю щонайменше 50 кВт, що відповідає місцевим електричним нормам;

– блок розподілу живлення (PDU): здатний розподіляти електроенергію на зарядні станції, керувати балансуванням навантаження та контролювати споживання електроенергії. PDU має підтримувати максимальне навантаження 50 кВт і забезпечувати підключення до даних для віддаленого моніторингу та керування;

– потужність. Загальна потужність зарядної станції повинна становити 50 кВт, що дозволяє одночасно заряджати кілька транспортних засобів із високою швидкістю заряджання;

– управління живленням і балансування навантаження: PDU має містити інтелектуальні алгоритми балансування навантаження, щоб забезпечити оптимальний розподіл електроенергії між зарядними станціями, уникаючи перевантаження та максимізуючи використання енергії;

– резервне живлення: інтегрована система безперебійного живлення (UPS) потужністю 20 кВт для забезпечення резервного живлення під час відключень електроенергії та забезпечення безперебійної зарядки;

– підключення до Інтернету. Щоб увімкнути бездротове підключення до Інтернету, користувач може отримати доступ до меню налаштувань у додатку або в інтерфейсі зарядної станції. У налаштуваннях користувач може знайти

налаштування Wi-Fi. У налаштуваннях бездротового підключення до Інтернету користувач побачить список доступних мереж Wi-Fi поблизу. Користувач може вибрати потрібну мережу та, якщо потрібно, ввести мережевий пароль або пройти автентифікацію за допомогою інших методів автентифікації для конкретної мережі. Після успішного підключення користувача до бездротової мережі система встановить підключення до Інтернету. Це підключення дозволяє користувачеві отримати доступ до додаткових онлайн-функцій, таких як оновлення даних у реальному часі, історія стягнень, платіжні операції та інтеграція з іншими службами чи платформами;

– безпека. Для забезпечення безпеки система може включати різні методи та протоколи, такі як Open Charge Point Protocol (OCPP). Це широко використовуваний протокол зв'язку для зарядних станцій і центральних систем керування. Це забезпечує безпечний і стандартизований зв'язок між зарядною станцією та серверною системою. Крім OCPP, система може впроваджувати інші заходи безпеки, включаючи шифрування конфіденційних даних, безпечні механізми автентифікації та безпечні платіжні шлюзи. Автентифікацію та авторизацію користувача можна виконати за допомогою таких методів, як комбінація пошти користувача та пароля. Крім того, система має відповідати відповідним стандартам безпеки та найкращим практикам, таким як шифрування даних у стані спокою та під час передачі, регулярні перевірки безпеки та оновлення, а також дотримання нормативних вимог щодо захисту особистої та фінансової інформації користувачів.

Апаратні компоненти:

- сервер: для розміщення програмного забезпечення, яке керує роботою зарядної станції, потрібен виділений сервер. Він повинен відповідати мінімальним вимогам до програмних компонентів, що використовуються;
- пам'ять: має бути достатньо оперативної пам'яті для виконання вимог програми до обробки та зберігання. Рекомендується мінімум 4 ГБ оперативної пам'яті;
- зберігання: для зберігання таких даних, як журнали сеансу зарядки, інформація про користувача та конфігурації системи, потрібен достатній простір для зберігання. Для швидшого доступу до даних краще використовувати SSD диск;
- процесор: для ефективної обробки обчислювального навантаження рекомендується сучасний багатоядерний процесор із тактовою частотою 2 ГГц або вище;
- мережеве обладнання: мережеве обладнання, таке як маршрутизатори, комутатори та кабелі, потрібне для встановлення з'єднання між зарядною станцією, сервером та іншими мережевими пристроями;
- системи захисту: для забезпечення електричної безпеки слід встановити засоби безпеки, включаючи пристрої захисту від перенапруги, вимикачі замикання на землю (GFCI).

Висновки до розділу 3

В цьому розділі було продемонстровано практичну реалізацію об'єкта проектування і тестування та налагодження інформаційно-комп'ютерної системи, виявлено помилки і їх усунення. Також була розроблена документація для програми під конкретну специфіку додатку.

ВИСНОВКИ

За результатами проведеного дослідження варто зробити наступні висновки, що дана робота, передбачала розробку програми з використанням передових технологій для задоволення потреб сфери зарядки електромобілів.

Додаток, розроблений під час цього проєкту, пропонує комплексне рішення для моніторингу стану зарядки електромобіля. Використовуючи такі технології, як Vue.js, Vuetify, SQL, i18n, WebSockets і Vuex, програма надає інформацію в режимі реального часу про стан зарядки, рівні потужності та різні параметри, характерні для різних роз'ємів електромобілів.

Практична реалізація об'єкта проєктування демонструє ефективність обраних технологій і методик. Використання Vue.js і Vuetify сприяло розробці динамічного та інтуїтивно зрозумілого інтерфейсу користувача, забезпечуючи безперебійну роботу користувача.

Інтеграція WebSockets забезпечує миттєве оновлення та зв'язок у режимі реального часу між додатком і зарядними станціями, забезпечуючи точну та актуальну інформацію. Крім того, підтримка i18n дозволяє легко локалізувати та інтернаціоналізувати, обслуговуючи користувачів із різних регіонів.

Протягом цієї роботи виконувалися різні завдання, щоб забезпечити його завершення. Ці завдання включали аналіз релевантності програми, вибір відповідних технологій та інструментів реалізації, створення технічного завдання, функціональний опис програми та визначення можливостей користувача. Також ретельно проводилась розробка інтерфейсу програми, налаштування середовища розробки та технологій, кодування, ретельне тестування.

Під час роботи було виконано конкретні поставлені завдання:

- проведено аналіз актуальності додатку;
- вибрано технології та засоби реалізації для створення проєкту, такі, як: Vue, Vuetify, Webscokets, i18n, SCSS.
- написання технічного завдання;
- перевірено дію написаного функціоналу;

- описано можливості комунікації користувачів із програмою;
- створено інтерфейс додатку, який сумісний із платформою та браузером;
- налаштовано середовище розробки та технології;
- написання коду для проєкту, що виконує поставлені завдання;
- проведено тестування додатку;
- розроблено документацію для програмного продукту.

Розроблена програма є свідченням інтеграції Vue.js, Vuetify, SQL, i18n, WebSockets і Vuex, демонструючи їхні можливості у створенні надійної та зручної системи моніторингу зарядки електромобілів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до виконання кваліфікаційної роботи бакалавра : метод. рек. / уклад.: Н. М. Ліщина, А. А. Ящук, О. М. Суринович. Луцьк, 2022.56 с.
2. Electrify America URL: <https://www.electrifyamerica.com/about-us/> (дата звернення: 25.04.2023).
3. Electrify America rebrands 350kW and 150kW chargers URL: <https://electrek.co/2022/09/08/electrify-america-branding-chargers-hyper-fast-ultra-fast-new-naming-standard/> (дата звернення: 25.04.2023).
4. Kempower URL: <https://kempower.com> (дата звернення: 25.04.2023).
5. Про компанію / UGV Chargers URL: <https://ugv.ua/uk/pro-kompaniyu/> (дата звернення: 25.04.2023).
6. EVgo Announces Opening of 800th EVgo Fast Charging Location URL: <https://www.prnewswire.com/news-releases/evgo-announces-opening-of-800th-evgo-fast-charging-location-301014046.html> (дата звернення: 25.04.2023).
7. Smart Electric Vehicle Charging in the Era of Internet of Vehicles, Emerging Trends, and Open Issues URL: <https://www.mdpi.com/1996-1073/15/5/1908> (дата звернення: 06.05.2023).
8. Vue.js документація URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 06.05.2023).
9. Vuetify документація, вступ / чому саме Vuetify URL: <https://vuetifyjs.com/en/introduction/why-vuetify/> (дата звернення: 25.04.2023).
10. Advantages and disadvantages of Websocket URL: <https://www.wallarm.com/what/websocket-vs-http-how-are-these-2-different> (дата звернення: 08.05.2023).
11. EV charging connector types URL: <https://www.power-sonic.com/blog/ev-charging-connector-types/> (дата звернення: 08.05.2023).
12. About Electrify America URL: https://en.wikipedia.org/wiki/Electrify_America (дата звернення: 08.05.2023).

13. Build a Basic CRUD App with Vue.js and Node and Sequelize URL: <https://developer.okta.com/blog/2018/02/15/build-crud-app-vuejs-node> (дата звернення: 08.05.2023).
14. Electric vehicle charging infrastructure | Guidelines for cities URL: <https://cleantechnica.com/files/2018/04/EV-Charging-Infrastructure-Guidelines-for-Cities.pdf> (дата звернення: 11.05.2023).
15. Handbook of electric vehicle charging infrastructure implementation URL: <https://www.niti.gov.in/sites/default/files/202108/HandbookforEVChargingInfrastructureImplementation081221.pdf> (дата звернення: 11.05.2023).
16. How To Manage State in a Vue.js Application with Vuex URL: <https://www.digitalocean.com/community/tutorials/how-to-manage-state-in-a-vue-js-application-with-vuex> (дата звернення: 16.05.2023).
17. Review of Electric Vehicle Charger Cybersecurity Vulnerabilities, Potential Impacts, and Defenses URL: <https://www.mdpi.com/1996-1073/15/11/3931> (дата звернення: 16.05.2023).
18. Sequence Diagram Tutorial – Complete Guide with Examples URL: <https://creately.com/guides/sequence-diagram-tutorial/> (дата звернення: 17.05.2023).
19. Types Of Software Testing: URL: <https://www.softwaretestinghelp.com/types-of-software-testing/> (дата звернення: 06.05.2023).
20. Vue.js 2 Cookbook: Build modern, interactive web applications with Vue.js by Andrea Passaglia (дата звернення: 18.05.2023).
21. Vue i18n: Building a multi-language app with locale switcher URL: <https://lokalise.com/blog/vue-i18n/> (дата звернення: 19.04.2023).
22. Vue I18n Internationalization plugin for Vue.js URL: <https://vue-i18n.intlify.dev> (дата звернення: 19.05.2023).

ДОДАТКИ

ДОДАТОК А

КОМПОНЕНТ «PowerLimitsSettingsTab»

```

<template>
  <div class="pb-2">
    <v-subheader>Maximum charging current</v-subheader>
    <v-divider />

    <div class="mx-4 my-2">
      <div
        v-for="(slider, index) in amperageSliders"
        :key="index"
        class="d-flex align-center pt-1 mb-5"
      >
        <v-slider
          v-model="slider.value"
          :min="slider.min"
          :max="slider.max"
          hide-details
          ticks
          @change="onAmperageSliderMove(slider, index)"
        >
          <template v-slot:label>
            <div class="amperage-slider__label">{{ slider.title }}</div>
          </template>
          <template v-slot:thumb-label="{ value }">
            <div class="text-subtitle-2">{{ value }}</div>
          </template>
        </v-slider>

        <div
          style="width: 80px"
          class="d-flex justify-center pl-3 text-h6"
        >
          <span>{{ slider.value }} / {{ slider.max }}</span>
        </div>
      </div>

    <div class="mx-4">
      <div class="d-flex">
        <v-switch
          v-model="powerLimits.allow_change_current_main_screen"

```

```

      :label="$t('settings.power.power_limits.allow_current_change')"
    />
  </div>
  <div
    v-if="false"
    class="d-flex"
  >
    <v-switch
      v-model="powerLimits.allow_change_phases_main_screen"
      :label="$t('settings.power.power_limits.allow_switch_phases')"
    />
  </div>
</div>

<div
  class="d-flex px-4 pt-3"
  :class="onlyOneConnector ? 'justify-start' : 'justify-space-around'"
>
  <div
    v-for="(toggle, index) in connectorsPhases"
    :key="index"
    class="connector-phases-wrapper"
  >
    <div class="d-flex justify-center text-uppercase text-body-2 mb-2">
      <span>{{ onlyOneConnector ? 'Phases' : toggle.title }}</span>
    </div>

    <PhasesToggle v-model="toggle.value"
    @change="onConnectorPhaseToggle($event, toggle, index)" />
  </div>
</div>
</div>
</template>

<script>
import { sendSocketMessage } from '@/utils/socket';
import { mapGetters } from 'vuex';
import { isEqual, cloneDeep } from 'lodash';

export default {
  name: 'PowerLimitsSettingsTab',
  components: {
    PhasesToggle: () => import('@/components/basic/PhasesToggle.vue'),
  },
  props: {

```

```

powerLimits: {
  type: Object,
  required: true,
},
},
data() {
return {
  amperageSliders: [
    {
      title: this.$t('settings.power.power_limits.general_amperage'),
      min: 12,
      max: 32,
      value: 32,
    },
    {
      title: this.$t('settings.power.power_limits.left_amperage'),
      min: 6,
      max: 16,
      value: 6,
    },
    {
      title: this.$t('settings.power.power_limits.right_amperage'),
      min: 6,
      max: 16,
      value: 16,
    },
  ],
},
connectorsPhases: [
  {
    title: this.$t('settings.power.power_limits.left_connector'),
    value: false,
  },
  {
    title: this.$t('settings.power.power_limits.right_connector'),
    value: false,
  },
],
};
},
computed: {
  ...mapGetters([
    'connectors',
  ]),

```

```

onlyOneConnector() {
  return this.connectors.length === 1;
},
},

watch: {
  powerLimits: {
    deep: true,
    handler() {
      this.updatePowerLimitsData();
    },
  },
},

beforeMount() {
  this.applyServerConfig();
  this.subscribeForServerUpdates();
},

methods: {
  subscribeForServerUpdates() {
    this.$store.subscribe(({ type, payload }) => {
      if (type !== 'setPowerLimits' || isEqual(cloneDeep(this.powerLimits),
cloneDeep(payload))) return;

      Object.assign(this.powerLimits, payload);
      this.applyServerConfig();
    });
  },

  applyServerConfig() {
    if (this.onlyOneConnector) this.applyConfigForOneConnector();
    else this.applyConfigForConnectors();
  },

  applyConfigForConnectors() {
    this.amperageSliders[0].value = this.powerLimits.overall_max_current;
    this.amperageSliders[0].max = this.powerLimits.overall_max_current_limit;
    this.amperageSliders[0].min = this.powerLimits.overall_min_current_limit;

    this.powerLimits.connectors.forEach(c => {
      this.amperageSliders[c.id].value = c.max_current;
      this.amperageSliders[c.id].max = c.max_current_limit;
      this.amperageSliders[c.id].min = c.min_current_limit;
    });
  }
}

```

```

    this.connectorsPhases[parseInt(c.id) - 1].value = c.use_three_phase;
  });
},

```

```

applyConfigForOneConnector() {
  const [c] = this.powerLimits.connectors;

```

```

    this.amperageSliders[0].value = c.max_current;
    this.amperageSliders[0].max = c.max_current_limit;
    this.amperageSliders[0].min = c.min_current_limit;
    this.amperageSliders.splice(1, this.amperageSliders.length);

```

```

    this.connectorsPhases[0].value = c.use_three_phase;
    this.connectorsPhases.splice(1, this.connectorsPhases.length);
  },

```

```

onConnectorPhaseToggle(value, toggle, index) {
  toggle.value = value;
  const idx = this.powerLimits.connectors.findIndex(c => c.id === index + 1);

```

```

  if (idx < 0) throw new Error('No connector index found to update');

```

```

  this.powerLimits.connectors[idx].use_three_phase = value;
},

```

```

onAmperageSliderMove(slider, index) {
  if (index === 0) {
    this.powerLimits.overall_max_current = slider.value;
    return;
  }

```

```

  const idx = this.powerLimits.connectors.findIndex(c => c.id === index);

```

```

  if (idx < 0) throw new Error('No connector index found to update');

```

```

  this.powerLimits.connectors[idx].max_current = slider.value;
},

```

```

updatePowerLimitsData() {
  const data = this.configureSendData();
  sendSocketMessage({ type: 'ChargePointPowerLimits', data });
},

```

```

configureSendData() {
  if (this.onlyOneConnector) return {
    ...this.powerLimits,

```

```
connectors: [  
  {  
    ...this.powerLimits.connectors[0],  
    max_current: this.powerLimits.overall_max_current,  
    max_current_limit: this.powerLimits.overall_max_current_limit,  
    min_current_limit: this.powerLimits.overall_min_current_limit,  
  },  
],  
};  
  
return this.powerLimits;  
},  
},  
};  
</script>
```