

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інтелектуальна робототехнічна система керування
паркуванням та зарядними станціями електромобілів у смарт-місті»**

Виконала:

Студентка IV курсу, групи ІК-11

Кудінова Анастасія Олексіївна

Керівник:

Доцент кафедри ІСТ, к.т.н.

Крилов Є.В.

Рецензент:

Доцент кафедри ІІІ, к.т.н, доцент

Лісовиченко О.І.

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ
на дипломний проєкт студентці
Кудіновій Анастасії Олексіївні

1. Тема проєкту «Інтелектуальна робототехнічна система керування паркуванням та зарядними станціями електромобілів у смарт-місті», керівник проєкту Крилов Євген Володимирович, доцент кафедри ІСТ, к.т.н, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «9» червня 2025 року
3. Вихідні дані до проєкту: система повинна забезпечувати автоматизований процес керування паркувальними місцями та зарядними станціями і отримувати заброньовані пакромісця та зарядні електростанції закріплених за певним користувачем, систем проводить моніторинг в даний паркувальної площі і поєднує це з клієнтською частиною.
4. Зміст пояснювальної записки: опис предметної області, аналіз існуючих рішень та обґрунтування обраної архітектури системи, вибір програмного забезпечення та створення бази даних, розробка системи та опис особливостей реалізації

5. Перелік графічного матеріалу (діаграма прецедентів, архітектура контексту база даних, реалізація вебзастосунку, алгоритм контролю доступу)

6. Дата видачі завдання 12.02.2025

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	14.10.24 - 30.10.24	
2	Аналіз існуючих рішень	31.10.24 - 22.11.24	
3	Постановка задачі	23.11.24 - 17.12.24	
4	Огляд існуючих рішень та їх вирішення	18.12.24 – 29.01.25	
5	Створення алгоритму системи	30.01.25 – 03.03.25	
6	Написання коду програми	04.03.25 – 10.05.25	
7	Тестування підсистеми	11.05.25 – 30.05.25	

Студентка

Анастасія КУДІНОВА

Керівник

Євген КРИЛОВ

АНОТАЦІЯ

Пояснювальна записка: 60 ст., 30 рисунків, 2 таблиці, 20 джерел.

Інтелектуальна робототехнічна система керування паркуванням та зарядними станціями електромобілів у смарт-місті

Мета дипломного проєкту: розробка та тестування інтелектуальної робототехнічної системи, яка представляє собою систему керування паркувальними місцями та зарядним станцією, використання, якої представляє собою концепцію Smart City на базі технології Інтернет речей.

У першому розділі описується предметна область та її характеристика. Проводиться аналіз існуючих рішень та їх вирішення та відбувається опис основного бізнес процесу, який підлягає автоматизації. Можливі функціональні рішення системи та огляд алгоритмів підсистеми.

У другому розділі обґрунтовується обрана архітектурна системи. Проводиться аналіз існуючих систем та принцип їх роботи, порівняння реалізації та вибір архітектурного підходу.

У третьому розділі описується стек технології, вибір мов програмування, архітектура контексту бази даних та робоча логіка системи з використанням фреймворків.

У четвертому розділі реалізується розроблена вебсистема з компонентою локальної програми для управління паркінгом та проводиться тестування програми.

Список ключових слів: робототехнічна система, паркування, зарядні станції, база даних, mysql, C#.

SUMMARY

Explanatory note: 60 pages. 30 figures, 2 tables 20 sources.

Intelligent robotic system for controlling parking and charging stations for electric vehicles in a smart city.

The purpose of the diploma project: development and testing of an intelligent robotic system, which is a parking space and charging station management system, the use of which represents the Smart City concept based on Internet of Things technology.

The first section describes the subject area and its characteristics. An analysis of existing solutions and their solutions is carried out and the main business process to be automated is described. Possible functional solutions of the system and an overview of the subsystem algorithms are presented.

The second section justifies the chosen architectural system. An analysis of existing systems and their working principle is carried out, implementation comparisons are made, and an architectural approach is chosen.

The third section describes the technology stack, the choice of programming languages, the database context architecture, and the operating logic of the system using frameworks.

In the fourth section, the developed web system with a local application component for parking management is implemented and the application is tested.

List of keywords: robotic system, parking, charging stations, database, mysql, C#.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			Документація загальна			
2						
3			Знову розроблена			
4	A4	ІК11.150БАК.006 ПЗ	Пояснювальна записка	60		
5			Інтелектуальна	1		
6	A3	ІК11.150БАК.006 Д1	робототехнічна система			
7			керування паркуванням та			
8			зарядними станціями			
9			електромобілів у смарт-місті			
10			Діаграма прецедентів			
11	A4	ІК11.150БАК.006 Д2	Інтелектуальна	1		
12			робототехнічна система			
13			керування паркуванням та			
14			зарядними станціями			
15			електромобілів у смарт-місті			
16			Діаграма діяльності			
17	A3	ІК11.150БАК.006 Д3	Інтелектуальна	1		
18			робототехнічна система			
19			керування паркуванням та			
20			зарядними станціями			
21			електромобілів у смарт-місті			
22			Схема архітектури бази даних			
23	A3	ІК11.150БАК.006 Д4	Інтелектуальна	1		
24			робототехнічна система			
25			керування паркуванням та			
26			зарядними станціями			
27			електромобілів у смарт-місті			
28			Схема архітектури системи			
Зм.	Аркуш	№ докум.	Підпис	Дата	ІК11.150БАК.006 ТП	
Розроб.		Кудінова А.О.			Літ.	Аркуш
Керівн.		Крилов Є.В.				
Затв.						
					Інтелектуальна робототехнічна система керування паркуванням та зарядними станціями електромобілів у смарт-місті. Відомість проекту	
					Літ.	
					Аркуш	
					Аркушів	
					1	
					1	
					КПІ ім. Ігоря Сікорського Група ІК-11	

**Пояснювальна записка
до дипломного проєкту
на тему: «Інтелектуальна робототехнічна система
керування паркуванням та зарядними станціями
електромобілів у смарт-місті»**

Київ – 2025 року

ЗМІСТ

ВСТУП.....	4
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Загальна характеристика предметної області.....	7
1.2 Аналіз проблеми та їх актуальність	7
1.3 Постановка задачі.....	9
1.4 Огляд існуючих рішень	10
1.5 Послідовність вирішення задач та під задач.....	12
1.6 Структура та компоненти системи	12
1.7 Огляд та аналіз існуючих рішень.....	16
1.8 Типи зарядних станцій.....	16
Висновки до розділу 1	18
2 АНАЛІЗ ІСНЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ОБРАНОЇ АРХІТЕКТУРИ СИСТЕМ.....	19
2.1. Існуючі типи систем та принципи їх роботи.....	19
2.2 Вимоги до ефективної системи.....	22
2.3 Порівняння реалізацій та вибір архітектурного підходу	31
Висновок до розділу 2.....	34

					ІК11.150БАК.006 ПЗ			
		№ докум.	Підпис		Інтелектуальна робототехнічна система керування паркування та зарядними.станціями електричних автомобілів у смарт-сіті Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Кудінова А.О							
Перевірив	Крилов Є.В.						2	60
						КПІ ім. Ігоря Сікорського Група ІК-11		
Затв.								

3 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА СТВОРЕННЯ БАЗИ ДАНИ.....	36
3.1. Вибір мов програмування та фреймворків	36
3.2 Використання засобів для керування базою даних	38
3.3 Структура бази даних	41
3.4 Логіка роботи системи.....	43
Висновок до розділу 3.....	45
4 РОЗРОБЛЕННЯ СИСТЕМИ ТА ОПИС ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЯ ...	47
4.1 Особливості реалізації коду	47
4.2 Реалізація вебзастосунку	47
4.3 Особливості локальної програми для управління паркінгом	48
4.4 Опис отриманого функціоналу та тестування системи.....	51
Висновок до розділу 4.....	56
ВИСНОВКИ.....	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ А	62

ВСТУП

Майже кожен житель великих міст зустрічався із проблемою забрудненого повітря. Прокидалися і отримували повідомлення від застосунку «КиївЦифровий», що зараз не найкращий час для прогулянок на свіжому повітрі і закрийте вікна у ваших домівках. Вихлопні гази автомобілів спричиняють такі великі забруднення особливо вранці під час величезних заторів. Тож щоб допомогти екосистемі і дихати чистішим повітрям потрібно обирати електромобілі, щоб у вас в майбутньому не було страху через проблему зарядних станцій, буде створено інтелектуальну робототехнічну систему керування паркуванням та зарядними станціями електромобілів у смарт-місті.

Також болюча проблема у паркуванні авто, для цього відведено замало місць і утворюються стихійні паркувальні зони у не відведених для цього місцях. Тож проектування системи включає в себе поєднання двох напрямків, що зробить наше життя комфортніше, а головне з користю для нашого здоров'я та планети.

В сучасному світі все стрімко розвивається, всі хочуть багато всього, проте майже ніколи не задумуються до яких фатальних наслідків це може призвести. Тож головною ідеєю цього проекту йде думка про себе і оточуючих.

Створення системи «ParkingReservation» дає змогу виправити всі проблеми. Дана розробка користуватиметься великою кількістю охочих до використання. Основа концепції полягає у створенні Smart Sity, це буде її компонентою, відбувається осучаснення міст за допомогою роботизації виконання процесів, які можна замінити людську працю з метою покращення умов життя.

Переваги даної системи будуть у налагодженій реалізації в звичному для нас житті, яка може працювати цілодобово і всі процеси будуть автоматизовані . Виключається так би мовити «людський фактор» помилки, які виникають через фізичні для нас проблеми: втому, хворобу тощо.

Увага зосереджена перш за все для комфорту людей, великих територій, а не тільки мегаполісів, щоб кожен, хто бажає міг з легкістю користуватися цією розробкою. Щоб система користувалася попитом, потрібно її зробити доступною у

					ІК11.150БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

використання. Як можна більше спростити алгоритм використання , тобто щоб в користувача не було великого списку послідовності дій, щоб виконати паркування, адже це відштовхує і будуть надавати перевагу стихійним паркуванням в не призначеному для цього місці.

У підсумку всі зважених причин та факторів маємо чітко сформульовано мету для реалізації даного проекту – для створення інтелектуальної робототехнічної системи керування паркування та електричними зарядними станціями у смарт місті. Сучасний світ не стоїть на місці і стрімко розвивається, тому такі системи стануть необхідними у майбутньому.

Загалом ця розробка стане новою для України, адже аналогів такої гібридної системи не існує на рівні великих паркувальних зон, відповідно з оснащеними засобами. Тому якщо зважати на стрімкий розвиток електромобілів – розроблена система в майбутньому стане над необхідною.

					ІК11.150БАК.006 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Сучасний світ не стоїть на місці і стрімко йде до розвитку, з'являється багато виробництв і нових технологій. Всі прагнуть до комфорту і хочуть все краще і більше, але забувають про одну важливу річ – про оточуючих навколо себе людей і нашу планет.

Міста-мільйонники України переповненні вихлопними газами та авто, які розводять стихійні парковки в не відведених для цього місцях. Свідомі громадяни розуміють про проблему, але наданий момент відсутні для цього зручності і надають перевагу бензиновим авто та паркуванню поза правилами. Тож потрібно аналізувати проблему і знаходити відповідні рішення в час роботизованих технологій – це стає значно легше і можна поєднати комфорт із корисним.

Електрокари – це як новий подих для нашої екосистеми, екологічність, безшумність, дешевше обслуговування та зручна експлуатація. Існує один великий фактор чому водії в Україні відмовляються від таких авто - відсутність зарядних станцій, так у містах їх стає більше, а як щодо автомагістралей. Ви зустрінете сотні бензинових, дизельних та газових заправок і зможете, наприклад, поїхати в командировку або на відпочинок за 1000 кілометрів та не будете зустрічатись із проблемою пошуку пального, а ось на електромобілі так. Паркування в не відведених для цього місцях також шкодить суспільству, досить часто є випадки, коли рятувальна пожежна служба фізично не може допомогти через стихійне паркування і від цього залежать життя людей і кожна хвилини важлива для порятунку, проте авто може цьому зашкодити

Розвиток зарядних електростанцій та смарт-парковки – це вірний шлях до розвитку, який буде корисний не тільки для розвитку бізнесу, а й для покращення екології та зручностей як у великих містах так і на магістралях, це значно зменшить страх для надання переваги електромобілям. Це зможе покращити рівень комфортного життя та безпеки, як і для екологічної ситуації, життєдіяльності так і для безпечного паркування та заряджання авто.

1.1 Загальна характеристика предметної області

Ринок інтелектуальних систем стрімко розвивається, ключовий напрям це створення паркувальних майданчиків з необхідною інфраструктурою та зарядними станціями предметна область зосереджена на розвитку Smart parking із електростанціями на базі роботизованих рішень. За даними Опендатабот посилання <https://opendatabot.ua/>, від початку 2024 року в Україну було ввезено 20 785 електрокарів, що стало рекордною кількістю. Зазначається, що загалом попит на електромобілі за рік збільшився у 2,1 рази. Електроавтівки складають 12,9% від загальної кількості імпортованих в Україну машин. Так, у 2024 році було імпортовано 5,5 тисяч нових та 15,9 тисячі б/у електрокарів. Ця статистика свідчить про актуальність встановлення зарядних систем та про досить великий відсоток користувачів.

В ході розробки ми поєднаємо в одну інформаційну систему два напрями розвитку паркування та станції для заряджання автомобілів, що досить спрощує пошук вільних місць, яка має великий потенціал до розвитку із створення масштабності проекту.

1.2 Аналіз проблеми та їх актуальність

Проблема звичайних парковок полягає в їхній не організованості, загалом це досить простий алгоритм, отримуєш талончик або ж купуєш його в певної відповідальної особи на майданчику і потім шукаєш між рядами вільне місце. Така процедура має безліч недоліків та призначена більше для фінансового потоку, а ніж для зручності, бо навіть при отриманій пропуску ніхто не гарантує про те, що там все таки буде вільне місце. Тож перша проблема полягає в моніторингу вільних місць, яка в подальшому буде вирішена. Досить актуально є попереднє бронювання паркомісця. Це корисно як і місцевим жителям так і туристам. Інтернет речей (IoT) може стати найкращим підходом для вирішення цієї проблеми. Кількість

					ІК11.150БАК.006 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

транспорту зростає створюючи проблеми. Відсутність автоматизованої системи призводить до постійних заторів. Тому така смарт система буде користуватися попитом і вирішить багато питань.



Рисунок 1.1 – Карта парковок міста Києва від компанії «Київтранспарксервіс» [13]

Згідно Рисунку 1.1, дані взято з компанії «КиївЦифровий.» можна отримати інформацію про паркувальні зони, вони виділенні синіми зонами, ця система передбачає в собі 12 000 місць, що замало для такого міста і вона не оснащена відповідними системи для зручності у використанні. [13].

В Україні остання статистика зарядних станцій з 2021 року становить 3,244, що досить мало, адже на одну таку претендують 7 авто і зазвичай вони розміщені у містах. Вирішення цієї проблеми комплексно поєднується із смарт паркування. Система, яка проводить моніторинг і бронювання вільних станцій, коли можна швидко переглянути за допомогою застосунку і провести оплату[15].

Таким чином в людей не буде виникати страху пов'язаного із нестачею заряду батареї, що надає перевагу купувати електромобілі. На рисунку 1.2 представлено кількість станції від провідної компанії «Автоентерпрайз». Наочно видна мізерна кількість особливо на магістрях [14].

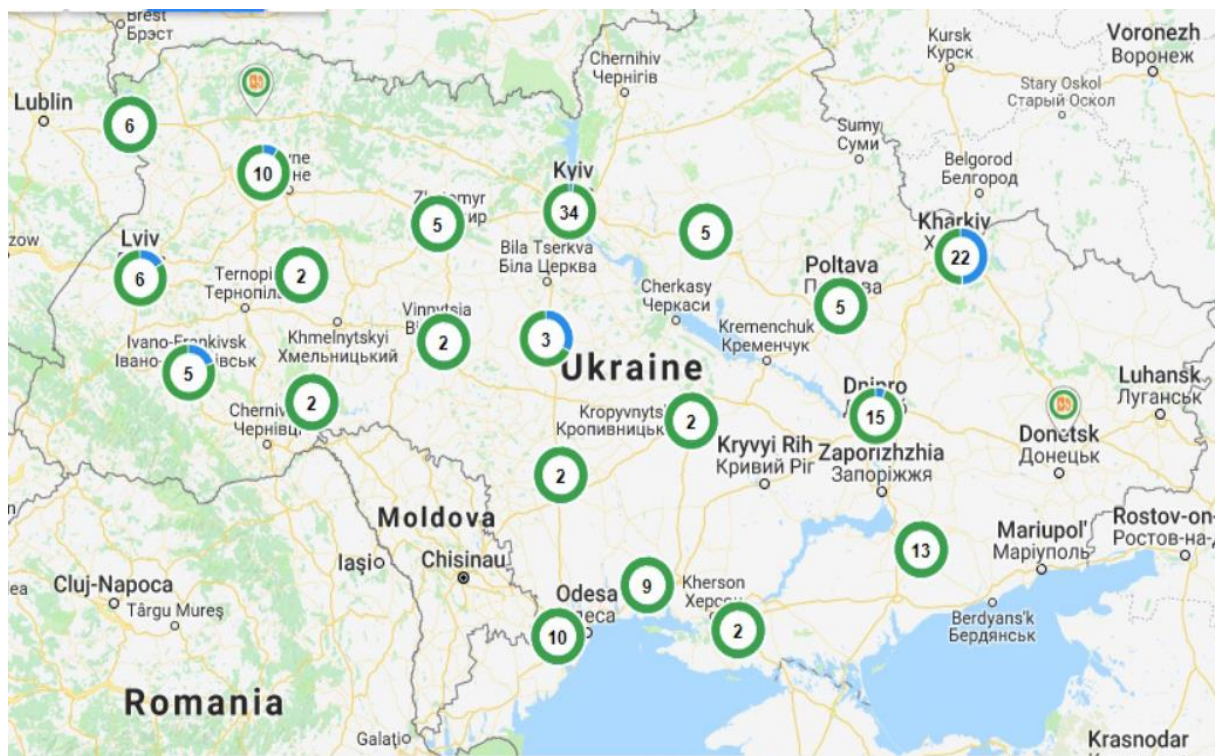


Рисунок 1.2 – Мапа зарядних станції компанії «Автоентерпрайз» [14]

В ході проекту полягають такі проблеми як регулювання, безпека, контроль, екологія. Спосіб вирішення – це створення інтелектуальної системи, яка поєднує в собі Smart Parking та зарядні станції, яка відповідає потребам сучасного суспільства, збільшує осучаснення трансформацій міст та покращує якість життя суспільства.

1.3 Постановка задачі

Основною задачею проекту є створення розумної системи на базі роботизованих рішень, яка включає в себе паркування та зарядні станції.

Користувач в системі може проводити моніторинг вільних місць, пошук найближчих парковок та попереднє бронювання. Система включатиме в себе моментальну оплату в результаті підтвердження отримується згенерований QR-код, який сканується за допомогою сканерів з налаштованою програмою, яка також входить в розробку систему паркування. Аналогічний функціонал поширюється на бронь зарядних електростанцій. До головної задачі проекту входить поширення популярності такої системи, включаючи великі дистанції. Система є повністю автоматизованою і не потребує людського контролю на майданчиках та працюватиме цілодобово.

Також розробляючи систему потрібно звернути увагу на легкість користування відповідним додатком. Задача перш за все полягає в покращених умовах для життя з доступним для цього навичками. Складність проекту полягає в тому, щоб розробка не відштовхувала від себе людей, щоб таку систему могли використовувати різні вікові категорії.

1.4 Огляд існуючих рішень

Потрібно вміти поєднувати приємне з корисним – це як червона нитка для всіх проєктів. Перш за все потрібно розглянути ринок в Україні, які компанії мають провідні рейтинг та віднайти їх недоліки (Таблиця 1.1).

З наведеної інформації можна прийти до висновку, що ринок починає тільки формуватися, наразі переважають більше виробник зарядних станцій, а ніж їх мережі. Лідирує в наведених даних AutoEnterprise (AE Charge), яка має більше тисячі зарядних станцій, проте використання поширена тільки в певних містах або ж в окремих областях і не мають зарядних станцій на певних довгих ділянках і це одразу влаштує великі незручності у використанні.

Навіть поодинокі зарядні станції вже матимуть великий попит у використанні. На даний момент ці станції працюють тільки у двох регіонах нашої країни.

					ІК11.150БАК.006 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

Компанія YASNO має компактні зарядні станції, які не потребують великою площі, одже будуть дуже зручні у налаштування на певних ділянках. Навіть поодинокі зарядні станції вже матимуть великий попит у використанні. На даний момент ці станції працюють тільки у двох регіонах нашої країни.

Таблиця 1.1 – Рейтинг електрозаправок в Україні

Назва	Мережа	Деталі
AutoEnterprise Charge) (AE	1000 зарядних станцій	Власне виробництво зарядних пристроїв, мобільний застосунок AE Charge Point.
ТОКА	Сотні точок у великих містах	Можливість віддаленого керування зарядкою через застосунок,
YASNO E-mobility	Зарядні станції у Києві та Дніпрі.	Можна оплачувати через мобільний застосунок YASNO.
Volt Charger	Київ, Львів, Одеса, Харків	Зручний застосунок для управління і розташовано поруч торговельних центрів

З наведеної інформації можна прийти до висновку, що ринок починає тільки формуватися, наразі переважають більше виробник зарядних станцій, а ніж їх мережі.

Основний процес нашої системи – це гібридність та унікальність. В Україні немає жодного аналогу, який би поєднував в собі два напрями розумне паркування з інфраструктурою для зарядних станцій. Популярність електрокарів та їх потреба зростає, то і збільшується попит на заправки. Автоматизація системи буде відбуватися для користувача за рахунок вебсайту, який виконує алгоритм дій для

бронювання певних послух, після чого проводить оплату. Дану систему потрібно оптимізовувати для подальшого розвитку в світі сучасних технологій. В результаті ця комбінація принесе великий прибуток і виконає основний бізнес- процес.

1.5 Послідовність вирішення задач та під задач

Спочатку відбувається уточнення нашої задачі. Ми чітко формує алгоритм дій основних задач:

- користувач бронює парко місце або зарядну станцію;
- обравши відповідні категорії підтверджує бронь;
- отримує згенерований QR – код, де вказано місце та час бронювання;
- прибувши на місце, сканує код, та отримує доступ до відповідної зони;
- на виїзді повторне сканування, відкриття доступу до виїзду;
- оплата послуг.

Для цього потрібно розробити вебзастосунок для користувача і систему для зчитування QR- коду за допомогою комп'ютерного зору. При розробці програми під задачі – це доповнення моніторингу вільних місць для всіх доступних послуг, віддалена керування електростанціями.

1.6 Структура та компоненти системи

Можливості нашої системи глобальні за допомогою Інтернету речей та робототехнічних систем. Smart Sity в ідеальному її представленні використовує сенсори та шлюзи для цілодобової роботи системи. Наша система поділяється на 4 групи: сенсорні вузли, клієнтські вузли, шлюзи та сервери так, як це зображено на рисунку 1.3. Пристрої з'єднані однією мережею. Сенсорний вузол збирає інформацію та будь-які інші деталі та відображає результати, які надійшли із сервера. Зв'язок між цими двома пристроями здійснюється через шлюз, який обмінює та обробляє повідомлення за участі декількох протоколів.

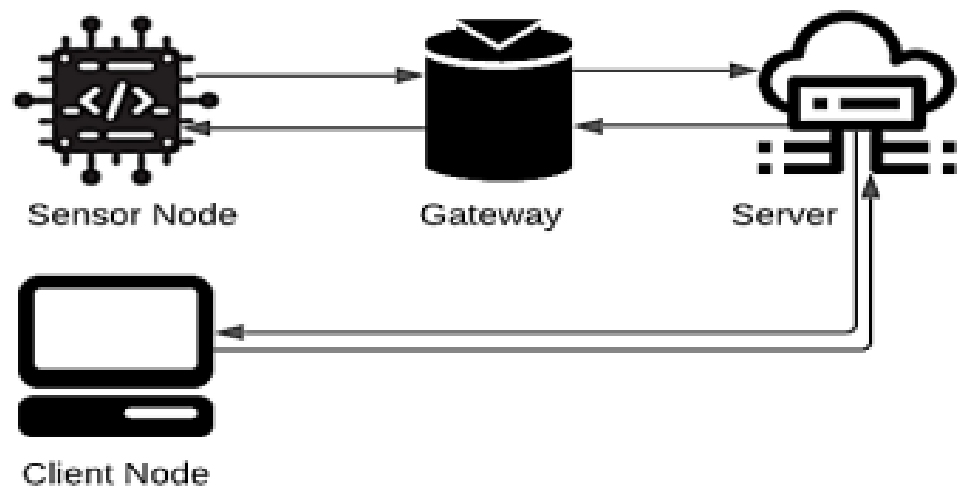


Рисунок 1.3 – Компоненти IoT-системи [18]

Клієнтський вузол – напряму зв’язується із сервером. Сервер збирає інформацію із сенсорних клієнтських вузлів, а також може надсилати запитувану інформацію назад до них [16]. Загалом система зображена на рисунок 1.4

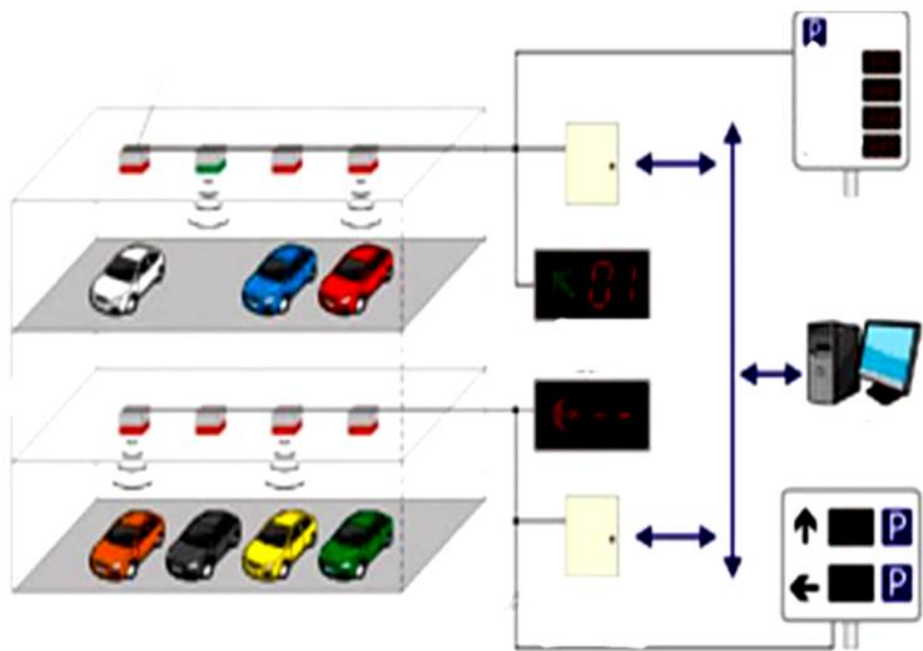


Рисунок 1.4 – Схематичний функціонал [17]

IoT — це парадигма, за допомогою якої машина може спілкуватися з іншою машиною. Він відомий як зв'язок між машинами (M2M). Він широко застосовується в багатьох сферах, наприклад, у домашній автоматизації, промисловій автоматизації, охорона здоров'я, навіть система паркування [17]. IoT також може дозволити звичайний електронний пристрій для дистанційного керування. Використовуючи датчики та шлюзи, він гарантує, що користувачі можуть керувати пристроями, обмінюючись інформацією між пристроями. Ця парадигма також згадується як розумні пристрої. Об'єднання інтелектуальних пристроїв в інтелектуальне середовище не тільки підвищує ефективність використання пристроїв, але також покращує взаємодію з користувачем. Фізична парковка оснащена відповідними датчиками та камерами рисунок 1.5



Риснок 1.5 – Альтернативний вигляд парковки з відповідними датчиками руху [17]

Для зручності користування та комфорту буде проведено маркування та поділ. Інформаційні знаки, які вказуватимуть напрям руху до конкретного паркомісця рисунок 1.6 та рисунок 1.7



Рисунок 1.6 - Зразковий вигляд парковки в середині [15]

Така паркувальна зона буде оснащена зарядними станціями для зарядження авто і матиме світлове маркування для показу напрямку руху, користувачу при заїзді будуть показуватися стрілки для напрямку для його персонального місця



Рисунок 1.7 – Зразковий вигляд парковки зовні [15]

Зм.	Лист	№ докум.	Підпис	Дата

1.7 Огляд та аналіз існуючих рішень

Існуючого рішення нашої гібридної системи не існує, тож розглянемо все поступово. Для прикладу візьмемо YASNO E-mobility від ДТЕК, яка має свою мережу в Києві та Дніпрі, з недоліків, вона відсутня на автомагістралях. Користування системою відбувається через додаток, проте без попереднього бронювання і невідомо стан зарядної станції, вони розташовані на території паркінгів торгових центрів, проте функціонально з майданчиком вони не пов'язані, тож як наслідок ви не будете мати гарантію чи поруч є вільне місце для підзарядки все досить стихійно на рисунку 1.8 зображено інтерфейс застосунку[12].

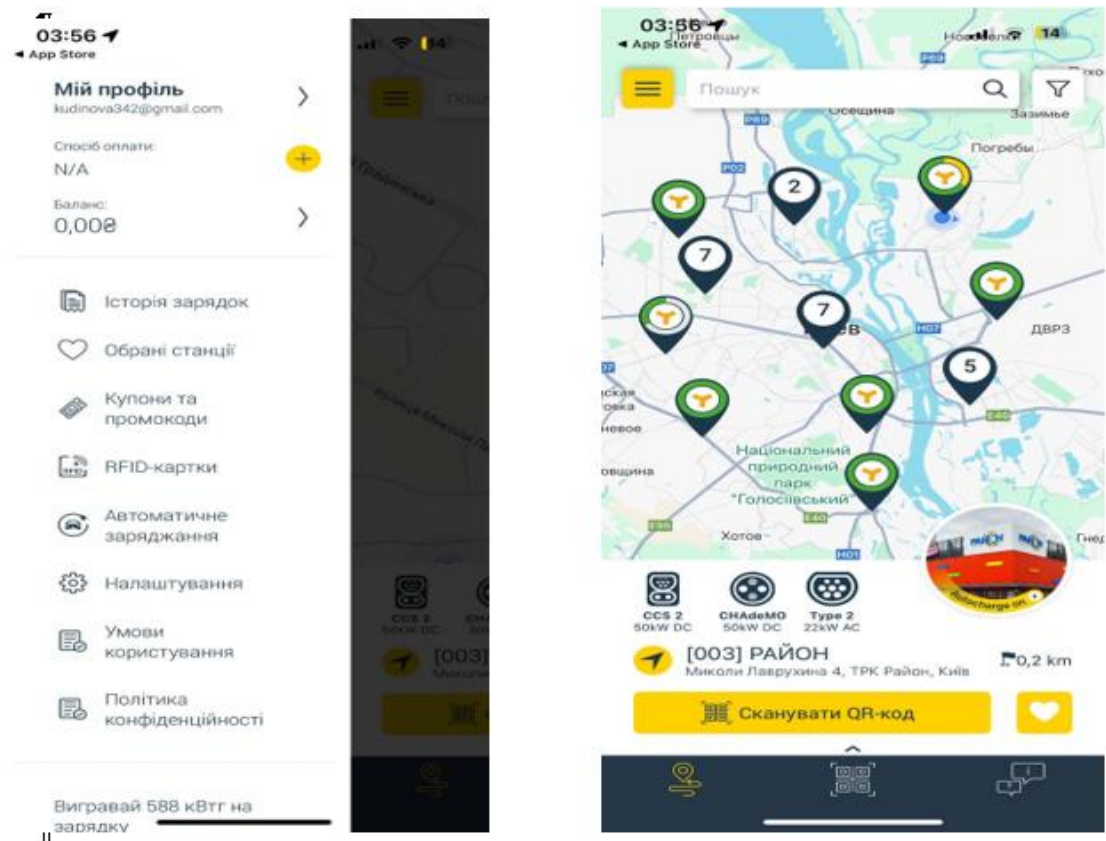


Рисунок 1.8 – Інтерфейс застосунку YASNO E-mobility від ДТЕК [12]

Аналог смарт-парковки є в додаток «КиївЦифровий» проте, там лише представлена комерційна частина без зв'язку з SmartParking, але якщо б у майбутньому була інтеграція із цим додатком, то це зменшило проблему стихійних

паркувань в непризначених місцях з уникненням заторів та економією часу в пошук вільного місця

Тож в результаті проведеного аналізу існуючих рішень, наша система буде конкурентоспроможною, адже поєднує в собі широкий функціонал із точністю.

1.8 Типи зарядних станцій

Для проектування даної системи розглянуто різні типи зарядних станцій. Найпоширенішою, проте досить слабкою є АС (повільна зарядна станція). Такий тип включає в себе два рівня зарядки. Перший рівень до 2,4 кВт та має змогу підключення до звичайної розетки, тому такий варіант не підходить для мережі зарядок, адже розрахований для домашнього використання. Другий рівень – це 3,7 – 22 кВт досить швидкий змінний струм і такий варіант вже починати використовувати для паркувальних майданчиків.

Розроблена паркувальна мережа буде включати більшість DC (швидка зарядка) рисунок 1.9, вона вважається третім рівнем і має потужність понад 50 - 350 кВт і може заряджати автомобіль дуже швидко приблизно за 40-60 хвилин, але через високу потужність системи зазвичай їх використовують на автомагістралях.



Рисунок 1.9 – Зарядна станція типу DC [20]

					ІК11.150БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		17

Висновки до розділу 1

При початковому аналізі майбутньої інтелектуальної системи керування паркуванням та зарядними станціями електромобілів у смарт-місті. Визначили основний бізнес-потік та користь для суспільства. Виокремили проблему, те що відсутність урегульованих та організованих парковок спричиняє проблеми в заторах та роботу невідкладних служб. Головним недоліком надання переваги електромашинам стає мізерна кількість зарядних станцій, що і стає головним фактором чому люди відмовляються від таких авто. Дослідили функціональні можливості вже масштабованої системи. Виокремили задачу та її під задачі для правильного проектування системи. Поставлена мета проекту полягає у розробці для користувача системи керування та управління паркувальними місцями та зарядним станціями віддалено. Виконавши всі умови проекту, потрібно буде додати спеціальні додаткові технічні засоби на паркувальні зони: такі як шлагбаум зі сканером, датчики руху, сенсори та камери з налаштованим штучним інтелектом для відстаження дій в паркувальній зоні.

Система буде включати цілу мережу «ParkingReservation», яка розташовуватиметься по автомагістралях та містах України, що значно покращить рівень продажів електромобілів і надасть перевагу до покращення екологічних умов

2 АНАЛІЗ ІСНЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ОБРАНОЇ АРХІТЕКТУРИ СИСТЕМИ

2.1 Типи систем та принципи їх роботи

У другій частині роботи буде проаналізовано існуючі типи систем керування паркуванням, розглянуто принципи їхньої роботи, технічні особливості та способи взаємодії з користувачем і обладнанням. За цим аналізу буде обґрунтовано вибір архітектурного підходу, який найбільше відповідає умовам реалізації проєкту. Приділено увагу питанням обліку вільних місць, контролю доступу, ідентифікації користувачів, а також безпеці обробки персональних даних і керування зарядними станцій та аналіз їх типів.

Інтелектуальні системи керування паркуванням і зарядними станціями є невід'ємною частиною концепції смарт-міста. На сьогоднішній день існує декілька основних підходів до реалізації таких систем, кожен з яких має власні технічні та організаційні особливості. Такі як централізовані API-системи, локальні автоматизовані рішення, а також комплексні системи, що поєднують паркування з електрозарядною інфраструктурою.

Керування паркомісця та зарядними станціями буде виконуватися за одним алгоритмом дій та не потребуватиме іншої системи керування, адже вона буде гібридною для двох систем одночасно. Управління системи буде здійснюватися віддалено і реалізується робототехнічна система, завдяки зв'язку відповідних датчиків руху, які замінюють людську працю і цей процес є автоматизацією.

Централізовані API-системи функціонують на основі хмарних або серверних платформ, які акумулюють дані від численних парковок. Кожна з таких парковок надає відкритий доступ до інформації про кількість вільних місць, тарифи та інші параметри через публічне API. Дані з цих API обробляються агрегатором, який формує загальний перелік доступних місць для користувачів на своєму сайті або в мобільному застосунку. Під час бронювання здійснюється зворотний запит на API з метою створення резервації. [11-14].

Структура роботи таких систем може бути описана у вигляді спрощеної діаграми потоку даних на рисунку 2.1:

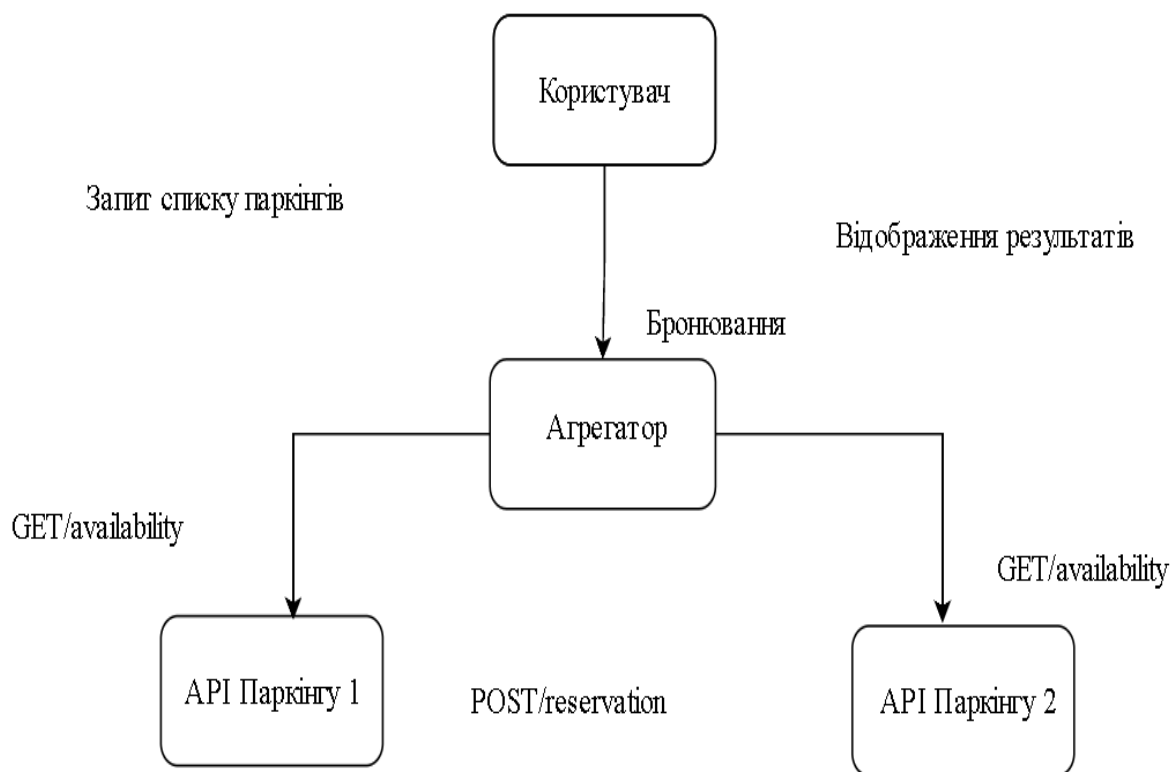


Рисунок 2.1 – Структура роботи систем агрегації для вебсистеми

Іншим поширеним типом є локальні системи, які функціонують автономно в межах одного паркінгу. Вони зазвичай не потребують зовнішнього підключення до інтернету та зосереджені на управлінні доступом і точному контролі кількості вільних місць. У таких системах використовуються технології ідентифікації на основі QR-кодів, RFID-карток, а також датчики, встановлені безпосередньо на паркомісцях, які визначають наявність автомобіля. В'їзд здійснюється через автоматизовані шлагбауми, які підключені до локального сервера. Сервер виконує перевірку права доступу та приймає рішення про відкриття або блокування в'їзду.[1- 4]

Схема взаємодії елементів локальної системи виглядає наступним чином рисунок 2.3:

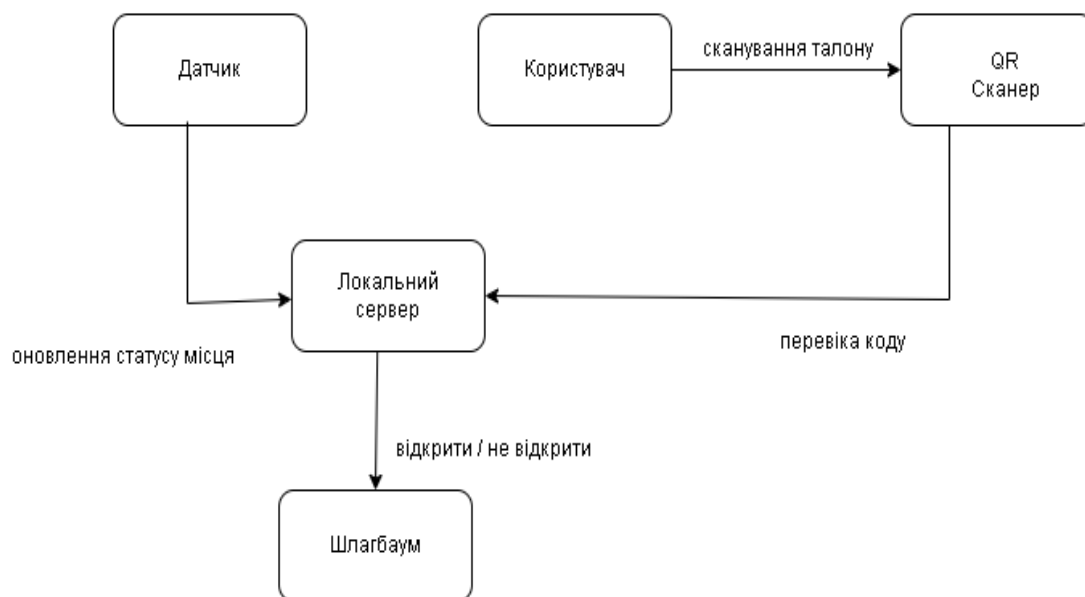


Рисунок 2.2 – Схема взаємодії елементів локальної системи

Третім важливим напрямком розвитку є інтеграція з електрозарядними станціями. Користувач має змогу не лише забронювати місце для паркування, але й отримати доступ до зарядного пристрою. Це особливо актуально у містах з високою концентрацією електромобілів. Резервація місця включає резерв зарядки на певний проміжок часу, що дозволяє оптимізувати навантаження на інфраструктуру.

У деяких системах зарядка активується автоматично після в'їзду або через авторизацію по QR-коду.[18]

У даного проєкті передбачається умовне включення зарядної функції до кожного паркомісця, що дає змогу в майбутньому реалізувати повноцінну інтеграцію із зарядними станціями без необхідності значного перепроєктування системи. Різноманіття можливих підходів до організації сучасної системи паркування, їх аналіз є необхідним кроком для обґрунтування власної архітектури, яка має поєднувати простоту впровадження, надійність та можливість масштабування.

2.2 Вимоги до ефективної системи

Проектування інтелектуальної системи паркування передбачає дотримання низки критичних вимог, без яких неможливо досягти її стабільної, масштабованої та безпечної роботи в умовах смарт-міста.

Ефективна система має не лише забезпечувати базову функцію бронювання місця, але й інтегруватися з фізичною інфраструктурою паркінгу, гарантувати точний облік місць, захист особистих даних користувачів та стабільну роботу в режимі реального часу.

Однією з умов ефективності є реальний облік вільних місць. У більшості сучасних систем паркування існує поділ між заброньованими місцями та фактично зайнятими. Це означає, що система має враховувати не лише ті слоти, які зарезервовані наперед, але й ті, що вже зайняті транспортними засобами без бронювання.

Для цього необхідно використовувати датчиків присутності, що вбудовуються у покриття кожного паркомісця або монтуються у вигляді камер, ультразвукових сенсорів, інфрачервоних детекторів тощо, як зображено на рисунку 2.3. Тож тепер така система буде відповідно навчаною і може фільтрувати броньовані місця та зайняті паркомісця і зарядні станції за іншим алгоритмом полегшеним.

Також система може бути покращена відео наглядом за допомогою влаштованого штучного інтелекту. Вони будуть аналізувати надане зображення на даний момент і за допомогою певних алгоритмів зможуть аналізувати чи паркувальна зона зайнята чи вона вільна.

Для зарядних станцій система працює так само, адже це клієнта система запитує чи потрібно вам зарядна станція і він відповідно може керувати нею, погодитись чи відмінити цю функцію. Також така камера покращує безпеку і дозволяє дистанційно нею керувати. За допомогою застосунку користувач може спостерігати за авто

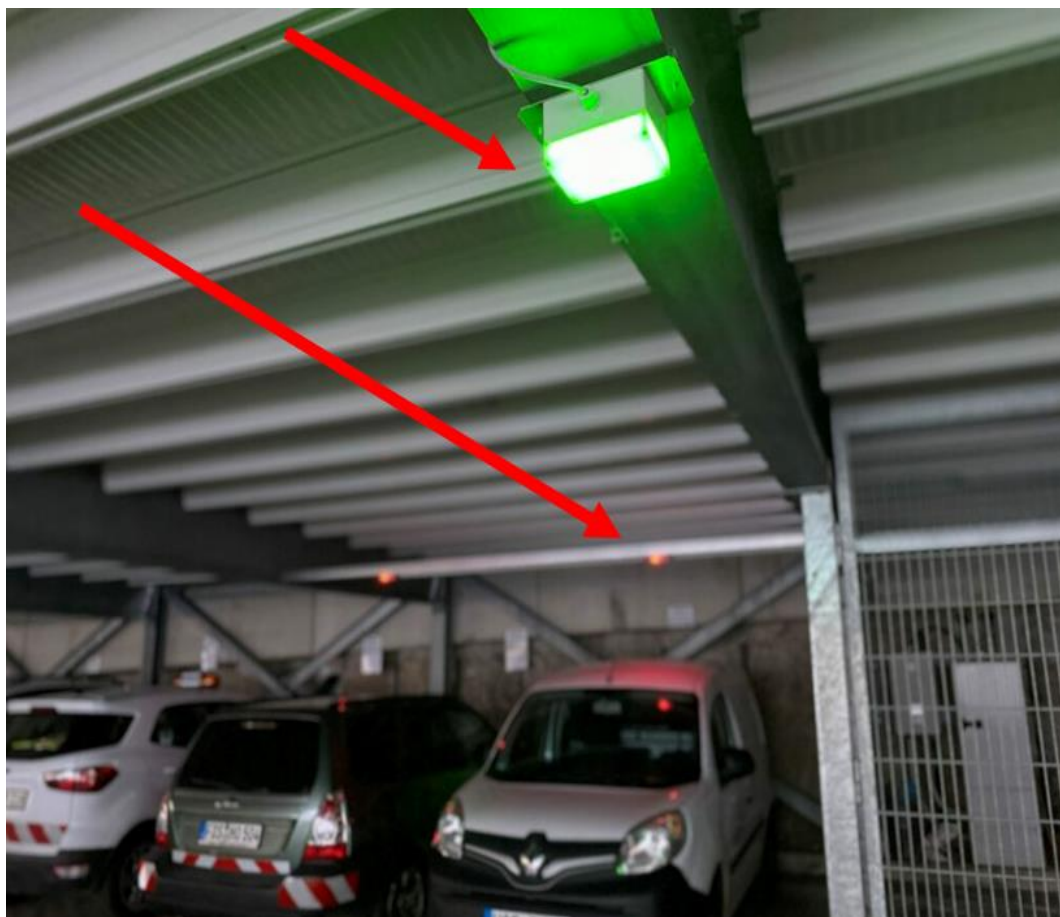


Рисунок 2.3 – Датчик розміщений на паркувальній зоні [17]

Паркувальна зона буде оснащена ультразвуковими датчиками з вбудованим індикатором LED-світла типу UDS.30 зображено на рисунку 2.4. Цей пристрій працює так: датчик випромінює ультразвукову хвилю вниз і якщо на шляху виникає перешкода, тобто автомобіль, хвиля відповідно починає відбиватися і датчик буде спрацьовувати на неї, позуючи що це місце зайняте приклад роботи показано на рисунку 2.5.

І так само цей алгоритм працює навпаки, якщо хвиля доходить до землі без перешкод і не відбивається, означає, що це місце вільне і може бути використаними для заряджання або ж для паркомісця.

Дані передаються через дротовий Ethernet або також через бездротовий зв'язок - Wi-Fi на сервер, який робить запит до бази даних і відповідно записується запит про те, що це місце зайнято без попереднього бронювання, за допомогою

Зм.	Лист	№ докум.	Підпис	Дата

такої система зникне плутанина і користувач може мати гарантій, що місце буде передбачено для нього.



Рисунок 2.4 – Датчик UDS.30 [19]

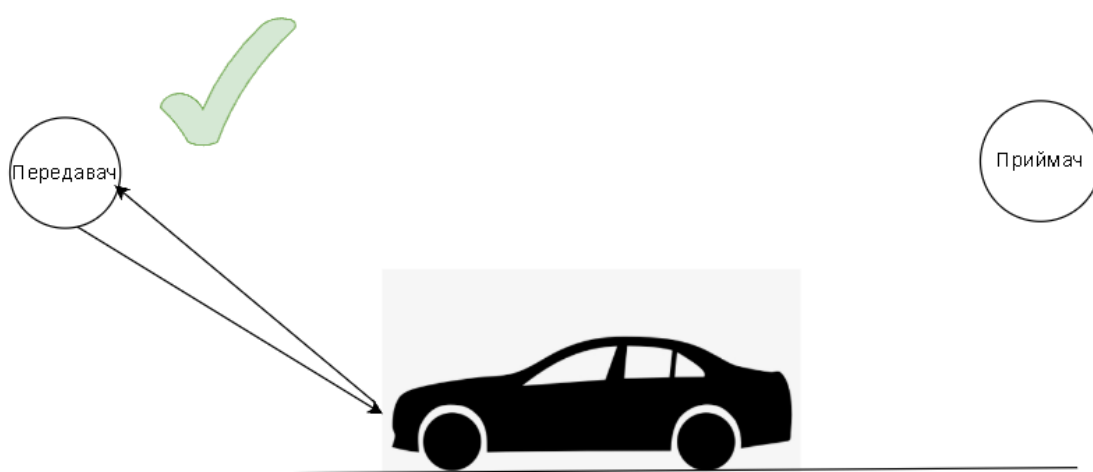


Рисунок 2.5 – Принцип роботи датчика на виявлення авто

Для керування зарядними станціями і моніторингу за допомогою системи використовують датчик зайнятості (Occupancy Sensors). Завдяки ним ми поєднаємо апаратне забезпечення, яка включає в себе станцію та програмне забезпечення, розроблену вебсистему. Датчики струму визначають чи споживається на даний момент струм і в цей момент вбудований в кабель сенсор розпізнає чи він

підключений до авто. Датчики оброблюють інформацію і передають через безпроводну мережу до серверів, які в свою чергу взаємодіють з системою і подають відповідний сигнал.

Приймач не отримав сигнал (рисунок 2.6) - ультразвукову хвилю і зафіксував наявність об'єкта, що означає – місце недоступне для паркування та заряджання. Тому система при бронюванні доступними зарядними станціями – видасть повідомлення, про те що місце на даний момент зайняте.

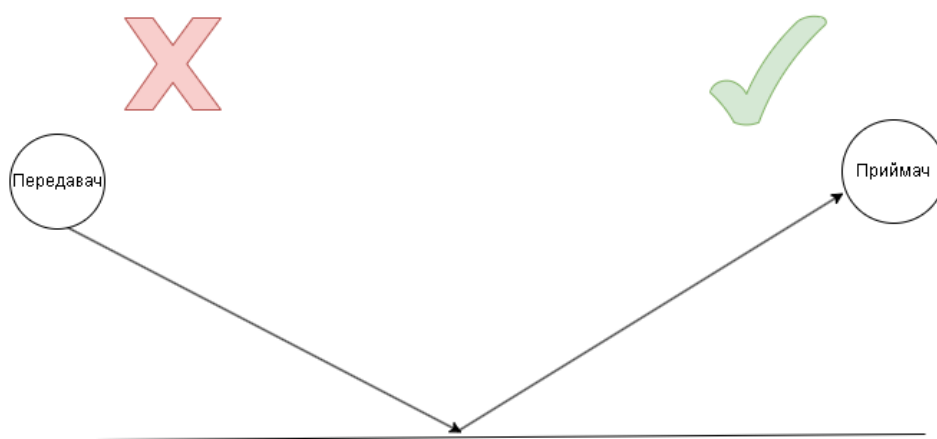


Рисунок 2.6 – Ультразвукова хвиля у разі відсутності авто

Завдяки таким пристроям система в реальному часі отримує інформацію про наявність автомобіля на конкретному місці, також можливе створення комбінованої логіки: при перевищенні порогу заповненості, що розраховується як сума активних бронювань та кількості реально зайнятих місць, в'їзд нових авто без броні блокується автоматично. Це запобігає перевантаженню та хаосу на території парковки. [13-15]

Модель взаємодії датчиків, системи бронювання та пристрою контролю в'їзду можна представити так рисунок 2.7 :

Ще одним достатньо необхідний компонентом є наявність можливості попереднього бронювання паркомісця. Система повинна забезпечувати зручний інтерфейс для користувачів, який дозволяє заздалегідь обрати дату, місце, час

перебування, а також отримати підтвердження про успішну резервацію. Після цього користувач має можливість пред'явити підтвердження у вигляді QR-коду або іншого ідентифікатора при в'їзді. Саме через можливість попереднього бронювання знижується рівень хаотичного заповнення паркінгу, особливо у години пік.

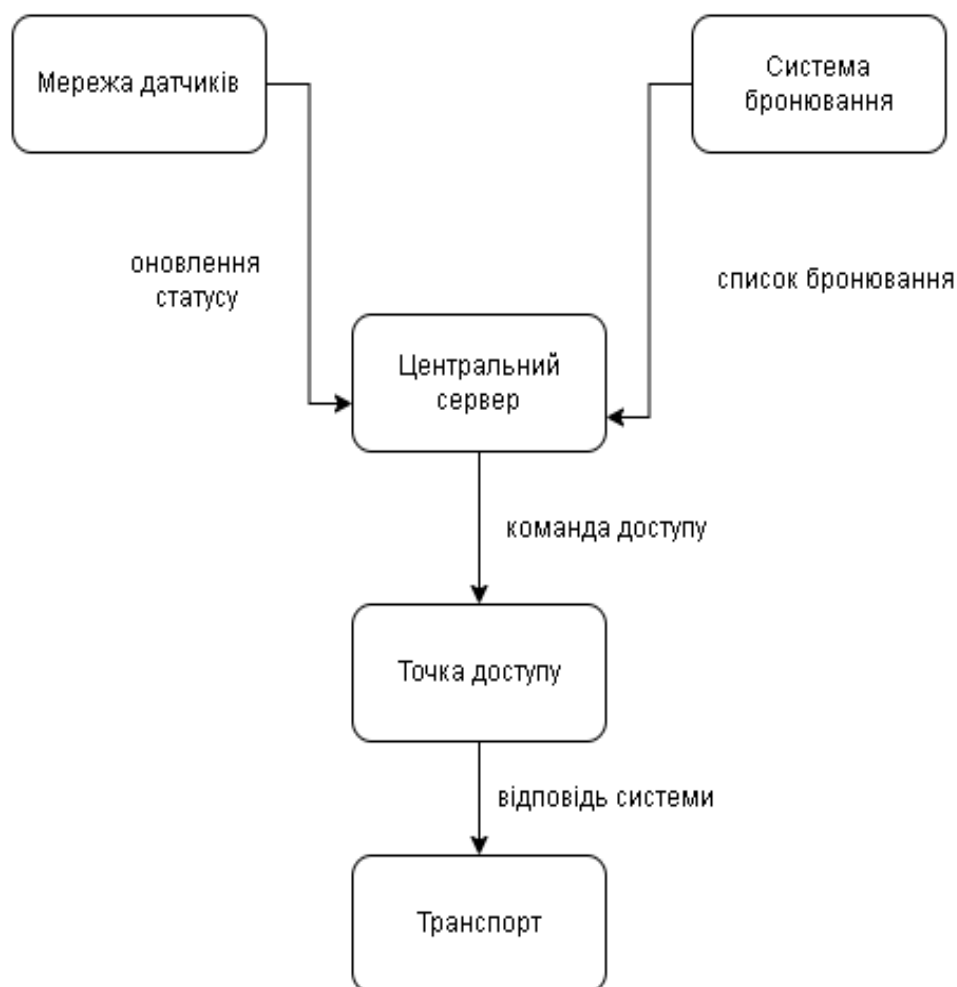


Рисунок 2.7 – Модель взаємодії датчиків

Важливою функціональною вимогою є аутентифікація та ідентифікація користувача. Якщо система дозволяє створення бронювання, зберігання історії, доступ до персональних кабінетів або здійснення оплат, вона має забезпечити надійні механізми авторизації. Це може бути реалізовано через стандартні методи:

логін і пароль, соціальні ідентифікатори (OAuth), або ж через двофакторну аутентифікацію. Усі дії користувача, пов'язані із створенням бронювання, повинні бути прив'язані до конкретного облікового запису. [5-7]

Існує інший практичний аспект, який стосується зручності та швидкості взаємодії з системою. У деяких випадках надмірні вимоги до ідентифікації можуть стати бар'єром для користувача. Наприклад, якщо бронювання є безкоштовним і короткостроковим, немає сенсу примушувати людину проходити повну реєстрацію. Це актуально у сценаріях, коли рішення про бронювання приймається спонтанно або в дорозі. У таких моментах можна використовувати спрощений механізм - розміщення дорожніх знаків із QR-кодом паркування на рисунку 2.8 заздалегідь на маршруті, за кілька хвилин до досягнення місця.



Рисунок 2.8 – Розташування на посилання до сайтів -агрегаторів на дорозі [16]

Після сканування QR-коду пасажир може здійснити бронювання буквально в кілька кліків без реєстрації чи авторизації. Таке бронювання буде обмеженим у часі

активним протягом однієї години, якщо користувач не приїде в зазначений проміжок, резерв автоматично скасовується.

Досягається баланс між безпекою і зручністю, де не задіяні чутливі дані чи транзакції, система може працювати без повноцінної авторизації, що значно покращує користувацький досвід.

Інтеграція з системами контролю доступу є ще одним технічним викликом. Йдеться про синхронізацію програмної частини із фізичними пристроями - автоматичними шлагбаумами, терміналами, камерами розпізнавання номерних знаків. Залежно від конфігурації, ці пристрої можуть отримувати команду на відкриття в'їзду лише після позитивної верифікації коду, QR-талону або номерного знаку. Включає в себе наявність серверної логіки, яка здатна обробляти запити з мінімальною затримкою та надавати однозначну відповідь щодо прав користувача на в'їзд у певний момент часу. У даній роботі буде реалізовано варіант швидкого бронювання на рисунку 2.9 без оплати, який не вимагає обов'язкової авторизації користувача. Такий підхід дозволяє забезпечити зручність і швидкість взаємодії з системою, особливо у випадках, коли користувач здійснює бронювання під час поїздки або безпосередньо перед прибуттям на паркувальний майданчик. Алгоритм цього процесу наведено нижче. Інтеграція із системами контролю доступу передбачає не лише перевірку QR-кодів або ідентифікаторів, а й контроль за фактичним наповненням паркінгу. Для цього застосовується комбінована логіка: враховуються як резервації (бронювання), так і вже заїхавші автомобілі.

На в'їзді система приймає рішення про відкриття шлагбаума лише у випадку, якщо загальна кількість автомобілів (броньованих і фізично присутніх) не перевищує загальної кількості місць на парковці. У роботі використовуються такі компоненти:

- датчики присутності автомобілів на кожному місці (або на в'їзді/виїзді)
- датчики зайнятості;
- контактні сенсори в роз'ємі;
- датчики струму

- система обліку бронювання;
- сервер контролю доступу;
- фізичний пристрій - шлагбаум;
- ідентифікатор користувача: QR-код або номер авто.

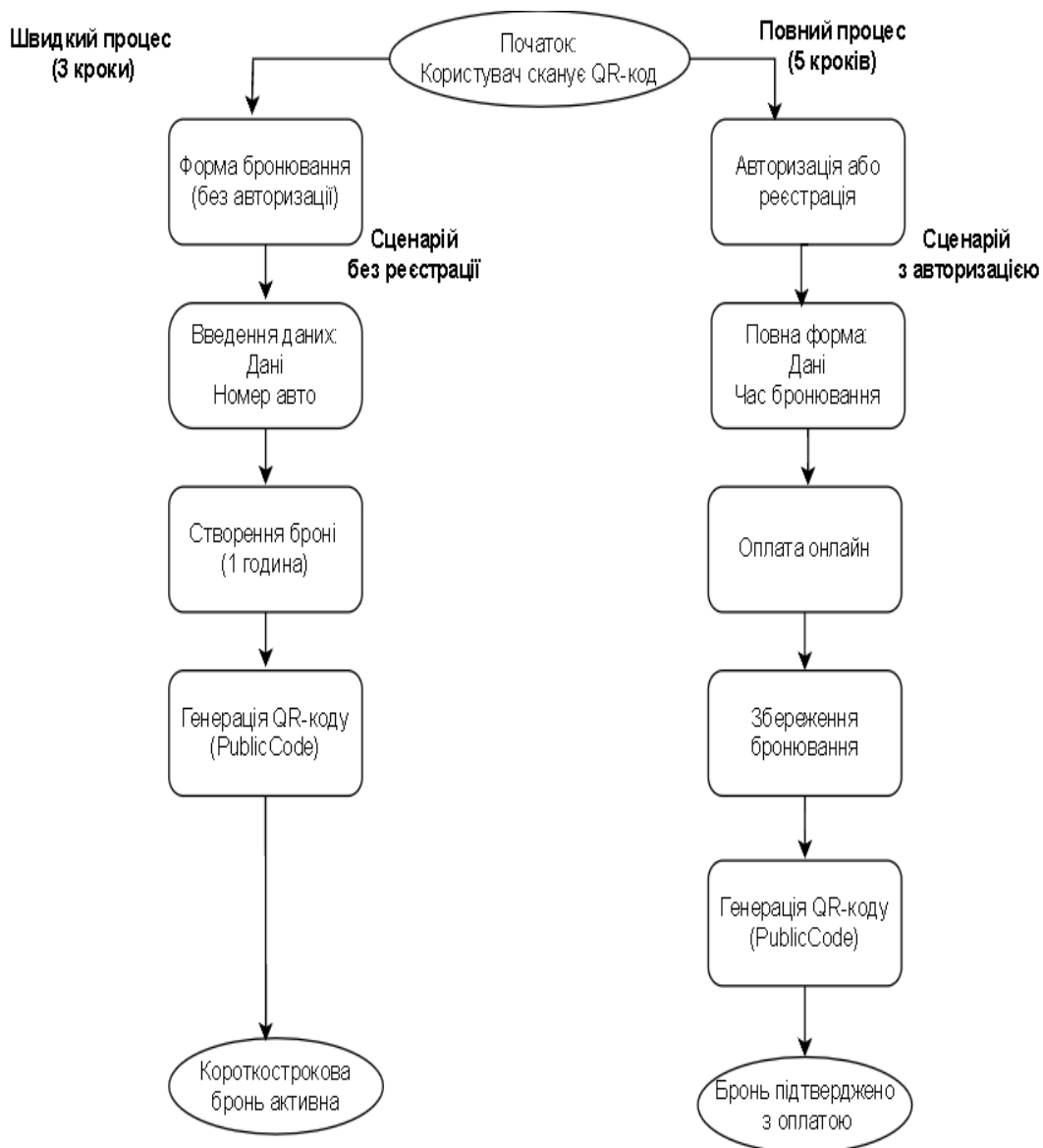


Рисунок 2.9 – Алгоритму швидкого бронювання без оплати

Логіка прийняття рішення:

- якщо користувач має дійсне бронювання — дозвіл на в'їзд.
- якщо користувач не має бронювання — перевірити, чи кількість заїхавших + кількість активних бронювань менше загальної місткості парковки.

- якщо так - дозволити в'їзд.
- якщо ні - заборонити.

Якщо на парковці 20 місць, а вже заїхало 15 машин і є ще 5 активних бронювань - доступ іншим автомобілям без резервації буде заборонено. Це виключає ситуації перенавантаження та сприяє точному обліку.

Цей метод є особливо ефективним у багатоповерхових або автоматизованих паркінгах, де вхід не контролюється оператором, а виконується через електронну систему у режимі реального часу.

Також дуже важливою є безпека обробки персональних даних. У системі зберігаються відомості про ПІБ користувача, державний номер автомобіля, історію бронювань, а також, можливо, платіжні дані. Алгоритму логіки контролю доступу проілюстровано на рисунку 2.10.



Рисунок 2.10 – Алгоритму логіки контролю

Усі ці елементи підпадають під дію законодавства про захист персональних даних, включаючи міжнародні стандарти (як-от GDPR, якщо система передбачає вихід на зовнішні ринки). Обов'язковою є реалізація таких елементів, як шифрування конфіденційної інформації, захист від несанкціонованого доступу, аудит дій у системі, регулярне резервне копіювання та контроль цілісності бази даних.

Проблематику організації безпечної, функціональної та масштабованої системи можна зобразити у вигляді дерева проблем рисунок 2.11: Для створення ефективної системи управління паркуванням необхідно передбачити цілу низку технічних, організаційних та правових елементів. Їх врахування на етапі архітектурного проектування дозволяє сформувати надійну, безпечну та зручну для користувача платформу, яка зможе адаптуватися до умов реального міського середовища

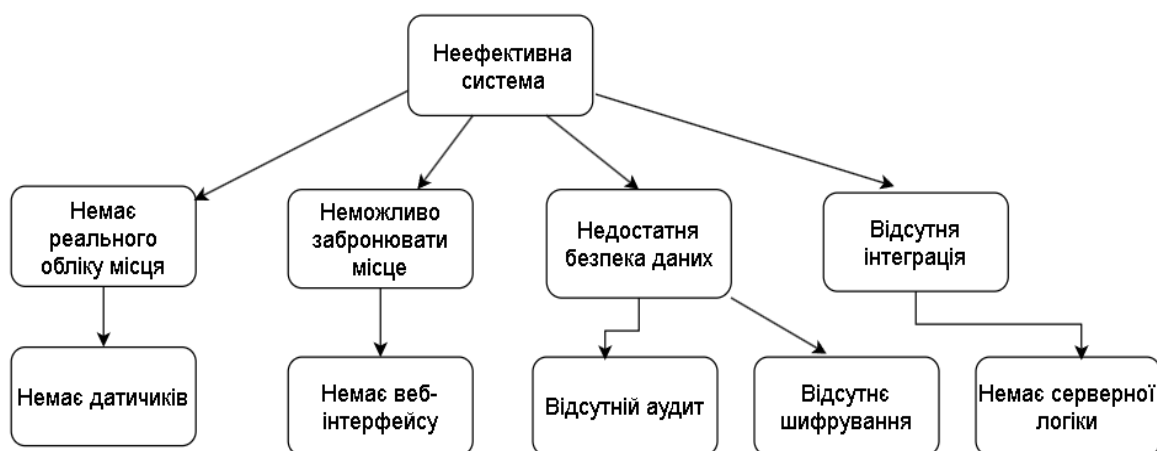


Рисунок 2.11 – Дерево проблем системи, яка створюється

2.3 Порівняння реалізацій та вибір архітектурного підходу

Побудова інтелектуальних систем передбачає в собі декілька типів паркування. Кожен з них має свої переваги та недоліки, які варто враховувати при

виборі архітектури для реалізації [8-11]. За порівнянням основних моделей в таблиці 2.1.

Таблиця 2.1 – Порівняльний аналіз систем

Модель реалізації	Опис	Переваги	Недоліки
Локальна система	Уся логіка зберігається на пристроях всередині парковки	Надійність, автономність	Обмежена доступність ззовні, складність оновлень
Агрегатор (хмарна система)	Клієнт звертається до зовнішнього сервісу через API	Широке покриття, масштабованість	Залежність від сторонніх сервісів
Гібридна публічно-локальна модель	Поєднання централізованого сервера і локального ПЗ у межах однієї БД	Гнучкість, відмовостійкість, контроль	Складність синхронізації у разі розриву зв'язку

У проєкті реалізовано комбінований підхід, який поєднує елементи централізованої архітектури з локальною обробкою критично важливих сценаріїв. Створено два типи клієнтів: вебінтерфейс для користувачів, що дозволяє переглядати доступні місця та здійснювати бронювання, та окрема локальна програма на основі Windows Forms, яка виконує роль контролера доступу при в'їзді на територію паркінгу.

Центральна логіка базується на вебзастосунку, який зберігає інформацію про паркінги, резервації та генерує QR-коди для верифікації. На стороні в'їзду використовується автономний застосунок, що має доступ до тієї ж бази даних і здатен самостійно перевіряти коди, фіксувати факти в'їзду або відмовляти в доступі. Якщо недоступний центральний вебсервіс, локальна програма продовжує роботу системи без втрати ключової функціональності.

Це забезпечує:

- централізоване управління з боку власника системи;
- можливість роботи як онлайн, так і в локальному режимі;
- зручність для користувача (через вебінтерфейс);
- надійність в обробці запитів на в'їзд навіть у випадку збою з'єднання;
- єдине джерело істини у вигляді загальної бази даних;
- простоту масштабування на інші паркінги.

Саме така архітектура дозволяє в подальшому додати додаткові компоненти - IoT-датчики, платіжні модулі або мобільний застосунок - без повної перебудови системи. Структура системи підтримує розширення, не ускладнюючи основні сценарії використання.

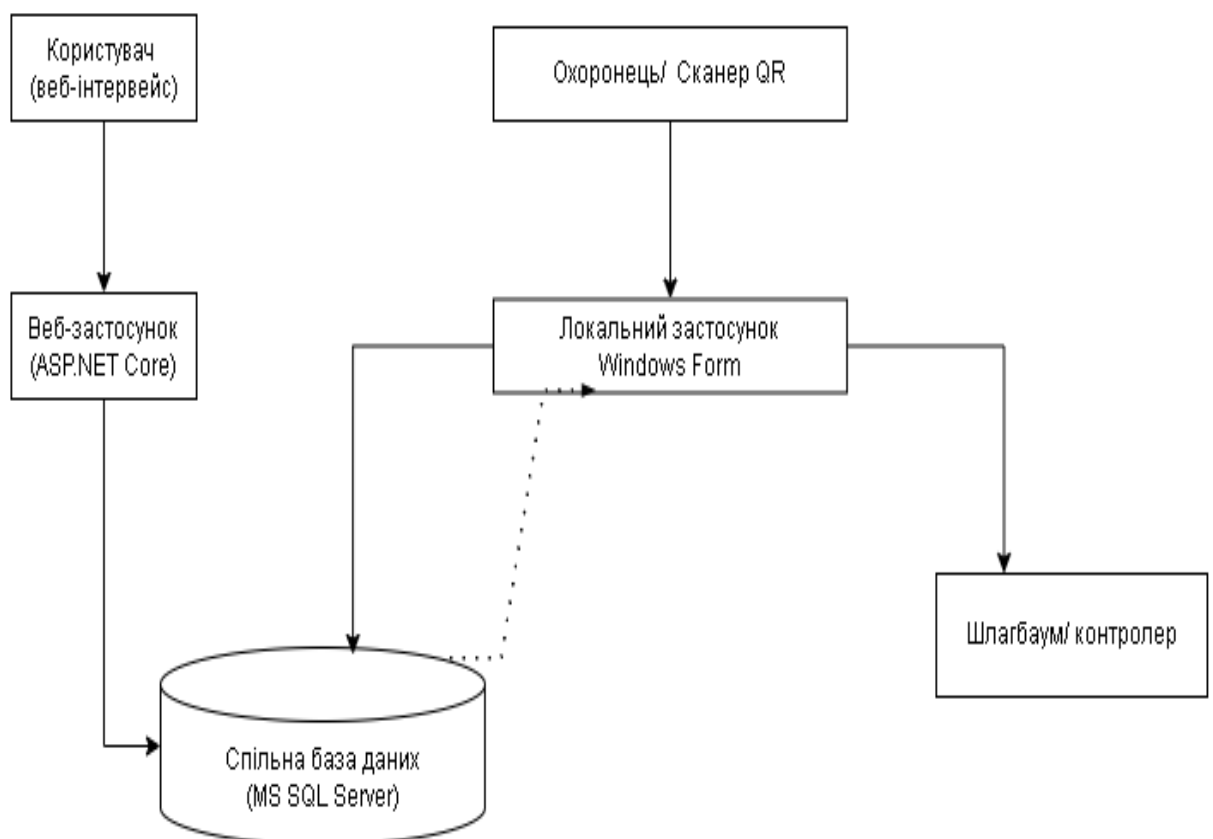


Рисунок 2.12 – Загальний алгоритм роботи системи с загальною базою даних

Така реалізація комбінує в собі простоту, контроль та стабільність подано на рисунку 2.12. Що досить важливо для ситуацій, де паркування пов'язане з доступом до приватної території або потребує високого рівня безпеки.

Висновок до розділу 2

Провівши аналіз типових архітектур сучасних систем паркування, отримали те, що найбільш придатною для реалізації у рамках проєкту - це модель із централізованим управлінням базою даних застосунками включає вебінтерфейс для користувача і локальним компонентом для взаємодії з відповідним сканером.

Через обмеження апаратної частини (відсутність реальних API-пристроїв, мережевих шлагбаумів або промислових сенсорів) система реалізується у вигляді програмної емуляції. Для перевірки наявності вільних і проведення моніторингу місць на парковці використовується логічна модель, де база даних містить перелік паркінгів, їхню місткість і бронювання з фіксованими датами початку та завершення. З цими даними виконується підрахунок зайнятих місць за обраний період часу.

Фактичне API, через яке в реальних умовах здійснювався б обмін даними між обладнанням і серверами, замінено на внутрішню логіку застосунків із прямим доступом до бази. Система працює у форматі моделі "клієнт-сервер", де серверна частина відповідає за обробку запитів, збереження інформації, перевірку QR-кодів і контроль вільних місць, а клієнтські застосунки (веб та локальний) - за взаємодію з користувачем та виконання сценаріїв входу на парковку.

Вибір QR-коду - це ідентифікації проста і має рішенням, що дозволяє передавати мінімальний набір інформації (наприклад, унікальний код резервації) і швидко здійснювати перевірку. Враховуючи, що система працює з локальною базою даних, усі операції перевірки відбуваються без необхідності звертання до зовнішніх сервісів, що підвищує її стабільність і дозволяє підтримувати функціональність навіть у разі втрати інтернет-з'єднання. Цей підхід поєднує

елементи централізованого керування даними з можливістю автономної роботи на стороні паркувального комплексу, що є оптимальним компромісом між функціональністю, простотою реалізації та відповідністю сучасним принципам побудови смарт-систем. Завдяки такій автоматизації система стає доступна та легка у використанні для будь-якого користувача. Основна логіка системи зосереджена у зберіганні інформації про доступні місця та їх моніторинг, резервацію обраної паркувальної зони та генерацію QR-коду для верифікації у системі. Класифікація паркувальних зон безпосередньо буде щзалежати від типу парковки та її вмісткості, для збільшених масштабів у кожного паркувального місця буде оснащено певними датчиками руху та спеціальних індикаторних позначка для зручності користувачів.

Такі паркувальні зони не потребуватимуть влаштованої людина за правила відповідальності за парковку та зарядні станції, адже ці процеси будуть повністю автоматизовані та матимуть дистанційне керування як і для керуючих так і користувачів.

3 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА СТВОРЕННЯ БАЗИ ДАНИХ

У цій частині дипломного проєкту розглянуто практичні аспекти реалізації вебсистеми для бронювання паркувальних місць. Зосереджено погляд на виборі інструментів розробки, мов програмування, фреймворків та бази даних, які стали основою для створення повноцінної функціональної системи. Описано архітектуру та логіку взаємодії між компонентами, структуру бази даних та особливості реалізації ключових модулів системи. Враховуючи, що проєкт включає як вебзастосунок для користувачів, так і локальну програму для верифікації резервацій через QR-код, вибір технологій мав забезпечити не лише функціональність, а й стабільність, безпеку та можливість масштабування у подальшому.

3.1 Вибір мов програмування та фреймворків

У сучасній практиці розробки програмного забезпечення існує велика кількість фреймворків та мов програмування, які дозволяють створювати як клієнтські, так і серверні застосунки. Серед популярних рішень для серверної частини (back-end) широко використовуються такі платформи, як Flask або Django (на базі мови програмування Python на рисунку 3.1), Node.js (на JavaScript), Spring Boot (на Java), Ruby on Rails, а також ASP.NET Core (на мові C#). Такі фреймворки мають свої сильні сторони та обмеження. Flask вирізняється легкістю налаштування та придатністю для невеликих проєктів, Django має потужну адміністративну панель і вбудовані механізми безпеки, а Node.js забезпечує високу швидкодію при роботі з асинхронними запитами.

Для проєкту було взяти рішення на користь ASP.NET Core у поєднанні з мовою програмування C#. Це високопродуктивна платформа від компанії Microsoft, яка дозволяє створювати масштабовані, безпечні та підтримувані вебзастосунки. Основними перевагами ASP.NET Core є:

					ІК11.150БАК.006 ПЗ	Арк.
						36
Зм.	Лист	№ докум.	Підпис	Дата		

- висока продуктивність при обробці запитів;
- потужна система маршрутизації та обробки HTTP-запитів;
- вбудована підтримка механізмів безпеки (аутентифікація, авторизація, захист від XSS/CSRF);
- ефективна інтеграція з ORM-фреймворком Entity Framework Core, що значно спрощує взаємодію з базою даних;
- підтримка системи міграцій, яка дозволяє керувати версіями схеми бази даних;
- активна спільнота розробників і наявність великої кількості документації та прикладів.

Важлив елементом для аргументування на користь вибору ASP.NET Core стала необхідність реалізації другого компоненту системи - локального застосунку для роботи з QR-кодами на базі Windows Forms. Обидва модулі могли бути реалізовані на одній мові (C#), що зменшило складність інтеграції та забезпечило узгодженість коду.

Для реалізації клієнтської частини системи обрано поєднання HTML/CSS та шаблонізатора Razor Pages, який є частиною ASP.NET Core. Альтернативними варіантами могли бути JavaScript-фреймворки (наприклад, React, Angular, Vue.js), які забезпечують побудову динамічних SPA-застосунків.

Для реалізації клієнтської частини системи обрано поєднання HTML/CSS та шаблонізатора Razor Pages, який є частиною ASP.NET Core. Альтернативними варіантами могли бути JavaScript-фреймворки (наприклад, React, Angular, Vue.js), які забезпечують побудову динамічних SPA-застосунків. Проте Razor Pages виявився оптимальним рішенням з огляду на цілі проєкту. Його переваги включають:

- чітке розділення логіки відображення та обробки подій;
- вбудовану підтримку серверного рендерингу, що позитивно впливає на продуктивність;
- глибоку інтеграцію з системою маршрутизації ASP.NET Core;

– зменшення залежності від сторонніх JavaScript-бібліотек.

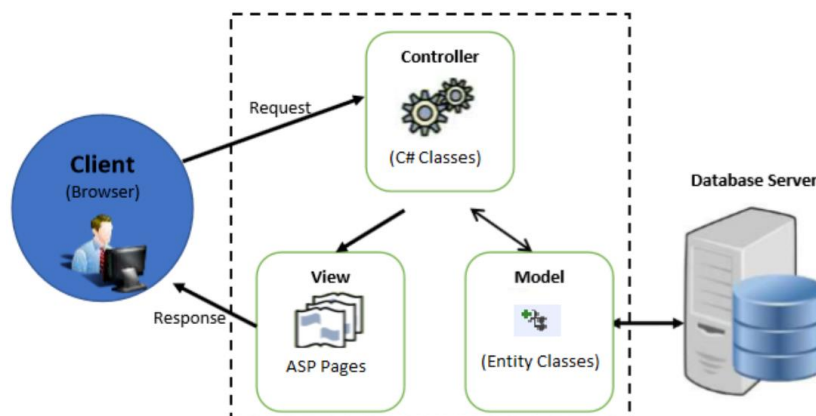


Рисунок 3.1 – MVC архітектура [5]

Razor Pages дозволяє розробляти сторінки з урахуванням принципів MVC (Model-View-Controller) на рисунку 3.1, що покращує структуру проєкту та полегшує його підтримку. Використання Razor Pages зменшує час завантаження сторінок і покращує індексацію сайту пошуковими системами, що може бути корисним у випадку публічного доступу до ресурсу.

3.2 Використання засобів для керування базою даних

Важливим етапом у проєкті є вибір системи керування базами даних (СКБД), яка відповідає вимогам до збереження та обробки інформації про користувачів, бронювання та паркувальні місця. У ході розробки було проаналізовано низку можливих СКБД, серед яких MySQL, PostgreSQL, SQLite, Oracle та Microsoft SQL Server.

Після порівняння переваг і недоліків різних варіантів було прийнято рішення на користь MS SQL Server. Основними аргументами стали:

- зручна інтеграція з платформою .NET та Entity Framework Core;
- висока продуктивність у багатокористувацьких сценаріях;

- підтримка транзакцій, представлень, тригерів та інших механізмів корпоративного рівня;

- хороша масштабованість — система може бути розгорнута як на локальному сервері, так і в хмарному середовищі;

- наявність розвинених засобів адміністрування (SQL Server Management Studio, аналітика запитів, бекапи тощо).

Для взаємодії з базою даних було використано ORM-фреймворк Entity Framework Core (EF Core). Альтернативним шляхом могла бути розробка системи з ручним написанням SQL-запитів або використанням маперів на кшталт Dapper. Такі підходи дають більше контролю, однак значно ускладнюють підтримку коду, особливо при змінах у структурі даних. EF Core забезпечує автоматичне перетворення об'єктно-орієнтованих моделей у таблиці БД та дозволяє працювати з даними через LINQ-запити, зменшуючи ризик помилок при формуванні SQL-інструкцій.

EF Core підтримує систему міграцій на рисунку 3.2, що дозволяє легко розгортати та оновлювати схему бази даних відповідно до змін у програмному коді. Міграції створюються з консолі або автоматично при запуску, забезпечуючи синхронізацію моделей і БД.

У реалізації проєкту створення бази даних є ключовим етапом, який може бути реалізований кількома різними способами. Вибір методу залежить від підходу до розробки, масштабу проєкту та потреб у подальшому оновленні структури бази. В розробленій системі використано Code First і представлено на кресленику ІК11.150БАК.006 Д2.

У сучасній практиці з використанням Entity Framework Core існують три основні підходи до створення бази даних:

- Database First — база даних створюється заздалегідь вручну або в середовищі СКБД, після чого виконується генерація моделей у коді;

- Code First із міграціями — розробник спочатку створює класи моделей, які відображають сутності, а потім за допомогою міграцій формує базу даних;

– Code First з автоматичним створенням бази — структура бази створюється одразу при запуску програми без використання механізму міграцій, через метод EnsureCreated().

Кожен із підходів має свої особливості:

– Database First є доцільним, коли вже існує або розробляється складна БД з великою кількістю зв'язків, і потрібна генерація моделей без ручного кодування;

– Code First з міграціями підходить для активної розробки, коли структура БД змінюється разом з еволюцією коду;

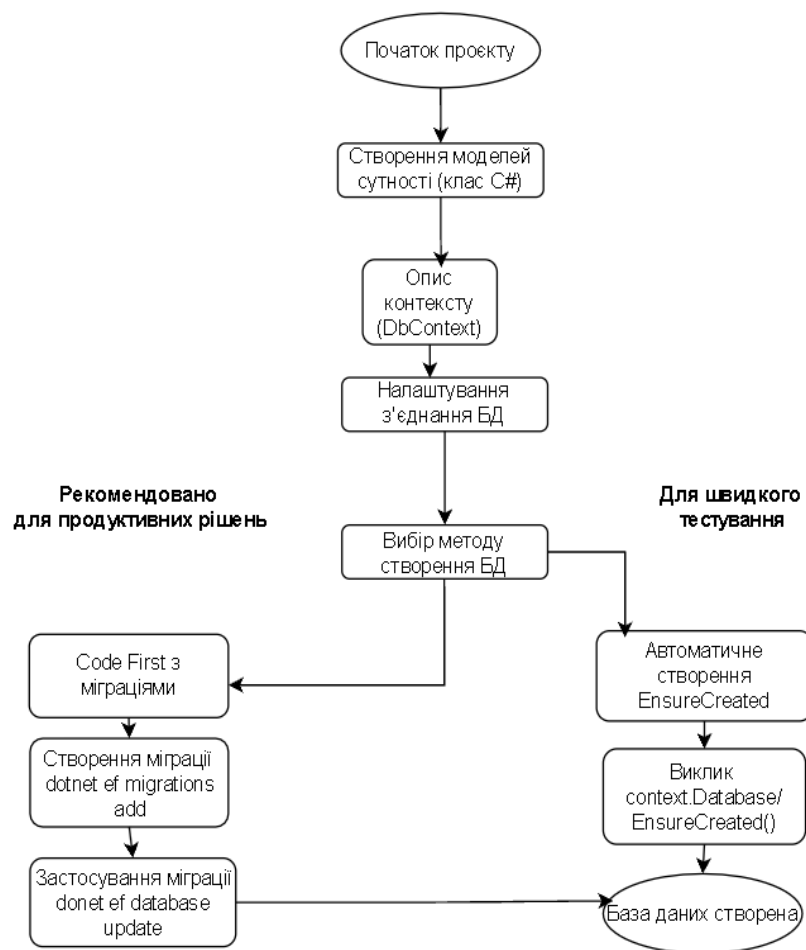


Рисунок 3.2– Основні підходи до створення бази даних

– EnsureCreated() підходить для невеликих проєктів, тестування або швидкого прототипування.

У проєкті застосовано підхід Code First, з підтримкою міграцій, оскільки він забезпечує гнучкість, контроль над структурою БД, можливість оновлення схеми без втрати даних, а також дозволяє команді працювати без необхідності ручного створення SQL-скриптів. [12]

3.3 Структура бази даних

База даних, яка лежить в основі розробленої вебсистеми бронювання паркувальних місць, є логічно структурованим сховищем, побудованим згідно з принципами реляційного моделювання. Її структура орієнтована на забезпечення ефективного зберігання, доступу та обробки інформації, що стосується паркувальних майданчиків та пов'язаних із ними бронювань. Реалізація здійснена з використанням ORM-фреймворку Entity Framework Core, що дозволило безпосередньо пов'язати об'єктно-орієнтовані моделі застосунку зі структурою таблиць у СКБД Microsoft SQL Server.

У системі визначено дві основні сутності: ParkingLot та Reservation, які відповідають відповідним таблицям у базі даних. Відповідність між моделями та таблицями забезпечується через механізми конфігурації контексту бази даних (DbContext), де задаються правила побудови зв'язків між сутностями.

Сутність ParkingLot репрезентує окремий паркувальний майданчик. Вона має такі атрибути:

- Id: первинний ключ, який однозначно ідентифікує кожен запис;
- DescriptionJson: текстове поле у форматі JSON, що містить опис паркування (розташування, зручності, координати);
- TotalSpots: загальна кількість доступних паркувальних місць;
- DailyRate: вартість одного дня паркування;
- Reservations: навігаційна властивість, що задає зв'язок "один-до-багатьох" із сутністю Reservation.

Модель дозволяє зберігати як загальні характеристики паркінгу, так і гнучко розширювати опис завдяки формату JSON, що є зручним при візуалізації або імпортуванні структурованих даних.

Сутність Reservation описує окрему бронь, пов'язану з конкретним паркінгом.

Вона включає наступні поля:

- Id: первинний ключ бронювання;
- ParkingLotId: зовнішній ключ, що вказує на пов'язану сутність ParkingLot;
- ParkingLot: навігаційна властивість, яка забезпечує об'єктний зв'язок між резервацією та паркінгом;
- PublicCode: унікальний публічний код (10 символів), який генерується при створенні бронювання та використовується в електронному талоні;
- FullName: ПІБ особи, яка здійснила бронювання;
- CarNumber: державний номер автомобіля;
- DateStart / DateEnd: період, на який здійснене бронювання;
- ServiceStartTime / ServiceEndTime: додаткові поля, які використовуються для фіксації фактичного часу надання послуги (наприклад, час в'їзду/виїзду), можуть бути null, якщо бронювання ще не активовано.

Сутність ReservationCharge описує бронювання електростанцій. Вона включає наступні поля:

- Id: первинний ключ зарядної станції;
- Charging Id: зовнішній ключ, що вказує на пов'язану сутність ChargingLot;
- ChargingLot: навігаційна властивість, яка забезпечує об'єктний зв'язок між резервацією та зарядною станцією;
- PublicCode: унікальний публічний код (10 символів), який генерується при створенні бронювання та використовується в електронному талоні;
- FullName: ПІБ особи, яка здійснила оплату зарядної станції;
- StatusCharge: показує залишок електроенергії;
- TypeCharge: вказує на тип зарядної станції;

- CarNumber: державний номер автомобіля;
- DateStartCharge / DateEndCharge: період, на який здійснене бронювання на зарядну станцію;
- ServiceStartTimeCharge/ServiceEndTimeCharge поля, які використовуються для фіксації фактичного часу надання послуги.

Зв'язок між сутностями реалізовано у методі OnModelCreating класу ParkingDbContext. Зокрема, задається відношення один-до-багатьох між ParkingLot та Reservation і ReservationCharge. Ця конфігурація означає, що при видаленні об'єкта ParkingLot та ChargingLot усі пов'язані з ними резервації також будуть автоматично видалені. Забезпечується цілісність даних і уникається накопичення "сирітських" записів у таблиці бронювань паркомісця та зарядної станції. Без особливих труднощів додати нові сутності, такі як User, Payment, AdminLog, або ж розширити поточні, не порушуючи цілісності системи.

Тобто архітектура розробленої бази даних здійснює зберігання даних для управління паркувальними зонами та зарядними станціями.

3.4 Логіка роботи системи

Логіка функціонування вебсистеми побудована навколо основного сценарію взаємодії користувача з сервісом бронювання паркувальних місць. Уся робота системи базується на чітко визначеному життєвому циклі резервації, який охоплює взаємодію з вебзастосунком, базою даних та програмою, встановленою на в'їзному пункті (Windows Forms-застосунок для охорони). Основні етапи життєвого циклу резервації на рисунку 3.3.

Програма починається з перегляду доступних паркомісць, користувач відкриває вебзастосунок і бачить список доступних паркінгів. На цьому етапі відбувається завантаження з бази даних сутностей ParkingLot, які включають загальну кількість місць, опис у форматі JSON, розцінки та іншу інформацію. Наступний крок це вибір дати, після вибору паркінгу користувач обирає бажану

дату або діапазон дат для бронювання. Цей крок є обов'язковим для подальшого обчислення наявності вільних місць.

Користувач здійснює перевірку зайнятих місць, а система виконує запит до таблиці Reservation, щоб визначити кількість уже зайнятих місць у заданий період. Фільтрація здійснюється за датами DateStart та DateEnd, які перетинаються з обраним інтервалом. Якщо доступні місця на зазначені дати відсутні, користувач отримує відповідне повідомлення. У разі відсутності активної сесії користувач повинен пройти автентифікацію. У випадку нових користувачів можлива реєстрація з подальшим збереженням даних у відповідній таблиці (наприклад, Users, якщо реалізовано).

Збереження нової резервації, якщо місце доступне, користувач переходить до форми бронювання, де вводить своє повне ім'я (FullName) та номер автомобіля (CarNumber).

Після натискання кнопки підтвердження створюється новий запис у таблиці Reservation, пов'язаний із відповідним ParkingLotId. Одночасно із збереженням резервації система генерує унікальний 10-символьний код (PublicCode). Архітектурну контекстну базу даних представлено на кресленику ІК11.150БАК.006 ДЗ, яка чітко розкриває структуру архітектури взаємодії розробленого застосунку.

Який виконує роль ключа перевірки на в'їзді. Код є унікальним і може бути використаний як у зашифрованому вигляді, так і як посилання на сервер для перевірки статусу бронювання.

Потім починається створення QR-коду - на основі PublicCode формується електронний талон у вигляді QR-коду. Для генерації зображення використовується бібліотека QRCode. У коді може бути зашифровано або сам PublicCode, або URL із параметром (наприклад, /check?code=XYZ1234567), що веде до сервера перевірки. На в'їзді охоронець або автоматизована система сканує QR-код за допомогою спеціального застосунку (Windows Forms), який надсилає HTTP-запит до серверної частини.

					ІК11.150БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

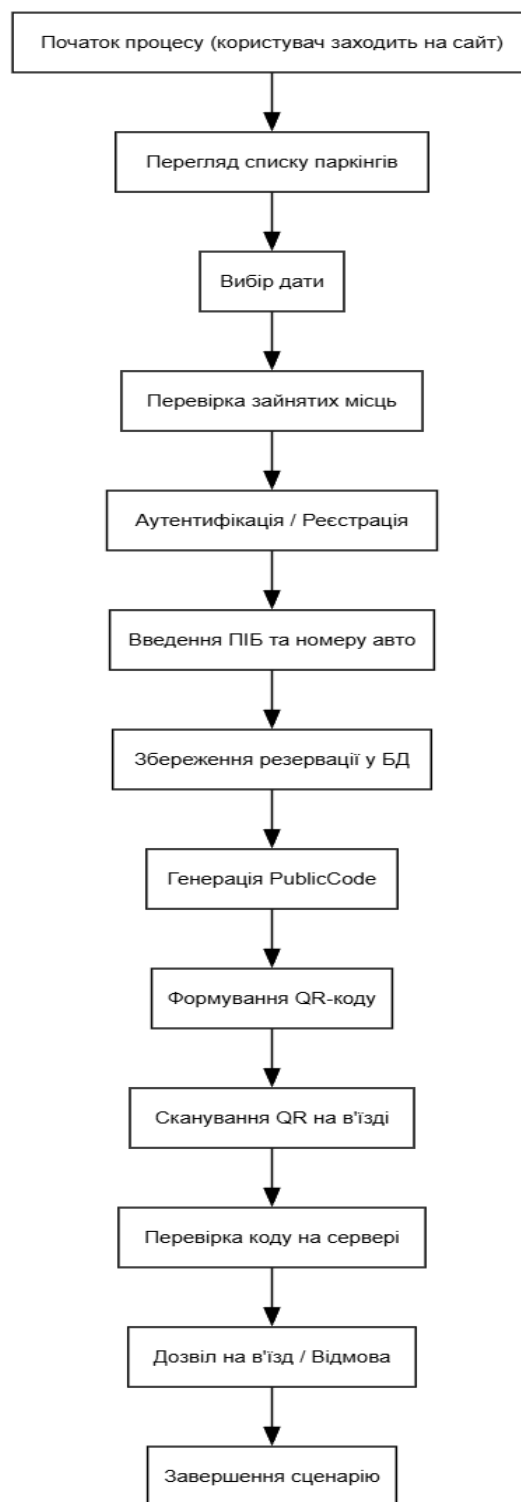


Рисунок 3.3– Основні етапи життєвого циклу резервації

Система виконує перевірку:

- чи існує бронювання з таким PublicCode;
- чи дата в'їзду (DateStart) відповідає поточному часу;

Зм.	Лист	№ докум.	Підпис	Дата

– чи резервація ще не завершена

Висновок до розділу 3

Розглянули практичні аспекти реалізації майбутньої вебсистеми. Проаналізували можливий функціонал та рішення. Для серверної частини розробки застосунка було обрано рішення на користь ASP.NET Core у поєднанні з мовою програмування C#. Це сучасна платформа від компанії Microsoft, яка дозволяє створювати масштабовані, безпечні та підтримувані вебзастосунки, яка чудово підходить для початкової реалізації як і для вебсайту, так і для компоненту програми - сканера, яка буде розроблена на базі Windows Form. Клієнтська частина поєднується HTML/CSS та шаблонізатор Razor Pages, в ході підбору також було проаналізовано інші вибір стеку технологій. Сформувавши дві частини і зібравши всі переваги, базою даних застосунку є СКБД MS SQL Server, яка дуже зручно інтегрується з платформою .NET та Entity Framework Core. З поєднаною компонентою для сканування коду верифікації – отримуємо повноцінну систему, яка призначена для керування інтелектуальною робототехнічною системою управління зарядними станціями в смарт-місті.

Інфраструктура проекту має потенціал для опрацювання великих даних за допомогою обраних засобів для розробки і додатково налаштованих датчиків, доданих нових пристроїв і оснащення паркувальних зон спеціалізованими датчиками.

Дана система є робототехнічною, адже в ній є компоненти такі як: сенсори та датчики з віддаленим керуванням. Також розробка включає в себе автоматизоване програмне забезпечення, яке автоматизує процес керування паркувальними місцями та зарядними станціями.

					ІК11.150БАК.006 ПЗ	Арк.
						46
Зм.	Лист	№ докум.	Підпис	Дата		

4 РОЗРОБЛЕННЯ СИСТЕМИ ТА ОПИС ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ

В цій частині дипломного проєкту буде проведено розробку систем та створено реалізацію вебсистеми та додаткового компоненту для сканування QR-кодів для відкриття доступу для заїзду шлагбаума. Для даної розробки буде загально два актори на користування. Система заключається у віддаленому керування як з боку адміністратора так і безпосередньо користувача. Функціональність робототехнічної системи представлена на кресленику ІК11.150БАК.006 Д1.

4.1 Особливості реалізації коду

У даному підрозділі розглянуто основні аспекти реалізації вебзастосунку для бронювання паркомісць. Реалізація здійснена з використанням фреймворку ASP.NET Core Razor Pages, що дозволяє ефективно поєднувати логіку, маршрутизацію та відображення в єдиній структурі. Для збереження даних застосовується Entity Framework Core.

4.2 Реалізація вебзастосунку

Вебзастосунок схема якого представлена на кресленику ІК11.150БАК.006 Д4 складається з кількох сторінок, головні з яких - **Index.cshtml** (головна сторінка з пошуком паркінгів) та **Details.cshtml** (оформлення бронювання) схему зображено на рисунку 4.1. Вони обробляються відповідними моделями сторінок (IndexModel та DetailsModel), які взаємодіють із базою даних через ParkingDbContext.

Генерація унікального коду PublicCode. Кожна резервація отримує унікальний публічний код, що використовується як ідентифікатор під час

верифікації на в'їзді. Генерація коду реалізована за допомогою GUID, обрізаного до 8 символів:

Цей код на рисунку 4.1 зберігається у полі PublicCode таблиці Reservation та включається у URL, який кодується у QR-форматі.

Розрахунок кількості доступних місць на рисунку 4.2. Під час вибору діапазону дат система виконує динамічний розрахунок кількості вільних місць на основі поля TotalSpots у таблиці ParkingLot та кількості записів у таблиці Reservation, що перетинаються за часом:

Користувач бачить лише ті паркінги, які мають наявні місця для заданого періоду. Після створення резервації користувача перенаправляють на сторінку Ticket, яка відображає згенерований QR-код на рисунку 4.3. QR-код кодує унікальний PublicCode, що міститься в URL типу /Check?code=XYZ12345. Для генерації QR-коду використовується стороння бібліотека QRCoder на рисунку 4.4.

4.3 Особливості локальної програми для управління паркінгом

Для автоматизованого контролю доступу до паркінгу реалізовано окрему клієнтську програму, яка імітує роботу сканера QR-кодів на в'їзді та виїзді. Ця програма розроблена як WinForms-додаток на платформі .NET і має доступ до тієї ж бази даних, що й основний вебдодаток. Зберігається цілісність даних між компонентами системи.

Користувач натискає кнопку "Сканування на в'їзд" або "Сканування на виїзд", що імітує початок зчитування QR-коду за допомогою вебкамери. Програма з використанням бібліотек AForge та ZXing аналізує відеопотік з камери, декодує QR-код та отримує унікальний публічний код резервації (PublicCode). За отриманим кодом здійснюється звернення до бази даних (через ParkingDbContext), де:

– перевіряється наявність відповідного запису резервації;

– перевіряється, чи поточна дата входить у дозволений діапазон між DateStart і DateEnd;

– у випадку в'їзду, фіксується час фактичного початку обслуговування у полі ServiceStartTime;

– у випадку виїзду, фіксується ServiceEndTime.

При успішній перевірці запускається відеофайл відкриття шлагбаума (Шлагбаум.mp4) у вікні програми. Для цього використовується елемент axWindowsMediaPlayer. Імітується фізичне відкриття доступу на територію паркінгу. Інформація про результат перевірки, включно з ПІБ користувача, держаним номером автомобіля та часом, виводиться у текстову консоль форми (RichTextBox), що може використовуватись як внутрішній журнал подій.

Це дозволить проводити демонстрацію системи без фізичного обладнання (шлагбауму та сканера), що особливо важливо в умовах тестування та розробки.

Зчитування QR-коду вебкамерою та використання бібліотеки AForge.NET: особливості реалізації. Одним із ключових компонентів клієнтської частини системи автоматизованого доступу до паркінгу є модуль зчитування QR-кодів за допомогою вебкамери. Цей модуль реалізовано з використанням бібліотеки AForge.NET, яка забезпечує взаємодію з відеопристроями, а також бібліотеки ZXing.NET - для декодування QR-кодів із захоплених кадрів [8].

Функцією цього модуля є безперервне зчитування відеопотоку з підключеної вебкамери, аналіз кожного кадру в реальному часі, та виявлення у ньому QR-коду. У випадку успішного декодування коду, процес зчитування припиняється, і програма переходить до подальшої обробки отриманого публічного коду (PublicCode), який використовується для перевірки резервації в базі даних.

AForge.NET - це бібліотека з відкритим кодом, призначена для комп'ютерного зору, обробки зображень і роботи з відеопотоком. У даному контексті вона використовується для доступу до відеопристроїв (зокрема вебкамер), захоплення кадрів у реальному часі, а також обробки цих кадрів для подальшого аналізу [20].

Реалізація відбувається так :

За допомогою класу `FilterInfoCollection` проводиться сканування всіх відеопристроїв, доступних у системі. Вибирається перша доступна камера (`VideoCaptureDevice`) і підключається обробник події `NewFrame`.

Подія `NewFrame` спрацьовує щоразу, коли з камери надходить новий кадр. Ці кадри надходять у вигляді об'єктів `Bitmap`; Отриманий кадр зберігається у змінну `currentFrame`, а також відображається на екрані користувача в `PictureBox`. У програмі використовується таймер на рисунку 4.6 (`System.Windows.Forms.Timer`), який із фіксованим інтервалом (500 мс) на рисунку аналізує останній отриманий кадр; Такий підхід дозволяє уникнути надмірного навантаження на систему, при цьому забезпечуючи достатню частоту оновлення для комфортної роботи.

Щоб уникнути конфліктів при доступі до змінної `currentFrame`, використовується механізм блокування (`lock`), що гарантує потокобезпечне клонування зображення перед передачею до QR-декодера.

Після отримання кадру, його обробляє об'єкт типу `BarcodeReader` із бібліотеки `ZXing.NET`. [19] Декодер здійснює пошук QR-коду на зображенні і, у випадку успіху, повертає розпізнаний текст (тобто публічний код резервації)

Особливості роботи QR-декодера:

- декодування здійснюється на повнокольоровому або чорно-білому зображенні;
- при успішному розпізнаванні подальша обробка відеопотоку зупиняється, щоб уникнути повторного зчитування;
- у випадку неуспішної спроби, процес продовжується автоматично до моменту зчитування або вручну зупиняється оператором.

Перевагою використання бібліотеки `AForge.NET` є її сумісність із різними моделями камер, висока стабільність при обробці відеопотоку та можливість масштабування для роботи з декількома відеопотоками одночасно. Вона також дозволяє керувати якістю кадрів, частотою оновлення та розміром відео, що є важливим у середовищах з обмеженими ресурсами.

Зчитування у реальному часі дозволяє оператору паркінгу здійснювати контроль доступу без необхідності втручання у внутрішню логіку системи — достатньо лише навести QR-код до камери, після чого всі перевірки і дії виконуються автоматично.

До недоліків реалізації можна віднести обмежену швидкість обробки, яка залежить від потужності комп'ютера та якості відео. У майбутньому можливе впровадження більш сучасних бібліотек, зокрема EmguCV (обгортка OpenCV), які забезпечують апаратне прискорення та ширші можливості аналізу зображень.

4.4 Опис отриманого функціоналу та тестування системи

У програмної частини системи було створено повноцінний функціональний цикл, який охоплює усі етапи взаємодії користувача з системою паркування - від попереднього бронювання місця до безконтактного в'їзду та виїзду з паркінгу за допомогою QR-коду. Це дозволяє комплексно перевірити коректність роботи всіх компонентів системи в інтегрованому середовищі, а також провести тестування логіки взаємодії між клієнтською частиною, сервером і базою даних.

Щоб протестувати систему у повному обсязі, слід здійснити повний цикл дій, що імітує типовий сценарій користувача. Першим кроком у цьому ланцюжку є створення нової резервації місця на паркінгу. Для цього користувач запускає головний вебдодаток, після чого на екрані з'являється стартова сторінка з інтерфейсом для вибору параметрів майбутньої резервації.

Інтерфейс надає можливість:

- обрати конкретну паркувальну локацію зі списку доступних рисунків 4.1;
- задати дату початку та завершення резервації із застосуванням зручного календарного компонента;
- В опціях бачимо, що для даного паркомісця передбачена зарядна станція і після сканування коду на місці ми отримуємо інформацію

- ввести основні персональні дані – рисунок 4.3 : ПІБ, номер телефону, номер транспортного засобу;
- ініціювати генерацію QR-коду, який буде використовуватись у майбутньому як цифровий талон для входу в систему автоматичного контролю доступу зображено на рисунку 4.4 та рисунок 4.5

Рисунок 4.1 – Перша сторінка WEB додатку – вибір дати та місця

Після заповнення всіх необхідних полів та підтвердження, система формує новий запис у базі даних, присвоює резервації унікальний ідентифікатор (PublicCode) та створює відповідне зображення QR-коду, яке користувач може зберегти або роздрукувати зображено на рисунку 4.2

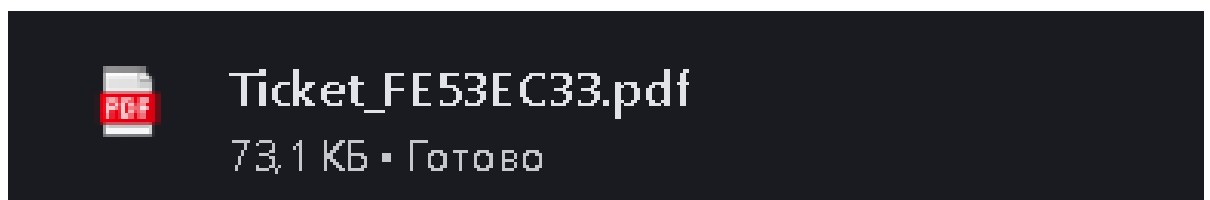


Рисунок 4.2 –Завантаження пдф квитка

ParkingReservation

Home

Privacy

Hello, !

Бронювання парковки

Адреса: ул. Харківська, 1

Період: 27.05.2025 — 28.05.2025

Опції: цілодобова, електрозарядка, крытая

Ціна: 120,00 грн

Ім'я користувача:

Анастасія

Номер авто:

A101170A

Номер картки (муляж):

0000 0000 0000 0000

Підтвердити бронювання

© 2025 - ParkingReservation - [Privacy](#)

Рисунок 4.3 – Друга сторінка WEB додатку – введення даних

Після заповнення всіх необхідних полів та підтвердження, система формує новий запис у базі даних, присвоює резервації унікальний ідентифікатор (PublicCode) та створює відповідне зображення QR-коду, яке користувач може зберегти або роздрукувати

ParkingReservation

Home

Privacy

Hello, !

Ваш талон на паркування

Ім'я: Анастасія

Номер авто: A101170A

З: 25.05.2025

По: 28.05.2025

Адреса паркінгу: ул. Харківська, 1



Збережіть цей талон або покажіть його при в'їзді.

Завантажити PDF

© 2025 - ParkingReservation - [Privacy](#)

Рисунок 4.4 – Третя сторінка WEB додатку – генерація талону

Паркувальний талон

Ім'я: Анастасія

Авто: AI0117OA

З: 25.05.2025

По: 28.05.2025

Адреса: ул. Харківська, 1



Покажіть QR-код на в'їзді

Рисунок 4.5 – Паркувальний талон завантажений у пдф

Відповідно цей талон користувач сканує при заїзді на паркувальну зону і отримує місце для паркування

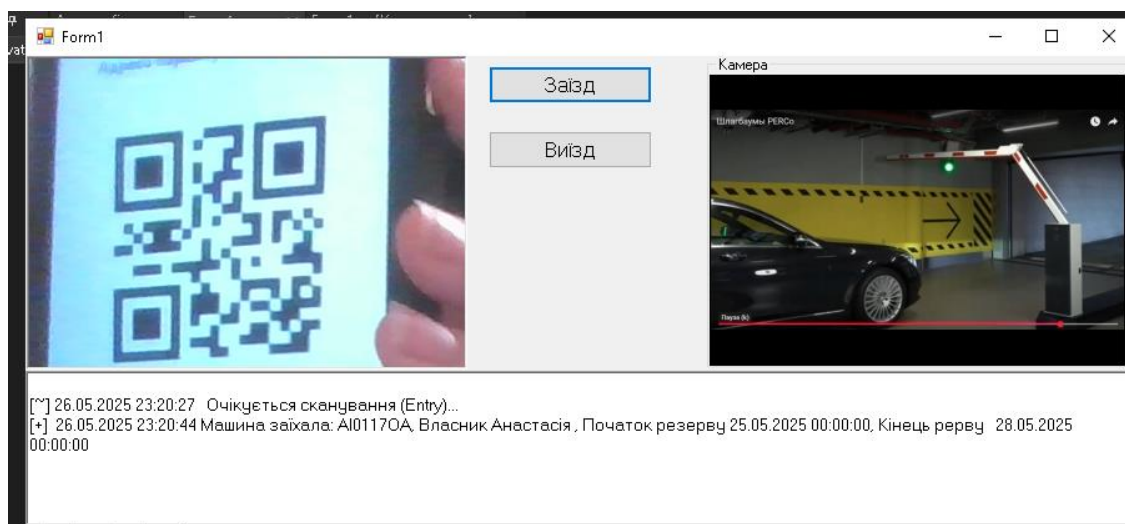


Рисунок 4.6 – Контроль талону заїзду

					ІК11.150БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		54

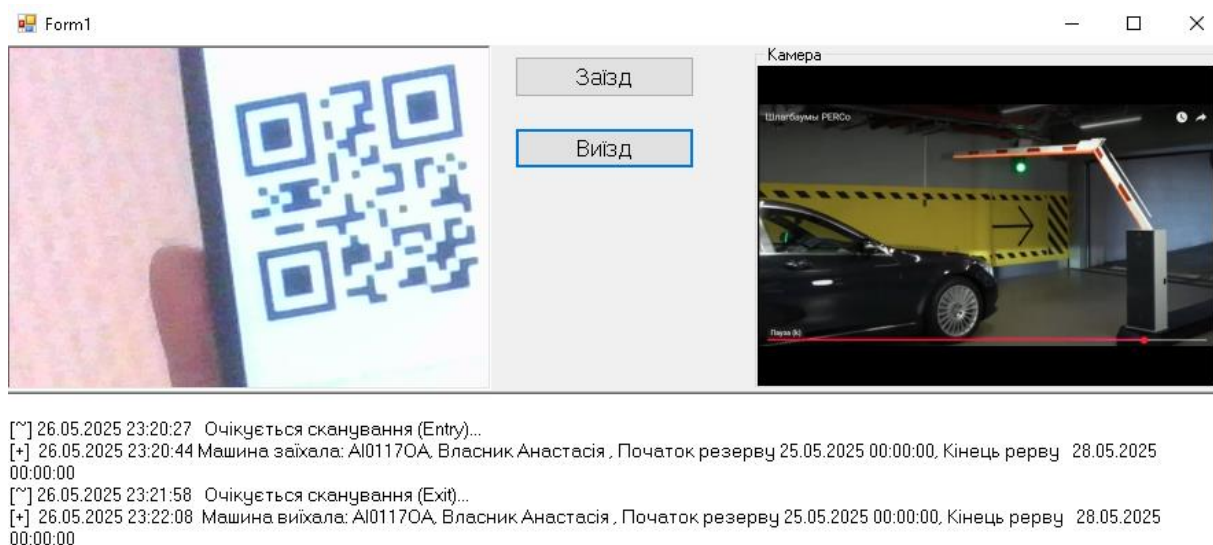


Рисунок 4.6 – Контроль талону виїзду

Після здійснення резервації та отримання користувачем QR-коду, наступним кроком є імітація реального сценарію в'їзду на територію паркінгу. Для цього використовується окрема клієнтська програма, яка виконує роль сканувального терміналу на в'їзді на рисунку 4.6 та рисунку 4.7.

Інтерфейс цієї програми розроблений у формі настільного Windows-додатку, що має інтуїтивно зрозуміле вікно. Основними візуальними елементами інтерфейсу є:

- відеопотік з вебкамери, який відображається у реальному часі (компонент PictureBox);
- дві кнопки керування режимами сканування — для в'їзду та виїзду з паркінгу;
- інформаційне текстове поле (RichTextBox), у якому відображаються хронологічні події та діагностичні повідомлення системи (наприклад, успішне зчитування, помилки часу, відсутність резервації тощо);
- відеоплеєр (Windows Media Player), який демонструє коротке відео відкриття шлагбаума після успішної авторизації транспортного засобу.

Для перевірки роботи програми в умовах, наближених до реального використання, користувач демонструє камері екран мобільного телефону з

відображеним QR-кодом, отриманим після резервації. Система виконує безперервне захоплення кадрів з камери у реальному часі з використанням бібліотеки AForge.Video, після чого кожні 500 мілісекунд запускається спроба розпізнавання QR-коду за допомогою бібліотеки ZXing.Net. У разі успішного розпізнавання коду програма:

- зупиняє відеопотік для економії ресурсів;
- надсилає запит до бази даних для пошуку резервації з відповідним PublicCode;
- перевіряє коректність діапазону дат, у межах яких дозволено в'їзд;
- залежно від режиму (в'їзд або виїзд) фіксує відповідну подію (ServiceStartTime або ServiceEndTime);
- відображає детальне повідомлення у журналі подій;
- запускає коротке відео з імітацією відкриття шлагбаума.

Цей функціонал наочно демонструє, як засоби комп'ютерного зору й доступу до бази даних інтегруються для автоматизованого пропуску транспортних засобів. Імітація демонструє сценарій, максимально наближений до реального паркувального термінала з камерою на в'їзді.

Форма додатку побудована так, щоб будь-який оператор або адміністратор міг легко контролювати процес сканування та при необхідності вручну перезапустити зчитування.

Висновок до розділу 4

У результаті реалізації третього розділу було створено повнофункціональну систему управління паркувальними резерваціями, що складається з двох окремих застосунків - вебсервісу для користувачів та адміністративної програми для контролю доступу. Обидва компоненти інтегруються через спільну базу даних. Було проведено тестування повного циклу взаємодії: від створення резервації до

зчитування QR-коду на в'їзді й фіксації подій у базі. Система успішно пройшла відлагодження та функціонує відповідно до заданих вимог.

В результаті ми отримуємо систему, де спочатку користувач заходить на сайт або сканує OR- код, поблизу паркувальної зони оснащеної зарядними станціями. Наступним чином відбувається обрання відповідної функції, що і включає в себе керування паркувальними місцями та зарядними станціями. Потім обираємо для потрібну категорію для клієнта категорію і вводимо дані. Якщо було обрано випадок із зарядною станцією, то користувач має змогу, вибрати тип, переглянути скільки лишилося електроенергії та отримати приблизний час до повної зарядки в залежності від обраної станції.

На виході нашої системи користувач отримує згенерований QR- код, який містить в собі основу інформацію, таку як:

- ПІБ;
- дата та час заїзду;
- заброньована місце;
- вказана адреса;
- тип зарядної станції;
- залишок електроенергії;
- номер зарядної станції;

Користувач може керувати зарядною станцією віддалено, зупиняти або подавити живлення, також за допомогою камер оснащених штучним інтелектом може час від часу робити перегляд активного запису з камер і слідкувати за своїм авто.

Відповідно потім він сканує код за допомогою розробленої компоненти, яка продемонстрована в роботі і активує бронювання і систему світлових показників на паркувальній зоні.

ВИСНОВКИ

Метою даної роботи було моделювання та реалізація імітованої системи бронювання паркувальних місць з подальшим контролем доступу до паркінгу. Робота виконувалась у контексті тенденцій «розумного міста» та автоматизації міської інфраструктури. У результаті реалізовано повнофункціональну демонстраційну модель, що складається з:

- вебзастосунку для створення та адміністрування резервацій паркувальних місць;
- серверної частини з централізованою базою даних;
- локального програмного забезпечення для управління в'їздом і виїздом на паркінг на основі зчитування QR-кодів у режимі реального часу.

Було розглянуто сучасні системи паркування, їх переваги, а також актуальні виклики у сфері організації паркувального простору, проведено аналіз існуючих архітектур і обґрунтовано вибір клієнт-серверної моделі з централізованим зберіганням даних і автономними компонентами доступу. Обрані технології дозволили реалізувати гнучку та розширювану структуру.

Уся поставлена мета роботи була досягнута. Розроблена система є робочим прототипом, який може бути масштабований та адаптований до реального середовища. Попри деякі спрощення, характерні для навчального проєкту, вона демонструє ключові принципи побудови сучасних паркувальних сервісів та відкриває перспективи для подальшого розвитку та впровадження в міську інфраструктуру. Завдяки розробленого процесу вирішується багато проблем, які пов'язані зі станом екології та стихійних паркувальних місць. Вебзастосунок зі своєю компонентою сканером представляє реалізацію, яка немає аналогів, сумісна двом напрямкам: паркування та зарядні станції.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Litman, T. (2022). Parking Management: Strategies, Evaluation and Planning. Victoria Transport Policy Institute. [Електронний доступ]. Режим доступу: https://www.vtpi.org/park_man.pdf
2. Sini K. (2020). Smart Parking Systems and Real Time Data Analytics. IEEE Xplore. [Електронний доступ]. Режим доступу: <https://ieeexplore.ieee.org/sta>
3. Gharbaoui, M. T., Hossain, M. S., & Muhammad, G. (2017). Intelligent Parking Cloud Services Based on IoT. [Електронний доступ]. Режим доступу: <https://arxiv.org/pdf/1706.01747.pdf>
4. Shaheen, S., & Cohen, A. (2020). Innovative Mobility: Carsharing Outlook. UC Berkeley. [Електронний доступ]. Режим доступу: <https://escholarship.org/>
5. European Commission. (2022). Urban Mobility: Smart Parking Policies. [Електронний доступ]. Режим доступу: <https://transport.europa.eu/system/.pdf>
6. IoT-Based Smart Parking System. International Journal of Engineering Research & Technology (IJERT). [Електронний доступ]. Режим доступу: <https://www.ijert.org/research/iot-based-smart-parking-system>
7. Ghosh, A., Basu, A., & Mahanti, N. C. (2019). Design and Implementation of QR-Code Based Car Parking System. [Електронний доступ]. Режим доступу: <https://www.irjet.net/archives/V6/i4/IRJET-V6I4350.pdf>
8. ZXing (Zebra Crossing) — Barcode Image Processing Library. [Електронний доступ]. Режим доступу: <https://github.com/zxing/zxing>
9. AForge.NET Framework Documentation. [Електронний доступ]. Режим доступу: <http://www.aforgenet.com/framework/>
10. Microsoft. (2023). Entity Framework Core Documentation. [Електронний доступ]. Режим доступу: <https://learn.microsoft.com/en-us/ef/core/>

11. Дуднік, О. І., & Хома, І. В. (2021). Розробка інформаційної системи бронювання паркувальних місць. Вісник Хмельницького національного університету. [Електронний доступ]. Режим доступу: <http://journals.khnu.km.ua>

12. Мобільний застосунок YASNO E-mobility. [Електронний доступ]. Режим доступу <https://yasno.com.ua/mobile-app>

13. Застосунок «Київ Цифровий » [Електронний доступ]. Режим доступу: <https://kyiv.digital/start>.

14. Український виробник зарядних станцій [Електронний доступ]. Режим доступу: <https://autoenterprise.ua/>

15. Реєстрація авто [Електронний доступ]. Режим доступу: https://opendatabot.ua/?utm_source=google&utm_medium=cpc&utm

16. Модель компоненти [Електронний доступ]. Режим доступу: <https://www.smartparking.com/uk>

17. Схема [Електронний доступ]. Режим доступу: <https://satinnovation.com>

18. Development of Smart Parking System using Internet of Things Concept [Електронний доступ]. Режим доступу: <https://www.researchgate.net/publication>

19. Комбінований ультразвуковий датчик. [Електронний доступ]. Режим доступу: https://www.google.com/url?sa=i&url=https%3A%2F%2Felhart.ru%2Fsensors_and_measuring_equipment%2Fultrasonic

20. Зарядна станція. [Електронний доступ]. Режим доступу: <https://energyup>