

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень


Серверна підсистема кіберфізичної системи «Розумна парковка»
Назва теми

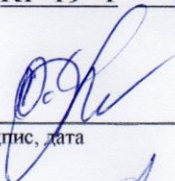
КвРКІ. 190128.19.01.24 ПЗ
Шифр

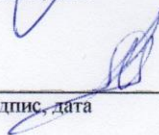
Галузь знань 12 «Інформаційні технології»
Шифр, назва

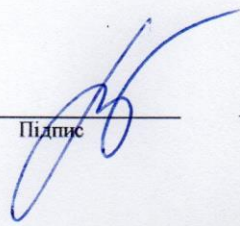
Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія»
Назва

Виконав: студент IV курсу, група КІ-19-1 
Підпис А. А. Кузьмін
Ініціали, прізвище

Керівник 
Підпис, дата О. О. Павлова
Ініціали, прізвище

Нормоконтролер 
Підпис, дата С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних систем 
Підпис Т.О. Говорущенко
Ініціали, прізвище

«1» червня 2023 р.

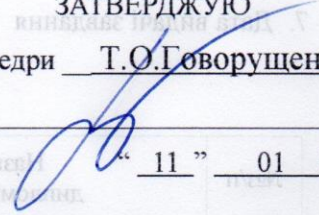
Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет <u>ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ</u>	
Кафедра <u>КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ</u>	
Освітній рівень <u>БАКАЛАВР</u>	
Галузь знань <u>12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ</u>	
Спеціальність <u>123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ</u>	
Освітня програма <u>ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»</u>	

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко



« 11 » 01 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Кузьмін Андрійю Андрійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Серверна підсистема кіберфізичної системи «Розумна парковка»
Керівник проекту (роботи) Павлова О.О., д.ф., ст.викл.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Огляд існуючих систем для розв'язання завдання

Обґрунтування вибору компонентів та середовища реалізації

Реалізація серверної підсистеми кіберфізичної системи «Розумна парковка»

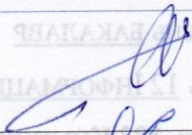
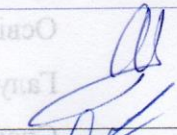
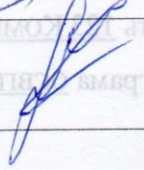
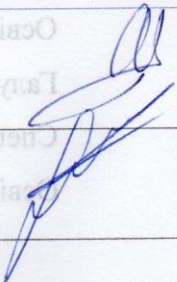
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Структурні схеми серверної підсистеми кіберфізичної системи «Розумна парковка»

Блок-схеми програми

Інтерфейси серверної підсистеми

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІС		
Антиплагиат	Нічпорок А.О., доцент кафедри КІС		

7. Дата видачі завдання « 11 » 01 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2023	виконано
3	Робота над розділом 1 – Огляд існуючих систем для розв'язання завдання	01.03.2023	виконано
4	Робота над розділом 2 – Обґрунтування вибору компонентів та середовища реалізації	01.04.2023	виконано
5	Робота над розділом 3 – Реалізація серверної підсистеми кіберфізичної системи “Розумна парковка”	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	31.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент


Підпис

А. А. Кузьмін

Ініціали, прізвище

Керівник проекту (роботи)


Підпис

О. О. Павлова

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Серверна підсистема кіберфізичної системи «Розумна парковка»».

Автор роботи: Кузьмін Андрій Андрійович.

Керівник роботи: Павлова Ольга Олександрівна.

Пояснювальна записка: 55 с., 29 рис., 4 табл., 4 дод., 65 джерел.

Графічна частина: 3 плакати.

РОЗУМНА ПАРКОВКА, КІБЕРФІЗИЧНА СИСТЕМА, СЕРВЕРНА ПІДСИСТЕМА, КАМЕРА ЗОВНІШНЬОГО СПОСТЕРЕЖЕННЯ, СТРУКТУРНА СХЕМА, АРХІТЕКТУРА КІБЕРФІЗИЧНОЇ СИСТЕМИ

Метою кваліфікаційної роботи є підвищення ефективності керування серверною підсистемою кіберфізичної системи «Розумна парковка».

Об'єктом дослідження є процес розробки серверної підсистеми кіберфізичної системи «Розумна парковка».

Предметом дослідження є серверна підсистема кіберфізичної системи «Розумна парковка».

Для досягнення поставленої мети використовуються такі методи дослідження, як методи синтезу, аналізу та моделювання процесів, принципи системного аналізу, теоретико-множинні підходи.



Підпис студента

26.05.2023р

Дата

ЗМІСТ

ВСТУП.....	4
1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ РОЗВ’ЯЗАННЯ ЗАВДАННЯ	6
1.1 Програмно-технічні засоби для розумних парковок в Україні та світі... 6	6
1.2 Порівняння існуючих програмно-технічних засобів.....	11
1.3 Аналіз користувацьких додатків для розумної парковки	15
1.4 Висновки	19
2 ОБҐРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ	20
2.1 Апаратне середовище реалізації.....	20
2.2 Функційні вимоги.....	23
2.3 Нефункційні вимоги.....	25
2.3.1 Вимоги до системи	26
2.3.2 Вимоги щодо взаємодії з навколишнім середовищем.....	26
2.4 Вибір методів та середовища для реалізації програмного забезпечення	
2.5 Висновки	38
3 РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНА ПАРКОВКА»	39
3.1 Принципи роботи серверної частини кіберфізичної системи «Розумна парковка»	39
3.2 Структурна схема та алгоритм роботи серверної частини кіберфізичної системи «Розумна парковка».....	47
3.3 Реалізація серверної підсистеми кіберфізичної системи «Розумна парковка»	53
3.4 Висновки	55
ВИСНОВКИ	57
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	59

					КВРКІ 190128.19.01.24 ПЗ					
Зм.	Арк.	№ док.ум.	Підпис	Дата	Серверна підсистема кіберфізичної системи «Розумна парковка»			Літера	Арк.вщ.	Арк.вщів
Виконав	Перевір.	Кузьмін А.А Павлова О.О.						у		
Н.контр.	Затвер.	Лисенко С.М. Говорущенко Т.О.			ХНУ КІ2-19-1					

ДОДАТОК А Копія креслення «Структурна схема серверної підсистеми кіберфізичної системи “Розумна парковка”»	66
ДОДАТОК Б Копія креслення «схема роботи компонентів кіберфізичної системи Розумна парковка».....	67
ДОДАТОК В Копія креслення «Архітектура серверної підсистеми кіберфізичної системи розумна парковка»	68
ДОДАТОК Г Лістинг коду програмного забезпечення серверної підсистеми кіберфізичної системи “Розумна парковка”	69

ВСТУП

Проблема вільних парковок часто зустрічається біля торгових центрів, магазинів, супермаркетів чи мінімаркетів, або ж біля житлових будинків чи ринків. Після проведення аналізу стає зрозуміло, що через брак паркомісць людям доводиться паркувати своє авто на обочині, газоні чи тротуарі, що в свою чергу заважає як водіям, так і пішоходам.

Наразі дана проблема існує у багатьох містах нашої країни. Через велику кількість авто та малу кількість парковок які видаляються на одне місто. Для вирішення проблеми запропоновано розгрузити паркомісця тим самим зменшити кількість “паркувальників” в непризначеному для них місці. Якщо ж водій бачитиме обладнанні спеціальні парковки, через додаток у своєму смартфоні, він зможе розуміти де поблизу є вільна парковка і йому не доведеться їздити по колу в пошуках вільного місця.

Особливу увагу можна приділити центральній частині міста. Майже завжди є проблема із паркуванням свого авто, і часто водіям доводиться паркуватись в незручних місцях або ж вуличках з яких потім важко виїжджати.

Також проблема часто спостерігається біля торгівельних центрів, якщо ж взяти до уваги великі торгові центри то там можна побачити велику парковку, щоправда в певний період часу водій починає кружляти по парковці в пошуках вільного місця, або ж стояти в очікуванні того, що хтось звільнить паркомісце. Це забирає чимало часу та може створити незручну чи навіть аварійну ситуацію.

Велика проблема із парковкою також спостерігається біля ринку, автівки залишають прямо на дорозі через брак місць. Через це може виникнути дискомфорт під час руху транспорту особливо під час великого потоку.

Актуальність роботи полягає у розробці серверної підсистеми для кіберфізичної системи розумної парковки. Метою дипломної роботи є підвищення ефективності керування серверною підсистемою кіберфізичної системи “Розумна парковка”.

					КвРКІ 190128.19.01.24 ПЗ	Арк. 4
Зм.	Арк.	№ докum.	Підпис	Дата		

Поставлена мета досягається розв'язанням таких основних задач:

- 1) провести огляд існуючих рішень та систем для розумних парковок;
- 2) виконати вибір компонентів та середовища для реалізації задачі;
- 3) розробити серверну підсистему кіберфізичної системи “Розумна парковка”.

Об'єктом дослідження є процес розробки серверної підсистеми кіберфізичної системи “Розумна парковка”. Предметом дослідження є серверна підсистема кіберфізичної системи “Розумна парковка”.

Практична цінність отриманих результатів полягає у розробці серверної підсистеми кіберфізичної системи для розумної парковки на базі алгоритмів із застосуванням штучних нейронних мереж та машинного зору для розпізнавання зображень.

За темою дипломної роботи було взято участь у Всеукраїнській науково-практичній конференції “Актуальні Проблеми Комп'ютерних Наук (АПКН-2022)”, м. Хмельницький, Всеукраїнській науково-практичній конференції Інформаційні технології та інженерія (IT&I-2023), м. Миколаїв та опубліковано тези у збірниках конференцій.

					КвРКІ 190128.19.01.24 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ РОЗВ'ЯЗАННЯ ЗАВДАННЯ

1.1 Програмно-технічні засоби для розумних парковок в Україні та світі

Ідея стосовно реалізації системи “розумна парковка” виникла вже давно. Її суть полягає в різносторонньому підході, адже можна пов'язувати такі технології як магнітні чи інфрачервоні датчики або ж камери.

До прикладу реалізації технологій “розумної парковки”, у місті Київ запустили першу парковку із безготівковим розрахунком [1], але для такої реалізації потрібні інспектори, які перевірятимуть чи була здійснена безготівкова оплата за місце паркування чи ні.

У Львові здійснено інший підхід, цю технологію реалізовано за допомогою датчиків [2], а завдяки додатку можна відслідкувати кількість вільних місць на парковці, а також зарезервувати одне із місць наперед.

Принцип реалізації такої технології здійснюється за допомогою магнітних та інфрачервоних датчиків або ж камер.

Магнітний датчик являє собою систему, яка працює за допомогою детекторів, що вимірюють зміни магнітного поля навколо свого радіусу дії. Монтуються в асфальтове дорожнє покриття на парковці. Коли водій припаркує свою автівку на одному із вільних місць, виникне порушення магнітного поля, що в свою чергу призведе до спрацювання датчику [3].

При такій реалізації з часом виникнуть проблеми, при заміні дорожнього покриття потрібно постійно демонтувати датчики, щоб їх не пошкодити.

Принцип роботи магнітного датчика показано на рисунку 1.1.

Також використовують інфрачервоні датчики, їхній алгоритм роботи пов'язаний на інфрачервоних сенсорах, які в свою чергу реагують на тепло [4].

Досить не практично використовувати такі датчики при проектуванні розумної парковки, адже інфрачервоні сенсори зчитують тепло, а саме інформацію з усіх тіл, що в свою чергу призведе до некоректної роботи програмного забезпечення.

					КВРКІ 190128.19.01.24 ПЗ	Арк. 6
Зм.	Арк.	№ докum.	Підпис	Дата		

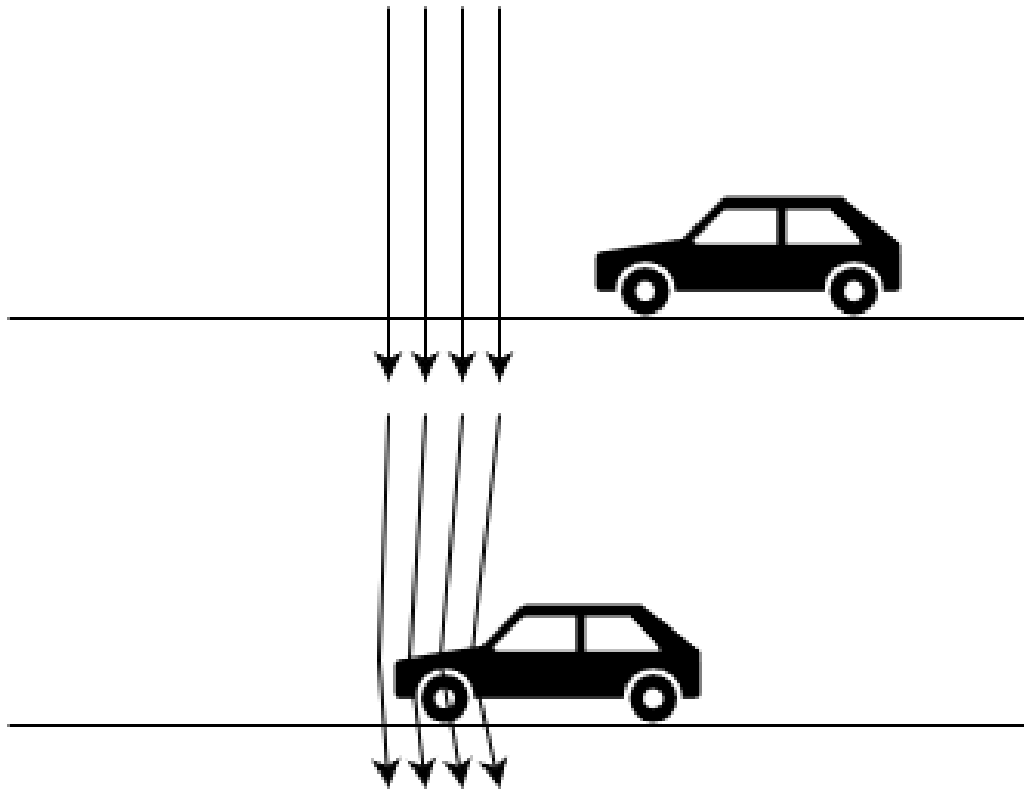


Рисунок 1.1 – Принцип роботи магнітного датчика

До переваг такого рішення відносяться:

1. Точне регулювання відстані.
2. Відсутність шкідливого випромінювання.
3. Відносно низька ціна.

До недоліків відносяться:

1. Помилкові спрацювання;
2. Погана робота при високих температурах;
3. Не спрацьовує через скляну поверхню.

Принцип роботи інфрачервоного датчика зображено на рисунку 1.2.

Також використовуються при проектуванні парковок не тільки датчики, які час від часу не практичні в різних умовах таких як, погодні умови, високі і низькі температури.

Камера - універсальна технологія, що в свою чергу використовується у повсякденному житті. В кожному телефоні чи ноутбуці вже одразу вбудована камера.

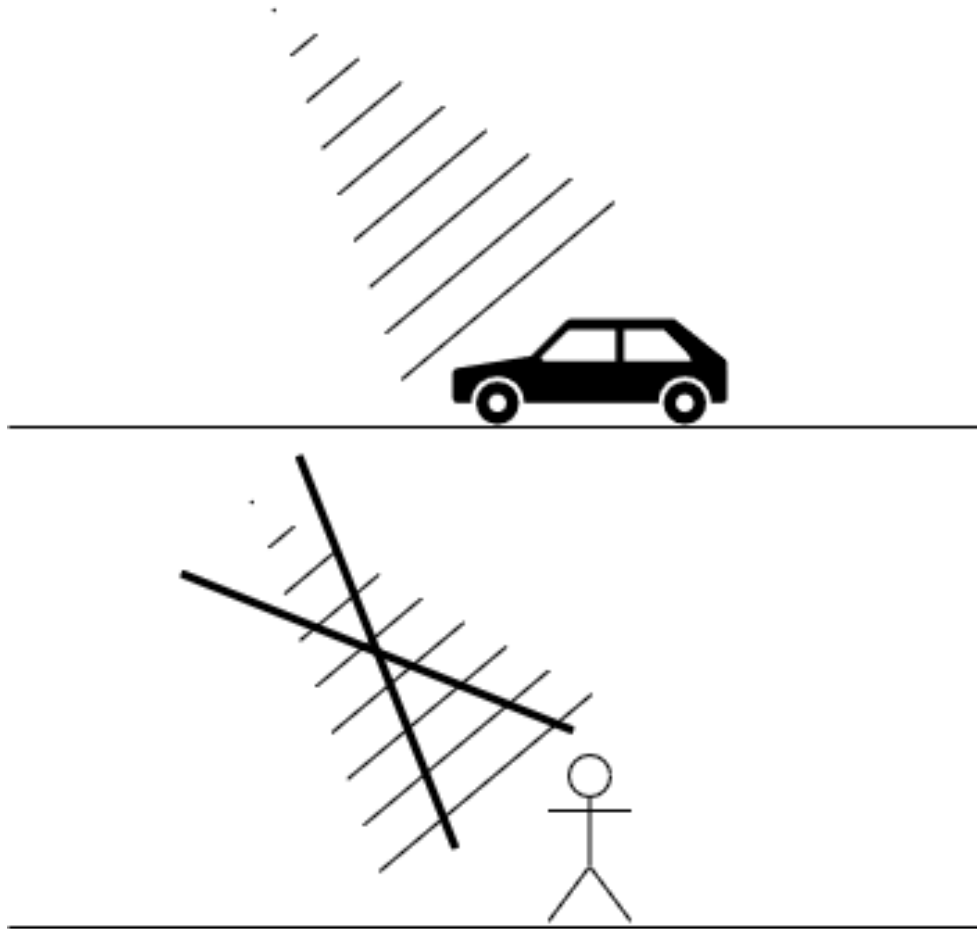


Рисунок 1.2 – Принцип роботи інфрачервоного датчика

Підприємці використовують камери відеоспостереження, щоб вберегти бізнес та товари від зловмисників. Існують камери відеофіксації, які можна зустріти на швидкісних шосе. Вони слугують для того, щоб фіксувати порушення допустимої швидкості на тому чи іншому відрізку шосе. У випадку фіксації порушення, камера передає дані автівки порушника до поліції.

В даному випадку використано алгоритм відеоспостереження, тобто зображення з відеокамери передається на екран мобільного пристрою за допомогою додатку.

Принцип реалізації за допомогою камери показано на рисунку 1.3.

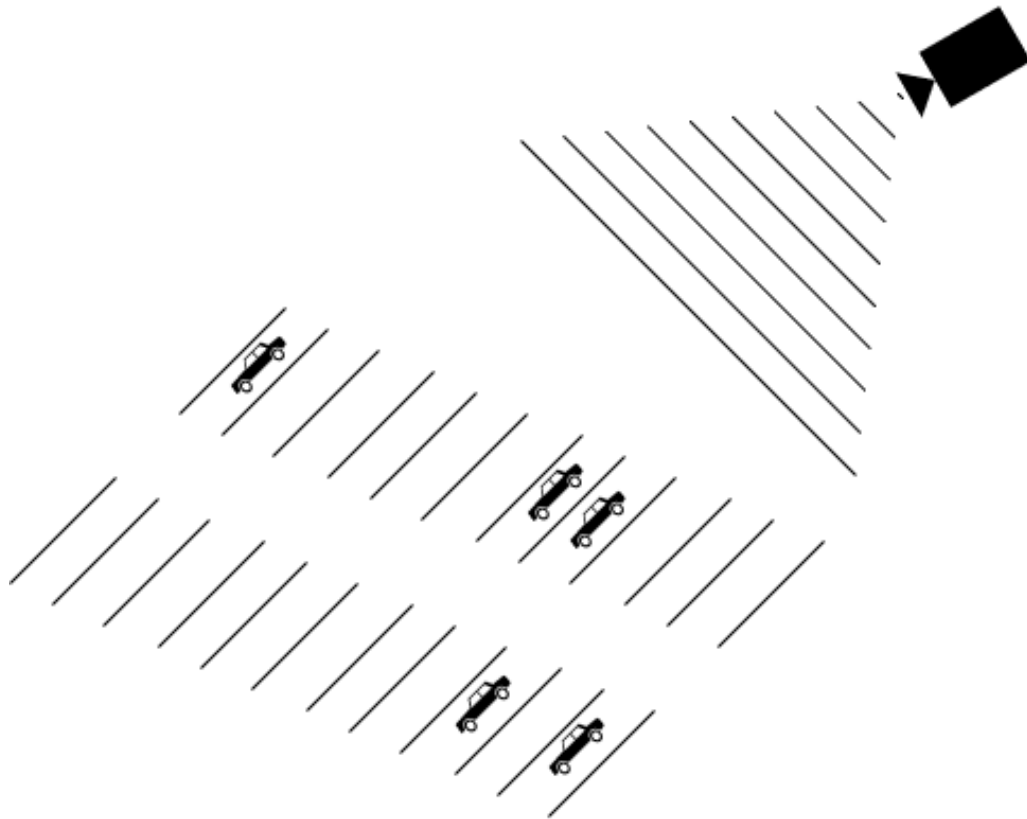


Рисунок 1.3 – Принцип реалізації за допомогою камери

Системи відеоспостереження бувають:

1. Простої структури.
2. Складної структури.

У першому випадку зображення із однієї камери автоматично передається на екран через додаток.

У другому випадку зображення із декількох камер може записуватись для подальшого вивчення, за цим може слідкувати оператор.

У випадку використання складної структури доведеться наймати додатково оператора для моніторингу камер відеоспостереження, що не зовсім зручно та практично, а також це займатиме багато часу на відміну від простої структури.

Систему відеоспостереження простої структури показано на рисунку 1.4.

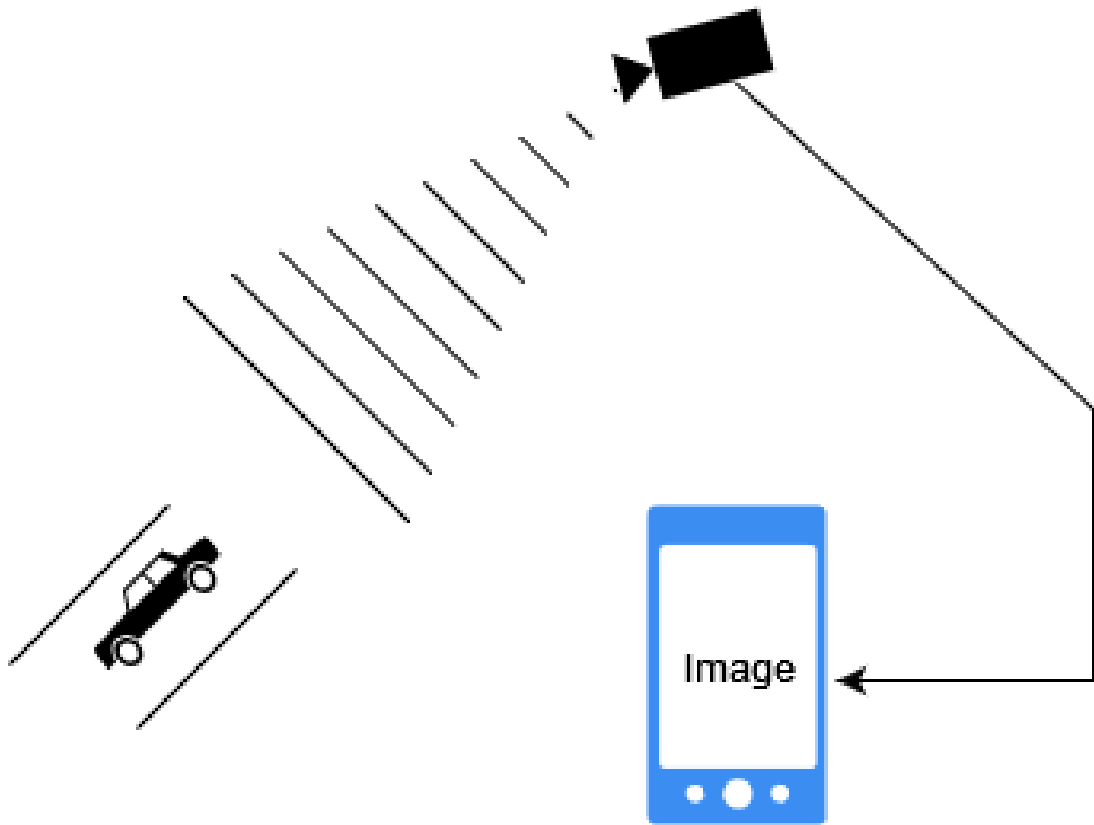


Рисунок 1.4 – Система відеоспостереження простої структури

Систему відеоспостереження складної структури показано на рисунку 1.5.

Використання алгоритму відеоспостереження простої структури для передачі інформації в режимі прямої трансляції на екран мобільного пристрою за допомогою додатку - найкращий варіант. Оскільки, це не займає багато часу, та не потребує додаткового персоналу. Зображення автоматично передається користувачу одразу із інформацією про вільну кількість паркувальних місць.

Хоча датчики широко використовуються для виявлення транспортних засобів, вони мають деякі недоліки. Один з основним недоліком є масштабованість, оскільки необхідність використання одного датчика на місце для паркування може бути дорогим рішенням. Щоб подолати цю проблему, можна використовувати системи на основі зору для виявлення транспортних засобів. Системи такого типу можуть забезпечити більш масштабоване рішення, ніж система на основі датчиків. А камеру, яка розміщена в місці, яке охоплює широкий огляд зони паркування визначити стан кількох паркувальних місць.

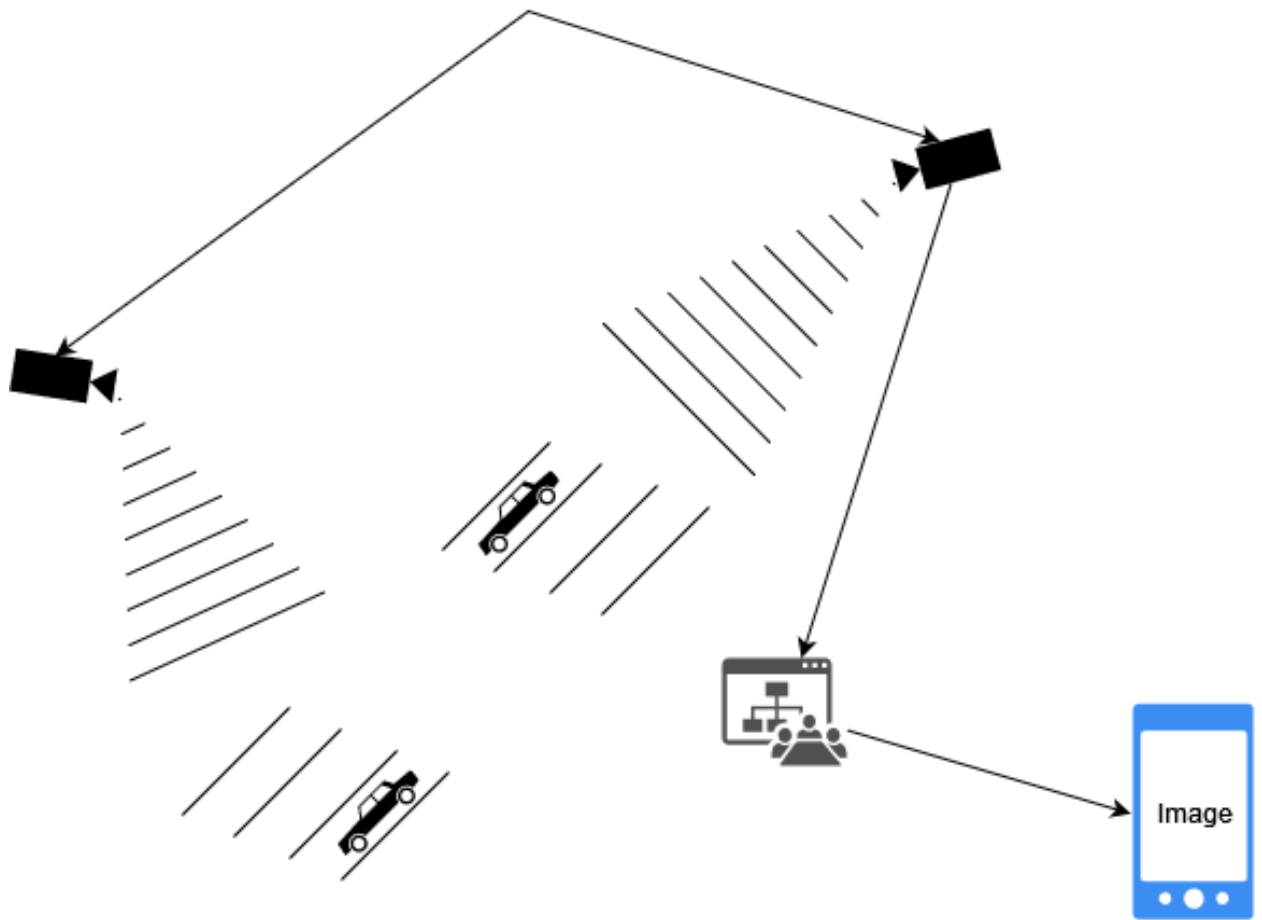


Рисунок 1.5 - Система відеоспостереження складної структури

1.2 Порівняння існуючих програмно-технічних засобів

Існує достатньо методів для реалізації подібної технології. Але кожен із цих методів несе в собі як переваги, так і недоліки. Якщо проаналізувати кожен з технологій, а саме - магнітний датчик, інфрачервоний датчик, та камера тоді буде зрозуміло, що найкраще може справитись з поставленою задачею.

На наступній діаграмі, яку зображено на рисунку 1.6 зображено графік по кількості використання технологій.

Надсилення зображень на сервер або хмарний сервіс через Інтернет також супроводжується проблемами безпеки. Щоб вирішити цю проблему безпеки, можна використовувати певний процес шифрування. Усі ці обмеження можуть негативно вплинути на зручність використання системи.



Рисунок 1.6 – Співвідношення використання технологій для розумних парковок

На даному графіку зрозуміло, що технологія камер або ж камер відеоспостереження використовується найчастіше, а саме підприємцями та державними службами щоб вберегти майно від порушників.

Магнітний та інфрачервоний датчики використовують не так часто адже вони досить не практичні. Магнітний датчик легко пошкодити хоча незважаючи на це, він зміг би чудово підійти для реалізації технології для розумної парковки. Інфрачервоний датчик менш практичний та використовується не так часто. Він більш практичний для домашнього використання.

Переваги та недоліки присутні в кожній із наведених технологій адже працюють вони за різних умов, та по різному принципу. Щоб побудувати динамічну розумну систему паркування, місця для паркування мають визначатися автоматично. Це завдання можна виконати за допомогою алгоритму виявлення об'єктів. На рисунку 1.7 показано алгоритм роботи магнітного датчика.



Рисунок 1.7 – Алгоритм роботи магнітного датчика

Магнітний датчик досить практична технологія для подібного проекту, та в ситуації, якщо один автомобіль займе два паркувальних місця, спрацює два датчика, адже буде порушено радіус дії.

Інфрачервоний датчик не практичний для подібного проекту, так як часто на нього впливають зовнішні фактори. Метою передачі даних в системі є передача даних на сервер, де можна зареєструвати інформацію про паркування. Передані дані мають на меті надати інформацію про стан паркінгу, тобто про те, скільки паркувальних місць доступно і скільки з цих місць для паркування незайнятих. На рисунку 1.8 зображено алгоритм роботи інфрачервоного датчика.



Рисунок 1.8 – Алгоритм роботи інфрачервоного датчика

Вплинути на роботу цього датчика може зовнішнє середовище, погодні умови або ж люди які просто проходять повз випадково потрапляючи у зону спрацювання інфрачервоного датчика.

Камера досить практична технологія, яка використовується постійно та регулярно. Завдяки цій технології не тільки знімаються фільми або ж моменти із власного життя, якими в результаті діляться у соцмережах. А також завдяки камерам виконується відеоспостереження, що в свою чергу передає зображення на пристрій. На рисунку 1.9 зображено діаграму роботи камери.

Ця технологія досить часто використовується. Але також має свої недоліки. Через нестабільне підключення може збоїти зображення та якість. Також до недоліків відноситься висока ціна такого обладнання.



Рисунок 1.9 – Алгоритм роботи методу на основі камери зовнішнього спостереження

1.3 Аналіз користувацьких додатків для розумної парковки

У ході роботи було проведено аналіз існуючих рішень у вигляді мобільних додатків для операційних систем Android та iOS, які доступні на ринку, а саме в популярних магазинах мобільних додатків, таких як Play Market та Apple App Store [16,17] (Таблиця 1.1).

Таблиця 1.1 – Аналіз існуючих програмних рішень для розумного паркування у формі мобільних застосунків

Назва додатку	Платний чи безкоштовний	Операційна система	Опис додатку	Країна та міста поширення
Smart Parking	Додаток безкоштовний, але послуга паркування платна	Android та iOS	Додаток Smart Parking допоможе знайти місце для паркування, використовуючи актуальну інформацію від системи датчиків Smart Parking, яка встановлена на паркувальних місцях по всьому місту. Використовуючи прості червоні, жовті та зелені маркери на карті додатка в реальному часі, додаток Smart Parking надає точну інформацію про те, де ви, швидше за все, зможете знайти вільне місце.	Австралія, Уельс, Майамі (США)

Зм.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 1.1 – Аналіз існуючих програмних рішень для розумного паркування у формі мобільних застосунків

Назва додатку	Платний чи безкоштовний	Операційна система	Опис додатку	Країна та міста поширення
Penang Smart Parking	Додаток безкоштовний, але послуга паркування платна	Android та iOS	Penang Smart Parking — це платформа для паркування та оплати в Пенангу. Він оновлює звичайний спосіб паркування за допомогою паркувального талона, автоматично усуває клопоти, пов'язані з пошуком кабінки з паркувальними купонами та подряпанням вашого купону. Ви можете знайти паркування навколо вибраних зон, легко поповнити свій гаманець і миттєво оплатити паркування та виклик.	Пенанг (Малайзія)

Зм.	Арк.	№ докum.	Підпис	Дата

КВРКІ 190128.19.01.24 ПЗ

Арк.
17

Отже, з таблиці 1.1 видно, що наразі є досить велика кількість мобільних додатків для розумних парковок по всьому світу. Проте, на жаль, таких додатків немає в Україні, хоча потреба у них, особливо у великих містах-мегаполісах та навіть обласних центрах, є величезною. Тому створення серверної частини, яка в подальшому буде в основі клієнт-серверної архітектури для системи розумної парковки і є метою даної роботи.

1.4 Висновки

У першому розділі проведено аналіз проблеми пов'язаної із паркуванням автомобілів у рамках концепції Розумних міст. Запропоновано методи вирішення даної проблеми за допомогою кіберфізичної системи “Розумна парковка”.

Також було проведено дослідження та опис методів реалізації та проаналізовано позитивні чи негативні сторони тих чи інших технологій. Також проведено детальний аналіз подібних технологій які використовують в інших країнах.

Крім того, наведено приклади вже існуючих обладнаних розумних парковок та клієнтського програмного забезпечення у вигляді мобільних додатків для операційних систем Android та iOS.

					КвРКІ 190128.19.01.24 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ОБҐРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ

2.1 Апаратне середовище реалізації

Для реалізації “розумної парковки” обрано метод використання камер зовнішнього відеоспостереження, оскільки даний спосіб є найменш затратним з точки зору кількості обладнання, портативності та економічності. Приклад системи для розумного паркування на основі камер зображено на рисунку 2.1.

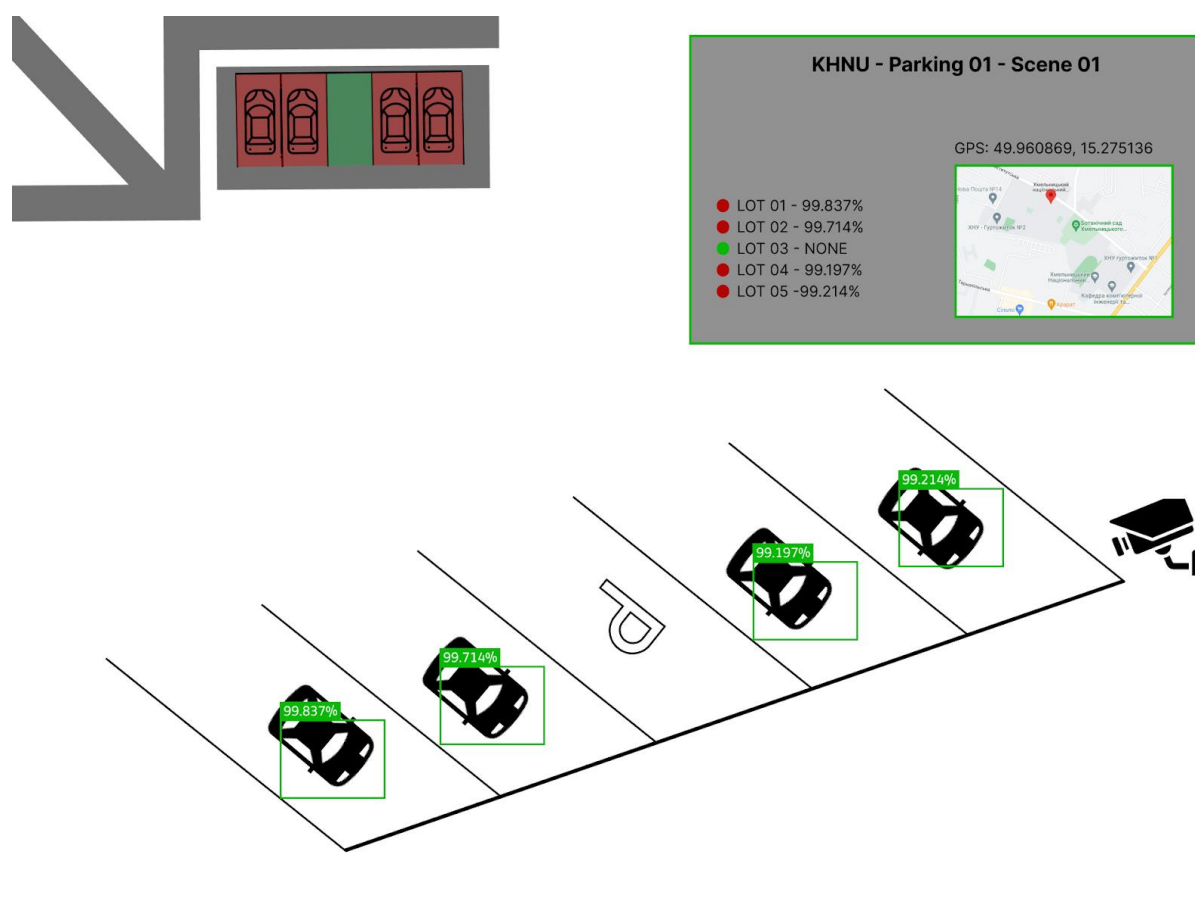


Рисунок 2.1 – Кіберфізична система для розумної парковки на основі використання камер зовнішнього спостереження

У розділі 1.3 та у [8] описано порівняння існуючих методів для розумного паркування, а саме на основі ультразвукових датчиків, інфрачервоних датчиків,

магнітних датчиків та на основі камер. Для того щоб переконатися у тому, що метод з використанням камер зовнішнього спостереження є доцільним, було визначено критерії, за якими було здійснено вибір апаратного середовища. Це вартість монтажу, вартість обслуговування, вартість обладнання, енергозатратність, портативність (можливість транспортування компонентів), масштабованість, надійність, розпізнавання транспортного засобу, розпізнавання перешкод, апаратна складність та складність програмного забезпечення та ідентифікація транспортних засобів. Щоб переконатися наскільки спосіб розумного паркування на основі камер зовнішнього спостереження задовольняє перелічені вище критерії, представимо це у вигляді таблиці 2.1.

Таблиця 2.1 – Відповідність способу розумної парковки на основі використання камер зовнішнього спостереження поставленим вимогам

Критерій оцінки	Показник
Вартість монтажу	Низька
Вартість обслуговування	Низька
Вартість обладнання (за одиниць)	Середня
Енергозатратність	Висока
Портативність	Легко

Кінець таблиці 2.1 – Відповідність способу розумної парковки на основі використання камер зовнішнього спостереження поставленим вимогам

Критерій оцінки	Показник
Складність програмного забезпечення	Висока
Масштабованість	Висока
Надійність	Висока
Ідентифікація	Так
Розпізнавання перешкод	Так
Розпізнавання транспортних	Так
Апаратна складність	Низька
Додаткові вимоги	Конфіденційність і безпека

У таблиці 2.1 різними кольорами було виділено ступінь задоволення критеріїв оцінки ефективності розумною парковкою із використанням камер зовнішнього спостереження як засобу розпізнавання зображень. Показники відповідають за позитивну, негативну та нейтральну оцінку критерію. Оскільки, із 12 критеріїв 9 було оцінено позитивно, то метод використання камер зовнішнього

спостереження для розпізнавання зображень автомобілів для розумної парковки є доцільним.

На рисунку 1.4 у розділі 1 зображена схема роботи розумної парковки, заснованої на роботі камер зовнішнього спостереження. Однак, для обладнання кіберфізичної системи розумної парковки потрібно ще багато деталей, а саме:

1. PoE-коммутатор TP-LINK TL-SG1005LP.
2. IP відеокамера Dahua DH-IPC-HDW2431TP-AS-S2.
3. Витя пара зовнішньої прокладки UTP с.5Е 4 х 2 х 0.51 мм² 25 м.
4. Кабель КППЕт-ВП (100) 4*2*0,51 (FTP-cat.5Е), ОК-net, СU, ізоляція ПЕ, екр., з трос. 7 * 0,5 для нар. роб.,305м.
5. 2х Спліттер PoE для CCTV камер Ethernet 5.5х2.1мм.
6. Конектор Cor-X RJ45 connector CAT5Е 8P8C 10 шт.
7. Інжектор PoE-INJECTOR Lite.

2.2 Функційні вимоги

Вимоги до програмного забезпечення - це набір вимог щодо властивостей, якості та функцій програмного забезпечення, яке буде розроблено, або знаходиться у розробці. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки вимог [18].

Функціональні вимоги (Functional Requirements) - це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: обчислення даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система [19], тобто іншими словами функціональні вимоги визначають що саме повинен робити програмний продукт і за що відповідають окремі його частини. Ці вимоги описуються в документі Специфікація вимог до програмного забезпечення.

					КВРКІ 190128.19.01.24 ПЗ	Арк. 23
Зм.	Арк.	№ докum.	Підпис	Дата		

На відміну від нефункціональних вимог, які визначають якою система повинна бути, функціональні вимоги визначають, що система повинна робити. Функціональні вимоги до програмного забезпечення визначаються на першій стадії процесу розробки ПЗ - на етапі аналізу вимог[19].

Оскільки метою даної роботи є проектування серверної частини для кіберфізичної системи “Розумна парковка”, розглянемо функціонал, який покладається на серверну частину.

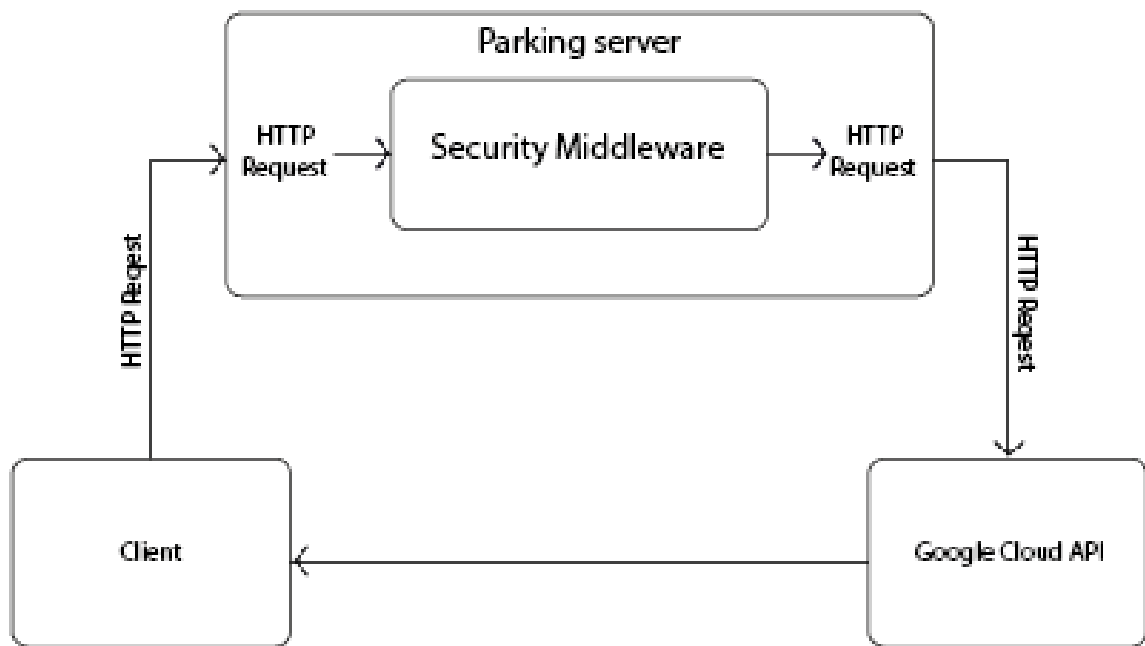


Рисунок 2. 2 – Архітектура програмного забезпечення кіберфізичної системи розумної парковки

З рисунку 2.2 видно, що програмна частина кіберфізичної системи розумної парковки складається з клієнтської частини (мобільного додатку), серверної частини та зовнішніх програм, які будуть взаємодіяти з системою через API - прикладний програмний інтерфейс. Серверна частина програмного забезпечення кіберфізичної системи розумної парковки складається з проміжного програмного забезпечення для перевірки безпеки HTTP-запитів, програми-обробника запитів та самого алгоритму на основі штучного інтелекту для розпізнавання зображень з камери. Наразі для тестування роботи програми будемо використовувати вже

готовий алгоритм розпізнавання зображень, а саме бібліотеку Google Cloud Vision, яка буде взаємодіяти з нашим серверним програмним забезпеченням через API.

Відтак, основні функціональні вимоги, які можна сформулювати для серверної частини ПЗ:

1. Наявність функціоналу розпізнавання зображення з камери зовнішнього спостереження.
2. Наявність алгоритму перевірки безпечності запитів від програми-клієнта.
3. Можливість спостереження за автомобілем в режимі реального часу (підключення до камери зовнішнього спостереження).
4. Надання висновку - паркомісце вільне чи зайняте.

2.3 Нефункційні вимоги

Нефункціональні вимоги (Non-Functional Requirements) — це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути. Нефункціональні вимоги до програмного забезпечення визначаються на ранніх етапах життєвого циклу ПЗ, а саме на етапі аналізу вимог до програмної системи, що проектується [20].

Нефункціональні вимоги можна поділити на дві категорії:

- Спрямовані на покращення системи (безпека, надійність, швидкодія, зручність у використанні);
- Спрямовані на вдосконалення (масштабування, відновлюваність) властивостей системи.

					КвРКІ 190128.19.01.24 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

2.3.1 Вимоги до системи

Оскільки в даній роботі ми не розглядаємо клієнтську частину системи, тому не враховуємо вимоги до інтерфейсу користувача. Однак, при проектуванні серверної частини не варто відкидати вимогу щодо зручності програмного інтерфейсу для взаємодії з клієнтською частиною та базою даних та комунікації через HTTP-запит.

Апаратні та програмні вимоги, тобто опис апаратної та програмної платформ, необхідних для роботи (і підтримки) системи. У нашому випадку це технічні характеристики камер зовнішнього спостереження (висока роздільна здатність, можливість запису відео у темноті, захист від несприятливих погодніх умов), постійний доступ до мережі Інтернет та безперебійне живлення для безперервного доступу до сервера, операційна система сервера - Ubuntu, наявність бази даних.

Вимоги щодо безпеки та конфіденційності серверної частини також відіграють важливу роль, так як несанкціонований доступ до сервера може призвести до помилки роботи алгоритму та неправильності результатів програми, а саме некоректного розпізнавання зображення та неправильних висновків щодо зайнятості чи не зайнятості паркомісця, витоку даних чи збоїв у роботі системи в цілому. Також при проектуванні серверної частини важливо дотримуватись вимог щодо надійності, тобто щоб система працювала без збоїв 24/7. А при можливому збої роботу системи та дані можна було легко відновити, тобто вимога відновлюваності. Не мало важливою є і вимога щодо продуктивності та швидкодії роботи серверної системи, адже від цього залежить швидкість отримання результату користувачем та репутація самої системи.

2.3.2 Вимоги щодо взаємодії з навколишнім середовищем

Оскільки система розумної парковки на основі камер зовнішнього спостереження може бути використана як для громадського користування, так і для комерційних організацій, важливими нефункційними вимогами також є її

					КвРКІ 190128.19.01.24 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

екологічність та безпека взаємодії з навколишнім середовищем. Так як для роботи серверної частини нам потрібні лише камери зовнішнього спостереження, якими часто вже оснащені паркомайданчики поблизу великих торгівельних центрів, немає потреби у встановленні додаткового обладнання, такого як магнітні чи інфрачервоні давачі. Відтак, немає потреби у проведенні додаткових будівельно-монтажних робіт, що може пошкодити тротуарну плитку чи покриття дороги або паркомісця. Для роботи алгоритму розпізнавання паркомісця на базі штучного інтелекту також потрібна стандартна паркувальна розмітка. Такою розміткою, як і камерами зовнішнього спостереження, часто вже оснащені паркомайданчики комерційних установ та закладів. В цьому і полягає перевага щодо екологічності та дружності до навколишнього середовища методу розумних парковок на основі камер зовнішнього спостереження, а також його економічний ефект, у порівнянні з іншими відомими методами. Варто також зауважити і перевагу щодо безпеки такого виду розумної парковки. Завдяки камерам зовнішнього спостереження та функційній вимозі щодо можливості спостереження за автомобілем в режимі реального часу (підключення до камери зовнішнього спостереження), користувач через клієнтську частину системи - мобільний додаток зможе спостерігати за автомобілем, наприклад, роблячи покупки у торговельному центрі або у кінозалі чи перебуваючи в університеті.

2.4 Вибір методів та середовища для реалізації програмного забезпечення

Для реалізації серверної підсистеми кіберфізичної системи розумної парковки з метою тестування роботи алгоритмів розпізнавання зображень з камери зовнішнього спостереження було обрано такий стек технологій:

1. Мова програмування Python та фреймворк Flask для роботи з бібліотекою OpenCV.
2. Бібліотека комп'ютерного зору OpenCV для розпізнавання зображень з камери.

					КвРКІ 190128.19.01.24 ПЗ	Арк. 27
Зм.	Арк.	№ докum.	Підпис	Дата		

3. Мова програмування PHP та фреймворк Laravel для роботи з REST API.

Для розробки серверної частини будуть використовуватися середовища PHP Storm та PyCharm IDE.

OpenCV було започатковано в Intel у 1999 році Гарі Брадскі, а перший випуск вийшов у 2000 році. У 2005 році OpenCV використовувався на автомобілі Stanley, який виграв у 2005 році DARPA Grand Challenge. Пізніше його активний розвиток продовжився за підтримки Willow Garage під керівництвом Гарі Брадскі та Вадима Писаревського. Тепер OpenCV підтримує безліч алгоритмів, пов'язаних із комп'ютерним зором і машинним навчанням, і з кожним днем розширюється.

Бібліотека OpenCV підтримує широкий спектр мов програмування, таких як C++, Python, Java тощо, і доступний на різних платформах, включаючи Windows, Linux, OS X, Android та iOS. Також активно розробляються інтерфейси для високошвидкісних операцій GPU на основі CUDA і OpenCL. Проте для даної роботи було обрано OpenCV-Python як один з найпростіших та найлегших у користуванні варіантів. Це API Python для OpenCV, що поєднує найкращі якості OpenCV C++ API і мови Python.

OpenCV-Python - це бібліотека прив'язок Python, розроблена для вирішення проблем комп'ютерного зору. Python - це мова програмування загального призначення, започаткована Гвідо ван Россумом, яка дуже швидко стала дуже популярною, головним чином через її простоту та читабельність коду. Це дозволяє програмісту висловлювати ідеї в меншій кількості рядків коду, не зменшуючи читабельності.

Порівняно з такими мовами, як C/C++, Python повільніший. Тим не менш, Python можна легко розширити за допомогою C/C++, що дозволяє нам писати код з інтенсивними обчисленнями на C/C++ і створювати оболонки Python, які можна використовувати як модулі Python. Це дає нам дві переваги: по-перше, код такий же швидкий, як оригінальний код C/C++ (оскільки це фактичний код C++, що працює у фоновому режимі), а по-друге, його легше кодувати на Python, ніж на C/C++. OpenCV-Python — це оболонка Python для оригінальної реалізації OpenCV

					КВРКІ 190128.19.01.24 ПЗ	Арк. 28
Зм.	Арк.	№ докum.	Підпис	Дата		

C++. OpenCV-Python використовує NumPy, яка є оптимізованою бібліотекою для числових операцій із синтаксисом у стилі MATLAB. Усі структури масивів OpenCV перетворюються в масиви NumPy і з них. Це також полегшує інтеграцію з іншими бібліотеками, які використовують NumPy, такими як SciPy і Matplotlib [38].

Також для роботи з веб-частиною буде застосовано фреймворк Flask це веб-фреймворк Python, створений з невеликим ядром і легкою для розширення філософією. Початківцям легко почати працювати з Flask, оскільки існує невеликий шаблонний код для запуску та запуску простої програми.

API є аббревіатурою від Application Programming Interface, що означає програмний інтерфейс програми. API представляє набір правил і функцій, що дозволяють двом різним програмам взаємодіяти один з одним. Подібні інтерфейси полегшують інтеграцію додатків, даючи можливість розробникам створювати потужні діджитал рішення. API є посередником між додатками, надсилаючи запити та відповіді [39]. Для надсилання та отримання тих самих запитів (функцій) та залежно від призначення API слід різним протоколам та стандартам. Розрізняють такі типи протоколів та архітектур:

1. XML-RPC — дозволяє обмінюватись функціями між двома або більше мережами. XML-RPC використовує XML для опису запитів та відповідей, і за допомогою протоколів HTTP передає інформацію від клієнта до сервера.

2. JSON-RPC - це полегшений RPC, схожий на XML. Протокол закодований в JSON, що дозволяє отримувати виклики на сервер з можливістю асинхронних відповідей.

3. SOAP - simple object access protocol - протокол для обміну структурованою інформацією під час реалізації веб-сервісів у комп'ютерних мережах. SOAP використовує XML для аутентифікації, авторизації та взаємодії процесів в операційних системах та дозволяє клієнтам викликати веб-сервіси та отримувати відповіді незалежно від мови та платформи.

4. REST API – репрезентативна передача стану (REST) – це архітектурний стиль, який здійснює реалізацію клієнта та сервера незалежно один від одного.

Сервіси REST API взаємодіють за протоколом HTTP. Принцип роботи API зображено на рисунку 2.3.

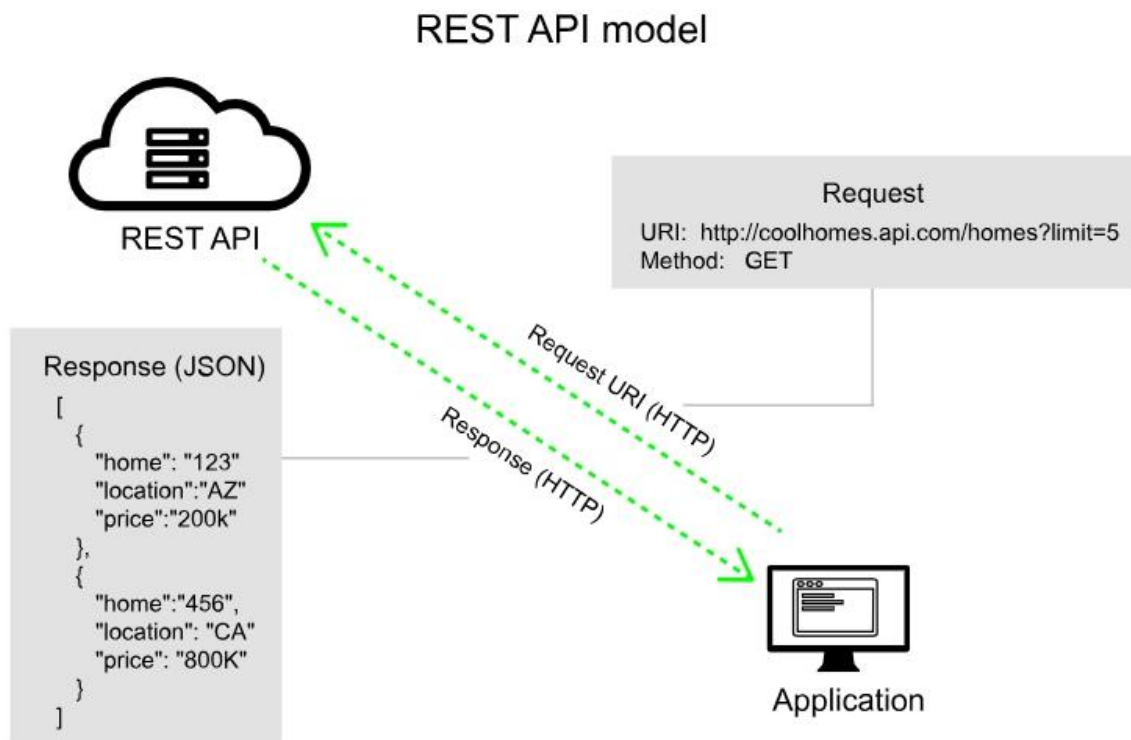


Рисунок 2.3 – Принцип роботи REST API

REST або Representational state transfer, що перекладається як передача стану подання - це архітектурний стиль проектування API з використанням протоколу HTTP. Головна перевага REST - велика гнучкість. REST API застосовується всюди, де є необхідність надання даних із сервера користувачеві веб-програми або сайту.

Головними компонентами REST API є:

- Client - клієнт або програма, яка запущена на стороні користувача (на його девайсі) та ініціює комунікацію.
- Server - сервер, який надає API як доступ до своїх даних та функцій.
- Resource - ресурс є будь-яким видом контенту (відео, текст, картинка), який сервер може передати клієнту.

REST API взаємодіє з допомогою HTTP запитів, виконуючи стандартні функції: створення, оновлення, читання, видалення записів у ресурсі. Існує чотири методи, що описують, що потрібно робити з ресурсом:

- POST - створення ресурсу.
- GET – отримання ресурсу.
- PUT – оновлення ресурсу.
- DELETE – видалення ресурсу.

Методи роботи з REST API зображені на рисунку 2.4.

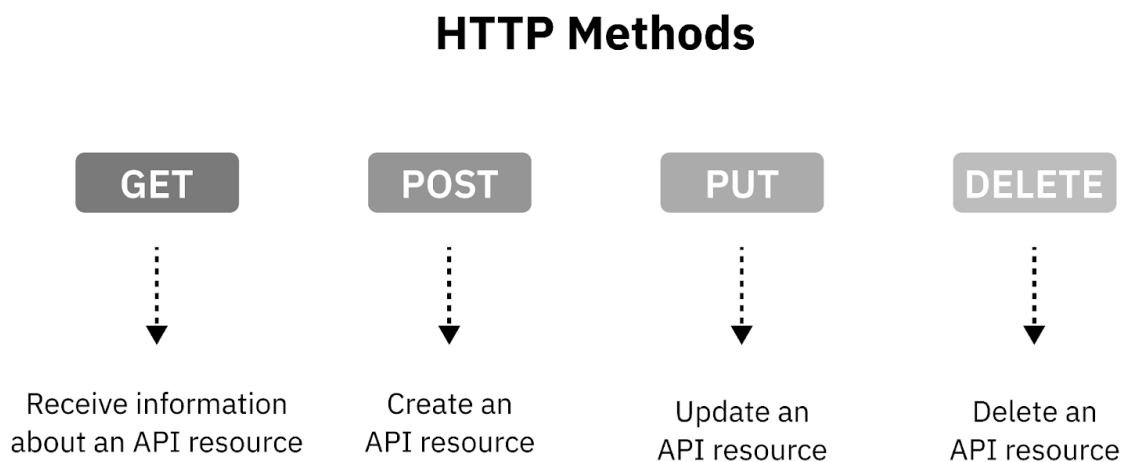


Рисунок 2.4 – Методи роботи з REST API

Ресурс – це ключова абстракція інформації. Будь-який вид інформації, який ми можемо використовувати, може бути ресурсом: документ, зображення, сервіс. Стан ресурсу будь-який певний момент називається уявленням ресурсу. Воно складається з:

- даних;
- метаданих, що описують дані;
- гіпермедіа посилань, які допомагають клієнтам перейти до наступного стану.

Ця інформація доставляється клієнту у будь-якому форматі: JavaScript (JSON), HTML, XLT, Python, PHP або простий текст.

JSON найбільш популярний, тому що він читабельний для машин та людей і не залежить від мови програмування. Щоб отримати доступ до ресурсу, клієнт надсилає запит. Сервер генерує відповідь із закодованими даними про ресурс.

Структура будь-якого запиту включає чотири основні компоненти: HTTP метод, ендпоінти, заголовки і тіло.

HTTP метод визначає, що має бути зроблено з ресурсом.

Ендпоінт містить URI - Uniform Resource Identifier (уніфікований ідентифікатор ресурсу), який вказує, де та як знайти ресурс в Інтернеті. Найпоширеніший тип URI — це URL або Uniform Resource Location, яка є повноцінною веб-адресою.

Заголовки зберігають інформацію, що стосується як клієнта, так і сервера. Головним чином, заголовки надають автентифікаційні дані: API ключ, назву або IP-адресу комп'ютера, на якому встановлений сервер, а також інформацію про формат відповіді.

Тіло необхідно для передачі серверу додаткової інформації: дані тіла body - це дані, які, наприклад, потрібно додати або замінити.

REST не прив'язаний до будь-якої конкретної технології або платформи, він не залежить від мови. Він також не вказує, як саме створювати API, але використовує 6 архітектурних обмежень, які мають бути дотримані, щоб інтерфейс можна було назвати RESTful. Саме ці принципи REST API описують, як сервер обробляє запити та відповідає на них.

Програма REST має клієнт-серверну архітектуру. Клієнтом є той, хто надсилає запити на ресурси і ніяк не пов'язаний із сховищем даних. Зберігання даних залишається всередині сервера. Сервери ж не пов'язуються з інтерфейсом користувача. Іншими словами, клієнт та сервер незалежні один від одного і можуть розвиватися окремо, що робить REST API більш гнучким та масштабованим.

Уніфікований інтерфейс чи єдиний інтерфейс – це головне, що відрізняє REST API від інших видів. Він передбачає наявність єдиного способу взаємодії з сервером незалежно від типу пристрою або програми.

Єдиний інтерфейс включає чотири основні принципи:

1. Identification of resources. Кожен ресурс REST має ідентифікатор, який не залежить від стану ресурсу. У ролі ідентифікатора виступає URL-адреса.

2. Manipulation of resources through representations (маніпулювання ресурсами через уявлення). Клієнт має уявлення ресурсу, що містить дані його видалення чи зміни. Клієнт відправляє подання на сервер (об'єкт JSON), який хоче змінити, видалити чи додати.

3. Self-descriptive messages (самодокументовані повідомлення). Такі повідомлення містять потрібну інформацію одержувача, щоб зрозуміти її. Жодної додаткової інформації не потрібна окрема документація або повідомлення. Кожне повідомлення має достатню інформацію для того, щоб сервер легко проаналізував запит.

4. Hypermedia as the engine of application state (гіпермедіа як двигун стану програми). Гіпермедіа має на увазі використання посилань для кожної відповіді, щоб клієнт міг знайти інші ресурси. Таким чином, всі взаємодії в REST проходять через гіпермедіа.

Сервер не містить жодної інформації про клієнта. Кожен запит включає всю необхідну інформацію для обробки. Інформація про сесію повністю зберігається за клієнта.

Кожна відповідь повинна містити дані про те, чи вона кешується чи ні і протягом якого часу відповідь може бути кешована на стороні клієнта. Якщо відповідь може бути кешована, то в наступних схожих запитах клієнт може використовувати ті ж дані без повторного звернення на сервер. При правильному використанні це покращує продуктивність та доступність.

У REST використовується багаторівнева система - ієрархія шарів, що створює певні обмеження на поведінку компонентів. У багаторівневій системі компоненти можуть бачити тільки ті компоненти, які розташовані на найближчих рівнях і з якими вони взаємодіють.

Code on demand (код на запит) - це опціональна функція, яка дозволяє клієнту завантажувати та виконувати код.

Насамперед, сама суть REST визначає його несумісність з іншими протоколами. REST — це архітектурний стиль, де архітектура REST представляє набір вимог, які потрібно дотримуватися, щоб надати веб-службу RESTful. А, наприклад, SOAP та RPC є протоколами обміну повідомленнями. Протокол описує ці повідомлення, коли архітектурний REST API лише визначає вимоги (у разі обмеження), які повідомлення мають виконувати.

Більшість програмних інтерфейсів працюють у форматі програма-додаток, коли REST діє за принципом Клієнт-Сервер. У разі клієнт і сервер залежать друг від друга, що дає більше гнучкості у роботі. Зазвичай API використовують конкретні формати повідомлень, наприклад SOAP використовує XML. REST не дотримується такого суворого принципу. Обмін повідомлень може відбуватися на основі JSON, XML або будь-якого іншого формату. Найпопулярнішим є JSON (JavaScript Object Notation). Популярність JSON цілком обґрунтована - це легко читається людиною і швидкий для аналізу формат обміну даними. JSON незалежний від мови та може бути використаний з будь-якою мовою програмування, крім JavaScript.

REST є гнучкішою архітектурою, завдяки чому і став таким популярним. Він вирішує проблеми, які викликає той же SOAP, будучи складним протоколом з розширеними функціями безпеки, що потребує більшої пропускнуої спроможності. REST складається із простих рекомендацій, даючи можливість розробникам реалізовувати вимоги у своєму форматі. Архітектурний стиль також має високу продуктивність. Що зробило особливо затребуваним для мобільних пристроїв, де швидкість завантаження є особливо важливою.

REST має певні переваги над іншими типами API. Саме тому всі великі компанії, такі як Twitter і Google, вже давно впровадили REST API для своїх продуктів. Адже це ідеальний спосіб передачі функцій та даних розробникам по

					КвРКІ 190128.19.01.24 ПЗ	Арк. 34
Зм.	Арк.	№ докum.	Підпис	Дата		

всьому світу та перевірений механізм для створення ефективних та масштабованих інтерфейсів для розробки ПЗ.

Також для роботи з веб-складовою нам потрібна серверна мова програмування PHP. Для роботи з нею використаємо фреймворк Laravel. Laravel - це PHP-фреймворк з відкритим кодом, надійний і простий для розуміння. Він відповідає шаблону проектування «модель-вид-контролер». Laravel повторно використовує існуючі компоненти різних фреймворків, що допомагає у створенні веб-додатків [40-42]. Розроблений таким чином веб-додаток більш структурований і прагматичний. Laravel пропонує багатий набір функціональних можливостей, які включають основні функції фреймворків PHP, таких як CodeIgniter, Yii та інших мов програмування, таких як Ruby on Rails. Laravel має дуже багатий набір функцій, які прискорюють веб-розробку.

Laravel має наступні переваги:

- Завдяки фреймворку Laravel веб-додаток стає більш масштабованим.
- Значна економія часу при розробці веб-програми, оскільки Laravel повторно використовує компоненти з інших фреймворків у розробці веб-програми.
- Він включає простори імен та інтерфейси, таким чином, допомагає організувати ресурси та керувати ними.

Composer — це інструмент, який містить усі залежності та бібліотеки. Це дозволяє користувачеві створювати проект з урахуванням згаданих фреймворків (наприклад, тих, що використовуються у встановленні Laravel).

Бібліотеки сторонніх розробників можна легко встановити за допомогою композитора. Усі залежності зазначені у файлі `composer.json`, який розміщено у вихідній папці. Інтерфейс командного рядка, який використовується в Laravel, називається `Artisan`. Він містить набір команд, які допомагають створити веб-додаток. Ці команди включено з фреймворку `Symphony`, що призводить до появи додаткових функцій у Laravel 5.1 (остання версія Laravel).

Laravel пропонує наступні ключові функції, що робить його ідеальним вибором для розробки веб-додатків. Перелік функцій наведено в таблиці 2.1

					КВРКІ 190128.19.01.24 ПЗ	Арк. 35
Зм.	Арк.	№ докum.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ КІБЕРФІЗИЧНОЇ СИСТЕМИ «РОЗУМНА ПАРКОВКА»

3.1 Принципи роботи серверної частини кіберфізичної системи «Розумна парковка»

На рисунку 3.1 зображена архітектура програмного забезпечення кіберфізичної системи “Розумна парковка”

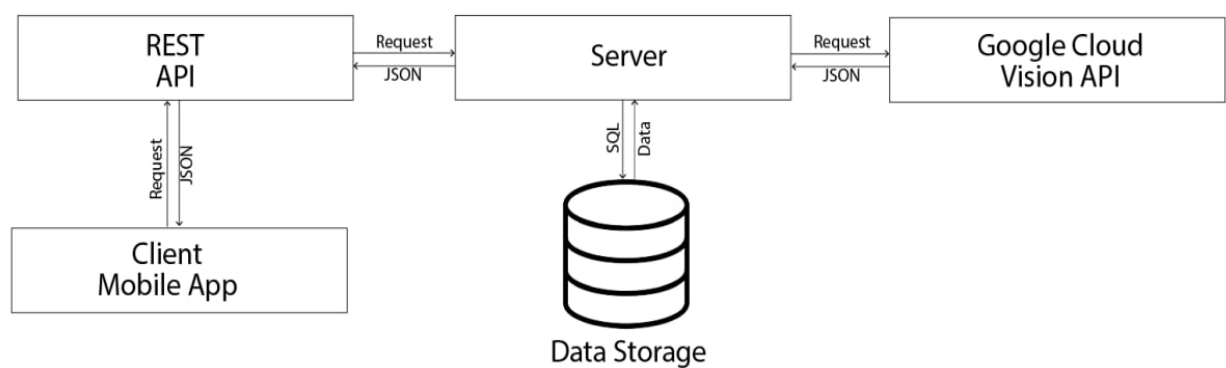


Рисунок 3.1 – Архітектура програмного забезпечення кіберфізичної системи “Розумна парковка”

З рисунку можна побачити, що все програмне забезпечення системи складається з програмного забезпечення клієнтської частини, яку ми не розглядаємо в даній роботі, програмних компонентів, які приєднуються через API - прикладний програмний інтерфейс - це REST API для зв'язку клієнта з сервером та бібліотека Google Cloud Vision для розпізнавання зображень автомобілів на основі штучного інтелекту, яка взаємодіє з сервером також за допомогою API [7].

В основі роботи системи розумної парковки, яка базується на використанні зображень з камер зовнішнього спостереження, лежить алгоритм розпізнавання образів на основі штучної нейронної мережі. При проектуванні системи постає завдання - використати вже готовий алгоритм у вигляді бібліотеки чи розробити власний алгоритм навчання штучної нейронної мережі. Оскільки другий спосіб є

доволі затратним у часі, вимагає значних апаратних та програмних ресурсів і спеціалізованої техніки, було прийнято рішення використати вже готову програмну бібліотеку для даної роботи.

Для розробки серверної частини програмного забезпечення, в основі якого лежить алгоритм розпізнавання зображень на основі штучного інтелекту, було складено план роботи:

1. Аналіз відомих технологій та штучних нейронних мереж для розпізнавання зображень.

2. Тестування вже існуючих нейронних мереж на датасетах (попередньо підготовлених наборах даних - зображень автомобільних парковок), які є у відкритому доступі.

3. Вибір найбільш ефективних технологій за результатами аналізу для подальшої роботи.

4. Збір власного датасету - набору зображень, знятих камерами зовнішнього спостереження з паркувальних майданчиків на території Хмельницького національного університету.

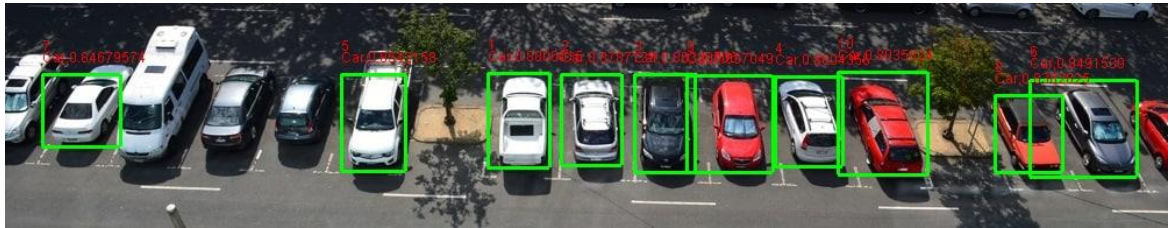
5. Тестування вибраних у пункті 3 нейронних мереж на власних датасетах - наборах реальних зображень автомобілів з парковки Хмельницького національного університету.

6. Вибір найефективнішої технології за результатами тестування на власноруч забраних датасетах для подальшої роботи під час розробки серверної частини.

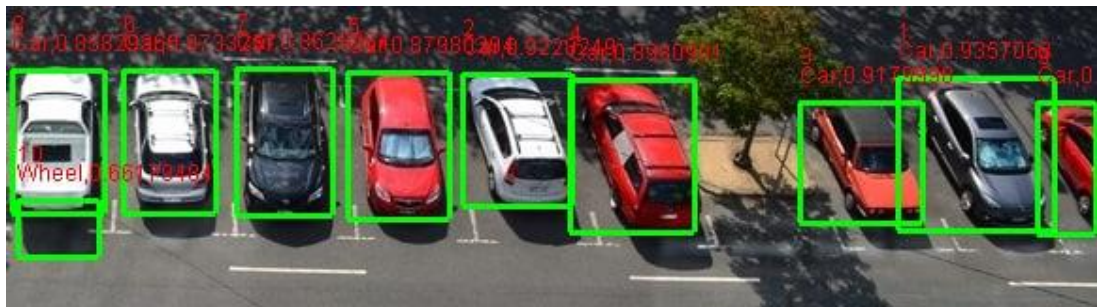
Серед відомих та широко застосовуваних технологій, які використовуються для розпізнавання зображень є бібліотеки на основі штучного інтелекту OpenCV [27] та Google Cloud Vision [26]. Для початку роботи було вирішено взяти попередньо підготовлені датасети із зображеннями парковок, які є у відкритому доступі [28-31] та спробувати протестувати ефективність роботи кожного з вже готових алгоритмів для розпізнавання зображень за допомогою штучного інтелекту. Для експерименту було використано бібліотеки OpenCV, Google Cloud

					КвРКІ 190128.19.01.24 ПЗ	Арк. 40
Зм.	Арк.	№ док.ум.	Підпис	Дата		

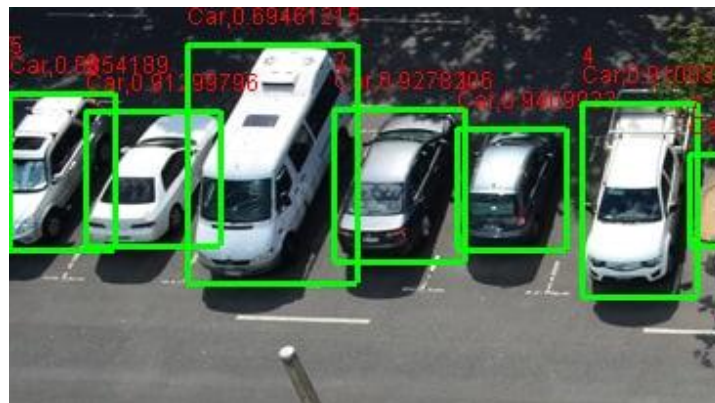
Vision та бібліотеку Detectron2 [32], написану мовою програмування Python. Результати тестування, а саме приклади зображень із датасетів [28-31] зображені на рисунку 3.2. Великою перевагою усіх протестованих алгоритмів є те, що вони показують точність, з якою розпізнається кожне зображення автомобіля. А бібліотека Google Cloud Vision розпізнає і окремі частини автомобіля, як от, наприклад, колесо на рисунку 3.2 б.



а)



б)



в)

Рисунок 3.2 – Принцип роботи алгоритмів для розпізнавання зображень кіберфізичною системою «Розумна парковка»

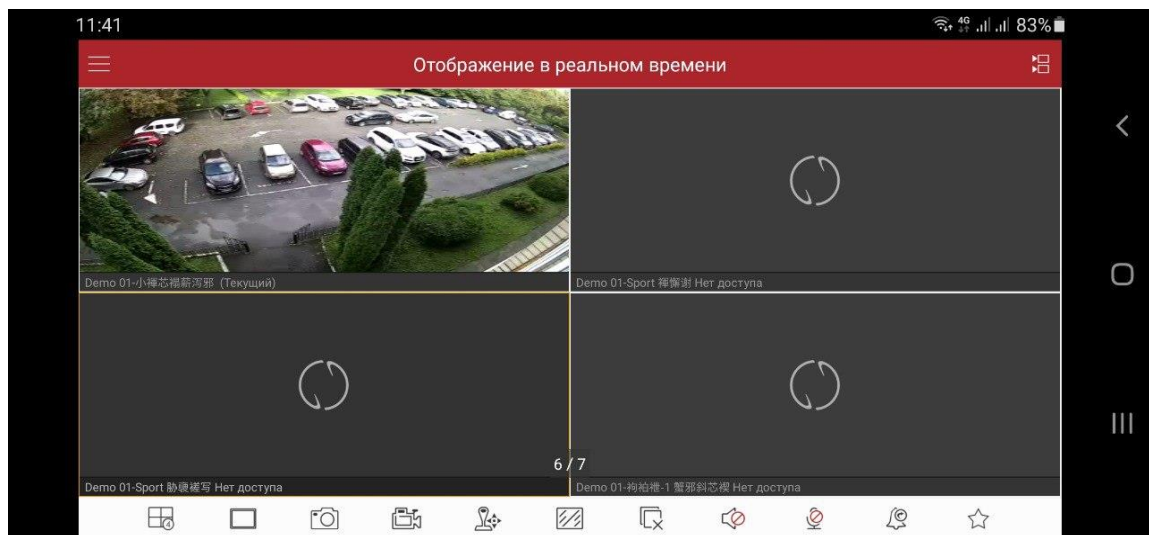
– частина датасету на дальній відстані;

- частина датасету при додатковій попередній обробці (обрізка зайвих частин та приближення);
- результат розпізнавання є найбільш точним при максимальному наближенні.

Отже, з рисунку 3.2 видно, що найкращі показники, тобто найвищу точність розпізнавання усі тестовані бібліотеки дають при максимальному наближенні зображень автомобілів та при обрізці зайвих частин зображення, які розсіюють фокус штучної нейронної мережі.

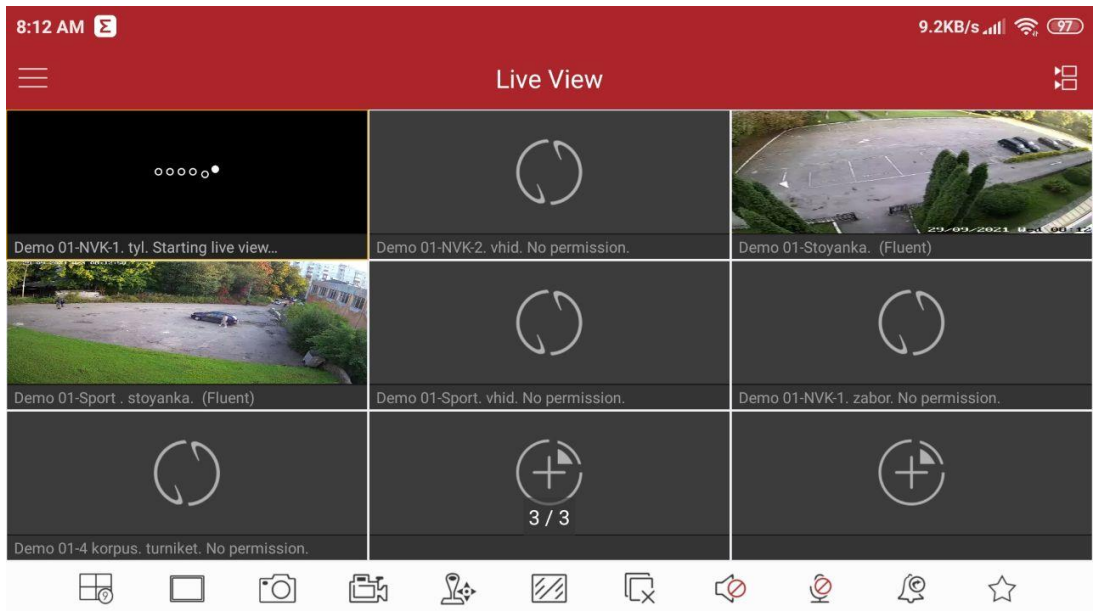
За результатами тестування найнижчу результативність та якість розпізнавання зображень показала бібліотека Detectron2[32], тому для подальшого тестування на власноруч зібраних датасетах ми її не використовуватимемо.

У ході роботи використаємо власноруч зібрані набори зображень, знятих камерами зовнішнього спостереження на паркувальних майданчиках Хмельницького національного університету[33]. Процес збору зображень для датасету показано на рисунку 3.3



а)

					КВРКІ 190128.19.01.24 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		



б)

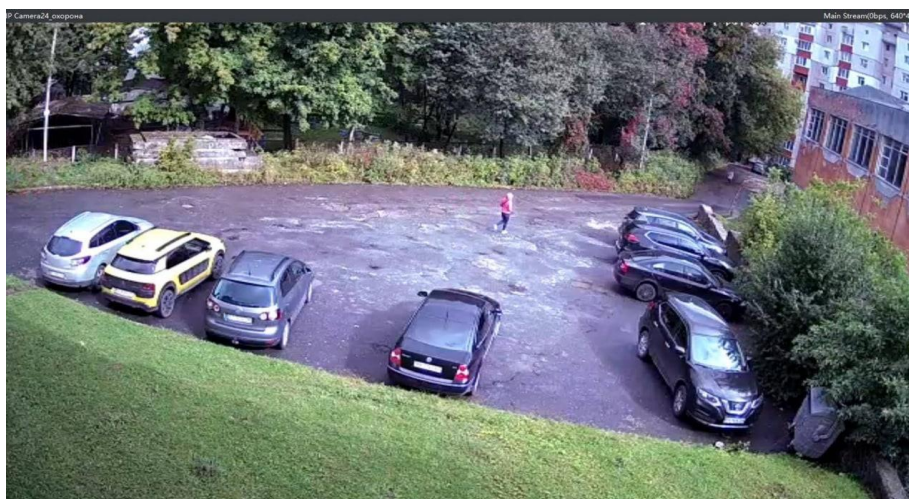


в)

Рисунок 3.3 – Процес збору наборів даних із зображень з камери зовнішнього спостереження Хмельницького національного університету

- а) підключення до камери зовнішнього спостереження за допомогою середовища iVMS-42000;
- б) отримання даних з усіх камер зовнішнього спостереження із видом на паркомайданчики;
- в) приклад зображення з камери для датасету.

Наступним етапом є тестування отриманого датасету за допомогою бібліотек, які показали найбільшу ефективність розпізнавання зображень на попередніх етапах, а саме OpenCV та Google Cloud Vision. Детально цей процес та результати описано в роботах [9-11]. Приклад зображення паркомайданчика із зібраного датасету та приклад тестування цього ж зображення за допомогою алгоритму для розпізнавання зображень на основі штучного інтелекту, а саме за допомогою бібліотеки OpenCV, зображено на рисунку 3.4 а) та б) відповідно.



а)



б)

Рисунок 3.4 - Приклад роботи підсистеми для розпізнавання зображення з камери зовнішнього спостереження паркомайданчика Хмельницького національного університету

Зм.	Арк.	№ докум.	Підпис	Дата

КВРКІ 190128.19.01.24 ПЗ

Арк.
44

- а) вихідне зображення парковки з камери зовнішнього спостереження;
- б) зображення після тестування за допомогою бібліотеки OpenCV.

Камери зовнішнього спостереження, які встановлено на паркомайданчиках ХНУ мають здатність розпізнавати об'єкти і в нічний час (рисунок 3.4 а) та б)), проте було вирішено не проводити тестування на предмет розпізнавання зображень автомобілів у нічний час, так як немає гострої потреби у користуванні паркомайданчиком вночі, так як це є у пікові години чи протягом робочого дня.



а)



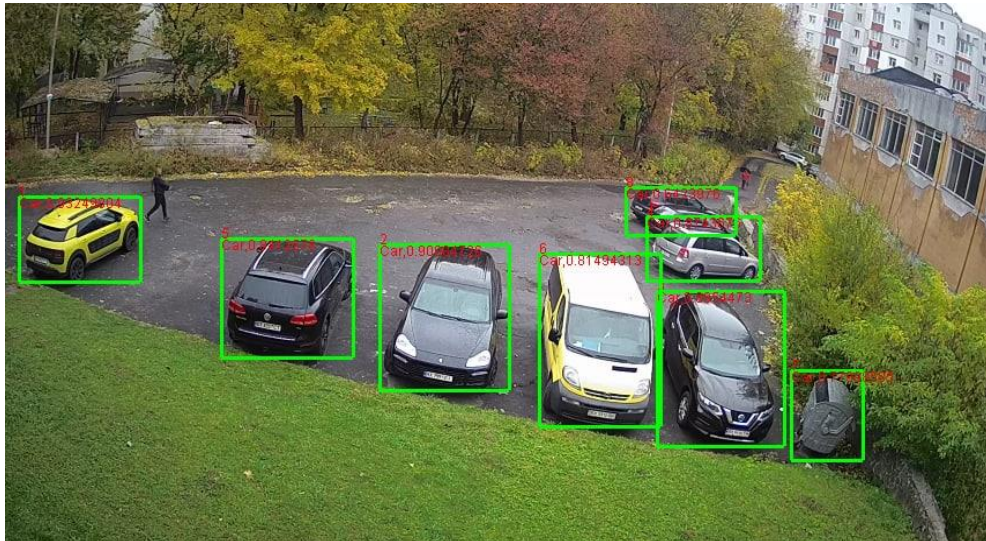
б)

Рисунок 3.5 – Зображення парковки з камер зовнішнього спостереження у нічний час

- а) автомобілі відсутні;
- б) наявний один автомобіль.

					КВРКІ 190128.19.01.24 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

З рисунку 3.4 б) можемо побачити, що алгоритм, який лежить в основі бібліотеки OpenCV дає значні похибки при розпізнаванні зображення з реального, а не з попередньо підготовленого та нормалізованого датасету. Приклади похибок при розпізнаванні зображень автомобілів кіберфізичною системою зображені на рисунку 3.6 а) та б).



а)



б)

Рисунок 3.6 – Похибки при розпізнаванні зображень автомобілів кіберфізичною системою

а) зображення паркомайданчика з датасету [33] при тестування за допомогою бібліотеки для розпізнавання зображень;

б) приклад розпізнавання штучним інтелектом сміттєвого баку як автомобіля з точністю 83%.

Отже, беручи до уваги результати попередніх досліджень [9-11], а саме точність розпізнавання кожної з бібліотек (згідно з [9] точність класифікації бібліотеки OpenCV+ЗНМ рівна 58,9%, а Google Cloud Vision - 77,08%) та результати тестування відомих бібліотек для розпізнавання образів на основі штучних нейронних мереж, а також ймовірність похибок при розпізнаванні, було прийнято рішення для подальшої роботи використовувати бібліотеку Google Cloud Vision, яка буде взаємодіяти з розробленим серверним програмним забезпеченням за допомогою прикладного програмного інтерфейсу

3.2 Структурна схема та алгоритм роботи серверної частини кіберфізичної системи «Розумна парковка»

Кіберфізична система для розумної парковки складається з таких трьох компонентів: зовнішньої частини (земельної ділянки для паркування автомобілів, розміченої паркувальною розміткою), апаратної частини (камери зовнішнього спостереження, мережеве з'єднання компонентів, сервера з безперебійним живленням, клієнтської частини у вигляді кросплатформного мобільного додатку) та програмної частини - програмного забезпечення для серверної та клієнтської частин. У ході роботи було побудовано структурну схему кіберфізичної системи для розумної парковки, для якої розробляється серверне програмне забезпечення. За основу було взято майданчики для паркування Хмельницького національного університету, оснащені камерами зовнішнього спостереження.

Структурна схема компонентів кіберфізичної системи для розумної парковки представлена на рисунку 3.7.

Camera Based Smart Parking Cyber-Physical System

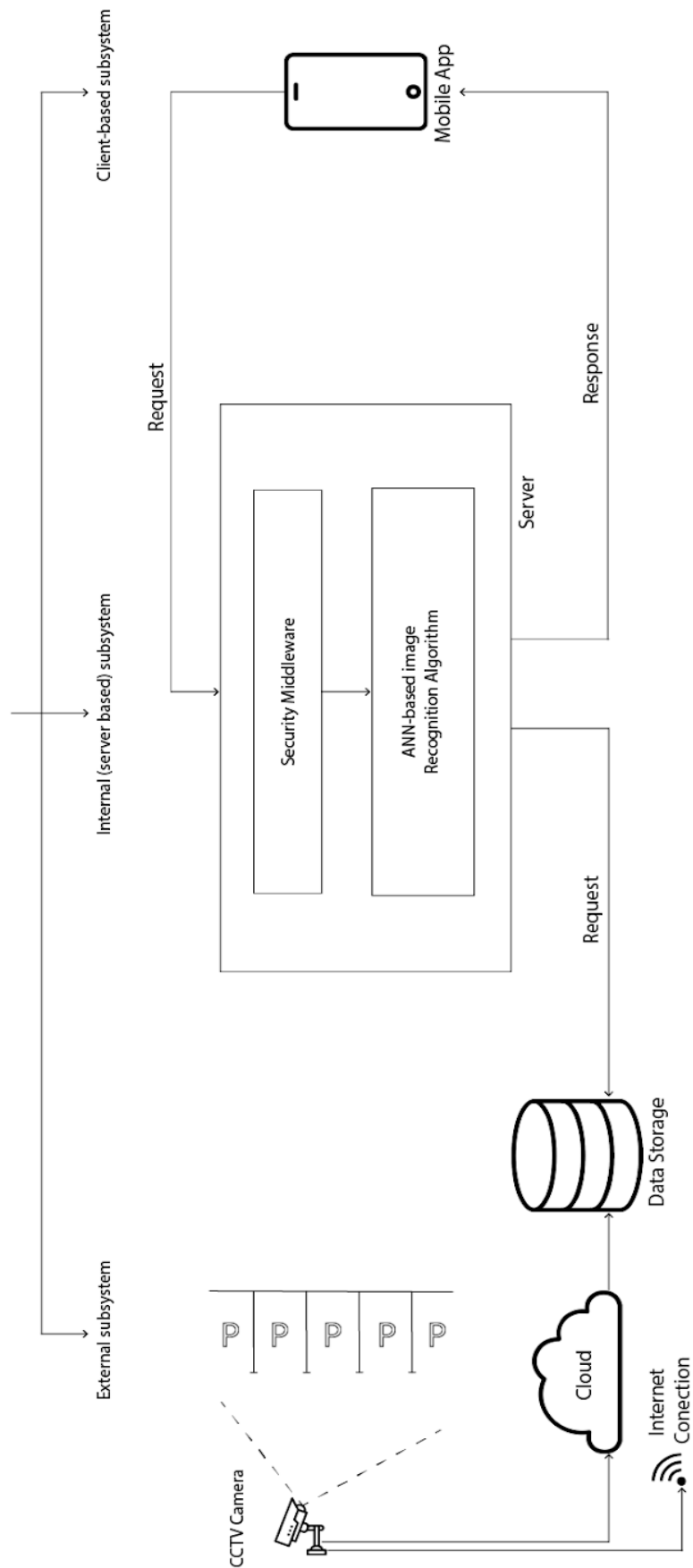


Рисунок 3.7 – Структурна схема компонентів кіберфізичної системи для розумної парковки

Зм.	Арк.	№ докум.	Підпис	Дата

На схемі присутні такі компоненти: камери зовнішнього спостереження з високою роздільною здатністю, підключення до мережі Інтернет, передача даних за допомогою хмарних технологій, сховище даних, серверна частина ПЗ, до складу якої входить проміжне ПЗ для перевірки безпеки запитів від клієнтського додатку, алгоритм для розпізнавання зображень на основі штучного інтелекту та програма-обробник та клієнтська частина у вигляді кросплатформного мобільного додатку.

Для розробки серверної частини ПЗ було складено блок-схему алгоритму роботи серверної частини кіберфізичної системи розумної парковки, яка представлена на рисунку 3.8.

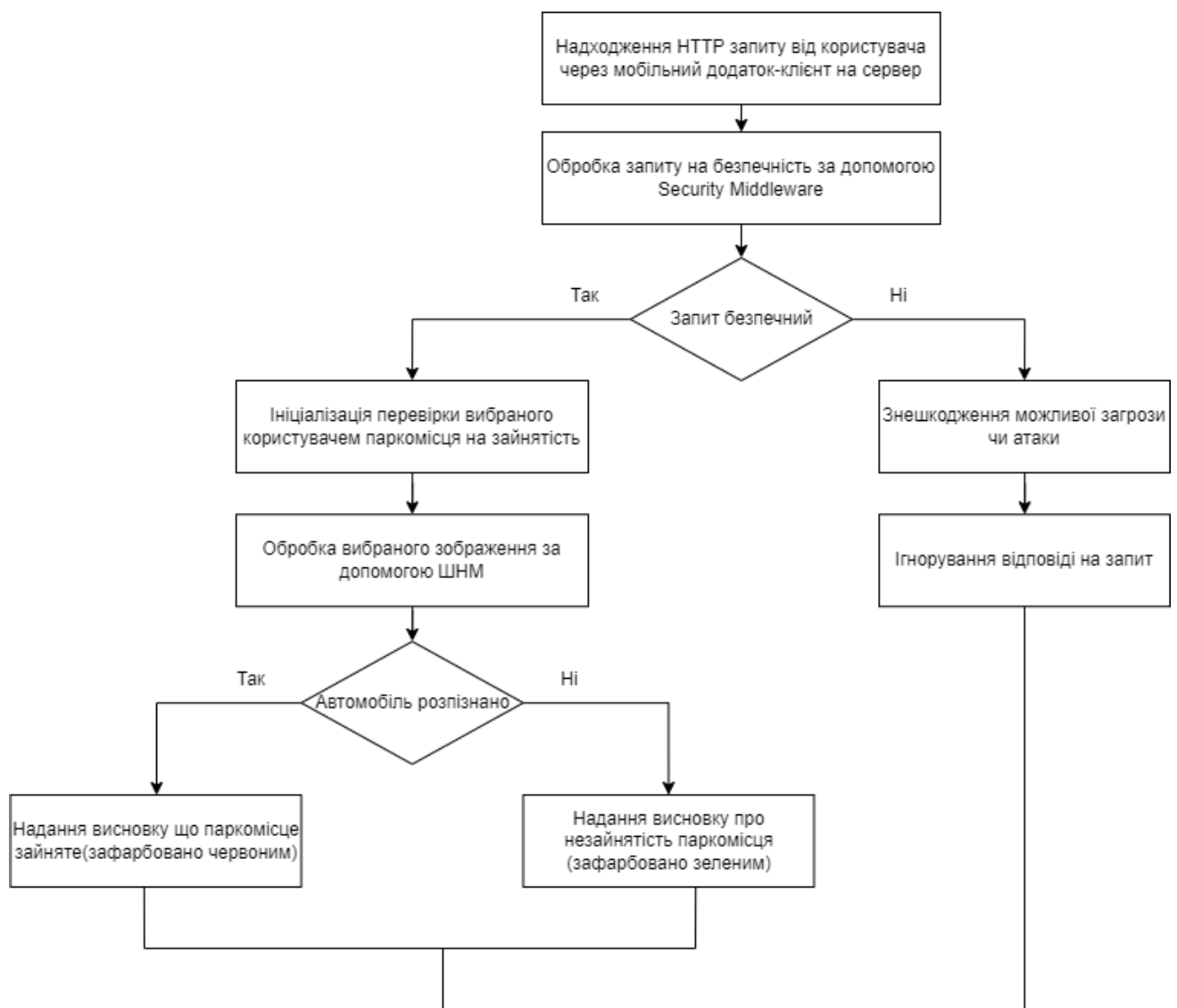


Рисунок 3.8 – Алгоритм роботи серверної частини кіберфізичної системи розумної парковки

Коли користувач заходить в мобільний додаток для знаходження вільних паркомісць на “розумній парковці” оснащених системою відеоспостереження, на сервер надходить HTTP-запит на сервер.

Далі програма-обробник на сервері опрацьовує запит на безпечність за допомогою програмної компоненти Security Middleware. У цьому випадку підозрілі та потенційно небезпечні запити не потрапляють в подальше обробку сервера, а знешкоджуються на цьому рівні. Це забезпечує додатковий захист системи від зловмисницького коду, SQL-ін’єкцій та ймовірних витоків даних чи внесення змін у вихідний код, який знаходиться на сервері.

Якщо запит безпечний, він відправляється на подальше опрацювання сервером. Відбувається перевірка вибраного користувачем паркомайданчика на предмет наявності/відсутності вільних паркомісць з їх візуалізацією за допомогою кольорів. Це забезпечує використання алгоритмів для розпізнавання зображень на базі штучного інтелекту у бібліотеці Google Cloud Vision, яка взаємодіє з програмою-обробником на сервері через прикладний програмний інтерфейс (API). Якщо автомобіль розпізнано, то паркомісце ідентифікується як зайняте і позначається червоним прямокутником на користувацькому інтерфейсі у клієнтському додатку. Якщо автомобіль не розпізнано, то паркомісце ідентифікується як вільне і позначається зеленим прямокутником на користувацькому інтерфейсі у клієнтському додатку.

Також користувач при натисканні на прямокутник-позначення паркомісця, може переглянути відео з камери зовнішнього спостереження вибраного паркомайданчика, щоб мати можливість заздалегідь побачити як зручніше заїхати на паркомісце та спостерігати за своїм автомобілем у режимі реального часу для додаткового забезпечення безпеки транспортного засобу на паркомайданчику. На рисунку 3.9 зображено процес взаємодії серверної та клієнтської частин кіберфізичної системи для розумної парковки на основі технології розпізнавання зображень з використанням штучної нейронної мережі. У лівому верхньому кутку діаграми зображений сервер обробки зображень Google Cloud Vision API, який був

обраний в результаті експериментів у розділі 2. У лівому нижньому кутку зображений клієнтський пристрій. Це може бути будь-який смартфон з операційною системою не старішою, ніж Android 7. По центру зображений фізичний сервер на базі операційної системи Linux та веб-сервер Apache Tomcat Server. З правого боку зображений сервер бази даних з СУБД MySQL.

Схема роботи компонентів кіберфізичної системи для розумної парковки з використанням штучної нейронної мережі на рисунку 3.9.

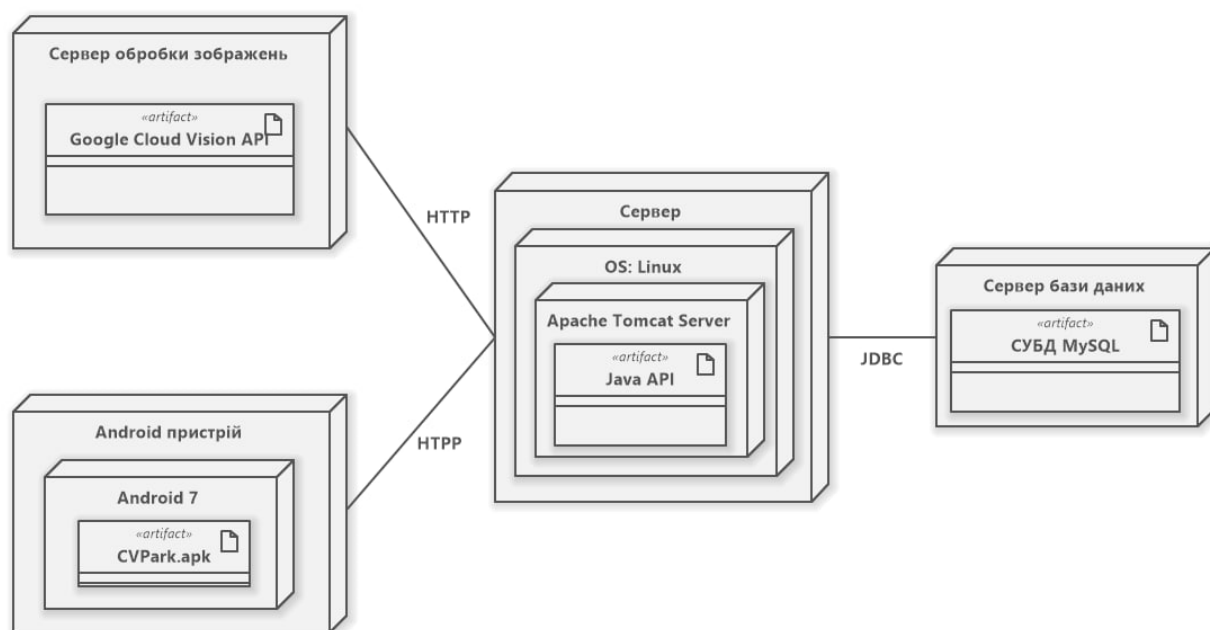


Рисунок 3.9 – Схема роботи компонентів кіберфізичної системи для розумної парковки

Для демонстрації роботи кіберфізичної системи для розумної парковки на основі розпізнавання зображень автомобілів використаємо ширококутну камеру зовнішнього спостереження, кут огляду якої охоплюватиме весь парко майданчик. Смуга розмітки парко майданчика має бути чітко видимою та накресленою з дотриманням стандартів. Пропонована у роботі схема паркомайданчика представлена на рисунку 3.10. Камера зовнішнього спостереження, яка використовується для кіберфізичної системи «Розумна парковка», представлена на рисунку 3.11.

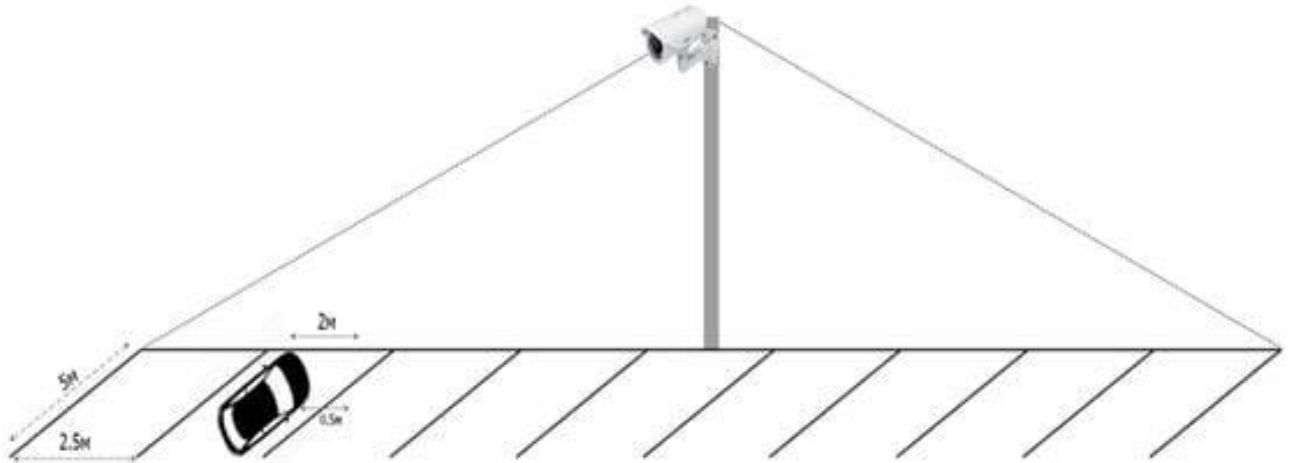


Рисунок 3.10 – Складові частини паркомайданчика кіберфізичної системи «Розумна парковка»



Рисунок 3.11 – Камера зовнішнього спостереження для кіберфізичної системи розумної парковки

3.3 Реалізація серверної підсистеми кіберфізичної системи «Розумна парковка»

Для реалізації серверної підсистеми кіберфізичної системи розумної парковки було обрано Python та фреймворк Flask, бібліотеку OpenCV та мову програмування PHP та фреймворк Laravel для роботи з REST API. Структуру файлів проєкту у середовищі розробки PHP Storm наведено на рисунку 3.12.

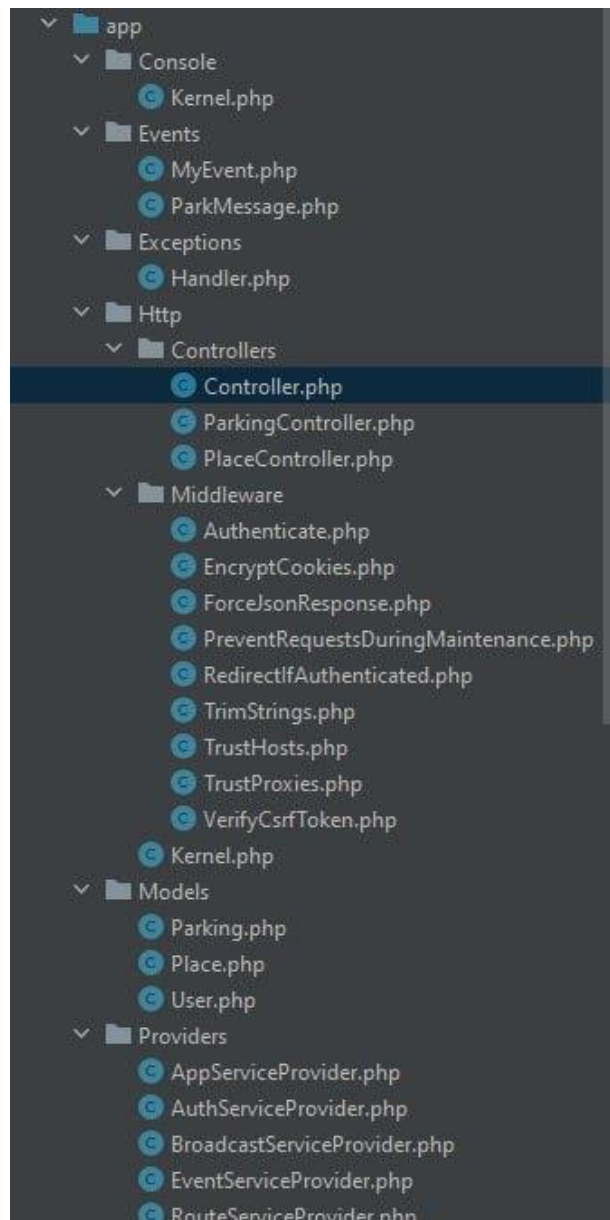
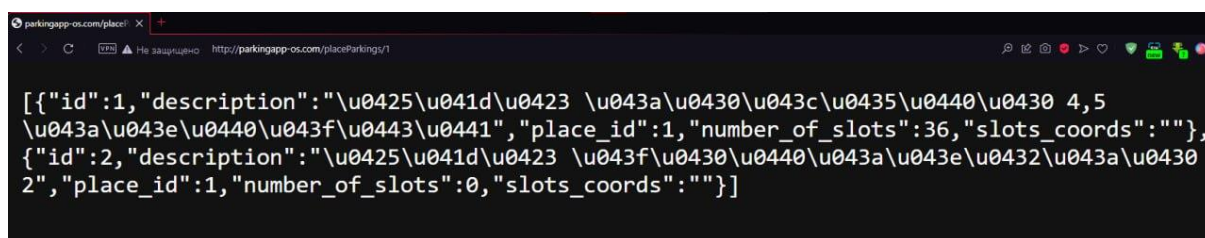


Рисунок 3.12 – Структура файлів проєкту серверної підсистеми кіберфізичної системи “Розумна парковка”

Оскільки програмне забезпечення розробленої кіберфізичної системи базуватиметься на принципах клієнт-серверної архітектури, однією з задач цієї роботи було забезпечення передачі інформації з сервера до клієнтського додатку, тобто забезпечення обміну даними через REST API. Тому було проведено визначення endpoints для REST API - endpoint placeParkings/{id}. Тобто результатом є json-файл, який повертає всі оброблені парковки у місця з вказаним ідентифікатором ID. Json-файл з результатами виконання програми наведено на рисунку 3.13.



```
[{"id":1,"description":"\u0425\u041d\u0423 \u0430\u0430\u043c\u0435\u0440\u0430 4,5\n\u0430\u043e\u0440\u0440\u0443\u0443\u0441","place_id":1,"number_of_slots":36,"slots_coords":""},\n{"id":2,"description":"\u0425\u041d\u0423 \u0430\u0430\u043c\u0435\u0440\u0430\u043e\u0432\u0430\u0430\u0430\n2","place_id":1,"number_of_slots":0,"slots_coords":""}]
```

Рисунок 3.13 – Json-файл з результатами виконання програми для серверної підсистеми кіберфізичної ситсеми «Розумна парковка»

Отже, можна дійти висновку, що розроблена серверна підсистема кіберфізичної системи розумної парковки працює добре та встановлює зв'язок з REST API для подальшого обміну даними з програмою-клієнтом.

Подальші зусилля будуть спрямовані на реалізацію клієнтської частини кіберфізичної сиситеми розумної парковки у вигляді кросплатформного мобільного застосунку.

3.4 Висновки

Отже, у даному розділі було запропоновано принцип роботи серверної підсистеми кіберфізичної системи “Розумна парковка” із застосуванням алгоритмів на базі штучних нейронних мереж та машинного зору. Згідно концепції архітектури кіберфізичної системи “Розумна парковка”, інтеграція серверної та клієнтської

частин буде відбуватися за допомогою використання прикладного програмного інтерфейсу (REST API).

Також було спроектовано структурну схему кіберфізичної системи для розумної парковки на основі технології розпізнавання зображень з використанням штучної нейронної мережі. Для розробки даного програмного забезпечення було обрано водоспадну модель життєвого циклу через її простоту та ефективність в проектах такого типу.

Розроблена програмна система дозволяє користувачу побачити вільні та зайняті паркомісця на громадських парковках, оснащених камерами. Тестові датасети було взято з камер зовнішнього спостереження між навчальними корпусами 3 та 4 Хмельницького національного університету.

Також було спроектовано структурну схему та алгоритм роботи серверної підсистеми кіберфізичної системи “Розумна парковка” та виконано реалізацію у вигляді скрипта, який забезпечує передачу інформації з сервера до клієнтського додатку через прикладний програмний інтерфейс. Для реалізації було використано такий стек технологій як Python та фреймворк Flask, бібліотеку OpenCV та мову програмування PHP та фреймворк Laravel для роботи з REST API. Також було проведено експеримент із застосуванням запропонованої в роботі інформаційної технології на території студмістечка Хмельницького національного університету та наведено результати тестування продуктивності розробленої системи

					КвРКІ 190128.19.01.24 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У дипломній роботі за результатами виконаних теоретичних та практичних досліджень було розроблено серверну підсистему для кіберфізичної системи для розумної парковки на основі алгоритмів із застосуванням штучних нейронних мереж та машинного зору для розпізнавання зображень автомобілів з камер зовнішнього спостереження, які ведуть на паркомайданчик Хмельницького національного університету.

У першому розділі проведено аналіз проблеми пов'язаної із паркуванням автомобілів у рамках концепції Розумних міст. Запропоновано методи вирішення даної проблеми за допомогою кіберфізичної системи "Розумна парковка". Також було проведено дослідження та опис методів реалізації та проаналізовано позитивні чи негативні сторони тих чи інших технологій. Також проведено детальний аналіз подібних технологій які використовують в інших країнах. Крім того, наведено приклади вже існуючих обладнаних розумних парковок та клієнтського програмного забезпечення у вигляді мобільних додатків для операційних систем Android та iOS.

У другому розділі було проведено обґрунтування вибору компонентів та середовища реалізації серверної підсистеми кіберфізичної системи розумної парковки. А саме, було виконано опис апаратного середовища реалізації - наведено переваги та недоліки існуючих способів реалізації розумних парковок та зроблено висновок, що спосіб реалізації за допомогою камер зовнішнього спостереження є найбільш економічно вигідним та простим у реалізації за рахунок легкого встановлення та можливості масштабування. Також було визначено функційні і нефункційні вимоги до серверної підсистеми кіберфізичної системи розумної парковки, вимоги до системи та вимоги щодо взаємодії з навколишнім середовищем. Важливо відмітити, що у ході дослідження проектованої системи серед переваг було визначено додаткову вимогу, яка забезпечується застосуванням камер - це конфіденційність та безпека. Також було обрано методи, стек технологій

					КвРКІ 190128.19.01.24 ПЗ	Арк. 57
Зм.	Арк.	№ докum.	Підпис	Дата		

та середовище для реалізації програмного забезпечення серверної підсистеми кіберфізичної системи розумної парковки.

У третьому розділі було запропоновано принцип роботи серверної підсистеми кіберфізичної системи “Розумна парковка” із застосуванням алгоритмів на базі штучних нейронних мереж та машинного зору. Згідно концепції архітектури кіберфізичної системи “Розумна парковка”, інтеграція серверної та клієнтської частин буде відбуватися за допомогою використання прикладного програмного інтерфейсу (REST API). Також було спроектовано структурну схему та алгоритм роботи серверної підсистеми кіберфізичної системи “Розумна парковка” та виконано реалізацію у вигляді скрипта, який забезпечує передачу інформації з сервера до клієнтського додатку через прикладний програмний інтерфейс. Для реалізації було використано такий стек технологій як Python та фреймворк Flask, бібліотеку OpenCV та мову програмування PHP та фреймворк Laravel для роботи з REST API.

Подальше дослідження буде присвячено розробці клієнтської частин у вигляді мобільного застосунку, який дозволяє користувачу побачити кількість та розміщення вільних та зайнятих паркомісць на парковках, які обладнані камерами зовнішнього спостереження.

Практична цінність отриманих результатів полягає у розробці серверної підсистеми кіберфізичної системи для розумної парковки на базі алгоритмів із застосуванням штучних нейронних мереж та машинного зору для розпізнавання зображень.

Результати, отримані під час досліджень в рамках дипломної роботи, пройшли апробацію у вигляді участі у Всеукраїнській науково-практичній конференції “Актуальні Проблеми Комп’ютерних Наук (АПКН-2022)”, м. Хмельницький та Всеукраїнській науково-практичній конференції Інформаційні технології та інженерія (IT&I-2023), м. Миколаїв, за результатами яких було опубліковано тези у збірниках конференцій.

					КВРКІ 190128.19.01.24 ПЗ	Арк. 58
Зм.	Арк.	№ докum.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Розумне паркування в Києві URL: <https://www.ukrinform.ua/rubric-kyiv/2283219-rozumne-parkuvanna-u-kievi-stali-do-roboti-persi-10-inspektoriv.html> (дата звернення: 12.07.2022).
2. У Львові з'явилась перша розумна парковка URL: <https://rubryka.com/2021/11/05/rozumna-parkovka-u-lvovi/> (дата звернення: 12.07.2022).
3. Інфрачервоні датчики руху: поради по використанню URL: <https://res.ua/blog/infrachervoni-datchiki-ruhu-poradi-po-vikoristannju.html> (дата звернення: 12.07.2022).
4. Чому датчики руху реагують на тварин та як цього уникнути URL: <https://ajax.systems.ua/blog/what-is-pet-immunity-in-motion-detectors-and-how-to-use-it-correctly/> (дата звернення: 12.07.2022).
5. Відеоспостереження URL: <https://uk.wikipedia.org/wiki/Відеоспостереження> (дата звернення: 12.07.2022).
6. Авсієвич В.Р., Кузьмін А.А. Дослідження вразливостей системи розумної парковки та способи їх усунення. *Актуальні Проблеми Комп'ютерних Наук* (АПКН-2021), Хмельницький, Україна, 18-19 жовтня 2022. Хмельницький: ХНУ, 2022. с. 11-14.
7. Pavlova O., Kovalenko V., Novorushchenko T., Avsiyevych V. Neural network based image recognition method for smart parking. *Comput. Syst. Inf. Technol.* 3, 2021. pp. 49–55.
8. Авсієвич В., Коваленко В. Аналіз інформаційних технологій для розумної парковки на основі штучних нейронних мереж. *Актуальні Проблеми Комп'ютерних Наук* (АПКН-2021), Хмельницький, Україна, 15-16 жовтня 2021. Хмельницький: ХНУ, 2021. С. 12-14

					КВРКІ 190128.19.01.24 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

9. Radiuk P., Pavlova O., El Bouhissi H., Avsiyevych V., Kovalenko V. Convolutional Neural Network for Parking Slots Detection. *CEUR Workshop Proceedings*, 2022. 3156, pp. 284–293.

10. Avsiyevych V., Kovalenko V. Cyber-physical system for smart parking based on computer vision technology Black Sea Science 2022. *Proceedings of the International Competition of Student Scientific Works*. Odesa National University of Technology. Odessa: ONUT. 2022. pp. 335-346.

11. Львівська компанія SoftServe почала тестування паркінг-системи на базі машинного навчання. zaxid.net. URL: https://zaxid.net/lvivska_kompaniya_softserve_pochala_testuvannya_parking_sistemi_na_bazi_mashinnogo_navchannya_n1471000 (дата звернення: 30.11.2022).

12. Паркінг в Буковелі. Трансфер до Карпат. URL: <https://transferdokarpat.com.ua/articles/bukovel-vartist-poslugiparkingiv> (дата звернення: 30.11.2022).

13. У Києві стартував пілот із впровадження «розумної» системи паркування. Офіційний портал Києва. URL: https://kyivcity.gov.ua/news/u_kiyevi_startuvav_pilot_iz_vprovadzhennya_rozumno_sistemi_parkuvannya/ (дата звернення: 30.11.2022).

14. Інтелектуальна система парковки Асер. URL: <https://www.acer.com/ac/ru/RU/content/acerdesign-smart-parking> (дата звернення: 30.11.2022).

15. Apple App Store URL: <https://www.apple.com/ua/app-store/> (дата звернення: 30.11.2022).

16. Google Play Market URL: <https://play.google.com/store> (дата звернення: 30.11.2022).

17. Вимоги до програмного забезпечення URL: https://uk.wikipedia.org/wiki/Вимоги_до_програмного_забезпечення (дата звернення: 03.12.2022).

18. Функціональні вимоги URL: https://uk.wikipedia.org/Функціональні_вимоги (дата звернення: 03.12.2022).

					КВРКІ 190128.19.01.24 ПЗ	Арк. 60
Зм.	Арк.	№ док.ум.	Підпис	Дата		

19. Нефункціональні вимоги [URL:https://uk.wikipedia.org/wiki/Нефункціональні_вимоги](https://uk.wikipedia.org/wiki/Нефункціональні_вимоги) (дата звернення: 03.12.2022).
20. Коваленко В. В. Кіберфізична система розумної парковки на основі технології комп'ютерного зору: кваліфікаційна робота магістра: 123 Комп'ютерна інженерія. ХНУ. Хмельницький, 2022. 113с.
21. Gollapudi S. Learn computer vision using opencv: with deep learning CNNs and RNNs : eBook. Berkeley, CA : Apress. 2019. 171 p.
22. Balaji G.V., Bharath K., Nithin S., Pranesh D.M., Shilpa S.B. Object detection using OpenCV and deep learning. *International Journal for Research in Applied Science and Engineering Technology*. 2021. Vol. 9, No. 1. pp. 3920-3923.
23. Parking Startups Are Cashing In on America's Traffic Surge [URL:https://www.bloomberg.com/news/articles/2021-07-22/parking-startups-cash-in-on-america-s-post-pandemic-traffic-surge-with-apps](https://www.bloomberg.com/news/articles/2021-07-22/parking-startups-cash-in-on-america-s-post-pandemic-traffic-surge-with-apps) (дата звернення: 07.12.2022)
24. Vision AI. Google Cloud. URL: <https://cloud.google.com/vision> (дата звернення: 02.12.2022)
25. Learn OpenCVC++ in 4 hours URL: <https://www.youtube.com/watch?v=2FYm3GOonhk> (дата звернення: 02.12.2022)
26. Amato G., Carrara F., Falchi F., Gennaro C., Vairo C. CNRPark+EXT. A Dataset for Visual Occupancy Detection of Parking Lots. URL:<http://cnrpark.it/> (дата звернення: 07.12.2022)
27. Parking Lot Database URL: <https://web.inf.ufpr.br/vri/databases/parking-lot-database/> (дата звернення: 07.12.2022)
28. Melbourne University Dataset URL: https://melbourne.figshare.com/articles/dataset/MATLABCodeCNNSVM_zip/1297893/2/1?file=24726374 (дата звернення: 07.12.2022)
29. Marek M. Official repository for the "Image-Based Parking Space Occupancy Classification: Dataset and Baseline" paper. URL: <https://github.com/martin-marek/parking-space-occupancy> (дата звернення: 07.12.2022)

					КВРКІ 190128.19.01.24 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

30. Detectron2 - a platform for object detection, segmentation and other visual recognition tasks. URL:<https://github.com/facebookresearch/detectron2> (дата звернення: 07.12.2022)

31. Репозитарій із датасетом зображень, зібраних з камер зовнішнього спостереження ХНУ URL: <https://github.com/soolstafir/khnu-parking-lot> (дата звернення: 02.12.2022)

32. IP-відеокамера та комплектуючі URL: https://rozetka.com.ua/hikvision_ds_2cd2121g0_is_c_2_8_mm/p322796167/characteristics/ (дата звернення: 07.12.2022)

33. IP-відеокамера Dahua DH IPC HDW2431 URL: https://rozetka.com.ua/dahua_dh_ipc_hd_w2431tp_as_s2_3_6_mm/p175174490/characteristics/ (дата звернення: 07.12.2022)

34. Келер А., Брадські Г. Вивчення OpenCV 3: комп'ютерний зір у C++ з бібліотекою OpenCV 1st Edition. O'Reilly Media .2017 . 1024 с.

35. Кузьмін А. А., Павлова О. О. Застосування комп'ютерного зору для кіберфізичної системи розумної парковки. *"Інформаційні технології та інженерія - 2023"*, 7-10 лютого 2023, Миколаїв, Україна, с. 45-47

36. Introduction to OpenCV-Python Tutorials URL: https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html (дата звернення: 19.02.2023)

37. How to Build an Effective API Security Strategy URL:<https://www.gartner.com/en/documents/3834704> (дата звернення: 19.02.2023)

38. The PHP Framework for Web Artisans URL:<https://laravel.com/> (дата звернення: 19.02.2023)

39. Laravel Framework GitHub URL: <https://github.com/laravel/framework> (дата звернення: 19.02.2023)

40. Laravel - Overview URL: https://www.tutorialspoint.com/laravel/laravel_overview.htm (дата звернення: 19.02.2023)

					КВРКІ 190128.19.01.24 ПЗ	Арк. 62
Зм.	Арк.	№ док.ум.	Підпис	Дата		

41. Hovorushchenko T., Pavlova O., Kostiuk M. Method of Increasing the Security of Smart Parking System. *JCSANDM*. vol. 12, no. 03, 2023. pp. 297–314.
42. Hovorushchenko T., Boyarchuk A., Pavlova O., Bobrovnikova K. Agent-Oriented Information Technology for Assessing the Initial Stages of the Software Life Cycle. *ICTERI Workshops*. 2019. pp. 617-632.
43. Understanding OWASP Mobile Top 10 Risks with Real-world Cases. URL: <https://appinventiv.com/blog/owasp-mobile-top-10-real-world-cases/amp> (дата звернення: 19.02.2023)
44. The top API security risks and how to mitigate them. URL: <https://appinventiv.com/blog/how-to-mitigate-api-security-risks/> (дата звернення: 19.02.2023)
45. Lopatto I., Hovorushchenko T., Popov P., Pavlova O. Intelligent Multi-Agent System for Improving the Quality of Software by Taking into Account the Information of the Subject Area at All Stages of its Development. *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IDAACS 2021. 2021. 1. pp. 548–551.
46. Amiri W. A., Baza M., Banawan K., Mahmoud M., Alasmay W., Akkaya K. Towards Secure Smart Parking System Using Blockchain Technology. 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). 2020. pp. 1-2. doi: 10.1109/CCNC46108.2020.9045674.
47. Waheed P., Krishna P.V. Comparing Biometric and Blockchain Security Mechanisms in Smart Parking System. *2020 International Conference on Inventive Computation Technologies (ICICT)*. 2020. pp. 634-638. doi: 10.1109/ICICT48043.2020.9112483.
48. Kumar M., Khan M. H., Umar M. S. Smart parking system using RFID and GSM technology. *2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, 2017. pp. 180-184. doi: 10.1109/MSPCT.2017.8364000.

					КВРКІ 190128.19.01.24 ПЗ	Арк. 63
Зм.	Арк.	№ докum.	Підпис	Дата		

49. Lee C., Leng FTJ., Habeeb RAA., Amanullah MAA, Rehman M., Edge computing-enabled secure and energy-efficient smart parking. *A review, Microprocessors and Microsystems*. Volume 93, 2022
50. Ahmed S., Soaibuzzaman M., Rahman S., Rahaman S. A Blockchain-Based Architecture for Integrated Smart Parking Systems. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2019, pp. 177-182, doi: 10.1109/PERCOMW.2019.8730772.
51. Biyik M., Allam M., Pieri G., Moroni G., O’Fraifer M., O’Connell M., Olariu M., Khalid M. Smart Parking Systems: Reviewing the Literature, Architecture and Ways Forward. *Smart Cities*. 2021. 4(2). pp. 623-642.
52. Imbugwa G.B., Mazzara M. Towards a Secure Smart Parking Solution for Business Entities. *Advanced Information Networking and Applications*. AINA 2021. Lecture Notes in Networks and Systems. vol 227. Springer. pp. 469-478.
53. Abdulkader O., Bamhdi A. M., Thayananthan A., Jambi K., Alrasheedi A. A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *2018 15th Learning and Technology Conference (L&T)*. 2018. pp. 102-106, doi: 10.1109/LT.2018.8368492.
54. Singh SK, Pan Y, Hyuk J. Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City Environment. *Sustainable Cities and Society*. vol. 76. 2022.
55. Garcia M., Rose P., Sung P., El-Tawab E., Secure Smart Parking at James Madison University via the Cloud Environment (SPACE). *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, 2016. pp. 271-276. doi: 10.1109/SIEDS.2016.7489313.
56. Anwar A., Ijaz-ul-Haq N., Saadati P. Smart Parking: Novel Framework of Secure Smart Parking Solution using 5G Technology. *2021 IEEE International Smart Cities Conference (ISC2)*. 2021, pp. 1-4. doi: 10.1109/ISC253183.2021.9562776.
57. Hovorushchenko T., Pavlova O., Avsiyevych V. Method of Assessing the Impact of External Factors on Geopositioning System Operation Using Android GPS

					КВРКІ 190128.19.01.24 ПЗ	Арк. 64
Зм.	Арк.	№ доквм.	Підпис	Дата		

API. 2021 *International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)*, 2021, 1. pp. 295–298

58. Waheed A., Krishna P. V., Gitanjali J., Sadoun B., Obaidat M. Learning automata and reservation based secure smart parking system. *Methodology and simulation analysis, Simulation Modelling Practice and Theory*. vol. 106. 2021.

59. Atif Y., Ding J., Jeusfeld MA., Internet of Things Approach to Cloud-based Smart Car Parking. *Procedia Computer Science*. vol. 98. 2016

60. Hakim I.M., Christover M., Jaya Marindra A.M. Implementation of an image processing based smart parking system using Haar-Cascade method. *2019 IEEE 9th Symposium on Computer Applications Industrial Electronics (ISCAIE-2019)*. pp. 222–227. IEEE Inc., Penang, Malaysia, 27-28 April 2019. DOI:10.1109/ISCAIE.2019.8743906

61. Manjula G., Rajulu G.G., Anand, J.T. Thirukrishna. Implementation of smart parking application using IoT and machine learning algorithms. *Computer Networks and Inventive Communication Technologies*. Springer Singapore, Singapore. 2022. pp. 247–257. doi:10.1007/978-981-16-3728-5_18.

62. Vakula D., Kolli Y.K.. Low cost smart parking system for smart cities. *International Conference on Intelligent Sustainable Systems*, 2017. DOI: 10.1109/ISS1.2017.8389415.

63. Tkachenko R., Izonin I., Dronyuk I., Logoyda M., Tkachenko P. Recovery of missing sensor data with grnn-based cascade scheme. *International Journal of Sensors, Wireless Communications and Control*. 2021, vol. 11, no.5. pp. 531–541

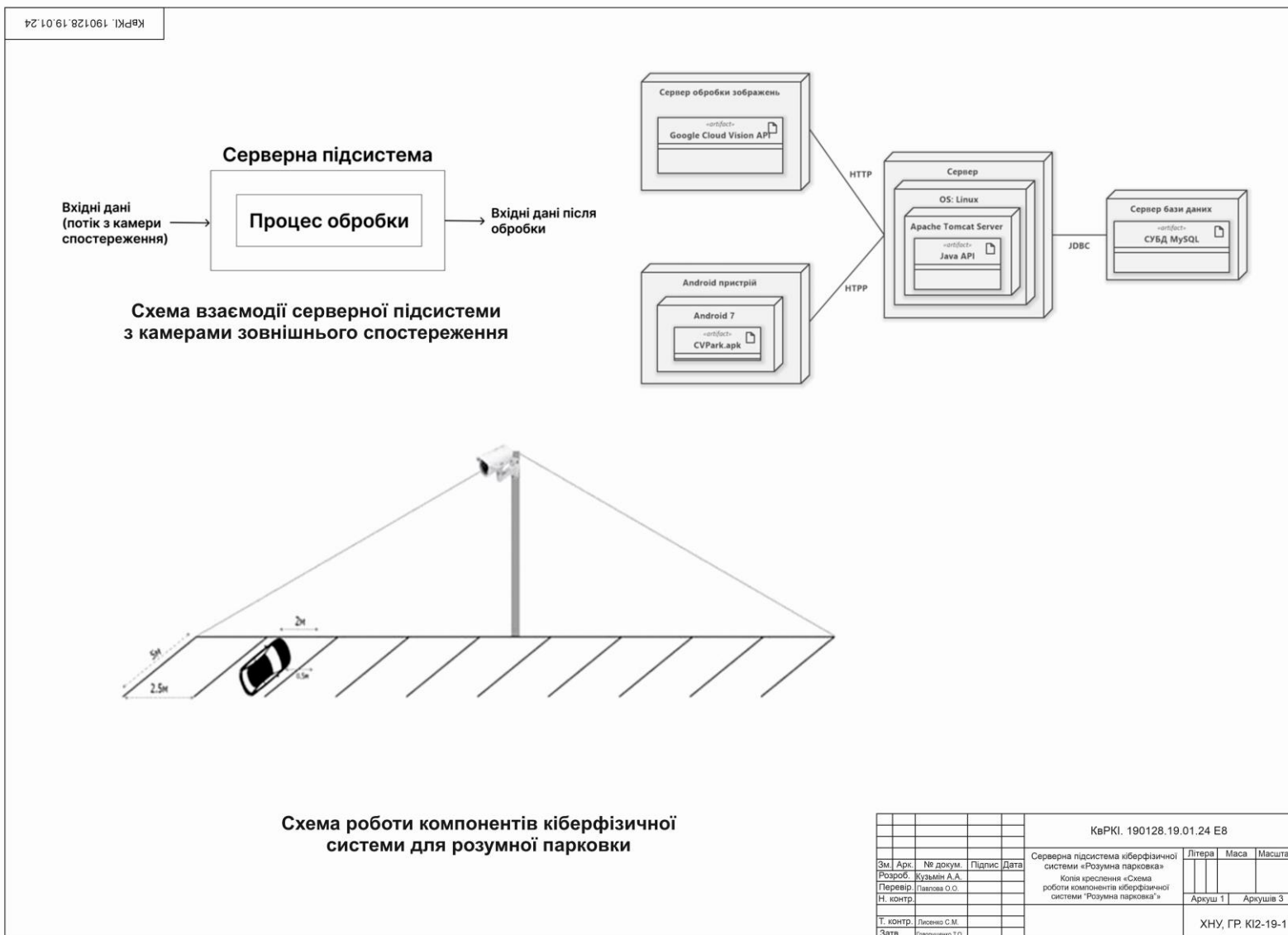
64. Zaitseva E., Levashenko V. Construction of a reliability structure function based on uncertain data. *IEEE Transactions on Reliability*. vol. 65, no. 4. 2016. pp. 1710 – 1723.

65. Zaitseva E., Levashenko V. Reliability analysis of multi-state system with application of multiple-valued logic. *International Journal of Quality and Reliability Management*. 2017. vol. 34, no. 6. P.862 – 878.

					КВРКІ 190128.19.01.24 ПЗ	Арк. 65
Зм.	Арк.	№ доквм.	Підпис	Дата		

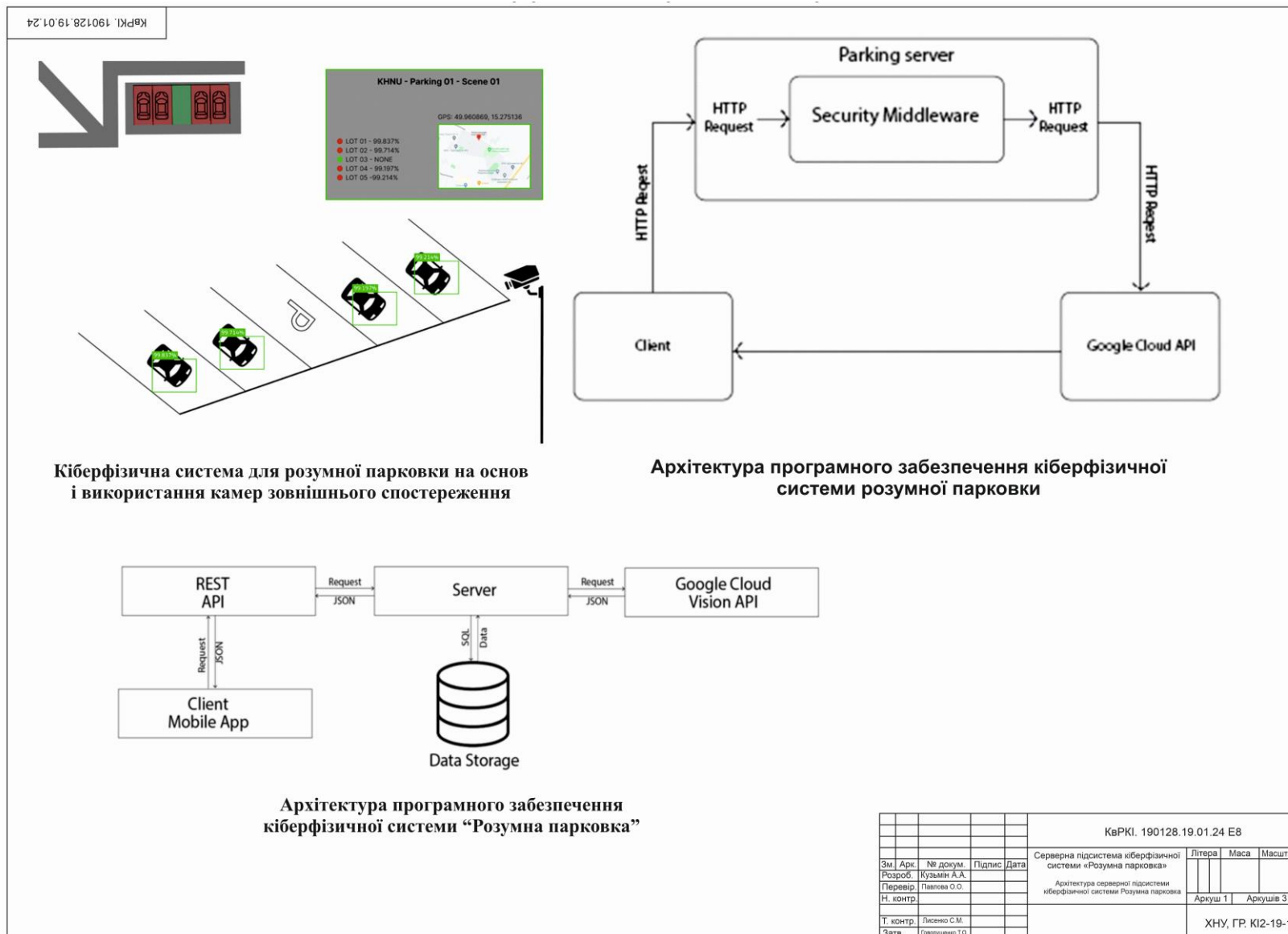
Додаток Б

Копія креслення «Схема роботи компонентів кіберфізичної системи Розумна парковка»



Додаток В

Копія креслення «Архітектура серверної підсистеми кіберфізичної системи Розумна парковка»



Додаток Г

Лістинг коду програмного забезпечення серверної підсистеми кіберфізичної системи “Розумна парковка”

Модуль обробки зображення з камери зовнішнього спостереження (мова програмування Python, фреймворк Flask)

```
from flask import Flask, abort
import cv2
import time
import numpy

app = Flask(__name__)

@app.route("/get_cam_photo/<rtsp_url>")
def imageroute():
    cap = cv2.VideoCapture(rtsp_url)

    if cap.isOpened():
        ret, frame = cap.read()

        encode_param = [int(cv2.IMWRITE_JPEG_QUALITY), 100]
        result, imgencode = cv2.imencode('.jpg', frame, encode_param)
        data = numpy.array(imgencode)
        stringData = data.tobytes()

        return jsonify({'image_data': stringData})

    else:
        abort(404, description="Camera is offline")

    cap.release()

if __name__ == "__main__":
    app.run()
```

Модуль встановлення зв'язку серверної підсистеми з REST API (мова програмування PHP фреймворк Laravel)
<?php

```
use App\Events\ParkMessage;
use App\Events\MyEvent;
use Illuminate\Support\Facades\Route;
use App\Models\Place;
use App\Models\Parking;
use App\Http\Controllers\ParkingController;

Route::get('/parkings', function () {

    $place = Place::all()->find(1);

    $s="";

    foreach ($place->Parkings as $parking)
    {
        $s.=$parking->description;
    }

    return $s;

});

Route::get('/test/{channel}/{event}', [ParkingController::class,'update']);

Route::get('/test2', function()
{
return response()->json(event(new MyEvent('hello')));
});
```

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорущенко Т. О.

Кузьмін Андрія Андрійовича

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-19-1

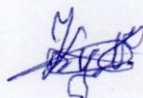
ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2023 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Серверна підсистема кіберфізичної системи «Розумна парковка»

Автор: Кузьмін Андрій Андрійович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Павлова Ольга Олександрівна, д.ф. ст.викладач

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 4.6% і адресується до 335 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС



О. О. Павлова

С. М. Лисенко

Т. О. Говорущенко

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 13%

| | | | | |
|---|----------|---------|-----------------------------|---------|
| ID: 112776
Назва: Серверна підсистема кіберфізичної системи «Розумна парковка»
Додано в БД: 2023-05-01
Автора: А. А. Кузьмін
Керівники: О. О. Павлова
Консультанти:
Опоненти: | Документ | | Сумарний збіг по Базі Даних | |
| | Символи | Лексеми | Символи | Лексеми |
| | 62748 | 495 | 1790 (3%) | 19 (4%) |

Джерело плагіату

| ID | Опис | Наявність плагіату в документі | |
|----|------|--------------------------------|---------|
| | | Символи | Лексеми |
| | | | |

Ім'я користувача:
Кафедра КІ

Дата перевірки:
01.05.2023 09:14:02 EEST

Дата звіту:
01.05.2023 09:17:55 EEST

ID перевірки:
1014863381

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Кузьмін_Серверна підсистема кіберфізичної системи «Розумна парковка»

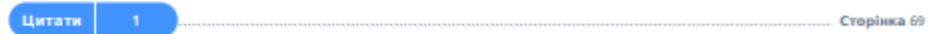
Кількість сторінок: 64 Кількість слів: 9012 Кількість символів: 71078 Розмір файлу: 9.34 MB ID файлу: 1014562924

14.2% Схожість

Найбільша схожість: 4.6% з Інтернет-джерелом (<http://elar.khmnu.edu.ua/jspui/bitstream/123456789/11951/1/%d0%94%>)



0.23% Цитат



Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

