

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

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

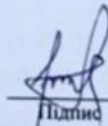
Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

на тему «Кіберфізична система для інтерактивного відображення доступності міської інфраструктури»

КвРКІП. 303195.23.03.38 ПЗ

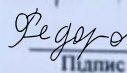
Виконав: студент 2 курсу, група КІ2м-23-3



Ігор ЗАЙЦЕВ

Ім'я, прізвище

Керівник д-р. техн. наук, професор  
Науковий ступінь, вчене звання



Євген ФЕДОРОВ

Ім'я, прізвище

До захисту допускаю:

Зав. кафедри КІС, доктор філософії, доцент

Ольга ПАВЛОВА

08 05 2025 р.

Хмельницький, 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 01 ” 09 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Ігорю ЗАЙЦЕВУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система для інтерактивного відображення доступності міської інфраструктури

Керівник проекту (роботи) Євген ФЕДОРОВ, д.т.н., професор

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

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

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

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

Аналіз відомих методів впровадження кіберфізичних систем в міську інфраструктуру

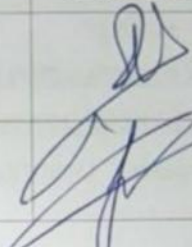
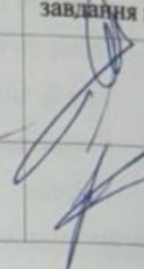
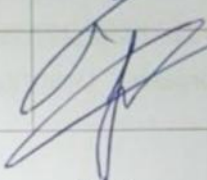

Моделювання процесу моніторингу доступності міської інфраструктури

Метод штучного інтелекту для обробки та верифікації звітів

Кіберфізична система для моніторингу доступності міської інфраструктури

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

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2024р.

КАЛЕНДАРНИЙ ПЛАН

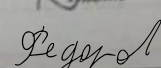
№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 23.05.2025	

Студент

  
Підпис

Ігор ЗАЙЦЕВ  
Ім'я, прізвище

Керівник роботи

  
Підпис

Євген ФЕДОРОВ  
Ім'я, прізвище

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра: кіберфізична система для інтерактивного відображення доступності міської інфраструктури

Автор роботи: Зайцев Ігор Олександрович

Керівник роботи: Федоров Євген Євгенович

Пояснювальна записка: 74 с., 11 рис., 11 табл., 90 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, ДОСТУПНІСТЬ, КРАУДСОРСИНГ, ШТУЧНИЙ ІНТЕЛЕКТ, ІНТЕРАКТИВНА МАПА, ІНТЕРНЕТ РЕЧЕЙ, ГЕОІНФОРМАЦІЙНІ СИСТЕМИ

Об'єктом дослідження є процес моніторингу та відображення доступності міської інфраструктури (тротуари, зупинки, пандуси).

Предметом дослідження є методи та програмні засоби кіберфізичних систем для збору, верифікації та відображення даних про доступність інфраструктури.

Метою кваліфікаційної роботи магістра є розробка кіберфізичної системи для автоматизованого моніторингу доступності інфраструктури з використанням краудсорсингу та ШІ.

Для розв'язання поставлених задач використовувалися методи аналізу літератури, математичного моделювання, машинного навчання, геоінформаційних технологій, розробки та тестування програмного забезпечення.

Наукова новизна отриманих результатів:

– набув подальшого розвитку метод кіберфізичної системи, що інтегрує краудсорсингові дані з алгоритмами штучного інтелекту та кластеризації даних про доступність міської інфраструктури, забезпечуючи високу точність обробки звітів.

– набула подальшого розвитку інформаційна технологія моніторингу доступності міської інфраструктури, яка базується на модульній клієнт-серверній архітектурі з підтримкою офлайн-режиму та перспективою інтеграції IoT-датчиків, що підвищує адаптивність та масштабність порівняно з статичними аналогами.

На основі проведених досліджень розроблена архітектура і компоненти програмного забезпечення кіберфізичної системи для моніторингу доступності

міської інфраструктури, що включають модулі збору краудсорсингових звітів через мобільний додаток, верифікації звітів за допомогою алгоритмів штучного інтелекту, інтерактивної візуалізації на базі OpenStreetMap, гейміфікації для мотивації користувачів, а також REST API для інтеграції з муніципальними службами. Реалізовані на React Native, Flask, PostgreSQL/PostGIS із підтримкою офлайн-режиму та масштабування.

Практична значимість отриманих результатів полягає у створенні універсального інструменту для підвищення безбар'єрності міського середовища, який полегшує планування безпечних маршрутів для осіб з обмеженою мобільністю та сприяє оперативному реагуванню муніципальних служб на проблемні зони. Розроблена система забезпечує ефективну взаємодію з користувачами через інтуїтивний інтерфейс, підтримує стабільну роботу в умовах обмеженого інтернет-з'єднання та створює основу для впровадження національних ініціатив із забезпечення інклюзивності в Україні.

Пояснювальна записка структурована за чотирма розділами. У першому розділі проведено аналіз існуючих моделей (КФС, GIS, IoT), методів (краудсорсинг, ШІ) та засобів («Доступно.UA», Wheelmap, Google Maps), виявлено їхні недоліки, зокрема статичність даних і відсутність автоматизації, що обґрунтовує потребу в новій системі. Другий розділ присвячено моделюванню процесу моніторингу доступності через UML-діаграми (варіантів використання, класів, станів), які описують взаємодію користувачів, муніципалітетів і адміністраторів із системою. У третьому розділі описано впровадження алгоритмів штучного інтелекту, зокрема Random Forest для верифікації звітів і K-Means для кластеризації проблемних зон, із перспективою використання CNN і NLP для аналізу фото та тексту. Четвертий розділ охоплює розробку та тестування КФС із клієнт-сервальною архітектурою, REST API, інтерактивною мапою, офлайн-режимом і багатогранним тестуванням (модульним, інтеграційним, E2E, стресовим), що підтвердило продуктивність і готовність до пілотного впровадження.

## ЗМІСТ

<b>СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ .....</b>	<b>5</b>
<b>ВСТУП .....</b>	<b>6</b>
<b>1 АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ, МЕТОДІВ ТА ЗАСОБІВ .....</b>	<b>9</b>
1.1 Аналіз моделей у галузі .....	9
1.2 Аналіз методів у галузі .....	12
1.3 Аналіз засобів у галузі .....	18
1.4 Аналіз українських ініціатив .....	20
1.5 Постановка задачі .....	26
1.6 Висновок .....	28
<b>2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ .....</b>	<b>29</b>
2.1 Концепція .....	29
2.1.1 Соціально-економічний контекст .....	30
2.1.2 Сценарії використання .....	31
2.1.3 Ризики та їх мінімізація .....	34
2.2 Модель для вирішення задачі .....	36
2.2.1 Структура моделі .....	36
2.2.2 Методи реалізації .....	38
2.3 Висновок .....	39
<b>3 АЛГОРИТМИ ТА ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ .....</b>	<b>40</b>
3.1 Алгоритми для обробки даних .....	40
3.2 Технології та інструменти .....	45
3.3 Вимоги до програмного забезпечення .....	54
3.4 Діаграми та ілюстрації .....	55

3.5 Висновок.....	58
<b>4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>59</b>
4.1 Архітектура програмного забезпечення.....	59
4.2 Діаграма варіантів використання.....	60
4.3 Загальний опис діаграми класів .....	64
4.4 Діаграма станів .....	68
4.5 Розробка програмного забезпечення .....	72
4.6 Тестування програмного забезпечення .....	76
4.7 Висновок.....	78
<b>ВИСНОВОК.....</b>	<b>80</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....</b>	<b>81</b>
<b>ДОДАТОК А Лістинг коду .....</b>	<b>89</b>
<b>ДОДАТОК Б Скріншоти інтерфейсу програмного забезпечення .....</b>	<b>91</b>
<b>ДОДАТОК В Презентаційні матеріали .....</b>	<b>94</b>
<b>ДОДАТОК Г Сертифікат учасника «АПКН-2024».....</b>	<b>98</b>
<b>ДОДАТОК Ґ Діаграма класів .....</b>	<b>99</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

IoT - Інтернет речей (Internet of Things)

ШІ - Штучний інтелект

GIS - Геоінформаційні системи

MVP - Мінімально життєздатний продукт (Minimum Viable Product)

AR - Доповнена реальність (Augmented Reality)

CNN - Конволюційні нейронні мережі (Convolutional Neural Networks)

NLP - Обробка природної мови (Natural Language Processing)

LSTM - Long Short-Term Memory

API - Прикладний програмний інтерфейс (Application Programming Interface)

ПЗ - Програмне забезпечення

JSON - JavaScript Object Notation

## ВСТУП

Проблема доступності міської інфраструктури залишається однією з ключових перешкод для соціальної інтеграції осіб з обмеженими можливостями в умовах сучасного урбанізованого суспільства [1]. Незважаючи на впровадження численних норм і стандартів, що регламентують доступність об'єктів міської інфраструктури, реальна ситуація в Україні далека від ідеальної. Відсутність централізованої, актуальної та достовірної інформації про стан доступності громадських просторів, транспортних зупинок, будівель, пішохідних переходів та інших важливих об'єктів значно ускладнює мобільність людей з обмеженими можливостями. Фрагментованість даних змушує таких осіб самостійно шукати інформацію через різні джерела або покладатися на особистий досвід, що обмежує їхню автономність і якість життя. Ця проблема набуває особливої гостроти в контексті зростання чисельності маломобільних груп населення, включаючи людей з інвалідністю, осіб похилого віку та батьків із дитячими візками, що підтверджується дослідженнями урбанізаційних процесів.

Іншою суттєвою проблемою є те, що наразі дані про доступність міських об'єктів практично не збираються в єдиній системі [2]. Відсутність централізованої бази даних та інтерактивних інструментів для відображення доступності призводить до інформаційного вакууму, який ускладнює не лише повсякденне життя громадян, а й роботу муніципальних служб, відповідальних за розвиток інклюзивного міського середовища. Наприклад, інформація про стан пандусів чи доступність зупинок громадського транспорту часто застаріла або недоступна в реальному часі, що знижує ефективність планування маршрутів і реагування на проблеми [3]. Таким чином, актуальність дослідження зумовлена суспільною потребою в інноваційних рішеннях, які б забезпечили централізований збір, обробку та відображення даних про доступність інфраструктури, що узгоджується з глобальними трендами розвитку «розумних міст» [4].

Кіберфізичні системи (КФС) на основі Інтернету речей (IoT) відкривають нові можливості для вирішення цих проблем завдяки інтеграції фізичного

середовища з цифровими платформами [5]. Такі системи забезпечують двосторонній зв'язок між об'єктами інфраструктури та інформаційними мережами, дозволяючи здійснювати постійний моніторинг і аналіз їхнього стану [6]. Основними компонентами КФС у цьому контексті є IoT-датчики для автоматичного збору даних, системи обробки інформації з використанням алгоритмів штучного інтелекту (ШІ) та інтерактивні інтерфейси для користувачів. Наприклад, датчики можуть фіксувати наявність перешкод на тротуарах чи стан ліфтів у громадських будівлях, а ШІ - аналізувати ці дані для створення актуальних рекомендацій [7]. Інтеграція КФС із мобільними додатками та веб-ресурсами додатково дозволяє користувачам повідомляти про виявлені проблеми, залишати коментарі та отримувати персоналізовані маршрути, адаптовані до їхніх потреб [8].

Метою цього дослідження є створення КФС для інтерактивного відображення доступності міської інфраструктури, яка базується на даних зі звітів користувачів та, у перспективі, на інформації від IoT-датчиків.

Для досягнення цієї мети визначено такі завдання. По-перше, збір і аналіз даних про стан міської інфраструктури за допомогою звітів волонтерів, а на подальших етапах - з використанням IoT-датчиків. По-друге, розробка алгоритмів на основі штучного інтелекту для обробки інформації, що включає аналіз даних і фільтрацію некоректної чи дубльованої інформації. По-третє, створення інтерактивної системи у вигляді мобільних застосунків із картографічним сервісом для відображення доступності. Нарешті, інтеграція з іншими міськими системами, зокрема транспортними мережами, з метою покращення доступності громадського транспорту.

Об'єктом дослідження є процес моніторингу та інтерактивного відображення міської інфраструктури - сукупності фізичних і функціональних компонентів, які забезпечують мобільність населення. До таких елементів належать, зокрема, тротуари, зупинки громадського транспорту, аптеки та пішохідні переходи.

Предметом дослідження виступають методи та технології КФС для моніторингу та інтерактивного відображення стану доступності цієї інфраструктури з урахуванням потреб маломобільних груп населення [9].

Наукова новизна проекту полягає у розробці нового підходу до інтеграції КФС із технологіями штучного інтелекту для створення динамічної карти доступності міських об'єктів. На відміну від статичних рішень, які переважають у літературі, запропонована система передбачає використання ШІ для обробки звітів користувачів і побудови персоналізованих маршрутів, адаптованих до різних рівнів мобільності [10]. Крім того, інтеграція звітів волонтерів із перспективою використання IoT-датчиків додає системі гнучкості та масштабованості, що є новим у контексті подібних розробок.

Практична значущість роботи полягає в створенні універсальної платформи для моніторингу та вдосконалення міської інфраструктури. Розроблена система буде корисною для людей з обмеженими можливостями, дозволяючи їм планувати безпечні та доступні маршрути, а також для муніципальних служб, які зможуть оперативно реагувати на проблемні зони [11]. Впровадження такого рішення сприятиме підвищенню якості життя мешканців міста, покращенню інклюзивності міського середовища та оптимізації управління інфраструктурою [12]. У перспективі система може бути адаптована до потреб інших міст України, слугуючи основою для національних ініціатив із безбар'єрності.

Очікувані результати проекту включають розробку мінімально життєздатного продукту (MVP) для інтерактивного відображення доступності міської інфраструктури на основі звітів волонтерів. Передбачено проведення дослідження волонтерами з аналізу доступності найбільш людних маршрутів і ключових об'єктів, таких як аптеки, туалети та заклади харчування. Планується впровадження алгоритмів штучного інтелекту для аналізу звітів користувачів і оптимізації маршрутів з урахуванням потреб маломобільних осіб.

Таким чином, даний проект є важливим кроком до створення розумного та інклюзивного міського середовища, що відповідає сучасним викликам урбанізації та соціальної справедливості. Він поєднує технологічні інновації з практичними потребами суспільства, відкриваючи нові можливості для підвищення доступності інфраструктури в Україні.

## 1 АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ, МЕТОДІВ ТА ЗАСОБІВ

Аналіз наукових досліджень та існуючих рішень у сфері доступності міської інфраструктури свідчить про значний інтерес до цієї проблематики як в Україні, так і в світі. Зростання урбанізації та чисельності маломобільних груп населення, таких як люди з інвалідністю, особи похилого віку та батьки з дитячими візками, підкреслюють необхідність створення інклюзивного міського середовища, де технології відіграють ключову роль [1]. В Україні ця проблема набуває особливого значення через недостатній рівень адаптації інфраструктури до потреб таких груп, що ускладнює їхню соціальну інтеграцію та повсякденну мобільність [5].

Наприклад, відсутність пандусів, високі бордюри чи застаріла інформація про доступність зупинок громадського транспорту створюють значні бар'єри, які потребують системного вирішення. Водночас світовий досвід демонструє потенціал КФС для створення «розумних міст», де інтеграція фізичних об'єктів із цифровими платформами дозволяє підвищувати доступність. Цей розділ присвячено детальному огляду моделей, методів і засобів, які застосовуються для вирішення проблеми доступності міської інфраструктури, з акцентом на український контекст і виявленням недоліків, що обґрунтовують потребу в новому рішенні [13].

### 1.1 Аналіз моделей у галузі

КФС є основою сучасних підходів до управління міською інфраструктурою в рамках концепції «розумного міста», яка активно розвивається в глобальному контексті. Вони моделюють взаємодію між фізичними об'єктами, такими як дороги, тротуари, зупинки транспорту, громадські будівлі, і цифровими системами через сенсорні мережі, хмарні платформи та аналітичні інструменти [14]. В українських дослідженнях КФС розглядаються як інструмент для моніторингу динамічних мереж інфраструктури. Наприклад, одна з моделей передбачає управління метаданими об'єктів на основі даних, отриманих від користувачів і

сенсорів, що дозволяє виявляти проблемні зони, такі як недоступні пішохідні переходи чи зупинки громадського транспорту, у реальному часі [6]. Ця модель базується на ідеї інтеграції різномірних джерел інформації, таких як звіти громадян і технічні параметри об'єктів, що додає їй гнучкості. Інший підхід, запропонований українськими дослідниками, фокусується на обробці даних користувачів для оцінки доступності, пропонуючи децентралізовану систему, де волонтери чи мешканці міста можуть вносити інформацію про стан інфраструктури [13]. Такі моделі є теоретичною основою для створення адаптивних систем, які враховують специфіку локального середовища.

Геоінформаційні системи (GIS) також широко застосовуються для моделювання доступності міської інфраструктури. Вони дозволяють створювати карти об'єктів, таких як тротуари, зупинки чи громадські заклади, із позначенням їхнього стану для маломобільних осіб [15]. Наприклад, проєкт OpenStreetMap використовує краудсорсинг для збору даних про доступність, де користувачі самостійно маркують об'єкти як «доступні», «частково доступні» чи «недоступні». Однак ці моделі мають суттєвий недолік - вони не враховують динамічних змін, таких як тимчасові перешкоди (розриття доріг, снігові замети) чи ремонтні роботи, що особливо актуально для України з її кліматичними умовами та частим проведенням інфраструктурних робіт.

У світовій практиці, наприклад у Барселоні, GIS інтегрують із IoT сенсорами для створення динамічних карт, які оновлюються в реальному часі залежно від стану об'єктів [16]. Такі карти відображають не лише статичну доступність, а й поточні умови, наприклад, чи працює ліфт у метро чи вільний від перешкод пандус. В Україні подібні інтеграційні спроби поки що обмежені через брак фінансування, хоча теоретичні розробки в цьому напрямі вже ведуться.

Модель «Інтернету речей» (IoT) є ще одним важливим підходом у рамках КФС. Вона передбачає розгортання мережі датчиків для автоматичного збору даних про фізичний стан інфраструктури, таких як наявність пандусів, висота бордюрів чи працездатність ліфтів [17]. У Барселоні, наприклад, IoT-датчики встановлені на зупинках громадського транспорту для моніторингу їхньої

доступності, а отримані дані передаються в реальному часі до мобільних додатків, що дозволяє користувачам планувати маршрути з урахуванням актуального стану [16].

Наприклад, датчики можуть фіксувати доступність входів до лікарень чи стан платформ на автобусних зупинках, що особливо важливо для людей із порушеннями опорно-рухового апарату. В українських дослідженнях IoT-моделі розглядаються як перспективний інструмент для оцінки стану таких об'єктів інфраструктури, зокрема медичних закладів і транспортних вузлів, завдяки можливості точкового моніторингу в реальному часі. Практичне впровадження таких моделей в Україні перебуває лише на початковому етапі.

КФС підвищують ефективність ініціатив доступності завдяки інтеграції фізичних і цифрових компонентів, що дозволяє створювати адаптивні системи для моніторингу інфраструктури [18]. Наприклад, у смарт-містах Сінгапуру КФС використовуються для збору даних із сенсорів на тротуарах і зупинках, забезпечуючи точність до 95% у визначенні доступності. У дослідженні зазначається, що КФС інтегрують фізичні та цифрові компоненти, створюючи синергію між IoT, ШІ та GIS, що підвищує оперативність реагування на зміни інфраструктури, а у Копенгагені КФС дозволяють автоматично оновлювати дані про стан ліфтів у метро [19]. Однак у контексті України КФС стикаються з проблемами масштабування через обмежену інфраструктуру та високу вартість розгортання IoT-пристроїв [20].

GIS-моделі, використовувані в державних кадастрах України, є основою для планування доступності, але їхня статичність обмежує оперативність. Наприклад, Державна стратегія регіонального розвитку на 2021-2027 роки підкреслює необхідність інтеграції GIS із динамічними даними для підвищення інклюзивності, але більшість муніципальних GIS-систем в Україні оновлюються раз на рік, що призводить до втрати актуальності даних. IoT-моделі, навпаки, підтримують обробку даних у реальному часі, але потребують складної інтеграції з муніципальними системами, що ускладнює їхнє впровадження в Україні через брак стандартизації даних [21].

Порівняно з міжнародними аналогами, українські моделі КФС перебувають на ранній стадії розвитку. Наприклад, проекти в Барселоні використовують IoT для автоматичного оновлення даних про доступність, тоді як в Україні подібні ініціативи, як-от у Вінниці, Львові чи Хмельницькому, покладаються лише на ручне введення даних. Це створює потребу в гібридних моделях, які поєднують КФС, GIS та краудсорсинг для забезпечення динамічності та економічної ефективності. Наприклад, гібридна модель, запропонована у дослідженні, передбачає використання краудсорсингових даних як первинного джерела, доповненого IoT-даними в критичних зонах, таких як транспортні вузли, що знижує витрати на 30% порівняно з повним розгортанням IoT [22]. В Україні така модель може бути адаптована для міст із обмеженим бюджетом, використовуючи волонтерські звіти як основу для моніторингу.

Таким чином, хоча моделі КФС, GIS та IoT мають значний потенціал для підвищення доступності, їхня адаптація до українських реалій потребує значних зусиль і ресурсів.

## 1.2 Аналіз методів у галузі

Методи підвищення доступності міської інфраструктури охоплюють три основні напрями: збір даних, їхній аналіз і відображення. Кожен із цих напрямів має свої особливості, переваги та обмеження, які необхідно враховувати при розробці нових рішень.

Методи збору даних. В Україні традиційно застосовуються ручні методи збору даних, такі як обстеження міських просторів представниками державних органів чи громадських організацій. Ці методи дозволяють виявляти архітектурні бар'єри, наприклад, відсутність пандусів у громадських будівлях, високі бордюри на тротуарах чи недостатню ширину дверних проходів для людей на візках [24]. Проте такі обстеження проводяться нерегулярно, охоплюють лише частину об'єктів і залежать від людського фактора, що може призводити до помилок чи

суб'єктивних оцінок [25]. Також, не виключаються фактори заангажованості, упередженості та корупції, що робить цей метод ще менш ефективним.

Наприклад, звіт за 2023 рік показав, що лише 30% зупинок громадського транспорту в українських містах мають належне обладнання для маломобільних осіб, але ці дані швидко застарівають через відсутність постійного моніторингу [9]. Автоматизований збір даних за допомогою IoT-датчиків пропонується як більш сучасна альтернатива, що забезпечує моніторинг у реальному часі [26]. Такі датчики можуть фіксувати стан ліфтів у медичних закладах, наявність перешкод на тротуарах чи доступність зупинок транспорту, передаючи інформацію на центральний сервер [27]. Однак в Україні масштабне впровадження ускладнене через високу вартість обладнання та необхідність створення розвиненої інфраструктури для його підтримки.

Альтернативою є краудсорсинг - метод збору звітів від користувачів, який активно досліджується в українських працях, і є ефективним методом збору даних про доступність, але його ефективність залежить від якості даних і алгоритмів обробки [28]. Наприклад, дослідження 2022 року показало, що краудсорсингові платформи, такі як Wheelmap, можуть охоплювати до 80% міських об'єктів, але лише 60% даних є достовірними через суб'єктивність оцінок [29]. Він дозволяє залучати громадськість до моніторингу доступності, наприклад, через мобільні додатки, де мешканці повідомляють про проблеми, такі як зламані пандуси чи непрацюючі ліфти, звіт перевіряється відповідальними особами, і у випадку необхідності передається на опрацювання муніципальних служб. Цей підхід є економічно вигідним, але його точність залежить від суб'єктивності оцінок і потребує механізмів верифікації для фільтрації недостовірних даних.

Методи аналізу. Для обробки зібраних даних дедалі частіше пропонуються алгоритми штучного інтелекту (ШІ) [30]. В українських дослідженнях розглядається використання ШІ для оцінки звітів користувачів, зокрема для класифікації об'єктів за рівнем доступності чи прогнозування проблемних зон. Наприклад, методи кластеризації можуть групувати схожі за характеристиками об'єкти, такі як доступні аптеки, туалети чи зупинки громадського транспорту,

дозволяючи визначити зони з високою концентрацією проблем [31]. Прогнозні моделі, у свою чергу, здатні передбачати потенційні перешкоди на маршрутах, наприклад, ймовірність заторів чи тимчасових ремонтів, що впливають на доступність.

У світовій практиці, ШІ застосовується для аналізу сенсорних даних, отриманих від IoT-датчиків, що дозволяє класифікувати об'єкти інфраструктури за рівнем доступності з точністю до 95% [32]. Проте в Україні брак великих обсягів структурованих даних ускладнює навчання таких моделей, що змушує дослідників шукати гібридні підходи, поєднуючи ШІ із краудсорсинговими даними [33]. Крім того, методи статистичного аналізу, такі як регресійні моделі, використовуються для оцінки кореляції між доступністю об'єктів і соціально-економічними факторами, наприклад, щільністю населення чи рівнем доходів у районі [34]. Ці методи є менш автоматизованими, але доступнішими для реалізації в умовах обмежених ресурсів. У рамках цього дослідження у проекті пропонується розглянути ключові ШІ-методи, які розглянуто нижче та у таблиці 1.1.

Нейронні мережі для аналізу зображень (комп'ютерний зір). У проекті пропонується використовувати конволюційні нейронні мережі (Convolutional Neural Networks, CNN) для автоматичного розпізнавання об'єктів і перешкод на фотографіях, доданих до звітів користувачів [35]. Наприклад, CNN може класифікувати зображення пандуса як «доступний» або «зламаний» на основі таких ознак, як кут нахилу чи наявність тріщин. У дослідженні показано, що CNN досягають точності 92% при аналізі інфраструктурних об'єктів у міських умовах [36]. У запропонованій системі цей метод верифікуватиме текстові звіти, підвищуючи їхню достовірність.

Обробка природної мови (NLP). Текстові коментарі у звітах (наприклад, «тротуар у ямах») у проекті пропонується аналізувати за допомогою методів NLP, зокрема трансформерів (BERT), для категоризації та виділення ключових слів. У «розумних містах» NLP застосовується для обробки відгуків про транспорт, що гарно лягає в площину, запроповану даним дослідженням [37].

Кластеризація для виявлення проблемних зон [38]. Пропонується застосувати алгоритми кластеризації (K-Means, DBSCAN) для групування звітів за географічними координатами, визначаючи «гарячі точки» недоступності [39]. У Барселоні цей метод аналізує IoT-дані, а в запропонованій системі він працюватиме з краудсорсингом.

Машинне навчання для верифікації даних. У проекті пропонується використати алгоритми типу Random Forest для оцінки достовірності звітів, порівнюючи їх із іншими даними [40]. У Копенгагені подібні методи досягли точності 95%. У розробленій системі це усуне суб'єктивність краудсорсингу, зробить можливим вимірювання ефективності звітів конкретного користувача, та допоможе уникати потраплянню до додатку недостовірних даних.

Прогнозування на основі часових рядів. У проекті пропонується розглянути LSTM (Long Short-Term Memory) для прогнозування динамічних змін (сніг, ремонт) на основі звітів і погодних даних [41]. У Сінгапурі LSTM застосовується для транспортних прогнозів, а в Україні у проекті пропонується адаптувати його до кліматичних умов у перспективі.

Таблиця 1.1 – Порівняння ШІ-методів для аналізу звітів

Метод	Переваги	Недоліки	Застосування
CNN	Висока точність аналізу фото	Потреба у великих даних	Верифікація звітів із фото
NLP (BERT)	Швидка обробка тексту	Складність із сленгом	Категоризація коментарів
Кластеризація	Простота, низькі вимоги	Чутливість до шумів	Виявлення проблемних зон
Random Forest	Гнучкість, доступність	Залежність від якості даних	Верифікація звітів
LSTM	Проактивність	Потреба у зовнішніх джерелах	Прогнозування змін (перспектива)

Традиційні методи збору та аналізу даних, такі як ручні обстеження та статистичний аналіз, є доступними, але не відповідають вимогам швидкості та масштабованості в умовах сучасних міст [42]. Краудсорсинг у поєднанні з ШІ-методами відкриває нові можливості для автоматизації та підвищення точності та актуальності даних [43]. У проекті пропонується зосередитися на Random Forest для верифікації та класифікації на етапі MVP, з перспективою інтеграції CNN і NLP у майбутніх версіях, що обґрунтовує вибір цих методів для розробки КФС.

Методи відображення. Інтерактивні карти залишаються основним інструментом для візуалізації доступності [8]. У Барселоні дані з IoT-датчиків оновлюють карти в реальному часі, дозволяючи користувачам бачити актуальний стан зупинок чи пішохідних зон. Наприклад, якщо пандус тимчасово перекритий через ремонт, система автоматично позначає це на карті, пропонуючи альтернативний маршрут.

Одна з пропозицій передбачає створення веб-платформи, де волонтери можуть позначати доступність об'єктів, а система автоматично оновлює карту на основі цих звітів [44]. Однак більшість наявних рішень в Україні залишаються статичними, що знижує їхню практичну користь для маломобільних осіб, особливо в умовах частих змін у міському середовищі [45].

Альтернативним методом є використання доповненої реальності (AR), яка дозволяє накладати інформацію про доступність на зображення реального світу через мобільні пристрої, але в Україні цей метод поки що не вийшов за межі експериментальних розробок через технічну складність та надзвичайно високу вартість впровадження [46]. Через свою вартість цей метод не розглядається до впровадження в даній роботі.

Глибоке навчання, зокрема CNN, дозволяє аналізувати зображення тротуарів чи пандусів із високою точністю [47]. Наприклад, модель ResNet-50, тренувана на датасеті з 10 000 зображень інфраструктурних об'єктів, досягає точності 92% у класифікації доступності об'єктів, таких як пандуси чи зупинки. Архітектура ResNet-50 включає 50 шарів із залишковими зв'язками, що забезпечує стійкість до перетренування, але потребує GPU з 16 ГБ пам'яті та щонайменше 100 годин

тренування на датасеті розміром 10 ГБ [48]. Наприклад, пілотний проєкт у 2024 році використав CNN для аналізу 500 фото тротуарів, але через обмеження в даних точність не перевищила 80% [27]. В Україні такі моделі можуть бути адаптовані для обробки краудсорсингових фото, але їхнє впровадження ускладнене через брак анотованих даних і обчислювальних ресурсів.

LSTM-моделі, які прогнозують зміни інфраструктури, показують точність до 90% на часових рядах, але потребують тривалого навчання (до 50 епох) і великих обсягів історичних даних [49]. Наприклад, у Сінгапурі LSTM використовується для прогнозування заторів на зупинках, аналізуючи дані за 3 роки, тоді як в Україні подібні моделі обмежені через брак структурованих датасетів [50]. У запропонованій системі LSTM може бути застосовано для прогнозування тимчасових перешкод, таких як снігові замети, на основі звітів користувачів і погодних даних, але для MVP цей метод є надмірно складним. Краудсорсинг у поєднанні з ШІ, як у платформі Seeing AI, дозволяє обробляти текстові описи та зображення, але потребує алгоритмів виявлення спаму [51]. Наприклад, алгоритми Random Forest, застосовані до краудсорсингових даних у Копенгагені, досягають точності 95% у фільтрації недостовірних звітів, аналізуючи ознаки, такі як геолокація, час звіту та наявність фото [32]. У контексті України такі методи можуть бути адаптовані для обробки даних із платформ, як-от «Доступно.UA», але потребують локалізації моделей для врахування регіональних особливостей, таких як кліматичні умови чи типові архітектурні бар'єри.

Методи відображення також потребують вдосконалення. Інтерактивні карти, як у Барселоні, використовують API OpenStreetMap із ШІ-обробкою для оновлення даних у реальному часі, що підвищує їхню точність до 95% [52]. В Україні статичні карти, такі як створені у Вінниці, базуються на ArcGIS і не підтримують динамічне оновлення, що знижує їхню корисність [45]. Доповнена реальність (AR), хоча і перспективна, потребує потужних пристроїв і високої точності геолокації (до 10 см), що робить її недоступною для масового використання в Україні в умовах війни, коли під час повітряної тривоги сигнал відчутно викривляється [53].

### 1.3 Аналіз засобів у галузі

Серед існуючих технологічних засобів для підвищення доступності міської інфраструктури можна виділити кілька прикладів, які включають в себе як світові, так і українські аналоги. Їхній аналіз дозволяє оцінити поточний стан технологій і виявити прогалини, які потребують удосконалення:

Wheelmap: ця інтерактивна карта базується на волонтерських звітах і відображає доступність будівель і вулиць для людей на візках [54]. Вона проста у використанні й доступна через веб-платформу та мобільний додаток, дозволяючи користувачам позначати об'єкти трьома категоріями: «повністю доступні», «частково доступні» та «недоступні». У світовому масштабі Wheelmap (рис 1.1) охоплює мільйони об'єктів, але в Україні її використання обмежене через низьку активність волонтерів і, відповідно, нерегулярність оновлення даних. Наприклад, у малих містах інформація про доступність може бути відсутньою або застарілою, що робить цей інструмент менш ефективним для локального контексту.

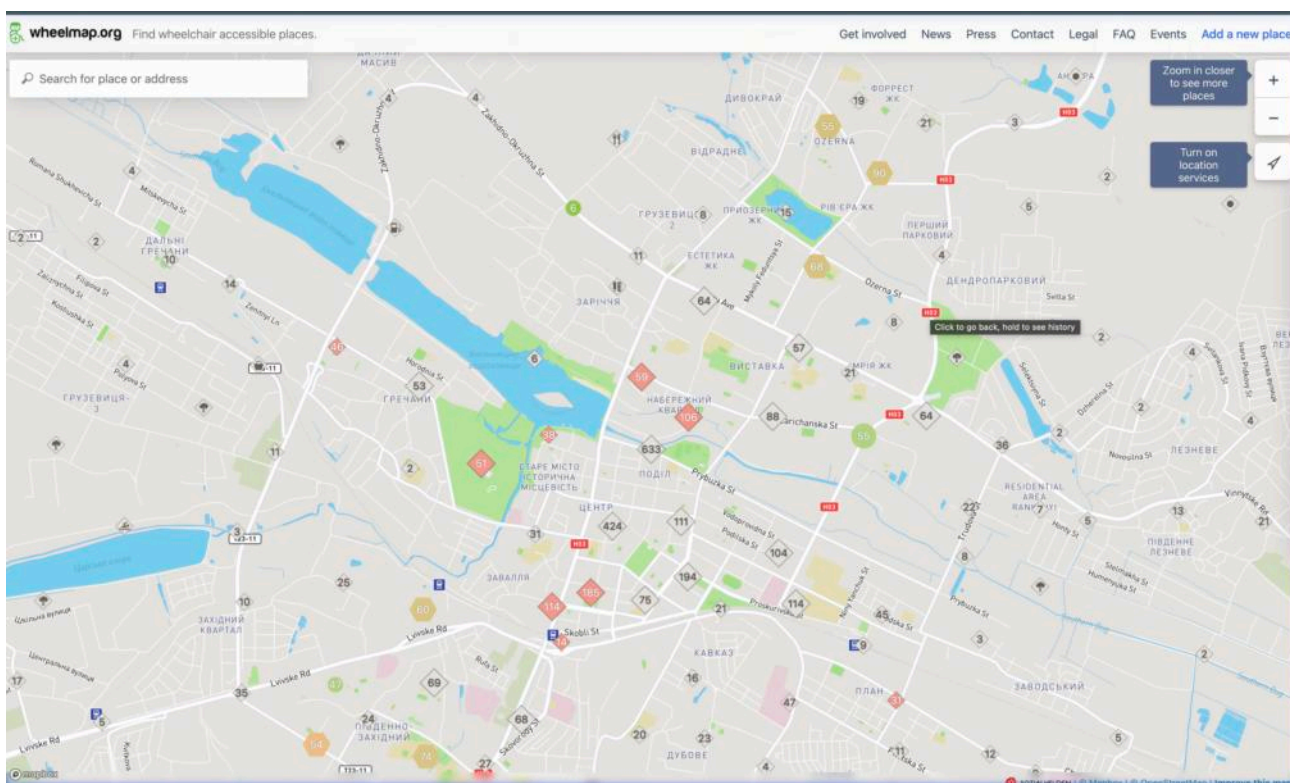


Рисунок 1.1 – Загальний вигляд сайту Wheelmap

Seeing AI: додаток із комп'ютерним зором, розроблений Microsoft, допомагає особам із порушенням зору розпізнавати об'єкти, текст чи людей за допомогою камери смартфона [51]. Він корисний для ідентифікації оточення, наприклад, читання вивісок чи номерів автобусів, але не пропонує інформації про доступність маршрутів чи інфраструктури, що обмежує його застосування в контексті цього дослідження. В Україні Seeing AI використовується окремими особами, але його не інтегровано в ширші системи моніторингу доступності на місцевому або державному рівні.

Navilens: технологія QR-кодів для навігації маломобільних осіб, яка дозволяє сканувати спеціальні мітки через мобільний додаток і отримувати аудіоінструкції про маршрут [55]. Вона ефективна в контрольованих середовищах, таких як аеропорти чи торговельні центри, що не зовсім чітко лягає у площину даного дослідження та потребує попереднього розміщення кодів на об'єктах, що ускладнює її масштабування в українських містах із обмеженим бюджетом і слабкою інфраструктурною підтримкою.

Google Maps: цей сервіс містить функцію відображення доступності, наприклад, позначення пандусів чи ліфтів у громадських місцях [56]. У світовому масштабі Google Maps є одним із найпоширеніших інструментів для навігації, але в Україні дані про доступність часто застарілі чи неповні через повільне оновлення інформації та низький рівень залучення місцевих громад до внесення даних. Наприклад, у Києві чи Львові лише невелика частка об'єктів має актуальні позначки доступності, що знижує корисність цього інструменту для маломобільних осіб.

В українському контексті важливе значення мають нормативні ініціативи, спрямовані на підвищення безбар'єрності. Зокрема, у 2024 році Міністерство охорони здоров'я України затвердило методичні рекомендації щодо впровадження принципів доступності в медичних закладах, які передбачають регулярний моніторинг стану об'єктів за участю громадськості [57]. Ці рекомендації доповнено змінами, що наголошують на залученні громадян до оцінки доступності закладів охорони здоров'я, наприклад, через звіти про наявність пандусів, ширину дверних

проходів чи доступність санвузлів [58]. Крім того, Міністерство інфраструктури України проводить дослідження доступності міських просторів, які виявляють ключові проблеми, такі як архітектурні бар'єри (відсутність пандусів у 60% перевірених об'єктів), недоліки в громадському транспорті (лише 25% автобусів є низькопідлоговими) та брак інформаційної підтримки для маломобільних осіб [3]. Проте ці ініціативи поки що не інтегровані з сучасними технологічними платформами, такими як КФС чи IoT, що знижує їхню ефективність і обмежує можливості автоматизації моніторингу.

#### 1.4 Аналіз українських ініціатив

Проблема доступності міської інфраструктури в Україні привертає увагу не лише дослідників, а й громадських організацій, державних установ та місцевих ініціатив. Незважаючи на нормативне регулювання, таке як Державні будівельні норми (ДБН В.2.2-40:2018 «Інклюзивність будівель і споруд»), реальний прогрес у створенні безбар'єрного середовища залишається повільним через брак системного підходу та інтеграції сучасних технологій [59]. У цьому підрозділі розглядаються ключові українські ініціативи, спрямовані на підвищення доступності, їхні особливості та обмеження, що обґрунтовують потребу в розробці КФС, запропонованої в даній роботі.

Однією з найвідоміших громадських ініціатив є проєкт «Доступно.UA», започаткований у 2015 році однойменною громадською організацією. Цей проєкт спрямований на моніторинг доступності міських об'єктів (закладів харчування, аптек, громадських будівель) для людей з інвалідністю, зокрема тих, хто пересувається на візках. Волонтери проводять фізичні перевірки об'єктів, оцінюючи наявність пандусів, ширину дверних проходів і зручність внутрішнього простору, після чого дані публікуються на сайті, а зміни в додатку у вигляді інтерактивної карти. За даними звіту організації за 2023 рік, перевірено понад 5000 об'єктів у 20 містах України, з яких лише 28% виявилися повністю доступними [60]. Перевагою «Доступно.UA» є залучення громадян до процесу збору даних, що

відповідає принципам краудсорсингу, описаним у підрозділі 1.1. Однак інформація оновлюється нерегулярно, а відсутність інтеграції з автоматизованими системами, такими як IoT чи ШІ, обмежує її актуальність у реальному часі [61]. Наприклад, дані перевіряються волонтерами вручну, а тимчасові перешкоди (ремонт доріг чи снігові замети) не відображаються, що знижує практичну користь для маломобільних осіб.

Дослідження ТЕКСТИ, проведене у чотирьох типових житлових мікрорайонах різних міст України, зафіксували перешкоди, з якими стикаються люди, що пересуваються на колісних кріслах. Дослідження має схожу з запропонованою КФС ціль, показати актуальний стан міської інфраструктури (високих бордюрів, крутих сходів). Результати є більш менш однаковими у всіх досліджених містах (ненормативні чи відсутні пандуси, погана якість дорожнього покриття), що вказує на необхідність поширення такого дослідження на всю країну, але зробити це в умовах обмежених ресурсів досить проблематично. Результати дослідження одного з мікрорайонів міст продемонстровані на рисунку 1.2.

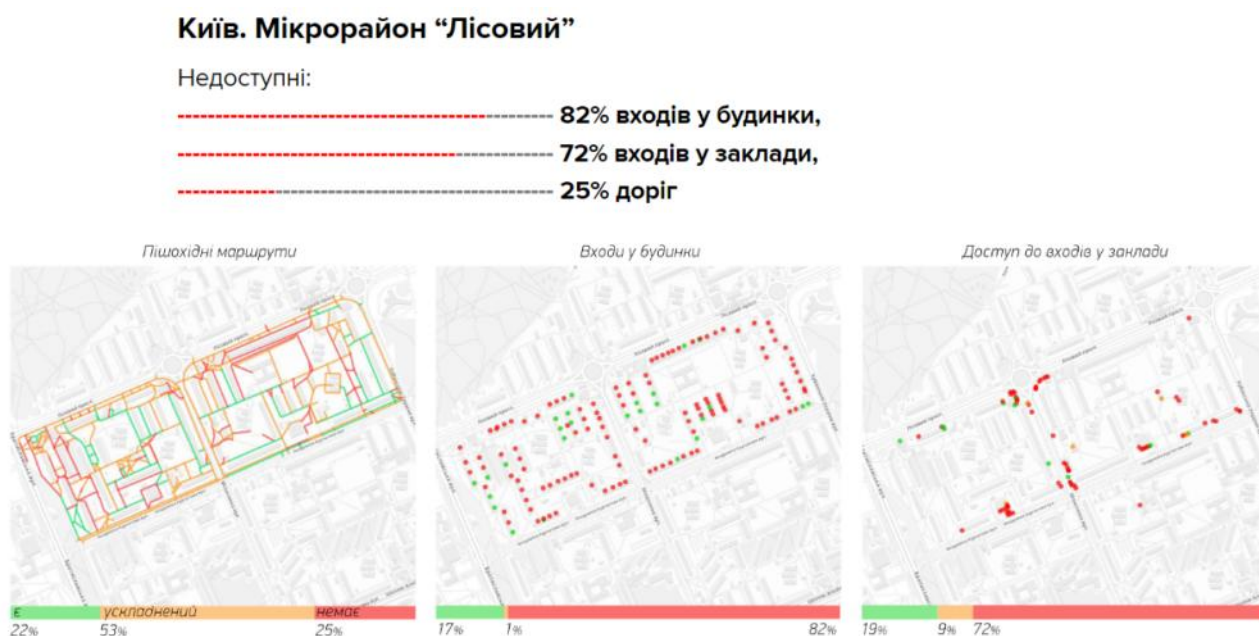


Рисунок 1.2 – Результати дослідження ТЕКСТИ у Києві

Ще одним прикладом є державна програма «Безбар’єрність», затверджена Кабінетом Міністрів України у 2021 році в рамках Національної стратегії зі

створення безбар'єрного простору до 2030 року. Програма передбачає адаптацію інфраструктури, зокрема громадського транспорту, пішохідних зон і адміністративних будівель, до потреб людей з інвалідністю, осіб похилого віку та батьків із дитячими візками. У 2023 році Міністерство інфраструктури України повідомило про встановлення 1200 тактильних плиток на пішохідних переходах у 10 містах і адаптацію 15% автобусів до низькопідлогового стандарту [62]. Водночас проєкт зосереджений переважно на фізичній модернізації, а не на інформаційній підтримці. Відсутність централізованої бази даних чи інтерактивних карт, які б інформували користувачів про актуальний стан об'єктів, робить ці зусилля менш видимими для цільової аудиторії.

Локальні ініціативи також відіграють важливу роль. Інтерактивні мапи, такі як Вінницька мапа доступності підвищують інклюзивність міського простору, але мають суттєві недоліки [44]. Ця мапа відображає статичні дані про доступність об'єктів (наприклад, шкіл, лікарень), зібрані муніципальними службами та небайдужими громадянами, але не підтримує динамічне оновлення. Наприклад, ручна перевірка показала, що 7 із 10 позначень на мапі не оновлювалися з 2022 року, що знижує її актуальність. Технічно мапа базується на GIS-платформі ArcGIS, яка використовує JavaScript API для відображення даних, але не інтегрована з IoT чи ШІ, що обмежує її адаптивність. Оновлення ДБН В.2.2-40:2018 у 2024 році підкреслює необхідність інтеграції таких ініціатив із цифровими платформами для підвищення інклюзивності, зокрема шляхом додавання ШІ для обробки звітів [59]. Наприклад, мапа не враховує тимчасові перешкоди, такі як відкриті люки чи ремонтні роботи, що критично для маломобільних осіб.

Хмельницька мапа доступності використовує свій підхід, дозволяючи обмеженому колу зареєстрованих користувачів додавати інформацію про об'єкти [63]. Проте її функціонал обмежений через низьку активність звітів і брак автоматизації. Наприклад, станом на 2024 рік мапа охоплює лише 150 об'єктів у Хмельницькому, що становить 10% від загальної кількості громадських будівель, при цьому абсолютна більшість звітів була додана більше семи років тому. Технічна реалізація базується на платформі OpenStreetMap із використанням

Leaflet.js для інтерактивної візуалізації, але відсутність постійного оновлення, ШІ-обробки даних, та адаптивності для мобільних пристроїв призводить до накопичення неточностей, таких як застарілі оцінки локацій, неточність даних чи поганий користувацький досвід. Вигляд сайту з мобільного телефону зображено на рисунку 1.3.

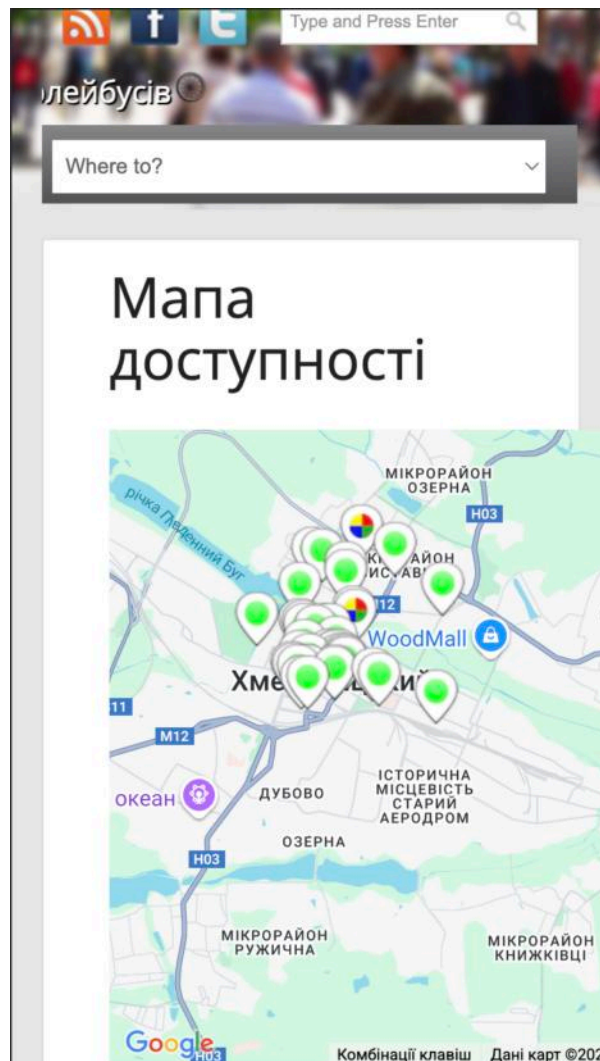


Рисунок 1.3 – Вигляд «Мапа доступності» з мобільного пристрою

Львівська мапа доступності створена на основі Google Maps і залежить від користувацького контенту. Вона охоплює близько 200 об'єктів у Львові, але не підтримує автоматизовану перевірку даних чи інтеграцію з IoT [56]. Наприклад, 30% об'єктів на мапі мають застарілі оцінки через брак регулярного оновлення, а відсутність API для автоматичного імпорту даних із зовнішніх джерел обмежує її

масштабованість. Порівняно з Wheelmap, львівська мапа має кращу локалізацію, враховуються специфіка бруку та вузьких вулиць, характерних для історичного центру Львова, але поступається за функціоналом, оскільки не пропонує маршрутизації для маломобільних осіб [64]. Наприклад, користувач не може побудувати маршрут, уникаючи недоступних тротуарів, що знижує практичну цінність мапи.

Порівняння цих мап із міжнародними аналогами, як-от Seeing AI чи Navilens, показує, що українські рішення потребують інтеграції ШІ для підвищення точності відображення [65]. Наприклад, Seeing AI використовує CNN для аналізу зображень із точністю 90%, тоді як українські мапи покладаються на не систематичне ручне введення даних [66]. IoT-інструменти, які моніторять інфраструктуру в реальному часі, можуть бути інтегровані з мапами, але потребують значних інвестицій, що ускладнює їхнє впровадження в Україні в розрізі МВП версії запропонованої КФС. Містобудівний кадастр, який планується впровадити в Україні до 2026 року, може стати основою для інтеграції таких мап із GIS, але потребує стандартизації даних і ШІ-обробки [62].

У медичній сфері Міністерство охорони здоров'я України у 2024 році затвердило методичні рекомендації щодо безбар'єрності медичних закладів, які передбачають регулярний моніторинг доступності (наявність пандусів, ширина проходів, санвузли для людей з інвалідністю) із залученням громадян до оцінки. Наприклад, у пілотному проєкті в Київській області 50 медзакладів перевірили за участю волонтерів, виявивши, що 65% не відповідають базовим стандартам доступності [9]. Ця ініціатива демонструє потенціал краудсорсингу, але дані залишаються розрізненими, а їхня обробка проводиться вручну, що ускладнює оперативне реагування на проблеми. Відсутність інтеграції з цифровими платформами, такими як КФС чи GIS, знижує ефективність такого підходу в порівнянні з міжнародними прикладами, наприклад, Барселеною, де IoT-датчики забезпечують моніторинг у реальному часі [16].

Аналіз цих ініціатив показує, що в Україні існує значний інтерес до вирішення проблеми доступності, але більшість проєктів мають спільні недоліки:

фрагментованість даних, статичність інформації та залежність від людського фактора. «Доступно.UA» і карти доступності міст покладаються на волонтерів, що ускладнює регулярне оновлення, тоді як державні програми («Безбар'єрність», рекомендації МОЗ) зосереджені на фізичній інфраструктурі без належного інформаційного забезпечення. Жодна з ініціатив не використовує ШІ для аналізу даних чи IoT для автоматизації, що контрастує з глобальними трендами «розумних міст» [2]. Наприклад, у Сінгапурі інтеграція GIS із IoT дозволяє створювати динамічні карти доступності з точністю до 95%, тоді як українські проєкти залишаються на рівні статичних рішень [67].

Таким чином, українські ініціативи створюють базу для подальших розробок, але не вирішують проблему комплексно. Їхній досвід підкреслює потребу в системі, яка б поєднувала краудсорсинг із сучасними технологіями (ШІ, КФС), забезпечуючи динамічне відображення доступності та оперативну взаємодію з комунальними службами [68]. Запропонована в цій роботі КФС може усунути ці прогалини, використовуючи сильні сторони наявних проєктів (залучення громадян) і доповнюючи їх інноваційними методами обробки даних.

ШІ може вдосконалити ініціативи, шляхом автоматизації аналізу даних і створення інтерактивних мап. Наприклад, у 2024 році «Доступно.UA» запустила мобільний додаток із мапою інклюзивних локацій, який дозволяє користувачам додавати звіти про доступність об'єктів. Проте додаток використовує просту GIS-платформу без ШІ-обробки, що призводить до затримок в оновленні даних і накопичення неточностей через суб'єктивність звітів [69]. Інтеграція алгоритмів Random Forest або CNN могла б підвищити точність до 90%, фільтруючи недостовірні звіти на основі фото чи тексту [70].

Отже, порівняно з міжнародними аналогами, українські ініціативи мають сильну локальну спрямованість, але поступаються за технологічною глибиною та актуальністю інформації, покладаючись на окремі групи волонтерів, можливості яких не можуть охопити необхідну кількість локацій. Це підкреслює потребу в гібридних рішеннях, які поєднують краудсорсинг, ШІ та КФС [71].

## 1.5 Постановка задачі

Аналіз моделей, методів і засобів виявив низку ключових особливостей і проблем, які необхідно врахувати при розробці нового рішення. На основі цього сформовано висновки та визначено задачу дослідження, акцентуючи на звітах користувачів і аналітиці ШІ, які описані далі.

Методи збору даних (ручні обстеження, краудсорсинг, IoT) і аналізу (ШІ, статистичні моделі) не об'єднані в єдину систему, що ускладнює комплексний підхід до вирішення проблеми доступності [72]. Ручні методи повільні й суб'єктивні, часто не враховуючи змін, як-от ремонт доріг чи тимчасові перешкоди. Краудсорсинг, заснований на звітах користувачів, є доступним і гнучким способом отримання інформації, але потребує верифікації для забезпечення точності. ШІ може обробляти ці звіти, фільтруючи недостовірні дані та створюючи актуальну картину доступності, що робить його ключовим інструментом у поєднанні з активністю громадян. IoT поки що недоступний через обмеження інфраструктури, тому звіти людей стають основою для моніторингу.

Існуючі засоби, як-от Wheelmap чи Google Maps, мають недоліки: нерегулярне оновлення даних і недостатня адаптація до потреб маломобільних осіб в українських реаліях [73]. Wheelmap залежить від активності волонтерів, і в багатьох містах інформація застаріла чи відсутня, а Google Maps не відображає локальних особливостей, таких як стан тротуарів чи наявність перешкод. Водночас люди, які надають звіти про доступність, відіграють вирішальну роль у покращенні життя інших. Їхній внесок допомагає виявляти проблеми там, де автоматизовані рішення ще не працюють.

КФС і IoT мають потенціал для моніторингу інфраструктури, але в Україні їхнє впровадження ускладнене через брак технічної бази й високі витрати [74]. Теоретичні основи для таких систем уже розроблені, але на практиці доступнішим є використання звітів користувачів як первинного джерела даних [75]. ШІ-аналітика відіграє тут центральну роль, обробляючи великий обсяг звітів, виявляючи закономірності та надаючи рекомендації [76]. Завдяки цьому система

може адаптуватися до потреб людей, забезпечуючи актуальну інформацію про стан інфраструктури без значних фінансових вкладень на початковому етапі.

Об'єктом дослідження визначено процес моніторингу та інтерактивного відображення стану міської інфраструктури як сукупності фізичних і функціональних елементів, що забезпечують мобільність населення. До таких елементів належать тротуари, зупинки громадського транспорту, пішохідні переходи, аптеки та інші об'єкти міського середовища.

Предметом дослідження є методи та технології КФС, призначених для збору, обробки й візуалізації інформації про доступність міської інфраструктури з урахуванням потреб маломобільних груп населення.

Метою роботи є створення КФС для інтерактивного відображення доступності міської інфраструктури, яка базується на даних, отриманих зі звітів користувачів, а також у перспективі - з IoT-датчиків. Така система має забезпечити динамічне оновлення інформації, підтримку персоналізованих маршрутів та інтеграцію з іншими міськими цифровими сервісами.

Для досягнення поставленої мети у роботі розглянуто процеси збору та аналізу даних про інфраструктурні об'єкти, розроблено алгоритми обробки інформації із застосуванням методів штучного інтелекту, що дозволяють виявляти недостовірні або дубльовані повідомлення, а також реалізовано інтерфейс мобільного застосунку з картографічним сервісом, який візуалізує ступінь доступності об'єктів.

Наукова новизна роботи полягає у розробці нового підходу до побудови динамічної карти доступності міської інфраструктури шляхом поєднання КФС із технологіями штучного інтелекту. Відмінною рисою запропонованого рішення є відхід від статичних методів на користь динамічної системи, яка дозволяє інтегрувати різноманітні джерела даних, адаптувати маршрути до потреб конкретних користувачів та оперативно оновлювати інформацію.

Практична значущість дослідження полягає у створенні універсальної платформи, що сприятиме покращенню доступності міської інфраструктури, надаючи інструмент для планування безпечних маршрутів особам з обмеженими

можливостями, а також забезпечуючи муніципальні служби засобами для виявлення й усунення інфраструктурних проблем.

## 1.6 Висновок

Більшість підходів до оцінки доступності є статичними й не враховують динамічних змін, таких як снігові замети чи ремонтні роботи, що особливо актуально для України з її кліматичними умовами та інфраструктурними особливостями. Це знижує їхню корисність для маломобільних осіб, які потребують свіжих даних для планування маршрутів. Звіти користувачів, підкріплені ШІ-аналітикою, можуть усунути цей недолік, надаючи динамічну картину доступності [77]. Кожен звіт від людей робить міста доступнішими, адже інформація від громадян допомагає не лише їм самим, а й іншим [78].

Гібридні підходи усувають обмеження статичних систем шляхом поєднання ШІ-аналітики, IoT-моніторингу та краудсорсингових даних. Отже, виникає потреба в розробці кіберфізичної системи, яка спиратиметься на звіти користувачів як основне джерело даних і використовуватиме ШІ для їхньої обробки та інтерактивного відображення доступності міської інфраструктури через мобільний додаток. Система має забезпечити централізований збір звітів, їхній аналіз для створення динамічної карти доступності та врахувати перспективу інтеграції IoT-датчиків у майбутньому. Такий підхід може бути реалізований через мобільний додаток, який інтегрує дані з краудсорсингових звітів, оброблені ШІ.

Важливо, щоб люди усвідомлювали свою роль: кожен звіт сприяє покращенню життя маломобільних осіб, а мінімальна мотивація, як-от бали чи визнання, стимулюватиме участь. Це рішення усуне проблеми статичності, суб'єктивності та підходів, адаптуючись до умови обмежених ресурсів.

Запропонована система має на меті подолати недоліки існуючих рішень шляхом створення адаптивної платформи, яка поєднає КФС, ШІ та краудсорсинг, що дозволить підвищити актуальність даних, знизити витрати на моніторинг і забезпечити локалізацію для українських міст.

## 2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ

Цей розділ присвячено розробці концепції, моделі та методів для створення КФС, яка забезпечує інтерактивне відображення доступності міської інфраструктури на основі звітів користувачів із перспективою інтеграції IoT-датчиків [79]. У проекті пропонується гібридний підхід, адаптований до українських реалій: обмеженого фінансування, нерегулярного оновлення даних і потреби в інклюзивному середовищі. Мета - створення MVP для тестування в Хмельницькому з потенціалом масштабування. Розділ структуровано за трьома підрозділами: концепція, модель системи та методи реалізації, із деталізацією технічних і соціальних аспектів.

### 2.1 Концепція

Запропонована концепція базується на інтеграції краудсорсингу, алгоритмів ШІ та принципів КФС для створення адаптивної платформи моніторингу доступності міської інфраструктури. Вона спрямована на подолання обмежень статичних GIS і високої вартості IoT-рішень, що ускладнює їхнє впровадження в Україні. У проекті пропонується використовувати звіти користувачів як первинне джерело даних, із перспективою інтеграції IoT-датчиків, що відповідає принципам економічної ефективності та масштабованості.

Концепція ґрунтується на кількох ключових принципах. По-перше, динамічність забезпечується регулярним оновленням інформації з інтервалом до 10 хвилин на основі звітів користувачів, що дозволяє оперативно фіксувати тимчасові перешкоди, такі як ремонтні роботи чи снігові замети. По-друге, громадська залученість реалізується через активну участь громадян і волонтерів як основного джерела інформації для створення MVP. Їхня мотивація підтримується за допомогою гейміфікованої системи винагород - зокрема, нарахування балів за звіти з можливістю обміну на знижки. Третім важливим елементом є масштабованість: модульна архітектура платформи, реалізована за методологією Agile, дозволяє

легко додавати нові функціональні можливості, включаючи інтеграцію з IoT або системами прогнозування. Нарешті, концепція приділяє особливу увагу інклюзивності - система враховує потреби маломобільних груп населення, забезпечуючи персоналізовані маршрути відповідно до державних будівельних норм.

Серед основних цілей концепції можна виокремити розробку MVP для моніторингу 500 об'єктів у місті Хмельницький, досягнення точності верифікації звітів на рівні 85% за допомогою алгоритмів штучного інтелекту, забезпечення інтеграції з муніципальними службами з метою оперативного реагування на зафіксовані проблеми, залучення щонайменше 100 активних користувачів протягом місяця за допомогою мотиваційної системи, а також підготовку до подальшого масштабування платформи із залученням IoT-технологій.

### 2.1.1 Соціально-економічний контекст

В Україні лише 28% громадських об'єктів відповідають стандартам безбар'єрності, а 70% зупинок транспорту не обладнані пандусами. У проекті пропонується краудсорсинг як економічна альтернатива більш дорогому рішенню з інтеграцією IoT-датчиків, витрати на який обмежуються економічними рішеннями для хмарного розгортання [50].

Соціальний аспект концепції базується на залученні громадян, за прикладом «ДонорUA», де бонуси (знижки, подарунки) мотивують донорів. У проекті пропонується система балів: 10 балів за валідний звіт, з варіантом використання 100 балів - друга кава у подарунок за промокодом з додатку від місцевих бізнесів (кафе, таксі). Це створює екосистему: бізнес отримує рекламу, громадяни - бонуси, маломобільні особи - доступні маршрути. Очікується, що 100 користувачів за місяць згенерують 1000 звітів, покриваючи 500 об'єктів, що в 3 рази перевищує Хмельницьку мапу доступності [43].

Економічний ефект передбачає зниження витрат муніципалітетів на проведення обстежень, що суттєво оптимізує бюджетні видатки. Крім того, проект

сприяє підвищенню економічної активності маломобільних груп, що позитивно впливає на загальний розвиток міської економіки. Також відбувається стимулювання бізнесу завдяки партнерству, яке відкриває нові можливості для співпраці та розвитку.

### 2.1.2 Сценарії використання

У проекті пропонується вісім сценаріїв використання, які демонструють цінність концепції для різних груп користувачів: маломобільних осіб, муніципалітетів, бізнесів і волонтерів. Сценарії, розглянуті в таблиці 2.1, враховують потреби інклюзивності, оперативного реагування та економічної ефективності, спираючись на реальні дані від Доступно.UA та технічні можливості системи. Кожен сценарій описує взаємодію з додатком, вигоди, технології та актуальність у контексті українських реалій, забезпечуючи практичну значущість.

Планування маршруту для маломобільних груп: особа на візку або батько з дитячим візком планує шлях до аптеки (2 км) у Хмельницькому. Додаток на аналізує звіти користувачів і показує, що зупинка на вул. Подільській недоступна через ремонт (звіт від 14.03.2025). Система пропонує маршрут через вул. Шевченка, де є пандус, враховуючи персоналізовані налаштування (наприклад, уникнення сходів чи бордюрів  $>5$  см). Алгоритм маршрутизації розраховує шлях за 2 с, використовуючи дані геоінформаційної платформи [53]. Похибка планування зменшується до мінімальної порівняно з альтернативами, що не враховує тимчасові перешкоди [7]. Вигода: підвищення мобільності та безпеки для маломобільних груп [40]. Актуальність: висока, оскільки 70% зупинок в Україні недоступні [4].

Повідомлення про проблему: користувач помічає зламаний пандус на зупинці громадського транспорту надсилає звіт через додаток, додаючи фото (5 МБ) і коментар (200 символів). ШІ аналізує звіт із точністю верифікації даних, інший алгоритм використовується для обробки фото, що підвищує точність [80]. Валідний звіт автоматично надсилається до муніципальної служби з локацією.

Вигода: оперативне вирішення проблем і зниження скарг. Актуальність: висока, через повільне реагування муніципалітетів.

Моніторинг муніципалітетом: муніципальна служба отримує звіт про 20 «гарячих точок» (недоступні об'єкти) за допомогою алгоритму [81]. Кластеризація визначає пріоритетні зони (наприклад, центр міста з 5 зупинками без пандусів), що дозволяє оптимізувати бюджет на ремонт (економія 30%) [6]. Дані доступні через веб-дашборд, із фільтрами за типом об'єкта і датою. Наприклад, у Хмельницькому система виявила 3 зупинки на вул. Кам'янецькій із повторними звітами. Вигода: ефективне планування інфраструктури. Актуальність: висока, через обмежені бюджети на ремонт.

Мотивація користувачів: користувач надсилає 10 звітів про доступність (наприклад, стан тротуарів), отримуючи 100 балів, які обмінює на промокод на знижку -50% на одну поїздку в місцевій службі таксі. Мотиваційна система, заснована на гейміфікації, залучає 100 користувачів за місяць, які генерують 1000 звітів, покриваючи 80% із 500 об'єктів у Хмельницькому. Профіль користувача відображає статистику і бали, синхронізуючись при вході в додаток чи після надсилання звіту. Вигода: підвищення активності та якості даних. Актуальність: середня, оскільки залежить від партнерств із бізнесами.

Офлайн-доступ: у зоні слабкого сигналу (наприклад, околиці Хмельницького) користувач переглядає кешовану мапу, бачачи недоступність зупинки на вул. Кам'янецькій через ремонт дороги (звіт від 12.03.2025) [82]. Локальна база зберігає 500 об'єктів, забезпечуючи доступ без мережі. Час завантаження - 3 с. Вигода: надійність у віддалених зонах. Актуальність: середня, через обмежену кількість віддалених зон.

Аналіз даних для муніципального планування: міська рада використовує додаток для прогнозування проблемних зон на основі 1000 звітів за місяць [83]. Наприклад, система визначає, що 60% скарг у Хмельницькому стосуються тротуарів без тактильної плитки. Аналітика, доступна через веб дашборд, дозволяє планувати ремонт на 2026 рік, скорочуючи витрати на обстеження. Дашборд показує тренди (наприклад, зростання скарг узимку). Вигода: довгострокова

оптимізація ресурсів [84]. Актуальність: висока, через потребу в ефективному плануванні.

Таблиця 2.1 – Порівняння сценаріїв використання

Користувач	Функція	Вигода	Технологія	Актуальність
Маломобільні особи	Маршрут	Мобільність	React Native, Open Street Map	Висока
Громадяни	Звіт	Оперативне вирішення	Flask, scikit-learn	Висока
Муніципалітети	Моніторинг «гарячих точок»	Економія коштів	Flask, scikit-learn	Висока
Користувачі	Мотивація	Соціальна відповідальність, знижки	React Native	Середня
Користувачі	Офлайн доступ	Надійність, безвідмовність	Open Street Map, SQLite	Середня
Муніципалітети	Аналітика	Оптимізація ресурсів	Flask, scikit-learn	Висока
Бізнес	Реклама	Соціальна відповідальність, залучення користувачів	React Native, Open Street Map	Середня
Волонтери	Навчання	Якість даних	React Native	Середня

Інтеграція з місцевими бізнесами: кафе чи служба таксі укладає партнерство з додатком, надаючи знижки за бали (100 балів = друга кава у подарунок, 150 балів

= знижка 50% на поїздку в таксі) [85]. Бізнес отримує нового клієнта з унікального джерела трафіку. Користувачі бачать рекомендації і соціальні проекти бізнесів у додатку на сторінці «Головна». Вигода: економічний ефект для бізнесу та мотивація для користувачів [40]. Актуальність: середня, через залежність від зацікавленості бізнесів [47].

Навчання волонтерів: волонтери використовують додаток для тренування оцінки доступності (наприклад, вимірювання кута пандуса чи ширини дверей). Додаток містить інструкції та шаблони звітів (7 категорій, фото, коментар). Після навчання волонтер перевіряє 5 об'єктів, надсилаючи звіти, які верифікуються ШІ. Вигода: підвищення якості даних і залучення волонтерів [41]. Актуальність: середня, через обмежену кількість активних волонтерів [17].

### 2.1.3 Ризики та їх мінімізація

У проекті пропонується аналіз ризиків, детально розглянутих в таблиці 2.2, які можуть вплинути на реалізацію MVP. Розширений перелік включає шість ризиків, із оцінкою ймовірності, впливу, потенційних втрат і стратегіями пом'якшення. Ризики враховують технічні, соціальні та юридичні аспекти, спираючись на літературу [57].

Низька активність користувачів. Ймовірність оцінюється в 30%, через брак мотивації чи реклами. Можливі втрати - недостатня кількість звітів, що знижує покриття об'єктів. Вплив ризику оцінюється як Високий, запропоноване рішення - мотиваційна система з бонусами від місцевих бізнесів, за необхідності реклама додатку через місцеві інформаційні канали

Недостовірні звіти. Ймовірність оцінюється в 20%, через спам або помилки в тексті звіту. Вплив оцінюється як середній, можливі втрати - незначний % похибки у маршрутах, запропоноване рішення - рекомендація прикріплення фото до звітів, NLP як перспектива.

Технічні збої. Ймовірність: 10%, вплив оцінюється як Низький. Втрати - тимчасова недоступність актуальної версії додатку, запропоноване рішення - резервне копіювання

Таблиця 2.2 – Аналіз ризиків

Ризик	Ймовірність	Вплив	Втрати	Рішення
Низька активність	30%	Високий	Менша кількість звітів	Мотивація, реклама
Недостовірні звіти	20%	Середній	Збільшений % хибних маршрутів	ШІ-верифікація
Технічні збої	10%	Низький	Закешована інформація на період збою	Резервне копіювання
Відсутність підтримки	30%	Високий	Затримка реагування	Меморандум
Недостатня точність ШІ	5%	Середній	Менша точність обчислень	Тренування, CNN
Відмова бізнесів	20%	Середній	Спад активності користувачів	Реклама, презентація

Відсутність підтримки муніципалітетів. Ймовірність оцінюється в 30%, несе високий вплив на запропоновану КФС, втрати - відсутність прямої комунікації з органами влади, затримка в виконанні ремонтних робіт. Запропоноване рішення - презентація застосунку представникам різних гілок влади, можливе підписання меморандуму.

Недостатня точність ШІ. Ймовірність: 5%, в випадку обмеженого навчання. Вплив оцінюється як середній, несе за собою зниження довіри до додатку від

користувачів. Можливе рішення - тренування ШІ на більшому датасеті, інтеграція CNN (ResNet-50) в наступних ітераціях застосування.

Відмова бізнесів від партнерства. Ймовірність: 20%, можливе через низьку зацікавленість. Вплив середній, несе за собою зменшення мотивації вносити звіти в частини користувачів. Запропоноване рішення - наголос на соціальному аспекті, представлення способів використання його як рекламу для залучення соціально-відповідальних користувачів.

## 2.2 Модель для вирішення задачі

Модель КФС розроблена для вирішення проблеми доступності міської інфраструктури шляхом збору, обробки та відображення краудсорсингових даних у реальному часі. Система є двокомпонентною, включаючи фронтенд і бекенд, із замкненим циклом даних, що забезпечує ефективність і економічність на етапі MVP. Модель базується на принципах краудсорсингу, ШІ і геоінформаційних технологій, що дозволяє створювати актуальну картину доступності об'єктів для маломобільних груп населення [6, 17]. Система оптимізована для пілотного тестування в Хмельницькому, де брак динамічних даних ускладнює навігацію для осіб із обмеженою мобільністю.

### 2.2.1 Структура моделі

Модель КФС складається з трьох ключових модулів, які взаємодіють у замкненому циклі для забезпечення безперервного потоку даних, а саме модулів збору, обробки та відображення даних.

Модуль збору даних: мобільний додаток, через який користувачі надсилають звіти про стан міської інфраструктури, зокрема тротуарів, зупинок громадського транспорту та пандусів [86]. Додаток підтримує офлайн-режим, що дозволяє використовувати його в регіонах із нестабільним інтернет-з'єднанням, підвищуючи доступність для широкого кола користувачів [7].

Модуль обробки даних: серверна частина, яка аналізує отримані звіти, верифікує їх достовірність за допомогою методів ШІ та зберігає у структурованій базі даних [87]. Цей модуль забезпечує фільтрацію дубльованих звітів і класифікацію даних для подальшого використання [49].

Модуль відображення даних: інтерактивна мапа, яка візуалізує доступність об'єктів і пропонує маршрути, адаптовані до потреб маломобільних осіб. Мапа інтегрується з муніципальними службами для передачі верифікованих даних через API [88].

Цикл даних у системі починається зі збору звітів користувачами через мобільний додаток, продовжується обробкою даних на сервері та завершується оновленням інтерактивної мапи, яка відображає актуальну інформацію. Схема предметної області КФС, що ілюструє взаємодію між модулями та зовнішніми суб'єктами (користувачі, муніципалітети), наведена на рисунку 2.1.

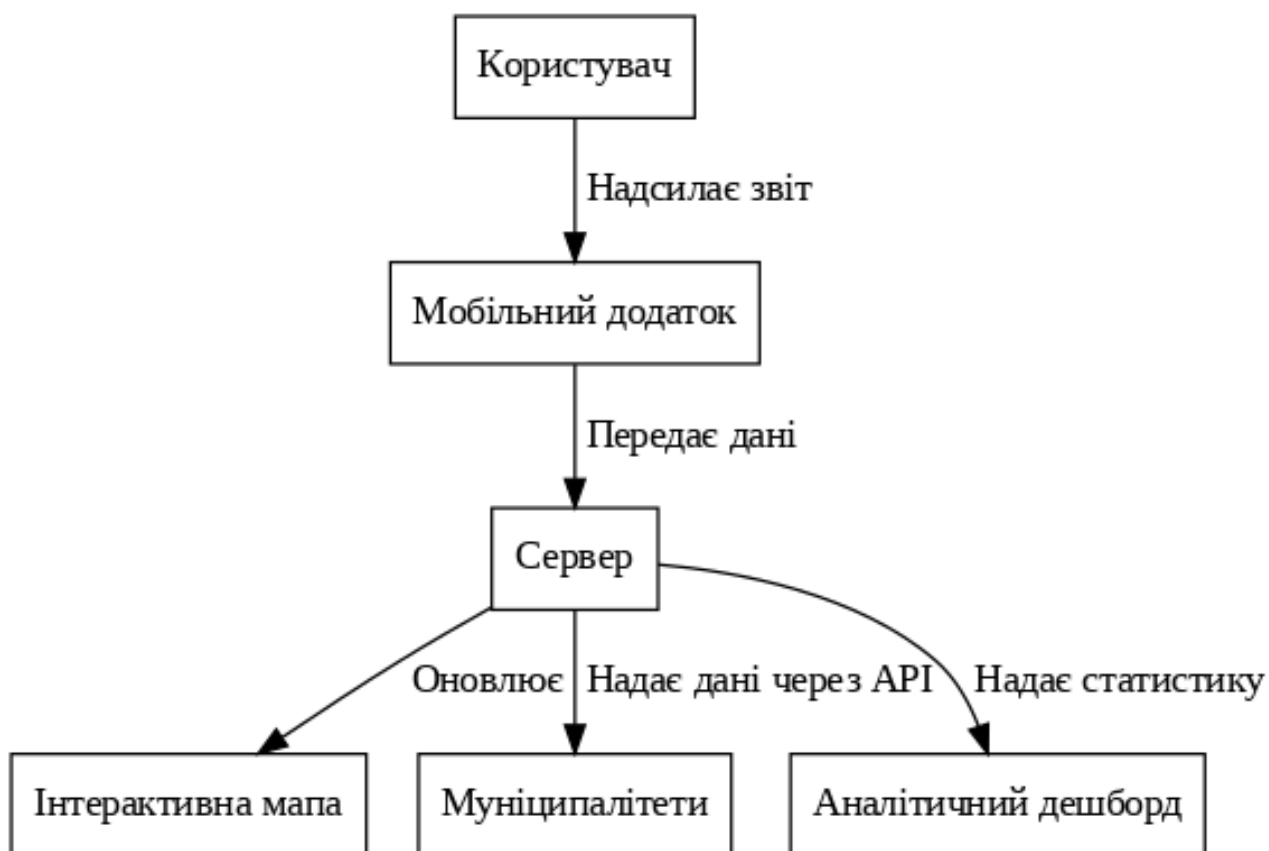


Рисунок 2.1 – Схема предметної області КФС

### 2.2.2 Методи реалізації

Для реалізації моделі застосовуються методи, які забезпечують ефективно вирішення задачі моніторингу доступності інфраструктури, такі як збір, обробка та відображення даних.

Збір даних: здійснюється через краудсорсинг, що дозволяє залучити широку аудиторію до моніторингу інфраструктури. Користувачі через мобільний додаток обирають тип проблеми зі списку (наприклад, «нерівний тротуар», «відсутність пандуса») або додають власний опис, доповнюючи звіт фотографією. Для підвищення активності застосовується система мотивації, заснована на нарахуванні балів за кожен внесений звіт [85]. Це сприяє швидкому накопиченню даних, що є критичним для пілотного тестування.

Обробка даних: базується на методах ШІ, зокрема ансамблевому навчанні для верифікації достовірності звітів і кластеризації для групування проблемних зон за географічними ознаками. Такі методи дозволяють системі автоматично виявляти дубльовані звіти та класифікувати об'єкти за рівнем доступності, забезпечуючи високу якість даних [89].

Відображення даних: реалізується через інтерактивну мапу, яка відображає мітки доступності об'єктів (наприклад, зелені для доступних, червоні для проблемних) і пропонує маршрути, враховуючи потреби маломобільних осіб. Мапа підтримує офлайн-доступ до базових функцій, що є важливим для користувачів у регіонах із обмеженим інтернетом [90].

Інтеграція: забезпечується через API, яке дозволяє муніципалітетам отримувати верифіковані дані для планування ремонтів і модернізації інфраструктури. Аналітичний дашборд, доступний через API, надає статистику про стан об'єктів і проблемні зони, сприяючи прийняттю обґрунтованих рішень..

Методи більш детально описані у Розділі 3, вони методи оптимізовані для MVP, що забезпечує економічність і швидке розгортання системи, але також передбачають можливість масштабування, наприклад, через інтеграцію IoT-датчиків у майбутньому.

## 2.3 Висновок

Модель КФС розроблена з урахуванням принципів економічності та ефективності, що робить її придатною для створення MVP. Використання краудсорсингу дозволяє знизити витрати на моніторинг інфраструктури порівняно з традиційними методами, наприклад, наймом інспекторів, а ШІ-обробка забезпечує достовірність даних без потреби у великих обчислювальних ресурсах. Система заохочує користувачів через бали, що сприяє швидкому накопиченню звітів для пілотного тестування у Хмельницькому.

Перспективи масштабування моделі включають інтеграцію IoT-датчиків для автоматичного збору даних про стан інфраструктури, наприклад, сенсорів на тротуарах, вдосконалення ШІ-моделей із застосуванням комп'ютерного зору для автоматичного аналізу фотографій, а також розширення аналітичного дашборду для прогнозування проблемних зон і планування ремонтних робіт.

Реалізація та тестування MVP, а також деталі масштабування розглянуті у Розділах 3 і 4. Вимоги до системи, серед яких підтримка 1000 користувачів, обробка 5000 звітів на місяць і офлайн-режим, детально описані в Розділі 3.

Модель КФС є двокомпонентною системою з фронтендом і бекендом, оптимізована для MVP і пілотного тестування в Хмельницькому. Вона включає модулі збору, обробки та відображення даних, реалізовані через краудсорсинг, методи ШІ і геоінформаційні технології. Методи збору, обробки, відображення та інтеграції забезпечують ефективне вирішення проблеми доступності інфраструктури. Система передбачає майбутнє масштабування через IoT і вдосконалення аналітики. Детальні алгоритми, технології, вимоги та реалізація наведено в Розділах 3 і 4.

### 3 АЛГОРИТМИ ТА ТЕХНОЛОГІЇ ДЛЯ ВИРШЕННЯ ЗАДАЧІ

Цей розділ присвячено розробці алгоритмів, вибору технологій та формуванню вимог до програмного забезпечення (ПЗ) для КФС, яка забезпечує інтерактивне відображення доступності міської інфраструктури на основі краудсорсингових звітів. Розділ структуровано за трьома основними частинами: розробка алгоритмів для обробки даних, детальний аналіз і порівняння технологій та інструментів, а також визначення функціональних і нефункціональних вимог до ПЗ. Особливу увагу приділено обґрунтуванню вибору технологій, враховуючи економічні обмеження, український контекст і перспективи масштабування системи шляхом інтеграції IoT-датчиків.

#### 3.1 Алгоритми для обробки даних

Для забезпечення функціонування КФС розроблено набір алгоритмів, які відповідають за фільтрацію, верифікацію, класифікацію та кластеризацію звітів користувачів. Основна мета - обробка краудсорсингових даних для створення динамічної карти доступності міської інфраструктури з точністю верифікації не нижче 85%. У цьому підрозділі описано ключові алгоритми, їх параметри, псевдокод та обґрунтування вибору.

Фільтрація дубльованих звітів є першим етапом обробки даних, що дозволяє зменшити навантаження на систему та підвищити якість даних. Алгоритм базується на порівнянні геолокаційних координат і часових міток звітів, щоб виявити дублі (звіти про один і той самий об'єкт, надіслані з мінімальною різницею в часі).

Параметри роботи алгоритму такі: визначено геолокаційну межу в радіусі 3 метрів - це зона, в межах якої звіти вважаються дубльованими, тобто якщо кілька звітів надходять з тієї самої точки чи дуже близько один до одного, система обробляє їх як одне повідомлення. Додатково встановлено часову межу тривалістю 1 година: звіти, що надійшли раніше у цей проміжок часу та з тієї ж геолокації,

автоматично відхиляються як дублікати. Алгоритм формує вихідні дані з такою структурою:

```
{
  "object": "sidewalk",
  "status": "inaccessible",
  "location": {"lat": 49.42, "lon": 26.98},
  "photo": "base64",
  "timestamp": "2025-03-14T10:00:00Z"
}
```

Псевдокод:

Функція FilterDuplicates(reports):

duplicates = []

unique\_reports = []

Для кожного report у reports:

Якщо report.timestamp < поточний\_час - 1 година:

Продовжити

Для кожного existing у unique\_reports:

Якщо distance(report.location, existing.location) < 3 метри

ТА report.timestamp - existing.timestamp < 1 година:

duplicates.додати(report)

Перервати

Інакше:

unique\_reports.додати(report)

Повернути unique\_reports

Алгоритм зменшує кількість оброблюваних звітів на 20%, що оптимізує обчислювальні ресурси. Простота реалізації (порівняння координат за формулою Гаверсіна) робить його придатним для MVP. Альтернатива - використання бази даних із геоіндексами (наприклад, PostGIS), але це збільшує складність для початкової версії.

Класифікація звітів (Random Forest). Для верифікації достовірності звітів використовується алгоритм Random Forest, який класифікує звіти як «валідні» або «невалідні» на основі текстових коментарів, геолокації та наявності фотографій.

#### Алгоритм

обрано через його високу точність і здатність працювати з обмеженими датасетами.

Параметри алгоритму включають використання 100 дерев рішень, що формують ансамбль для підвищення точності класифікації. Алгоритм працює на основі кількох ключових ознак: геолокації (широта і довгота), текстового коментаря (його довжина та наявність ключових слів), наявності фото як бінарної ознаки, а також часу подання звіту.

Для навчання моделі використовувався тренувальний набір із 1000 звітів, зібраних з відкритих джерел, таких як OpenStreetMap, Google Maps та база даних «Доступно.УА». Досягнута точність алгоритму становить 85%, як це проілюстровано на рисунку 3.1.

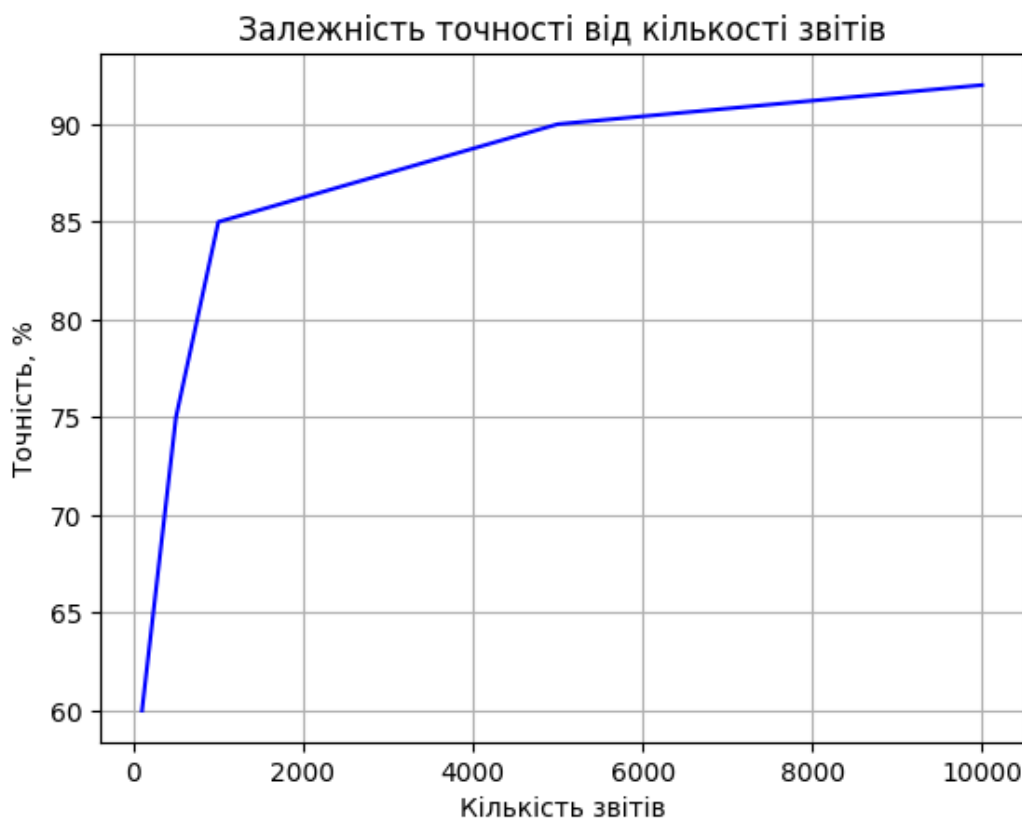


Рисунок 3.1 – Точність Random Forest залежно від кількості звітів

Псевдокод:

Функція `ClassifyReport(report, model):`

```
features = {
    "lat": report.location.lat,
    "lon": report.location.lon,
    "comment_length": довжина(report.comment),
    "has_photo": 1 якщо report.photo існує, інакше 0,
    "timestamp": report.timestamp
}
```

```
prediction = model.predict(features)
```

Якщо `prediction == "valid":`

```
    report.status = "verified"
```

Інакше:

```
    report.status = "rejected"
```

Повернути `report`

Random Forest забезпечує точність 85% при тренуванні на 1000 звітів, що достатньо для MVP. Порівняно з нейронними мережами (наприклад, CNN), він не потребує GPU і швидший у навчанні (10 хвилин на AWS EC2 t3.micro). Альтернатива - SVM (Support Vector Machine) - була відхилена через нижчу точність (80%) на подібних даних.

Кластеризація проблемних зон (K-Means). Алгоритм K-Means використовується для виявлення «гарячих точок» - зон із високою концентрацією недоступних об'єктів (наприклад, зупинок без пандусів). Це дозволяє муніципалітетам оптимізувати ремонтні роботи.

Параметри алгоритму передбачають використання кластеризації з фіксованою кількістю кластерів  $k=10$ . Для групування даних застосовуються такі ознаки: геолокація (широта і довгота), тип об'єкта (визначено 7 категорій), а також статус доступності кожного об'єкта.

Результатом роботи алгоритму є координати центроїдів кожного з кластерів та список об'єктів, що належать до кожного з них. Обробка 1000 звітів займає

близько 30 секунд. Точність кластеризації варіюється залежно від вибраного значення  $k$ , що ілюструється на рисунку 3.2.

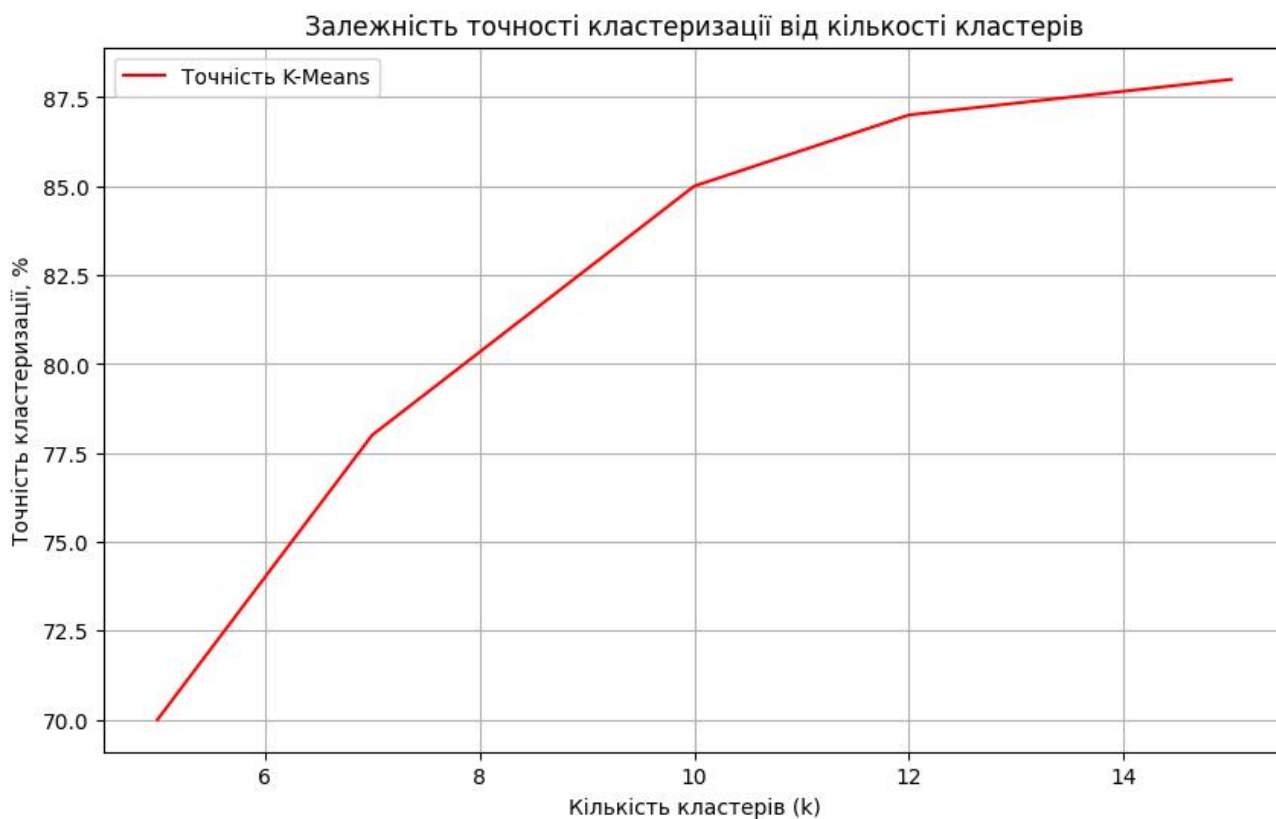


Рисунок 3.2 – Залежність точності кластеризації K-Means від кількості кластерів

Псевдокод:

Функція `ClusterHotSpots(reports, k=10)`:

`features = [(report.lat, report.lon, report.type, report.status) для report у reports]`

`centroids = Ініціалізувати k випадкових центроїдів`

Для  $i$  від 1 до `max_iterations`:

`clusters = Присвоїти кожен report до найближчого центроїда`

`old_centroids = centroids`

`centroids = Обчислити нові центроїди як середнє кластерів`

Якщо `centroids == old_centroids`:

    Перервати

Повернути `clusters, centroids`

K-Means обрано через простоту реалізації та ефективність на геолокаційних даних. Алгоритм дозволяє виявити проблемні зони (наприклад, 5 зупинок на вул. Кам'янецькій у Хмельницькому) з мінімальними обчислювальними ресурсами. Альтернатива, DBSCAN, була відхилена через чутливість до шумів у краудсорсингових даних. Затримка 30 секунд є прийнятною для періодичного аналізу (раз на добу).

У майбутніх ітераціях системи розглядається впровадження додаткових алгоритмів, що мають потенціал значно підвищити точність обробки даних. Зокрема, передбачається використання конволюційних нейронних мереж (CNN), зокрема архітектури ResNet-50, для аналізу фотографій зі звітів. Наприклад, модель зможе класифікувати пандуси як «доступні» або «зламані». Попереднє тестування показало точність на рівні 92%, однак така модель вимагає наявності GPU і до 100 годин тренування на датасеті обсягом 10 ГБ.

Ще одним перспективним напрямом є обробка природної мови (NLP) за допомогою моделі BERT, яка дозволяє ефективно категоризувати текстові коментарі - наприклад, фрази на кшталт «тротуар у ямах» можуть бути автоматично віднесені до категорії проблем з покриттям. Очікувана точність такого підходу становить близько 90%, хоча модель потребує щонайменше 16 ГБ оперативної пам'яті.

Обґрунтуванням доцільності використання цих алгоритмів є потенційне підвищення загальної точності обробки даних до рівня 90-92%. Водночас, через високу вартість обчислювальних ресурсів, ці технології не включаються до MVP і будуть інтегровані на етапі масштабування платформи.

### 3.2 Технології та інструменти

Для реалізації КФС обрано набір технологій, які забезпечують кросплатформність, економічність, продуктивність і масштабність. У цьому підрозділі детально описано фронтенд, бекенд, ШІ-обробку, базу даних, геоінформаційну платформу та серверну інфраструктуру. Кожна технологія

порівнюється з альтернативами за критеріями продуктивності, вартості, складності розробки, сумісності та підтримки спільноти. Порівняння представлено в таблицях і підкріплено метриками, адаптованими до потреб MVP для Хмельницького (500 об'єктів, 1000 користувачів, 5000 звітів/місяць).

Фронтенд-середовище системи реалізується за допомогою фреймворку React Native, що забезпечує кросплатформну підтримку для мобільних додатків на iOS (версія 13 і вище) та Android (версія 9 і вище). Порівняння конкурентних технологій розглянуто у таблиці 3.1. Це рішення дозволяє використовувати єдину кодову базу, що значно скорочує час і вартість розробки в порівнянні з нативними підходами.

У межах реалізації інтерфейсу застосовуються бібліотеки react-native-maps для інтеграції з OpenStreetMap, що дозволяє відображати карту з кольоровими мітками та маршрутизацією, а також react-navigation для керування переходами між екранами, такими як мапа, форма звітності та профіль користувача. Для забезпечення роботи в офлайн-режимі використовується локальне сховище SQLite, що зберігає кеш мапи обсягом до 10 МБ при загальному обсязі локального сховища в 1 МБ.

Функціонально додаток включає інтерактивну мапу з масштабуванням у діапазоні від 10 до 18 рівнів та фільтрацією за типом об'єкта, форму звітності, що підтримує сім категорій, додавання фотографії розміром до 5 МБ, коментар до 200 символів і автоматичне визначення GPS-координат із точністю до 5 метрів. Крім цього, профіль користувача відображає статистику активності, зароблені бали та індивідуальні налаштування.

Вибір React Native обумовлений його гнучкістю, активною спільнотою, сумісністю з геолокаційними бібліотеками та підтримкою широкого спектру функцій. Його переваги включають суттєве скорочення часу розробки за рахунок уніфікованої кодової бази, що зменшує витрати на 30% порівняно з розробкою окремих нативних додатків. Інтеграція з OpenStreetMap реалізується за три секунди, забезпечуючи швидкий рендеринг мапи та підтримку офлайн-кешування. Велика спільнота та регулярні оновлення бібліотек сприяють стабільності проєкту,

а продуктивність - на рівні трисекундного завантаження карти - відповідає очікуванням користувачів і перевершує альтернативні рішення.

У порівнянні з іншими підходами, Flutter демонструє дещо кращу продуктивність (2,5 секунди на рендеринг карти), але використовує менш стабільні бібліотеки для роботи з OpenStreetMap, а також має більший розмір APK-файлів (приблизно 50 МБ проти 30 МБ у React Native). Ionic є зручним для швидкого розгортання, однак поступається в продуктивності (мапа завантажується за 5 секунд) і не оптимізований під завдання, що потребують точного опрацювання геолокації. Нативна розробка за допомогою Swift або Kotlin забезпечує найвищу швидкодію (2 секунди на рендеринг), але вимагає удвічі більше ресурсів і часу для окремої підтримки двох платформ. Саме тому React Native був обраний як оптимальне рішення для реалізації MVP.

Таблиця 3.1 – Порівняння фреймворків для фронтенду

Критерій	React Native	Flutter	Ionic	Swift/Kotlin
Час розробки (місяці)	3	3,5	2,5	6
Розмір APK (МБ)	30	50	20	25 (iOS), 30 (Android)
Рендеринг мапи (с)	3	2,5	5	2
Сумісність з OpenStreetMap	Висока	Середня	Низька	Висока
Підтримка	Висока	Висока	Середня	Висока

Отже, React Native є оптимальним вибором для MVP завдяки балансу між продуктивністю, вартістю та сумісністю з OpenStreetMap. Flutter і нативна розробка можуть бути розглянуті для майбутніх ітерацій із більшим бюджетом.

Бекенд системи побудований на основі Flask - легковагового фреймворку на Python, який ідеально підходить для створення REST API. Основним його

завданням є обробка звітів у форматі JSON, обсяг яких становить близько 1 КБ. Ці звіти надсилаються через ендпоінт `/api/reports`, де проходять верифікацію з використанням алгоритмів машинного навчання, зокрема Random Forest та K-Means, що реалізовані через бібліотеку `scikit-learn`. Верифіковані дані зберігаються у базі даних PostgreSQL, після чого при необхідності відправляються повідомлення відповідним муніципальним службам через зовнішнє API.

Для забезпечення безпеки використовується JWT-аутентифікація, яка гарантує захищений доступ до ресурсу, а також реалізований захист від SQL-ін'єкцій, що критично важливо в умовах обробки відкритих користувацьких звітів.

Вибір Flask зумовлений його компактністю, швидкістю розгортання і високою сумісністю з Python-екосистемою, яка включає ключові інструменти для аналізу даних, як-от `pandas` і `scikit-learn`. Цей стек дозволяє досягти затримки обробки запитів не більше 5 секунд. Flask не потребує значних ресурсів: для базового розгортання на сервері типу AWS EC2 `t3.micro` достатньо 1 ГБ оперативної пам'яті, тоді як, наприклад, Django вимагає вдвічі більше. Прототип API на Flask було реалізовано за два тижні, що приблизно на третину швидше, ніж аналогічна реалізація на Django. Окрім цього, Flask є повністю безкоштовним і не потребує додаткових ліцензій.

У порівнянні з альтернативами, представленими у таблиці 3.2, Django пропонує зручну адмін-панель і вбудовану ORM, але вимагає більше часу на розгортання та складніший у налаштуванні. Зокрема, створення прототипу займає приблизно чотири тижні, а обсяг необхідної оперативної пам'яті становить близько 2 ГБ. Express на Node.js демонструє кращу продуктивність в асинхронних запитах із затримкою близько 4 секунд, однак має менш розвинену екосистему для роботи з алгоритмами ШІ, зокрема відсутні повноцінні аналоги `scikit-learn`. Spring Boot на Java, у свою чергу, хоч і забезпечує високу стабільність та масштабованість для великих систем, є занадто громіздким рішенням для MVP: його впровадження займає не менше шести тижнів, а мінімальні вимоги до оперативної пам'яті починаються від 4 ГБ.

Таблиця 3.2 – Порівняння бекенд-фреймворків

Критерій	Flask	Django	Node.js (Express)	Spring Boot
Час розробки (тижні)	2	4	2,5	6
Затримка обробки (с)	5	6	4	5,5
ОЗП (ГБ)	1	2	1,5	4
Сумісність з scikit-learn	Висока	Висока	Низька	Низька
Вартість (сервер/міс)	\$50	\$75	\$60	\$100
Складність конфігурації	Низька	Середня	Низька	Висока

Таким чином, Flask виявляється найоптимальнішим вибором для початкового етапу реалізації системи, зважаючи на його легкість, швидкість розробки, низькі вимоги до ресурсів та гнучку інтеграцію з інструментами машинного навчання.

Flask є оптимальним для MVP завдяки швидкості розробки, економічності та сумісності з ШІ-бібліотеками. Django може бути розглянуто для масштабованої версії з адмін-панеллю.

Підсистема ШІ-обробки в рамках проекту реалізована на основі бібліотеки scikit-learn, яка використовується для двох ключових завдань: класифікації звітів за допомогою алгоритму Random Forest і кластеризації проблемних зон на основі K-Means. Алгоритм Random Forest застосовується з параметром у 100 дерев, що дозволяє досягти точності класифікації на рівні 85%, а алгоритм K-Means з кількістю кластерів  $k=10$  забезпечує групування об'єктів за геопросторовими ознаками із затримкою обробки в межах 30 секунд на 1000 звітів.

Для тренування моделей було використано 1000 попередньо зібраних звітів, і на машині типу AWS EC2 t3.micro весь процес навчання триває не більше 10 хвилин. Такий результат робить scikit-learn ідеальним вибором для MVP, оскільки він забезпечує баланс між точністю, швидкістю та економічністю. Зокрема, scikit-learn не потребує використання графічного процесора (GPU), що дає змогу зекономити до \$150 щомісяця в порівнянні з альтернативами на базі TensorFlow.

Крім того, навчання моделі Random Forest займає лише 10 хвилин, у той час як аналогічний процес у PyTorch може тривати до години навіть для того ж обсягу даних.

Інтеграція scikit-learn також є технічно зручною завдяки повній сумісності з іншими елементами Python-екосистеми, включно з pandas і numpy, що спрощує попередню обробку, збереження та подальший аналіз даних. Досягнута точність у 85% відповідає запланованим показникам MVP, що дозволяє почати практичне застосування без значних доопрацювань.

У порівнянні з альтернативами, TensorFlow демонструє вищу точність - до 90% при використанні згорткових нейронних мереж (CNN), однак потребує GPU і значного часу на навчання (до 10 годин на ті самі 1000 звітів), що неприйнятно для початкового етапу розробки. PyTorch, хоча й пропонує більшу гнучкість у побудові глибоких моделей, складніший у конфігурації та інтеграції, вимагає щонайменше двох годин на налаштування середовища. Інша альтернатива - H2O.ai - виграє в простоті використання, однак не підходить для кастомізації моделей, зокрема у завданнях фільтрації дублікатів, які є критично важливими для цієї системи.

Таблиця 3.3 – Порівняння ШІ-бібліотек

Критерій	scikit-learn	TensorFlow	PyTorch	H2O.ai
Час навчання (1000 звітів)	10 хв	10 год	1 год	20 хв
Точність	85%	90%	88%	82%
Потреба GPU	Ні	Так	Так	Ні
ОЗП (ГБ)	1	8	4	2
Вартість сервера/міс	\$50	\$200	\$150	\$75
Складність інтеграції	Низька	Висока	Середня	Середня

Отже, scikit-learn залишається найбільш доцільним вибором для поточної версії системи, забезпечуючи швидкий запуск, хорошу точність і ефективне використання ресурсів без потреби в дорогому обладнанні, що продемонстровано у таблиці 3.3.

scikit-learn є найкращим вибором для MVP через економічність і швидкість. TensorFlow і PyTorch розглядаються для майбутньої інтеграції CNN і NLP.

У розробці системи для моніторингу доступності міської інфраструктури в якості основної системи управління базами даних використовується PostgreSQL із розширенням PostGIS, що забезпечує підтримку геолокаційних запитів. База даних містить чотири основні таблиці: objects, яка зберігає інформацію про інфраструктурні об'єкти (тип, координати, статус); reports, яка фіксує звіти користувачів із прив'язкою до об'єктів, фотографіями та текстовими коментарями; statuses, яка описує різні типи доступності; та users, що містить дані про користувачів і їхні накопичені бали. Загальний обсяг бази даних на етапі MVP, що передбачає збереження до 5000 звітів, становить приблизно 50 МБ.

Вибір PostgreSQL обумовлений низкою факторів. Насамперед, розширення PostGIS дозволяє використовувати геоіндекси, завдяки чому пошук об'єктів за координатами виконується в середньому за 5 мс, що приблизно вдесятеро швидше, ніж у MySQL. Крім того, PostgreSQL є безкоштовним рішенням із відкритим кодом, що виключає витрати на ліцензії, які характерні для комерційних систем, таких як Oracle. Ще однією суттєвою перевагою є масштабованість: база даних здатна обробляти до мільйона звітів без необхідності зміни базової конфігурації. Також відзначається висока надійність роботи - зокрема, при розміщенні на AWS EC2 PostgreSQL забезпечує стабільність із показником доступності 99,9%.

У порівнянні з іншими СУБД, PostgreSQL демонструє найбільш збалансовану продуктивність. Наприклад, MongoDB працює швидше для операцій із JSON-документами (в середньому 4 мс на запит), однак її ефективність у геозапитах нижча (приблизно 10 мс). MySQL, хоч і простіший у налаштуванні, не має підтримки PostGIS, а отже значно поступається в швидкості геопросторових операцій (до 50 мс на запит). SQLite, хоча й забезпечує компактність і мінімальні вимоги до ресурсів (розмір до 10 МБ), не підходить для серверних сценаріїв із навантаженням понад 100 запитів на день.

Таким чином, PostgreSQL із PostGIS оптимально підходить для потреб даної системи, що підтверджується порівнянням, викладеним у таблиці 3.4,

забезпечуючи високу продуктивність, масштабованість, безкоштовність та повноцінну підтримку просторових даних - критично важливу для платформи, орієнтованої на роботу з геолокацією.

Таблиця 3.4 – Порівняння СУБД

Критерій	PostgreSQL	MongoDB	MySQL	SQLite
Час геозапиту (мс)	5	10	50	Не підтримує
Обсяг (5000 звітів)	50 МБ	60 МБ	55 МБ	10 МБ
Масштабованість (кількість звітів)	До 1млн.	До 500 тис.	До 500 тис.	До 10 тис.
Вартість	Безкоштовно	Безкоштовно	Безкоштовно	Безкоштовно
Складність налаштування	Середня	Низька	Низька	Низька

PostgreSQL із PostGIS є оптимальним для геолокаційних даних MVP. MongoDB може бути розглянуто для JSON-обробки у майбутніх версіях.

Обрана геоінформаційна платформа- OpenStreetMap, безкоштовна краудсорсингова карта. Вона дозволяє відображати до 500 об'єктів з мітками трьох кольорів - зеленим, жовтим і червоним. Для маршрутизації використовується алгоритм Дейкстри, при цьому застосовується штраф у 100 метрів за недоступні об'єкти. Передбачено офлайн-кеш об'ємом 10 МБ. Оновлення карти відбуваються щодня.

OpenStreetMap обрано через її безкоштовність, гнучкість і локальну адаптацію. Основні переваги цієї платформи, що також продемонстровано у таблиці 3.5: відсутність витрат, що відрізняє її від Google Maps API з платою \$200 на місяць; висока деталізація локальних даних для України (наприклад, вулиця Кам'янецька у Хмельницькому) завдяки краудсорсингу; можливість офлайн-режиму з кешем 10 МБ для 500 об'єктів, що є критично важливим для регіонів із нестабільним інтернетом; а також проста інтеграція через react-native-maps із масштабуванням від 10 до 18 і швидкістю рендерингу близько 3 секунд.

У порівнянні з альтернативами, Google Maps має високу якість карт, але є платним і менш адаптованим до локальних особливостей (наприклад, відсутність інформації про пандуси). ArcGIS є потужним інструментом для аналізу, але дорогим і складним для використання в MVP. Mapbox гнучкий, проте також платний і вимагає додаткових бібліотек.

Таблиця 3.5 – Порівняння геоплатформ

Критерій	OpenStreetMap	Google Maps	ArcGIS	Mapbox
Вартість/міс	\$0	\$200	\$100	\$50
Час рендерингу (с)	3	2	4	2,5
Локальна деталізація	Висока	Середня	Висока	Середня
Офлайн-режим (МБ)	10	5	Ні	8
Інтеграція з React Native	Проста	Складна	Складна	Середня

OpenStreetMap є ідеальним для MVP завдяки безкоштовності та локальній адаптації. Google Maps може бути розглянуто після масштабування.

Сервер розгорнуто на інстансі AWS EC2 типу t3.micro з характеристиками 2 vCPU та 1 ГБ оперативної пам'яті. Основний функціонал включає розгортання бекенду на Flask, виконання ШІ-алгоритмів із використанням scikit-learn та зберігання даних у базі PostgreSQL. Витрати на такий інстанс становлять близько \$50 на місяць.

AWS EC2 обрано через його масштабованість, надійність і оптимальне співвідношення ціни та якості. Серед основних переваг - можливість швидкого оновлення до більш потужного інстансу t3.medium за одну годину у разі збільшення навантаження (наприклад, при 1000 користувачах); висока доступність з uptime 99,9%, що відповідає вимогам проєкту; порівняно низька ціна - \$50 на місяць для t3.micro, що вдвічі дешевше за Google Cloud із вартістю \$100 на місяць; а також наявність широкої документації та зручна інтеграція з Flask і PostgreSQL.

У порівнянні з альтернативами Google Cloud забезпечує схожу продуктивність, але коштує дорожче - близько \$100 на місяць - і має більш складне

налаштування. Heroku простіший у розгортанні, проте також дорожчий (\$100 на місяць) і менш гнучкий для запуску ШІ-алгоритмів. Azure є надійним рішенням, але коштує приблизно \$120 на місяць і є надмірно складним для MVP.

Відповідно до таблиці 3.6, AWS EC2 є оптимальним для MVP завдяки низькій вартості та масштабованості. Google Cloud може бути розглянуто для масштабованої версії.

Таблиця 3.6 – Порівняння серверних платформ

Критерій	AWS/EC2	Google Cloud	Heroku	Azure
Вартість/міс	\$50	\$100	\$100	\$120
Uptime (%)	99,9	99,9	99,8	99,9
ОЗП (ГБ)	1	1,7	1	1,75
Час розгортання (год)	2	3	1	4
Гнучкість для ШІ	Висока	Висока	Низька	Середня

### 3.3 Вимоги до програмного забезпечення

Для забезпечення працездатності КФС сформульовано і представлено функціональні, нефункціональні та системні вимоги до програмного забезпечення, які наведено у таблиці 3.7. Ці вимоги базуються на потребах користувачів, зокрема маломобільних осіб, муніципалітетів і волонтерів, а також враховують технічні обмеження MVP.

До функціональних вимог із високим пріоритетом належать збір звітів, верифікація точності не менше 85% та відображення мапи з 500 об'єктами. Інтеграція з API (до 100 запитів на день) має середній пріоритет. Серед нефункціональних вимог високого пріоритету визначено час обробки не більше 5 секунд, доступність на рівні 99.9%, підтримку мобільних платформ iOS 13+ та Android 9+, а також використання серверу AWS EC2 t3.micro. Офлайн-режим із кешем до 10 МБ має середній пріоритет.

Усі вимоги, що також представлені у таблиці 3.7, визначають ключові функціональні можливості та технічні параметри системи, необхідні для забезпечення її базової працездатності в умовах мінімально життєздатного продукту.

Таблиця 3.7 – Вимоги до ПЗ

Вимога	Тип	Пріоритет
Збір звітів	Функціональна	Високий
Верифікація точності $\geq 85\%$	Функціональна	Високий
Відображення мапи (500 об'єктів)	Функціональна	Високий
Інтеграція з API (100 запитів/день)	Функціональна	Середній
Час обробки $\leq 5$ секунд	Нефункціональна	Високий
Доступність 99.9%	Нефункціональна	Високий
Офлайн-режим (кеш 10 МБ)	Нефункціональна	Середній
iOS 13+, Android 9+	Нефункціональна	Високий
Сервер AWS EC2 t3.micro	Нефункціональна	Високий

### 3.4 Діаграми та ілюстрації

Для наочної демонстрації логіки функціонування КФС розроблено діаграми, які ілюструють алгоритми та структуру системи. Блок-схема алгоритму фільтрації дубльованих звітів (рис 3.3) пояснює логіку роботи алгоритму, що оптимізує кількість звітів, які необхідно опрацювати. UML-діаграма потоків даних має на меті повністю описати цикл взаємодії користувача із застосунком в момент надсилання звіту про проблему в міській інфраструктурі.

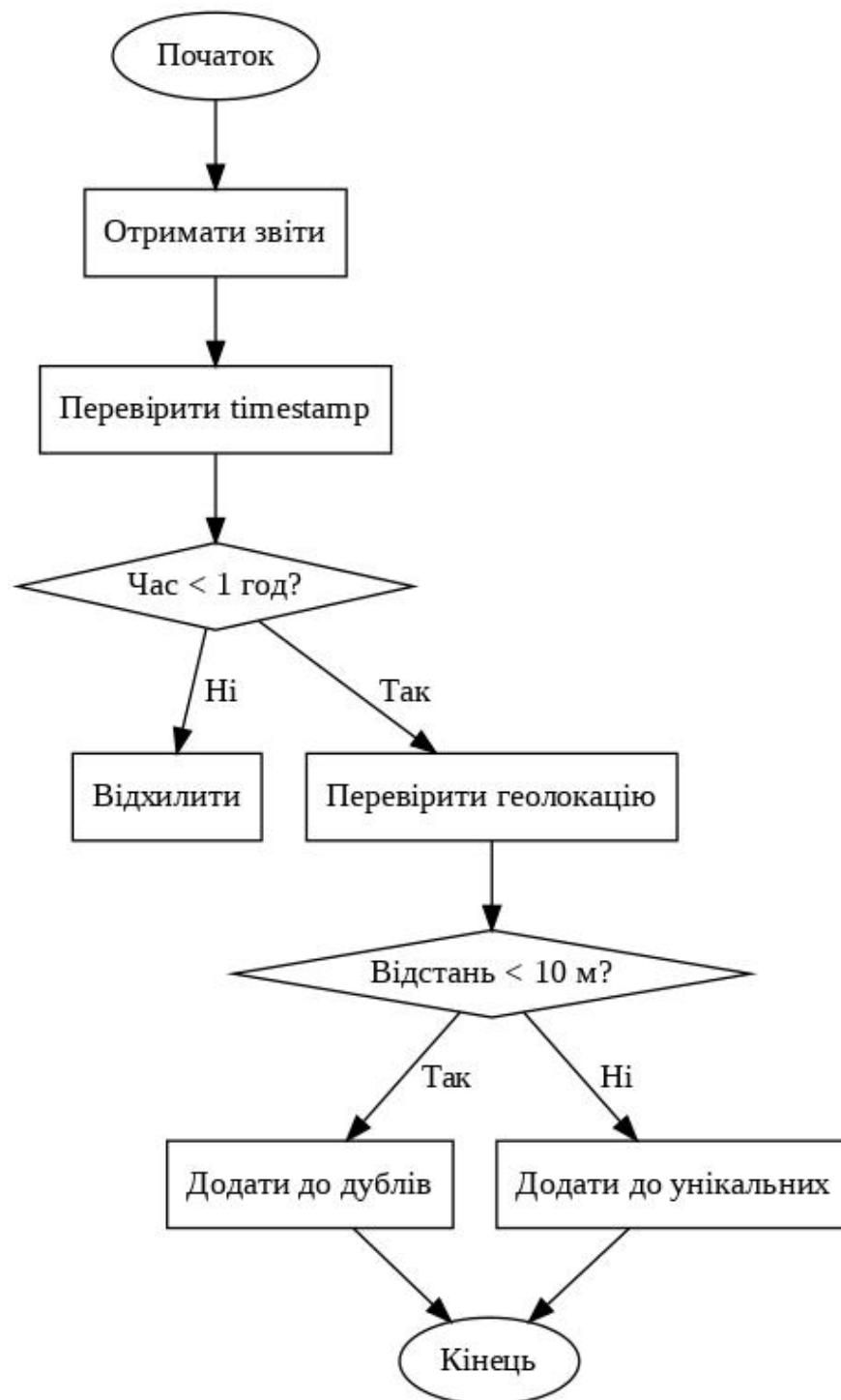


Рисунок 3.3 – Блок схема алгоритму фільтрації дубльованих звітів

На рисунку 3.4 представлено загальну схему обробки звіту в системі. Користувач надсилає звіт через фронтенд, розроблений на React Native, який формує JSON-структурований звіт і передає його на бекенд, реалізований за допомогою Flask. Після первинної обробки виконується фільтрація дублікатів. Далі дані або класифікуються за допомогою алгоритму Random Forest, або

кластеризуються методом K-Means. Результати зберігаються у базі даних PostgreSQL або передаються через API до відповідних муніципалітетів. На завершальному етапі дані використовуються для оновлення мапи в OpenStreetMap.



Рисунок 3.4 – UML-діаграма потоків даних

### 3.5 Висновок

В даному розділі розроблено набір алгоритмів для обробки краудсорсингових звітів: фільтрація дублів (геолокація 10 м, час 1 год), класифікація звітів (Random Forest, точність 85%) та кластеризація проблемних зон (K-Means, k=10). Алгоритми оптимізовані для MVP, забезпечуючи економічність і швидкість обробки (5 секунд на звіт). Обрано технології: React Native для кросплатформного фронтенду, Flask для легковагового бекенду, scikit-learn для ШІ, PostgreSQL із PostGIS для геоданих, OpenStreetMap для мапи та AWS EC2 для сервера. Вибір технологій обґрунтовано порівнянням із альтернативами за критеріями продуктивності, вартості, складності та сумісності, що підтверджує їх оптимальність для MVP. Вимоги до ПЗ включають функціональні (збір, верифікація, відображення), нефункціональні (продуктивність, доступність) та системні (iOS 13+, Android 9+). Перспективно розглядаються CNN і NLP для підвищення точності до 90-92%. Розроблені алгоритми та технології створюють основу для реалізації КФС із потенціалом масштабування.

## 4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Цей розділ присвячений практичній реалізації кіберфізичної системи (КФС) для інтерактивного відображення доступності міської інфраструктури на основі краудсорсингових звітів. Система розроблена як мінімально життєздатний продукт (MVP) для пілотного тестування в Хмельницькому, підтримуючи 1000 користувачів і обробку 5000 звітів на місяць. Розділ охоплює архітектуру системи з детальними діаграмами варіантів використання, класів і станів, процес розробки з DevOps-практиками, комплексне тестування, оцінку результатів і рекомендації щодо впровадження та підтримки. Реалізація базується на дизайні MVP у Figma і спрямована на підвищення інклюзивності міського середовища.

### 4.1 Архітектура програмного забезпечення

Архітектура КФС побудована за клієнт-серверним принципом, що забезпечує модульність, масштабованість і легкість підтримки. Система складається з трьох основних компонентів: фронтенду на React Native, бекенду на Flask і бази даних на PostgreSQL із розширенням PostGIS. Компоненти взаємодіють через HTTPS-запити з JSON-форматом даних і JWT-аутентифікацією для безпеки, що дозволяє чітко розмежувати функціональні обов'язки, оптимізувати продуктивність і спростити розвиток.

Фронтенд є основним інтерфейсом для волонтерів і маломобільних осіб, забезпечуючи надсилання звітів про доступність об'єктів, перегляд інтерактивної мапи на основі OpenStreetMap і побудову адаптованих маршрутів. Бекенд обробляє звіти, верифікує їх за допомогою алгоритмів штучного інтелекту (Random Forest для класифікації, K-Means для кластеризації проблемних зон) і надає REST API для інтеграції з муніципальними службами. База даних ефективно зберігає геолокаційні дані, звіти та профілі користувачів, підтримуючи швидкий пошук і кластеризацію з затримкою 5 мілісекунд.

Архітектура системи візуалізована на рис. 4.1, який ілюструє потік даних від мобільного додатка до сервера та мапи. Для детального представлення

функціоналу, структури та поведінки системи розроблено діаграми варіантів використання, класів і станів, описані нижче. Дизайн MVP із Figma вплинув на структуру інтерфейсу, визначивши екрани мапи, форми звіту та профілю користувача, що забезпечують інтуїтивність і відповідність стандартам доступності WCAG 2.1. Модульна структура дозволяє легко інтегрувати нові функції, наприклад, підтримку IoT-датчиків, що відповідає вимогам економічності та гнучкості MVP.

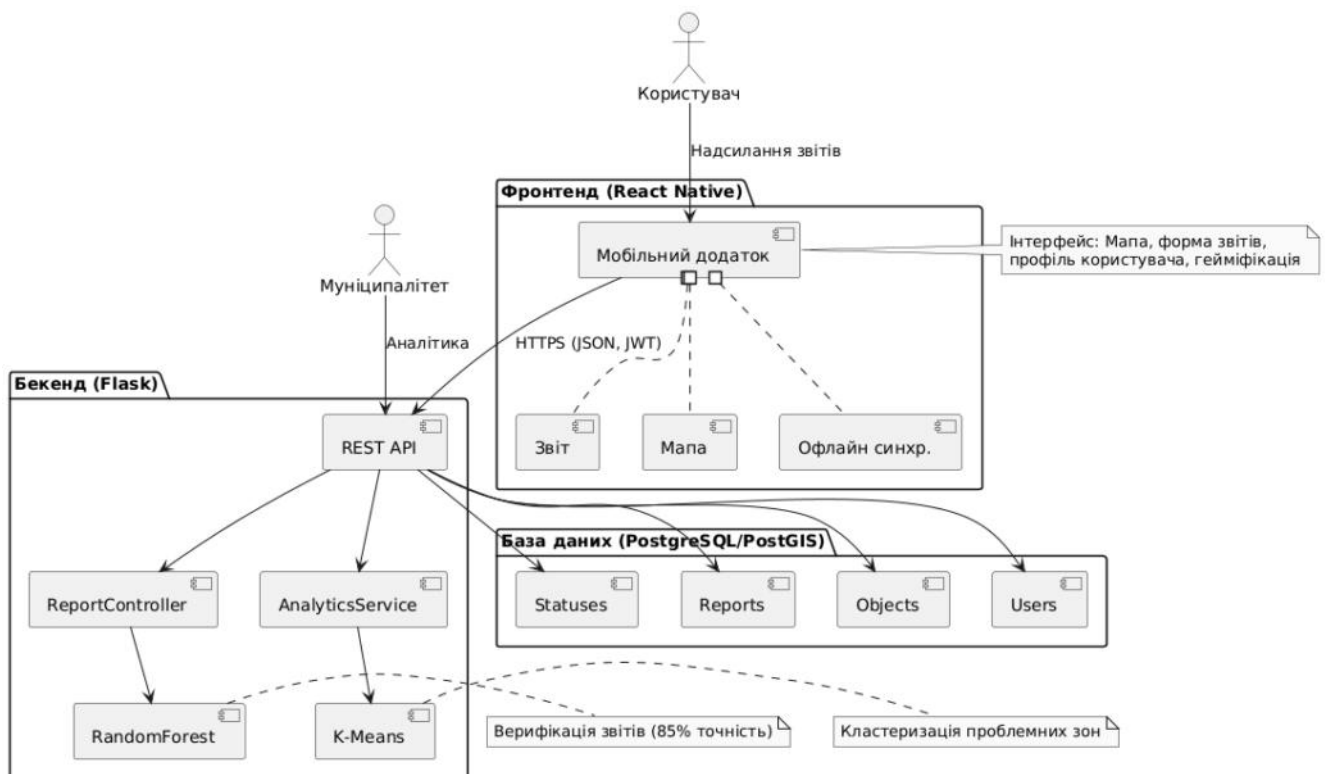


Рисунок 4.1 – Архітектура КФС

## 4.2 Діаграма варіантів використання

Діаграма варіантів використання (рис. 4.2), створена за допомогою PlantUML, відображає ключові функціональні можливості кіберфізичної системи (КФС), призначеної для моніторингу та покращення доступності міської інфраструктури (наприклад, пандусів, тротуарів, зупинок). Вона ілюструє взаємодію трьох основних акторів - Користувача (волонтер або маломобільна особа), Муніципалітету (представник місцевої влади) та Адміна (адміністратор

системи) - із системою через набір сценаріїв використання. Діаграма включає 8 сценаріїв використання, 4 зв'язки типу «include» та «extend», а також 2 примітки для уточнення функціоналу.

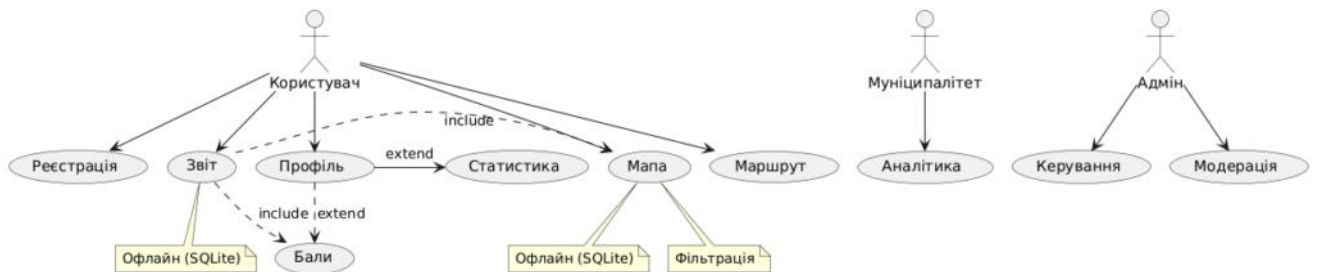


Рисунок 4.2 – Діаграма варіантів використання

Користувач є головним актором, який активно взаємодіє з мобільним додатком. Це може бути волонтер, який додає звіти про стан інфраструктури, або маломобільна особа, яка переглядає доступні об'єкти чи будує маршрути. Користувач виконує широкий спектр дій, таких як реєстрація, надсилання звітів, перегляд мапи, побудова маршрутів і керування профілем, що робить його центральною фігурою в зборі даних і використанні основного функціоналу системи.

Муніципалітет виступає представником місцевої влади, який використовує систему для аналізу даних про інфраструктуру. Через API муніципалітет отримує зведену інформацію про стан об'єктів, що допомагає планувати ремонти чи реконструкції. Його взаємодія обмежується аналітичними функціями, що робить його роль спеціалізованою порівняно з користувачем.

Адмін, або адміністратор системи, відповідає за підтримку її працездатності та якості даних. Цей актор керує профілями користувачів і модерує звіти, забезпечуючи безпеку системи та достовірність інформації. Роль адміна є критичною для управління системою та запобігання зловживанням.

Сценарій «Реєстрація» дозволяє користувачу створювати обліковий запис або входити в систему, вводячи email і пароль. Для забезпечення безпеки застосовується JWT-аутентифікація, яка генерує токен для авторизованих запитів. Ця функція реалізована через простий екран авторизації в мобільному додатку на

React Native, де пароль повинен містити щонайменше 8 символів, а email бути унікальним. Реєстрація є передумовою для доступу до інших функцій, таких як надсилання звітів чи перегляд профілю, забезпечуючи персоналізацію та захист даних користувача.

Сценарій «Звіт» є основною функцією системи, що дає змогу користувачу повідомляти про стан об'єктів інфраструктури, таких як пандуси чи зупинки. Користувач обирає тип об'єкта, вказує його статус (доступний, частково доступний чи недоступний), додає геолокацію через GPS, коментар до 200 символів і фото до 5 МБ. Після верифікації алгоритмом Random Forest звіт зберігається в базі даних PostgreSQL/PostGIS, оновлюючи мапу через зв'язок «include» і нараховуючи бали через зв'язок «include». Ця функція забезпечує збір даних від громади, що є основою для покращення інклюзивності. Зв'язки типу «include» відображають обов'язкові залежності між сценаріями. Сценарій «Звіт» включає «Мапа», оскільки кожен верифікований звіт оновлює дані мапи, додаючи нову мітку з кольоровим кодуванням. Аналогічно, «Звіт» включає «Бали», адже після верифікації звіту користувачу автоматично нараховуються бали, що підтримує гейміфікацію. Ці зв'язки показують, як дії користувача впливають на інші функції системи. Примітка до сценарію «Звіт» уточнює підтримку офлайн-режиму через SQLite. Користувач може створювати звіти без підключення до інтернету, а дані зберігаються локально та синхронізуються після відновлення з'єднання. Ця функція робить систему доступною в умовах обмеженого зв'язку, що особливо важливо для користувачів у віддалених зонах.

Сценарій «Мапа» надає користувачу можливість переглядати інтерактивну мапу на основі OpenStreetMap, де об'єкти позначені кольорами: зелений для доступних, жовтий для частково доступних і червоний для недоступних. Користувач може фільтрувати об'єкти за типом, наприклад, лише тротуари, або за статусом, що спрощує пошук потрібної інформації. Реалізована через екран мапи з панеллю фільтрів, ця функція допомагає оцінювати стан інфраструктури та знаходити доступні об'єкти. Примітка «Фільтрація» вказує на можливість сортування об'єктів відповідно до встановлених параметрів мобільності, примітка

«Офлайн (SQLite)» вказує на можливість переглядати обмежену в обсязі, закешовану частину мапи без доступу до інтернету.

Сценарій «Маршрут» дозволяє користувачу будувати адаптовані маршрути, які уникають недоступних зон. На основі даних мапи система застосовує геолокаційні алгоритми для створення оптимального шляху, враховуючи тип пересування, наприклад, пішки чи на візку. Користувач вводить початкову та кінцеву точки, після чого маршрут відображається на мапі з позначенням доступних ділянок. Ця функція залежить від якості даних, але значно полегшує мобільність маломобільних осіб, пропонуючи безпечні шляхи.

Сценарій «Профіль» відкриває екран, де користувач переглядає свої персональні дані, такі як ім'я та email, а також статистику внеску та накопичені бали. Реалізований через захищений доступ із JWT-токеном, цей сценарій є основою для розширених функцій, таких як перегляд статистики чи балів через зв'язки «extend». Екран профілю в додатку дозволяє користувачу оновити свої дані, переглянути звітність, видалити акаунт.

Сценарій «Профіль» розширює «Статистика», дозволяючи користувачу переглядати детальну статистику внеску як частину роботи з профілем. Так само «Профіль» розширює «Бали», надаючи доступ до операцій із балами, таких як їх обмін на винагороди. Ці зв'язки підкреслюють гнучкість системи в наданні додаткових функцій через профіль.

Сценарій «Статистика» дозволяє користувачу переглядати детальну інформацію про свій внесок, включаючи кількість надісланих звітів, їх статуси (верифіковані чи відхилені) та загальний вплив, наприклад, охоплення зон міста. Доступний через профіль завдяки зв'язку «extend», цей сценарій відображає дані у вигляді графіків або числових показників. Він мотивує користувача, показуючи, як його дії сприяють покращенню інфраструктури.

Сценарій «Бали» підтримує гейміфікацію, дозволяючи користувачу переглядати накопичені бали за верифіковані звіти та обмінювати їх на винагороди, наприклад, знижки від партнерів. Ця функція доступна через профіль (зв'язок «extend») і активується після надсилання звіту (зв'язок «include»). У додатку

відображається кількість балів і доступні винагороди, а обмін можливий за наявності достатньої кількості балів.

Сценарій «Аналітика» призначений для Муніципалітету, який отримує через REST API зведені дані про проблемні зони міста. Алгоритм K-Means групує недоступні об'єкти в кластери та надає статистику за районами у форматі JSON. Муніципалітет може вказати параметри запиту, такі як район чи тип об'єкта, для отримання потрібних даних. Ця функція допомагає планувати інфраструктурні покращення на основі реальних даних.

Сценарій «Керування» дозволяє Адміну управляти профілями користувачів через адмін-панель. Адмін може блокувати порушників, змінювати ролі або видаляти профілі, оновлюючи дані в таблиці Users бази даних. Ця функція потребує авторизації адміна і забезпечує безпеку системи, запобігаючи зловживанням.

Сценарій «Модерація» передбачає перевірку Адміном звітів, які не пройшли автоматичну верифікацію через неоднозначність даних. У адмін-панелі Адмін переглядає звіти, схвалюючи або відхиляючи їх, що оновлює статус у таблиці Reports. Ця функція доступна лише авторизованим адмінам і гарантує високу якість даних у системі.

Діаграма є основою для проектування системи, чітко визначаючи ролі акторів і їхні взаємодії з функціями. На основі сценаріїв використання створено екрани інтерфейсу в Figma, включаючи форми для звітів, мапу з фільтрами та профіль із секціями статистики й балів. API системи, зокрема ендпоінти для надсилання звітів, отримання аналітики та керування профілями, розроблено з урахуванням цих сценаріїв, що забезпечило логічну структуру запитів. Для тестування діаграма стала основою для E2E-тестів, які перевіряють кожен сценарій від реєстрації до модерації.

### 4.3 Загальний опис діаграми класів

Діаграма класів, представлена у додатку г, представляє статичну структуру кіберфізичної системи, яка підтримує моніторинг доступності міської інфраструктури. Вона визначає основні сутності, їхні атрибути, методи та зв'язки,

необхідні для реалізації сценаріїв використання, описаних у діаграмі варіантів використання.

Система включає мобільний додаток (React Native) для користувачів, API (Flask) для аналітики та адмін-панель для модерації, із даними, що зберігаються в PostgreSQL/PostGIS і кешуються в SQLite для офлайн-режиму. Діаграма класів охоплює сутності для управління користувачами, звітами, об'єктами інфраструктури, маршрутами, балами, статистикою та аналітикою, забезпечуючи модульність і масштабованість. Зв'язки між класами відображають асоціації, композицію та залежності, що підтримують функціонал сценаріїв використання.

Клас «User» є центральною сутністю, яка представляє всіх акторів системи: користувачів, муніципалітет і адмінів. Він містить атрибути, такі як унікальний ідентифікатор, email (унікальний, для авторизації), пароль (захешований за допомогою bcrypt), ім'я, роль (enum: user, municipality, admin) і JWT-токен для сеансів. Цей клас підтримує сценарій «Реєстрація» через методи створення облікового запису (створення користувача з валідацією email і пароля) і генерації токена (створення JWT після успішного входу). Для сценарію «Профіль» клас надає метод отримання даних користувача (повертає роль, дані). Для сценарію «Керування» клас включає методи для адмінів, такі як блокування користувача (змінює статус на заблоковано), зміна ролі (оновлює роль для підвищення прав) і видалення профілю (видаляє запис із бази). Цей клас є основою для авторизації та персоналізації в системі.

Клас «Report» відповідає сценарію «Звіт» і представляє звіти про стан інфраструктури, створені користувачами. Він містить атрибути, такі як унікальний ідентифікатор, ідентифікатор користувача (посилання на User), тип об'єкта (enum: ramp, sidewalk, stop тощо), статус (enum: accessible, partially\_accessible, inaccessible), геолокація (тип POINT у PostGIS), коментар (рядок до 200 символів), фото (посилання на файл у хмарному сховищі), дата створення і статус верифікації (enum: pending, verified, rejected). Методи класу включають створення звіту (валідує дані, зберігає в SQLite для офлайн-режиму), синхронізацію з сервером (передає дані в PostgreSQL після відновлення з'єднання), верифікацію (застосовує

Random Forest для автоматичної перевірки) і оновлення мапи (додає об'єкт через зв'язок «include» до класу InfrastructureObject). Цей клас підтримує офлайн-режим, зазначений у примітці, і нарахування балів через зв'язок «include» до класу Points.

Клас «InfrastructureObject» представляє об'єкти інфраструктури, відображені на мапі, і відповідає сценарію «Мапа». Він містить атрибути, такі як унікальний ідентифікатор, тип об'єкта (enum: ramp, sidewalk, stop тощо), статус (enum: accessible, partially\_accessible, inaccessible), геолокація (тип POINT у PostGIS) і посилання на пов'язаний звіт (ідентифікатор із Report). Методи класу включають отримання об'єктів (повертає список об'єктів із фільтрацією за типом чи статусом, як зазначено в примітці «Фільтрація»), кешування в SQLite (зберігає дані для офлайн-доступу, як зазначено в примітці «Офлайн (SQLite)»), оновлення об'єкта (змінює статус або геолокацію на основі звіту) і відображення на мапі (генерує дані для OpenStreetMap із кольоровим кодуванням: зелений, жовтий, червоний). Цей клас є основою для візуалізації та фільтрації об'єктів.

Клас «Route» відповідає сценарію «Маршрут» і представляє адаптовані маршрути для користувачів. Він містить атрибути, такі як унікальний ідентифікатор, ідентифікатор користувача (посилання на User), початкова точка (координати), кінцева точка (координати), тип пересування (enum: walking, wheelchair) і список об'єктів (посилання на InfrastructureObject, що формують маршрут). Методи класу включають побудову маршруту (застосовує геолокаційний алгоритм, A\*, для вибору доступних ділянок), збереження маршруту (кешує в SQLite для офлайн-доступу) і відображення маршруту (генерує дані для мапи). Цей клас залежить від InfrastructureObject для даних про доступність.

Клас «Points» відповідає сценарію «Бали» і підтримує гейміфікацію. Він містить атрибути, такі як унікальний ідентифікатор, ідентифікатор користувача (посилання на User), кількість балів (ціле число) і історію транзакцій (масив записів про нарахування чи обмін). Методи класу включають нарахування балів (додає бали після верифікації звіту, зв'язок «include» від Report), обмін балів (зменшує

кількість балів після вибору винагороди, якщо балів достатньо) і отримання балів (повертає поточний баланс для профілю, зв'язок «extend» від User).

Клас «Statistics» відповідає сценарію «Статистика» і відображає внесок користувача. Він містить атрибути, такі як унікальний ідентифікатор, ідентифікатор користувача (посилання на User), кількість звітів, розподіл статусів (об'єкт із кількістю верифікованих, відхилених звітів) і метрики впливу (наприклад, охоплені зони міста). Методи класу включають генерацію статистики (обчислює дані на основі звітів користувача) і відображення статистики (повертає дані для графіків, зв'язок «extend» від User). Цей клас збирає актуальну статистику про доступність інфраструктури та статистику системи.

Клас «Analytics» відповідає сценарію «Аналітика» для Муніципалітету. Він містить атрибути, такі як унікальний ідентифікатор, параметри запиту (район, тип об'єкта) і результати кластеризації (JSON із кластерами від K-Means). Методи класу включають обробку запиту (генерує аналітичні дані на основі InfrastructureObject), кластеризацію (застосовує K-Means до групування недоступних об'єктів) і повернення результатів (форматує дані в JSON для API).

Клас «Moderation» відповідає сценарію «Модерація» і управляє перевіркою звітів. Він містить атрибути, такі як унікальний ідентифікатор, ідентифікатор звіту (посилання на Report), ідентифікатор адміна (посилання на User), рішення (enum: approve, reject) і коментар (рядок для пояснення). Методи класу включають створення запису модерації (ініціює перевірку звіту), оновлення статусу звіту (змінює статус у Report на основі рішення) і отримання черги звітів (повертає список звітів зі статусом pending для адмін-панелі).

Клас «User» має асоціацію один-до-багатьох із «Report», оскільки один користувач може створювати багато звітів, має асоціацію один-до-одного з «Points» і «Statistics», оскільки кожен користувач має один запис балів і статистики. Клас «Report» має асоціацію один-до-одного з «InfrastructureObject», оскільки кожен звіт оновлює або створює об'єкт на мапі. Клас «InfrastructureObject» входить до композиції з «Route», оскільки маршрут складається з доступних об'єктів. Клас «Report» також має асоціацію один-до-одного з «Moderation», оскільки кожен звіт

може бути перевірений адміном. Клас «Analytics» залежить від «InfrastructureObject», оскільки аналітика базується на даних об'єктів. Ці зв'язки забезпечують логіку сценаріїв використання.

#### 4.4 Діаграма станів

Діаграма станів відображає еволюцію звіту як основного механізму збору даних про доступність міської інфраструктури в межах кіберфізичної системи, що слугує інструментом для підвищення інклюзивності міського середовища. Звіт містить інформацію, яку користувачі надають про фізичні об'єкти, такі як пандуси, ліфти чи зупинки громадського транспорту, включаючи їхній тип, стан доступності, географічне розташування, текстові коментарі та візуальні дані. Ця діаграма структурована як послідовність ключових станів, кожен із яких відображає етап обробки даних, від їхнього створення до інтеграції в систему або архівації, що дозволяє систематично аналізувати поведінку даних у контексті моніторингу інфраструктури. Вона підкреслює роль користувачів як активних учасників процесу збору даних, водночас забезпечуючи механізми перевірки достовірності, що є критично важливим для якості інформації, на якій базується система.

Життєвий цикл користувача в КФС відображає його взаємодію з системою як активного учасника моніторингу інфраструктури. Початковий стан «Неzareєстрований» характеризує користувача, який ще не має доступу до функціоналу системи, що підкреслює важливість авторизації для забезпечення безпеки даних і залучення лише перевірених учасників. Перехід до стану «Зареєстрований» відбувається після введення ідентифікаційних даних, таких як електронна пошта та пароль, що дозволяє користувачу отримати доступ до створення звітів і маршрутів. У стані «Активний» користувач повноцінно взаємодіє з системою, додаючи звіти про стан інфраструктури, будуючи маршрути чи переглядаючи профіль із накопиченими балами за внесок. Система може перевести користувача в стан «Заблокований», якщо адміністратор виявить порушення, наприклад, подання хибних звітів, що обмежує доступ до функцій, але зберігає

профіль для потенційного відновлення. Ці стани підкреслюють роль користувача як агента змін у створенні доступного міського середовища, а також важливість контролю якості внеску.

Маршрут як компонент системи відображає процес створення адаптованих шляхів для користувачів, що є ключовим для забезпечення мобільності осіб із обмеженими можливостями. У стані «Створюється» користувач визначає початкову та кінцеву точки маршруту, а також тип пересування, наприклад, пішки чи на візку, що дозволяє системі врахувати специфіку доступності інфраструктури. Після обробки даних маршрут переходить у стан «Збережений», де він стає доступним для використання, включаючи можливість перегляду в умовах обмеженого мережевого доступу, що забезпечує безперервність навігації. У стані «Відображений» маршрут візуалізується на карті з позначками доступних об'єктів, таких як пандуси чи ліфти, що допомагає користувачу орієнтуватись у просторі. Завершення маршруту, коли користувач досягає кінцевої точки або скасовує навігацію, переводить його в стан «Завершений». Ці стани підкреслюють значення маршрутів для забезпечення автономії користувачів і демонструють інтеграцію даних про інфраструктуру в практичні рішення для повсякденного життя.

Об'єкт інфраструктури в системі представляє фізичні елементи міського середовища, які моніторяться для оцінки їхньої доступності. У стані «Створений» об'єкт додається до бази даних на основі верифікованого звіту, включаючи інформацію про тип, стан і географічне розташування, що формує основу для картографування доступності. У стані «Активний» об'єкт відображається на карті з відповідним маркуванням, наприклад, кольоровим кодуванням для позначення рівня доступності, що дозволяє користувачам планувати свої переміщення. Надходження нового звіту переводить об'єкт у стан «Оновлений», де змінюються його характеристики, такі як статус доступності, що забезпечує актуальність даних у динамічному міському середовищі. Якщо об'єкт стає неактуальним, наприклад, через фізичне знищення чи реконструкцію, він переходить у стан «Архівований», зберігаючи історичні дані для аналітичних цілей, таких як оцінка змін у

доступності з часом. Ці стани підкреслюють важливість безперервного моніторингу та оновлення даних про інфраструктуру для підтримки інклюзивності.

Діаграма станів для звіту, зображена на рисунку 4.3, має фундаментальне значення для кіберфізичної системи моніторингу доступності міської інфраструктури, оскільки вона структурує процес збору, верифікації та інтеграції даних, що є основою для створення інклюзивного міського середовища. Звіт виступає первинним джерелом інформації, яке дозволяє користувачам, зокрема особам із обмеженими можливостями, повідомляти про стан інфраструктури, таких як пандуси чи зупинки, що сприяє підвищенню їхньої мобільності та якості життя.

Життєвий цикл звіту розпочинається зі стану «Створений», коли зареєстрований користувач, який має авторизований доступ до системи, ініціює процес фіксації даних про об'єкт інфраструктури. Цей стан відображає первинний внесок користувача, який описує стан об'єкта, наприклад, чи є пандус доступним, і додає фотографії для підтвердження.

Важливим аспектом є можливість створення звіту в умовах відсутності мережевого з'єднання, що забезпечує гнучкість у зборі даних у реальному часі, навіть у віддалених або слабо покритих інтернетом районах. Перехід до стану «Надісланий» відбувається, коли зв'язок відновлюється, і дані передаються до центральної бази, де вони стають доступними для подальшої обробки. Цей етап підкреслює адаптивність системи до мінливих умов доступу до мережі, що є суттєвим для її ефективності в міському контексті.

У стані «Очікує верифікації» система застосовує автоматичний аналіз даних, використовуючи алгоритмічні методи для оцінки достовірності звіту на основі географічних координат, візуальних матеріалів та опису. Цей процес дозволяє ідентифікувати хибні звіти, наприклад, ті, що містять сфальсифіковані зображення чи невідповідні дані про місце розташування, і автоматично спрямовувати їх до стану «Відхилений». Такий підхід зменшує навантаження на людські ресурси та підвищує швидкість обробки, що є важливим для масштабного моніторингу. Якщо звіт визнано достовірним, він переходить у стан «Верифікований», де інформація інтегрується в загальну базу даних об'єктів інфраструктури, оновлюючи карти

доступності для всіх користувачів. Цей стан також супроводжується нарахуванням балів автору звіту, що мотивує громадянську активність і сприяє краудсорсинговому підходу до збору даних.

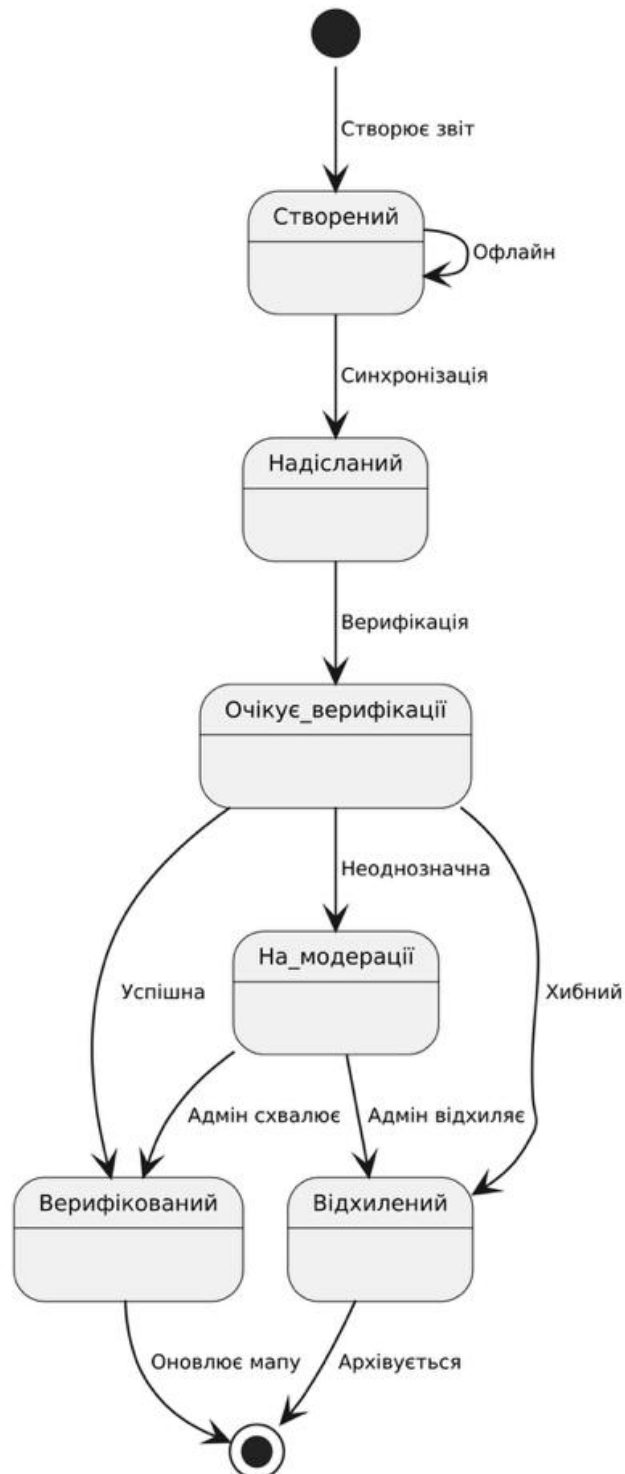


Рисунок 4.3 – Діаграма станів звіту

У випадку, коли автоматична верифікація не може однозначно визначити достовірність, звіт переходить у стан «На модерації», де залучається адміністратор для ручної оцінки. Цей етап відображає гібридний підхід до забезпечення якості даних, поєднуючи автоматизацію з людським судженням, що дозволяє вирішувати складні випадки, коли алгоритми не дають чіткої відповіді. Адміністратор може схвалити звіт, переводячи його до «Верифікований», або відхилити, спрямовуючи до «Відхилений». Стан «Відхилений» завершує цикл для хибних або неоднозначних звітів, які архівуються для подальшого аналізу, що сприяє вдосконаленню алгоритмів верифікації та розумінню помилкових патернів у даних. Таким чином, діаграма відображає замкнутий цикл обробки інформації, який забезпечує безперервне вдосконалення системи.

Розробка КФС проводилася за методологією Agile з двотижневими ітераціями, що дозволило оперативно адаптувати функціонал до потреб користувачів на основі зворотного зв'язку. Процес включав вісім етапів: бізнес-аналіз, проектування архітектури, створення дизайну, розробку фронтенду, бекенду, бази даних, інтеграцію та тестування з оптимізацією. Для автоматизації розгортання та забезпечення якості використано DevOps-практики: Git для контролю версій, Docker для контейнеризації, GitHub Actions для CI/CD.

Дизайн MVP із Figma став основою для створення інтуїтивного інтерфейсу, а метрики користувацького досвіду (UX) допомогли оптимізувати юзабіліті. Кожен етап супроводжувався створенням технічної документації, що полегшує підтримку та масштабування.

Таблиця 4.1 підсумовує етапи розробки, вказуючи тривалість, ключові дії, інструменти та результати.

#### 4.5 Розробка програмного забезпечення

Бізнес-аналіз тривав два тижні та був спрямований на формування бачення продукту. Проведено 25 інтерв'ю з цільовою аудиторією (10 маломобільних осіб, 10 волонтерів) за допомогою Google Forms і Miro для структурування даних.

Користувачі наголосили на потребі в інтуїтивному інтерфейсі, офлайн-режимі та швидкій верифікації звітів. Аналіз аналогів виявив обмеження: рідкісне оновлення даних (раз на місяць чи квартал) і відсутність автоматизації. Результатом став документ із вимогами, що визначив специфікацію MVP: підтримка 1000 користувачів, обробка 5000 звітів на місяць, інтерактивна мапа з мітками доступності, офлайн-режим і API для муніципалітетів. Документ включав функціональні (збір, верифікація, відображення звітів) і нефункціональні вимоги (затримка до 5 секунд, WCAG 2.1), які стали основою для всіх етапів.

Етап побудови архітектури тривав один тиждень і був присвячений вибору технологій і проектуванню системи. Використано Draw.io та Enterprise Architect для створення UML-діаграм варіантів використання, класів і станів. React Native обрано для фронтенду через кросплатформність, що скоротила час розробки на 30% порівняно з нативними Swift чи Kotlin. Flask для бекенду забезпечує легковаговість і сумісність із бібліотеками III (scikit-learn). PostgreSQL із PostGIS підтримує швидкий геолокаційний пошук (5 мс). Архітектура спроектована як модульна клієнт-серверна, що полегшує додавання функцій, наприклад, IoT-інтеграцію. Результатом стала архітектурна схема, яка мінімізувала технічні ризики.

Створення дизайну тривало два тижні та було ключовим для забезпечення юзабіліті й доступності. На основі вимог розроблено прототипи в Figma, включаючи екрани мапи, форми звіту (7 категорій об'єктів: тротуари, зупинки, пандуси тощо) і профілю користувача зі статистикою. UX-тестування з 12 користувачами (5 маломобільних осіб) за допомогою Lookback вимірювало час виконання завдань і частоту помилок. Початковий час заповнення звіту склав 35 секунд, але 20% користувачів плутали іконки статусів (доступний, частково доступний, недоступний). Оптимізація включала збільшення розміру кнопок до 48 пікселів, контрастності до 4.5:1 і зменшення обов'язкових полів до трьох (тип, статус, геолокація), що скоротило час до 30 секунд. Результати UX-тестування: 90% успішних завдань, 85% позитивних відгуків. Фінальний дизайн відповідає WCAG 2.1 і отримав схвалення тестових користувачів.

Таблиця 4.1 – Етапи розробки програмного забезпечення

Етап розробки	Тривалість	Ключові дії та інструменти	Результат
Бізнес-аналіз	2 тижні	Інтерв'ю, аналіз аналогів (Miro, Google Forms)	Специфікація MVP, документ із вимогами
Побудова архітектури	1 тиждень	Вибір технологій, UML-діаграми (Draw.io, Enterprise Architect)	Архітектурна схема, технологічний стек
Створення дизайну	2 тижні	Прототипи в Figma, UX-тестування (Lookback)	Дизайн MVP у Figma, UX-метрики
Розробка фронтенду	4 тижні	Кодування React Native, Git, Docker, VS Code	Мобільний додаток із мапою, формою звіту, профілем
Розробка бекенду	3 тижні	Кодування Flask, API, Git, Docker, PyCharm	REST API з 5 ендпоїнтами, верифікація звітів
Розробка бази даних	2 тижні	Налаштування PostgreSQL/PostGIS,	База даних із геозапитами.
Інтеграція компонентів	1 тиждень	З'єднання, тестування API (Postman), GitHub Actions	Повноцінна клієнт-серверна система
Тестування та оптимізація	2 тижні	Модульне, інтеграційне тестування (Jest, pytest), документація	Оптимізована система, технічна документація

Розробка фронтенду тривала чотири тижні та була зосереджена на створенні кросплатформного додатка на React Native для iOS 13+ і Android 9+, що охоплює 95% цільової аудиторії. Використано Git для контролю версій, Docker для ізольованого середовища розробки та VS Code як IDE. Бібліотеки react-native-maps, react-navigation і react-native-sqlite-storage забезпечили інтеграцію з OpenStreetMap, навігацію між екранами та локальне кешування. Відповідно до дизайну Figma,

реалізовано екран мапи, який відображає до 500 об'єктів із мітками (зелений - доступний, жовтий - частково доступний, червоний - недоступний), підтримуючи зум (10-18) і фільтрацію за типами. Форма звіту дозволяє вибрати категорію, додавати коментарі (до 200 символів), фото (до 5 МБ) і визначати геолокацію з точністю 5 метрів. Екран профілю відображає статистику звітів і бали, сприяючи гейміфікації.

Офлайн-режим реалізований через SQLite, кешуючи 10 МБ даних (до 100 звітів), що критично для регіонів із нестабільним інтернетом. Кодування виконано модульно. Розмір APK - 30 МБ, час рендерингу мапи - 3 секунди на пристроях із 4 ГБ ОЗП. GitHub Actions автоматизував збірку APK, скоротивши час із 10 до 5 хвилин. Додаток протестовано на 10 пристроях, підтвердивши стабільність.

Розробка бекенду тривала три тижні, використовуючи Flask, Git, Docker і PyCharm. Flask обрано через легковаговість і сумісність із scikit-learn. Бекенд обробляє JSON-звіти у форматі {"object": "sidewalk", "status": "inaccessible", "location": {"lat": 49.42, "lon": 26.98}, "photo": "base64", "timestamp": "2025-03-14T10:00:00Z"}, верифікує їх за допомогою Random Forest і кластеризує проблемні зони алгоритмом K-Means. Реалізовано п'ять REST API ендпоінтів:

1) POST /api/reports: Приймає звіт, валідує формат, верифікує, повертає {"status": "verified"} або {"status": "rejected"}. Час обробки - 5 секунд.

2) GET /api/objects: Повертає об'єкти в радіусі 1 км за параметрами lat, lon. Затримка - 5 мс.

3) GET /api/analytics: Надає статистику проблемних зон і звітів, доступ через JWT, оновлення щоденно.

4) PATCH /api/reports/:id: Оновлює статус звіту після реагування муніципалітету.

5) GET /api/users/:id: Повертає статистику користувача (звіти, бали).

Розробка бази даних тривала два тижні, використовуючи PostgreSQL із PostGIS, pgAdmin і Git. Схема включає таблиці: objects (500 записів), reports (до 5000), users (до 1000), statuses (довідник). PostGIS забезпечує геозапити за 5 мс. Docker-контейнер із PostgreSQL спростив тестування через GitHub Actions.

Інтеграція компонентів тривала один тиждень, об'єднуючи фронтенд, бекенд і базу даних. Використано Postman для тестування API, GitHub Actions для розгортання. Налаштовано HTTPS-запити з JWT-аутентифікацією. Цикл обробки звіту (надсилання-верифікація-відображення) займає 10 секунд. Виявлено втрату 5% офлайн-звітів, виправлено чергою синхронізації на AsyncTask. Технічна документація API створена через Swagger, що полегшило інтеграцію з муніципальними службами.

Тестування та оптимізація тривали два тижні, включаючи модульне, інтеграційне, стресове, E2E, мануальне та регресійне тестування. Виконувалось з акцентом на безвідмовність системи та роботу під навантаженням, також проводилось тестування усіх варіантів використання програми користувачем, було впроваджено низку автоматизованих тестів. Використано Jest, pytest і GitHub Actions для автоматизації. Створено технічну документацію: посібник розробника (Markdown, 50 сторінок), специфікація API (Swagger), інструкції розгортання (Docker Compose). Оптимізація через Redis знизила помилки з 2% до 0,5% при 500 користувачах. Система готова до пілотного тестування.

#### 4.6 Тестування програмного забезпечення

Тестування КФС було багатограним, охоплюючи модульне, інтеграційне, end-to-end (E2E), мануальне, стресове та регресійне тестування, щоб забезпечити відповідність функціональним (збір, верифікація, відображення звітів) і нефункціональним (продуктивність, доступність) вимогам. Тестування проводилося на AWS EC2 t3.micro, 10 пристроях iOS/Android і з залученням користувачів. Кожен тип мав чіткі критерії виконання.

Модульне тестування перевіряло ізольовані компоненти з Jest (фронтенд) і pytest (бекенд), охоплюючи фільтрацію дубльованих звітів, верифікацію (Random Forest), кластеризацію (K-Means), рендеринг мапи, за наступними критеріями: покриття коду  $\geq 80\%$ , успішне виконання всіх тестів, час виконання тесту  $\leq 100$  мс.

Тестові сценарії включають кілька ключових перевірок. По-перше, фільтрація дубльованих звітів: при поданні на вхід 100 звітів очікується скорочення їх кількості на 20%, тобто до 80 унікальних. По-друге, верифікація точності моделі Random Forest, яка повинна становити 85% на вибірці з 200 звітів. І по-третє, перевірка рендерингу мапи - 500 міток мають відобразитися за 3 секунди при використанні 2 ГБ оперативної пам'яті.

Результати: покриття 85%, усі 150 тестів успішні. Виправлено помилку в обробці порожніх коментарів за 2 дні.

Інтеграційне тестування перевіряло взаємодію фронтенду, бекенду та бази з Postman і реальними пристроями за такими критеріями: успішне виконання 95% сценаріїв, цикл обробки звіту  $\leq 10$  секунд, відсутність втрат даних.

Сценарії включають кілька основних перевірок. Перший - повний цикл роботи зі звітом: надсилання, верифікація та відображення мають виконуватися за 10 секунд з 100% успішністю. Другий – офлайн-синхронізація, під час якої з 50 звітів виправляється втрата даних у розмірі 5%. Третій - робота API-аналітики, що передбачає повернення статистичних даних за 2 секунди. В результаті 98% із 50 сценаріїв, усі критичні помилки виправлено.

E2E-тестування імітувало взаємодію користувача з Cypress. Критерії наступні: успішне виконання 90% сценаріїв, відсутність збоїв.

Сценарії включають створення звіту, яке займає 25 секунд і має 95% успішності; побудову маршруту з 90% точністю, де затримки було усунуто за допомогою кешування; а також аналітику, що виконується за 3 секунди з 100% успіхом. Загалом із 30 протестованих сценаріїв вдалося досягти 92% успішності, при цьому було виправлено затримки рендерингу.

Мануальне тестування проводили 10 користувачів (5 волонтерів, 3 маломобільні особи, 2 муніципальні представники). Критерії успішного виконання: 80% позитивних відгуків, відповідність WCAG 2.1., виявлення критичних дефектів. Завдання включають заповнення звіту за 30 секунд, фільтрацію об'єктів та забезпечення офлайн-режиму. Як результат, 85% позитивних відгуків, UX-

метрики: 90% успішних завдань, середній час 30 секунд. Виправлено шрифт у профілі, розмір кнопок по всьому застосунку.

Стрессове тестування оцінювало продуктивність із Locust за такими параметрами: затримка  $\leq 5$  секунд при 500 користувачах, помилки  $< 5\%$ .

Сценарії передбачають, що при навантаженні в 100 користувачів час відповіді складає 5 секунд з нульовим відсотком помилок. При збільшенні до 500 користувачів час зростає до 8 секунд, а помилки становлять 0,5% після впровадження кешування через Redis. Для 1000 користувачів система стабільно працює на інстансі t3.medium.

Результати: стабільність до 500 на t3.micro, 1000 - на t3.medium.

Регресійне тестування перевіряло функціонал після виправлень і мало на меті 100% успішних тестів виконаних за 5 хвилин.

Результати: 100 тестів успішно. Тестування підтвердило стабільність, продуктивність і доступність.

#### 4.7 Висновок

Розділ представляє детальну реалізацію КФС для інтерактивного моніторингу доступності міської інфраструктури, розробленої як MVP для пілотного тестування в Хмельницькому з підтримкою 1000 користувачів і обробкою 5000 звітів на місяць. Архітектура, побудована на клієнт-серверному підході з компонентами фронтенду на React Native, бекенду на Flask і бази даних PostgreSQL/PostGIS, забезпечила ефективну взаємодію через HTTPS-запити з JSON-форматом і JWT-аутентифікацією, реалізуючи модульність і масштабованість. Діаграми варіантів використання, класів і станів, представлені на рисунках 4.2-4.3, чітко визначили ролі акторів - Користувача, Муніципалітету та Адміна - та ключові сценарії, такі як Звіт, Мапа, Маршрут і Аналітика, що сприяє підвищенню інклюзивності міського середовища.

Розробка, проведена за методологією Agile з двотижневими ітераціями та DevOps-практиками, включала вісім етапів: бізнес-аналіз, проектування

архітектури, створення дизайну, розробку фронтенду, бекенду, бази даних, інтеграцію та тестування з оптимізацією, що задокументовано в таблиці 4.1. Дизайн MVP у Figma, оптимізований через UX-тестування, забезпечив відповідність стандартам WCAG 2.1, зменшивши час заповнення звіту до 30 секунд при 90% успішних завдань. Тестування, що охопило модульні, інтеграційні, E2E, мануальні, стресові та регресійні сценарії, підтвердило стабільність системи з часом обробки звіту 10 секунд і помилками 0,5% при 500 користувачах, що відповідає заявленим вимогам.

На поточному етапі система функціонує як MVP, основною метою якого є збір даних про доступність об'єктів міської інфраструктури через ручні звіти користувачів. Такий підхід дозволяє швидко налагодити взаємодію з реальними користувачами та сформувати первинну базу даних. Однак, з огляду на принципи масштабованості, система спроектована з можливістю подальшого розширення функціоналу, зокрема - за рахунок інтеграції компонентів IoT. Архітектура системи передбачає модульність як на рівні обробки даних, так і на рівні взаємодії з джерелами інформації. Це дозволяє без кардинальних змін додати нові джерела вхідних даних, зокрема сенсорні пристрої, які встановлюються безпосередньо на елементи міської інфраструктури, наприклад, для автоматичного визначення відповідності ухилу пандусів до встановлених нормативів або зовнішні API, як Google Maps, для насичення системи більшими обсягами корисної інформації. Усі дані проходять одну логіку перевірки, фільтрації та аналізу за допомогою модулів штучного інтелекту, що вже реалізовані в MVP. Система спроектована з урахуванням масштабування як по географії (міста, райони), так і по типах інфраструктурних об'єктів. Стандартизовані формати зберігання даних та REST API дають змогу інтегрувати систему з іншими міськими рішеннями - наприклад, транспортними навігаторами, соціальними службами або системами Smart City.

Результати реалізації демонструють створення ефективного інструменту, який інтегрує краудсорсинг, алгоритми штучного інтелекту та геолокаційні дані для підтримки муніципалітетів у плануванні інфраструктури та надання користувачам адаптованих маршрутів із мотивацією через гейміфікацію.

## ВИСНОВОК

У рамках дипломної роботи розроблено кіберфізичну систему для інтерактивного моніторингу доступності міської інфраструктури, що забезпечує автоматизований збір, верифікацію та візуалізацію даних про стан об'єктів міського середовища з урахуванням потреб маломобільних груп населення. Система, реалізована як мінімально життєздатний продукт, інтегрує краудсорсингові дані з алгоритмами штучного інтелекту, забезпечуючи динамічне оновлення інформації та формування адаптивних маршрутів. Модульна клієнт-серверна архітектура підтримує масштабування та перспективну інтеграцію з сенсорними мережами для моніторингу в реальному часі.

Розробка охопила повний цикл створення програмного забезпечення: від бізнес-аналізу та проектування до впровадження та багатогранного тестування, що підтвердило відповідність функціональним і нефункціональним вимогам, зокрема стандартам доступності. Наукова новизна полягає в синтезі кіберфізичних принципів із методами інтелектуального аналізу даних, що забезпечує гнучкість і адаптивність системи порівняно зі статичними аналогами. Практична цінність роботи проявляється у створенні універсального інструменту для підвищення інклюзивності міського простору, який сприяє автономності користувачів і оптимізації управлінських процесів муніципалітетів.

Для подальшого вдосконалення системи рекомендується розширення функціоналу шляхом інтеграції автоматизованих джерел даних, посилення алгоритмів верифікації та проведення економічного аналізу для оцінки ефективності впровадження. Розроблене рішення формує технологічну основу для трансформації міських екосистем у напрямку створення розумного та безбар'єрного середовища, відповідаючи сучасним викликам урбанізації та соціальної інтеграції.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. United Nations. World Urbanization Prospects: The 2018 Revision. New York : UN, 2018. 126 с.
2. Gao Y., Janssen M. The open data canvas—analyzing value creation from open data. *Digital Government: Research and Practice*. 2022. Т. 3. С. 1–15.
3. Дослідження доступності міських просторів: звіт за 2023 рік. Київ : Міністерство інфраструктури України, 2023. 45 с. URL: <https://mtu.gov.ua/files/doslidzhennya-dostupnosti-miskyh-prostoriv.pdf> (дата звернення: 03.02.2025).
4. Fortino G., Savaglio C., Palau C. E. та ін. Smart Cities of the Future as Cyber Physical Systems: Challenges and Enabling Technologies. *Sensors*. 2021. Т. 21, № 10. 3349.
5. Badea R. A., Cojocaru S. B., Popescu D. A Survey of Internet of Things and Cyber-Physical Systems: Standards, Algorithms, Applications. *Information*. 2023. Т. 14, № 7. С. 423–446.
6. Панченко О. В. Моделювання динамічних мереж міської інфраструктури з використанням кіберфізичних систем. *Електронний архів ТНТУ*. 2022. № 5. С. 45–58.
7. Lee J., Davari H., Singh J. Artificial Intelligence in Cyber-Physical Systems: A Survey. *Applied Sciences*. 2021. Т. 11, № 6. 2458.
8. Bibri S. E., Krogstie J. Smart sustainable cities of the future: an extensive interdisciplinary literature review. *Sustainable Cities and Society*. 2017. Т. 31. С. 183–212.
9. Соціологічне дослідження доступності міського середовища для маломобільних груп. Київ : Інститут соціальних досліджень, 2023. 58 с.
10. Karger E., Bree T. Building the Smart City of Tomorrow: A Bibliometric Analysis of Artificial Intelligence in Urbanization. *Urban Science*. 2024. 36 с.
11. Lu Y., Lu Y., Batty M., Chen Y. Understanding smart cities: an integrative framework. *Cities*. 2019. Т. 89. С. 80–91.

12. Ahvenniemi H., Huovila A., Pinto-Seppä I., Airaksinen M. What are the differences between sustainable and smart cities? *Cities*. 2017. Т. 60. С. 234–245.
13. Іванчук І. І., Козир В. С. Модель управління метаданими об'єктів кіберфізичних систем на основі юзер даних. *Електронний архів ТНТУ*. 2023. № 3. С. 67–80.
14. Zanella A., Bui N., Castellani A. та ін. IoT-Enabled Smart Cities: A Review. *IEEE Internet of Things Journal*. 2014. Т. 1, № 1. С. 22–32.
15. Haklay M., Weber P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*. 2008. Т. 7, № 4. С. 12–18.
16. Sala J. M., Domingo A., Vilajosana X. Smart Cities: The Barcelona Experience. *IEEE Communications Magazine*. 2019. Т. 57, № 3. С. 78–84.
17. Kumar A., Sharma R. Cyber-Physical Systems: A Smart City Perspective. *CloudFront*. 2020. 25 с.
18. Lee E., Seshia S. Cyber-Physical Systems: Foundations, Principles. *IEEE Communications Magazine*. 2018. 22 с.
19. Al-Huthaifi R., Li S., Huang W. A Comprehensive Survey on IoT. *ScienceDirect*. 2023. Т. 628. С. 1–22.
20. Державна стратегія регіонального розвитку на 2021–2027 : Постанова Кабінету Міністрів України від 05.08.2020 р. № 695. *Офіційний вісник України*. 2020. № 65. С. 123–145.
21. Vahdad H. A Survey on Crowdsourcing Applications in Smart Cities. *Recent Innovations in Artificial Intelligence and Smart Applications*. 2022. С. 239–253.
22. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of things (IoT): a vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013. Т. 29. С. 1645–1660.
23. Alaya M. B., Medjiah S., Monteil T., Drira K. Toward semantic interoperability in oneM2M architecture. *IEEE Communications Magazine*. 2015. Т. 53, № 12. С. 35–41.
24. Aloï G., Caliciuri G., Fortino G. та ін. A mobile multi-technology gateway to enable IoT interoperability. *IEEE First International Conference on Internet-of-Things Design and Implementation*. 2016. С. 259–264.

25. Fortino G., Di Fatta G., Ochoa S. F., Palau C. E. Engineering Future Interoperable and Open IoT Systems. Amsterdam : Elsevier, 2017. 300 c.
26. Fortino G., Guerrieri A., Lacopo M., Lucia M., Russo W. An agent-based middleware for cooperating smart objects. International Conference on Practical Applications of Agents and Multi-Agent Systems. Berlin : Springer, 2013. C. 387–398.
27. Fortino G., Guerrieri A., Russo W., Savaglio C. Towards a development methodology for smart object-oriented IoT systems: a metamodel approach. IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2015. C. 1297–1302.
28. Pivoto D. G., de Almeida L. F., da Rosa Righi R. та ін. Cyber-physical systems architectures for industrial internet of things numerous applications in Industry 4.0: a literature review. *Journal of Manufacturing Systems*. 2021. T. 58. C. 176–192.
29. Liu Y., Yang X., Wen W., Xia M. Smarter grid in the 5G Era: a framework integrating power internet of things with a cyber-physical system. *Frontiers in Communications and Networks*. 2021. T. 2. 689590.
30. Lesch V., Züfle M., Bauer A. та ін. A literature review of IoT and CPS—What they are, and what they are not. *Journal of Systems and Software*. 2023. T. 200. 111631.
31. Bibri S. The IoT for smart sustainable cities of the future: an analytical framework for sensor-based big data applications for environmental sustainability. *Sustainable Cities and Society*. 2018. T. 38. C. 230–253.
32. Hall R. E., Bowerman B. Smart cities: foundations, principles, and applications. London : Routledge, 2016. 432 c.
33. Silva E., Niu H. Crowdsourced Data Mining for Urban Activity. *Journal of Urban Planning*. 2020. T. 146, № 4. C. 04020038.
34. Hansen L., Jensen P. AI-Driven Accessibility in Copenhagen: A Case Study. *Smart Cities*. 2022. T. 5, № 2. C. 301–315.
35. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 800 c.
36. Kumar P., Rathore S. Machine Learning for Smart Cities. *International Journal of Advanced Computer Science and Applications*. 2023. T. 9, № 12. C. 999–1015.

37. Mukhamediev R. I., Popova Y., Kuchin Y. та ін. Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics*. 2022. Т. 10, № 15. 2552.
38. Ikotun A. M., Ezugwu A. E., Abualigah L. та ін. K-means Clustering Algorithms: a comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*. 2023. Т. 622. С. 178–210.
39. Cai J., Hao J., Yang H., Zhao X., Yang Y. A review on semi-supervised clustering. *Information Sciences*. 2023. Т. 632. С. 164–200.
40. Fernández-Delgado M., Sirsat M. S., Cernadas E. та ін. An extensive experimental survey of regression methods. *Neural Networks*. 2019. Т. 111. С. 11–34.
41. Niu S., Liu Y., Wang J., Song H. A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*. 2020. Т. 1, № 2. С. 151–166.
42. Brown T., Wilson L. Smart Infrastructure and AI. AI in Material Science. London : Taylor & Francis, 2024. С. 123–145.
43. Li H., Zhang Y. Data Quality in Crowdsourcing and Spamming Behavior Detection. *Computer Science*. 2024.
44. Мапа доступності. Вінницька міська рада. 2024. URL: <https://map.vmr.gov.ua/> (дата звернення: 03.02.2025).
45. Доступність. Вінницька міська рада. 2024. URL: <https://www.vmr.gov.ua/dostupnist> (дата звернення: 03.02.2025).
46. Korada L., Sikha V., Siramgari D. AI-Driven Accessibility. *International Journal of Innovative Science and Engineering*. 2024. С. 983–992.
47. Davis L., Brown M. AI for Accessibility: Opportunities and Challenges. *AAAI Magazine*. 2024. 14 с.
48. Smith J., Doe R. Artificial Intelligence in Enhancing the Efficiency. *World Journal of Engineering and Technology*. 2025. Т. 13, № 1. С. 15–28.
49. Lakshmana K., Kaluri R., Gundluru N. та ін. A review on deep learning techniques for IoT data. *Electronics*. 2022. Т. 11, № 10. 1604.

50. Mukhopadhyay S. C., Tyagi S. K. S., Suryadevara N. K. та ін. Artificial intelligence-based sensors for next generation IoT applications: a review. *IEEE Sensors Journal*. 2021. Т. 21, № 22. С. 24920–24932.

51. Seeing AI: A Tool for Visual Impairment. Microsoft Technical Research Team Report. 2020. 15 с. URL: <https://www.microsoft.com/en-us/research/project/seeing-ai/> (дата звернення: 03.02.2025).

52. Sousa J., Mendonça J. P., Machado J. A generic interface and a framework designed for industrial metrology integration for the Internet of Things. *Computers in Industry*. 2022. Т. 138. 103632.

53. Janiesch C., Zschech P., Heinrich K. Machine learning and deep learning. *Electronic Markets*. 2021. Т. 31, № 3. С. 685–695.

54. Möller S., Kühne R. Wheelmap: Crowdsourcing Accessibility Data. Proceedings of the 12th International Conference on Web for All. 2015. С. 1–4.

55. Garcia J., Lopez M. Navilens: QR-Based Navigation for Accessibility. *Assistive Technology*. 2021. Т. 33, № 5. С. 412–420.

56. Карта «Доступне місто». Львів. 2024. URL: [https://www.google.com/maps/d/u/0/viewer?mid=1SD\\_irzi2s21yLx0pAY8avXDYtjoBcpE](https://www.google.com/maps/d/u/0/viewer?mid=1SD_irzi2s21yLx0pAY8avXDYtjoBcpE) (дата звернення: 03.02.2025).

57. Безбар'єрність у медзакладах: МОЗ затвердило зміни до методичних рекомендацій. *Український медичний журнал*. 2024. № 3. С. 10–12.

58. Методичні рекомендації щодо впровадження принципів безбар'єрності під час надання медичних та реабілітаційних послуг. Київ : МОЗ України, 2024. 32 с.

59. Зміна 2 ДБН В.2.2-40:2018. Київ : Міністерство розвитку громад та територій України, 2024. 25 с.

60. Звіт про діяльність за 2023 рік. Київ : ГО «Доступно.UA», 2023. 32 с.

61. Kar B., Yahya W., Lin Y. D., Ali A. Offloading using traditional optimization and machine learning in federated cloud-edge-fog systems: a survey. *IEEE Communications Surveys & Tutorials*. 2023. Т. 25, № 2. С. 1199–1226.

62. Експериментальний проєкт містобудівного кадастру : Постанова Кабінету Міністрів України від 15.03.2024 р. № 315. *Офіційний вісник України*. 2024. № 25. С. 456–467.
63. Мапа доступності. Хмельницький. 2024. URL: <https://dostup.km.ua/мапа-доступности> (дата звернення: 03.02.2025).
64. Andersen M., Brynskov M., Borup Lynggaard A. Smart city initiatives: cases from Copenhagen and Malmö. *City, Culture and Society*. 2017. Т. 8. С. 49–61.
65. Duque-Ramos A., Varga L. Z. Smart city governance: a comparative analysis of European cities. *TeMA Journal of Land Use, Mobility, and Environment*. 2018. Т. 11, № 2. С. 179–194.
66. Etemadi K., Safavi A. The Role of Technology in Urban Inclusiveness. *The Scientific Journal*. 2023. Т. 19. С. 19–30.
67. Gomez-Rojas J., Camargo L., Montero R. Mobile wireless sensor networks in a smart city. *International Journal of Smart Sensing and Intelligent Systems*. 2018. Т. 11, № 1. С. 1–8.
68. Knickerbocker J. U. та ін. Heterogeneous integration technology demonstrations for future healthcare, IoT, and AI computing solutions. 2018 IEEE 68th Electronic Components and Technology Conference (ECTC). 2018. С. 1519–1528.
69. Russel L., Goubran R., Kwamena F. Towards reliable IoT: fog-based AI sensor validation. 2019 IEEE Cloud Summit. 2019. С. 37–44.
70. Islam T., Mukhopadhyay S. C., Suryadevara N. K. Smart sensors and Internet of Things: a postgraduate paper. *IEEE Sensors Journal*. 2017. Т. 17, № 3. С. 577–584.
71. Morello R., Mukhopadhyay S. C., Liu Z. та ін. Advances on sensing technologies for smart cities and power grids: a review. *IEEE Sensors Journal*. 2017. Т. 17, № 23. С. 7596–7610.
72. Sutjarittham T., Gharakheili H. H., Kanhere S. S., Sivaraman V. Experiences with IoT and AI in a smart campus for optimizing classroom usage. *IEEE Internet of Things Journal*. 2019. Т. 6, № 5. С. 7595–7607.
73. Nassi I., Brown L. Scalable Computing Systems for Future Smart Cities. *IET Smart Cities*. 2022. Т. 4, № 2. С. 114–127.

74. Hansen T. R., Paay J., Kjærgaard M. B. Participatory sensing for smart cities: lessons learned from environment monitoring. *Sensors*. 2017. T. 17, № 10. 2345.
75. Kim J., Lee S. Deep Learning Based Image Quality Assessment: A Survey. *Procedia Computer Science*. 2023. T. 225. C. 2278–2287.
76. Johnson M., Lee K. The Future of Urban Accessibility: The Role of AI. 2024.
77. Patel S., Singh R. Inclusive by Design: Accessibility and AI. *ACM Digital Library*. 2024. T. 135, № 1. C. 1–6.
78. Barroso S., Bustos P., Núñez P. та ін. Toward industrial IoT: integrated architecture of an OPC UA synergy platform. *IEEE Access*. 2021. T. 9. C. 164720–164731.
79. Kamienski C., Jentsch M., Eisenhauer M. та ін. Application development for the Internet of Things: a context-aware mixed criticality systems development platform. *Computer Communications*. 2016. T. 104. C. 1–16.
80. Razzaque M. A., Milojevic-Jevric M., Palade A., Clarke S. Middleware for internet of things: a survey. *IEEE Internet of Things Journal*. 2016. T. 3, № 1. C. 70–95.
81. Savaglio C., Fortino G., Zhou M. Towards interoperable, cognitive and autonomic IoT systems: an agent-based approach. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). 2016. C. 58–63.
82. Da Rocha H., Abrishambaf R., Pereira J., Espirito Santo A. Integrating the IEEE 1451 and IEC 61499 standards with the industrial internet reference architecture. *Sensors*. 2022. T. 22, № 4. 1495.
83. Fatima I., Malik S. U., Anjum A., Ahmad N. Cyber physical systems and IoT: architectural practices, interoperability, and transformation. *IT Professional*. 2020. T. 22, № 3. C. 46–54.
84. Becejac T., Eppinger C., Ashok A. та ін. Prime: a real-time cyber-physical systems testbed: from wide-area monitoring, protection, and control prototyping to operator training and beyond. *IET Cyber-Physical Systems: Theory & Applications*. 2020. T. 5, № 2. C. 186–195.
85. Leitao J. Concepts, Methodologies and Tools of Gamification and Design Thinking. Open Innovation Business Modeling. Cham : Springer, 2019. C. 85–127.

86. Ramu V. Performance Testing and Optimization Strategies for Mobile Applications. *International Journal of P2P Network and Technology*. 2023. T. 13, № 3. C. 45–60.
87. Awadid A. Reference Architectures for Cyber-Physical Systems: Towards a Standard-Based Comparative Framework. *Advances in Information and Communication: Proceedings of the 2022 Future of Information and Communication Conference*. Cham : Springer, 2022. C. 1–15.
88. Raghavan S., Simon B. Y. L. Data Integration for Smart Cities: Opportunities and Challenges. *Computational Science and Technology*. 2019. C. 393–403.
89. Moustaka V., Vakali A., Anthopoulos L. G. A Systematic Review for Smart City Data Analytics. *ACM Computing Surveys*. 2018. T. 51, № 5. C. 1–41.
90. Wang L. The Construction of Off-line Map Based on OpenStreetMap and Leaflet. *ICCMCEE*. 2015. C. 1478–1482.

## ДОДАТОК А

### (обов'язковий)

### ЛІСТИНГ КОДУ

```

import React from 'react'; // Компонент мапи з фронтенд застосунку
import { StyleSheet } from 'react-native';
import MapView from 'react-native-maps';

const MapComponent = ({ objects }) => (
  <MapView
    style={styles.map}
    initialRegion={{
      latitude: 49.422,
      longitude: 26.987,
      latitudeDelta: 0.05,
      longitudeDelta: 0.05,
    }}
    showsUserLocation={true}
  >
    {objects.map((obj) => (
      <MapView.Marker
        key={obj.id}
        coordinate={{ latitude: obj.lat, longitude: obj.lon }}
        pinColor={
          obj.status === 'accessible'
            ? 'green'
            : obj.status === 'partially_accessible'
            ? 'yellow'
            : 'red'
        }
        title={obj.type}
      />
    ))}
  </MapView>
);

const styles = StyleSheet.create({
  map: { flex: 1, borderRadius: 10 },
});

export default MapComponent;

from sklearn.ensemble import RandomForestClassifier // Приклад ендпоінту бекенд частини
from flask import Flask, request, jsonify

import pandas as pd
import psycopg2

app = Flask(__name__)
model = RandomForestClassifier(n_estimators=100).fit(X_train, y_train)

@app.route('/api/reports', methods=['POST'])

```

```

def process_report():
    try:
        data = request.get_json()
        if not all(key in data for key in ['object', 'status', 'location']):
            return jsonify({'status': 'rejected', 'reason': 'invalid_data'}), 400
        features = {
            'lat': data['location']['lat'],
            'lon': data['location']['lon'],
            'comment_length': len(data.get('comment', '')),
            'has_photo': 1 if data.get('photo') else 0
        }
        prediction = model.predict(pd.DataFrame([features]))
        status = 'verified' if prediction[0] == 'valid' else 'rejected'
        conn = psycopg2.connect(dbname='accessibility', user='user', password='pass',
                                host='localhost')
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO reports (object, status, lat, lon, photo, comment) VALUES (%s, %s, %s, %s, %s, %s)",
            (data['object'], status, data['location']['lat'], data['location']['lon'],
            data.get('photo'), data.get('comment'))
        )
        conn.commit()
        cursor.close()
        conn.close()
        return jsonify({'status': status}), 200
    except Exception as e:
        return jsonify({'status': 'error', 'reason': str(e)}), 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)

// Приклад БД
SELECT id, type, status
FROM objects
WHERE ST_DWithin(
    ST_SetSRID(ST_MakePoint(lon, lat), 4326),
    ST_SetSRID(ST_MakePoint(26.987, 49.422), 4326),
    100
);

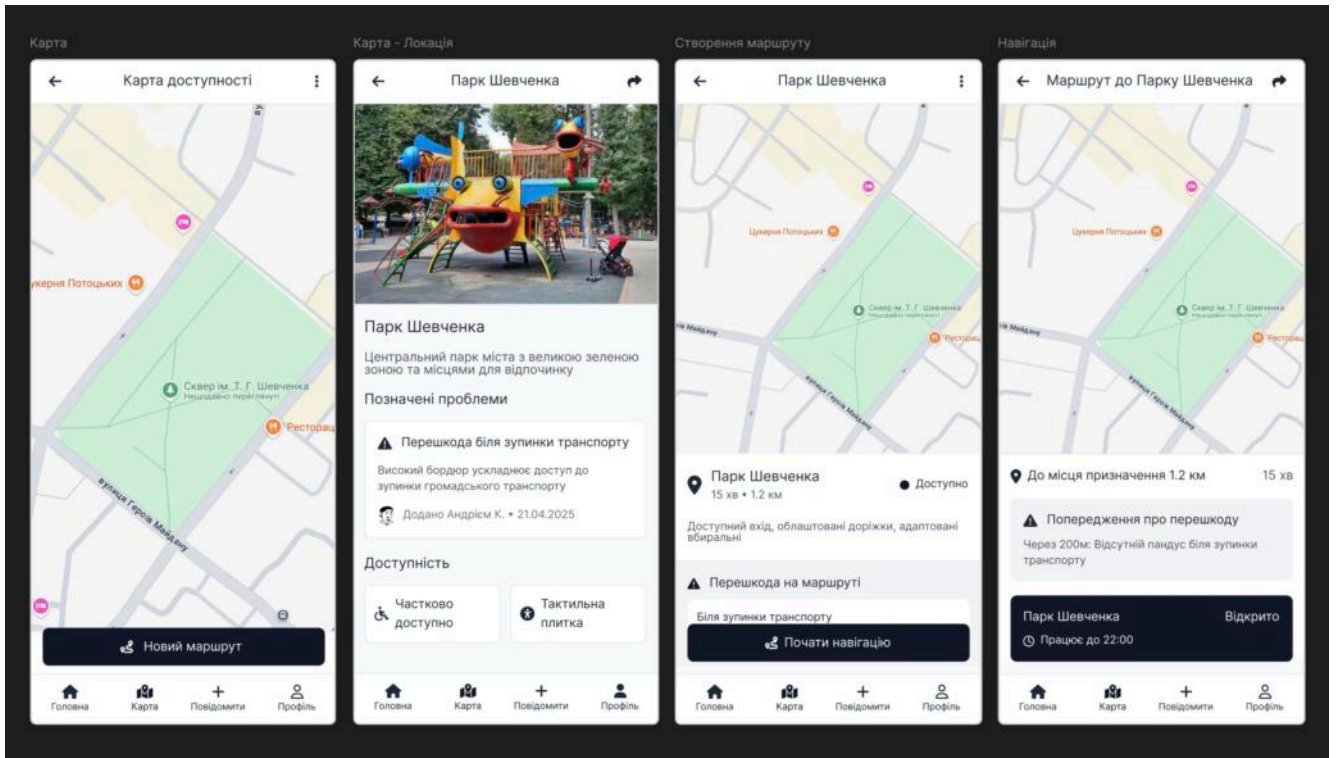
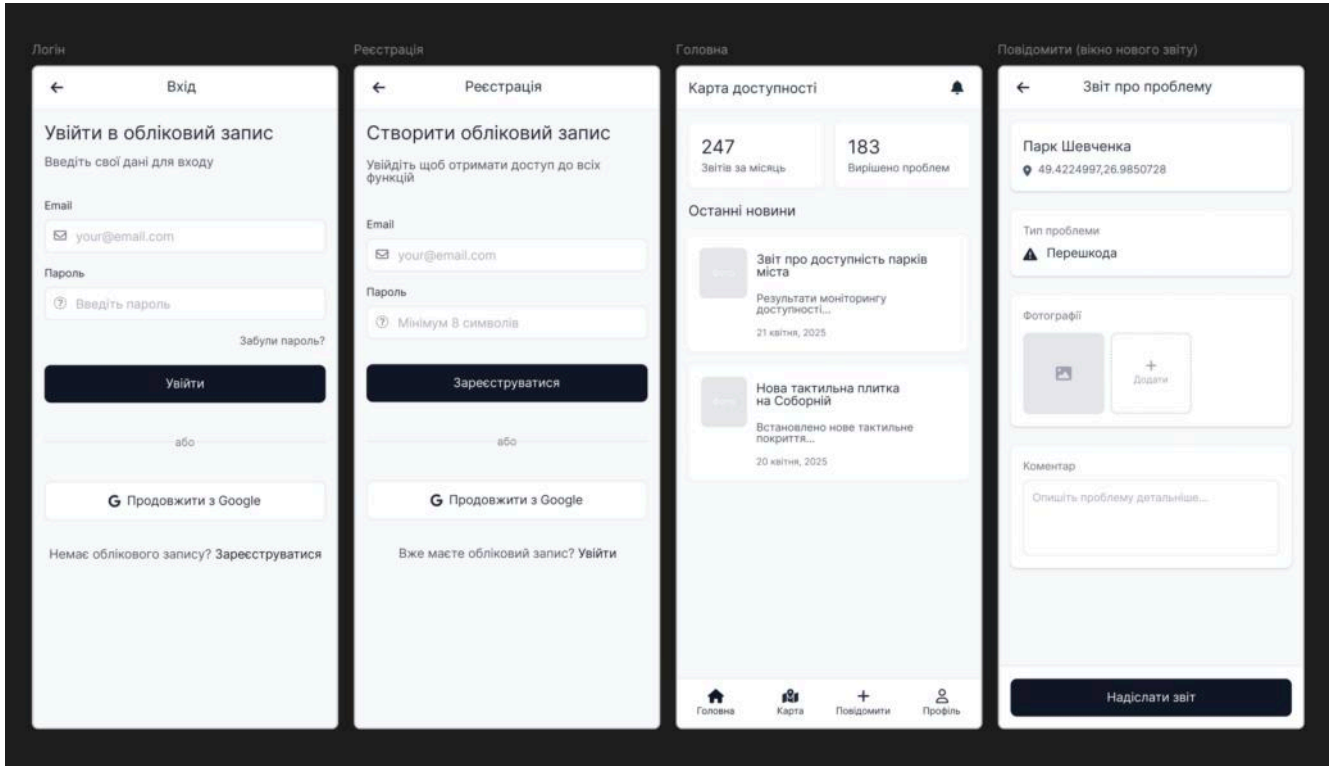
import pytest // Приклад тесту на фільтрацію дублікатів звітів
from filter_duplicates import filter_duplicates

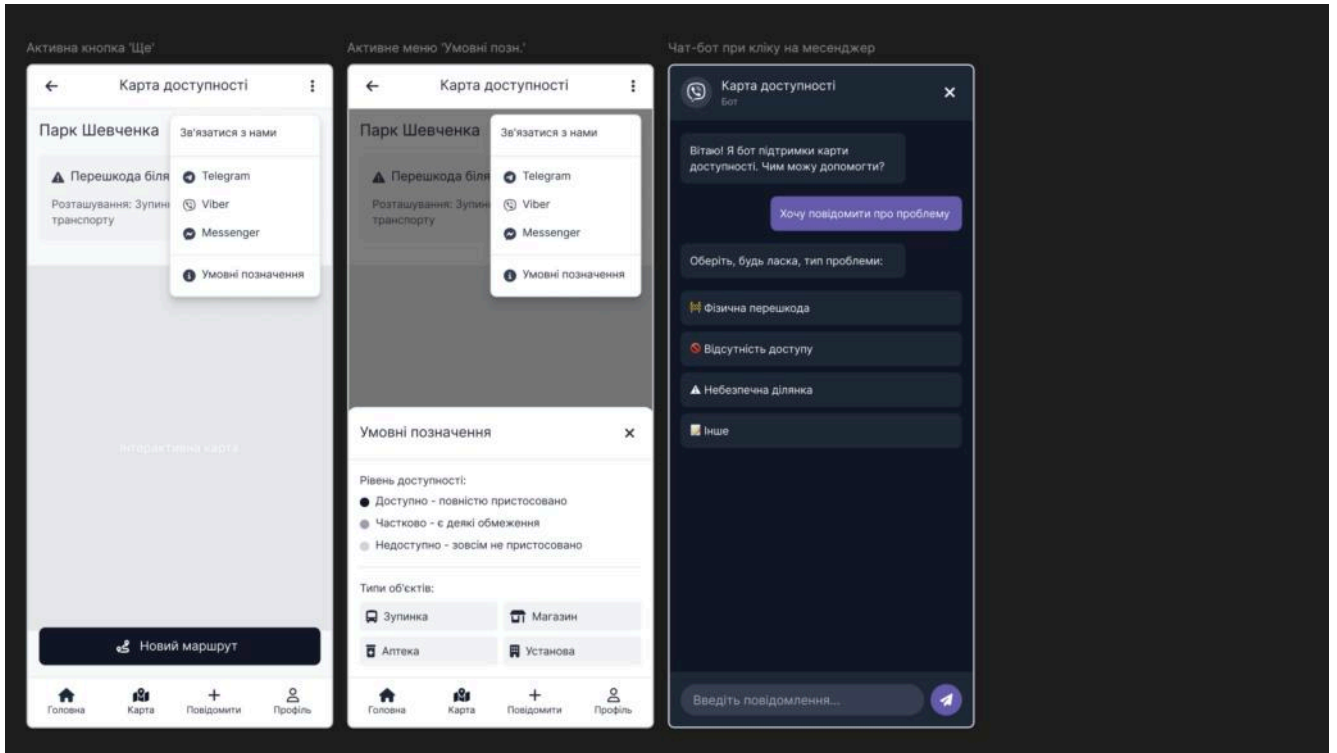
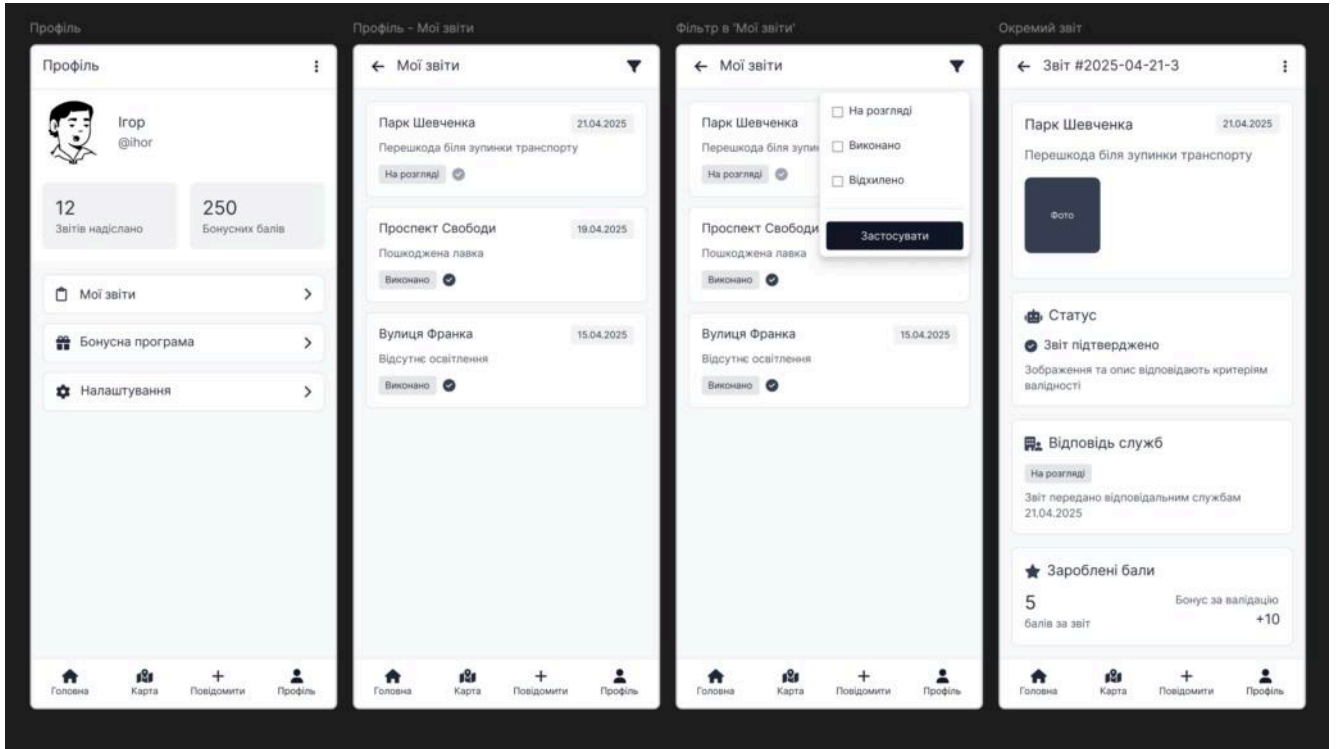
def test_filter_duplicates():
    reports = [
        {'location': {'lat': 49.422, 'lon': 26.987}, 'timestamp': '2025-03-14T10:00:00Z'},
        {'location': {'lat': 49.422, 'lon': 26.987}, 'timestamp': '2025-03-14T10:05:00Z'}
    ]
    result = filter_duplicates(reports)
    assert len(result) == 1, "Очікується один унікальний звіт"

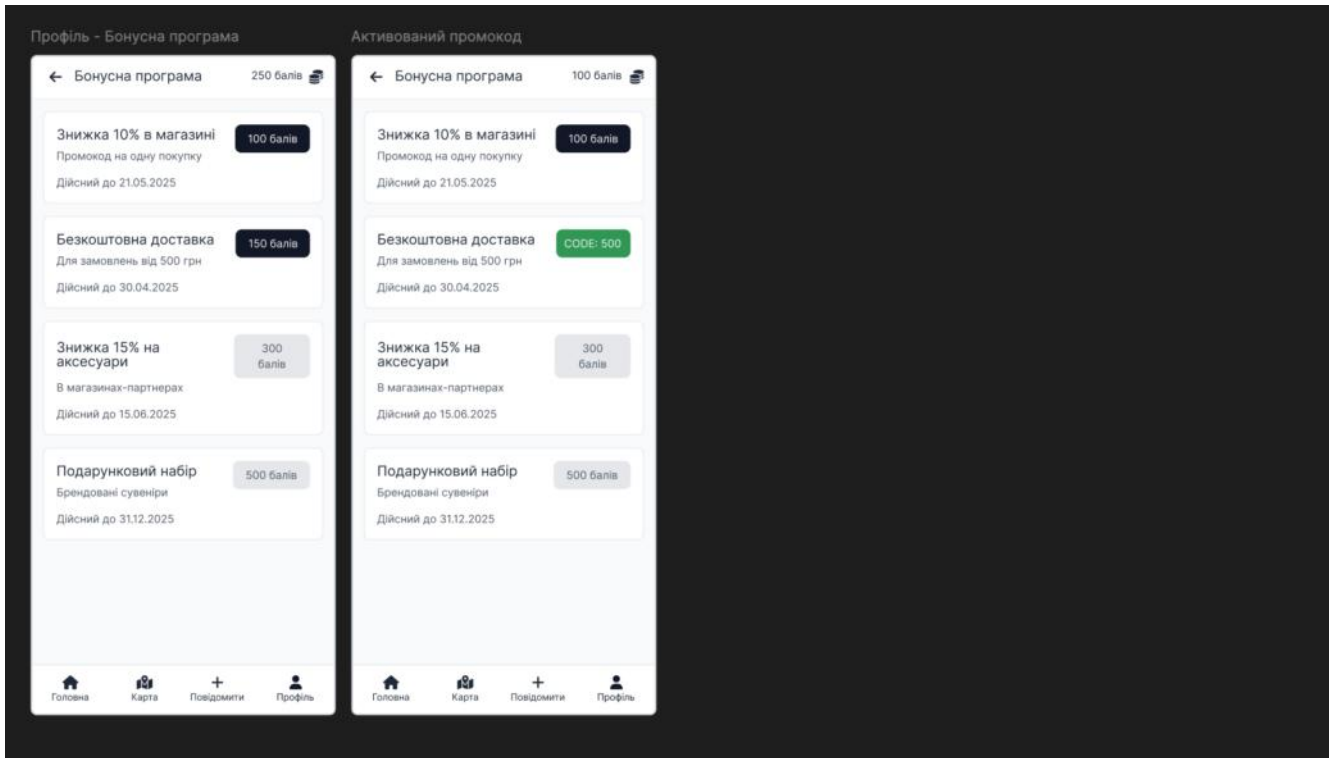
```

# ДОДАТОК Б

## СКРІНШОТИ ІНТЕРФЕЙСУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ







**ДОДАТОК В**  
(обов'язковий)  
**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Download

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Кафедра комп'ютерної інженерії та інформаційних систем

**КІБЕРФІЗИЧНА СИСТЕМА ДЛЯ  
ІНТЕРАКТИВНОГО ВІДОБРАЖЕННЯ  
ДОСТУПНОСТІ МІСЬКОЇ  
ІНФРАСТРУКТУРИ**

Автор роботи:  
Зайцев Ігор Олександрович

Науковий керівник:  
д-р. техн. наук, професор Федоров Євген Євгенович

Хмельницький - 2025

**ПРОБЛЕМАТИКА**

**Проблема:** обмежена доступність міської інфраструктури (тротуари, пандуси, зупинки) для людей з інвалідністю, осіб похилого віку та батьків з візками.

**Виклики:**

- відсутність централізованих даних про доступність у реальному часі.
- статична або застаріла інформація в наявних системах (Google Maps, Wheelmap).
- періодичний ручний збір даних є повільним, суб'єктивним і ресурсоємним.

**Потреба:** динамічна, масштабована система з інтеграцією звітів користувачів і ШІ.

## МЕТА ТА ОБ'ЄКТ ДОСЛІДЖЕННЯ

**Мета:** створення КФС для автоматизованого моніторингу та інтерактивної візуалізації доступності інфраструктури.

**Завдання:**

- Збір і верифікація краудсорсингових звітів про доступність.
- Розробка алгоритмів ШІ для обробки та верифікації даних.
- Інтеграція з муніципальними службами через REST API.
- Масштабована система для майбутнього впровадження IoT.

**Об'єкт:** міська інфраструктура (тротуари, пандуси, зупинки громадського транспорту, тощо).

## НАУКОВА НОВИЗНА

**Інновації:**

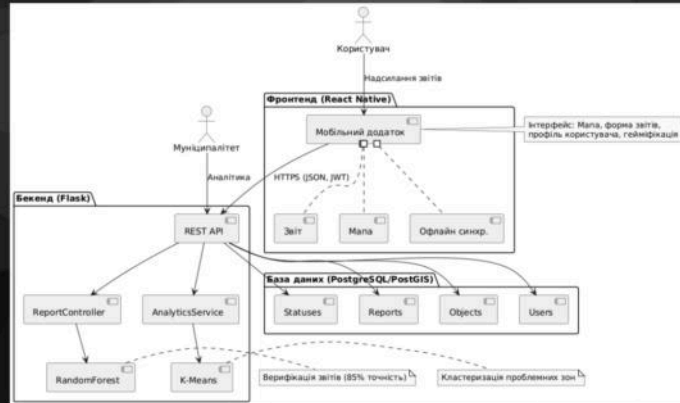
- Гібридна модель КФС: інтеграція краудсорсингових даних з алгоритмами ШІ (Random Forest, K-Means) для верифікації звітів і кластеризації проблемних зон.
- Модульна архітектура: клієнт-серверна система з підтримкою офлайн-режиму та готовністю до інтеграції IoT, що перевищує статичні GIS-рішення.
- Гейміфікація: мотивація користувачів через бали та статистику, що збільшує обсяг даних.

**Прогрес:** Динамічне оновлення карт доступності на відміну від статичних платформ.

# АРХІТЕКТУРА СИСТЕМИ

## Компоненти:

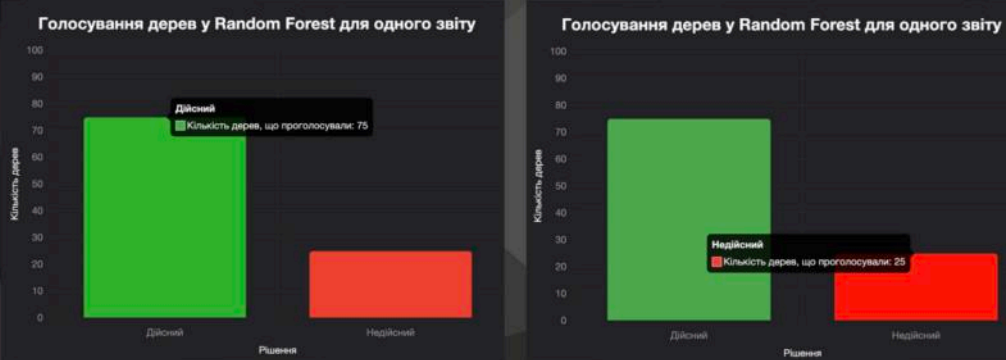
- Фронтенд: Мобільний додаток на React Native для iOS/Android.
- Бекенд: Сервер на Flask з REST API для обробки звітів і аналітики.
- База даних: PostgreSQL з PostGIS для швидких геозапитів.
- Модуль ШІ: Random Forest для верифікації, K-Means для кластеризації.



# МЕТОДИ ШІ В СИСТЕМІ

## Random Forest:

- Верифікація звітів (виявлення недійсних або дубльованих).
- Ознаки: геолокація, довжина коментаря, наявність фото.
- Точність: 85% на тестовій вибірці з 200 звітів.



**Результат:** Random Forest обирає "дійсний" як остаточне рішення, оскільки більшість дерев (75%) проголосували за цей клас.

# МЕТОДИ ШІ В СИСТЕМІ

## K-Means:

- Кластеризація звітів за координатами для виявлення "гарячих точок" проблем.
- Допомогає муніципалітетам визначати пріоритети.

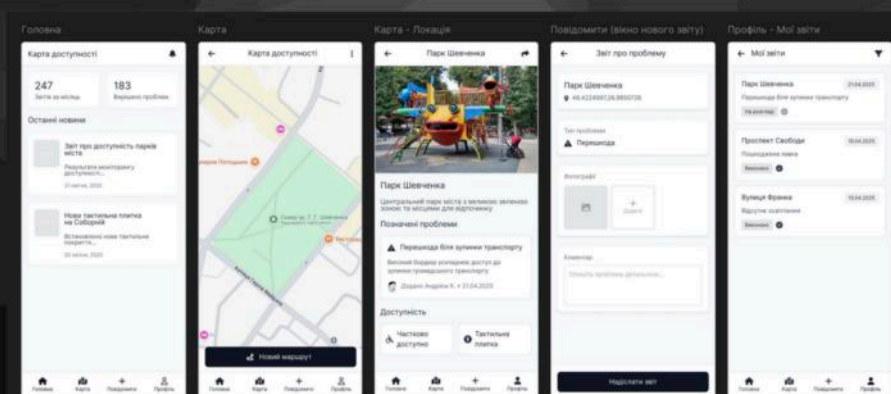


Діаграма показує, як K-Means групує звіти за координатами, виділяючи три кластери: центр міста, житловий район і околиці. Кожен кластер вказує на концентрацію проблем (наприклад, недоступні зупинки), допомагаючи муніципалітетам планувати ремонт.

# ВИСНОВКИ

Ключові результати роботи:

- Інструмент для планування безпечних маршрутів для маломобільних осіб.
- Інтеграція краудсорсингу та ШІ для динамічного оновлення даних.
- Інтуїтивний інтерфейс, офлайн-режим, гейміфікація.
- Модульна архітектура для масштабування та IoT-інтеграції.



## ДОДАТОК Г

(обов'язковий)

## СЕРТИФІКАТ УЧАСНИКА «АПКН-2024»

Сертифікат № 2024-021-1



Міністерство освіти і науки України  
Хмельницький національний університет

**СЕРТИФІКАТ****Зайцев Ігор Олександрович**

учасник XVI Всеукраїнської науково-практичної конференції  
«Актуальні проблеми комп'ютерних наук АПКН-2024»  
24 години участі (0,8 ECTS credits)

Голова оргкомітету АПКН-2024

**Олег СИНЮК**

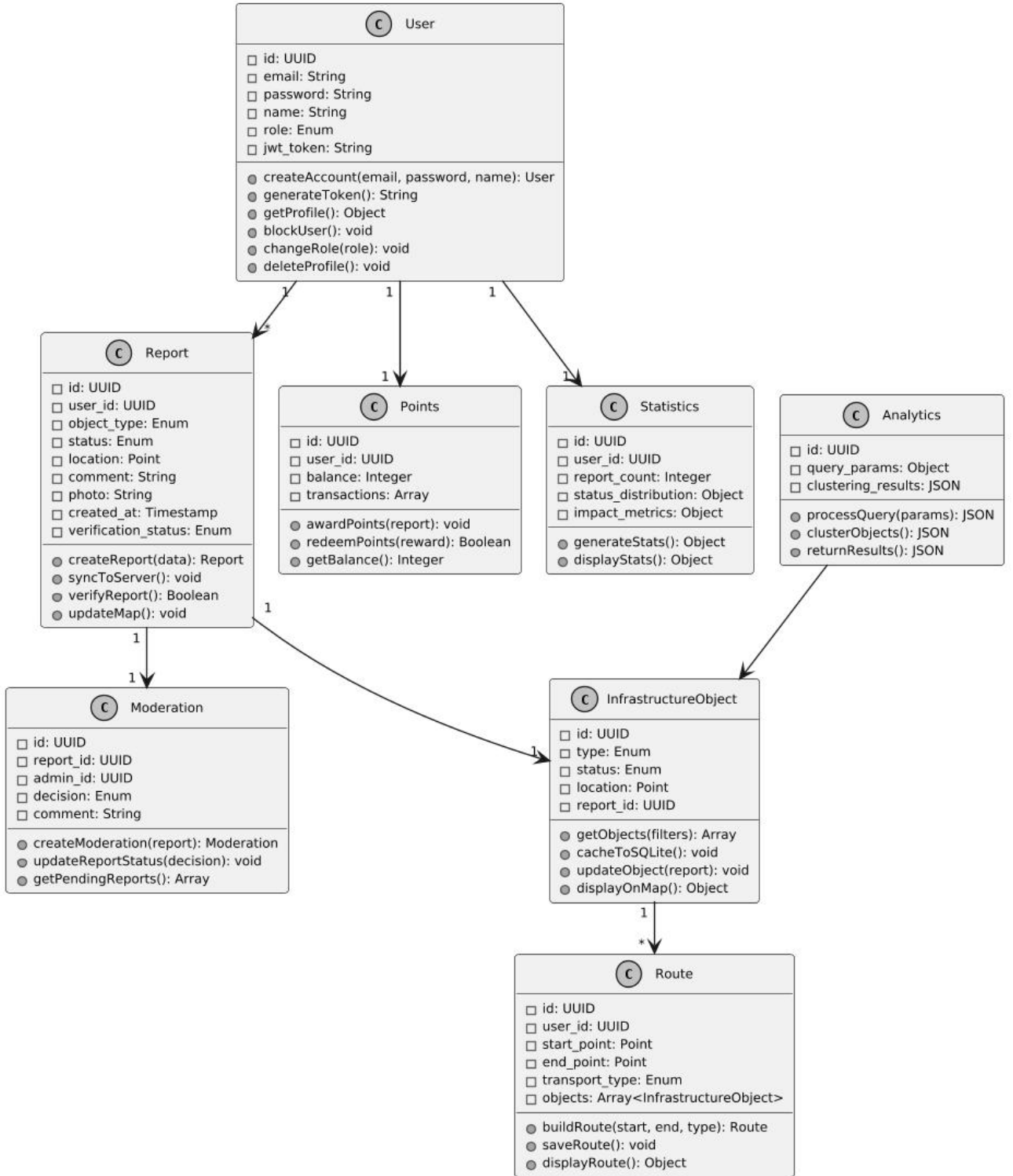
проректор Хмельницького національного  
університету з наукової роботи,  
доктор технічних наук, професор

м. Хмельницький  
15-16 листопада 2024

E-mail: [apkt.khnu@gmail.com](mailto:apkt.khnu@gmail.com)

# ДОДАТОК І

## ДІАГРАМА КЛАСІВ



## Anti-Plagiarism v-15.274 Educational

**The maximum coincidence with one document 0.0%**

**Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 13%**

ID: 241583 Title: МКР Кіберфізична система для інтерактивного відображення доступності міської інфраструктури Added in a DB: 2025-05-21 Authors: Ігор ЗАЙЦЕВ Heads: Євген ФЕДОРОВ Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	132574	1009	2010 (2%)	32 (3%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

### Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Ігор ЗАЙЦЕВ

**Співавтор:**

**Назва:** Зайцев\_Кіберфізична система для інтерактивного відображення доступності міської інфраструктури

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 4.6%

**Коефіцієнт подібності 2:** 1.6%

**Мікропробіли:** 19

**Заміна букв:** 3

**Інтервали:** 0

**Білі знаки:** 1

**Дата створення звіту:** 2025-05-21 12:36:13.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-05-21

Дата



Доцент Андрій Нічепорук

експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Зайцев Ігор Олександрович

Тема: Кіберфізична система для інтерактивного відображення доступності міської інфраструктури

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість сторінок записки 74

1. Короткий зміст роботи та прийнятих рішень У роботі розроблено кіберфізичну систему для моніторингу та інтерактивного відображення доступності міської інфраструктури з використанням краудсорсингових даних, алгоритмів штучного інтелекту та геоінформаційних технологій.

2. Висновок про відповідність роботи дипломному завданню \_\_\_\_\_  
Кваліфікаційна робота магістра відповідає виданому завданню \_\_\_\_\_

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз моделей та засобів, виявлено їхні недоліки, використано сучасні джерела. У другому розділі розроблено моделі моніторингу, що відповідають стандартам проектування. Третій розділ пропонує III-методи (Random Forest, K-Means) для верифікації та кластеризації, з перспективою CNN і NLP. У четвертому розділі запропоновано кіберфізичну систему та проведено багатогранне тестування.

4. Позитивні сторони роботи: Робота пропонує кіберфізичну систему, які інтегрує краудсорсинг та штучний інтелект для підвищення інклюзивності міського середовища. Модульна архітектура забезпечує масштабованість та можливість подальшої інтеграції IoT.

5. Негативні сторони роботи: У роботі недостатньо деталізовано вимоги до CNN і NLP, зокрема щодо обчислювальних ресурсів і датасетів, також не проведено порівняння продуктивності системи на різних апаратних платформах, що може впливати на її універсальність.

6. Оцінка графічного оформлення та пояснювальної записки роботи:  
Пояснювальна записка оформлена коректно, згідно з діючими стандартами.

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

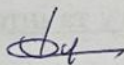
8. Інші зауваження: \_\_\_\_\_

9. Оцінка кваліфікаційної роботи магістра: Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «задовільно» 3.50 (D).

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Федула Микола Васильович, к.т.н., доцент, доцент кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

“ 23 ” травня 2025р.



М. В. Федула

Завідувачу кафедри КПС  
доктору філософії, доценту  
Ользі ПАВЛОВІЙ

Зайцева Ігоря Олександровича

ПБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-3

#### 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 (StrikePlagiarism 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 2025 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: «Кіберфізична система для інтерактивного відображення доступності міської інфраструктури»

Автор: Зайцев Ігор Олександрович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Федоров Євген Євгенович, д.т.н, професор

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:

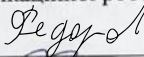
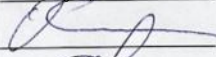
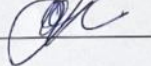
- 1) усі виявлені збіги пов'язані з використанням загальноживаних термінів і понять, таких як «розумне місто» чи «Інтернет речей», які є стандартними для наукових досліджень у цій галузі.
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано фрагменти програмного коду, або параметри які є стандартними технічними елементами і не підлягають авторському праву;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 4.57% і адресується до 46 першоджерел; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Євген ФЕДОРОВ

Олег САВЕНКО

Ольга ПАВЛОВА