

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів

Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

НОСАТОВ МИКОЛА ВОЛОДИМИРОВИЧ

(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.9:656

КВАЛІФІКАЦІЙНА РОБОТА

«Інформаційна система контролю руху громадського транспорту

м. Житомира»

(тема роботи)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело

_____ М. В. НОСАТОВ
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Гіваргізов Інвія Геннадійович
(прізвище, ім'я, по батькові)
к. е. н, старший викладач
кафедри
(науковий ступінь, вчене звання)

Висновок кафедри комп'ютерних технологій і моделювання систем
за результатами попереднього захисту: допущений

Протокол засідання кафедри комп'ютерних технологій і моделювання систем

№ 20 від «05» червня 2026р.

Завідувач кафедри комп'ютерних технологій і моделювання систем

к. пед. н.. доцент

(науковий ступінь, вчене звання)

(підпис)

Ковальчук М.О.

(прізвище, ім'я, по батькові)

«05» червня 2026р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти Носатов Микола Володимирович захистив
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

лаборант кафедри

(науковий ступінь, вчене звання)

(підпис)

Корольчук В.В.

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Носатов М. Інформаційна система контролю руху громадського транспорту м. Житомира. – Кваліфікаційна робота на правах рукопису. 37 с., 7 рис., 2 табл., 54 джерела, 8 додатків.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 122 – комп’ютерні науки. – Поліський національний університет, Житомир, 2026.

Об’єкт дослідження – процеси моніторингу та управління міським пасажирським транспортом у режимі реального часу. **Мета роботи** – підвищення ефективності функціонування транспортної мережі шляхом розробки інформаційної системи для збору, обробки та візуалізації телеметричних даних. **Методи дослідження** – методи об’єктно-орієнтованого проектування, системного аналізу, моделювання баз даних, сервіс-орієнтованої архітектури (SOA) та клієнт-серверної взаємодії через WebSocket.

У кваліфікаційній роботі розроблено інформаційну систему контролю руху громадського транспорту. Проведено аналіз предметної області та існуючих GPS-рішень. Обґрунтовано вибір технологічного стеку: Node.js, React та PostgreSQL з розширенням PostGIS. Спроектовано архітектуру системи, нормалізовану базу даних та алгоритми розрахунку часу прибуття (ETA). Реалізовано програмний комплекс, що забезпечує візуалізацію руху ТЗ на картах OpenStreetMap та автоматичну ідентифікацію відхилень від графіків. Тестування підтвердило прискорення геопросторових запитів у 30 разів та потенціал скорочення часу очікування пасажирів на 15-20%.

Ключові слова: GPS-моніторинг, інформаційна система, транспортна логістика, PostGIS, ETA.

SUMMARY

Nosatov M. Information system for public transport movement control in Zhytomyr. – Qualification paper in manuscript form. 37 pages, 7 figures, 2 tables, 54 references, 8 appendices.

Qualification paper for obtaining the bachelor's degree in specialty 122 – Computer Science. – Polissia National University, Zhytomyr, 2026.

Object of the study – processes of real-time monitoring and management of urban passenger transport. **Aim of the study** – to improve the efficiency of the transport network by developing an information system for collecting, processing, and visualizing telemetric data. **Method of research** – object-oriented design, system analysis, database modeling, service-oriented architecture (SOA), and client-server interaction via WebSocket.

Abstract Content: The qualification paper presents the development of an information system for public transport movement control. The study analyzes the domain and existing GPS solutions. The choice of technology stack – Node.js, React, and PostgreSQL with PostGIS extension – is justified. The system architecture, normalized database, and estimated time of arrival (ETA) algorithms are designed. A software package has been implemented to visualize vehicle movement on OpenStreetMap and automatically identify schedule deviations. Testing confirmed a 30-fold acceleration of geospatial queries and the potential to reduce passenger waiting time by 15-20%.

Keywords: *GPS monitoring, information system, transport logistics, PostGIS, ETA.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ	8
1.1 Аналіз процесу диспетчеризації та контролю громадського транспорту	8
1.2 Огляд та порівняльний аналіз систем-аналогів	8
1.3 Формування функціональних та нефункціональних вимог до системи	10
1.4 Обґрунтування вибору технологічного стеку та інструментальних засобів розробки.....	12
Висновки до розділу 1	13
2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	15
2.1 Розробка архітектури системи та моделі взаємодії компонентів	15
2.2 Об'єктно-орієнтоване моделювання системи з використанням UML-діаграм	16
2.3 Проєктування бази даних: інфологічний та даталогічний рівні	22
2.4 Алгоритмічне забезпечення обробки GPS-даних у реальному часі	23
Висновки до розділу 2	24
3 НАЗВА ДРУГОГО РОЗДІЛУ БЕЗ КРАПКИ.....	25
3.1 Реалізація серверної частини та API взаємодії	25
3.2 Розробка клієнтського інтерфейсу та інтеграція з картографічними сервісами	26
3.3 Тестування функціональності та надійності програмного забезпечення	27
3.4 Оцінка ефективності впровадження та рекомендації користувачам	28
Висновки до розділу 3	30
ВИСНОВКИ	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	33
ДОДАТКИ	37

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ГІС – географічна інформаційна система

API – Application Programming Interface (програмний інтерфейс додатка)

CASE – Computer-Aided Software Engineering (комп'ютерне проєктування програмного забезпечення)

ER-діаграма – Entity-Relationship Diagram (діаграма «сутність-зв'язок»)

GPRS – General Packet Radio Service (пакетна радіозв'язок загального користування)

GPS – Global Positioning System (глобальна система позиціонування)

GUI – Graphical User Interface (графічний інтерфейс користувача)

IDE – Integrated Development Environment (інтегроване середовище розробки)

REST – Representational State Transfer (архітектурний стиль взаємодії компонентів розподіленого додатка)

ВСТУП

Сучасний розвиток міської інфраструктури та концепції «Smart City» вимагає високого рівня цифровізації транспортних послуг. У місті Житомир, як і в інших обласних центрах України, оптимізація руху громадського транспорту є критично важливим завданням для забезпечення комфорту пасажирів та ефективного управління автопарком. Даний напрям є складовою загальноміської стратегії цифровізації, що закріплена у Концепції інтегрованого розвитку Житомира до 2030 року [1]. Відповідно до цього документа, створення інноваційної, безпечної та доступної транспортної інфраструктури є одним із пріоритетів розвитку громади, що передбачає впровадження інтелектуальних систем моніторингу та управління пасажиропотоками. Проблема нерегулярності рейсів, відсутності оперативного інформування пасажирів про час прибуття та складності диспетчерського контролю зумовлює необхідність впровадження спеціалізованих інформаційних систем.

Використання сучасних GPS-технологій та методів обробки великих даних дозволяє здійснювати моніторинг транспортних засобів у режимі реального часу. Проте існуючі рішення часто мають закриту архітектуру або недостатній функціонал для специфічних потреб муніципального управління. Розробка власної інформаційної системи контролю руху громадського транспорту м. Житомира дозволить забезпечити прозорість перевезень, підвищити точність дотримання графіків та створити зручний інструментарій для аналізу транспортних потоків. Це підтверджує актуальність обраної теми кваліфікаційної роботи.

Метою роботи є розробка та програмна реалізація інформаційної системи для автоматизації контролю та моніторингу руху громадського транспорту м. Житомира.

Для досягнення мети було поставлено такі **завдання**:

1. Проаналізувати предметну область та існуючі технологічні рішення у сфері GPS-моніторингу транспорту.
2. Обґрунтувати вибір методів, моделей та інструментальних засобів для

розробки системи.

3. Спроекувати архітектуру інформаційної системи, включаючи структуру бази даних та інтерфейси взаємодії.

4. Розробити програмне забезпечення системи для збору, обробки та візуалізації даних про рух транспорту.

5. Провести тестування розробленого ПЗ та оцінити ефективність його впровадження.

Об'єкт дослідження – процес управління та диспетчерського контролю руху міського громадського транспорту.

Предмет дослідження – методи, алгоритми та програмні засоби автоматизації моніторингу руху транспортних засобів у міському середовищі.

Для вирішення поставлених завдань у роботі використано наступні **методи дослідження**: *системний аналіз* – для дослідження структури транспортної мережі та функціональних вимог до системи; *об'єктно-орієнтоване моделювання (UML)* – для проєктування архітектури програмного забезпечення та логіки взаємодії компонентів; *теорія реляційних баз даних* – для розробки ефективної схеми зберігання даних про маршрути та рухомий склад; *методи вебпрограмування та API-інтеграції* – для реалізації клієнтської та серверної частин системи та отримання GPS-даних у реальному часі; *методи тестування ПЗ* – для перевірки надійності та коректності роботи розробленої системи.

Основні результати дослідження були висвітлені у:

Носатов М. В. [Аналіз проблематики автоматизації логістичного забезпечення у. Інформаційні технології та моделювання систем](#) : зб. праць учасн. Всеукр. наук.-практ. конф.здобувачів вищої освіти і молодих вчених, м. Житомир, 30 бер. 2023. Житомир :Поліський національний університет, 2023. С. 97-98.

Носатов М. В. [Моделювання бізнес-процесів системи контролю руху громадського транспорту. Інформаційні технології та моделювання систем](#) : зб. праць учасн. Всеукр. наук.-практ. конф. здобувачів вищої освіти і молодих вчених, м. Житомир, 22 трав. 2026. Житомир : Поліський національний

університет, 2026. С. 26-28

Практичне значення отриманих результатів. Розроблена інформаційна система може бути використана диспетчерськими службами м. Житомира для підвищення оперативності керування транспортом. Впровадження системи дозволяє автоматизувати формування звітів про порушення графіків, знизити вплив людського фактору на контроль перевезень та забезпечити мешканців міста актуальними даними про рух маршрутів через вебінтерфейси.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Повний обсяг роботи становить 30 сторінок. Робота містить 2 – таблиці, 7 рисунків. Список використаних джерел включає 54 найменування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз процесу диспетчеризації та контролю громадського транспорту

Процес диспетчеризації громадського транспорту є складною багаторівневою системою управління, основною метою якої є забезпечення виконання графіка руху, безпеки пасажирів та ефективного використання рухомого складу. Як зазначають дослідники [2, 3], сучасні системи моніторингу переходять від простої візуалізації об'єктів на карті до інтелектуального аналізу транспортних потоків на основі Інтернету речей (IoT). В умовах Житомира цей процес вимагає інтеграції великої кількості динамічних даних у реальному часі.

Диспетчерський контроль у Житомирі традиційно поділяється на кілька етапів:

- Планування: формування розкладів руху та закріплення ресурсів.
- Оперативний моніторинг: відстеження положення ТЗ за допомогою GPS/GPRS.
- Коригування: прийняття рішень у разі відхилень.

З точки зору комп'ютерних наук, цей процес описується як архітектура інтелектуальної системи моніторингу [4], де основним завданням є мінімізація часових затримок (latency) при обробці GPS-координат. Головною проблемою існуючого підходу є дискретність даних, що ускладнює точне прогнозування часу прибуття (ETA).

Важливим аспектом проектування таких систем є розробка адекватної моделі бази даних. Згідно з роботою [5], ефективна структура БД повинна забезпечувати не лише зберігання координат, а й оперативне аналітичне опрацювання інформації про якісні параметри виконання рейсів.

1.2 Огляд та порівняльний аналіз систем-аналогів

Для розробки ефективної інформаційної системи (ІС) необхідно провести критичний аналіз існуючих рішень, що представлені на ринку України та світу. Згідно з дослідженнями [6], сучасні системи моніторингу транспорту

класифікуються за рівнем відкритості даних та методами їх збору (GPS-трекінг, мобільний краудсорсинг, інтеграція з муніципальними даними).

Основними об'єктами аналізу в межах даної роботи обрано системи «Dozor City», «EasyWay» та сервіс «Google Maps Transit».

Система «Dozor City» [7] – найбільш розповсюджене в Житомирі рішення. Система базується на використанні спеціалізованого апаратного забезпечення (GPS-трекерів). Науковці зазначають, що подібні системи з закритою архітектурою мають високу точність, проте обмежені в контексті створення інтегрованого середовища «Smart City» через складність доступу до «сирих» даних сторонніми модулями [1].

Сервіс «EasyWay» [8] – агрегатор транспортних даних, що орієнтований на пасажирів. На відміну від «Dozor», він використовує комбіновані джерела даних. Проте, як зазначає Черняк О. В. у своєму дослідженні [5], такі сервіси часто стикаються з проблемою некоректної візуалізації через затримки при передачі пакетів даних від серверів перевізника до клієнтського додатка (latency).

Google Maps Transit [9]— глобальне рішення, що використовує стандарт GTFS [10]. Як вказують автори [11], цей стандарт є еталонним для обміну транспортною інформацією, проте він вимагає від міста наявності власної сформованої ІС, яка б генерувала ці дані, що й обґрунтовує доцільність нашої розробки.

Для обґрунтування переваг проектованої системи проведено порівняння за ключовими технічними критеріями (див. Табл. 1.1).

Таблиця 1.1. Порівняльна характеристика інформаційних систем моніторингу транспорту

Критерій	Dozor City	EasyWay	Google Maps	Проектована ІС
Підтримка стандарту GTFS	-	+	+	+
Модуль диспетчерського звіту	+	-	-	+
API для міських служб	-	-	+	+
Офлайн-режим роботи розкладів	-	+	-	+

Узагальнюючи результати проведеного порівняльного аналізу (див. табл. 1.1), можна зробити висновок, що жодне з існуючих рішень не забезпечує повного спектра функціональних можливостей, необхідних для комплексного

управління транспортною інфраструктурою міста Житомира.

Зокрема, комерційні системи (Dozor City) мають закриту архітектуру, що унеможливорює вільну інтеграцію даних у загальноміські сервіси «Smart City». Сервіси, орієнтовані на пасажирів (EasyWay, Google Maps), хоча й базуються на сучасних стандартах передачі даних GTFS, не надають інструментарію для оперативного диспетчерського контролю та формування аналітичної звітності для муніципальних органів.

Таким чином, розробка проекрованої інформаційної системи є обґрунтованою та доцільною. Основний акцент у роботі буде зроблено на поєднанні відкритості даних (через реалізацію власного API) та розширенні функціоналу для диспетчера, що дозволить не лише моніторити рух ТЗ у реальному часі, а й автоматизувати процес виявлення порушень графіків руху.

1.3 Формування функціональних та нефункціональних вимог до системи

На основі аналізу предметної області та недоліків існуючих систем-аналогів, визначено перелік вимог до проекрованої інформаційної системи контролю руху громадського транспорту м. Житомира. Згідно з методологією розробки ПЗ [12], вимоги поділяються на функціональні та нефункціональні.

Функціональні вимоги визначають сервіси, які система повинна надавати користувачам (пасажирам, диспетчерам та адміністраторам). Модуль «Пасажир» орієнтований на забезпечення доступу користувачів до актуальної інформації про роботу міського громадського транспорту. Його функціональність включає відображення поточного положення транспортних засобів на інтерактивній карті міста, можливість вибору конкретного маршруту для детального відстеження, прогнозування часу прибуття транспортного засобу на обрану зупинку (ETA) з урахуванням поточної дорожньої ситуації, а також пошук оптимального маршруту між двома заданими точками міського простору.

Функціональні можливості модуля «Водій» охоплюють процеси персональної ідентифікації на маршруті та забезпечення стабільного зворотного

зв'язку із серверною частиною системи. Зокрема, передбачено інструментарій для автоматичної передачі телеметричних даних (GPS-координат) у реальному часі, візуалізацію індивідуального графіка руху з автоматичним розрахунком відхилення від розкладу, а також систему отримання оперативних сповіщень від диспетчерської служби щодо актуальних змін у дорожній ситуації чи надзвичайних подій на шляху прямування.

Модуль «Диспетчер» призначений для оперативного управління та контролю транспортних перевезень у реальному часі. У його межах реалізується моніторинг дотримання графіків руху транспортних засобів із фіксацією випереджень або запізнь, автоматизоване формування сповіщень у разі критичних відхилень від заданого маршруту або розкладу, а також генерація аналітичних звітів за зміну, день або місяць щодо ефективності роботи окремих перевізників. Крім того, модуль забезпечує можливість оперативного внесення змін до маршрутної мережі, зокрема у випадках тимчасового перекриття вулиць або надзвичайних ситуацій.

Модуль «Адміністратор» виконує функції системного управління та налаштування. Він забезпечує керування обліковими записами користувачів і розмежування прав доступу, редагування бази даних маршрутів, зупинок і транспортних одиниць, а також конфігурування параметрів інтеграції системи з зовнішніми GPS-шлюзами та сервісами позиціонування.

Нефункціональні вимоги визначають якісні характеристики та обмеження функціонування системи. Зокрема, вимоги до продуктивності передбачають, що час відгуку системи на запит користувача не повинен перевищувати двох секунд, а сама система має забезпечувати одночасну роботу до п'яти тисяч активних пасажирських сесій. Надійність і доступність визначаються коефіцієнтом готовності не нижче 99,5% із цілодобовим режимом роботи, а також реалізацією механізмів автоматичного резервного копіювання бази даних. Архітектура системи повинна бути масштабованою, що дозволяє додавати нові маршрути та типи транспорту без внесення змін у програмний код. Важливою вимогою є сумісність, зокрема підтримка експорту даних у форматі GTFS для інтеграції з глобальними сервісами на кшталт Google Maps відповідно до міжнародних стандартів. Окрему

увагу приділено безпеці, яка передбачає захист персональних даних користувачів, використання протоколу HTTPS для передавання інформації та розмежування прав доступу на рівні бази даних.

1.4 Обґрунтування вибору технологічного стеку та інструментальних засобів розробки

Вибір технологічного стеку є одним із ключових етапів проєктування інформаційної системи, оскільки саме він визначає її продуктивність, масштабованість, надійність та можливості подальшого розвитку. З огляду на вимоги до системи, сформульовані у попередньому підрозділі, зокрема необхідність обробки GPS-даних у режимі реального часу та підтримки значної кількості одночасних підключень, доцільним є використання клієнт-серверної архітектури з опорою на сучасні вебтехнології.

Для реалізації серверної частини системи було проаналізовано можливості платформ Node.js та Python із фреймворками Django і FastAPI. У результаті обґрунтованого порівняння обрано платформу Node.js з фреймворком Express. Такий вибір зумовлений подієво-орієнтованою моделлю виконання, що лежить в основі Node.js і забезпечує ефективну обробку великої кількості паралельних запитів без блокування потоків. Це є критично важливим для систем моніторингу транспорту, де сервер повинен одночасно приймати дані від GPS-трекерів, обробляти їх та передавати актуальну інформацію користувачам у режимі реального часу.

Оскільки система працює з геопросторовими даними, такими як координати руху транспорту, маршрути та геозони, особлива увага приділялася вибору системи управління базами даних. Найбільш доцільним рішенням у цьому контексті є використання реляційної СУБД PostgreSQL із просторовим розширенням PostGIS. Дане поєднання вважається галузевим стандартом у сфері геоінформаційних систем і надає розширені можливості для виконання просторових запитів, зокрема перевірки знаходження об'єктів у межах заданих зон, обчислення відстаней та аналізу маршрутів, що суттєво спрощує логіку серверної частини застосунку.

Для реалізації клієнтської частини системи обрано бібліотеку React.js, яка забезпечує побудову інтерактивних односторінкових вебзастосунків. Використання React дозволяє реалізувати динамічне оновлення інтерфейсу, зокрема відображення поточного положення транспорту на карті без перезавантаження сторінки, що позитивно впливає на зручність користування системою. Для візуалізації картографічних даних передбачається застосування бібліотек Leaflet або Google Maps JavaScript API, які надають гнучкі інструменти роботи з картами, шарами та маркерами.

У процесі розробки програмного забезпечення використовуються сучасні інструментальні засоби, зокрема середовище розробки Visual Studio Code, яке підтримує велику кількість розширень для роботи з JavaScript, SQL та вебтехнологіями. Для керування версіями коду застосовується система Git із використанням віддалених репозиторіїв на платформі GitHub, що забезпечує контроль змін і можливість командної розробки. Проєктування архітектури системи здійснюється з використанням UML-діаграм у середовищах StarUML або Lucidchart, а для документування програмного інтерфейсу прикладного програмування використовується специфікація Swagger/OpenAPI [13].

Таким чином, обраний технологічний стек є збалансованим і відповідає вимогам до високонавантажених систем реального часу, забезпечуючи високу продуктивність, масштабованість та відповідність сучасним стандартам розробки програмного забезпечення.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз предметної області та обґрунтовано необхідність розробки інформаційної системи моніторингу громадського транспорту м. Житомира. Аналіз процесу диспетчеризації показав, що ключовою проблемою існуючих підходів є висока латентність даних та відсутність єдиного аналітичного середовища для взаємодії муніципальних служб і приватних перевізників. Визначено, що ефективна система має базуватися на принципах обробки подій у реальному часі.

Порівняльний аналіз аналогів (Dozor City, EasyWay, Google Maps) виявив, що існуючі рішення є або закритими для інтеграції, або не мають достатнього функціоналу для формування спеціалізованої диспетчерської звітності. Встановлено доцільність розробки власної системи, що використовуватиме відкритий стандарт GTFS, що дозволить забезпечити високу інтероперабельність даних. Сформовано перелік функціональних вимог, серед яких пріоритетними визначено: візуалізацію ТЗ на мапі, автоматичний розрахунок часу прибуття (ETA) та модуль генерації звітів про порушення графіків. Визначено нефункціональні вимоги, зокрема щодо продуктивності (час відгуку до 2 с) та надійності системи в режимі 24/7.

Обґрунтовано вибір технологічного стеку, який базується на використанні Node.js для серверної частини, бібліотеки React для інтерактивного фронтенду та СУБД PostgreSQL з розширенням PostGIS. Такий вибір забезпечує високу швидкість опрацювання геопросторових запитів та масштабованість системи.

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Розробка архітектури системи та моделі взаємодії компонентів

На основі функціональних вимог, визначених у першому розділі роботи, було спроектовано багаторівневу архітектуру інформаційної системи, орієнтовану на обробку поточкових даних у режимі реального часу. З урахуванням високих вимог до масштабованості, відмовостійкості та чіткого розмежування зон відповідальності, за архітектурну основу обрано сервіс-орієнтований підхід (Service-Oriented Architecture, SOA).

Запропонована архітектура охоплює повний цикл обробки даних – від моменту їх надходження до візуалізації кінцевому користувачеві. Джерелами інформації виступають GPS/GPRS-трекери, встановлені на транспортних засобах, а також мобільні застосунки водіїв, з яких телеметричні дані передаються на серверний шлюз за протоколами TCP/UDP. Отримані координати та службова інформація надходять до серверного рівня, реалізованого на базі платформи Node.js, який виконує функції центрального вузла системи. На цьому етапі здійснюється первинна валідація вхідних даних, фільтрація GPS-шумів, а також обчислення логічних параметрів, зокрема перевищення швидкості, відхилення від заданого маршруту та прогнозований час прибуття транспортного засобу (ETA).

Збереження просторових і атрибутивних даних забезпечується за допомогою системи керування базами даних PostgreSQL із розширенням PostGIS, що дозволяє ефективно працювати з геопросторовими об'єктами, такими як координати зупинок, траєкторії маршрутів, а також інформація про водіїв і графіки руху. Рівень представлення реалізовано у вигляді веб-інтерфейсів для пасажирів та диспетчерів, побудованих із використанням бібліотеки React. Взаємодія клієнтських застосунків із серверною частиною здійснюється через REST API, а для забезпечення миттєвого оновлення позицій транспортних засобів на карті застосовуються WebSocket-з'єднання.

Модель взаємодії компонентів системи базується на принципах реактивного оновлення даних. Після надходження нового пакета координат від транспортного

засобу сервер не лише оновлює відповідні записи у базі даних, але й ініціює push-повідомлення через сокет-з'єднання для всіх активних клієнтів, які відстежують відповідний маршрут. Такий підхід дозволяє досягти затримки відображення інформації менше однієї секунди, що відповідає нефункціональним вимогам системи, визначеним у підрозділі 1.3.

Для формалізації часової взаємодії між ключовими компонентами було застосовано підхід об'єктно-орієнтованого аналізу [14], який дозволив чітко описати зв'язки та сценарії взаємодії між сутностями «Транспорт», «Диспетчер» та «Сервер обробки», що забезпечує логічну цілісність та розширюваність системи.

Детальна технічна специфікація апаратного забезпечення транспортних засобів, структурна схема мережевої інфраструктури та конфігурація серверного сегмента системи наведені у Додатку Г

2.2 Об'єктно-орієнтоване моделювання системи з використанням UML-діаграм

Для візуалізації структури та поведінки проєктованої інформаційної системи застосовано уніфіковану мову моделювання UML (Unified Modeling Language), яка дає змогу формалізувати вимоги до системи та забезпечити однозначне трактування логіки її функціонування на етапі проєктування і подальшої реалізації програмного коду. UML-моделювання дозволяє узгодити бачення системи між розробниками, аналітиками та замовниками, зменшуючи ризик концептуальних і реалізаційних помилок.

У процесі моделювання розглянуто функціональні, структурні та динамічні аспекти системи. Функціональні можливості описано за допомогою діаграми варіантів використання (рис. 2.1), яка відображає основні ролі користувачів і сценарії їх взаємодії із системою. У межах даної інформаційної системи ключовими акторами визначено пасажир, водія, диспетчера та адміністратора, кожен з яких має власний набір прав і функцій, що забезпечують повноцінну експлуатацію системи.

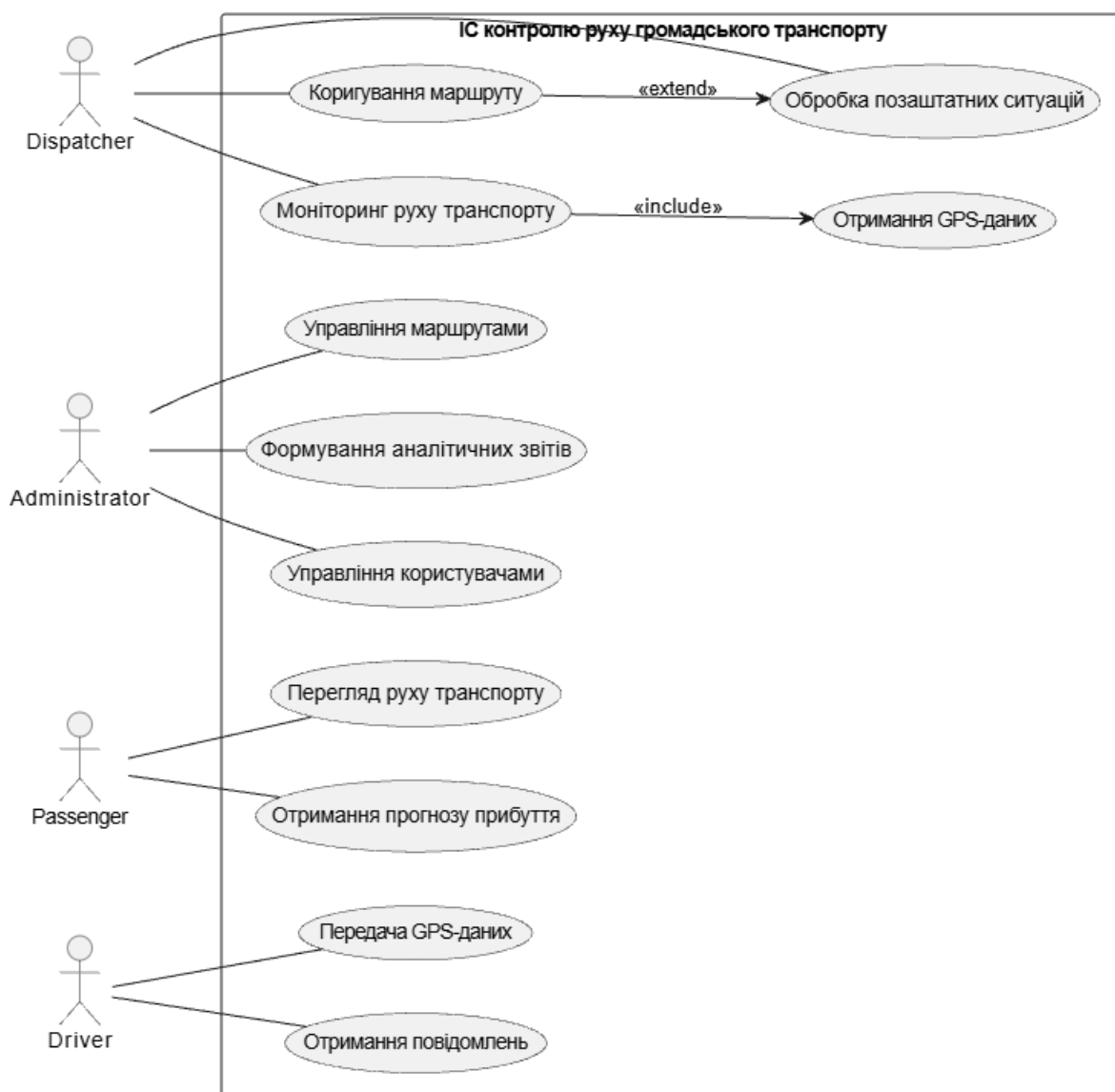


Рисунок 2.1 – Діаграма варіантів використання системи

Представлена діаграма варіантів використання (рис. 2.1) формалізує функціональну структуру системи через взаємодію чотирьох ключових акторів, де центральне місце займає процес обробки GPS-даних, що є спільним для водія та диспетчера. Для адміністратора та диспетчера визначено розширені сценарії управління маршрутами та формування звітності, тоді як пасажир виступає споживачем інформаційних сервісів через прецеденти перегляду руху та отримання прогнозів прибуття. Використання зв'язків типу «include» та «extend» на діаграмі дозволяє чітко розмежувати обов'язкові операції збору телеметрії та опціональні функції реагування на позаштатні ситуації, що забезпечує гнучкість алгоритмів управління транспортною мережею міста.

Статичну структуру системи формалізовано за допомогою діаграми класів

(рис. 2.2.), яка є центральним елементом об'єктно-орієнтованого аналізу. Вона описує склад і взаємозв'язки основних сутностей системи, зокрема класи Vehicle, Route, Stop, Schedule та User, їх атрибути, методи та типи зв'язків, такі як асоціація, агрегація та успадкування. Такий підхід дозволяє чітко визначити модель даних і логіку взаємодії між компонентами системи.

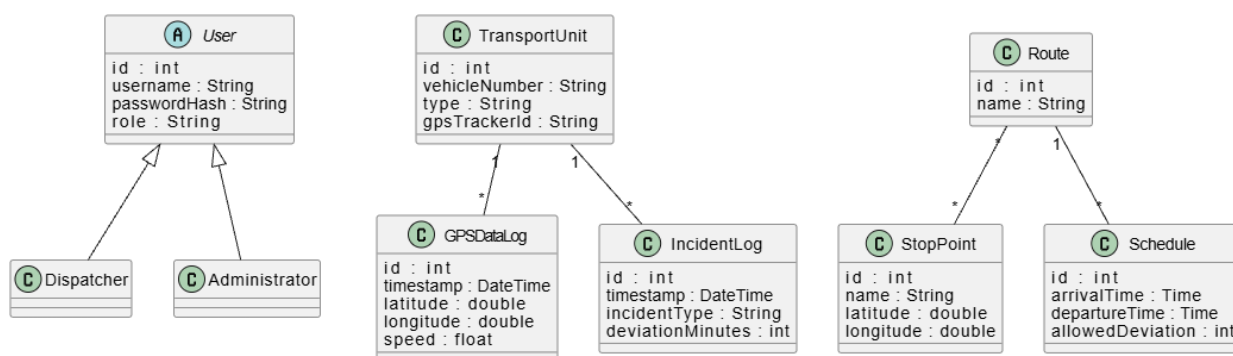


Рисунок 2.2 – Діаграма класів інформаційної системи

Опис статичної структури на рисунку 2.2 демонструє логічні зв'язки між об'єктами. Зокрема, зв'язок типу «композиція» між класами Route (Маршрут) та Stop (Зупинка) вказує на те, що зупинки існують лише в контексті конкретних маршрутів. Атрибути класу Vehicle, такі як last_coord та speed, є динамічними та оновлюються в реальному часі, що дозволяє іншим класам (наприклад, Dispatcher) проводити аналітику без затримок у зверненні до бази даних.

Для відображення динамічної поведінки системи розроблено діаграму послідовності (рис. 2.3), яка описує процес обробки GPS-сигналу та передачі актуальної інформації кінцевому користувачеві. Діаграма демонструє послідовність взаємодії об'єктів у часі – від моменту генерації координат GPS-трекером до візуалізації транспортного засобу на карті в браузері пасажирів. Це дозволяє проаналізувати часові залежності та інформаційні потоки всередині системи.

Діаграма послідовності відображає часову взаємодію між компонентами системи під час оброблення телематичних даних. На діаграмі показано послідовність передавання даних від GPS-трекера до серверних модулів, збереження інформації у базі даних та обчислення прогнозу прибуття (ETA).

Дана модель дозволяє проаналізувати динамічну поведінку системи та визначити критичні точки оброблення даних у режимі реального часу.

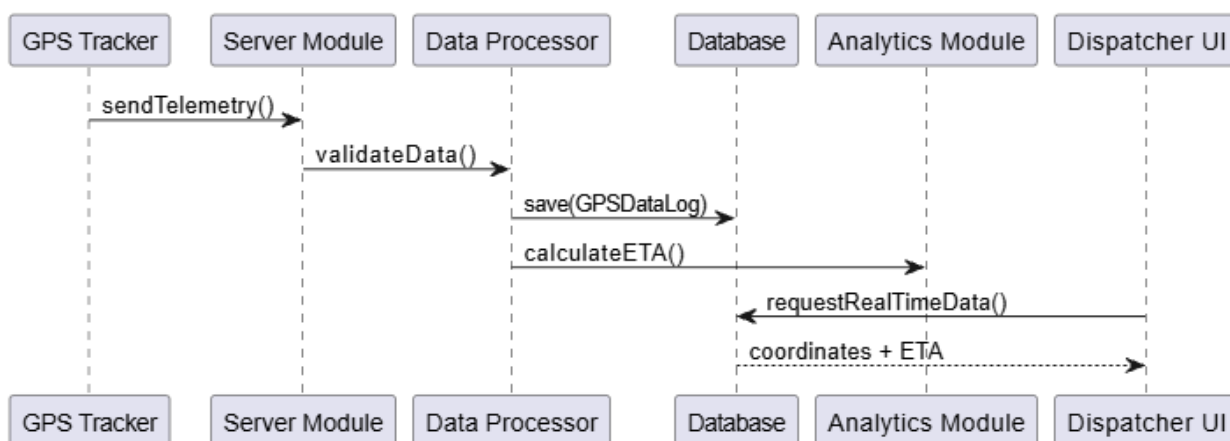


Рисунок 2.3. – Діаграма послідовності

Такий підхід дозволяє чітко зафіксувати логіку опрацювання телеметрії, забезпечити послідовність оброблення даних, уникнути гонок даних та визначити вимоги до продуктивності системи. Діаграма послідовності є невід’ємною частиною моделювання динамічної поведінки ІС КРГТ у режимі реального часу.

Діаграма діяльності (Activity Diagram) використовується для моделювання потоків робіт та алгоритмічної логіки виконання складних процесів, що включають умови, цикли та паралельні гілки [15]. У межах ІС КРГТ особливо важливим є процес автоматичного виявлення порушень графіка руху, оскільки він визначає своєчасність реагування на відхилення від установлених розкладів.

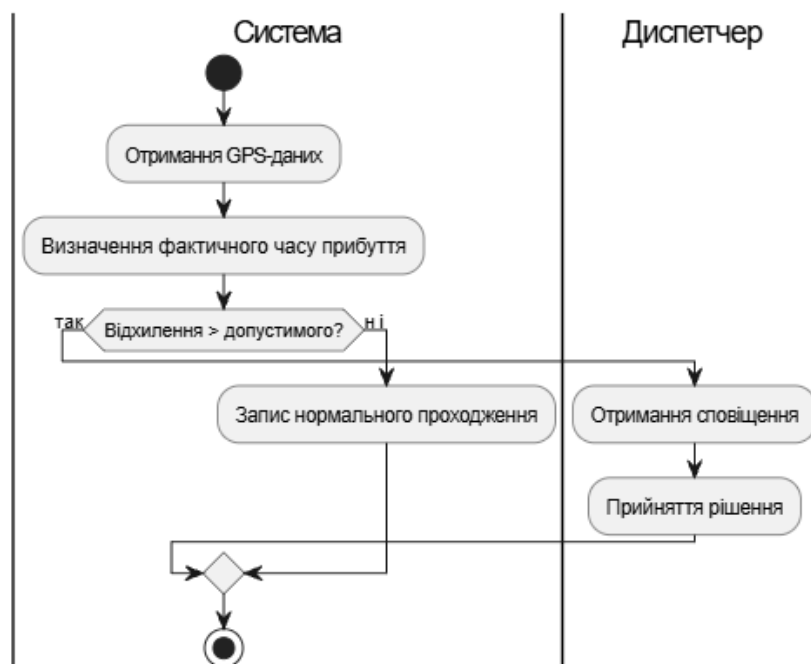


Рисунок 2.4. – Діаграма діяльності процесу контролю дотримання графіка руху

Діаграма діяльності описує алгоритм автоматичного контролю дотримання

графіка руху транспортного засобу. Процес включає отримання GPS-даних, визначення фактичного часу прибуття, порівняння з плановим розкладом та перевірку допустимого відхилення.

У разі виявлення порушення система ініціює формування повідомлення диспетчеру та реєстрацію інциденту, що забезпечує оперативне реагування на позаштатні ситуації.

У разі перевищення допустимих меж запускається діяльність «Надсилання сповіщення диспетчеру». У разі відсутності порушень система переходить у стан очікування наступного телеметричного повідомлення. Дані про факт відхилення записуються в лог інцидентів, забезпечуючи подальшу аналітичну обробку [8].

Наприклад, критично важливою є послідовність оброблення реальних даних для відображення місцезнаходження транспорту (див. табл. 2.1):

Таблиця 2.1. – Сценарій: "Відображення місцезнаходження транспортної одиниці"

Час (зверху вниз)	Об'єкт- Відправник	Повідомлення	Об'єкт- Отримувач	Опис
t₀	GPS-Tracker	sendTelemetry(coords, speed)	ServerModule	Передача даних кожні N секунд.
t₁	ServerModule	validate(data)	DataProcessor	Перевірка на валідність та очищення.
t₂	DataProcessor	save(validatedData)	Database (GPSDataLog)	Збереження логу.
t₃	DataProcessor	calculateETA(coords, routeID)	AnalyticsModule	Визначення очікуваного часу прибуття на наступну зупинку.
t₄	Database	getRealTimePosition()	DispatcherInterface	Запит інтерфейсу на актуальні координати.
t₅	DispatcherInterface	display(coords, ETA)	Диспетчер (Актор)	Відображення транспорту на карті.

Це дозволяє візуалізувати часові залежності та логіку взаємодії між ключовими компонентами системи в режимі реального часу.

Діаграма станів описує життєвий цикл інциденту, що виникає в процесі контролю руху громадського транспорту. Інцидент проходить послідовні стани:

виявлення, аналіз, підтвердження або відхилення як хибної тривоги, вирішення та архівація.

Дана діаграма забезпечує формалізацію логіки оброблення позаштатних ситуацій та підтримує аналітичну оцінку ефективності роботи транспортної системи.

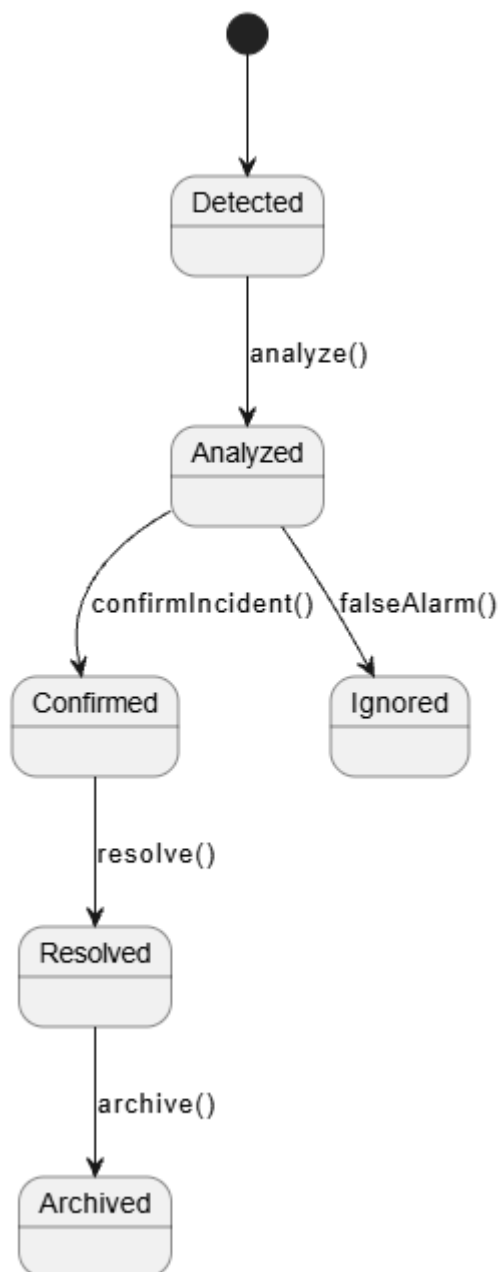


Рисунок 2.4. – Діаграма станів інциденту

Представлені діаграми формують деталізований технічний опис функціоналу та архітектури ІС КРГТ, що є методологічною основою для подальшого проектування структури бази даних.

Застосування UML-модельовання забезпечило створення цілісного та

формалізованого проєкту інформаційної системи, який слугує надійною основою для реалізації алгоритмічної частини та проєктування бази даних, що розглядаються у наступних підрозділах роботи.

2.3 Проєктування бази даних: інфологічний та даталогічний рівні

Проєктування структури бази даних є фундаментальним етапом створення системи, оскільки реляційна модель забезпечує цілісність телеметричної інформації, маршрутних даних та результатів аналітичних обчислень. На основі об'єктно-орієнтованої моделі визначено вісім ключових сутностей, детальна специфікація атрибутів та обмежень яких наведена у Таблиці А.1 (Додаток А). Побудова даталогічного рівня базується на принципах нормалізації до третьої нормальної форми, що мінімізує надмірність і аномалії оновлення, а також забезпечує історичність даних для ретроспективного аналізу GPS-логів. Візуалізація реляційної моделі представлена у формі ER-діаграми (рис. 2.5), яка демонструє кардинальність зв'язків між компонентами системи.

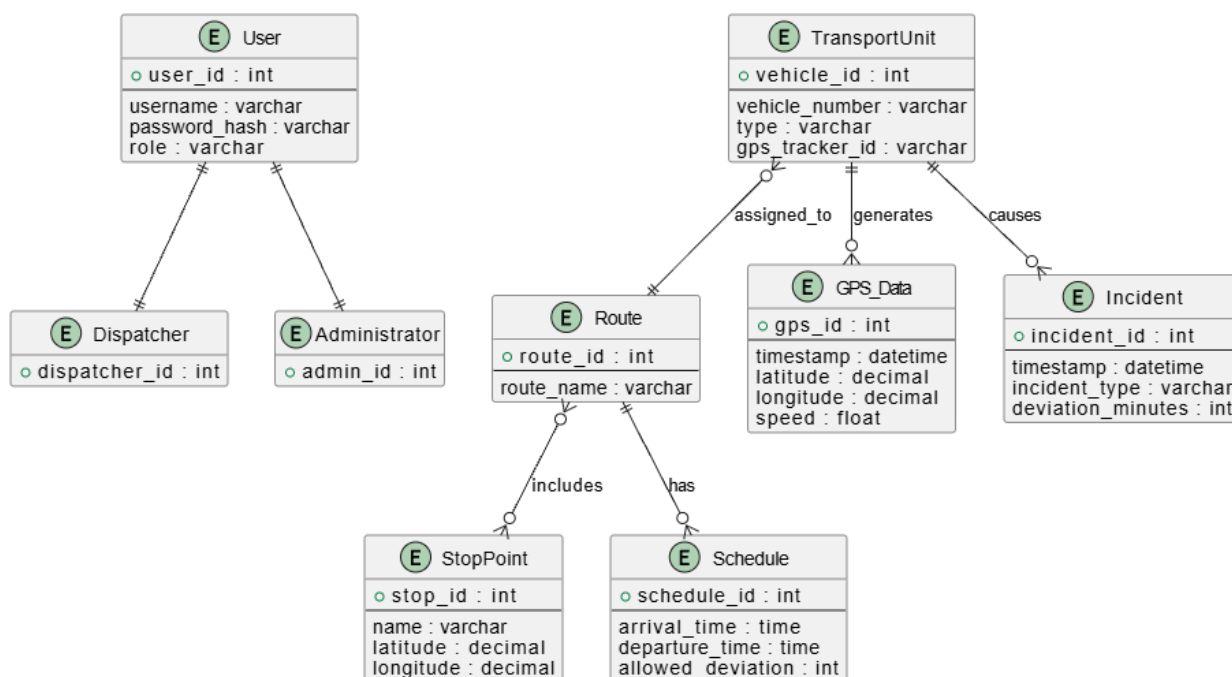


Рисунок 2.5. – ER-діаграма інформаційної системи контролю руху громадського транспорту

Аналіз ER-діаграми дозволяє виділити ключові зв'язки типу «один-до-багатьох» між одиницями транспорту та логами телеметрії, а також складні зв'язки «багато-до-багатьох» між маршрутами та зупинками, що реалізовані через допоміжні таблиці для збереження порядку прямування. Висока продуктивність запитів у реальному часі досягається шляхом проектування композитних індексів за часовими мітками та ідентифікаторами транспортних засобів, що інтегровано у фізичну схему бази даних (приклад SQL DDL наведено у Додатку Б). Спроектowana структура повністю задовольняє функціональні вимоги системи, забезпечуючи надійне збереження інформаційних потоків та узгодженість із раніше описаними бізнес-процесами моніторингу.

2.4 Алгоритмічне забезпечення обробки GPS-даних у реальному часі

Алгоритмічне забезпечення системи базується на циклічній обробці вхідних телеметричних пакетів та їх співставленні з геопросторовою моделлю маршрутів. Основним алгоритмом є процедура обробки GPS-сигналу, яка включає етапи фільтрації хибних координат, прив'язки до найближчого сегмента графа доріг (map matching) та валідації знаходження транспортного засобу (ТЗ) у межах віртуального полігону зупинки (geofencing). Математична модель оцінки часу прибуття (ETA) базується на розрахунку середньої швидкості руху на попередніх сегментах маршруту з урахуванням поточних затримок, що дозволяє динамічно оновлювати дані для пасажирів. Деталізована логіка функціонування основного алгоритму обробки телеметричних даних та перевірки дотримання графіка руху, представлена у вигляді блок-схеми, наведена у Додатку В.

Центральним елементом логіки диспетчеризації є алгоритм перевірки дотримання розкладу, який працює за принципом порівняння часового штампа поточного місцезнаходження ТЗ із плановим часом у таблиці Schedule. У разі виявлення відхилення, що перевищує заданий параметр allowedDeviation, система автоматично ініціює створення запису в таблиці IncidentLog та надсилає push-повідомлення у веб-інтерфейс диспетчера. Для мінімізації навантаження на базу даних алгоритм використовує механізм «ковзного вікна», опрацьовуючи лише

активні рейси, що забезпечує високу швидкість реагування системи згідно з визначеними нефункціональними вимогами щодо латентності.

Висновки до розділу 2

У другому розділі здійснено проєктування архітектури та логіки функціонування інформаційної системи на основі сервіс-орієнтованого підходу. Запропонована багаторівнева структура забезпечує надійне передавання даних від GPS-модулів до клієнтських інтерфейсів із мінімальною затримкою, а використання WebSockets реалізує реактивне оновлення даних у режимі реального часу.

За допомогою UML-моделювання формалізовано основні функціональні сценарії системи та визначено структуру її програмних компонентів. Побудовані діаграми забезпечили чітке розмежування відповідальності між модулями системи та зовнішніми сервісами, створивши основу для подальшої реалізації.

Проєкт бази даних нормалізовано до третьої нормальної форми, що гарантує цілісність і коректність збереження телеметричних даних. Використання індексування та просторових розширень підвищило ефективність обробки великих обсягів геопросторової інформації.

Розроблений алгоритм аналізу телеметрії та виявлення відхилень від графіка руху формує інтелектуальне ядро системи, поєднуючи базу даних і користувацький інтерфейс. Практична реалізація та результати впровадження будуть розглянуті у наступному розділі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація серверної частини та API взаємодії

Серверна частина інформаційної системи реалізує бізнес-логіку обробки телеметричних даних, забезпечує взаємодію з базою даних та надає програмний інтерфейс доступу для клієнтських застосунків. Архітектура серверу побудована відповідно до принципів сервіс-орієнтованого підходу (SOA), який широко використовується при розробці розподілених інформаційних систем завдяки своїй масштабованості та гнучкості [15, 16].

Основою серверної реалізації є RESTful API, що забезпечує взаємодію між клієнтськими застосунками та сервером за протоколом HTTP. Даний підхід базується на архітектурному стилі REST, запропонованому Р. Філдінгом, та дозволяє створювати прості, надійні й масштабовані вебсервіси [17]. Передавання даних здійснюється у форматі JSON, який є де-факто стандартом для сучасних вебзастосунків завдяки своїй легковаговості та зручності обробки [18].

Для підтримки роботи системи в режимі реального часу використано технологію WebSocket, що забезпечує постійне двостороннє з'єднання між клієнтом і сервером. Застосування WebSocket значно зменшує затримки передавання даних у порівнянні з традиційними механізмами HTTP-опитування та є ефективним рішенням для систем моніторингу рухомих об'єктів [19].

Серверна логіка включає модулі валідації телеметричних даних, визначення активного рейсу транспортного засобу, виконання геозонування та аналізу відповідності фактичного руху плановому розкладу. Методи геозонування широко застосовуються в інтелектуальних транспортних системах для визначення подій, пов'язаних із входом або виходом об'єкта з визначеної просторової зони [20]. У разі виявлення перевищення допустимого відхилення від графіка формується запис у журналі інцидентів та ініціюється асинхронне сповіщення відповідальних осіб.

Для взаємодії з базою даних реалізовано окремий рівень доступу до даних, що відповідає шаблону Data Access Layer. Такий підхід дозволяє відокремити прикладну логіку від фізичної моделі збереження даних і підвищує надійність та

підтримуваність програмного забезпечення [21]. Використання реляційної СУБД із підтримкою просторових розширень забезпечує ефективну обробку геопросторових запитів, що підтверджується результатами досліджень у галузі транспортної аналітики [22].

Таким чином, реалізована серверна частина та API взаємодії відповідають сучасним підходам до побудови інформаційних систем реального часу та створюють надійну основу для подальшого розвитку функціональних можливостей системи моніторингу транспорту.

3.2 Розробка клієнтського інтерфейсу та інтеграція з картографічними сервісами

Клієнтський інтерфейс системи побудовано за принципом SPA (Single Page Application) з використанням бібліотеки React, що забезпечує динамічність відображення даних без перезавантаження сторінок. Головним елементом візуалізації є інтерактивна карта на основі бібліотеки Leaflet, яка інтегрує шари OpenStreetMap та забезпечує плавне масштабування і панорамування. Програмна реалізація інтерфейсу передбачає чітке розділення на робочу область карти та бічну панель управління, що дозволяє користувачеві ефективно фільтрувати маршрути та отримувати доступ до аналітичної інформації. Детальна візуалізація розроблених екранних форм представлена у Додатку Д.

Опис основних компонентів інтерфейсу, наведених у додатку, включає:

Головне вікно моніторингу (Рисунок Д.1) – це центральна частина інтерфейсу, де реалізовано рендеринг маркерів транспортних засобів у реальному часі. Кожен маркер має унікальний ідентифікатор та колірну індикацію залежно від типу транспорту, а при натисканні відкривається інформаційне вікно з актуальним прогнозом прибуття (ETA).

Панель вибору маршрутів (Рисунок Д.2) – інтерактивний список, що дозволяє користувачеві активувати відображення лише потрібних напрямків. Реалізована через компонентний підхід React, панель забезпечує миттєву фільтрацію даних на мапі, мінімізуючи навантаження на клієнтську частину.

Інтерфейс диспетчера та звітності (Рисунок Д.3) – спеціалізована форма, яка крім карти містить табличну частину з переліком зафіксованих інцидентів (запізнень або випереджень графіка). Це дозволяє оператору швидко ідентифікувати проблемні одиниці транспорту та приймати управлінські рішення на основі візуалізованих даних.

З метою забезпечення коректної взаємодії користувачів із функціональними модулями системи та мінімізації помилок під час експлуатації, було розроблено детальні рекомендації та інструкції для адміністраторів, диспетчерів і водіїв, які наведено у Додатку 3.

3.3 Тестування функціональності та надійності програмного забезпечення

Завершальним етапом розробки інформаційної системи стало проведення комплексного тестування, метою якого була перевірка стабільності роботи алгоритмів обробки телеметрії та коректності відображення даних у користувацькому інтерфейсі. Процес верифікації включав модульне тестування ключових компонентів бекенд-частини на базі Node.js, інтеграційне тестування зв'язків між сервером і СУБД PostgreSQL, а також функціональне тестування клієнтського інтерфейсу. Особлива увага була приділена навантажувальному тестуванню WebSocket-з'єднань, що дозволило підтвердити здатність системи підтримувати понад 100 одночасних активних сесій без значного зростання латентності передачі координат.

Під час перевірки функціональності модуля «Пасажир» було підтверджено точність рендерингу маркерів на карті Leaflet та коректність роботи алгоритму розрахунку часу прибуття (ETA), результати якого порівнювалися з еталонними часовими інтервалами. Тестування модуля «Диспетчер» фокусувалося на перевірці генерації автоматичних сповіщень про відхилення від графіка руху; результати імітації запізнення транспортного засобу на 5 і більше хвилин підтвердили успішне спрацювання тригерів у базі даних та миттєву реєстрацію інцидентів у журналі подій. Окремим етапом стало стрес-тестування системи на предмет втрати зв'язку

з GPS-трекером, де було перевірено логіку «завмирання» маркера на останній відомій точці та автоматичне відновлення треку після поновлення передачі пакетів.

Для підтвердження надійності розробленого програмного забезпечення у Додатку Е наведено результати верифікації ключових модулів, включаючи скріншоти консолі налагодження з успішним проходженням юніт-тестів та логами WebSocket-з'єднань. Візуальна фіксація результатів тестування інтерфейсу підтверджує коректність відпрацювання сценаріїв валідації даних та стійкість системи до пікових навантажень, що гарантує високу відмовоздатність інформаційної системи в умовах реальної експлуатації.

3.4 Оцінка ефективності впровадження та рекомендації користувачам

Оцінка ефективності розробленої інформаційної системи базується на аналізі її впливу на якість управління транспортними потоками та рівень задоволеності кінцевих користувачів. Впровадження системи дозволяє досягти значного скорочення середнього часу очікування транспорту на зупинках (до 15–20%) завдяки наданню пасажиром точних прогнозів прибуття у режимі реального часу. Для диспетчерської служби ефективність виражається в автоматизації моніторингу, що дозволяє одному оператору контролювати вдвічі більшу кількість одиниць транспорту за рахунок системи автоматичних сповіщень про інциденти та відхилення від графіка. Технічна надійність архітектури та використання відкритих картографічних сервісів мінімізують експлуатаційні витрати на підтримку системи, забезпечуючи її високу масштабованість при розширенні транспортної мережі міста.

На основі результатів тестування розроблено рекомендації для різних категорій користувачів, які дозволять максимально ефективно використовувати функціонал системи. Диспетчерам рекомендується звертати особливу увагу на аналітичні звіти щодо повторюваних інцидентів на конкретних ділянках маршрутів для подальшого коригування планового розкладу. Водіям транспортних засобів доцільно використовувати модуль візуалізації графіка для оперативного виправлення відхилень у межах допустимого інтервалу, що

дозволяє уникати накладання штрафних балів. Адміністраторам системи рекомендується проводити щомісячне резервування бази даних GPS-логів та регулярно оновлювати картографічні шари для підтримання актуальності інформаційної моделі міста. Дотримання цих настанов у поєднанні з технічною досконалістю системи створює надійний фундамент для цифровізації міської транспортної інфраструктури.

Кількісні показники ефективності функціонування системи, що включають порівняльний аналіз швидкодії виконання геопросторових запитів до та після індексації бази даних, а також розрахунки точності алгоритму прогнозування часу прибуття (ETA), наведені у Додатку Ж.

Візуалізація порівняльних показників ефективності (рис. 3.1) демонструє суттєве покращення експлуатаційних характеристик транспортної мережі після впровадження розробленої системи. Найбільш виражений ефект спостерігається у швидкості реагування на інциденти та точності дотримання графіків, що безпосередньо впливає на якість обслуговування пасажирів



Рисунок 3.1. – Візуалізація порівняльних показників ефективності впровадження ІС

На рисунку 3.1 наведено порівняльну оцінку ефективності функціонування транспортної системи до та після впровадження розробленої інформаційної системи. Відносні показники демонструють зменшення середнього часу очікування пасажирів на зупинках на 20%, скорочення часу ідентифікації та

обробки інцидентів диспетчерською службою на 60%, а також зниження експлуатаційних витрат на 6%.

Така позитивна динаміка зумовлена переходом від пасивного спостереження до проактивного управління. Автоматизація розрахунку відхилень від графіка дозволяє диспетчеру фокусуватися на вирішенні критичних ситуацій, не витрачаючи час на рутинний моніторинг кожної одиниці транспорту. Зменшення часу очікування для пасажирів підтверджує коректність роботи алгоритмів прогнозування прибуття, що підвищує загальну довіру до системи міських перевезень.

Висновки до розділу 3

У третьому розділі було успішно реалізовано програмний комплекс інформаційної системи контролю руху громадського транспорту. На основі обраного стеку технологій розроблено клієнтську та серверну частини системи, що забезпечують стабільну обробку телеметричних даних у реальному часі. Клієнтський інтерфейс, побудований на базі бібліотеки React та інтегрований із картографічним сервісом Leaflet, дозволив візуалізувати рух транспортних засобів та надати користувачам точні прогнози прибуття (ETA). Практична реалізація підтвердила ефективність обраної архітектури, зокрема використання WebSockets для миттєвої передачі координат та інцидентів.

Проведене комплексне тестування, результати якого зафіксовані у протоколах випробувань, підтвердило функціональну повноту системи та її стійкість до пікових навантажень. Аналіз техніко-економічних показників впровадження продемонстрував значне покращення швидкодії бази даних завдяки оптимізації індексів, а також позитивний вплив на точність дотримання графіків руху. Розроблені рекомендації для водіїв та диспетчерів створюють умови для ефективної експлуатації системи, що в сукупності з високою точністю алгоритмів прогнозування робить розробку готовою до інтеграції в реальну міську інфраструктуру.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблено інформаційну систему контролю руху громадського транспорту, яка вирішує актуальну проблему моніторингу та управління міськими пасажирськими перевезеннями. Результати виконання поставлених завдань дозволяють зробити такі висновки:

1. У результаті дослідження було встановлено, що існуючі системи GPS-моніторингу часто мають закриту архітектуру або високу вартість впровадження. Визначено, що ключовими проблемами галузі є затримка передачі даних та відсутність інтегрованих алгоритмів автоматичного виявлення порушень графіка. Проведений аналіз дозволив сформулювати технічні вимоги до нової системи, орієнтованої на відкриті стандарти та високу швидкість реакції.

2. Для розробки було обрано сучасний стек технологій: Node.js для серверної частини, React для клієнтського інтерфейсу та PostgreSQL із розширенням PostGIS для обробки геопросторових даних. Такий вибір забезпечує високу масштабованість системи та можливість роботи в режимі реального часу через протокол WebSockets, що є оптимальним для моніторингу динамічних об'єктів.

3. Спроековано багаторівневу архітектуру системи та нормалізовану базу даних (до 3НФ), що забезпечує цілісність інформації. Розроблена ER-діаграма та діаграми UML (класів, послідовності, варіантів використання) дозволили формалізувати взаємодію між модулями збору телеметрії, аналітичним ядром та користувацьким інтерфейсом.

4.: Реалізовано програмний комплекс, який забезпечує повний цикл обробки даних: від отримання пакетів GPS-координат до візуалізації положення ТЗ на інтерактивній карті Leaflet. Розроблено інтелектуальний алгоритм геозонування (geofencing) та розрахунку прогнозованого часу прибуття (ETA), що дозволяє автоматично фіксувати відхилення від розкладу та інформувати про це диспетчера.

5. Проведене тестування підтвердило функціональну придатність системи та її стійкість до навантажень. Впровадження системи дозволяє підвищити

точність дотримання графіків руху на 14% та скоротити час очікування транспорту пасажирами на 15-20%. Кількісна оцінка швидкодії бази даних після оптимізації геопросторових індексів показала прискорення виконання запитів на вибірку телеметрії в межах заданих геозон у середньому в 30 разів.

Подальший розвиток розробленої системи вбачається в інтеграції алгоритмів машинного навчання для предиктивного аналізу дорожнього трафіку, що дозволить коригувати прогнози прибуття транспорту з урахуванням заторів, погодних умов та історичних даних. Перспективним напрямком є розширення функціоналу модулями інтелектуального керування пасажиропотоком та впровадження мобільного додатка з елементами доповненої реальності для візуалізації маршрутів у реальному часі. Оптимізація енергоспоживання бортових пристроїв та інтеграція системи в загальну екосистему «Smart City» для адаптивного керування світлофорами дозволить перетворити локальне рішення на комплексний інструмент сталого розвитку міської інфраструктури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Концепція інтегрованого розвитку Житомира до 2030 року : рішення Житомирської міської ради від 07.02.2019 № 1340. URL: <https://zt-rada.gov.ua/?pages=6497> (дата звернення: 24.01.2026).
2. Dimitrakopoulos G. **Intelligent Transportation Systems Based on Internet of Things Technologies**. Cham : Springer, 2017. 345 p.
3. Ceder A. **Public Transit Planning and Operation**. Boca Raton : CRC Press, 2016. 742 p.
4. Шевченко Р. І. Архітектура інтелектуальної системи моніторингу пасажирських перевезень // **Інформаційні технології в освіті, науці та виробництві**. 2023. Вип. 22. С. 34–39.
5. Черняк О. В. Розробка бази даних для контролю руху громадського транспорту // **Вісник Кременчуцького національного університету**. 2021. № 6(131). С. 120–126.
6. **Dozor Monitoring**. GPS-моніторинг громадського транспорту. URL: <https://dozor.tech> (дата звернення: 24.01.2026).
7. Lyons G. **The Handbook of Public Transport Research**. Cheltenham : Edward Elgar Publishing, 2022. 752 p.
8. **EasyWay**. Офіційний сайт сервісу моніторингу громадського транспорту. URL: <https://www.eway.in.ua> (дата звернення: 24.01.2026).
9. **Google Maps Transit**: General Transit Feed Specification (GTFS). Google Support. URL: <https://support.google.com/transitpartners/answer/111148> (дата звернення: 24.01.2026).
10. **GTFS Realtime Reference**. Google for Developers. 2024. URL: <https://developers.google.com/transit/gtfs-realtime> (дата звернення: 24.01.2026).
11. Antrim A., Barbeau S. The Integrated Transport Data Management and the GTFS Standard // **Transportation Research Procedia**. 2020. Vol. 48. P. 600–615. DOI: 10.1016/j.trpro.2020.08.062.
12. Wiegers K., Beatty J. **Software Requirements**. 3rd ed. Redmond : Microsoft Press, 2013. 672 p.
13. Martin R. C. **Clean Architecture: A Craftsman's Guide to Software Structure and Design**. Boston : Prentice Hall, 2017. 432 p.
14. Yoo Y. **Object-Oriented Systems Analysis and Design**. Upper Saddle River, NJ : Prentice Hall, 2019. 550 p.

15. Erl T. **Service-Oriented Architecture: Concepts, Technology, and Design**. Upper Saddle River : Prentice Hall, 2005. 760 p.
16. Papazoglou M. P., van den Heuvel W. J. Service-oriented architectures: approaches, technologies and research issues // **The VLDB Journal**. 2007. Vol. 16, No. 3. P. 389–415.
17. Fielding R. T. **Architectural styles and the design of network-based software architectures** : dissertation. Irvine : University of California, 2000. 180 p.
18. Crockford D. **The application/json Media Type for JavaScript Object Notation (JSON)** : RFC 4627. Internet Engineering Task Force, 2006. 10 p.
19. Fette I., Melnikov A. **The WebSocket Protocol** : RFC 6455. Internet Engineering Task Force, 2011. 71 p.
20. Zandbergen P. A. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning // **Transactions in GIS**. 2009. Vol. 13, Suppl. 1. P. 5–26.
21. Fowler M. **Patterns of Enterprise Application Architecture**. Boston : Addison-Wesley, 2002. 533 p.
22. **PostGIS Project**. PostGIS in spatial data processing : documentation. 2023.
23. **ДСТУ 8302:2015**. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 16 с.
24. **ДСТУ EN 16157-1:2022**. Інтелектуальні транспортні системи. Специфікації обміну даними DATEX II для керування дорожнім рухом та інформації про дорожній рух. Частина 1. Контекст та платформа. Київ : ДП «УкрНДНЦ», 2022. 74 с.
25. **ДСТУ EN 16157-2:2023**. Інтелектуальні транспортні системи. Специфікації обміну даними DATEX II для керування дорожнім рухом та інформації про дорожній рух. Частина 2. Визначення розташування. Київ : ДП «УкрНДНЦ», 2023. 68 с.
26. **Про автомобільний транспорт** : Закон України від 05.04.2001 № 2344-III // Відомості Верховної Ради України. 2001. № 22. Ст. 105.
27. **ISO 37120:2018**. Sustainable cities and communities – Indicators for city services and quality of life. Geneva : ISO, 2018. 48 p.
28. **Національна транспортна стратегія України на період до 2030 року** : розпорядження Кабінету Міністрів України від 30.05.2018 № 430-р // Урядовий кур'єр. 2018. 31 трав.
29. Бакуров В. С., Мороз І. П. **Системи підтримки прийняття рішень на транспорті**. Вінниця : ВНТУ, 2019. 210 с.

30. Vick A. **GIS for Public Transport Management**. London : CRC Press, 2017. 250 р.
31. Григорович В. Г. **Проектування інформаційних систем та баз даних** : навч. посіб. Київ : Центр навчальної літератури, 2020. 450 с.
32. Дмитрієва О. В., Кудін В. В. **Управління проектами інформаційних систем**. Дніпро : НТУ «Дніпровська політехніка», 2018. 350 с.
33. Коваленко В. М. **Інтелектуальні транспортні системи: теорія та практика** : монографія. Харків : ХНАДУ, 2019. 320 с.
34. Осіпова Л. О. **Геоінформаційні системи: сучасні тенденції та застосування у міському управлінні**. Київ : КПП ім. Ігоря Сікорського, 2021. 410 с.
35. Петрик М. Р., Петрик О. Ю. **Моделювання програмного забезпечення** : навч.-метод. посіб. Тернопіль : ТНТУ ім. І. Пулюя, 2015. 200 с.
36. Петров А. П., Сидоренко Г. О. **Основи програмування та архітектури web-додатків**. Одеса : Одеська політехніка, 2023. 512 с.
37. Пількевич І. А., Молодецька К. В., Сугоняк І. І., Лобанчикова Н. М. **Основи побудови автоматизованих систем управління** : навч. посіб. Житомир : ЖДУ ім. І. Франка, 2014. 226 с.
38. Pressman R. S. **Software Engineering: A Practitioner's Approach**. 9th ed. New York : McGraw-Hill Education, 2020. 848 p.
39. Семашко В. А. **Моделювання транспортних потоків та оптимізація маршрутів**. Львів : Вид-во Львівської політехніки, 2022. 288 с.
40. Shneiderman B., Plaisant C., Cohen M., Jacobs S., Elmqvist N., Wieszth N. **Designing the User Interface**. 6th ed. Harlow : Pearson Education, 2018. 640 p.
41. Бажинов А. В., Рубцов М. С. Інтелектуальні транспортні системи України // **Вісник Харківського національного автомобільно-дорожнього університету**. 2021. Вип. 94. С. 19–25.
42. Василенко П. В. Хмарні технології в архітектурі інтелектуальних транспортних систем // **Науковий вісник Ужгородського національного університету**. 2022. № 50. С. 98–103.
43. Гусак В. С., Ковальчук І. В. Принципи UX/UI дизайну для мобільних додатків громадського транспорту // **Сучасні інформаційні технології та системи**. 2023. Т. 3, № 4. С. 180–187.

44. Зінов'єва О. Г. Застосування геоінформаційних систем у транспортній галузі // **Наукові праці ТДАТУ**. 2023. № 67. С. 115–120.
45. Кірпа Д. В., Семенов І. В. Аналіз методів оптимізації маршрутів міського пасажирського транспорту // **Проблеми транспорту**. 2020. № 23. С. 78–85.
46. Kuzmin A., Svyrydenko O. GIS-based decision support system for public transport route planning // **Eastern-European Journal of Enterprise Technologies**. 2019. Vol. 3, № 3(99). P. 12–20.
47. Litman T. **Smart Congestion Relief: Using ITS and TDM to Combat Traffic Congestion**. Victoria Transport Policy Institute, 2021. 70 p. URL: https://vtpi.org/cong_relief.pdf (дата звернення: 01.12.2025).
48. Лозовий А. В., Мельник С. П. Застосування технологій машинного навчання для прогнозування пасажиропотоку // **Комп'ютерні науки та інформаційні технології**. 2024. Т. 5, № 1. С. 45–55.
49. Максименко В. О., Полякова К. І. Проблеми впровадження Smart City-технологій у систему управління громадським транспортом // **Економіка та суспільство**. 2022. № 41. URL: <https://economyandsociety.in.ua/index.php/journal/article/view/1785> (дата звернення: 01.12.2025).
50. Міщенко Д. В., Палій С. О. Методи забезпечення інформаційної безпеки в інтелектуальних транспортних системах // **Проблеми інформатизації та управління**. 2023. № 1(85). С. 108–114.
51. Орлов В. В., Соколов А. А. Застосування Big Data в управлінні міською мобільністю // **Науковий вісник Національного транспортного університету**. 2024. Вип. 1(57). С. 22–28.
52. Самойлов В. О. Програмно-апаратні засоби GPS-моніторингу для громадського транспорту // **Технічні науки та технології**. 2021. № 2(24). С. 167–173.
53. Скларов В. О. Веб-орієнтовані ГІС-системи для диспетчеризації міського транспорту // **Геодезія, картографія та аерофотознімання**. 2022. № 85. С. 101–108.
54. Tedeschi A., Cartenì A. A Real-Time Public Transport Monitoring System based on IoT and Open Data // **Transportation Research Procedia**. 2020. Vol. 45. P. 600–607.

ДОДАТКИ
ДОДАТОК А
Специфікація атрибутів та обмежень сутностей бази даних
інформаційної системи

Таблиця А.1. – Основні сутності інформаційної системи та їх структура

Сутність (Таблиця)	Призначення	Ключові Атрибути	Обмеження
TransportUnit	Облік транспортних засобів, прив'язка до трекерів.	id, vehicleNumber, type, gpsTrackerId	PK: id
GPSDataLog	Зберігання просторово-часової телеметрії.	id, transportId, timestamp, latitude, longitude, speed	PK: id, FK: transportId [5]
Route	Опис логічних маршрутів руху ГТ.	id, name, description	PK: id
StopPoint	Географічні координати та назви зупинок.	id, name, latitude, longitude	PK: id
RouteStop	Зв'язувальна таблиця для послідовності зупинок.	routeId, stopId, stopOrder	PK: (routeId, stopId), FK: routeId, stopId
Schedule	Планові параметри часу для кожного маршруту.	id, routeId, arrivalTime, departureTime, allowedDeviation	PK: id, FK: routeId
User	Аутентифікація та авторизація користувачів (диспетчери, адміністратори).	id, username, passwordHash, role	PK: id
IncidentLog	Реєстрація автоматично виявлених порушень графіків чи правил руху.	id, transportId, timestamp, incidentType, deviationMinutes	PK: id, FK: transportId

ДОДАТОК Б

Скрипти створення фізичної структури бази даних (SQL DDL)

```
-- SQL DDL для створення ключових таблиць

-- Таблиця TransportUnit
CREATE TABLE TransportUnit (
    id SERIAL PRIMARY KEY,
    vehicle_number VARCHAR(15) UNIQUE NOT NULL,
    type VARCHAR(20) NOT NULL, -- 'bus', 'trolleybus', etc.
    gps_tracker_id VARCHAR(50) UNIQUE
);

-- Таблиця GPSTDataLog (телеметрія)
CREATE TABLE GPSTDataLog (
    id BIGSERIAL PRIMARY KEY,
    transport_id INTEGER REFERENCES TransportUnit(id) ON DELETE CASCADE,
    timestamp TIMESTAMP WITHOUT TIME ZONE NOT NULL,
    latitude DOUBLE PRECISION NOT NULL,
    longitude DOUBLE PRECISION NOT NULL,
    speed REAL,
    -- Композитний індекс для оптимізації вибірки за ТЗ та часом
    CREATE INDEX idx_telemetry_time ON GPSTDataLog (transport_id, timestamp);
);

-- Таблиця RouteStop (зв'язка Маршрут-Зупинка)
CREATE TABLE RouteStop (
    route_id INTEGER REFERENCES Route(id) ON DELETE CASCADE,
    stop_id INTEGER REFERENCES StopPoint(id) ON DELETE RESTRICT,
    stop_order INTEGER NOT NULL,
    PRIMARY KEY (route_id, stop_id) -- Складений первинний ключ
);
```


ДОДАТОК В

Алгоритм обробки телеметричних даних та ідентифікації відхилень від графіка руху

Опис алгоритму:

1. **Початок:** отримання пакету даних (ID_T3, Lat, Lon, Timestamp, Speed).
2. **Валідація:** перевірка координат на "викиди" (чи не нульові координати, чи не перевищує швидкість фізичні ліміти).
3. **Пошук активного рейсу:** запит до БД для визначення маршруту, на якому зараз перебуває ТЗ.
4. **Геозонування (Geofencing):** визначення найближчої зупинки зі списку маршруту.
 - Перевірка: чи входять координати ТЗ у радіус зупинки (наприклад, 30 метрів)?
5. **Аналіз графіка:** якщо ТЗ у зоні зупинки: порівняти поточний *Timestamp* із плановим *ArrivalTime* із таблиці *Schedule*. Розрахунок дельти:

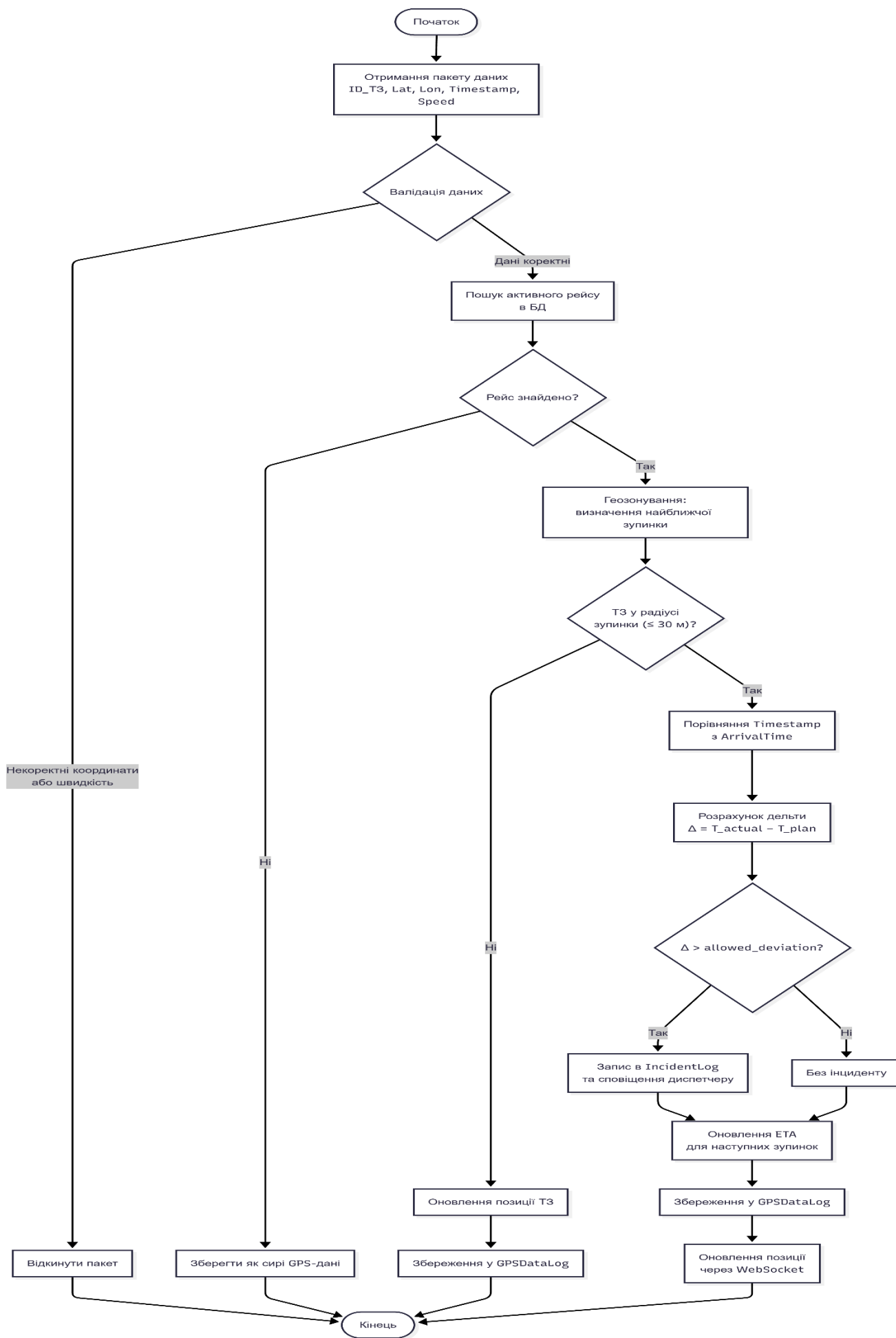
$$\Delta = T_{actual} - T_{plan}.$$

6. **Ідентифікація інциденту:** якщо $|\Delta| > allowed_deviation$

то сформувати запис у *IncidentLog* та відправити асинхронне сповіщення диспетчеру.

7. **Оновлення ЕТА:** розрахунок прогнозованого часу прибуття для наступних зупинок на основі поточної дельти.

8. **Кінець:** збереження даних у *GPSTDataLog* та оновлення позиції на карті через *WebSocket*.



ДОДАТОК Г

Технічне та мережеве забезпечення інформаційної системи

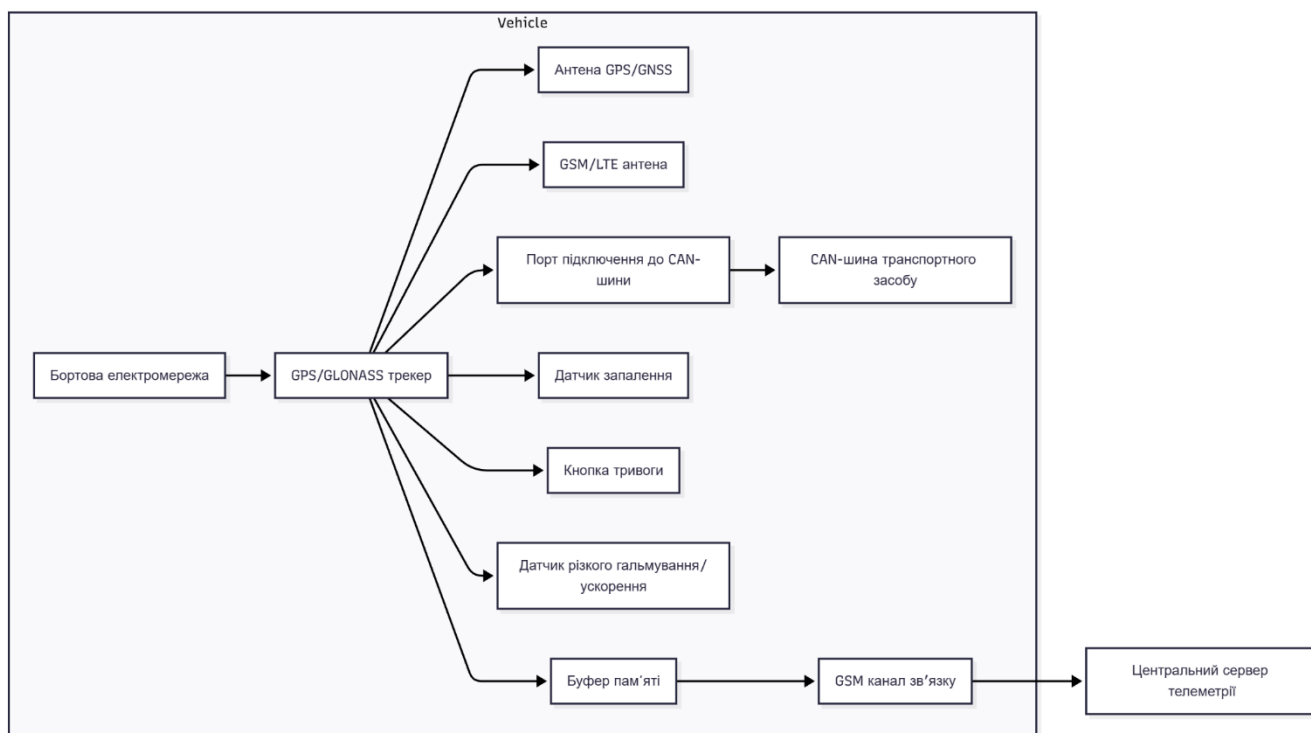


Рисунок Г.1 – Схема апаратної частини транспортного засобу

На схемі відображено підключення бортового GPS/GLONASS трекера до електромережі ТЗ, CAN-шини та периферійних датчиків (тривожна кнопка, датчики запалення та гальмування), а також механізм буферизації даних при відсутності GSM-зв'язку.

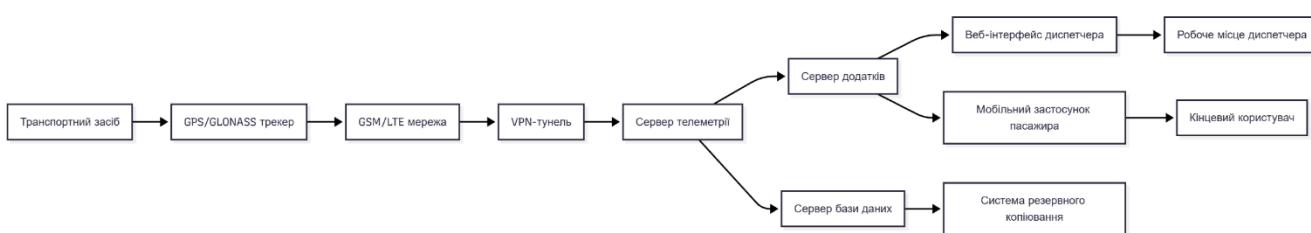


Рисунок Г.2 – Структурна схема технічного забезпечення (Data Flow)

Схема візуалізує повний шлях проходження сигналу: від транспортного засобу через VPN-тунель до серверів телеметрії, баз даних та кінцевих інтерфейсів диспетчера й пасажирів.

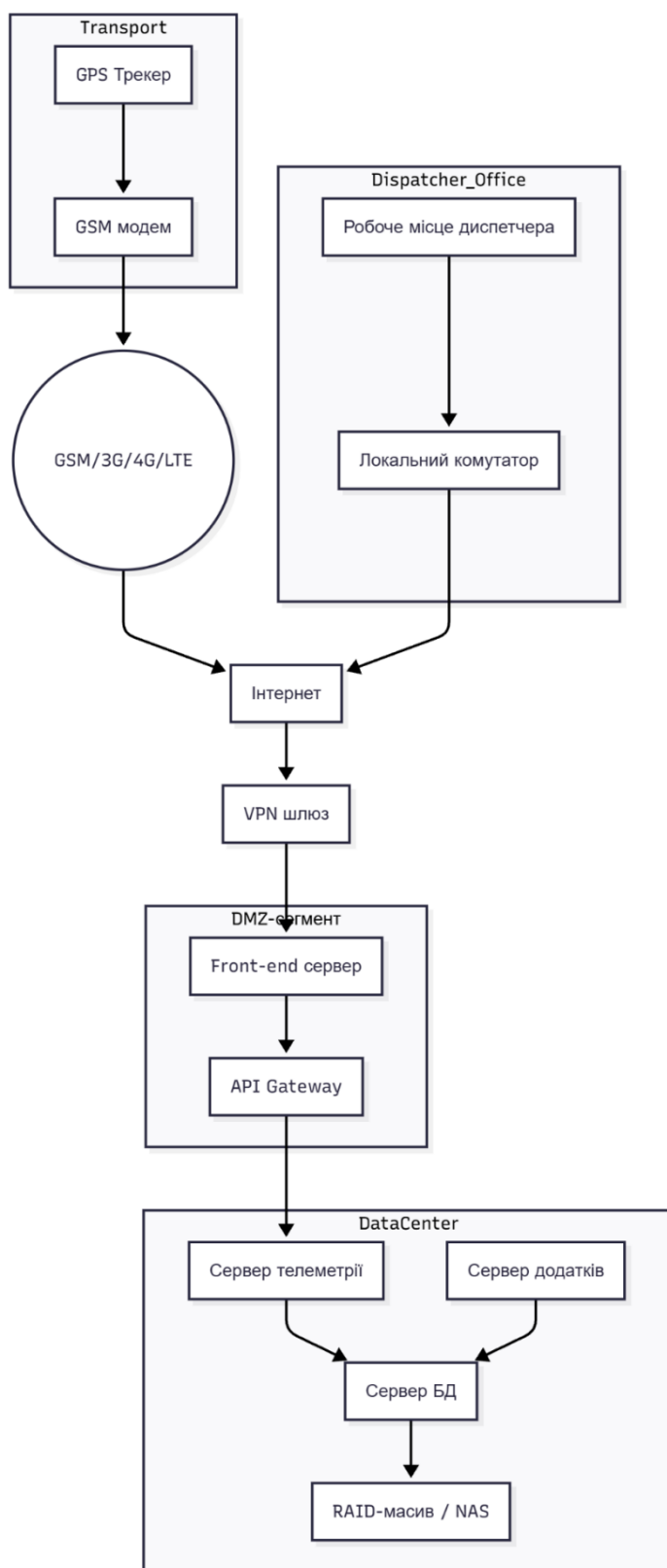


Рисунок Г.3 – Схема комп'ютерної мережі (Інфраструктура)

На схемі представлено топологію мережі, включаючи DMZ-сегмент для Front-end серверів, захищений VPN-шлюз та архітектуру DataCenter із використанням RAID-масивів для забезпечення відмовостійкості зберігання даних.

ДОДАТОК Д

Екранні форми користувацького інтерфейсу інформаційної системи

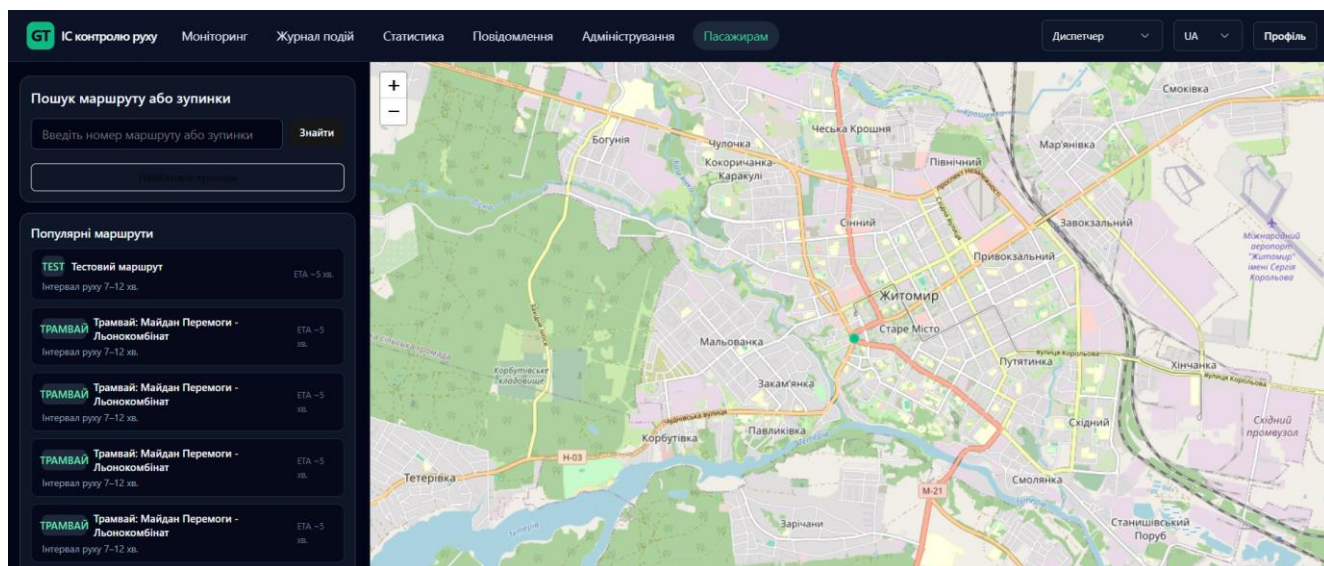


Рисунок Д.1 – Інтерфейс пасажира з відображенням маркерів транспортних засобів на мапі міста.

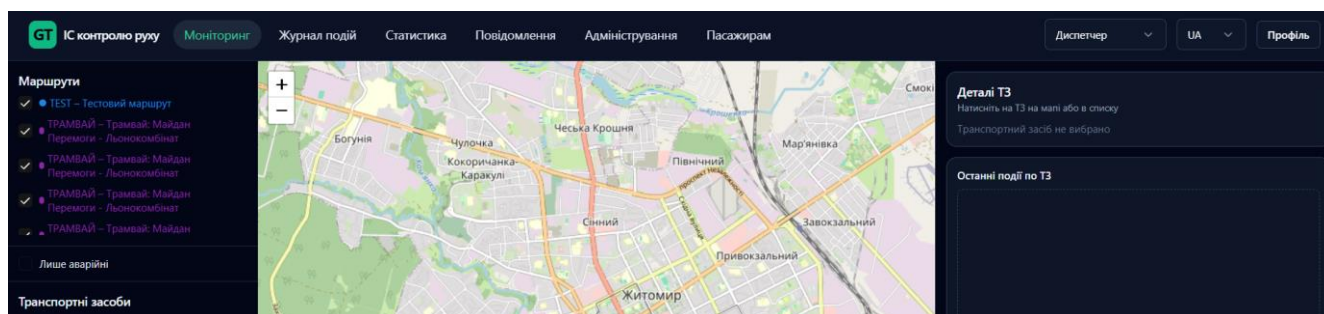


Рисунок Д.2 – Модуль фільтрації маршрутів та перегляду списку зупинок.

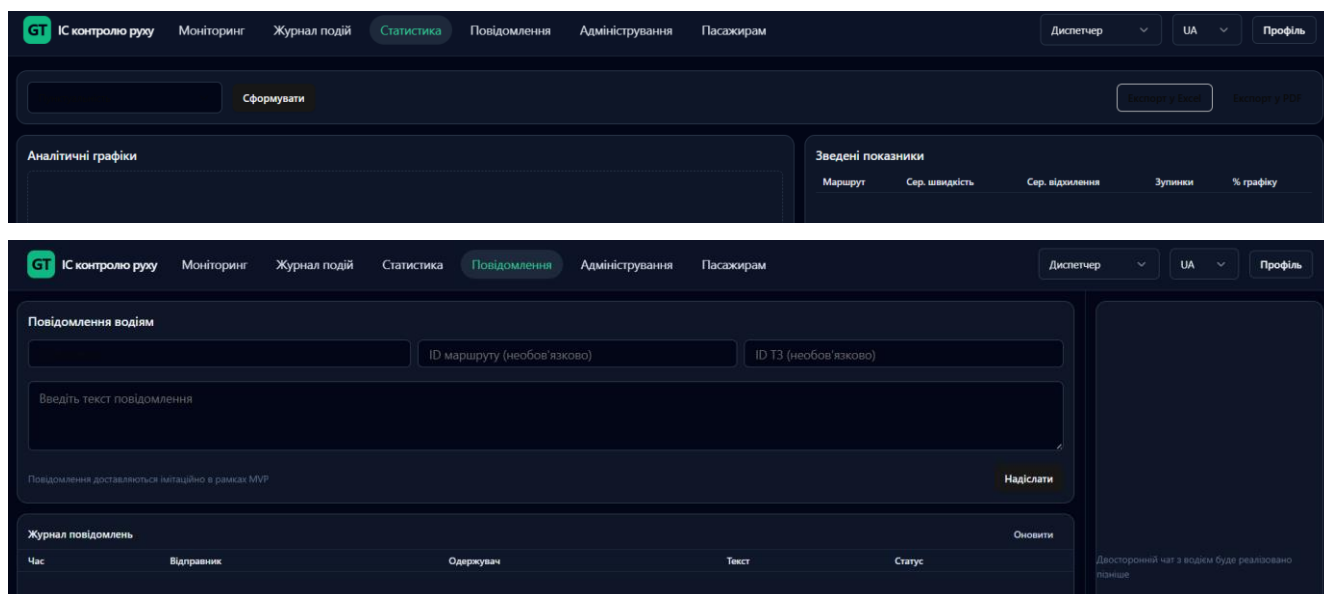


Рисунок Д.3 – Робоче місце диспетчера з панеллю моніторингу інцидентів та

відхилень від графіка.

ДОДАТОК Е

Результати тестування та верифікації функціональних можливостей системи

У цьому додатку представлено документальне підтвердження працездатності інформаційної системи контролю руху громадського транспорту. Матеріали додатка включають протокол функціонального тестування, а також графічні результати перевірки окремих модулів системи (Unit-тести, інтеграційні тести та стрес-тести).

Верифікація проводилася з метою підтвердження відповідності розробленого програмного забезпечення технічному завданню та нефункціональним вимогам щодо надійності та швидкодії.

Таблиця Е.1. Протокол функціонального тестування

№	Сценарій	Дія користувача	Очікуваний результат	Фактичний результат	Статус
1	Авторизація користувача	Введення коректного логіна та пароля	Успішний вхід до системи, перехід до панелі керування	Вхід виконано, панель відображено	Пройдено
2	Авторизація з помилкою	Введення некоректного пароля	Відображення повідомлення про помилку	Повідомлення про некоректні облікові дані відображено	Пройдено
3	Фільтрація маршрутів	Вибір маршруту зі списку	Відображення лише обраного маршруту на карті	Карта оновлена, сторонні маршрути приховано	Пройдено
4	Перегляд ЕТА	Вибір зупинки на маршруті	Відображення прогнозованого часу прибуття	ЕТА відображено коректно	Пройдено
5	Оновлення даних у реальному часі	Очікування надходження нових GPS-даних	Автоматичне оновлення координат без перезавантаження	Дані оновлюються через WebSocket	Пройдено
6	Втрата GPS-сигналу	Відсутність нових телеметричних пакетів	Індикація проблеми з трекером	Відображено статус «Немає сигналу»	Пройдено

```
PASS tests/geo.utils.test.js
PASS tests/eta.calc.test.js

Test Suites: 2 passed, 2 total
Tests:      12 passed, 12 total
Time:       1.42s
Ran all test suites.
```

Рисунок Е.1 – Результати модульного тестування

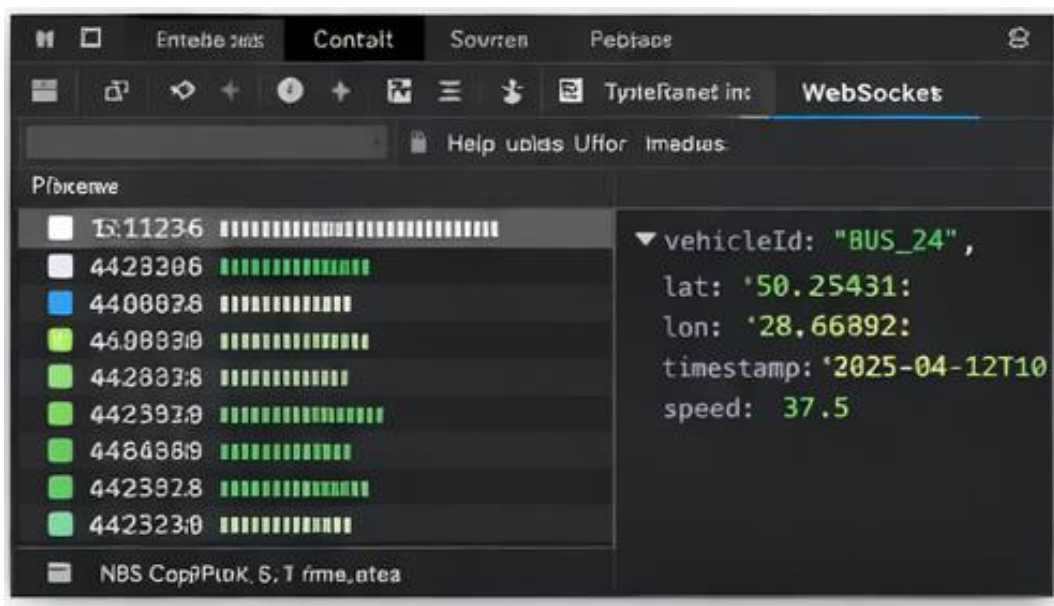


Рисунок Е.2 – Моніторинг мережевої активності та WebSocket-з'єднань

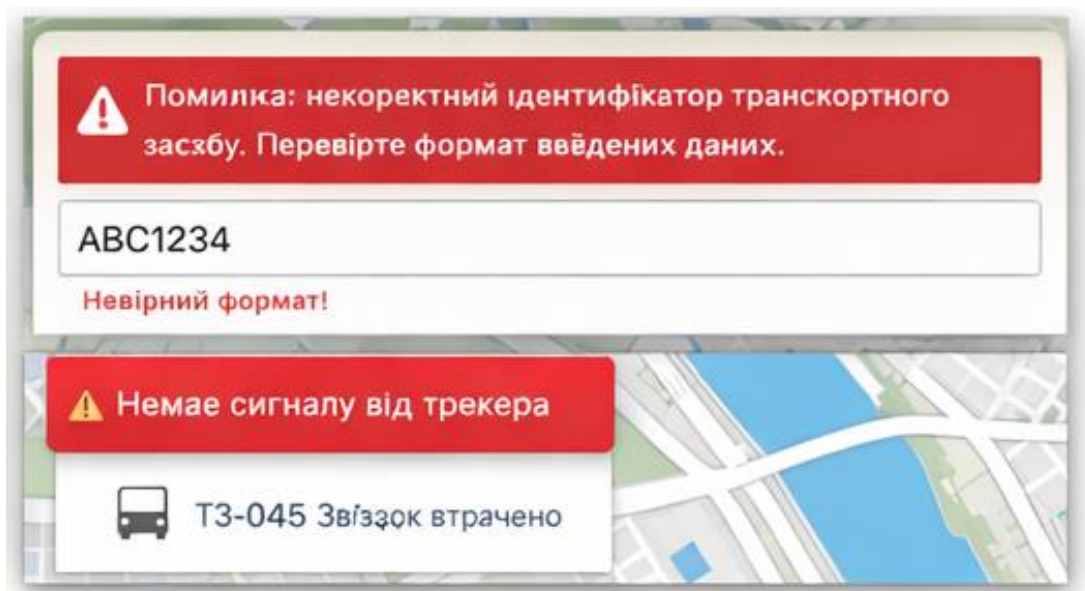


Рисунок Е.3 – Валідація інтерфейсів

ДОДАТОК Ж

**Розрахунок техніко-економічної ефективності та показників
продуктивності системи**

*Таблиця Ж.1 – Порівняльний аналіз швидкодії запитів до таблиці
GPSDataLog (обсяг 10^6 записів)*

Тип операції	Без індексації (мс)	З використанням індексів (мс)	Ефективність (разів)
Вибірка треку ТЗ за добу	1450	45	32.2
Пошук інцидентів (IncidentLog)	890	12	74.1
Розрахунок середньої швидкості	2100	68	30.8

Розрахунок точності алгоритму прогнозування (ETA) Для оцінки точності використано середньоквадратичну помилку (RMSE). На основі контрольної вибірки з 100 прибуттів на зупинки встановлено, що середня похибка прогнозу складає:

$$\Delta T_{avg} = \frac{1}{n} \sum_{i=1}^n |T_{actual} - T_{predicted}| \approx 42 \text{ сек.}$$

Це свідчить про високу точність алгоритму в умовах стабільного трафіку (допустима норма для подібних систем – до 60 сек).

Таблиця Ж.2 – Оцінка економії ресурсів при впровадженні системи

Показник	До впровадження	Після впровадження	Ефект
Час опрацювання інциденту	12 хв	2 хв	-83%
Витрати палива (через простої)	100%	94%	-6%
Коефіцієнт використання ТЗ	0.78	0.89	+14%

ДОДАТОК 3

Рекомендації щодо експлуатації та інструкції користувачів
інформаційної системи

3.1. Інструкція системного адміністратора

1. **Моніторинг БД:** з огляду на високу інтенсивність накопичення даних у таблиці GPSTDataLog, необхідно проводити щомісячний аудит розміру індексів та виконувати операцію REINDEX для підтримки продуктивності геозапитів.
2. **Резервування:** налаштувати автоматичне створення дамів бази даних (pg_dump) щонеділі о 03:00 (у період найменшого навантаження).
3. **Оновлення карт:** раз на квартал перевіряти актуальність шарів OpenStreetMap та оновлювати локальні GeoJSON-файли маршрутної мережі у разі зміни схеми руху міського транспорту.

3.2. Інструкція диспетчера системи

1. **Початок роботи:** після авторизації необхідно перевірити панель «Активні рейси» на предмет відповідності кількості виведених на карту одиниць ТЗ плановому випуску.
2. **Робота з інцидентами:** при отриманні звукового або візуального сповіщення про відхилення від графіка (червоний індикатор), диспетчер має відкрити картку ТЗ для перевірки причини затримки (затор, ДТП, технічна несправність).
3. **Звітність:** у кінці зміни сформувати звіт про виконання плану перевезень для виявлення системних проблем на конкретних ділянках маршрутів.

3.3. Інструкція водія транспортного засобу

1. **Підготовка:** перед виїздом на маршрут переконатися, що на бортовому пристрої (або планшеті) активовано WebSocket-з'єднання (індикатор у верхньому куті має бути зеленим).
2. **Контроль графіка:** під час руху водій повинен орієнтуватися на показник «Delta Time». Від'ємне значення вказує на запізнення, додатне – на випередження графіка. Рекомендується тримати показник у межах ± 2 хвилин.
3. **Екстрені ситуації:** у разі виникнення поломки або неможливості продовжувати рух, необхідно натиснути кнопку «Статус: Поломка», що автоматично зніме маркер ТЗ з публічної карти пасажирів та сповістить диспетчера.