

УДК 004.02; 004.942

DOI: 10.15587/1729-4061.2017.92831

Розроблення моделей та засобів серверної частини системи обліку пасажиропотоку громадського транспорту “розумного” міста

О. Ю. Борейко, В. М. Теслюк, А. Я. Зелінський, О. М. Березький

Розроблено структуру серверної частини системи обліку пасажиропотоку громадського транспорту, алгоритм функціонування та спеціалізоване програмне забезпечення. Розроблено і реалізовано інформаційне та технічне забезпечення, що забезпечує низьку ціну проектного рішення та є оптимальним рішенням з широкими функціональними можливостями. Наведено результати параметрів обліку пасажиропотоку громадського транспорту, отриманих з використанням розробленої системи

Ключові слова: “розумне” місто, система обліку пасажиропотоку громадського транспорту, інформаційна та програмна моделі

1. Вступ

Сучасний розвиток суспільства, проблеми з енергоносіями та жорсткі умови конкурентної боротьби за ринки збуту, потребують від виробників продукції розроблення інтелектуальних систем з використанням сучасної елементної бази. Однією з таких областей, яка динамічно розвивається, – є технології “розумного” міста [1], які, з однієї сторони, дають змогу суттєво економити енергоносії та підвищити рівень управління в місті, а з іншої – покращити рівень комфорту та якість надання послуг користувачам.

“Розумне” місто поєднує в собі аспекти технологічної інфраструктури та “громадських технологій”. Нові методи з використанням смартфонів, ріст популярності онлайн-транзакцій, низька вартість обладнання та P2P-технології створюють передумови для спільної роботи муніципальних органів з містянами для якнайкращого використання ресурсів, збору даних та прийняття ефективних рішень [2].

В структурному аспекті “розумне” місто – це система взаємодіючих систем. До складових технологій “розумного” міста належать високошвидкісні оптичні, сенсорні, провідні та безпроводні мережі [3].

На сьогодні однією з основних технологій для організації “розумних” міст стає технологія Інтернет речей (IoT). На даний час IoT можна розглядати як ключовий тренд світової економіки найближчого десятиліття [4]. Важливу роль у розвитку технології Інтернету речей відіграють інтелектуальні рішення в області міжмашинних комунікацій (M2M). Дана концепція передбачає інтеграцію комунікаційного обладнання з різноманітними пристроями [5].

Щодо побудови автоматизованих систем управління, то “розумне” місто можна розглядати як сукупність кількох вертикальних ринків, таких як транспорт [3], медицина, промислова автоматизація, освіта та енергетика, безпека й логістика [6]. Усі напрацювання в даних галузях можуть бути використані для

побудови ефективного “розумного” міста.

Одним з цікавих прикладів використання технологій міжмашинних комунікацій M2M може виступати система обліку пасажиропотоку громадського транспорту [3]. Такі системи будують на основі використання спеціалізованого обладнання, такого як контролери з давачами для збору та опрацювання інформації, систем позиціонування тощо. Системи обліку пасажиропотоку дають змогу контролювати оплату за проїзд, визначити завантаженість транспорту на маршрутах та оптимізувати роботу громадського транспорту залежно від днів тижня [7].

Отже, технологія “розумного” міста є складною системою. Відповідно, реалізація системи “розумного” міста пов’язана з цілою низкою технічних проблем [8], а саме: проблема з реалізацією низки інтелектуальних функцій, необхідність забезпечення низької вартості технічної системи, забезпечення заданої надійності технічного рішення [9], забезпечення масштабованості проектованої системи тощо.

Відповідно, проектування та реалізація системи обліку пасажиропотоку громадського транспорту (СОПГТ) [3] як складової системи “розумного” міста є актуальною задачею сьогодення.

2. Аналіз літературних даних та постановка проблеми

Основою роботи системи обліку пасажиропотоку громадського транспорту “розумного” міста є клієнт-серверна взаємодія. Така особливість передбачає встановлення та виконання певних вимог, що покладаються як на клієнтську, так і на серверну частину системи для забезпечення коректного та надійного функціонування системи обліку пасажиропотоку в цілому.

Здебільшого системи, призначені для підрахунку пасажиропотоку за критерієм сторони, де безпосередньо відбувається процес підрахунку, можна поділити на два типи [10].

В першому випадку [11], збір і опрацювання даних відбувається на клієнтській стороні системи, а на сервер надсилається конкретний результат, число. Сервер в такому разі служить лише для зберігання даних від клієнтів та відображення зрозумілої статистики та аналітичної інформації по конкретному клієнту, групах клієнтів.

В другому випадку [12], на клієнтській стороні здійснюється лише збір та попереднє опрацювання вхідних даних. Дані структуруються та надсилаються на сервер у “сирому” вигляді. Процес підрахунку тоді відбувається на стороні сервера. Таким чином серверна сторона має більшу кількість функцій і, відповідно, вимоги до надійності та ефективності її роботи зростають.

Запропонована система розроблена з орієнтацією на другий тип, тобто процес підрахунку відбувається на серверній стороні.

Спершу, на стороні клієнта, здійснюється запис відео пасажиропотоку з відеокамер, які розміщені над передніми та задніми дверима транспортного засобу. Після цього, відбувається фотофіксація пасажирів з правом на пільговий проїзд [13], визначаються GPS-координати та інші системні дані. Всі процеси відбуваються шляхом взаємодії контролера, в основі якого працює одноплат-

ний міні-комп'ютер Raspberry Pi [14], із периферійними пристроями клієнтської сторони [15].

Наступними етапами є попереднє опрацювання даних від периферії, структурування, формування пакетів та надсилання пакетів на сервер (WEB та FTP сервер) [16] для їхнього збереження та подальшого опрацювання.

3. Ціль та задачі дослідження

Ціллю роботи є розроблення моделей та засобів серверної частини системи автоматизованого обліку пасажиропотоку громадського транспорту, яка характеризується широкими функціональними можливостями та низькою ціною.

Для досягнення поставленої цілі необхідно розв'язати наступні задачі:

- розробити структуру серверної частини системи обліку пасажиропотоку міського громадського транспорту, яка має ґрунтуватися на модульному принципі;
- побудувати алгоритм функціонування та розробити спеціалізоване програмне забезпечення серверної частини СОПГТ, яке ґрунтується на використанні об'єктно-орієнтованого підходу;
- розробити інформаційну модель, яка забезпечить надійний обмін даними між клієнтом та сервером розробленої системи;
- побудувати технічне забезпечення серверної частини системи обліку пасажиропотоку громадського транспорту, яке забезпечує низьку ціну технічного рішення.

4. Структура системи обліку пасажиропотоку «розумного» міста

В основі розробленої структурної схеми системи обліку пасажиропотоку громадського транспорту “розумного” міста лежить клієнт-серверна взаємодія, що зображено на рис. 1. До клієнтської частини відноситься безпосередньо контролер, що змонтований у транспортному засобі й отримує дані від давачів та водія. Серверна сторона складається із апаратної та програмної реалізації сервера, а також операторів, що контролюють процес опрацювання даних, які надходять від контролерів (клієнтів).

Збір та опрацювання даних у системі обліку пасажиропотоку громадського транспорту відбувається у кілька етапів.

На першому етапі кожен контролер, розміщений на одиниці громадського транспортного засобу, отримує дані з периферії (GPS-модуль, камери). На цьому ж етапі контролер отримує дані від водія. Пізніше, водій натисканням відповідної кнопки активує запуск камери для фіксації пасажирів з правом на пільговий проїзд [17].

Наступним етапом є передача контролером отриманих даних на опрацювання серверу. На стороні сервера усі дані структуруються та зберігаються для подальшого опрацювання їх системою та при потребі операторами.

Завершальним є етап опрацювання даних з подальшим відображенням детальної та чіткої статистики й аналітики про динаміку пасажиропотоку в громадському транспорті за відповідний проміжок часу.

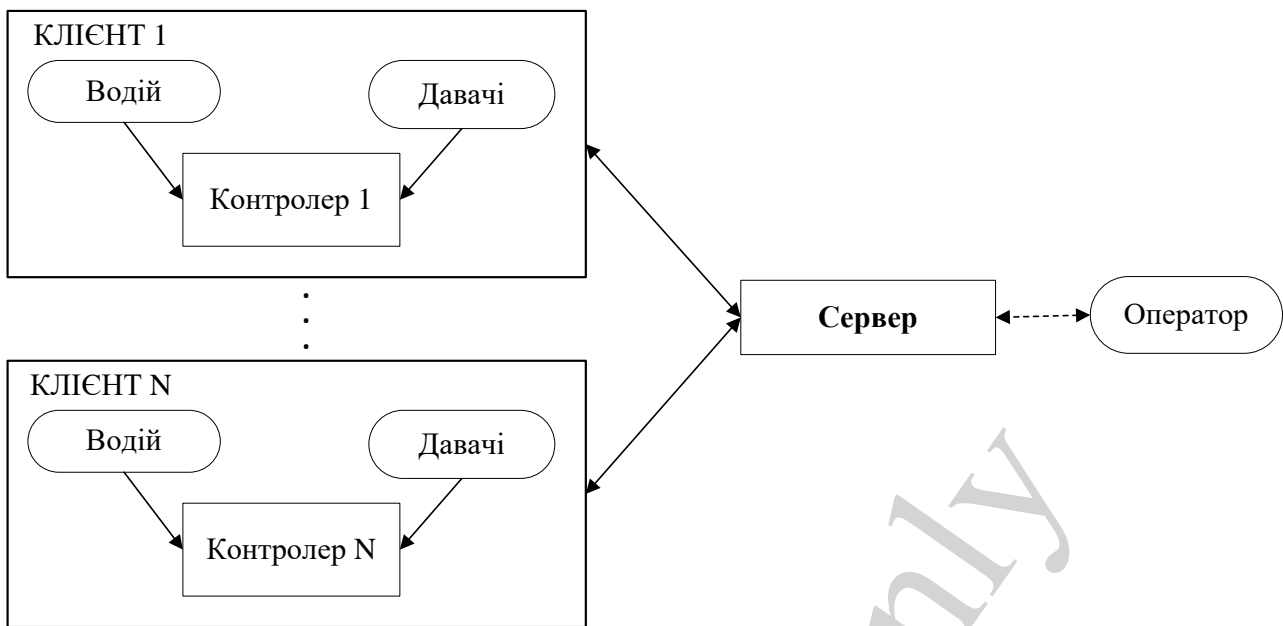


Рис. 1. Структура системи обліку пасажиропотоку “розумного” міста

В процесі реалізації системи обліку пасажиропотоку громадського транспорту “розумного” міста використано модульний принцип, який дає змогу ефективно організувати вдосконалення та модифікацію проектованої системи.

5. Інформаційна модель системи обліку пасажиропотоку громадського транспорту “розумного” міста

Розроблена серверна частина системи включає апаратний сервер з програмними реалізаціями сервера баз даних, сервера збереження файлів та веб-сервера. Опрацювання даних, які надходять від клієнтів, відбувається на серверній стороні. Неповними або неоднозначними даними займаються оператори.

Розроблена інформаційна модель має на меті описати інформаційні процеси, параметри та властивості системи обліку пасажиропотоку громадського транспорту “розумного” міста (рис. 2).

Модель представлена у вигляді схеми, на якій зображені усі основні складові елементи системи, взаємодію між ними та описано рух даних і етапи інтерпретації їх в завершену інформаційну аналітику.

В загальному випадку, від кожного контролера-клієнта надходять дані про подію. Структури даних про подію можуть бути двох типів. В першому випадку подією може виступати факт відкриття передніх або задніх дверей транспортного засобу та натискання спеціальної кнопки для фіксації пільговиків. По факту відкриття дверей, як і по факту натисканням водієм кнопки фіксації пільговиків, на контролері формується звіт, що включає в себе комплекс певних даних. Такими даними є:

- конкретні координати з GPS-модуля місця, де відбулася подія;
- точний час, коли відбулася подія;
- назва відео (якщо відкривалися двері) або фото (після активації кнопкою камери для фіксації пільговиків);

– службові дані (передні/задні двері, посвідчення та інше).

До другого типу належать безпосередньо мультимедійні дані, тобто відео і фото. Таким чином, дані, що надсилаються від клієнта до сервера, – відносяться до першого типу. Вони надходять у вигляді POST-запиту до веб-сервера. Мультимедійні дані, що надходять для збереження на файловий сервер, відносяться до другого типу. Спрощена схема розроблених структур даних наведена на рис. 3.

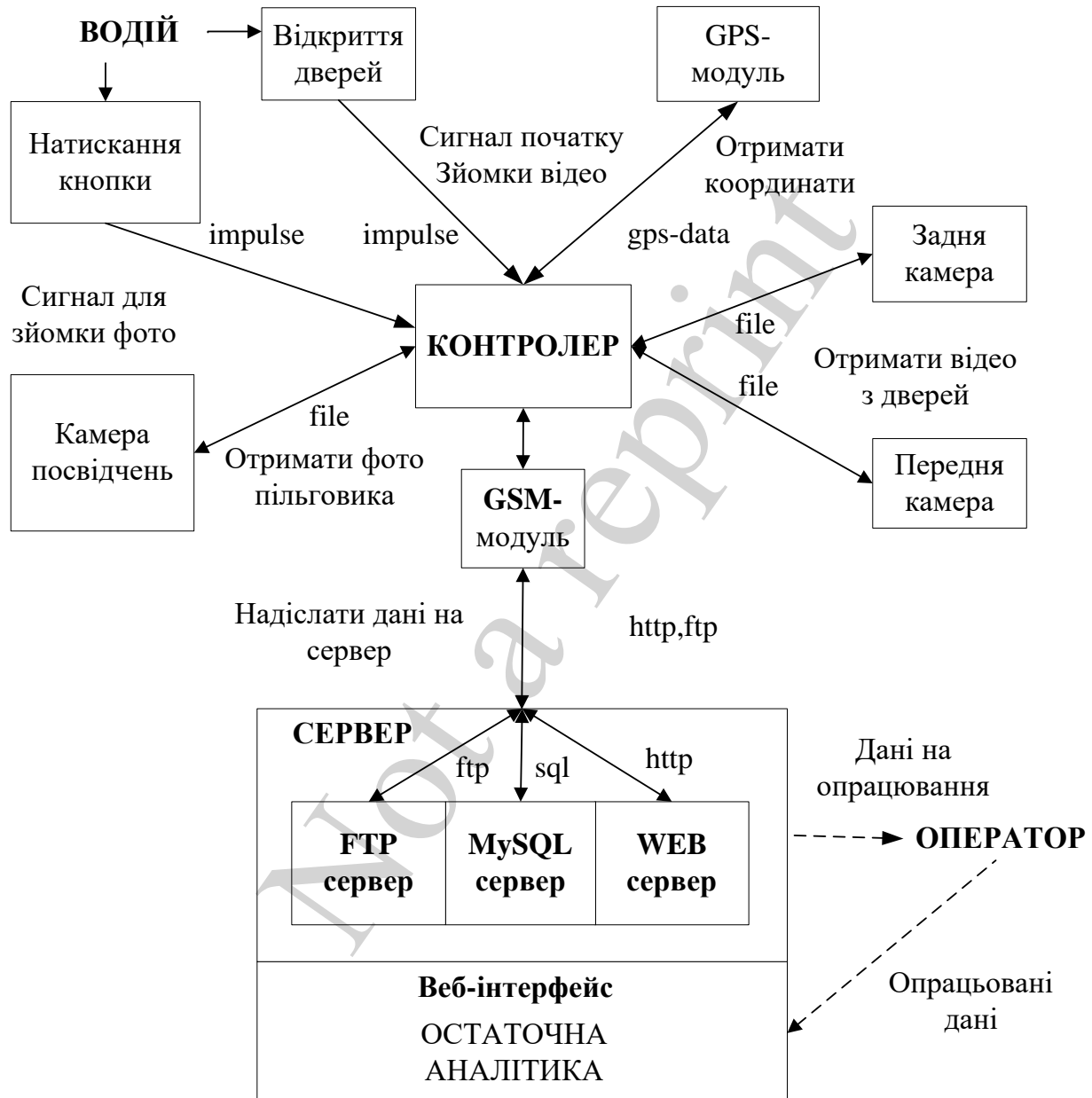


Рис. 2. Структура руху даних в інформаційній моделі системи обліку пасажиропотоку громадського транспорту

Звіти містять інформацію про час події, її місце і сам зміст (інформація про те чи подія є відео з дверей і яких саме чи фото з камери фіксації пільговиків). Звіти формуються у POST-запити до веб-сервера, а мультимедійні файли (фото пільговиків, відео з дверей) надсилаються на FTP-сервер. На сервері всі POST-запити від контролерів зберігаються у базі даних, а файли конвертуються для

опрацювання та зберігаються на дисковому просторі сервера. Опрацьовані статистичні дані зберігаються у вигляді XML-файлів [18].

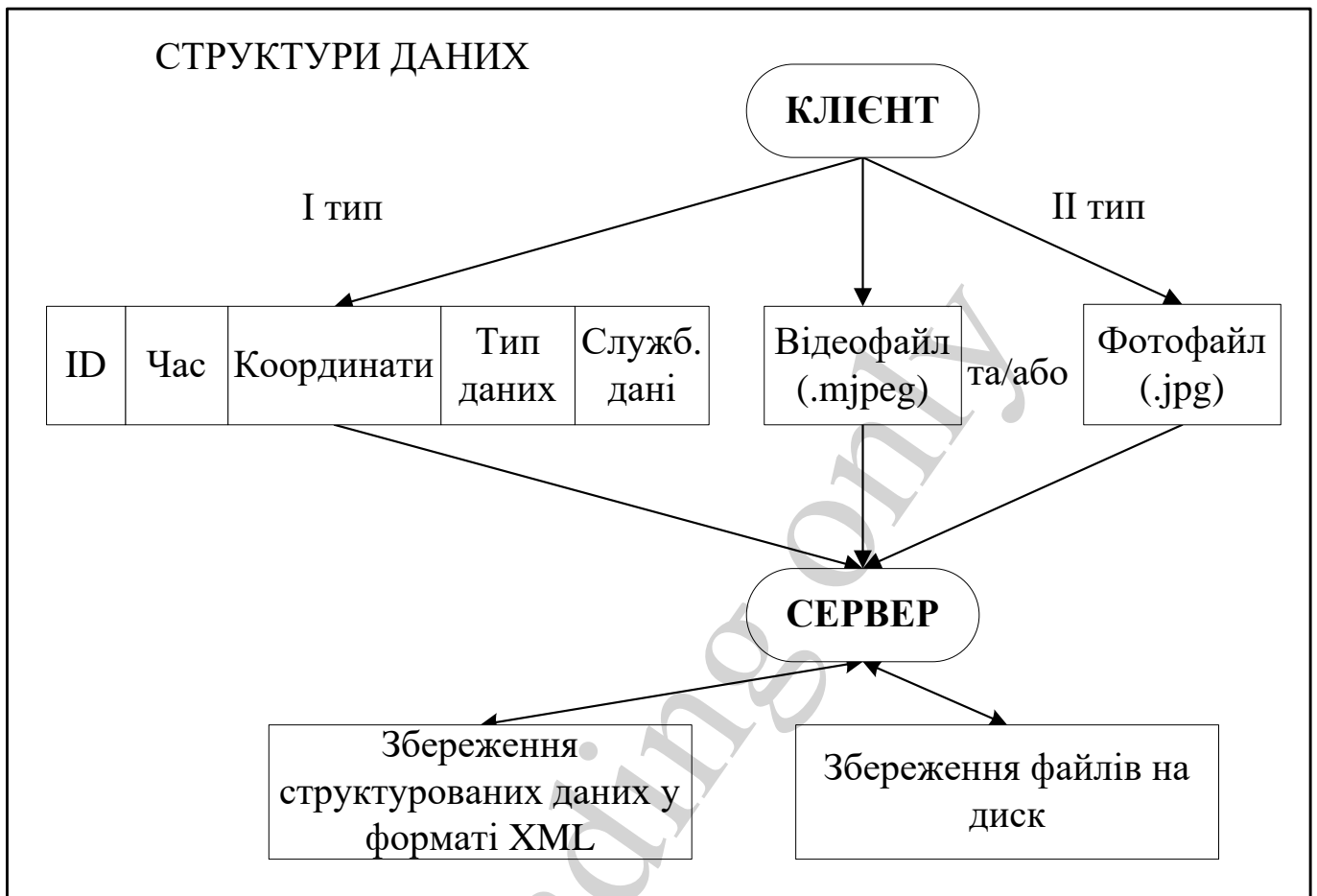


Рис. 3. Типи структур даних системи обліку пасажиропотоку громадського транспорту

Отже, розроблені структури даних двох типів дають змогу контролювати, зберігати та ефективно опрацьовувати параметри пасажиропотоку громадського транспорту “розумного” міста. В процесі реалізації інформаційної моделі використано спискові структури даних.

6. Алгоритм роботи серверної частини системи

Дані, отримані від клієнтської сторони та збережені на серверній частині, перевіряють на їх цілісність, достовірність і коректність. Коли перевірку завершено, частина даних потрапляють на автоматичне опрацювання, а дані, яким характерна якась неоднозначність, – на опрацювання оператору.

Фрагмент блок-схеми алгоритму процесу опрацювання даних від клієнтів на серверній частині зображено на рис. 4.

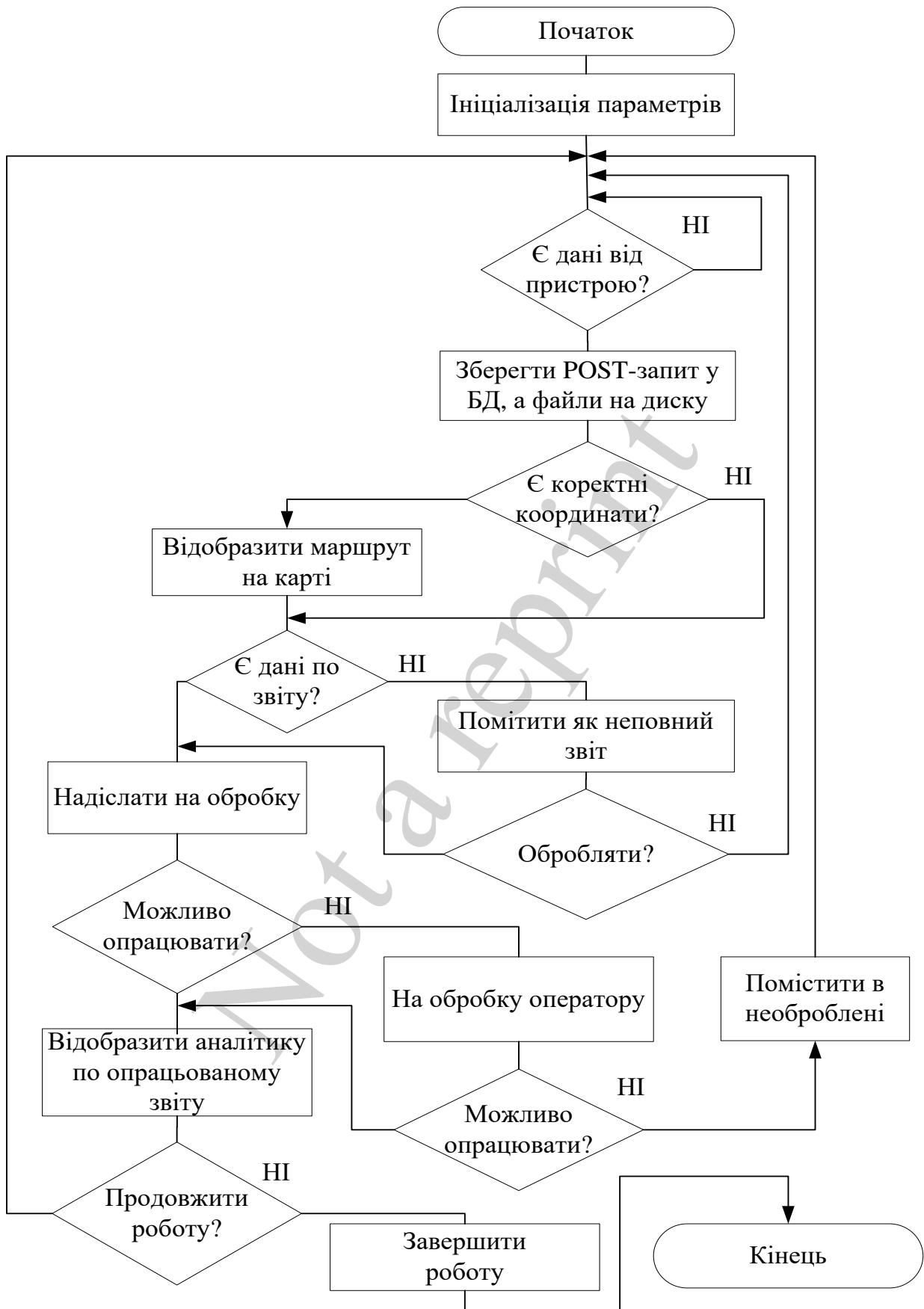


Рис. 4. Блок-схема алгоритму роботи сервера

Розроблений алгоритм роботи серверної частини системи обліку пасажирів

ропотоку громадського транспорту “розумного” міста включає такі кроки (етапи):

1. Ініціалізація параметрів – підготовка усіх апаратних та програмних засобів сервера до отримання даних від клієнтів (запуск програм, завантаження початкових даних і т. п.).

2. Перевірка наявності вхідних даних – програмна перевірка факту здійснення запитів від клієнтів до сервера (WEB-сервера, FTP-сервера).

3. Збереження даних – у разі отримання запитів та даних від клієнтів, зберегти їх у базу даних та дисковий простір сервера, інакше очікувати на запити від клієнтів.

4. Перевірка коректності та повноти даних – аналіз отриманих від клієнтів даних та оцінка можливості подальшої роботи з ними. Наприклад, якщо коректні GPS-координати – будувати GPS-трек для відображення маршруту транспортного засобу на карті, інакше пропустити; надсилати на опрацювання лише дані з доведеною надійністю, повнотою та коректністю.

5. Перевірка можливості автоматичного опрацювання даних – якщо дані отримані від клієнтів відповідають усім заявленим вимогам, то здійснити їх автоматичне опрацювання, інакше надіслати на ручне опрацювання оператором.

6. Обробка даних – власне опрацювання серверною частиною системи даних, отриманих від клієнтів у автоматичному або ручному (за необхідності) режимі.

7. Відображення результатів – звіт та виведення усієї статистичної та аналітичної інформації по кожному конкретному клієнту на основі даних, отриманих від нього.

7. Особливості розроблення програмного забезпечення серверної сторони системи

Основу структури програмного забезпечення (ПЗ) серверної частини системи, становить середовище серверної операційної системи Ubuntu [19] та програмних реалізацій серверів (рис. 5). Мова програмування серверної частини PHP [20]. Веб-фреймворк – Zend Framework [21].

Модель організації програмного забезпечення серверної сторони системи наведено на рис. 6. Дана модель включає три ієрархічні рівні опрацювання та збереження даних від клієнтської сторони. На першому рівні система отримує запити від клієнтів, що надходять через Інтернет. На другому рівні всі запити сортуються у відповідності до їх призначення (POST-запити для веб-сервера, файли на FTP-сервер). На третьому рівні за допомогою засобів мови програмування PHP здійснюється збереження та, за необхідності, вивід через веб-сервер даних отриманих від клієнтської сторони системи.

PHP (Hypertext PreProcessor) або препроцесор гіпертексту – скриптова мова програмування, яку використовують для виконання на серверній стороні. PHP створена для побудови динамічних та інтерактивних інтернет сайтів. Дана мова програмування виявилась дуже гнучкою і потужною, тому стала досить популярною і почала використовуватися у різномасштабних проектах: почина-

ючи від елементарного блогу до великих веб-додатків. Перевагами даної мови є:

- PHP відноситься до вільного програмного забезпечення, яке розповсюджується під спеціальною ліцензією (PHP license);
- легка в освоєнні на кожному з етапів;
- користується підтримкою серед користувачів та розробників;
- присутні суттєві напрацювання по підтримці баз даних (БД);
- реалізовано багато бібліотек, а також розширень даної мови;
- пропонує припустимі ресурси розроблення веб-сесій, програмний інтерфейс розширень;
- може бути розгорнута практично на будь-якому сервері;
- є портованою під велику кількість апаратних та програмних платформ.

Zend Framework – розроблений компанією Zend (USA) (компанія яка здійснює підтримку і координацію проекту PHP) каркас веб-додатку. Zend Framework розширює мову PHP зберігаючи її дух, його головний критерій простота, використані кращі прийоми об'єктно-орієнтованого програмування, дружня ліцензія, і добре протестований код, що швидко виконується. Важливим є те, що у Zend Framework закладено можливість проектування добре захищених, надійних та сучасних додатків WEB 2.0 і веб-сервісів, та всепоглинаючих широко доступних API-функцій від команд лідерів у даній сфері таких як Google, Yahoo!, Amazon, Flickr (USA).

У якості веб-сервера обрано сервер Apache [22]. Сервер Apache є додатком веб-сервера з відкритим вихідним кодом, розробленим Apache Software Foundation (USA). Програмне забезпечення сервера вільно поширюється і дана ліцензія з відкритим вихідним кодом означає, що користувачі можуть редагувати базовий код, щоб налаштувати продуктивність і сприяти розробці програми.

Apache є кросплатформним, та найчастіше використовують у комбінації з операційною системою Linux, яка також поширюється з відкритим вихідним кодом. Ці два компоненти, об'єднані з MySQL [23], базою даних і мовою сценаріїв PHP, утворюють популярне рішення для веб-сервера під назвою LAMP (комплекс серверного програмного забезпечення).

Основними перевагами Apache і серверних систем LAMP є:

- низькі витрати, так як ліцензія на програмне забезпечення безкоштовна;
- гнучке програмування через відкритий вихідний код;
- поліпшена безпека, так як Apache був розроблений для Unix-подібних операційних системи.

Сервер баз даних (SQL-сервер) представлений пакетом MySQL. Даний пакет є добре пристосованим для використання у середовищі веб СУБД (система управління базами даних). Зазвичай на більшості хостингових майданчиків для виконання додатків клієнта, провайдери надають не дуже велику кількість ресурсів (обчислювальних, дискових). Саме тому важливим є використання високоефективної СУБД, якій притаманна висока надійність (переважно веб-додатки і сайти мають працювати у цілодобовому режимі без вихідних).

У зв'язку з переліченими причинами, MySQL перетворилася на непорушний стандарт в області СУБД для веб, а тепер в ній також розвиваються можли-

вості для використання в різних критичних бізнес-додатках.

До переваг MySQL належать:

- багато поточність та можливість підтримки кілька одночасних запитів;
- можливість оптимізувати зв'язки з приєднанням багатьох даних за єдиний прохід;
- підтримка записів з фіксованою або змінною довжиною;
- драйвер ODBC;
- використання гнучкої системи та привілеїв;
- гнучка підтримка, форматів чисел і міток часу;
- швидка робота, легка масштабованість;
- можливий інтерфейс з мовами C і Perl, PHP;
- присутня сумісність з ANSI SQL;
- хороша підтримка провайдерами послуг хостингу;
- швидка підтримка транзакцій завдяки механізму InnoDB.

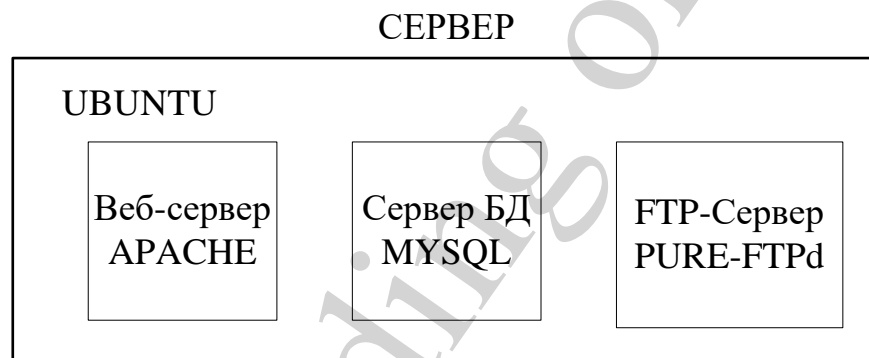


Рис. 5. Структурна модель програмного забезпечення сервера

У якості FTP-сервера використовується Pure-FTPd-сервер [24]. Pure-FTPd – вільний FTP-сервер для операційних систем сімейства UNIX, поширюється по ліцензії BSD. Основна увага приділяється безпеці та простоті настройки.

З вихідного коду компілювався для Linux, OpenBSD, NetBSD, DragonFly BSD, FreeBSD, Solaris, Tru64, Darwin, IRIX і HP-UX, також є під Android.

Відмінною особливістю є те, що сервер не читає налаштування безпосередньо з конфігураційних файлів, а приймає їх тільки з командного рядка. Але можливість використання конфігураційних файлів існує.

Отже, розроблене ПЗ серверної частини системи обліку пасажиропотоку громадського транспорту “розумного” міста забезпечує задану функціональність та є безкоштовним.

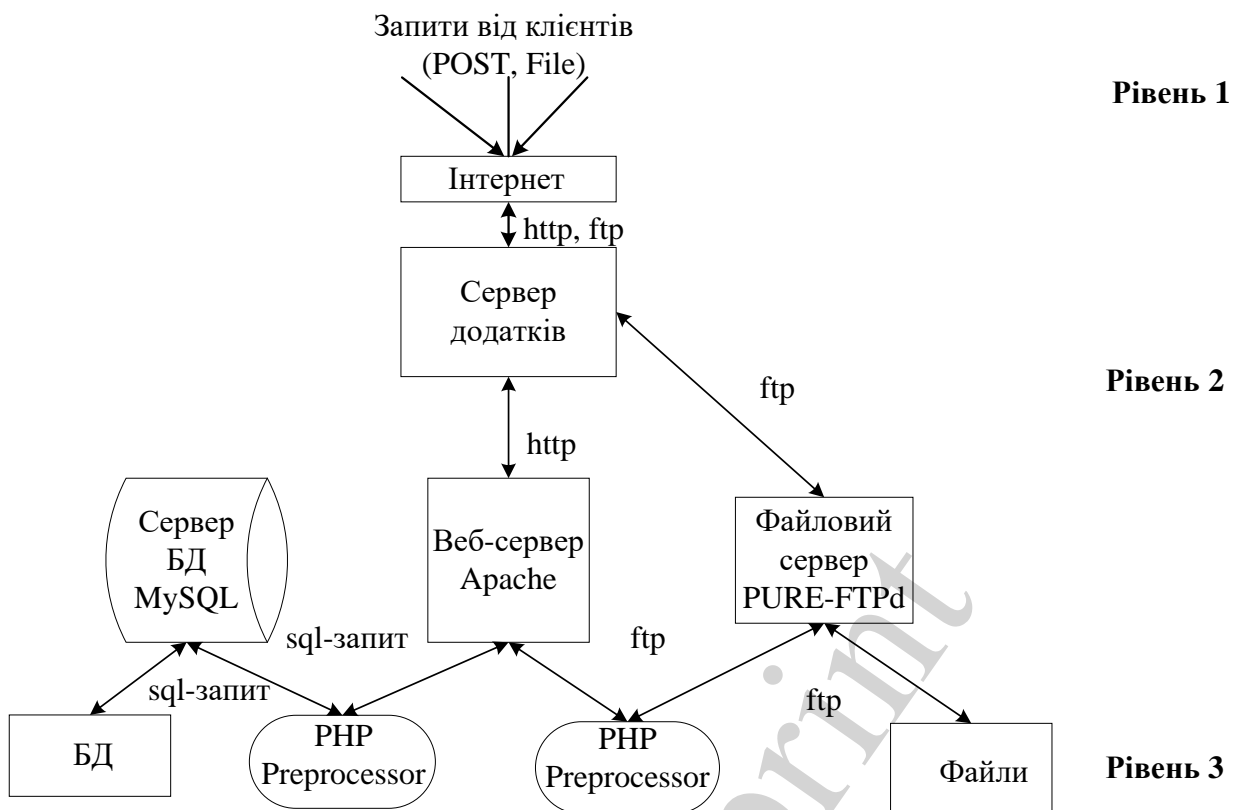


Рис. 6. Багаторівнева модель організації програмного забезпечення серверної частини системи

8. Розроблення апаратного забезпечення серверної частини системи

Аналіз технічного завдання, а також вимог програмного забезпечення серверної частини системи обліку пасажиропотоку громадського транспорту “розумного” міста, дав змогу робити висновки про вимоги до апаратного забезпечення серверної частини системи. Важливою умовою залишається збереження можливості постійного масштабування, нарощення технічних можливостей обладнання для подальшого розвитку проекту.

Основними параметрами сервера, відповідно і критеріями його вибору, є:

- 1) ЦП (центральний процесор), на основі якого він працює;
- 2) ТИП та обсяг ОЗП (оперативний запам'ятовуючий пристрій);
- 3) обсяг ПЗП (постійний запам'ятовуючий пристрій).

Отже, обраним центральним процесором для роботи сервера є Intel Pentium G3260 (USA) [25]. Основні технічні характеристики наведені у табл. 1.

Таблиця 1

Технічні характеристики процесора Intel Pentium G3260

Кількість ядер	2
Кількість потоків	2
Базова тактова частота процесора	3.30 GHz
Кеш-пам'ять	3 MB
Частота системної шини	5 GT/s DMI2
Розрахункова потужність	53 W

Оперативний запам'ятовуючий пристрій представлено Kingston DDR3-1600 8 Гб (USA) (дві мікросхеми по 4 Гб) [26]. Параметри пристрою наведено в табл. 2.

Таблиця 2
Технічні характеристики ОЗП Kingston DDR3–1600 4 Гб

Обсяг пам'яті	4 Гб
Тип пам'яті	DDR3 SDRAM
Напруга живлення	1.5 В
Частота пам'яті	1600 МГц
Ефективна пропускна здатність	12800 Мб/с
Схема таймінгів пам'яті	CL11

Постійний запам'ятовуючий пристрій становить обсяг 500 Гб. Всі основні характеристики апаратного забезпечення сервера СОПГТ “розумного” міста зведено у табл. 3.

Таблиця 3
Технічні характеристики сервера

Процесор	Intel Pentium G3260 3.3ГГц (2 ядра)
Чіпсет	Intel H81
Оперативна пам'ять	8 Гб DDR3
Відео	Intel HD Graphics
Аудіо	Realtek ALC662, HD, 5.1-channel
Мережевий адаптер	1 x 10/100/1000 Mbit
Порти	На передній панелі: 2 x USB 2.0 На задній панелі: 2 x PS/2 1 x VGA 1 x DVI 4 x USB 2.0 1 x RJ-45 3 x audio
ПЗП	500 Гб HDD
Блок живлення	Зовнішній мережевий адаптер на 60 Вт
Розміри	210×62×210 мм

Отже, розроблене технічне забезпечення серверної частини СОПГТ: забезпечує низьку ціну технічного рішення, надає можливості постійного масштабування та нарощення потужності обладнання в процесі розвитку системи.

9. Обговорення результатів: приклади застосування та перспективи розвитку системи

Розроблені засоби системи дають змогу відобразити маршрут транспортного засобу в реальному масштабі часу, що, як приклад, зображено на рис. 7. В будь-який момент часу можна отримати інформацію про місце знаходження вибраного транспортного засобу. Разом з тим, побудована система дає можливість отримати звіт за конкретним маршрутом, де відображені дані про дату, початковий і кінцевий час руху, прізвище водія тощо (рис. 8).

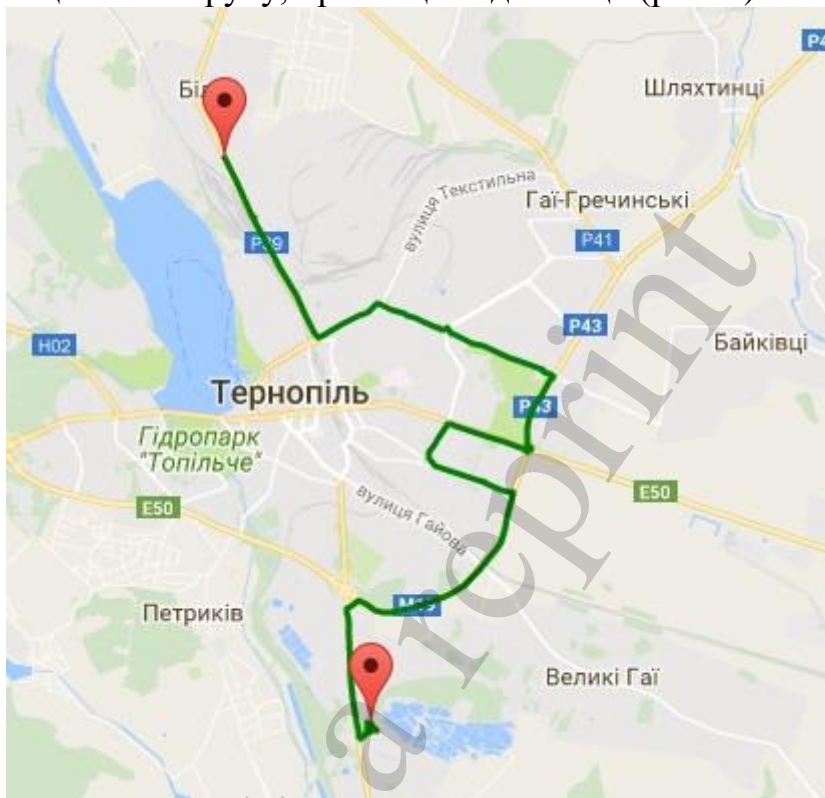


Рис. 7. Приклад відображення маршруту переміщення транспортного засобу

Опрацювання подій								Транспортні засоби		Історія опрацювання		Аналітика		Статистика		ВИХІД	
Виберіть транспортний засіб		Виберіть маршрут															
Фільтрувати								Вчора		Сьогодні							
Дата	Транспортний засіб	Водій	Маршрут	Віезд		Опрацьовано											
				З (год)	До (год)												
2016-12-08	BO 3524 AA (ver. 0.6) (детальна аналітика)	-	Карпенка Подоляни (19)	02:38:16	08/12/2016	21:36:26	08/12/2016	100 %									
2016-12-08	BO 0769 AA (ver. 1.1) (детальна аналітика)	-		00:00:00	08/12/2016	23:59:59	08/12/2016	Не опрацьовано									
2016-12-08	BO 0775 AA (ver 0.6) (детальна аналітика)	-	Карпенка Подоляни (19)	02:26:52	08/12/2016	22:29:11	08/12/2016	100 %									
2016-12-08	BO 0701 AA (ver 0.6) (детальна аналітика)	-	Карпенка Подоляни (19)	03:06:31	08/12/2016	22:52:48	08/12/2016	100 %									

Рис. 8. Приклад меню програмного засобу СОПГТ “розумного” міста з параметрами маршрутів

Приклад меню з статистикою маршруту зображено на рис. 9. Відповідно, сформовані звіти з використанням програмної частини системи дають повну картину параметрів пасажиропотоку кожного з маршрутів та їхні сумарні показники.

Дата	Транспортний засіб	Водій	Маршрут	Візд		Опрацьовано
				З (год)	До (год)	
2016-04-21	BO 3524-AA (new) (детальна аналітика)	-	Миру - Миськлікарня №2 (3)	05:37:08 21/04/2016	20:04:36 21/04/2016	100 %

Коло	Візд		Трек	Тривалість	Простій	Пройдений шлях	Кількість пасажирів					
	З (год)	До (год)					Повні	Діти	діти н/в	Пильговики	Учнівські	Всього
1	05:37:08 21/04/2016	08:05:26 21/04/2016	🌐	2:28:18	00:09:27	15.2 км.	53	1	0	9	4	67
2	08:05:26 21/04/2016	09:14:30 21/04/2016	🌐	1:9:4	00:01:10	16.02 км.	77	3	0	7	3	90
3	09:16:26 21/04/2016	10:23:11 21/04/2016	🌐	1:8:41	00:02:32	16.64 км.	47	4	0	14	2	67
4	10:27:44 21/04/2016	11:37:06 21/04/2016	🌐	1:13:55	00:02:29	16.66 км.	63	2	0	11	1	77

© 2015 ProtectionGroup Всі права захищено

Рис. 9. Приклад меню програмної системи з статистикою маршрутів

Наведені сформовані звіти з використанням розробленої системи, відображають повну картину пасажиропотоку маршруту транспортного засобу. Розроблена система обліку пасажиропотоку громадського транспорту апробована на АТП «Менс-Авто» та «Еталон» у м. Тернопіль (Україна).

Стосовно подальшого розвитку системи, то він може здійснюватися шляхом оптимізації її роботи по двох основних напрямках – програмному та апаратному.

Можливий перехід з мови програмування PHP на краще синтаксично та структурно організовану мову Python [27]. Це в свою чергу приведе до використання одного з популярних й ефективних Python веб-фрейворків. Для прикладу, такими фреймворками можуть бути Django, Flask [28, 29] і т. д. Збір даних про роботу серверної частини системи та їх детальний аналіз є важливим етапом функціонування розробленої системи. Такі дані можуть вказати на певні недоліки в процесі збільшення кількості клієнтів. Це в свою чергу може підняти питання про зміну веб-сервера. Вирішенням проблеми може бути використання альтернативного веб-сервера, наприклад Nginx [30].

Щодо розвитку апаратного забезпечення СОПГТ “розумного” міста, він полягає в масштабуванні апаратних потужностей. Подібне масштабування може досягатися шляхом збільшення обсягів оперативного запам’ятовуючого пристрою і постійного запам’ятовуючого пристрою. Також можлива серйозніша оптимізація у вигляді використання апаратного забезпечення, яке ґрунтується на роботі потужнішого центрального процесора.

10. Висновки

1. Розроблено структуру серверної частини системи обліку пасажиропотоку міського громадського транспорту. Дана структура ґрунтується на модульному принципі, що забезпечує просту й негайну заміну конкретного модуля в разі виходу його з ладу. Як наслідок, досягається підвищена надійність функціонування системи в цілому, а також безперешкодне масштабування та нарощування розробленої системи в майбутньому.

2. Побудовано алгоритм функціонування серверної частини СОПГТ. Його характерними особливостями є використання системного підходу щодо здійснення опрацювання вхідних даних та автоматизації роботи людино-машинної системи. Завдяки цим особливостям доступною стала можливість повної перевірки коректності опрацьованих вхідних даних та чіткого й зрозумілого відображення результатів обчислення параметрів пасажиропотоку. Застосування автоматизації опрацювання вхідних даних дало змогу підвищити ефективність роботи системи за рахунок покриття множини випадків неповноти та/або пошкодженості вхідних даних. Розроблено спеціалізоване програмне забезпечення серверної частини системи обліку пасажиропотоку громадського транспорту. ПЗ реалізує усі вищезазначені функції пристрою та ґрунтується на використанні сучасних підходів об'єктно-орієнтованого програмування, в тому числі використанні веб-фреймворків.

3. Розроблено інформаційне забезпечення, яке дає змогу забезпечити надійний обмін даними між клієнтом та сервером системи. Інформаційне забезпечення включає комплекс сучасних технологій та протоколів. До таких технологій відносяться збір відеоданих з використанням IP-камер, передача даних з допомогою 3G, збереження їх у реляційній БД та на дисковому просторі FTP-сервера, опрацювання даних з використанням спискових структур даних та збереження статистики у формі XML-файлів. Відповідно, розроблене інформаційне забезпечення ґрунтується на використанні сучасних протоколів збору, передачі, опрацювання та збереження даних (TCP/IP, MySQL Client/Server Protocol, HTTP, FTP тощо). Отже, використані технології та протоколи дають змогу ефективно реалізувати передачу і опрацювання даних та відповідають усім заявленим до системи вимогам.

4. Розроблено технічне забезпечення серверної частини СОПГТ. Завдяки використанню сучасних підходів до розроблення ПЗ та оптимізації БД, вдалося зменшити навантаження на ЦП й ОЗП сервера. За рахунок цього стало можливим здійснити підбір оптимального, у питанні ціна-якість, апаратного сервера. Також зроблено акцент на модульності апаратного забезпечення серверної частини для безперешкодного подальшого масштабування. Відповідно, розроблене технічне забезпечення забезпечує низьку ціну технічного рішення і ґрунтується на використанні недорогих, але таких, що повністю забезпечують надійне функціонування системи.

Література

1. Zhuhadar, L. The next wave of innovation – Review of smart cities intelligent operation systems [Text] / L. Zhuhadar, E. Thrasher, S. Marklin, P. O. de

Pablos // *Computers in Human Behavior*. – 2017. – Vol. 66. – P. 273–281. doi: 10.1016/j.chb.2016.09.030

2. Byun, J. Smart City Implementation Models Based on IoT Technology [Text] / J. Byun, S. Kim, J. Sa, S. Kim, Y.-T. Shin, J.-B. Kim // *Advanced Science and Technology Letters*. – 2016. – Vol. 129. – P. 209–212. doi: 10.14257/astl.2016.129.41

3. Mazur, V. Planning of routes based on distribution of passenger flows in time and space [Text] / V. Mazur // *2016 XII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. – 2016. doi: 10.1109/memstech.2016.7507541

4. Gauer, A. Smart city architecture and its applications based on IoT [Text] / A. Gauer, B. Scotney, G. Parr, S. McClean // *Procedia Computer Science*. – 2015. – Vol. 52. – P. 1089–1094. doi: 10.1016/j.procs.2015.05.122

5. Park, Y. Analysis on Smart City service technology with IoT [Text] / Y. Park, S. Rue // *Korea institute of information Technology Review*. – 2015. – Vol. 13, Issue 2. – P. 31–37.

6. Nowicka, K. Smart City Logistics on Cloud Computing Model [Text] / K. Nowicka // *Procedia – Social and Behavioral Sciences*. – 2014. – Vol. 151. – P. 266–281. doi: 10.1016/j.sbspro.2014.10.025

7. Boreiko, O. Structural model of passenger counting and public transport tracking system of smart city [Text] / O. Boreiko, V. Teslyuk // *2016 XII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. – 2016. doi: 10.1109/memstech.2016.7507533

8. Stefanovych, T. The reliability model for failure cause analysis of pressure vessel protective fittings with taking into account load-sharing effect between valves [Text] / T. Stefanovych, S. Shcherbovskykh, P. Drozdziel // *Diagnostyka*. – 2015. – Vol. 16, Issue 4. – P. 17–24.

9. Mulyak, O. Influence of software reliability models on reliability measures of software and hardware systems [Text] / O. Mulyak, V. Yakovyna, B. Volochiy // *Eastern-European Journal of Enterprise Technologies*. – 2015. – Vol. 4, Issue 9 (76). – P. 53–57. doi: 10.15587/1729-4061.2015.47336

10. Olaverri-Monreal, C. Intelligent Technologies for Mobility in Smart Cities [Text] / C. Olaverri-Monreal // *Hiradastechnika Journal*. – 2016. – Vol. 71. – P. 29–34.

11. Jamil, M. S. Smart Environment Monitoring System by Employing Wireless Sensor Networks on Vehicles for Pollution Free Smart Cities [Text] / M. S. Jamil, M. A. Jamilb, A. Mazharc, A. Ikrama, A. Ahmeda, U. Munawar // *Procedia Engineering*. – 2015. – Vol. 107. – P. 480–484. doi: 10.1016/j.proeng.2015.06.106

12. Gilmore, S. Validation of Automatic Vehicle Location Data in Public Transport Systems [Text] / S. Gilmore, D. Reijnsbergen // *Electronic Notes in Theoretical Computer Science*. – 2015. – Vol. 318. – P. 31–51. doi: 10.1016/j.entcs.2015.10.018

13. Pasquale, G. D. Innovative Public Transport in Europe, Asia and Latin America: A Survey of Recent Implementations [Text] / G. Di Pasquale, A. S. dos Santos, A. G. Leal, M. Tozzi // *Transportation Research Procedia*. – 2016. – Vol. 14.

– P. 3284–3293. doi: 10.1016/j.trpro.2016.05.276

14. Leccese, F. A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX [Text] / F. Leccese, M. Cagnetti, D. Trinca // Sensors. – 2014. – Vol. 14, Issue 12. – P. 24408–24424. doi: 10.3390/s141224408

15. Kamble, K. P. Smart vehicle tracking system [Text] / K. P. Kamble // International Journal of Distributed and Parallel systems. – 2012. – Vol. 3, Issue 4. – P. 91–98. doi: 10.5121/ijdps.2012.3410

16. Bischof, S. Semantic Modelling of Smart City Data [Text] / S. Bischof, A. Karapantelakis, C.-S. Nechifor, A. Sheth, A. Mileo, P. Barnaghi // Proc. W3C Workshop on the Web of Things. – 2014. – P. 1–5. – Available at: <http://www.w3.org/2014/02/wot/papers/karapantelakis.pdf>

17. Boreiko, O. Developing a controller for registering passenger flow of public transport for the "smart" city system [Text] / O. Boreiko, V. Teslyuk // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 6, Issue 3 (84). – P. 40–46. doi: 10.15587/1729-4061.2016.84143

18. Denysyuk, P. Usage of XML for Fluidic MEMS Database Design [Text] / P. Denysyuk // 2007 International Conference on Perspective Technologies and Methods in MEMS Design. – 2007. doi: 10.1109/memstech.2007.4283450

19. Scale out with Ubuntu Server [Electronic resource]. – Ubuntu. – Available at: <https://www.ubuntu.com/server>

20. Котеров, Д. В. PHP 5 [Текст] / Д. В. Котеров, А. Ф. Костарев. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2008. – 1104 с.

21. Documentation [Electronic resource]. – Available at: <https://framework.zend.com/learn>

22. Laurie, B. Apache: The Definitive Guide [Text] / B. Laurie, P. Laurie. – 3-rd ed. – Publisher: O'Reilly Media, 2002. – 590 p.

23. Яргер, Р. MySQL и mSQL. БД для небольших предприятий и Интернет [Текст] / Р. Яргер. – СПб.: Символ-Плюс, 2000. – 560 с.

24. Pure-FTPd [Electronic resource]. – Available at: <https://www.pureftpd.org/project/pure-ftpdoc>

25. Intel Pentium Processor G3260 (3M Cache, 3.30 GHz) [Electronic resource]. – Intel. – Available at: http://ark.intel.com/ru/products/87356/Intel-Pentium-Processor-G3260-3M-Cache-3_30-GHz

26. Серверная память Kingston [Электронный ресурс]. – Kingston Technology Corporation. – Режим доступа: <http://www.kingston.com/ru/memory/server>

27. Лутц, М. Программирование на Python. Т. 1 [Текст] / М. Лутц. – 4-е изд. – СПб.: Символ-Плюс, 2011. – 992 с.

28. Гринберг, М. Разработка веб-приложений с использованием Flask на языке Python [Текст] / М. Гринберг. – М.: ДМК, 2014. – 272 с.

29. Чан, У. Django. Разработка веб-приложений на Python [Текст] / У. Чан, П. Биссекс, Д. Форсье. – СПб.: Символ-Плюс, 2015. – 456 с.

30. Welcome to NGINX Wiki's documentation! [Electronic resource]. – Available at: <https://www.nginx.com/resources/wiki/>