

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Комп'ютерних наук

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавра

(назва освітнього ступеня)

на тему: Інтелектуальна інформаційно-технологічна система паркування
автомобілів університетського кампусу

Виконав(ла): студент(ка) 4 курсу, групи СНС-42
спеціальності 122 "Комп'ютерні науки"

(шифр і назва спеціальності)

Парій В.І.

(підпис)

(прізвище та ініціали)

Керівник

Мацюк О.В.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Шимчук Г.В.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

Рецензент

Стухляк П.Д.

(підпис)

(прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
« » 20__ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавра
(назва освітнього ступеня)

за спеціальністю 122 "Комп'ютерні науки"

(шифр і назва спеціальності)

студенту Парію Віталію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна інформаційно-технологічна система паркування автомобілів університетського кампусу

Керівник роботи Мацюк Олександр Васильович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 16 » березня 2022 року № 4/7-161

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Інтелектуальна система паркування автомобілів IoT 1.1. Інтелектуальні системи паркування 1.2. Зразок системи паркування для університетського містечка 1.3. Архітектура системи паркування 2. Огляд реалізацій ІСП 2.1. Хмарний рівень 2.2. Рівень веб-серверів OSGi 2.3. Рівень мобільних додатків 3. Безпека життєдіяльності, основи хорони праці.

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема роботи 2. Приклад платформи високого рівня розумного міста на основі IoT

3. Інтелектуальна система паркування автомобіля IoT для розумного міста 4. Операційна платформа інтелектуальних автостоянок 5. Система паркування автомобілів університету на базі infoStation 6. Система паркування автомобілів університету на базі infoStation 7. Вид високого рівня прикладного рівня хмарної системи паркування автомобілів 8. Хмарні рішення для послуги паркування 9. Схема послідовності веб-додатків 10.-11. Діаграма класів мобільного додатка 12. Висновок

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Безпека життєдіяльності, основи охорони праці | Гурик Олег Ярославович, к.т.н., доцент, доцент кафедри МТ | | |
| | | | |
| | | | |

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|-------|---|--------------------------------|-----------------|
| 1. | Ознайомлення з завданням до кваліфікаційної роботи | 24.01.2022 | <i>Виконано</i> |
| 2. | Підбір джерел по темі дослідження | 04.01.2022-30.01.2022 | <i>Виконано</i> |
| 3. | Переклад та опрацювання джерел по темі дослідження | 31.01.2022-06.02.2022 | <i>Виконано</i> |
| 4. | Виконання дослідження щодо інформаційних-технологічної системи паркування | 12.06.2021-13.06.2021 | <i>Виконано</i> |
| 5. | Оформлення розділу «Інтелектуальна система паркування автомобілів IoT» | 14.02.2022-06.03.2022 | <i>Виконано</i> |
| 6. | Оформлення розділу «Огляд реалізацій ІСП» | 07.03.2022-03.04.2022 | <i>Виконано</i> |
| 7. | Виконання завдання до підрозділу «Безпека життєдіяльності» | 04.04.2022-17.04.2022 | <i>Виконано</i> |
| 8. | Виконання завдання до підрозділу «Основи охорони праці» | 18.04.2022-01.05.2022 | <i>Виконано</i> |
| 9. | Оформлення кваліфікаційної роботи | 02.05.2022-15.05.2022 | <i>Виконано</i> |
| 10. | Нормоконтроль | 16.05.2022-22.05.2022 | <i>Виконано</i> |
| 11. | Перевірка на плагіат | | <i>Виконано</i> |
| 12. | Попередній захист кваліфікаційної роботи | | <i>Виконано</i> |
| 13. | Захист кваліфікаційної роботи | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент

(підпис)

Парій В.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Мацюк О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Інтелектуальна інформаційно-технологічна система паркування автомобілів університетського кампусу // Кваліфікаційна робота// Парій Віталій Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2022 // сторінки __-, рисунки ____, таблиць ____, джерел ____.

Ключові слова: Інтернет речей, розумні міста, інтелектуальні транспортні системи, автостоянка, проміжне програмне забезпечення.

У роботі представлена загальна концепція використання хмарних сервісів інтелектуального паркування автомобілів у розумних містах як важливого застосування парадигми Інтернету речей.

Цей тип послуг є невід'ємною частиною загальної операційної платформи IoT для розумних міст завдяки своїм функціям, які мають виключно бізнес-орієнтацію.

Окреслено види високого рівня запропонованого проміжного програмного забезпечення та проілюстрована відповідна операційна платформа. Описано хмарну інтелектуальну систему паркування автомобілів для використання в університетському містечку разом із деталями дизайну, впровадження та експлуатації.

ANNOTATION

Intelligent information technological car parking system for the university campus // Qualification work // Parii Vitalii Ihorovych // Ivan Pulyu Ternopil National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Science, group. CHc-42 // Ternopil, 2022 // pages ____, figures ____, tables ____, sources ____.

Key words: Internet of Things, smart cities, intelligent transport systems, parking, middleware.

The paper presents a general concept of using cloud services for intelligent car parking in smart cities as an important application of the Internet of Things paradigm.

This type of service is an integral part of the overall IoT operating platform for smart cities due to its business-oriented features.

The types of high level of the proposed middleware are outlined and the corresponding operating platform is illustrated. A cloud intelligent car parking system for use on campus is described, along with details of design, implementation and operation.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ШІ – штучний інтелект.

ІКТ – інформаційно-комунікаційні технології.

CC – Cloud Computing – хмарні обчислення.

IoT – Internet of Things – інтернет речей.

SC – Smart City – розумні міста.

SaaS – Software as a Service – програмне забезпечення як послуга.

POP – Post Office Protocol – протокол, що використовується клієнтом для доступу до повідомлень електронної пошти на сервері.

SMTP – Simple Mail Transfer Protocol – комунікаційний протокол для пересилання електронної пошти.

HTTP – Hyper Text Transfer Protocol – протокол передачі даних.

Зміст

| | |
|-------|---|
| | Вступ |
| 1 | Інтелектуальна система паркування автомобілів IoT |
| 1.1 | Інтелектуальні системи паркування |
| 1.2 | Зразок системи паркування для університетського містечка |
| 1.3 | Архітектура системи паркування |
| 1.3.1 | Хмарний рівень |
| 1.3.2 | Рівень веб-серверів OSGi |
| 1.3.3 | Рівень мобільних додатків |
| 2 | Огляд реалізацій ІСП |
| 2.1 | Хмарний рівень |
| 2.2 | Рівень веб-серверів OSGi |
| 2.3 | Рівень мобільних додатків |
| 2.4 | Результати |
| 2.4.1 | Хмарний рівень |
| 2.4.2 | Рівень веб-серверів OSGi |
| 2.4.3 | Рівень мобільних додатків |
| 3 | Безпека життєдіяльності, основи охорони праці |
| 3.1 | Санітарно-гігієнічні вимоги до умов праці на комп'ютеризованих робочих місцях |
| 3.2 | Психофізіологічне розвантаження для працівників |
| | Висновки |
| | Перелік використаних джерел |

ВСТУП

Пошук місця для паркування на стоянці в комерційних зонах є дуже неприємним заняттям для багатьох людей. Більше проблем посилюється з високими темпами зростання реєстрації нових автомобілів у всьому світі. Є багато паркомісць за сучасними технологіями, але багато водіїв не знають ні про паркувальні блоки, ні про систему паркування. Щоб подолати ці проблеми перевантаження транспорту та заощадити паливо, унікальним кроком вирішення є розумна парковка.

Інтелектуальну або розумну систему паркування необхідно запропонувати для пошуку вільних місць для паркування або збираються звільнитися.

У системі розумного паркування користувачі можуть отримати доступ до даних, щоб визначити наявність місць для паркування, а потім оплатити їх за допомогою мобільних телефонів. Також додаток надає всі можливі дефіцитні маршрути для паркування з будь-якого району міста.

Таким чином, розумне паркування покращує використання наявних паркінгів, що призводить до більшого доходу для власників паркувань. Це також приносить користь для навколишнього середовища та відіграє важливу роль у створенні екологічно чистого середовища..

Кілька років тому зусилля щодо індивідуальних систем паркування були локальними успішними, але нескоординованими, працюючи у власних підприємницьких вакуумах, не скориставшись перевагами універсально застосовних ідей для масштабування своєї діяльності в усьому місті або в усьому світі.

Розрив між існуючими програмами паркування та більш поширеним плануванням транспортної системи в глобальному масштабі є величезною втраченою можливістю для міст зменшити викиди, пов'язані з транспортом.

Завдяки смартфонам, які охоплюють все більшу і більшу частину світового ринку телекомунікацій як у країнах, що розвиваються, так і в

розвинених країнах, підприємці програмного забезпечення можуть збирати й аналізувати дані та надавати споживачам ідеї та інформацію абсолютно новими способами, які не вимагають встановлення нового обладнання.

Основною проблемою у впровадженні систем Smart Parking є системна інтеграція через широку різноманітність залучених апаратних і програмних платформ і, отже, є великою загрозою чи проблемою для масштабованості системи.

Технологічна платформа, що підтримує ці системи, що складається з безлічі апаратних датчиків, систем динамічного обміну повідомленнями та пристроїв контролю трафіку, бездротових, комп'ютерних клієнтів, серверів, драйверів обладнання та інтерфейсів програм.

Дозволити всім цим пристроям від тисяч різних виробників спілкуватися та об'єднати їх разом в одну платформу – це найбільша проблема у зниженні вартості та складності інтелектуального паркування. Різноманітність інфраструктурних апаратних і програмних систем, які необхідно інтегрувати, величезна і доповнює її.

Ще один важливий момент походить від постачальників електронних платежів. Ці системи обробки платежів надають електронні платежі на основі дозволів, як правило, за плату.

Занадто велика кількість цих рішень для розміщення - це масштабованість, здатність процесора транзакцій підтримувати широкі географічні, ринкові та сервісні зони з мінімальними витратами.

1 ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПАРКУВАННЯ АВТОМОБІЛІВ ІоТ

1.1 Інтелектуальні системи паркування

ІСП та інші системи, такі як системи електричної енергії, системи водопостачання та газопостачання, міські системи протипожежного захисту та безпеки, системи eHealth/mHealth тощо, надають інтелектуальні послуги ІоТ, щоб робить місто розумнішим [17]. ІСП розташовується на бізнес-рівні ІоТ, обмінюється даними з хмарним інформаційним центром «розумного міста» і надає користувачам «найкращі» транспортні послуги, такі як моніторинг і контроль руху, планування маршрутів, послуги паркування тощо .

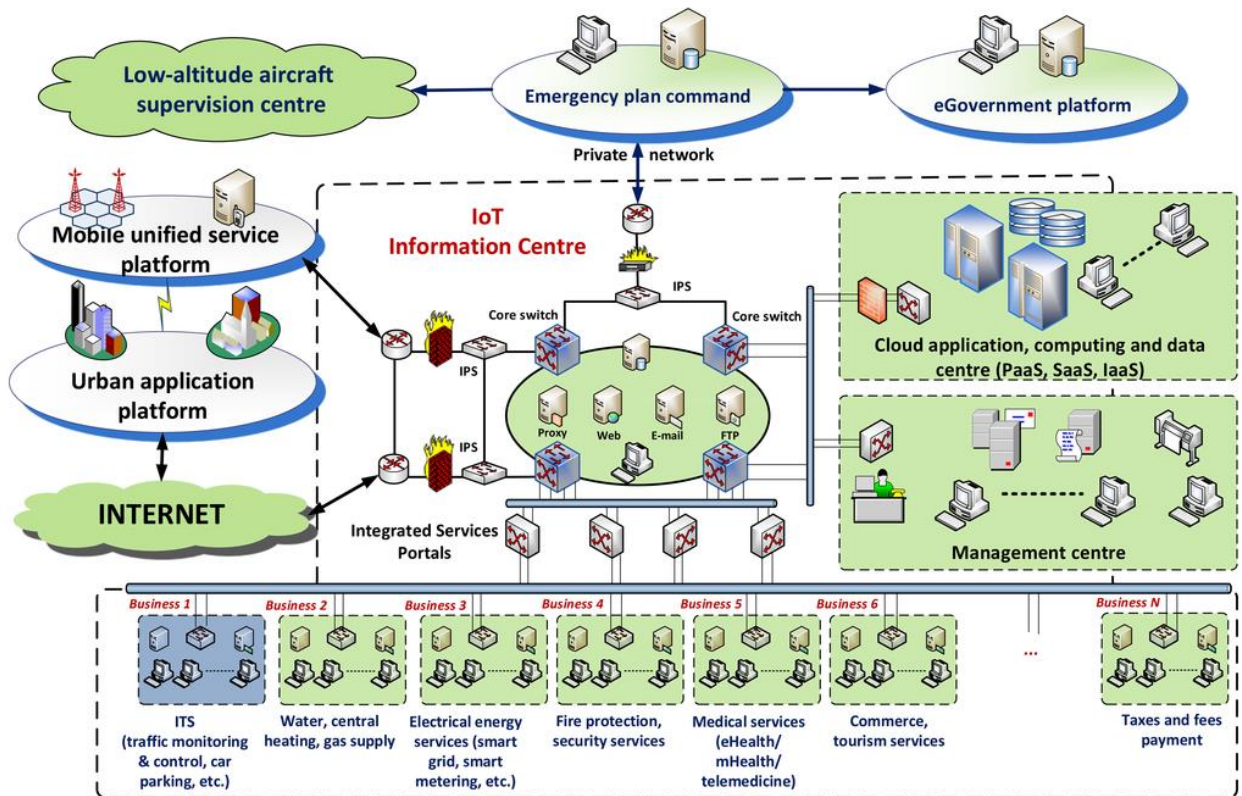


Рисунок 1.1 – Приклад платформи високого рівня розумного міста на основі ІоТ

На рисунку 1.1 зображено високорівневий вигляд централізованої платформи ІоТ, яка може служити загальною архітектурною основою для створення, функціонування, адміністрування та управління «розумним

містом». Цей загальний архітектурний дизайн верхнього рівня може уніфікувати розробку бізнес-додатків як ефективний та економічний процес.

Наприклад, якщо постачальник послуг ІПС хоче розгорнути послугу паркування автомобіля в спільній хмарній інфраструктурі розумного міста, йому потрібно буде зосередитися лише на операційній моделі, щоб усвідомити розумність паркування.

Тут пропонується інтелектуальна система паркування автомобілів для інтеграції в архітектуру ІоТ розумного міста, яка складається з трьох шарів - сенсорного, комунікаційного та прикладного рівня (рисунок 1.2).

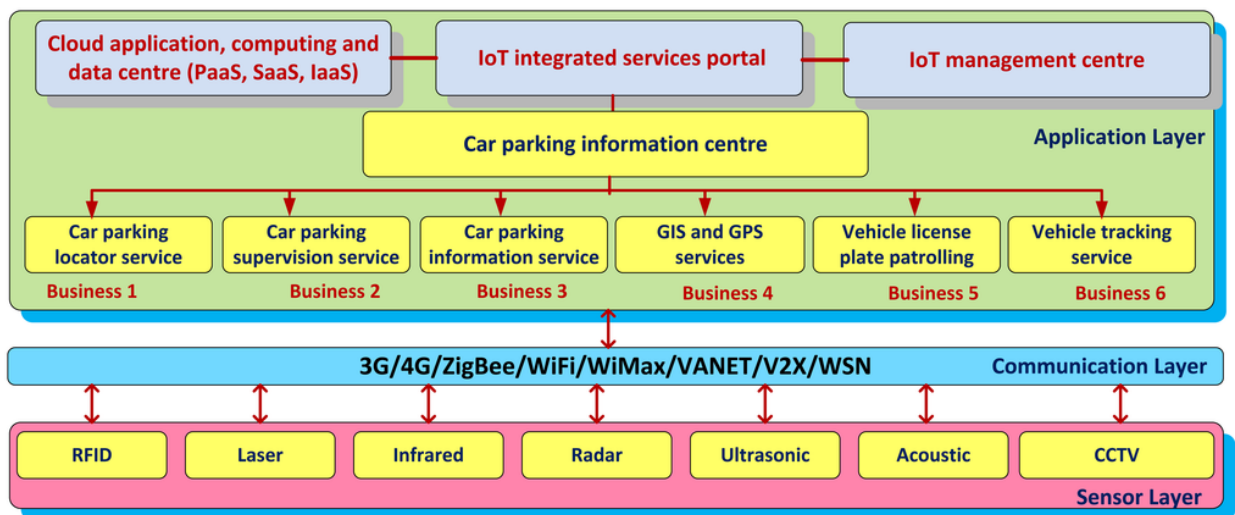


Рисунок 1.2 – Інтелектуальна система паркування автомобіля ІоТ для розумного міста.

На прикладному рівні інформаційний центр надає:

- хмарні послуги [18], тобто платформа як послуга (PaaS),
- програмне забезпечення як послуга (SaaS)
- інфраструктура як послуга (IaaS).

Центр управління ІоТ керує розумним містом через портал інтегрованих послуг ІоТ. Ряд бізнес-сервісів досліджують загальний інтерфейс до рівня зв'язку. Сюди входять пошук паркування, нагляд та інформаційні послуги, послуги ГІС/GPS, послуги відстеження автомобілів тощо.

На комунікаційному рівні різні бездротові технології забезпечують з'єднання між прикладним і сенсорним рівнем на основі комунікаційної парадигми ABC&S. 3-рівнева мережева архітектура на базі InfoStation [10] може бути інтегрована в цей рівень, щоб увімкнути функціональні можливості зв'язку «будь-коли» у розумних містах.

Різні сенсорні технології можуть бути використані на рівні датчиків для вбудованих рішень для паркування, таких як

- радіочастотна ідентифікація (RFID) для контролю доступу до паркування;
- лазерні датчики;
- пасивні інфрачервоні датчики;
- мікрохвильові радари;
- ультразвукові датчики;
- пасивні акустичні датчики;
- телебачення (CCTV) з опрацюванням відеозображення для визначення стану автостоянок;
- номерні знаки з встановленим модулем зв'язку 3G/4G для відстеження автомобілів.

Щоб система автомобільного паркування могла працювати як операційна платформа в розумному місті, слід розрізнити різні зони паркування для забезпечення «найкращих» автостоянок, виконуючи різні бізнес-ролі та програми.

Залежно від властивостей паркінги можна розділити на чотири основні категорії (рис.1.3):

- зона транспортного вузла;
- житловий/громадський район;
- земельна/вулична ділянка;
- торговий центр/готель/ресторан.

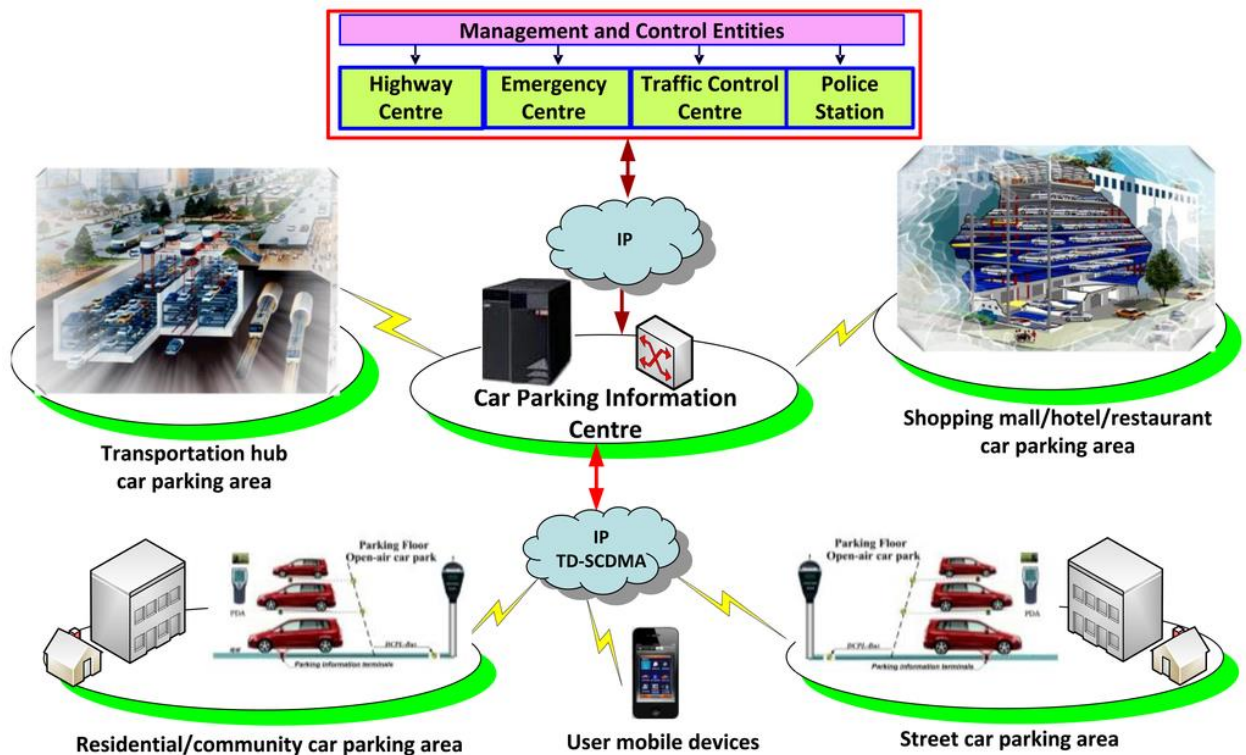


Рисунок 1.3 – Операційна платформа інтелектуальних автостоянок.

Відповідні органи управління та контролю, включаючи центр автомобільних доріг, центр екстреної допомоги, центр контролю дорожнього руху та поліцію, можуть отримати доступ до інформації, якою керує інформаційний центр паркування автомобілів з високими повноваженнями.

Давачі, встановлені в зоні паркування, періодично надсилають оновлену інформацію про наповненість автостоянок на паркомати, які передають ці дані до інформаційного центру.

Користувачі можуть взаємодіяти з системою, встановивши відповідний додаток для паркування на свої мобільні пристрої.

За сприяння персонального помічника кожен користувач може налаштувати особистий профіль, який буде використовуватися програмою для пошуку, розподілу, резервування та оплати «найкращого» паркінгу в кожному конкретному сценарії. Збережений у базі даних No-SQL на основі пам'яті, профіль користувача буде динамічно оновлюватися, щоб відобразити зміни в контексті та поведінці користувача, які аналізуються системою. Завдяки ефективним алгоритмам/правилам розподілу автостоянок система завжди

може надати мобільному користувачеві «найкращу» доступну автостоянку відповідно до парадигми ABC&S.

1.2 Зразок системи паркування для університетського містечка

У кожному великому університеті є кілька різних місць для паркування автомобілів, наприклад, для відвідувачів, студентів, співробітників тощо. Університетська автостоянка відноситься до категорії житлових/комунальних автостоянок. Кожен робочий день студенти та співробітники зазвичай витрачають багато часу на те, щоб знайти вільне місце для паркування. Це не тільки трудомісткий і енерговитратний процес, але також може спричинити автомобільні затори. Завдяки інтелектуальному хмарному сервісу паркування, який пропонується тут, можна досягти ефективного використання доступних паркувальних засобів у середовищі «розумного університету».

Одним із способів досягти цього є оснащення кожної автостоянки давачем, який зможе фіксувати присутність автомобіля на ньому. Інформаційна станція (InfoStation), що працює в зоні паркування, періодично збирає та об'єднує інформацію про присутність автомобіля з усіх давачів, розгорнутих у зоні, наприклад, за допомогою Wi-Fi, ZigBee або іншої бездротової технології малої дальності.

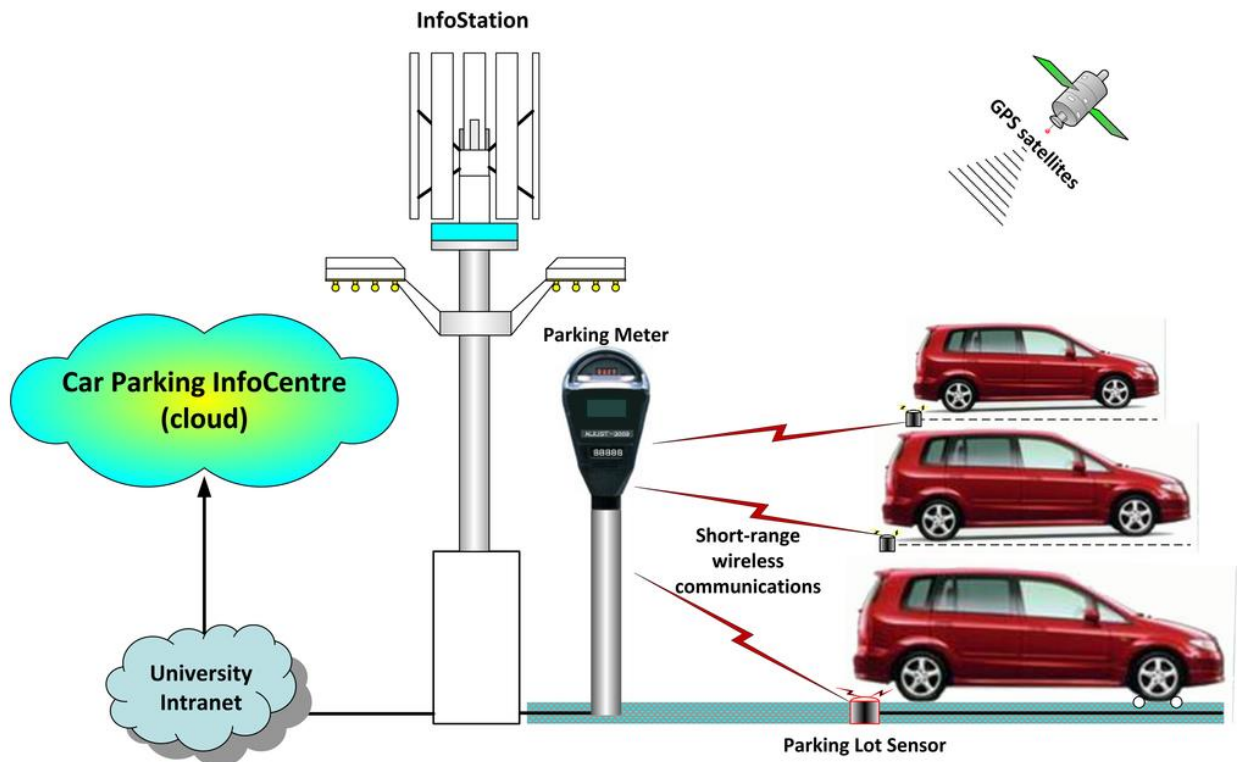


Рисунок 1.4 – Система паркування автомобілів університету на базі infoStation.

У разі платних автостоянок між InfoStation і давачами можуть працювати додаткові паркомати. Коли статус зайнятості автостоянки змінюється, інформація про це передається InfoStation до інформаційного центру паркування автомобілів (InfoCentre) у хмарі через інтранет університету (рис.1.4). Далі зосередимося лише на програмній реалізації цієї системи.

1.3 Архітектура системи паркування

З точки зору високого рівня розумного міста на основі IoT і багаторівневої інтелектуальної системи паркування автомобілів (рис.1.3 і 1.4), прикладний рівень служби паркування автомобілів, наприклад, в межах університетського містечка, може бути розгорнутий на трьох рівнях, як показано на рис.1.5.

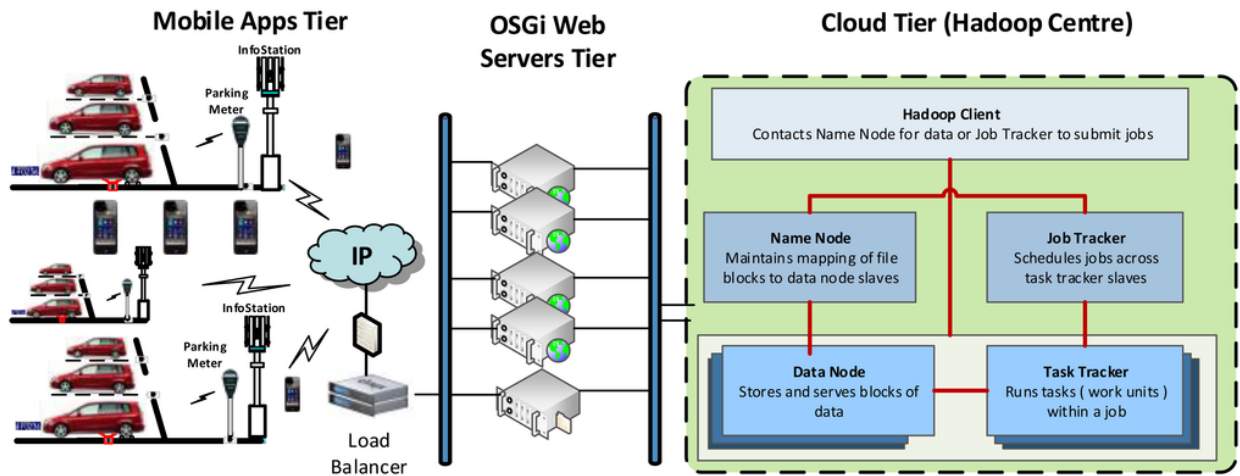


Рисунок 1.5 – Вид високого рівня прикладного рівня хмарної системи паркування автомобілів.

На хмарному рівні веб-додатки відбираються дані в центр Hadoop [20]; на рівні веб-серверів програми, розроблені як пакети, динамічно реєструються в рамках ініціативи Open Service Gateway (OSGi) [21]; а на рівні мобільних програм мобільні пристрої отримують доступ до веб-програм і надають своїм користувачам «найкращі» послуги паркування автомобілів.

1.3.1 Хмарний рівень

Хмара забезпечує зберігання даних та обчислювальні ресурси для служби паркування автомобілів. Зберігає «великі дані» про доступні автостоянки, площу паркування, місцезнаходження автомобілів, місцезнаходження та профілі користувачів тощо.

Найновіші дані зазвичай зберігаються в базі даних Hadoop Hbase [22] для підтримки запитів у реальному часі, тоді як дані історії серіалізуються в Hive [23] (сховище в Hadoop).

Для обчислень використовується ряд алгоритмів Map/Reduce [24], таких як:

- алгоритм рекомендацій для пропонування користувачам «найкращих» автостоянок,

- алгоритм рекомендацій на основі пропозицій друзів щодо паркування автомобілів,
- алгоритм оновлення профілю на основі про історію паркування користувачів тощо.

Для побудови ефективної та масштабованої системи використовується механізм правил Drools [25] для швидкого та надійного прийняття рішень на основі фактів.

1.3.2 Рівень веб-серверів OSGi

Цей рівень діє як міст між рівнем мобільних додатків і хмарним рівнем. Враховуючи велику кількість веб-програм/сервісів, що працюють на цьому рівні, розгортання нової/оновленої програми має бути можливим без зупинки/перезапуску відповідного веб-контейнера/сервера.

OSGi забезпечує середовище для модуляції веб-додатків на пакети, які можуть динамічно реєструватися в контексті виконання пакета. Щоб надати користувачам високопродуктивні послуги паркування автомобілів на вимогу, базу даних NoSQL на основі ключів - Redis [26] - використовується на цьому рівні для забезпечення масштабованих і розподілених черг завдань. Щоб оптимізувати використання веб-ресурсів, балансувальник навантаження розподіляє запити користувача по кластеру веб-серверів. Розподілена система збирає дані журналу веб-серверів і відправляє їх у хмару.

1.3.3 Рівень мобільних додатків

Дослідження Allied Business Intelligence (ABI) [27] описує, що операційна система Android [28] відіграла домінуючу роль на ринку смартфонів у 2013 році (з часткою 81%). Таким чином, для мобільних пристроїв Android розроблено першу версію мобільного додатку для паркування автомобілів.

Користувач наближається до кампусу університету, програма надсилає автоматичний запит (від імені користувача) на веб-сервер OSGi для паркування автомобілів із запитом на автостоянку.

Сервер знаходить «найкращу» доступну автостоянку для цього конкретного користувача на основі його/її уподобань, зазначених у профілі користувача, і (за бажанням) резервує її. Після цього користувачеві надсилаються маршрути проїзду разом із детальною картою, наприклад, за допомогою програми Google Map із Android API.

На рис.1.6 зображені основні компоненти системи.

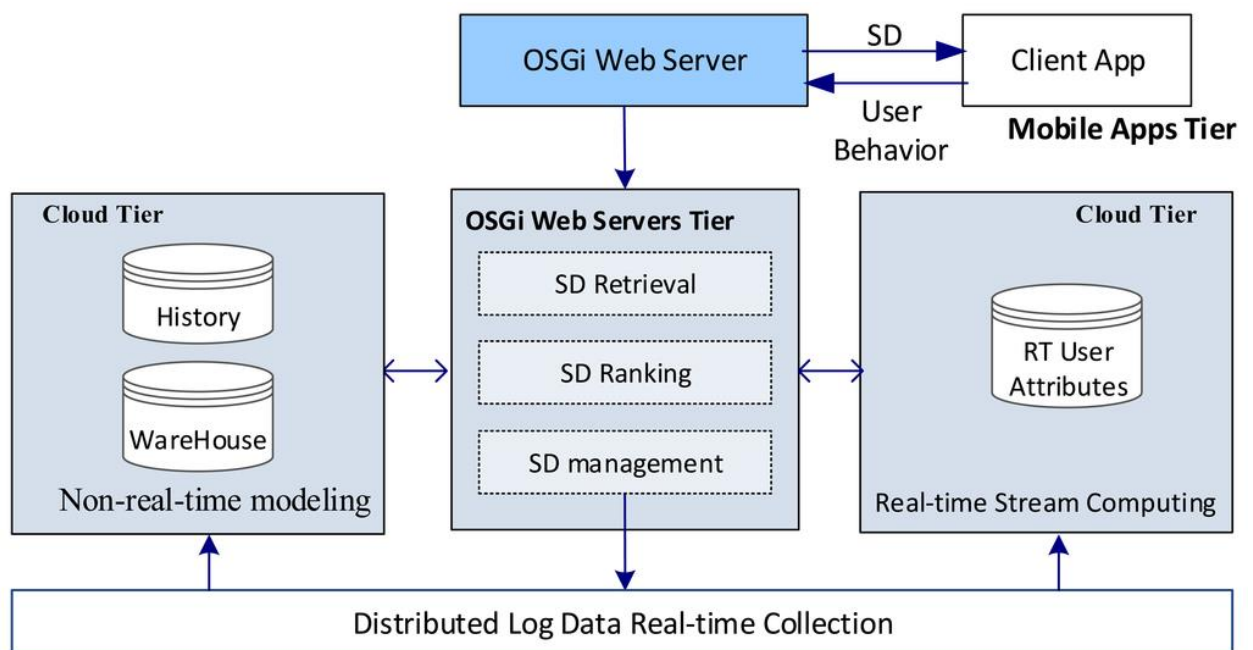


Рисунок 1.6 – Основні компоненти хмарної системи паркування автомобілів.

Хмарний рівень включає в себе частину поточкових обчислень у режимі реального часу для оновлення поведінки користувачів у режимі реального часу, стягнення плати за паркування тощо; і частина моделювання не в реальному часі для інтелекту даних і управління сховищем.

Частина збору даних розподіленого журналу діє як високошвидкісний канал даних у системі.

Рівень веб-серверів OSGi в центрі діє як міст між двома іншими рівнями. Можуть бути надані різні послуги паркування; вони описані їхніми описами послуг (SD) [17,29].

2 ОГЛЯД РЕАЛІЗАЦІЙ ІСП

2.1 Хмарний рівень

Hadoop — це фреймворк з відкритим вихідним кодом для розподіленої обробки «великих даних» на кількох комп'ютерах із використанням моделей програмування Map/Reduce. Це високонадійний і масштабований для паралельної обробки наборів «великих даних».

У службі автостоянки деякі функції знаходяться в автономному режимі та залежать від алгоритмів Map/Reduce, наприклад, алгоритми оновлення профілю, тоді як інші є функціями навігації в реальному часі.

Розподілений набір даних дозволяє ефективно обробляти дані, завдяки чому рекомендації щодо «найкращих» доступних автостоянок можуть бути доставлені різним користувачам з різних серверів. Щоб задовольнити цю вимогу, хмарне проміжне програмне забезпечення повинно містити три кластери:

- Kafka (kafka.apache.org) [30],
- Storm (storm-project.net) [31],
- розподілена файлова система Hadoop (HDFS, hadoop.apache.org) [32] - як показано на рис.2.1.

Kafka — це високопродуктивна платформа розподіленого обміну повідомленнями, яка використовується як кластер балансування навантаження для паралельного завантаження даних у Hadoop.

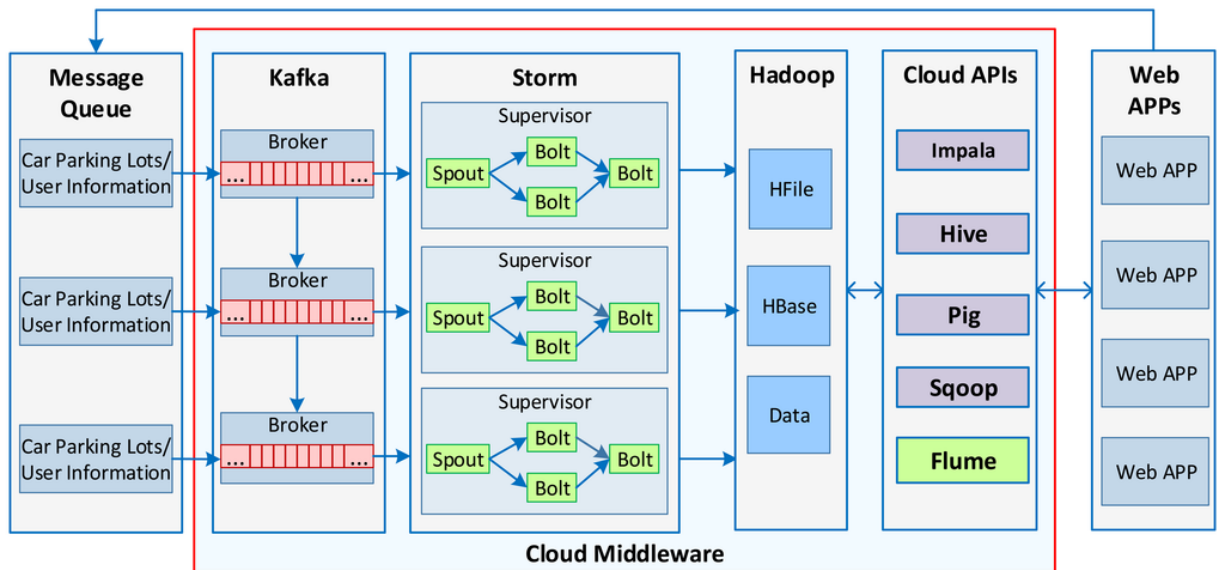


Рисунок 2.1 – Хмарні рішення для послуги паркування.

За допомогою цих хмарних рішень інформація про автостоянки - разом з іншою додатковою інформацією про користувачів - збирається в режимі реального часу, а сервіс доступний через хмарні API, наприклад, Hive, Pig, Impala тощо.

На рис.2.2 показано діаграму послідовності веб-програм, які розміщують набори даних у хмарі. У модулі Kafka спочатку ініціалізується `ProducerConfig` з такими параметрами: список брокерів для визначення лідера кожної теми, серіалізатор для кодування повідомлення для передачі та роздільник для визначення теми повідомлення.

Нарешті, генерується `kafka.javaapi.producer`, який знаходить головного брокера для певної теми та розділу, отримує відповідні повідомлення та метадані та надсилає їх до Storm

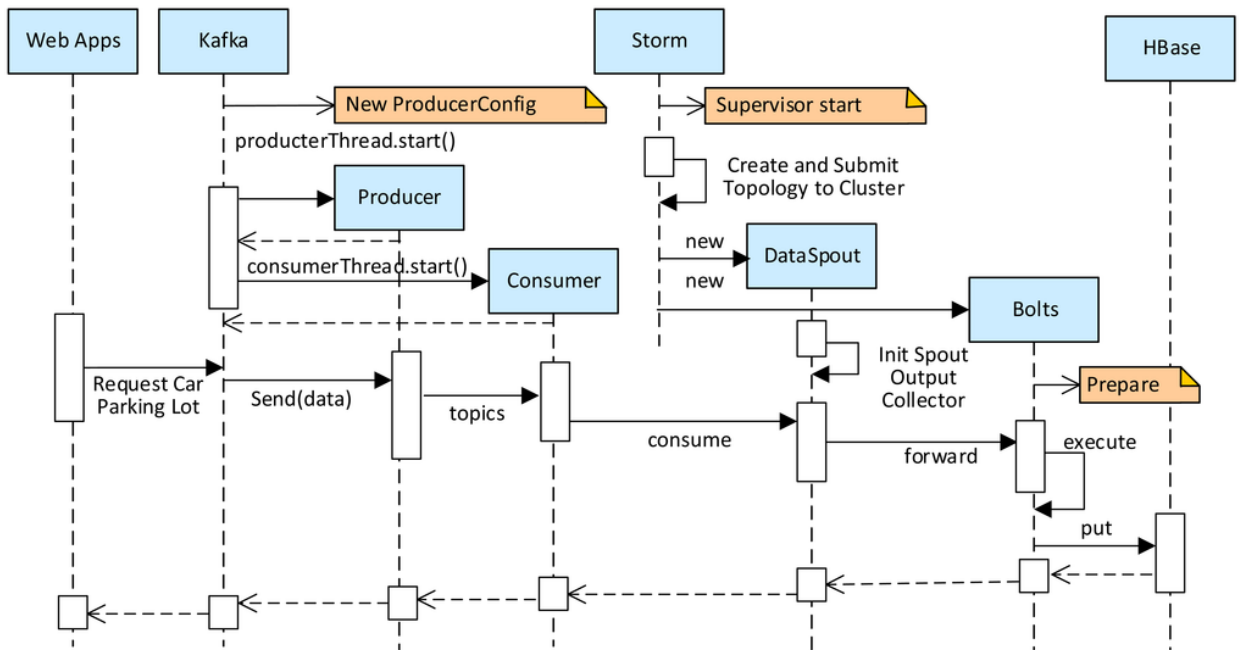


Рисунок 2.2 – Схема послідовності веб-додатків, які розміщують набори даних у хмарі.

На рис.2.3 показана UML-діаграма основних рекомендаційних класів Drools, що використовуються в частині механізму правил. Клас `RecommendImpl` реалізує інтерфейс `Recommend` і отримує рекомендовані SD паркування автомобіля через інтерфейс сортування.

За допомогою цього шаблону проектування інтерфейсу, коли до системи застосовується нове правило, систему не потрібно перекомпілювати, таким чином забезпечується слабке з'єднання.

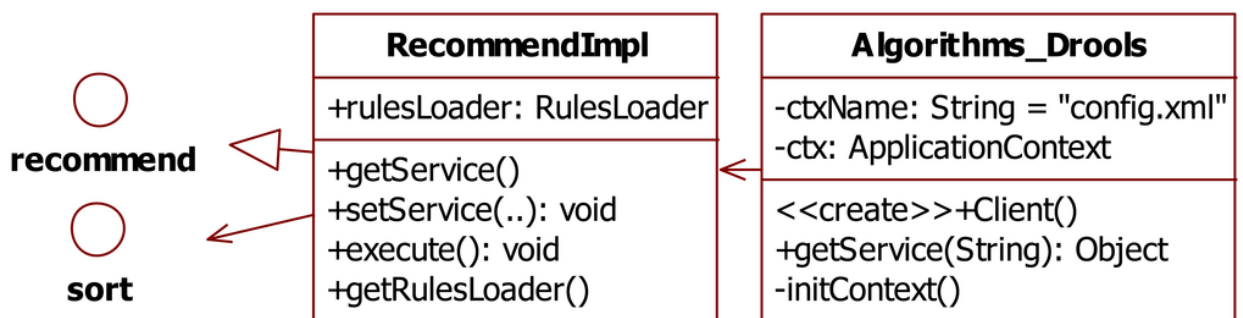


Рисунок 2.3 – UML-діаграма рекомендаційних класів на основі drools.

Клас `rulesLoader`, визначений мовою правил `Drools`, використовується для завантаження файлу «.drl» та для опису кількох правил, запитів і функцій. Клас `Algorithms_Drools` визначає контекст програми, створює дані користувача (клієнта) паркування та надає користувачеві функції `getService`.

2.2 Рівень веб-серверів OSGi

Через велику кількість веб-програм на базі OSGi, які працюють на цьому рівні, пакети веб-додатків повинні мати можливість динамічно реєструватися в контейнері або видалятися з контейнера. Нижче наведено етапи впровадження пакетів OSGi для паркування автомобілів:

- визначити інтерфейс пакета `QueryService` для паркування та додати його до експортного пакета в `MANIFEST.MF`.
- реалізувати інтерфейс `QueryService` і опис автостоянки. (Відповідні операції включають створення об'єкта `ServiceRegistration` в активаторі OSGi, реєстрацію `QueryService` у функції запуску та видалення `QueryService` з функції `stop`.)
- Реалізувати пакет відповідей HTTP `QueryService` для паркування. Об'єкти веб-сервлетів реєструються як `HttpService` в активаторі, а `org.osgi.framework.BundleContext` пов'язаний із сервлетом Java для надання функції запиту автостоянки (рис.2.4).

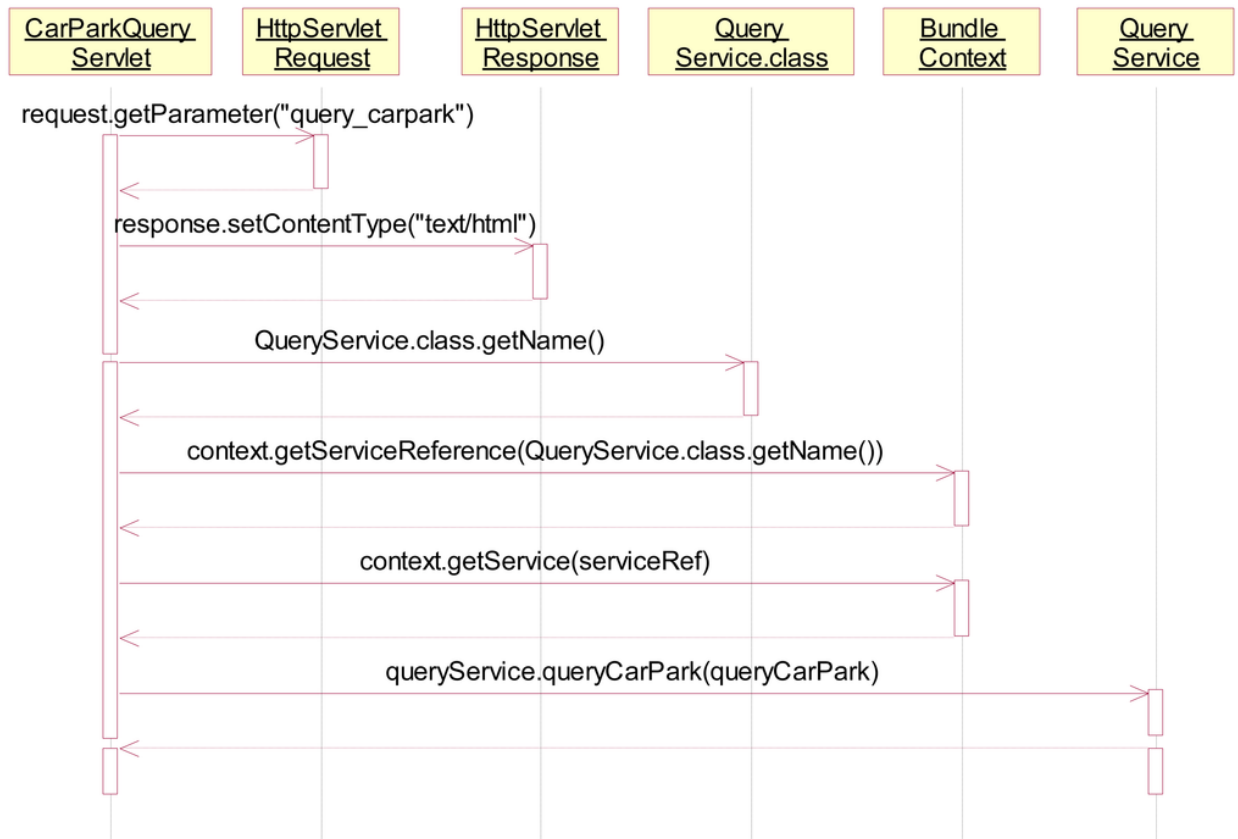


Рисунок 2.4 – Функція запиту автостоянки з пакетами OSGi.

Для збору веб-журналів Flume на основі Hadoop використовується для переміщення великих обсягів даних журналів у хмару.

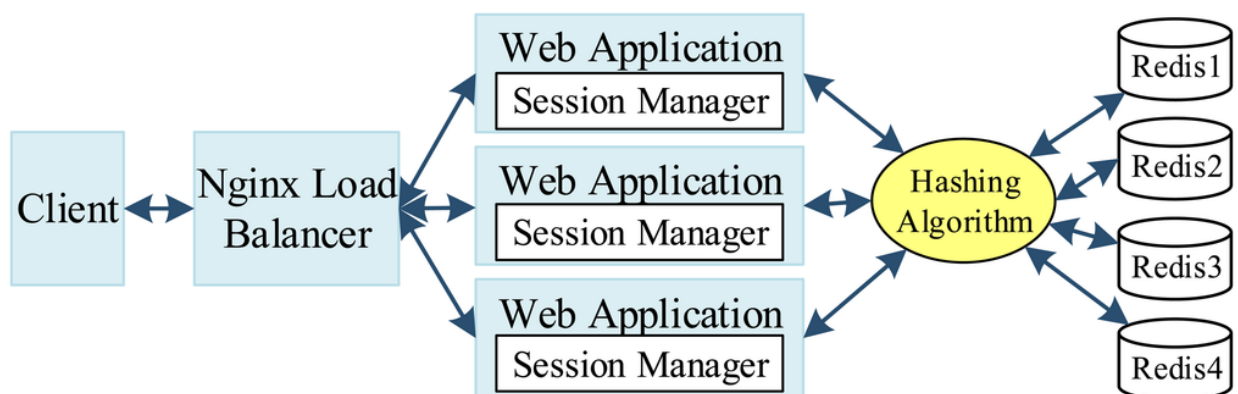


Рисунок 2.5 – Балансувальник навантаження, менеджер сеансів і розподілений redis на рівні веб-серверів OSGi.

Для підвищення продуктивності веб-сервера передбачено ряд рішень, зокрема балансувальник навантаження Nginx [35], що використовується для

прискорення відповідей користувачів, послідовний алгоритм хешування, що працює в розподіленій базі даних пам'яті Redis, і менеджери сеансів, які синхронізують сеанси різних веб-додатки (рис.2.5).

Zookeeper використовується для підтримки статусу менеджерів сеансів для забезпечення синхронізації даних.

Алгоритм хешування розроблений на основі функції Java ConcurrentHashMap для розподілу операцій читання/запису на кластері Redis.

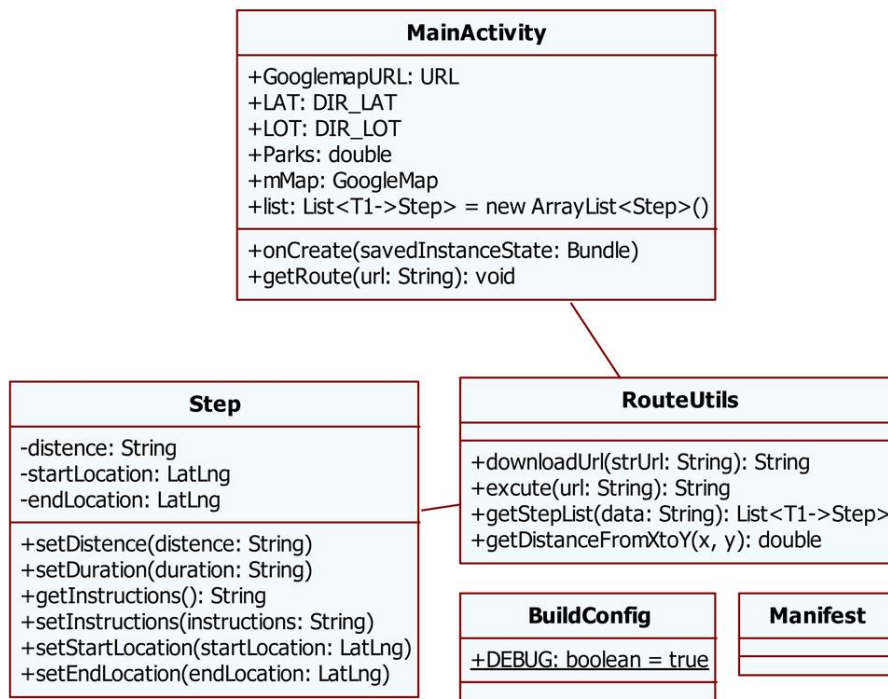
2.3 Рівень мобільних додатків

Розроблений мобільний додаток для паркування автомобілів для Android — це програма для одного перегляду, яка в основному включає три класи Java:

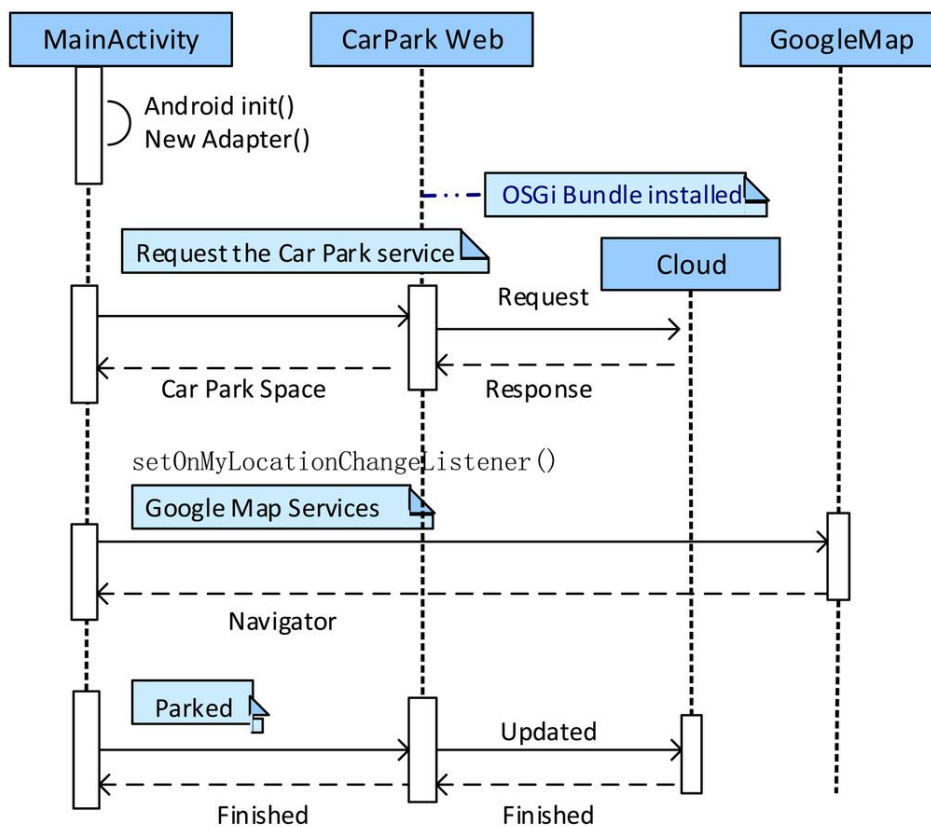
- Android Activity, названий MainActivity, який забезпечує інтерфейс користувача;
- простий старий об'єкт Java (POJO) з іменем step, який визначає властивості маршруту за допомогою методів get і set;
- диспетчер контролера, названий RouteUtils, який отримує дані з веб-рівня.

Коли об'єкт RouteUtils активний, він співпрацює з API карт Google і надсилає запит послуги паркування автомобіля на веб-сервер OSGi.

Після отримання інформації про паркування автомобіля з рівня хмари, інформація повертається на об'єкт і запускається навігація по карті Google.



(a)



(b)

Рисунок 2.6 – Діаграма класів мобільного додатка для паркування автомобілів для Android та діаграма послідовності основної діяльності паркування автомобіля.

На рис.2.6 проілюстровано діаграму класів програми для паркування автомобіля для Android та зображена схема послідовності функції MainActivity.

2.4 Результати

Розробка системи повинна здійснюватися за методологією процесу персонального програмного забезпечення (PSP). Для PSP використовуються методи розробки на основі тестів і функцій. На рис.2.7 показано розгортання системи паркування автомобіля, яка включає наступні компоненти:

- кеш Varnish, що діє як зворотний проксі-сервер для прискорення HTTP-відповідей;
- веб-сервер на основі ряду серверів Nginx для балансування навантаження;
- споживач і виробник послуг, що діють як розподілена структура сервісу для управління послугами (функції обміну повідомленнями синхронізації над асинхронністю та запитом-відповіддю, можливості балансування навантаження/відмови /кластеризації з механізмом віддаленого виклику процедур (RPC) тощо);
- доглядач реєстру послуг, що використовується для реєстрації послуг паркування та публікації/передплати сервісних подій;
- кластер Redis, що складається з ряду серверів Redis, що забезпечують розподілені операції для інших рівнів;
- індекс Sphinx, що забезпечує функціональність повнотекстового пошуку для користувачів/веб-додатків.

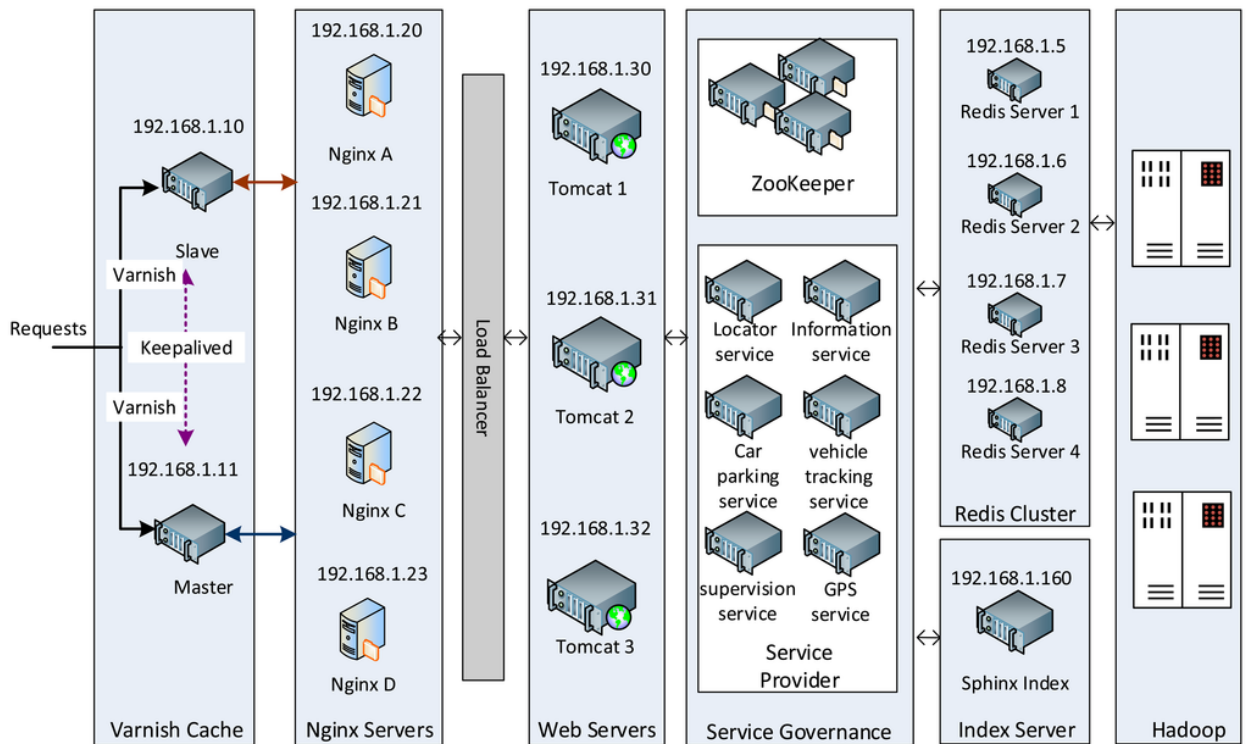


Рисунок 2.7 – Вид високого рівня розгортання системи паркування автомобіля.

2.4.1 Хмарний рівень

На хмарному рівні використовуються чотири сервери (Master, Slave1, Slave2 і Slave3) з встановленим Hadoop 1.2. Кожен сервер був встановлений на ПК Intel XEON з процесором E3-1220L та 8 ГБ оперативної пам'яті. Розмір повідомлення був встановлений на 200 байт. Майстер служить вузлом імені для Hadoop, а інші вузли даних.

Три сервери Kafka працюють на кожному підпорядкованому вузлі і один супервізор Storm працює на кожному сервері.

Storm nimbus і інтерфейс користувача ініціалізуються в Slave1.

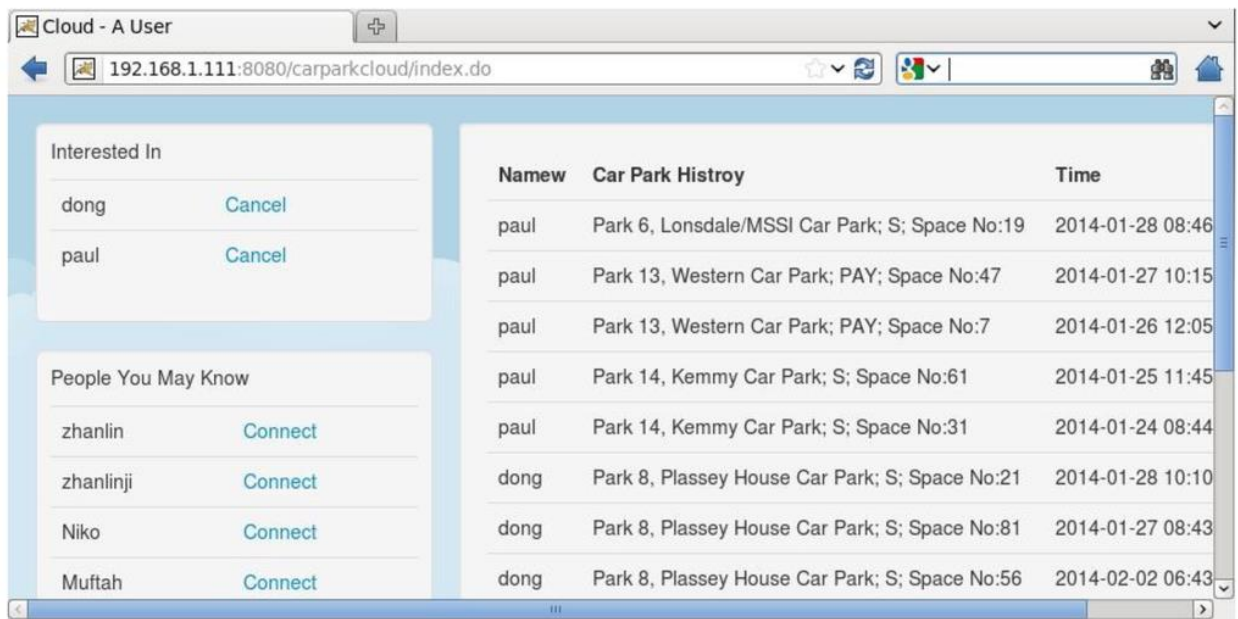
Визначена топологія подається у Slave1 за допомогою Storm jar-команда.

Автостоянки та користувачі описуються буфером протоколу Google. Для демонстраційних цілей 2000 автостоянок і 2000 користувачів генеруються випадковим чином і зберігаються в базі даних HBase.

Додатки хмарного рівня, включаючи керування автостоянками, планування алгоритмів Map/Reduce, обслуговування особистих профілів тощо, розміщені в Інфоцентрі.

З міркувань безпеки кінцеві користувачі не можуть отримати доступ до програм хмарного рівня; до них може отримати доступ лише адміністратор служби паркування або окремі програми рівня веб-серверів.

На рис.2.8 показано фрагмент історії використання паркування.



The screenshot shows a web browser window with the address bar displaying '192.168.1.111:8080/carparkcloud/index.do'. The page content is divided into two main sections. On the left, there is a sidebar with two sections: 'Interested In' and 'People You May Know'. The 'Interested In' section lists two users, 'dong' and 'paul', each with a 'Cancel' button. The 'People You May Know' section lists five users: 'zhanlin', 'zhanlinji', 'Niko', and 'Muftah', each with a 'Connect' button. On the right, there is a table with three columns: 'Name', 'Car Park History', and 'Time'. The table contains ten rows of data, each representing a user's parking history entry.

| Name | Car Park History | Time |
|------|--|------------------|
| paul | Park 6, Lonsdale/MSSI Car Park; S; Space No:19 | 2014-01-28 08:46 |
| paul | Park 13, Western Car Park; PAY; Space No:47 | 2014-01-27 10:15 |
| paul | Park 13, Western Car Park; PAY; Space No:7 | 2014-01-26 12:05 |
| paul | Park 14, Kemmy Car Park; S; Space No:61 | 2014-01-25 11:45 |
| paul | Park 14, Kemmy Car Park; S; Space No:31 | 2014-01-24 08:44 |
| dong | Park 8, Plassey House Car Park; S; Space No:21 | 2014-01-28 10:10 |
| dong | Park 8, Plassey House Car Park; S; Space No:81 | 2014-01-27 08:43 |
| dong | Park 8, Plassey House Car Park; S; Space No:56 | 2014-02-02 06:43 |

Рисунок 2.8 – Зразок GUI історії використання паркування на рівні хмари.

Було проведено п'ять тестів з різною кількістю виробників (клієнтів) Kafka. Кожен виробник асинхронно надсилає певну кількість повідомлень у хмару.

Snappy був використаний як алгоритм стиснення. Розмір пакету був встановлений на 2000 повідомлень, а кількість розділів Kafka — на 9. Пропускна здатність системи, тобто середня кількість оброблених повідомлень за секунду, досягнута в кожному тесті, зображена на рис.2.9.

Результати показують, що оптимальною кількістю виробників є 3, оскільки перевищення цієї кількості не призводить до значного збільшення

пропускної здатності системи. Для цього оптимального випадку досягнута пропускна здатність становить близько 400 000 повідомлень в секунду.

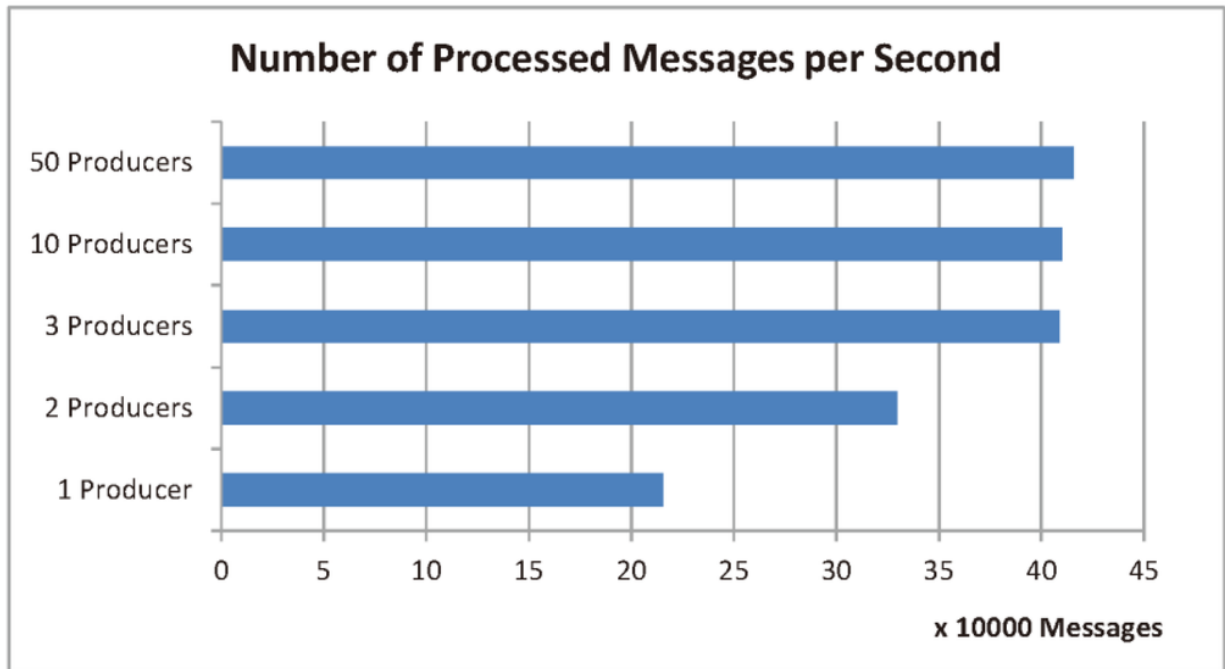


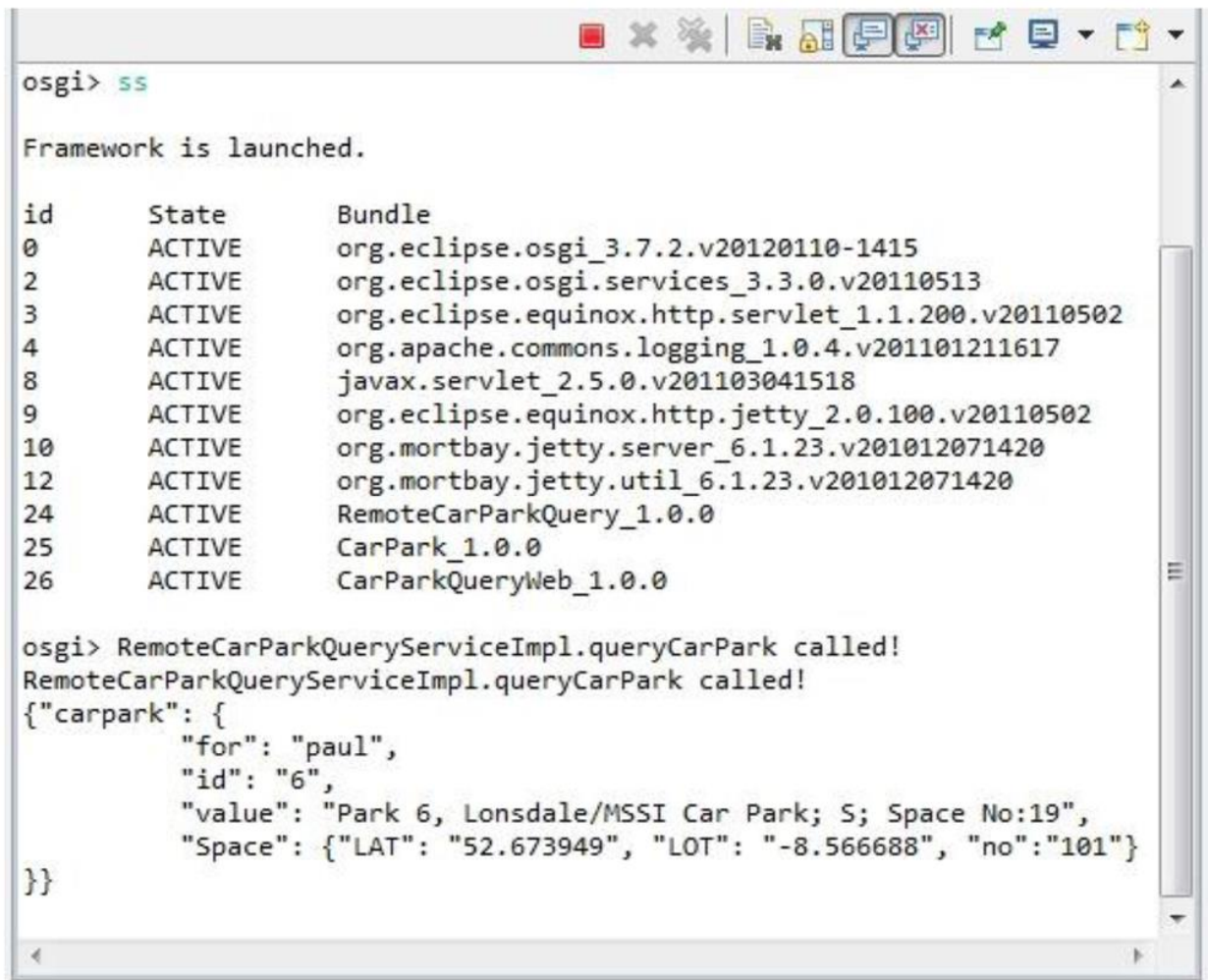
Рисунок 2.9 – Кількість оброблених повідомлень за секунду для різної кількості виробників (клієнтів) kafka на рівні хмари.

2.4.2 Рівень веб-серверів OSGi

На цьому рівні три веб-сервери (рис.2.10) налаштовані для роботи з CentOS 6.5 і контейнером Equinox OSGi. Один із серверів налаштований за допомогою балансувальника навантаження Nginx. Пакети розгорнуті на цих трьох веб-серверах.

На рис.2.10 показано пакети RemoteCarParkQuery, CarPark і CarParkQueryWeb у середовищі OSGi. Залежні пакети, наприклад, служба OSGi, Jetty тощо, також активні в середовищі.

При запиті CarPark отриманий запит, пакет пересилає запит на рівень хмари. Коли відповідь надходить, вона форматується за допомогою JavaScript Object Notation (JSON) і надсилається назад до клієнтської програми Android.



```
osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.7.2.v20120110-1415
2       ACTIVE    org.eclipse.osgi.services_3.3.0.v20110513
3       ACTIVE    org.eclipse.equinox.http.servlet_1.1.200.v20110502
4       ACTIVE    org.apache.commons.logging_1.0.4.v201101211617
8       ACTIVE    javax.servlet_2.5.0.v201103041518
9       ACTIVE    org.eclipse.equinox.http.jetty_2.0.100.v20110502
10      ACTIVE    org.mortbay.jetty.server_6.1.23.v201012071420
12      ACTIVE    org.mortbay.jetty.util_6.1.23.v201012071420
24      ACTIVE    RemoteCarParkQuery_1.0.0
25      ACTIVE    CarPark_1.0.0
26      ACTIVE    CarParkQueryWeb_1.0.0

osgi> RemoteCarParkQueryServiceImpl.queryCarPark called!
RemoteCarParkQueryServiceImpl.queryCarPark called!
{"carpark": {
  "for": "paul",
  "id": "6",
  "value": "Park 6, Lonsdale/MSSI Car Park; S; Space No:19",
  "Space": {"LAT": "52.673949", "LOT": "-8.566688", "no": "101"}
}}
```

Рисунок 2.10 – Вихід пакету OSGi служби запиту на паркування автомобіля.

Кластер Redis розгорнутий на чотирьох комп'ютерах з IP-адресами 192.168.1.5~8 (рис.2.10).

П'ять серверів Redis працюють на кожному комп'ютері через порти 5001~5005 відповідно. Кожен з цих портів призначений для одного конкретного типу служби паркування автомобілів, тобто служби локації, інформаційної служби, служби GPS, служби відстеження транспортних засобів та служби спостереження. Коли запит доставляється в кластер Redis, він буде відправлено на сервер Redis за допомогою послідовного алгоритму хешування.

2.4.3 Рівень мобільних додатків

На цьому рівні, коли користувач на автомобілі в'їжджає до університетського містечка через одну з його воріт, мобільний додаток для паркування автомобіля, встановлений на мобільному пристрої користувача, надсилатиме автоматичний HTTP-запит через точку бездротового доступу воріт до веб-сервера, і буде повернуто відповідь JSON, що містить інформацію про «найкращу» доступну автостоянку. Для мобільного пристрою з підтримкою GPS RouteUtils генерує маршрути (кроки), за якими слідуватиме водій, і відображає їх на карті Google. Для мобільних пристроїв, які не підтримують GPS, програма служби коротких повідомлень (SMS) відображає користувачеві інформацію про автостоянку.

Результати підтверджують здатність системи паркування обслуговувати велику кількість клієнтів, зберігаючи середній час відгуку на рівні мілісекунд. Незначні коливання в часі відповіді системи в основному зумовлені затримками пакетів, викликаними мережею.

3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

3.1 Санітарно-гігієнічні вимоги до умов праці на комп'ютеризованих робочих місцях

Гігієна – це галузь медицини, яка вивчає вплив умов життя на здоров'я людини і розробляє заходи профілактики захворювань, забезпечення оптимальних умов існування, збереження здоров'я та продовження життя.

Гігієна праці – це підгалузь загальної гігієни, яка вивчає вплив виробничого середовища на функціонування організму людини і його окремих систем. Організм людини формувався в умовах реального природного середовища. Основними чинниками цього середовища є мікроклімат, склад повітря, електромагнітний, радіаційний і акустичний фон, світловий клімат тощо.

Санітарія – це сукупність практичних заходів, спрямованих на оздоровлення середовища, що оточує людину.

Виробнича санітарія – це галузь санітарії, спрямована на впровадження комплексу санітарно-оздоровчих заходів щодо створення здорових і безпечних умов праці.

Згідно з ДСТУ 2293-99, виробнича санітарія – це система організаційних, гігієнічних і санітарно-технічних заходів та засобів запобігання впливу на працівників шкідливих виробничих факторів. Сфера дії виробничої санітарії – запобігання професійній небезпеці (шкідливості) яка може призвести до професійних або професійно зумовлених захворювань, у тому числі і смертельних, під час дії в процесі роботи таких факторів, як випромінювання електромагнітних полів, іонізуючого випромінювання, шумів, вібрацій, хімічних речовин, зниженої чи підвищеної температури тощо.

Згідно з діючим законодавством забезпечення санітарного благополуччя досягається такими основними заходами:

- пред'явленням відповідних гігієнічних вимог до проектування, забудови та експлуатації будівель, споруд, приміщень, територій, розробкою та впровадженням нових технологій і обладнання;
- державною санітарно-гігієнічною експертизою проєктів, технологічних регламентів, інвестиційних програм і діючих об'єктів;
- включенням вимог безпеки щодо здоров'я та життя людини в державні стандарти та нормативно-технічну документацію всіх сфер діяльності суспільства;
- гігієнічною регламентацією та контролем (моніторингом) усіх шкідливих і небезпечних факторів навколишнього та виробничого середовища;
- ліцензуванням видів діяльності, пов'язаних з потенційною небезпекою для здоров'я людей;
- контролем та аналізом стану здоров'я населення та працівників;
- профілактичними санітарно - лікувальними заходами;
- запровадженням санкцій до відповідальних осіб за порушення санітарно-гігієнічних вимог.

На сучасному етапі розвитку гігієни праці як науки гігієністи під час вирішення питань охорони здоров'я працівників дотримуються так званого порогового принципу: до якогось критичного відхилення певного фактору виробничого середовища від природної фізіологічної норми для людини відхилення не спричиняє небажаних змін в організмі працівника і не матиме генетичних наслідків.

Згідно із цим гігієністи за окремими факторами виробничого середовища встановлюють науково обґрунтовані граничні нормативи (гранично допустимі концентрації, рівні тощо), які в установленому порядку затверджують відповідні центральні органи державного управління. На основі цих нормативів здійснюється аудит гігієнічних умов праці на їх відповідність чинній нормативно-правовій базі.

Для комплексної оцінки умов праці – з урахуванням фізіологічних і гігієнічних умов – Київський інститут медицини праці розробив Гігієнічну класифікацію умов праці, затверджену наказом Міністра охорони здоров'я України засновану на принципі диференціації умов праці залежно від фактично діючих рівнів факторів виробничого середовища і трудового процесу порівняно із санітарними нормами, правилами, гігієнічними нормативами, а також можливим впливом їх на стан здоров'я працівників. Відповідно до Гігієнічної класифікації клас умов праці визначають тим фактором виробничого середовища, напруженості або тяжкості праці, який має найбільше відхилення від нормативних вимог. Реальні умови праці мають виключати передумови для виникнення травм і професійних захворювань.

Фактори, що зумовлюють умови праці, поділяють на чотири групи:

- перша група – Санітарно-гігієнічні фактори. Вона включає показники, що характеризують виробниче середовище робочої зони. Вони залежать від особливостей виробничого обладнання і технологічних процесів, можуть бути оцінені кількісно і нормовані.
- друга група – становлять психофізіологічні елементи, зумовлені самим процесом праці. З цієї групи лише частина факторів може бути оцінена кількісно.
- третя група – належать естетичні фактори, що характеризують сприйняття працівником навколишньої обстановки та її елементів; кількісно вони оцінені бути не можуть.
- четверта група – містить соціально-психологічні фактори, що характеризують психологічний клімат у трудовому колективі; кількісній оцінці також не підлягають.

Отже, несприятливий вплив на людину санітарно-гігієнічних факторів спричинює відволікання внутрішніх ресурсів працівника від основного трудового процесу, несприятливо впливає на психофізіологічний стан людини, її працездатність і, як наслідок, відбивається на техніко-економічних показниках підприємства.

3.2 Психофізіологічне розвантаження для працівників

При організації праці, пов'язаної з використанням ЕОМ і ПЕОМ, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності передбачаються внутрішньозмінні регламентовані перерви для відпочинку.

Внутрішньозмінні режими праці і відпочинку містять додаткові нетривалі перерви в періоди, що передують появі об'єктивних і суб'єктивних ознак стомлення і зниження працездатності.

Впродовж робочої зміни мають передбачатися:

- перерви для відпочинку і вживання їжі (обідні перерви);
- перерви для відпочинку і особистих потреб (згідно з трудовими нормами);
- додаткові перерви, що вводяться для окремих професій з урахуванням особливостей трудової діяльності.

За характером трудової діяльності розрізняють три професійні групи, згідно з діючим класифікатором професій (ДК-003-95 і Зміна N1 до ДК-003-95):1):

- розробники програм (інженери-програмісти) виконують роботу з ЕОМ та документацією при необхідності інтенсивного обміну інформацією з ЕОМ і високою частотою прийняття рішень.
- оператори електронно-обчислювальних машин виконують роботу, пов'язану з обліком інформації, одержаної з ВДТ за попереднім запитом, або тієї, що надходить з нього, супроводжується перервами різної тривалості, пов'язана з виконанням іншої роботи і характеризується напруженням зору, невеликими фізичними зусиллями, нервовим напруженням середнього ступеня та виконується у вільному темпі;
- оператор комп'ютерного набору виконує одноманітні за характером роботи з документацією та клавіатурою і нечастими нетривалими

переключеннями погляду на екран дисплея, з введенням даних з високою швидкістю.

Правилами встановлюються такі внутрішньо змінні режими праці та відпочинку при роботі з ЕОМ при 8-годинній денній робочій зміні в залежності від характеру праці:

- для розробників програм із застосуванням ЕОМ слід призначати регламентовану перерву для відпочинку тривалістю 15 хвилин через кожну годину роботи за ВДТ;
- для операторів із застосуванням ЕОМ слід призначати регламентовані перерви для відпочинку тривалістю 15 хвилин через кожні дві години;
- для операторів комп'ютерного набору слід призначати регламентовані перерви для відпочинку тривалістю 10 хвилин після кожної години роботи за ВДТ.

Для зниження нервово-емоційного напруження, стомлення зорового аналізатора, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно деякі перерви використовувати для виконання комплексу вправ, які наведені у Державних санітарних правилах і нормах роботи з ПК електроннообчислювальних машин ДСанПІН 3.3.2.007-98.

При проведенні сеансів психофізіологічного розвантаження рекомендується використовувати деякі елементи методу аутогенного тренування, який ґрунтується на свідомому застосуванні комплексу взаємопов'язаних прийомів психічної саморегуляції й виконанні нескладних фізичних вправ із словесним самонавіюванням. Головна увага при цьому приділяється набуванню й закріпленню навичок м'язового розслаблення (релаксації). У рекомендованому сеансі, який має проводитися в кімнаті психофізіологічного розвантаження з відповідним інтер'єром та кольоровим оформленням, виділяються три періоди, що відповідають фазам відновлювального процесу.

Перший період - абстрагування працівників від виробничої обстановки - відповідає фазі залишкового збудження.

Другий період - заспокоєння - відповідає фазі відновлювального гальмування.

Третій період - активізація - відповідає фазі підвищеної збудженості.

Сеанси психологічного розвантаження можуть проводитись за єдиною програмою через індивідуальні навушники і складатись із двох періодів по 5 хвилин кожний:

- повне розслаблення;
- активізація працездатності.

У разі потреби, на фоні музичних програм можуть вимовлятися окремі фрази навіювання відпочинку, гарного самопочуття і, на заключному етапі, бадьорості.

Після сеансів психофізіологічного розвантаження у працівників зменшується відчуття втоми, з'являються бадьорість, гарний настрій. Загальний стан відчутно поліпшується.

ВИСНОВКИ

У кваліфікаційній роботі здійснено огляд хмарних інтелектуальних система паркування автомобілів з використанням IoT.

Важливим компонентом інтелектуальної системи для розумних міст є система паркування автомобіля, яка складається з трьох рівнів: сенсорного, комунікаційного та прикладного.

Здійснено огляд програмного забезпечення системи та відповідних операційних платформ.

Розглянуто системи, які використовуються для надання послуг паркування автомобіля для кампусу університету разом із допоміжними хмарними додатками, веб-додатками на базі OSGi та мобільними додатками Android.

Послуга надає користувачеві (водію) інформацію про «найкращу» доступну автостоянку, яка повертається в мобільний додаток користувача, дотримуючись парадигми «Завжди найкращий зв'язок і найкраще обслуговування» (ABC&S).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vermesan, O.; Friess, P.; Guillemin, P.; Gusmeroli, S.; Sundmaeker, H.; Bassi, A.; Jubert, I.S.; Mazura, M.; Harrison, M.; Eisenhauer, M. Internet of things strategic research roadmap. *Int. Things-Global Technol. Soc. Trends* 2011, 1, 9–52.
2. Rudin-Brown, C.M. ‘Intelligent’ in-vehicle intelligent transport systems: Limiting behavioural adaptation through adaptive design. *IET Intell. Transp. Syst.* 2010, 4, 252–261.
3. O'Droma, M.; Ganchev, I. The creation of a ubiquitous consumer wireless world through strategic ITU-T standardization. *IEEE Commun. Mag.* 2010, 48, 158–165.
4. O'Droma, M.; Ganchev, I. Toward a ubiquitous consumer wireless world. *IEEE Wirel. Commun.* 2007, 14, 52–63.
5. Bilodeau, V.P. Intelligent Parking Technology Adoption. Ph.D. Thesis, University of Southern Queensland, Queensland, Australia, 2010.
6. Choeychuen, K. Automatic parking lot mapping for available parking space detection. Proceedings of the 5th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 31 January–1 February 2013; pp. 117–121.
7. Li, T.S.; Ying-Chieh, Y.; Jyun-Da, W.; Ming-Ying, H.; Chih-Yang, C. Multifunctional intelligent autonomous parking controllers for carlike mobile robots. *IEEE Trans. Ind. Electron.* 2010, 57, 1687–1700.
8. Keat, C.T.M.; Pradalier, C.; Laugier, C. Vehicle detection and car park mapping using laser scanner. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 2054–2060.
9. Ganchev, I.; O'Droma, M.; Meere, D. Intelligent car parking locator service. *Int. J. ITK* 2008, 2, 166–173.

10. Benson, J.P.; O'Donovan, T.; O'Sullivan, P.; Roedig, U.; Sreenan, C.; Barton, J.; Murphy, A.; O'Flynn, B. Car-park management using wireless sensor networks. Proceedings of 31st IEEE Conference on the Local Computer Networks, Tampa, FL, USA, 14–16 November 2006; pp. 588–595.
11. Di Lecce, V.; Amato, A. Route planning and user interface for an advanced intelligent transport system. IET. *Intell. Transp. Syst.* 2011, 5, 149–158.
12. Caicedo, F.; Vargas, J. Access control systems and reductions of driver's wait time at the entrance of a car park. Proceedings of the 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), Singapore, 18–20 July 2012; pp. 1639–1644.
13. Boussier, J.M.; Estrailier, P.; Sarramia, D.; Augeraud, M. Using agent-based of driver behavior in the context of car park optimization. Proceedings of the 3rd International IEEE Conference on Intelligent Systems, London, UK, 4–6 September 2006; pp. 395–400.
14. Bitam, S.; Mellouk, A. Its-cloud: Cloud computing for intelligent transportation system. Proceedings of the IEEE Global Communications Conference, Anaheim, CA, USA, 3–7 December 2012; pp. 2054–2059.
15. ITU-T Study Group 13 on Future Networks including Cloud Computing, Mobile and Next-Generation Networks. Available online: <http://www.itu.int/en/ITU-T/about/groups/Pages/sg13.aspx> (accessed on 23 May 2022).
16. Ji, Z.; Ganchev, I.; O'Droma, M. An iWBC consumer application for 'always best connected and best served': Design and implementation. *IEEE. Trans. Consum. Electron.* 2011, 57, 462–470.
17. Domingo, A.; Bellalta, B.; Palacin, M.; Oliver, M.; Almirall, E. Public open sensor data: Revolutionizing smart cities. *IEEE Technol. Soc. Mag.* 2013, 32, 50–56.
18. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I. A view of cloud computing. *Commun. ACM* 2010, 53, 50–58.

19. Deka, G.C. A Survey of Cloud Database Systems. *IT. Prof.* 2014, 16, 50–57.]
20. White, T. *Hadoop: The Definitive Guide*; O'Reilly Media Inc.: Sebastopol, CA, USA, 2012.
21. Hall, R.; Pauls, K.; McCulloch, S.; Savage, D. *OSGi in Action: Creating Modular Applications in Java*; Manning Publications: Greenwich, CT, USA, 2011.
22. George, L. *Hbase: The Definitive Guide*; O'Reilly Media Inc.: Sebastopol, CA, USA, 2011.
23. Thusoo, A.; Sarma, J.S.; Jain, N.; Shao, Z.; Chakka, P.; Zhang, N.; Antony, S.; Liu, H.; Murthy, R. *HIVE-A petabyte scale data warehouse using hadoop*. Available online: <http://infolab.stanford.edu/~ragho/hive-icde> (accessed on 24 May 2022).
24. Dean, J.; Ghemawat, S. Mapreduce: A flexible data processing tool. *Commun. ACM* 2010, 53, 72–77.
25. Michal, B. *Drools JBoss Rules 5.0 Developer's Guide*; Packt Publishing Ltd.: Birmingham, UK, 2009.
26. Carlson, J.L. *Redis in Action*; Manning Publications: Greenwich, CT, USA, 2013.
27. Meier, R. *Professional Android 4 Application Development*; John Wiley & Sons: New York, NY, USA, 2012.
28. Zhanlin, J.; Ganchev, I.; O'Droma, M. Performance Evaluation of 'WBC over DVB-H' System. *IEEE Trans. Consum. Electron.* 2009, 55, 754–762.]
29. Kreps, J.; Narkhede, N.; Rao, J. In *Kafka: A distributed messaging system for log processing*. Proceedings of the Sixth International Workshop on Networking Meets Databases Workshop, Athens, Greece, 12–16 June 2011; pp. x.1–x.7.
30. Borthakur, D. *The Hadoop Distributed File System: Architecture and Design*; The Apache Software Foundation: Forest Hill, MD, USA, 2007.]
31. Chauhan, A. *Learning Cloudera Impala*; Packt Publishing Ltd.: Birmingham, UK, 2013.

32. Hoffman, S.; D'Souza, S. Apache Flume: Distributed Log Collection for Hadoop; Packt Publishing Ltd.: Birmingham, UK, 2013.
33. Reese, W. Nginx: The high-performance web server and reverse proxy. *Linux J.* 2008, 9, 1–4.
34. Humphrey, W.S. Personal Software Process (PSP); Wiley Online Library: Hoboken, NJ, USA, 2002.
35. Graziano, P. Speed up your web site with Varnish. *Linux. J.* 2013, 6, 1–5.]
36. Hunt, P.; Konar, M.; Junqueira, F.P.; Reed, B. Zookeeper: Wait-free coordination for internet-scale systems. *Proceedings of the USENIX Annual Technical Conference, Boston, MA, USA, 23–25 June 2010*; pp. x.1–x.14.