

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Комп'ютерних інтелектуальних технологій та систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Інтелектуальна система моніторингу сервісу оренди особистого транспорту.

Виконав:

студент 2 курсу, групи КІТм-21-1

Бова І.В.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник проф. Аксак Н.Г.

Допускається до захисту

(підпис)

Зав. кафедри

(підпис)

проф. Руденко О.Г.

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти другий (магістерський)
Спеціальність 123 – Комп'ютерна інженерія
Тип програми освітньо-професійна
Освітня програма Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Бові Івану Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна система моніторингу сервісу оренди особистого транспорту

затверджена наказом по університету від “ 07 ” листопада 2022 р. № 1455 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2022 р.

3. Вхідні дані до роботи _____

1) Робоча машина на базі процесора AMD Ryzen 7 2700X, відео-картою GTX 1070Ti та 32 гігабайтів оперативної пам'яті

2) Операційна система Windows 10 (64 bit)

МаaS – Мобільність як послуга

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз існуючих рішень мобільності як послуги

2) Вибір комплектуючих

3) Розробка функціональної схеми

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 12 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та актуальності сервісу оренди особистого транспорту	08.11 – 13.11	
2	Огляд сучасної літератури за напрямком магістерської роботи	13.11 – 20.11	
3	Аналіз існуючих рішень мобільності як послуги	21.11 – 23.11	
4	Огляд та вибір комплектуючих майбутньої системи	24.11 – 26.11	
5	Розробка програмного забезпечення системи моніторингу оренди особистого транспорту	26.11 – 30.12	
6	Оформлення пояснювальної записки	01.12 – 07.12	
7	Оформлення графічного матеріалу	08.12 – 10.12	

Дата видачі завдання 07 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Проф. Аксак Н.Г.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 74 с., 12 рис., 2 табл., 1 дод., 21 джерел.

ОРЕНДА ОСОБИСТОГО ТРАНСПОРТУ, МОНІТОРИНГ, СЕРВІС

Метою роботи є розробка системи моніторингу сервісу оренди особистого транспорту.

У магістерській роботі пропонується формалізація процесу обслуговування особистого транспорту, яка складається із структурної та функціональної схем блоку життєвого циклу послуги «Виклик особистого транспорту», що дозволить ефективно використовувати спільні послуги мобільності (використання автомобілів, велосипедів тощо).

Розроблено специфікації системи моніторингу сервісу оренди особистого транспорту.

Об'єктом дослідження є процеси надання сервісу оренди особистого транспорту.

Предметом дослідження є математичні моделі та методи, що орієнтовані на обробку інформації в процесі моніторингу особистого транспорту.

ABSTRACT

Master's thesis: 74 pages, 12 figures, 2 tables, 1 appendices, 21 sources.

RENTAL OF PERSONAL VEHICLES, MONITORING, SERVICE

The purpose of the work is to develop a monitoring system for personal transport rental service.

The master's thesis proposes the formalization of the personal transport service process, which consists of structural and functional diagrams of the life cycle block of the "Bicycle Call" service, which will allow the effective use of shared mobility services (use of cars, bicycles, etc.).

The specifications of the personal transport rental service monitoring system have been developed.

The object of the research is the processes of providing the personal transport rental service.

The subject of the research is mathematical models and methods focused on information processing in the process of personal transport monitoring.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ Комп'ютерних інтелектуальних технологій та систем _____

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти другий (магістерський)

Інтелектуальна система моніторингу сервісу оренди особистого

Виконав:

студент 2 курсу, групи КІТм-21-2

Бова І.В.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник проф. Аксак Н.Г.

2022 р.

АНОТАЦІЯ

Актуальність теми дослідження. Зростаючий тиск на системи міського пасажирського транспорту збільшив попит на нові та інноваційні рішення для підвищення його ефективності. Одним із підходів для вирішення цієї проблеми став повільний, але постійний перехід до спільних послуг мобільності (спільне використання автомобілів, велосипедів тощо). Спираючись на ці нові види транспорту та розвиток інформаційно-комунікаційних технологій, нещодавно з'явилася концепція «Мобільність як послуга» (MaaS), яка пропонує зручне транспортування від дверей до дверей без необхідності витратити кошти на зберігання та технічне обслуговування приватного транспортного засобу. Термін «Мобільність як послуга» (MaaS) означає купівлю послуг мобільності на основі потреб споживачів замість покупки засобів мобільності. В останні роки в усьому світі з'явилися різні схеми MaaS.

Термін «Мобільність як послуга» означає купівлю послуг мобільності як пакетів на основі потреб споживачів замість покупки транспортних засобів. За допомогою систем «Мобільність як послуга» споживачі можуть купувати послуги мобільності, які надаються одним або різними операторами, використовуючи лише одну платформу та єдиний платіж. З точки зору бізнесу все складніше: потрібен постійний контроль, як за технічним станом, так і за розміщенням усіх транспортних засобів. Для підтримки належного стану потрібні майстри, а для того, щоб знати положення транспорту потрібна автоматизована система моніторингу.

Платформи MaaS зазвичай надають планувальник інтермодальних подорожей (забезпечуючи комбінації різних видів транспорту: спільне використання автомобілів, прокат автомобілів, метро, залізниця, автобус, спільне використання велосипедів, таксі тощо), систему бронювання, легку оплату та реальну інформацію про час. Користувачі MaaS можуть використовувати послугу як Pay-As-You-Go, або вони можуть придбати пакети мобільності відповідно до своїх потреб у подорожах.

Перспективним підходом до зменшення обсягу перевезень, підвищення ефективності використання ресурсів і водночас задоволення індивідуального попиту на мобільність може бути впровадження бізнес-моделі «продаж використання замість продажу продукту» в індивідуальних перевезеннях. Це можна реалізувати за допомогою мережі постачальників послуг, які пропонують клієнтам різні транспортні засоби за якістю, часом і місцем розташування, наприклад громадський транспорт, компанії з прокату автомобілів, мотоциклів і велосипедів. Тут декілька клієнтів можуть спільно використовувати один і той же засіб пересування, наприклад, автомобіль, у різний час.

Метою роботи є розробка системи моніторингу сервісу оренди особистого транспорту.

У магістерській роботі пропонується формалізація процесу обслуговування ОТ, яка складається із структурної та функціональної схем блоку життєвого циклу послуги «Виклик ОТ», що дозволить ефективно використовувати спільні послуги мобільності (використання автомобілів, велосипедів тощо).

Розроблено проєкт системи моніторингу сервісу оренди особистого транспорту.

Об'єктом дослідження є процеси надання сервісу оренди особистого транспорту.

Предметом дослідження є математичні моделі та методи, що орієнтовані на обробку інформації в процесі моніторингу особистого транспорту.

У **першому розділі** розглянуто актуальність та проблеми соціальної мобільності суспільства.

Всесвітня рада бізнесу зі сталого розвитку визначає стійку мобільність як «здатність задовольняти потреби суспільства вільно пересуватися, отримувати доступ, спілкуватися, торгувати та встановлювати відносини без шкоди для інших основних людських або екологічних цінностей сьогодні та в майбутньому».

Соціальна мобільність суспільства, що означає здатність пересуватися в соціальних ієрархіях, часто прямо пов'язана з просторовою мобільністю. З огляду на те, що до 2030 року половина населення країн, що розвиваються, проживатиме в

містах, створюються величезні нові вимоги до мобільності. Завдяки прагненню людей до доступних товарів, виняткових продуктів або передових технологій світова торгівля різко зростає, а регіональні ринки розширюються до глобальних масштабів.

Також зроблено аналіз проблеми моніторингу транспортних засобів

Одним з прикладів придатних транспортних засобів у цій мережі може бути ОТ із інтегрованою інформаційною системою супроводу продукту, так званою блоком життєвого циклу (БЖЦ). Наприклад, моніторинг положення ОТ за допомогою глобальної системи позиціонування (GPS), передача його через глобальну систему мобільного зв'язку (GSM) на станцію керування та надання даних в Інтернеті може дозволити клієнту знайти ОТ поблизу його розташування. Це також може підвищити ефективність роботи ОТ для постачальника послуг, наприклад, шляхом полегшення технічного обслуговування на основі стану.

У другому розділі зрблений аналіз технологій для реалізації системи оренди транспортних засобів.

До інтелектуальної системи моніторингу сервісу оренди транспортних засобів можна висунути наступні вимоги:

- Здатність отримувати великі об'єми даних від БЦЖ транспортних засобів у реальному часі.
- Здатність надійно зберігати дані.
- Здатність проводити обробку та аналіз даних у режимі близькому до реального часу.
- Можливість візуалізації оброблених даних у інформаційні панелі, відправка попереджень про збої, поломки, небажану активність, скупчення транспортних засобів в одному місці або нестачу в певних районах.

Відповідно до висунутих потреб необхідно підібрати технології, які дозволять вести розробку швидко та зручно і дозволять виконати усі поставлені завдання. Потрібно обрати сучасну мову програмування, з можливістю розробки з використанням Big Data технологій, такою мовою може стати Scala.

Виконувати пакетну обробку і обробку в реальному часі дозволить фреймворк

Spark, який є однією з найпоширеніших технологій для розподілених обчислень. Для надійного зберігання даних було вирішено обрати Delta Lake, що є сховищем розробленим для використання у поєднанні з фреймворком Spark. Для потокової передачі та зберігання повідомлень можна використати платформу Apache Kafka, яка також має інтерфейс для інтеграції зі Scala і Spark.

Одним із сховищ стане розподілена база даних Cassandra, вона призначена для зберігання великих обсягів даних. Для оркестрації потоків даних було вирішено використовувати програму Airflow, яка використовує мову програмування Python для організації пайплайнів.

Для організації інформації і подій у зручні та інформативні панелі було вирішено використовувати систему Elasticsearch.

У **третьому розділі** пропонується обґрунтування вибору архітектури системи оренди ОТ.

Архітектура програмного забезпечення відноситься до фундаментальних структур програмної системи та дисципліни створення таких структур та систем. Кожна структура включає елементи програмного забезпечення, відносини з-поміж них, і навіть властивості як елементів, і відносин. Архітектура програмної системи – це метафора, аналогічна архітектурі будівлі. Вона функціонує як план для системи та проекту розробки, в якому викладаються завдання, які мають бути виконані командами розробників.

Чим старшим стає проект, тим більшим є технічний борг, навіть якщо функціональність проекту не сильно змінюється, то технічний борг все одно накопичується з часом і в якийсь критичний момент виправлення дефектів займатиме занадто багато ресурсів, а на нову функціональність часу просто не вистачатиме.

Найголовніше, необхідно зібрати правильні вимоги до архітектури, що враховують усі складності розробки, ці вимоги можуть бути абсолютно різними,

Четвертий розділ присвячений розробці специфікації вимог до онлайн-системи оренди автомобілів

Кожен тип особистого транспорту має різну вартість оренди на день. Вартість

оренди залежить від кількості днів, марки та швидкості руху ОТ. Система повинна мати такі функції:

Оренда: система, обладнана для відповідей на запити *Клієнта* щодо наявності та вартості оренди різних «*Типів*» ОТ на певні дати в майбутньому. Коли клієнт приймає рішення щодо «*Типу*» ОТ та дат, система повинна мати можливість «*Зарезервувати*» або «*Визначити*» тип ОТ для запиту на запитані дати. Клієнту слід надати «номер підтвердження»

Підбирання: система обробляє підбірку ОТ. Клієнт заходить і надає підтвердження: номер або ім'я. Система має отримати всю інформацію про бронювання цього клієнта. Потім клієнта просять надати водійські права.

Повернення: система запускає повернення. Система повинна зафіксувати дату та час повернення, і в залежності від цих параметрів розрахувати остаточну суму оренди.

Основна мета полягає в такому: розробити веб-систему, яка дозволить клієнтам онлайн реєструвати та бронювати автомобіль, а компанії – ефективно керувати бізнесом з оренди автомобілів.

ОРЕНДА ОСОБИСТОГО ТРАНСПОРТУ, МОНІТОРИНГ, СЕРВІС

Список опублікованих робіт за темою магістерської роботи:

1. Н. Аксак , І. Бова. Огляд Big Data та аналітики у медицині та охороні здоров'я середовища. *Традиції та нові наукові стратегії у Центральній та Східній Європі*: Матеріали V Міжнародної науково-практичної конференції (м. Київ, 24–25 червня 2022 р.) / ГО «Інститут інноваційної освіти»; Науково-навчальний центр прикладної інформатики НАН України. – Київ : ГО «Інститут інноваційної освіти», 2022. -С. 125 -127.

Використані публікації керівника та співробітників кафедри, що становлять теоретичну базу роботи:

1. Аксак Н.Г. Дистанционный мониторинг пациента на основе концепции агент-группа-роль. *Системні дослідження та інформаційні технології*. – 2018. – № 3. – С. 7–18.

2. Axak N. Development of multi-agent system of neural network diagnostics and remote monitoring of patient. *Eastern-European Journal of Enterprise Technologies*. – 2016. – № 4/9 (82) – P. 4–11.

3. Аксак Н.Г. Мультиагентная система нейросетевой диагностики и удаленного мониторинга пациента. *Інформаційні технології: проблеми та перспективи: монографія*. за заг. ред. В. С. Пономаренка. Х. : Вид. Рожко С.Г., 2017. – С. 325–340.

4. Аксак Н.Г. Система віддаленого нейромережного діагностування меланоми. *Інформаційні системи та технології в медицині*. I Міжнародна науково-практична конференція (ISM-2018). Збірник наукових праць. ХНУРЕ. Харків: «Друкарня Мадрид», 2018.- 116–118.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	10
Вступ	11
1 Аналіз предметної області	14
1.1 Соціальна мобільність суспільства	14
1.2 Аналіз проблеми моніторингу транспортних засобів	15
1.3 Мобільність як послуга МaaS	17
1.4 Огляд літератури	21
2 Обґрунтування вибору технологій для розробки системи оренди транспортних засобів	24
2.1 Аналіз технологій для реалізації системи оренди транспортних засобів	24
2.1.1 Scala	25
2.1.2. Spark	26
2.1.3. Delta Lake	27
2.1.4 Kafka	28
2.1.5 Cassandra	29
2.1.6 Elasticsearch	30
2.1.7 Airflow	31
2.1.8 Docker	33
2.1.9 Apache Parquet	34
2.2 Аналіз сервісів оренди та продажу ОТ	35
2.3 Послуга «Виклик ОТ»	37
3 Розробка архітектури системи оренди особистого транспорту	40
3.1 Обґрунтування вибору архітектури системи оренди ОТ	40
3.1.2 Сервісна архітектура	41
3.2 Розробка архітектури	43
3.2 Формалізація процесу обслуговування особистого транспорту	44
4 Експериментальне моделювання	47

4.1 Специфікація вимог до онлайн-системи оренди ОТ	47
4.2 Функції користувачів в онлайн-системі	48
4.3 Спеціальні вимоги	49
4.4 Нефункціональні вимоги	52
Висновки	59
Перелік використаних джерел	60
Додаток А	62
Графічний матеріал атестаційної роботи	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІКТ – інформаційно-комунікаційні технології

БЖЦ – блок життєвого циклу

МaaS – Мобільність як послуга

GPS – глобальна система позиціонування

GSM – глобальна система мобільного зв'язку

NMS – нові послуги мобільності

ШІ – штучний інтелект

МАС – мультиагентна система

МСС – головний центр управління

ОТ – особистий транспорт

ВСТУП

Життя у великому місті потребує від людини мобільності. У пересічного мешканця Харкова щоденно виникає потреба в пересуванні з одного місця в інше, наприклад, з дому на роботу, в магазин, та ще багато куди. Зазвичай у людини є вибір між особистим і громадським транспортом, обидва мають свої переваги і недоліки.

Громадський транспорт дешевий і екологічний за рахунок перевезення великої кількості пасажирів за одну поїздку, не накладає ніяких додаткових обов'язків, але обмежує переміщення чітко визначеними маршрутами. Особистий транспорт надає власнику повну свободу і комфорт, але вимагає місце для зберігання, передбачає витрати на технічне обслуговування, а також автомобілі і двоколісний транспорт з двигуном внутрішнього згоряння забруднюють навколишнє середовище і потребують специфічної інфраструктури: доріг і паркових майданчиків. Також міста, які побудовані з розрахунком на переважне використання автомобільного транспорту є незручними, а іноді й небезпечними для пішоходів і водіїв ОТ.

Зростаючий тиск на системи міського пасажирського транспорту збільшив попит на нові та інноваційні рішення для підвищення його ефективності. Одним із підходів для вирішення цієї проблеми став повільний, але постійний перехід до спільних послуг мобільності (спільне використання автомобілів, велосипедистів тощо). Спираючись на ці нові види транспорту та розвиток інформаційно-комунікаційних технологій, нещодавно з'явилася концепція «Мобільність як послуга» (MaaS), яка пропонує зручне транспортування від дверей до дверей без необхідності витратити кошти на зберігання та технічне обслуговування приватного транспортного засобу. Термін «Мобільність як послуга» (MaaS) означає купівлю послуг мобільності на основі потреб споживачів замість покупки засобів мобільності. В останні роки в усьому світі з'явилися різні схеми MaaS.

Термін «Мобільність як послуга» означає купівлю послуг мобільності як пакетів на основі потреб споживачів замість покупки транспортних засобів. За

допомогою систем «Мобільність як послуга» споживачі можуть купувати послуги мобільності, які надаються одним або різними операторами, використовуючи лише одну платформу та єдиний платіж. З точки зору бізнесу все складніше: потрібен постійний контроль, як за технічним станом, так і за розміщенням усіх транспортних засобів. Для підтримки належного стану потрібні майстри, а для того, щоб знати положення транспорту потрібна автоматизована система моніторингу.

Платформи МaaS зазвичай надають планувальник інтермодальних подорожей (забезпечуючи комбінації різних видів транспорту: спільне використання автомобілів, прокат автомобілів, метро, залізниця, автобус, спільне використання велосипедистів, таксі тощо), систему бронювання, легку оплату та реальну інформація про час. Користувачі МaaS можуть використовувати послугу як Pay-As-You-Go, або вони можуть придбати пакети мобільності відповідно до своїх потреб у подорожах.

Перспективним підходом до зменшення обсягу перевезень, підвищення ефективності використання ресурсів і водночас задоволення індивідуального попиту на мобільність може бути впровадження бізнес-моделі «продаж використання замість продажу продукту» в індивідуальних перевезеннях. Це можна реалізувати за допомогою мережі постачальників послуг, які пропонують клієнтам різні транспортні засоби за якістю, часом і місцем розташування, наприклад громадський транспорт, компанії з прокату автомобілів, мотоциклів і велосипедів. Тут декілька клієнтів можуть спільно використовувати один і той же засіб пересування, наприклад, автомобіль, у різний час.

Адекватне використання засобів інформаційно-комунікаційних технологій (ІКТ) у мережі постачальників послуг, їхнього обладнання та клієнта є важливим для надання конкурентоспроможних послуг.

Одним із відповідних транспортних засобів у цій мережі може бути ОТ із інтегрованою інформаційною системою, що супроводжує продукт, так званим блоком життєвого циклу (БЖЦ). БЖЦ інтегровані в продукти та здатні отримувати, обробляти та передавати відповідні дані про продукт і процес оцінки протягом усього життєвого циклу продукту.

Інтегрований у ОТ БЖЦ має завдання оцінювати, прогнозувати, діагностувати та контролювати ОТ та його компоненти. Наприклад, моніторинг положення ОТ за допомогою глобальної системи позиціонування (GPS), передача інформації через глобальну систему мобільного зв'язку (GSM) на станцію керування та надання даних в Інтернеті може дозволити клієнту знайти ОТ поблизу його розташування.

Це також може підвищити ефективність роботи ОТ для постачальника послуг, наприклад, шляхом полегшення технічного обслуговування на основі стану. Крім того, у БЖЦ інтегрована функція керування живленням на основі умов. Це забезпечить розподіл наявної обмеженої електричної енергії на різні навантаження на ОТ, наприклад, БЖЦ та освітлення.

Для того, щоб управляти подібним бізнесом потрібна система моніторингу переміщення та стану кожної одиниці транспортного засобу. Така система зможе приймати повідомлення від інтегрованих БЖЦ, обробляти їх і представляти інформацію у зручному вигляді, а також робити прогнози на основі отриманих даних. Це дозволить персоналу реагувати на можливі проблеми.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Соціальна мобільність суспільства

Всесвітня рада бізнесу зі сталого розвитку визначає стійку мобільність як «здатність задовольняти потреби суспільства вільно пересуватися, отримувати доступ, спілкуватися, торгувати та встановлювати відносини без шкоди для інших основних людських або екологічних цінностей сьогодні та в майбутньому».

Соціальна мобільність суспільства, що означає здатність пересуватися в соціальних ієрархіях, часто прямо пов'язана з просторовою мобільністю. З огляду на те, що до 2030 року половина населення країн, що розвиваються, проживатиме в містах, створюються величезні нові вимоги до мобільності. Завдяки прагненню людей до доступних товарів, виняткових продуктів або передових технологій світова торгівля різко зростає, а регіональні ринки розширюються до глобальних масштабів.

Збільшення мобільності може створити економічне процвітання в усьому світі, позбавляючи мільйони людей від бідності та надаючи їм доступ до кращих медичних послуг, можливостей роботи та освіти, а також забезпечуючи швидке та надійне перевезення продуктів харчування та інших товарів. У той же час мобільність призводить до більшого споживання ресурсів (енергії, матеріалів тощо) і забруднення навколишнього середовища через збільшення викидів парникових газів. Соціальна напруга може виникнути через неминучий нерівний розподіл багатства. Але сьогоднішні перешкоди мультимодального транспорту вже помітні у вантажних перевезеннях. Застаріла технологія та низька ефективність, навіть у такій промислово розвиненій країні, як Німеччина, призвели до одностороннього переходу від внутрішнього водного та залізничного транспорту до більш гнучкого, але також більш ресурсозатратного автомобільного транспорту. Адекватне використання засобів доступних інформаційно-комунікаційних технологій (ІКТ) у мережі постачальників послуг, їхнього обладнання та клієнта є важливим для

надання конкурентоспроможних послуг.

ІКТ також можуть створити конкурентоспроможні нові форми колективного транспортування вантажів, наприклад, повідомлення в плящі, що забезпечує кожне домогосподарство щоденними товарами, такими як продукти харчування чи косметика. Але також нове використання поважних екологічно чистих засобів транспорту дає шанс підвищити їхню конкурентоспроможність. Потяги на магнітній подушці могли одночасно перевозити пасажирів і вантажі з досі неперевершеною швидкістю над землею і таким чином напряду конкурувати з літаками.

Щоб реалізувати бачення багатонаціональних мультимодальних перевезень пасажирів і вантажів, необхідно здійснити прагматичні зміни в інфраструктурі та ефективно перехресне зв'язування клієнтів, постачальників послуг, виробників, урядових і неурядових організацій у транспортних відносинах. Якщо це буде успішно, це може стати прикладом і дати зростання економіки менш розвинутих країнам і країнам, що розвиваються, вони матимуть шанс задовольнити потреби своїх громадян у більшій мобільності з меншим споживанням ресурсів. У той же час вони можуть отримати вигоду від знань, які вони повинні отримати, щоб розвивати, адаптувати та використовувати ці сучасні засоби транспорту та їхні ІКТ.

1.2 Аналіз проблеми моніторингу транспортних засобів

Одним з прикладів придатних транспортних засобів у цій мережі може бути ОТ із інтегрованою інформаційною системою супроводу продукту, так званою блоком життєвого циклу (БЖЦ). Наприклад, моніторинг положення ОТ за допомогою глобальної системи позиціонування (GPS), передача його через глобальну систему мобільного зв'язку (GSM) на станцію керування та надання даних в Інтернеті може дозволити клієнту знайти ОТ поблизу його розташування. Це також може підвищити ефективність роботи ОТ для постачальника послуг, наприклад, шляхом полегшення технічного обслуговування на основі стану.

ОТ є дуже мобільним транспортним засобом, і його можна легко адаптувати. ОТ оснащені БЖЦ, сприяють більшій продуктивності використання та

мінімальному негативному впливу на навколишнє середовище, наприклад, забруднення.

Автоматизована система оренди транспорту передбачає, що клієнт використовує свій смартфон зі встановленим на нього застосунком компанії прокату для того, щоб знайти найближчий до нього припаркований на вулиці транспортний засіб, забронювати його, обрати та сплатити тариф, після чого людина зможе їхати. Діставшись певного місця, користувач знову використовує додаток для того, щоб підтвердити місце паркування, на цьому поїздка завершується. Така модель поїздок потребує особливої обробки і моніторингу стану та положення.

Через те, що не існує чітко визначених місць паркування для транспортних засобів, необхідно відстежувати кожний з них за допомогою системи GPS. Це забезпечить можливість знайти кожний самокат в кінці дня, коли йому буде потрібна зарядка або інше технічне обслуговування. Крім того, може статися ситуація, коли в певному місці накопичується велика кількість одиниць транспортних засобів або десь їх зовсім немає, у такому випадку система відстеження дозволить провести аналіз та відреагувати на це. Також відстеження положення дозволить попередити крадіжку, наприклад, при переміщенні не розблокованого транспорту ввімкнеться сигналізація. В цілому автоматизована система відстеження, моніторингу, аналізу даних про поїздки може надати бізнесу важливу інформацію і вберегти від збитків.

У мегаполісах кількість транспортних засобів, доступних для оренди, досягатиме десятків тисяч, а кількість повідомлень, які вони генерують може нараховувати мільйони щодня. Такі об'єми даних потребують спеціального підходу. Традиційні комп'ютерні системи, побудовані на основі одного сервера, що займається усіма операціями не підходять коли кількість даних обчислюється в терабайтами.

Існує два підходи для вирішення проблеми недостатньої обчислювальної потужності системи: вертикальний та горизонтальний ріст. Вертикальний ріст передбачує збільшення потужності одного комп'ютера за рахунок встановлення більш потужних компонентів наприклад процесора, або більшої кількості пам'яті.

Горизонтальний ріст системи передбачує об'єднання комп'ютерів у кластери, і при нестачі ресурсів до кластера приєднується ще одна машина. Горизонтальне масштабування дозволяє позбутися обмежень потужності одної обчислювальної машини, але накладає додаткові складності пов'язані з організацією взаємодії машин у кластері.

При великих об'ємах даних переваги використання розподілених систем очевидні, а проблеми організації вирішуються різними готовими інструментами. При побудові такої системи важливо правильно оцінити усі потреби бізнесу і правильно обрати інструменти для розробки. Аналіз задачі та правильне планування архітектури дозволять уникнути зайвих витрат, та розробити стійку систему.

1.3 Мобільність як послуга МaaS

Останніми роками збільшення кількості транспортних послуг, що пропонуються в містах, а також розвиток технологій та ІКТ запровадили інноваційну концепцію мобільності як послуги (МaaS) [1]. Він поєднує різні види транспорту, щоб запропонувати індивідуальний пакет мобільності, подібний до місячного контракту на мобільний телефон, і включає інші додаткові послуги, такі як планування поїздок, бронювання та платежі через єдиний інтерфейс. Таке об'єднання способів мобільності означає відхід від існуючої транспортної системи, заснованої на власності, до транспортної системи, що базується на доступі. Він пропонує користувачам індивідуальне гіперзручне рішення для мобільності з багатообіцяючою перспективою замінити приватний автомобіль.

Враховуючи багатообіцяючі перспективи, навколо концепції все ще існує високий ступінь неоднозначності з кількома джерелами, які змагаються, щоб запропонувати визначення МaaS, багато з яких можуть суперечити одне одному або розглядати різні аспекти концепції взагалі. Крім того, хоча кілька схем МaaS було впроваджено в усьому світі, бракує системи оцінки, яка систематично класифікувала б їхні унікальні характеристики. «Що таке концепція МaaS?»

Щоб досягти стійкості в транспортному секторі, нові технології мають бути доповнені новими бізнес-моделями економіки спільного використання. Перш ніж запроваджувати політику сприяння їх розповсюдженню, необхідно розглянути сегментацію ринку та позиціонування.

Інтегрована та бездоганна мобільність, ідея, що лежить в основі мобільності як послуги, була баченням міського транспорту майбутнього. МaaS базується на трьох основних елементах, які в поєднанні забезпечують користувачам безперервні інтермодальні подорожі. Це:

- Інтеграція квитка та платежу: коли одна смарт-карта або квиток можуть використовуватися для доступу до всіх режимів, які беруть участь у службі, і один обліковий запис стягується за використання цих послуг;
- Пакет мобільності: коли клієнти можуть передоплатити певну суму (за часом або відстанню) комбінації послуг мобільності;
- Інтеграція ІКТ: коли є єдина програма або онлайн-інтерфейс, який можна використовувати для доступу до інформації про режими.

Досі ще не достатньо досліджень спільного впливу цих елементів на поведінку під час подорожі та вибір способу.

Мобільність як послуга (MaaS) –це зовсім нова концепція мобільності. Його можна розглядати як концепцію (нова ідея для уявлення про мобільність), явище (виникає з появою нових моделей поведінки та технологій) або як нове транспортне рішення (яке об'єднує різні доступні види транспорту та послуги мобільності). Перше вичерпне визначення MaaS запропоновано Hietanen в 2014р. Він описує MaaS як модель розподілу мобільності, яка забезпечує транспортні потреби користувачів через єдиний інтерфейс постачальника послуг. Він поєднує різні види транспорту, щоб запропонувати індивідуальний пакет мобільності, як-от місячний контракт на мобільний зв'язок. Ця інтерпретація охоплює деякі основні характеристики MaaS: орієнтованість на потреби клієнта, об'єднання послуг, кооперацію та взаємозв'язок між видами транспорту та постачальниками послуг.

У таблиці 1.1 наведено аналіз пілотних проектів і тематичних досліджень, які були реалізовані в контексті MaaS і описані відповідно до основних характеристик.

Таблиця 1.1 – Опис основних характеристик МaaS

Основна характеристика	Опис
1. Інтеграція видів транспорту	Метою схем МaaS є заохочення використання послуг громадського транспорту, об'єднуючи мультимодальні перевезення та дозволяючи користувачам вибирати та сприяючи їм у їхніх інтермодальних поїздках. Можуть бути включені такі види транспорту: громадський транспорт, таксі, спільне використання автомобілів, спільне використання поїздок, спільне використання ОТ, оренда автомобілів, автобусні послуги на замовлення.
2. Тарифний варіант	Платформа МaaS пропонує користувачам два типи тарифів для доступу до своїх послуг мобільності: «пакет мобільності» та «оплата за ходом». Пакет пропонує набори різних видів транспорту та включає певну кількість кілометрів/хвилин/балів, які можна використати в обмін на щомісячну оплату.
3. Одна платформа	МaaS спирається на цифрову платформу (мобільний додаток або веб-сторінку), за допомогою якої кінцеві користувачі можуть отримати доступ до всіх необхідних послуг для своїх подорожей: планування поїздок, бронювання, оформлення квитків, оплата та інформація в реальному часі. Користувачі також можуть отримати доступ до інших корисних послуг, таких як прогноз погоди, синхронізація з особистим календарем активності, звіт про історію подорожей, виставлення рахунків і зворотній зв'язок.
4. Кілька акторів	Екосистема МaaS побудована на взаємодії між різними групами учасників за допомогою цифрової платформи: споживачами мобільності (наприклад, приватні клієнти або бізнес-клієнти), постачальниками транспортних послуг (наприклад, державними чи приватними) і власниками платформ. Інші учасники також можуть співпрацювати, щоб забезпечити функціонування служби та підвищити її ефективність: місцеві органи влади, клірингові компанії, телекомунікаційні компанії та компанії з управління даними.
5. Використання технологій	Для забезпечення МaaS поєднуються різні технології: пристрої, такі як мобільні комп'ютери та смартфони; надійна мережа мобільного Інтернету (WiFi, 3G, 4G, LTE); GPS; система електронного продажу квитків та електронних платежів; система управління базами даних та інтегрована інфраструктура технологій (тобто IoT).

Закінчення таблиці 1.1

1	2
6. Орієнтація на попит	MaaS — це орієнтована на користувача парадигма. Він прагне запропонувати транспортне рішення, найкраще з точки зору клієнта, завдяки функції планування мультимодальних поїздок і включенню послуг, що реагують на попит, наприклад таксі.
7. Вимоги до реєстрації	Кінцевий користувач має приєднатися до платформи, щоб отримати доступ до доступних послуг. Обліковий запис може бути дійсним для однієї особи або, у деяких випадках, для всієї родини. Підписка не тільки полегшує користування послугами, але й дозволяє персоналізувати послугу.
8. Персоналізація	Персоналізація гарантує, що вимоги та очікування кінцевих користувачів задовольняються ефективніше та ефективніше, враховуючи унікальність кожного клієнта. Система надає кінцевому користувачеві конкретні рекомендації та індивідуальні рішення на основі його/її профілю, виражених уподобань і минулої поведінки (наприклад, історії подорожей). Крім того, вони можуть підключити свої профілі в соціальних мережах до свого облікового запису MaaS.
9. Налаштування	Налаштування дозволяє кінцевим користувачам змінювати пропонований варіант послуги відповідно до своїх уподобань. Це може підвищити привабливість MaaS серед мандрівників, а також задоволеність і лояльність клієнтів. Вони можуть вільно скласти певну ланцюгову поїздку або створювати свій пакет мобільності з різним обсягом використання певних видів транспорту, щоб краще досягти бажаного досвіду подорожі

Усі ці модельні системи, засновані на діяльності, були побудовані на припущенні, що рішення про подорожі приймаються в умовах визначеності. Цю гіпотезу можна вважати нереалістичною, оскільки на стан транспортних систем впливають різноманітні фактори невизначеності, а особи, які приймають рішення, не мають досконалого знання щодо свого вибору. Отже, щоб підвищити точність і надійність таких моделей, цікаві теорії та моделі людського вибору та прийняття рішень в умовах ризику та невизначеності були досліджені та розроблені в аналізі поведінки під час подорожей [2].

MaaS – це нова компанія і її впровадження в реальному світі все ще обмежене, існує дуже невелика кількість досліджень, які вивчають моделювання попиту на

подорожі для MaaS. В роботах [3, 4] вивчали зміни в поведінці подорожей, виборі режиму та рівні задоволеності користувачів шляхом збору та аналізу даних під час шестимісячного польового тестування UbiGo.

Таким чином, огляд літератури підтвердив випадок MaaS як інструмент, що сприяє вибору сталого транспорту. Результати свідчать про те, що багато спостережень про використання особистого транспорту не обов'язково виправдовуються в контексті послуг короткострокової оренди, особливо щодо соціально-економічних характеристик, таких як вік, сімейний статус, вид будинку та домоволодіння.

1.4 Огляд літератури

Метою статті [5] є дослідження існуючих послуг мобільності та розробити індекс для оцінки рівня інтеграції мобільності для кожної з них на основі припущення, що вищий рівень інтеграції більш привабливий для мандрівників. Огляд, представлений у цьому документі, дозволяє порівняти схеми та надає передумови та ключові моменти систем MaaS, які дослідницьке співтовариство, яке можна використовувати для розробки опитувань. Він також надає важливу інформацію для транспортних операторів і органів влади щодо елементів, які вони повинні взяти до уваги, щоб застосувати привабливу схему MaaS, яка могла б ефективно відвернути попит від приватних транспортних засобів.

У дослідженні [6] зроблено висновок, що впровадження нових послуг мобільності (NMS) пропонує потенціал для підвищення ефективності, стійкості, соціальної справедливості та якості життя. Основні висновки цього дослідження служать ідеальною відправною точкою для постачальників послуг мобільності та політиків, які стурбовані зростаючим попитом на чисті та зелені міста. Це дослідження пояснює ключові елементи того, як розглядати оцінку впливу NMS з точки зору соціальних, екологічних та економічних аспектів послуг стійкої мобільності.

У [7] представлено всебічний огляд літератури, зосереджений на пропозиції послуг мобільності, надаючи відповідну інформацію на концептуальному, операційному та моделювальному рівнях. Визначення спочатку беруться з парадигми мобільності як послуги через її переважання в літературі. Далі слідує оцінка операційних особливостей низки послуг мобільності, зокрема спільного використання автомобілів, велопробігу, послуг транспортування та транспорту. На завершення огляду повідомляється про сучасні підходи до моделювання послуг мобільності на різних рівнях складності та інтеграції. Три найважливіших висновки та аргументи з цієї статті свідчать про те, що існує високий ступінь загальності для операційних характеристик послуг мобільності; що важливо розрізняти мобільність як послугу та послугу мобільності окремо; поряд з аргументом про те, що роль людини слід ретельно враховувати під час моделювання процесів прийняття рішень як агентом користувача, так і агентом драйвера. Нарешті, пропонуються ключові міркування щодо майбутньої розробки концептуальної основи для моделювання сторони пропозиції послуг мобільності, яка матиме загальну модель постачальника послуг як основний компонент.

Агентно-базовані обчислення є однією з потужних технологій для розробки розподілених складних систем [8]. Багато дослідників вважають, що агенти являють собою найважливішу нову парадигму розробки програмного забезпечення з часів об'єктно-орієнтованого проектування [9], і концепція інтелектуальних агентів вже знайшла різноманітне застосування у виробництві, системах управління в реальному часі, електронній комерції, мережах. управління, транспортні системи, управління інформацією, наукові обчислення, охорона здоров'я та розваги. Причина зростаючого успіху агентної технології в цих областях полягає в тому, що властивий розподіл дозволяє природну декомпозицію системи на кілька агентів, які взаємодіють один з одним для досягнення бажаної глобальної мети. Агентна технологія може значно покращити проектування та аналіз проблемних областей за наступних трьох умов [10]: 1) проблемна область географічно розподілена; 2) підсистеми існують у динамічному середовищі; 3) підсистеми повинні більш гнучко взаємодіяти одна з одною.

Область руху та транспортних систем добре підходить для підходу на основі агентів через його географічно розподілену природу та чергування робочих характеристик «зайнято-неактивно» [11, 12]. З точки зору управління рухом і транспортуванням, найбільш привабливими характеристиками агентів є автономність, співпраця та реактивність.

Ця функція допомагає впроваджувати автоматизовані системи контролю та управління дорожнім рухом. У багатоагентній системі (MAS) агенти спілкуються з іншими агентами в системі для досягнення глобальної мети. Агенти також можуть сприймати навколишнє середовище та своєчасно реагувати на зміни навколишнього середовища. Транспортні системи на основі агентів дозволяють розподіленим підсистемам співпрацювати одна з одною для здійснення контролю та управління трафіком на основі умов руху в реальному часі. Тестовий стенд розподіленого моніторингу транспортних засобів, представлений у [13].

Останнім часом з'являється все більше додатків для транспортування та транспортування на основі агентів. Дослідження [14] показує, що прийоми та методи, отримані в області MAS, були застосовані до багатьох аспектів дорожнього руху та транспортних систем, включаючи моделювання та імітацію, інтелектуальний контроль і управління дорожнім рухом, динамічну маршрутизацію та управління заторами.

У роботах [17, 18, 19] розглянута організаційна концепція агент-група-роль для дистанційного моніторингу та критерії ефективності системи віддаленого моніторингу; а в [20, 21] – розроблені хмарні обчислення для віддаленого моніторингу та діагностування;

Таким чином, метою роботи є розробка системи моніторингу сервісу оренди особистого транспорту.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити існуючі сервіси оренди особистого транспорту;
- створити формалізацію процесу оренди особистого транспорту;
- розробити специфікацію вимог до програмного забезпечення системи оренди особистого транспорту.

2 ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ СИСТЕМИ ОРЕНДИ ТРАНСПОРТНИХ ЗАСОБІВ

2.1 Аналіз технологій для реалізації системи оренди транспортних засобів

До інтелектуальної системи моніторингу сервісу оренди транспортних засобів можна висунути наступні вимоги:

- Здатність отримувати великі об'єми даних від БЦЖ транспортних засобів у реальному часі.
- Здатність надійно зберігати дані.
- Здатність проводити обробку та аналіз даних у режимі близькому до реального часу.
- Можливість візуалізації оброблених даних у інформаційні панелі, відправка попереджень про збої, поломки, небажану активність, скупчення транспортних засобів в одному місці або нестачу в певних районах.

Відповідно до висунутих потреб необхідно підібрати технології, які дозволять вести розробку швидко та зручно і дозволять виконати усі поставлені завдання. Потрібно обрати сучасну мову програмування, з можливістю розробки з використанням Big Data технологій, такою мовою може стати Scala.

Виконувати пакетну обробку і обробку в реальному часі дозволить фреймворк Spark, який є однією з найпоширеніших технологій для розподілених обчислень. Для надійного зберігання даних було вирішено обрати Delta Lake, що є сховищем розробленим для використання у поєднанні з фреймворком Spark. Для потокової передачі та зберігання повідомлень можна використати платформу Apache Kafka, яка також має інтерфейс для інтеграції зі Scala і Spark.

Одним із сховищ стане розподілена база даних Cassandra, вона призначена для зберігання великих обсягів даних. Для оркестрації потоків даних було вирішено використовувати програму Airflow, яка використовує мову програмування Python для організації пайплайнів.

Для організації інформації і подій у зручні та інформативні панелі було вирішено використовувати систему Elasticsearch.

2.1.1 Scala

Scala — мова програмування, що поєднує в собі об'єктно-орієнтовану та функціональну парадигми програмування. Назва Scala утворена зі слів «scalable» (масштабовна) та «language» (мова), для того щоб задекларувати, що мова може рости разом з вимогами користувачів. Scala призначена для вираження загальних шаблонів програмування в стислий, елегантний і безпечний спосіб. Вона ідеально поєднує функції об'єктно-орієнтованих і функціональних мов.

Scala є чистою об'єктно-орієнтованою мовою в тому сенсі, що кожне значення є об'єктом. Типи та поведінка об'єктів описуються класами та ознаками (traits). Класи можна розширити за допомогою підкласів і використання гнучкого механізму композиції на основі mixin як більш зручної і безпечної заміни множинного успадкування.

Scala має властивості функціональної мови програмування, а саме – кожна функція є значенням. Синтаксис Scala дозволяє легко визначати анонімні функції, підтримує функції вищого порядку, підтримує вкладені функції, і навіть каррування. Scala має вбудовану підтримку алгебраїчних типів даних, які використовуються в більшості функціональних мов програмування (ця підтримка базується на механізмі зіставлення з прикладом, де як приклад виступають класи зразки). Є можливість створення об'єктів без об'явлення класу.

Крім того, концепція зіставлення з прикладом логічно переноситься на обробку XML-даних використовуючи як приклад регулярні вирази, з підтримкою функціоналу об'єктів екстракторів. Для більшої зручності обробки даних представлена схема формування запитів з використанням for-вираження. Такі можливості роблять Scala ідеальним рішенням для розробки програм типу веб-сервісів.

Найчастіше розробка застосунків для дуже специфічних областей вимагає

специфічних для цих галузей мовних можливостей чи окремих спеціалізованих мов програмування. Натомість Scala пропонує унікальні механізми, для легкої модифікації та розширення самої мови.

У багатьох випадках це можна зробити без використання засобів мета-програмування, таких як макроси. Наприклад:

- Неявні класи дозволяють додавати нові методи до існуючих.
- Інтерполяція рядків дозволяє додавати обробку рядків (розширюється розробником за допомогою інтерполяторів).

Scala повністю сумісний із популярним середовищем Java Runtime Environment (JRE). Взаємодія з основною об'єктно орієнтованою мовою програмування Java відбувається максимально гладко. Нові функції Java, такі як SAM, лямбди, анотації та дженерики, мають прями аналоги у Scala.

Ті функції Scala, які не мають аналогів у Java, такі як стандартні параметри та іменовані параметри, компілюються якомога ближче до Java. Scala має таку ж компіляційну модель (окрема компіляція, динамічне завантаження класів), як у Java, що дозволяє отримати доступ до тисяч вже існуючих високоякісних бібліотек.

2.1.2. Spark

Apache Spark – фреймворк з відкритим вихідним кодом для реалізації розподіленої обробки неструктурованих і слабоструктурованих даних, що входить до екосистеми проектів Hadoop. На відміну від класичного обробника з ядра Hadoop, що реалізує дворівневу концепцію MapReduce зі зберіганням проміжних даних на накопичувачах, Spark працює в парадигмі резидентних обчислень (in-memory computing) - обробляє дані в оперативній пам'яті, завдяки чому дозволяє отримувати значний вигаш для деяких класів завдань, зокрема, можливість багаторазового доступу до завантажених в пам'ять даних робить бібліотеку привабливою для алгоритмів машинного навчання.

Проект надає програмні інтерфейси для мов Java, Scala, Python, R. Спочатку написаний на Scala, згодом додана істотна частина коду Java для надання

можливості написання програм безпосередньо Java. Складається з ядра та кількох розширень, таких як Spark SQL (дозволяє виконувати SQL-запити над даними), Spark Streaming (надбудова для обробки поточкових даних), Spark MLlib (набір бібліотек машинного навчання), GraphX (призначений для розподіленої обробки графів). Може працювати як в середовищі кластера Hadoop під управлінням YARN, так і без компонентів ядра Hadoop, підтримує кілька розподілених систем зберігання - HDFS, OpenStack Swift, NoSQL-СУБД Cassandra, Amazon S3.

Ключовий автор — румунсько-канадський вчений в галузі інформатики Матей Захарія (англ. Matei Zaharia), розпочав роботу над проектом у 2009 році, будучи аспірантом Університету Каліфорнії в Берклі. У 2010 році проект опубліковано під ліцензією BSD, у 2013 році передано фонду Apache і переведено на ліцензію Apache 2.0, у 2014 році прийнято до проектів верхнього рівня Apache. 2022 року проект отримав щорічну премію SIGMOD у номінації «Системи».

2.1.3. Delta Lake

Delta Lake є сховищем з відкритим кодом, основною метою його розробки було підвищення надійності сховищ великих даних. Delta Lake підтримує транзакції ACID та масштабовану обробку метаданих, а також дозволяє об'єднати обробку поточкових даних та пакетну обробку. Delta Lake повністю сумісний з усіма API Apache Spark.

Зокрема, Delta Lake підтримує наведені нижче функції та можливості.

Транзакції ACID в Spark: рівні ізоляції, що серіалізуються, гарантують, що читачі не зіткнуться з неузгодженими даними.

Масштабована обробка метаданих: розподілена обчислювальна потужність Spark використовується для ефективної обробки всіх метаданих таблиць, обсяг яких може досягати петабайт, а кількість файлів - мільярдів штук.

Об'єднання потокової передачі та пакетної обробки: таблиця в Delta Lake - це таблиця пакетної обробки, а також джерело та приймач потокової передачі. Прийом поточкових даних, виконання пакетів заднім числом та інтерактивні запити

працюють без додаткового налаштування.

Застосування схеми: автоматична обробка варіантів схеми, щоб запобігти вставці неприпустимих записів під час прийому.

Перехід за часом: керування версіями даних дозволяє виконувати відкати, вести повноцінні історичні журнали аудиту та проводити відтворювані експерименти машинного навчання.

Операції `upsert` та видалення: підтримуються операції злиття, оновлення та видалення, що дозволяють реалізувати складні сценарії використання, такі як відстеження змінених даних, потокові операції `upsert` і т.д.

2.1.4 Kafka

Apache Kafka — це розподілена потокова платформа передачі повідомлень із відкритим кодом, яка використовується тисячами компаній для високопродуктивних конвеєрів даних, потокової аналітики, інтеграції даних і критично важливих програм. Написана мовами програмування Java та Scala.

Спроектвано як розподілену, горизонтально масштабовану систему, що забезпечує нарощування пропускнуєї спроможності як при зростанні числа та навантаження з боку джерел, так і кількості систем-підписників. Підписники можуть бути об'єднані у групи. Підтримується можливість тимчасового зберігання даних для подальшого пакетного оброблення. Однією з особливостей реалізації інструменту є застосування техніки, подібної до журналів транзакцій, які у системах управління базами даних. Спочатку розроблений компанією LinkedIn, вихідні коди проекту відкриті на початку 2011 року, включення до Apache Incubator відбулося 23 жовтня 2012 року.

Ключові можливості Kafka:

- Доставка повідомлень з обмеженою пропускнуєю здатністю мережі, використовуючи кластер машин із затримкою всього 2 мс.
- Масштабування виробничих кластерів до тисячі брокерів, трильйонів повідомлень на день, петабайтів даних, сотень тисяч розділів.

- Безпечне збереження потоків даних у розподіленому, довговічному, відмовостійкому кластері.
- Обробка потоків подій за допомогою об'єднань, агрегацій, фільтрів, перетворень тощо, використовуючи обробку подій і одноразову обробку.
- Готовий інтерфейс Connect від Kafka інтегрується з сотнями джерел подій і приймачів подій, включаючи Postgres, JMS, Elasticsearch, AWS S3 тощо.
- Читання, запис та обробляйте потоки подій у величезній кількості мов програмування.
- Велика екосистема інструментів з відкритим кодом.
- Підтримка критично важливих випадків використання з гарантованим упорядкуванням, нульовою втратою повідомлень і ефективною одноразовою обробкою.
- Багата документація, навчальні посібники, відео, зразки проектів, Stack Overflow тощо.

2.1.5 Cassandra

Cassandra — це безкоштовна система керування базами даних NoSQL із відкритим вихідним кодом, розподіленим сховищем із широкими стовпцями, призначена для обробки великих обсягів даних на багатьох серверах, забезпечуючи високу доступність без єдиної точки збою. Cassandra пропонує підтримку кластерів, що охоплюють кілька центрів обробки даних, з асинхронною реплікацією без керування, що дозволяє виконувати операції з низькою затримкою для всіх клієнтів. Cassandra була розроблена для реалізації комбінації розподіленого зберігання та реплікації Dynamo від Amazon у поєднанні з даними Google Bigtable та моделлю механізму зберігання. Промислові рішення на базі Cassandra розгорнуті для забезпечення сервісів таких компаній, як Cisco, IBM, Cloudkick, Reddit, Digg, Rackspace і Twitter.

Основні властивості:

- Розподілена Кожен вузол у кластері має однакову роль. Немає єдиної точки

відмови. Дані розподіляються по кластеру (тому кожен вузол містить різні дані), але немає головного, оскільки кожен вузол може обслуговувати будь-який запит.

- Підтримує реплікацію та реплікацію кількох центрів обробки даних. Стратегії реплікації можна налаштувати. Cassandra розроблена як розподілена система для розгортання великої кількості вузлів у кількох центрах обробки даних. Ключові особливості розподіленої архітектури Cassandra спеціально створені для розгортання кількох центрів обробки даних, резервування, відновлення після відмови та аварійного відновлення.

- Масштабованість. Розроблено таким чином, щоб пропускна здатність читання та запису зростала лінійно в міру додавання нових машин, щоб уникнути простоїв або перерв у роботі програм.

- Відмовостійкість. Дані автоматично копіюються на кілька вузлів для відмовостійкості. Підтримується реплікація в кількох центрах обробки даних. Вийшли з ладу вузли можна замінити без простою.

- Регульована консистенція. Cassandra зазвичай класифікується як система AP, що означає, що доступність і толерантність до розділів, як правило, вважаються важливішими за послідовність у Cassandra. Запис і читання пропонують настроюваний рівень узгодженості, від «запис ніколи не завершується» до "блок для читання всіх реплік", з рівнем кворуму посередині.

- Підтримка MapReduce. Cassandra має інтеграцію Hadoop із підтримкою MapReduce. Існує також підтримка Apache Pig і Apache Hive.

2.1.6 Elasticsearch

Elasticsearch — пошукова система, заснована на бібліотеці Lucene. Представляє собою розподілену систему повнотекстового пошуку з підтримкою кількох користувачів, документів JSON без схем, має веб-інтерфейс HTTP. Elasticsearch розроблено на Java. Офіційні клієнти доступні на Java, .NET (C#), PHP, Python, Ruby та багатьох інших мовах. Згідно з рейтингом DB-Engines, Elasticsearch є найпопулярнішою корпоративною пошуковою системою.

Elasticsearch можна використовувати для пошуку будь-яких документів. Він забезпечує масштабований пошук, має пошук майже в режимі реального часу та підтримує одночасний доступ кількох користувачів. Elasticsearch є розподіленим, що означає, що індекси можна розділити на сегменти, і кожен сегмент може мати нуль або більше реплік. Кожен вузол містить один або більше сегментів і діє як координатор для делегування операцій правильному сегменту. Перебалансування та маршрутизація виконуються автоматично. Пов'язані дані часто зберігаються в одному індексі, який складається з одного або кількох основних сегментів і нуля чи кількох реплік. Після створення індексу кількість первинних фрагментів не можна змінити. Elasticsearch має можливість розподілення, індекси можуть бути розділені на сегменти, при чому кожен сегмент може мати нуль чи більше реплік. Кожен вузол містить один чи більше сегментів і діє як координатор делегування операцій на потрібний сегмент. Балансування та маршрутизація виконується автоматично.

Приклади використання:

- Реалізація внутрішнього пошуку на вебсайті.
- Зберігання журналів подій чи транзакцій, аналізування і добування даних для отримання тенденцій, статистик, висновків, аномалій.
- Для впровадження аналітики/Бізнес-аналітики в проект, коли треба швидко досліджувати, аналізувати, візуалізувати надзвичайно великі об'єми даних (мільйони чи мільярди записів).

2.1.7 Airflow

Apache Airflow — це платформа керування робочим процесом із відкритим вихідним кодом для конвеєрів обробки даних. Розробку було розпочато в Airbnb у жовтні 2014 року як рішення для управління все більш складними робочими процесами компанії. Створення Airflow дозволило Airbnb програмно створювати та планувати свої робочі процеси та контролювати їх через вбудований інтерфейс користувача Airflow. З самого початку проект мав відкритий код, ставши проектом Apache Incubator у березні 2016 року та проектом верхнього рівня Apache Software

Foundation у січні 2019 року. Airflow написаний мовою Python, а робочі процеси створюються за допомогою сценаріїв Python. Airflow розроблено за принципом «конфігурація як код». Хоча інші платформи робочих процесів «конфігурація як код» існують із використанням мов розмітки, таких як XML, використання Python дозволяє розробникам імпортувати бібліотеки та класи, щоб допомогти їм створювати робочі процеси.

Airflow використовує напрямлені ациклічні графіки (DAG) для керування оркестрацією робочого процесу. Завдання та залежності визначаються з використанням коду Python, а потім Airflow керує плануванням і виконанням. DAG можна запускати за певним розкладом (наприклад, щогодини чи щодня) або на основі тригерів зовнішніх подій (наприклад, файл, що з'являється в файловій системі). Попередні планувальники на основі DAG, такі як Oozie та Azkaban, зазвичай покладалися на кілька файлів конфігурації та дерев файлової системи для створення DAG, тоді як у Airflow DAG часто можна було записати в одному файлі Python. Оператор з параметрами для його запуску описує задачу. Задачу можна виконувати багато разів, і кожне виконання називають екземпляром задачі. Екземпляр задачі може перебувати в різних станах: запланована, виконується, успішно виконана, невдача, невдача залежності, і т. д.

2.1.8 Docker

Docker — це набір продуктів «платформа як послуга» (PaaS), які використовують віртуалізацію на рівні ОС для доставки програмного забезпечення в пакетах, які називаються контейнерами. Сервіс має як безкоштовний, так і преміальний рівні. Програмне забезпечення, яке містить контейнери, називається Docker Engine. Вперше він був запусканий у 2013 році та розроблений Docker, Inc.

Docker може упакувати програму та її залежності у віртуальний контейнер, який можна запускати на будь-якому комп'ютері Linux, Windows або macOS. Це дає змогу програмі працювати в різних місцях, наприклад локально, у публічному або приватному хмарному сервісі. Під час роботи в Linux Docker використовує функції

ізоляції ресурсів ядра Linux (наприклад, контрольні групи та простори імен ядра) та файловою системою з можливістю об'єднання (наприклад, OverlayFS), щоб дозволити контейнерам працювати в одному екземплярі Linux, уникаючи накладні витрати на запуск і підтримку віртуальних машин. Docker на macOS використовує віртуальну машину Linux для запуску контейнерів.

Підтримка ядром Linux для просторів імен здебільшого ізолює погляд програми на операційне середовище, включаючи дерева процесів, мережу, ідентифікатори користувачів і змонтовані файлові системи, тоді як контрольні групи ядра забезпечують обмеження ресурсів для пам'яті та ЦП. Починаючи з версії 0.9, Docker включає власний компонент libcontainer для використання можливостей віртуалізації, наданих безпосередньо ядром Linux, на додаток до використання абстрактних інтерфейсів віртуалізації через libvirt, LXC і systemd-nsprawn.

Docker реалізує високорівневий API для забезпечення легких контейнерів, які запускають процеси в ізоляції. Контейнери Docker є стандартними процесами, тому можна використовувати функції ядра для моніторингу їх виконання, включаючи, наприклад, використання таких інструментів, як strace, для спостереження та втручання в системні виклики.

Основні інструменти Docker:

- Docker Compose — це інструмент для визначення та запуску багатоконтейнерних програм Docker. Він використовує файли YAML для налаштування служб програми та виконує процес створення та запуску всіх контейнерів за допомогою однієї команди. Утиліта CLI docker-compose дозволяє користувачам запускати команди для кількох контейнерів одночасно, наприклад, створювати зображення, масштабувати контейнери, запускати контейнери, які були зупинені, тощо. Команди, пов'язані з маніпулюванням зображеннями, або параметрами взаємодії з користувачем, не мають відношення до Docker Compose, оскільки вони звертаються до одного контейнера. Файл docker-compose.yml використовується для визначення служб програми та містить різні параметри конфігурації. Наприклад, опція збірки визначає параметри конфігурації, такі як шлях до файлу Docker, опція command дозволяє перевизначити стандартні команди

Docker тощо.

- Docker Swarm надає власні функції кластеризації для контейнерів Docker, які перетворюють групу механізмів Docker в єдиний віртуальний механізм Docker. У Docker 1.12 і вище режим Swarm інтегровано з Docker Engine. Утиліта `docker swarm` CLI дозволяє користувачам запускати контейнери Swarm, створювати маркери виявлення, перераховувати вузли в кластері тощо. Утиліта `docker node` CLI дозволяє користувачам запускати різні команди для керування вузлами в swarm групі, наприклад, перераховувати вузли в групі, оновлювати вузли та видаляти вузли з групи. Docker керує групами за допомогою консенсусного алгоритму Raft. Алгоритм передбачає, що оновлення буде виконано, якщо більшість вузлів Swarm погодяться на оновлення.

- Docker Volume полегшує незалежне збереження даних, дозволяючи їм залишатися навіть після видалення або повторного створення контейнера.

2.1.9 Apache Parquet

Apache Parquet — це безкоштовний формат зберігання даних із відкритим кодом, орієнтований на стовпці, в екосистемі Apache Hadoop. Він схожий на RCFile і ORC, інші формати файлів зі стовпцями зберігання в Hadoop, і сумісний із більшістю фреймворків обробки даних навколо Hadoop. Він забезпечує ефективне стиснення даних і схеми кодування з підвищеною продуктивністю для масової обробки складних даних.

Apache Parquet реалізовано за допомогою алгоритму подрібнення записів і складання, який враховує складні структури даних, які можна використовувати для зберігання даних. Значення в кожному стовпці зберігаються в безперервних розташуваннях пам'яті, що забезпечує такі переваги:

- Стиснення по стовпцях є ефективним для збереження дискового простору
- Можна використовувати методи кодування та стиснення, специфічні для типу даних у кожному стовпці
- Запити, які отримують певні значення стовпців, не потребують читання

всього рядка, що покращує продуктивність

Apache Parquet реалізовано за допомогою фреймворку Apache Thrift, що підвищує його гнучкість; він може працювати з кількома мовами програмування, такими як C++, Java, Python, PHP тощо.

Parquet підтримує інфраструктури обробки великих даних, зокрема Apache Hive, Apache Drill, Apache Impala, Apache Crunch, Apache Pig, Cascading, Presto та Apache Spark.

2.2 Аналіз сервісів оренди та продажу ОТ

Всесвітня рада бізнесу зі сталого розвитку визначає стійку мобільність як «здатність задовольняти потреби суспільства вільно пересуватися, отримувати доступ, спілкуватися, торгувати та встановлювати відносини без шкоди для інших основних людських або екологічних цінностей сьогодні та в майбутньому».

Соціальна мобільність суспільства, що означає здатність пересуватися в соціальних ієрархіях, часто прямо пов'язана з просторовою мобільністю. З огляду на те, що до 2030 року половина населення країн, що розвиваються, проживатиме в містах, створюються величезні нові вимоги до мобільності. Завдяки прагненню людей до доступних товарів, виняткових продуктів або передових технологій світова торгівля різко зростає, а регіональні ринки розширюються до глобальних масштабів.

Збільшення мобільності може створити економічне процвітання в усьому світі, позбавляючи мільйони людей від бідності та надаючи їм доступ до кращих медичних послуг, можливостей роботи та освіти, а також забезпечуючи швидке та надійне перевезення продуктів харчування та інших товарів. У той же час мобільність призводить до більшого споживання ресурсів (енергії, матеріалів тощо) і забруднення навколишнього середовища через збільшення викидів парникових газів. Соціальна напруга може виникнути через неминучий нерівний розподіл багатства. Але сьогоднішні перешкоди мультимодального транспорту вже помітні у вантажних перевезеннях. Застаріла технологія та низька ефективність, навіть у такій

промислово розвиненій країні, як Німеччина, призвели до одностороннього переходу від внутрішнього водного та залізничного транспорту до більш гнучкого, але також набагато більш ресурс затратного автомобільного транспорту. Адекватне використання засобів доступних інформаційно-комунікаційних технологій (ІКТ) у мережі постачальників послуг, їхнього обладнання та клієнта є важливим для надання конкурентоспроможних послуг.

ІКТ також можуть створити конкурентоспроможні нові форми колективного транспортування вантажів, наприклад, повідомлення в плящі, що забезпечує кожне домогосподарство щоденними товарами, такими як продукти харчування чи косметика. Але також нове використання поважних екологічно чистих засобів транспорту дає шанс підвищити їхню конкурентоспроможність. Потяги на магнітній подушці могли одночасно перевозити пасажирів і вантажі з досі неперевершеною швидкістю над землею і таким чином напряду конкурувати з літаками.

Щоб реалізувати бачення багатонаціональних мультимодальних перевезень пасажирів і вантажів, необхідно здійснити прагматичні зміни в інфраструктурі та ефективно перехресне зв'язування клієнтів, постачальників послуг, виробників, урядових і неурядових організацій у транспортних відносинах. Якщо це буде успішно, це може стати прикладом і дати зростання економіки менш розвинутих країн і країн, що розвиваються, мають шанс задовольнити потреби своїх громадян у більшій мобільності з меншим споживанням ресурсів. У той же час вони можуть отримати вигоду від знань, які вони повинні отримати, щоб розвивати, адаптувати та використовувати ці сучасні засоби транспорту та їхні ІКТ.

Одним з прикладів придатних транспортних засобів у цій мережі може бути велосипед із інтегрованою інформаційною системою супроводу продукту, так званою блоком життєвого циклу (БЖЦ). Наприклад, моніторинг положення велосипеда за допомогою глобальної системи позиціонування (GPS), передача його через глобальну систему мобільного зв'язку (GSM) на станцію керування та надання даних в Інтернеті може дозволити клієнту знайти ОТ поблизу його розташування. Це також може підвищити ефективність роботи ОТ для постачальника послуг, напр. шляхом полегшення технічного обслуговування на основі стану.

ОТ є дуже мобільним транспортним засобом, і його можна легко адаптувати. ОТ оснащені БЖЦ, сприяють більшій продуктивності використання та мінімальному негативному впливу на навколишнє середовище, наприклад, забруднення.

2.3 Послуга «Виклик ОТ»

ОТ є індивідуальним і дуже популярним засобом пересування, який використовується в усьому світі для пересування, спорту тощо. Популярність ОТ зростає в історії, і зі збільшенням населення виробництво ОТ загалом зросло.

ОТ мобільний, адаптивний до змін, надійний і простий в обслуговуванні. Це також зберігає здоров'я людей. Порівняно з іншими транспортними засобами на вулицях, ОТ є транспортним засобом із повільним розгоном. У великих містах, де динамічне перемикання транспортного засобу від зупинки до руху є одним із найважливіших факторів безпечнішого та ефективнішого транспортування, ОТ є потенційно небезпечним транспортним засобом. Безпеку водіїв ОТ підвищують спеціальні смуги руху для ОТ. Кількість жертв на дорогах мінімальна (табл. 2.1).

Переваги ОТ, такі як мобільність, роблять його дуже корисним транспортним засобом, але все ж продуктивність використання ОТ низька. Нові підходи та реалізації ОТ повинні збільшити продуктивність використання.

Також цікаво те, що збільшення продуктивності використання велосипедів може зменшити виробництво велосипедів і зменшити використання доступних ресурсів

Таблиця 2.1 – Використання та безпека їзди на велосипеді в окремих країнах

Країна	Велосипедні кілометри на людину за день	Вбиті велосипедисти на 100 мільйонів кілометрів
Україна	0.01	1.7
Великобританія	0.1	6.0
Італія	0.2	11.0

Австрія	0.4	6.8
Норвегія	0.4	3.0
Швейцарія	0.5	3.7
Фінляндія	0.7	5.0
Німеччина	0.8	3.6
Швеція	0.9	1.8
Данія	1.7	2.3
Нідерланди	3.0	1.6

Так, наприклад, служба прокату велосипедів доступна у великих містах Німеччини. Це відрізняється від інших служб прокату велосипедів, оскільки велосипеди розташовані не в кількох центральних центрах міста. З такою послугою велосипеди доступні у багатьох, помітних місцях міста: на вулицях, площах, автостанціях тощо та прив'язані електронним замком до нерухомих об'єктів. Люди можуть користуватися велосипедами, лише якщо вони зареєстровані в сервісі та мають відповідний персональний ідентифікаційний номер (PIN), наданий службою прокату. Для залучення велосипеда (щойно знайденого в місті) користувач запитує пароль від замка велосипеда в Головному центрі управління (MCC - Main Control Center) служби найму за допомогою системи коротких повідомлень (SMS). Користувач також надсилає свій PIN-код і номер велосипеда до MCC. MCC перевіряє PIN-код, номер велосипеда та наявність велосипеда та надсилає пароль для замка велосипеда. З цього моменту велосипед взято напрокат.

Користувач має право взяти велосипед напрокат стільки, скільки йому потрібно. Коли він/вона вирішує припинити прийом на роботу, користувач вмикає блокування, і в інтерфейсі користувача (UI) відображається новий пароль. Користувач повинен надіслати SMS до MCC із своїм PIN-кодом, номером велосипеда та новим паролем з інтерфейсу користувача. MCC перевіряє ці дані, припиняє прокат велосипеда та надсилає SMS користувачеві з підтвердженням про закінчення прокату та ціну прокату. Велосипеди та MCC не мають зв'язку.

Інформація про велосипеди, доступна для МСС, стосується лише статусу велосипеда (доступний чи ні). МСС не має даних про положення та стан велосипеда. Велосипед може бути несправним, але МСС не буде поінформовано про це, доки мобільні ремонтні служби оренди не виявлять його та не повідомлять про несправність.

3 РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ ОРЕНДИ ОСОБИСТОГО ТРАНСПОРТУ

3.1 Обґрунтування вибору архітектури системи оренди ОТ

Архітектура програмного забезпечення відноситься до фундаментальних структур програмної системи та дисципліни створення таких структур та систем. Кожна структура включає елементи програмного забезпечення, відносини з-поміж них, і навіть властивості як елементів, і відносин. Архітектура програмної системи – це метафора, аналогічна архітектурі будівлі. Вона функціонує як план для системи та проекту розробки, в якому викладаються завдання, які мають бути виконані командами розробників.

Чим старшим стає проект, тим більшим є технічний борг, навіть якщо функціональність проекту не сильно змінюється, то технічний борг все одно накопичується з часом і в якийсь критичний момент виправлення дефектів займатиме занадто багато ресурсів, а на нову функціональність часу просто не вистачатиме.

Найголовніше, необхідно зібрати правильні вимоги до архітектури, що враховують усі складності розробки, ці вимоги можуть бути абсолютно різними, ось лише деякі з них:

- відмовостійкість (надійність);
- швидкість розробки;
- простота деплою;
- масштабованість за кількістю людей у проекті;
- масштабованість за кількістю запитів на секунду;
- швидкий вхід/вихід у розробку;
- швидкість розгортання програми;
- позбавитися старого коду;
- складність нових інтеграцій.

3.1.1 Монолітна архітектура

При монолітній архітектурі програма знаходиться в одному репозиторії і по суті є окремим, самостійним проектом, який можна запусити.

Плюси моноліту:

- Проста розробка та розгортання програми.
- Менше проблем із мережею та роботою між серверами.
- Краще продуктивність.
- Легко зробити прототип.

Але моноліт має і мінуси:

- Масштабованість.

Насправді масштабованість це головна та чи не єдина проблема монолітів. І Мається на увазі масштабованість як у плані продуктивності (скільки користувачів може одночасно користуватися додатком), так і в плані розміру команди, адже чим більше проект, тим складніше зберегти хорошу якість коду та прибрати взаємозв'язки між різними частинами проекту. Зараз це важливо як ніколи, особливо коли сервіси перетворилися на гігантів.

3.1.2 Сервісна архітектура

Мікросервісна архітектура має на увазі що ми розбиваємо наш додаток на кілька невеликих, незалежних додатків які в теорії можуть працювати самостійно, але все ж таки таке не завжди виходить, але в ідеальному світі при відмові одного сервісу, це не повинно сильно вплинути систему і всі інші сервіси повинні продовжити роботу без перебоїв.

Складності мікросервісної архітектури:

- Складне розгортання програми (потрібно розгорнути декілька сервісів та баз даних та налаштувати їх взаємодію.
- Отримуємо новий шар на кшталт мережі (багато хто не враховує це при проектуванні архітектури, але у великих системах саме рівень мережі може бути

"пляшковою шийкою", в яке впирається система при масштабуванні)

- При великій кількості мікросервісів навантаження на мережу сповільнює всю систему.
- Складна горизонтальна масштабованість (потрібно правильно налаштувати взаємодію всіх сервісів).
- Збір огів від декількох сервісів складно реалізувати та відстежувати.
- Тестувати мікросервісну архітектуру складніше.
- Конфігурація розподіленого сховища для сервісів .
- Контроль доступу (також потрібно відстежувати кожен сервіс і намагатися зрозуміти, який сервіс має доступ у відкритий світ, який працює тільки з іншими мікросервісами).

Як правило мікросервісну архітектуру обирають заради масштабованості. Правильно побудована мікросервісна архітектура здатна масштабуватись майже без обмежень.

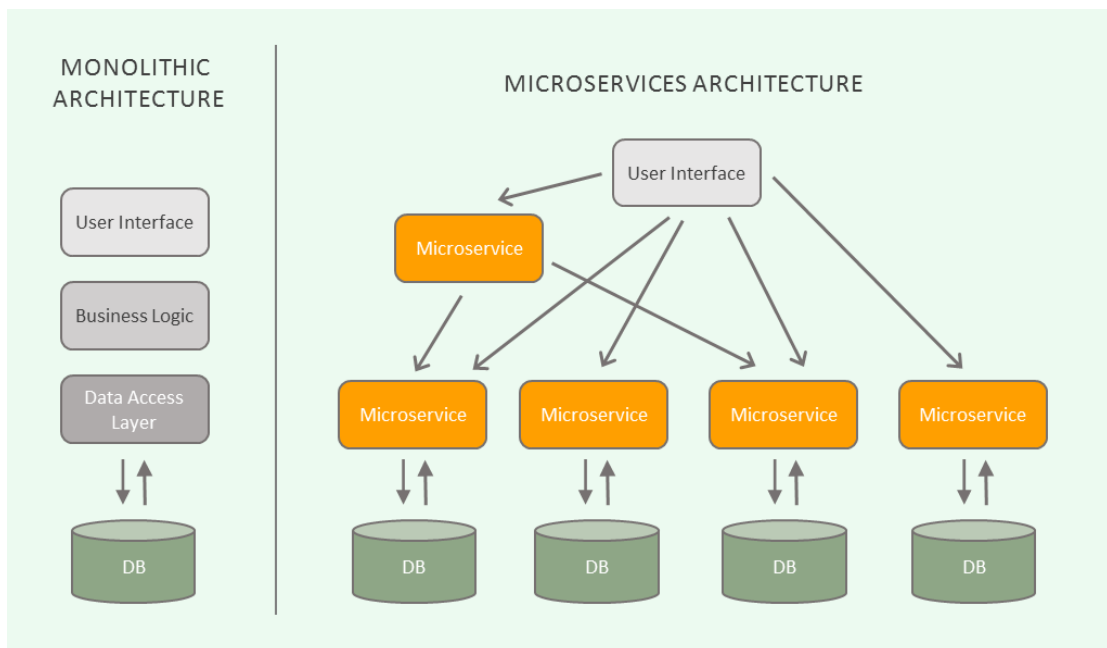


Рисунок 3.1 – Порівняння архітектур

На рисунку 3.1 у вигляді схеми продемонстровано відмінність систем монолітної та сервісної архітектур. Можна побачити, що мікросервісна архітектура

складається з декількох програм. Ці програми спілкуються за допомогою мережі, кожна з них може бути запущена на окремій машині, або навіть сервері, це дозволяє нарощувати потужність для кожного з компонентів системи за необхідності.

3.2 Розробка архітектури

Після проведення аналізу технологій та можливих архітектур, було прийняте рішення щодо використання сервісної архітектури. Такий підхід надасть майбутній системі гнучкості. Це дозволить розділити систему на підпрограми, що виконують одну задачу. Кожний з блоків можна буде сконфігурувати окремо. Також масштабувати систему побудовану на мікросервісах набагато легше, що важливо для пристосування до змін в кількості даних.

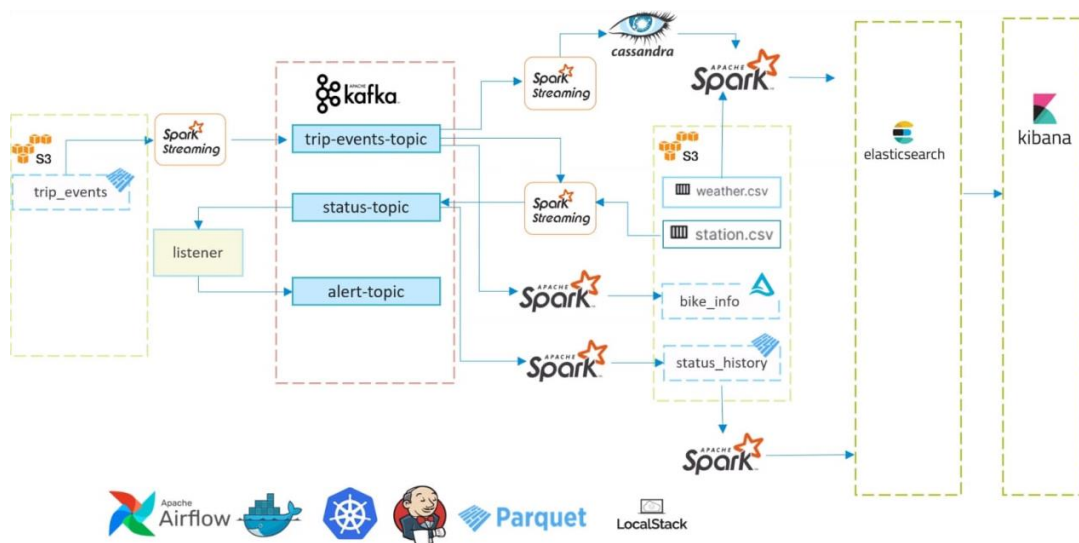


Рисунок 3.2 – Архітектурна схема

Уся система буде поділена на модулі, кожний з яких виконує ту чи іншу операцію (читання та запис даних, агрегації, потокова передача, візуалізація). Для розробки буде створено багатомодульний проект. Для того щоб спростити керування залежностями, автоматизувати компіляцію, та збірку кожного модуля буде використано систему зборки для платформи Java – maven. Ця програма

безкоштовна і має відкритий код.

3.2 Формалізація процесу обслуговування особистого транспорту

Простіша реалізація надання послуг, таких як «Виклик ОТ», може бути досягнута за допомогою БЖЦ.

Загальні завдання БЖЦ полягають у отриманні, обробці та передачі відповідних даних про продукт і процес протягом усього життєвого циклу продукту. Інтегрований у ОТ БЖЦ має завдання оцінювати, прогнозувати, діагностувати та контролювати велосипед та його компоненти. Структурна схема БЖЦ показана на рисунку 3.3.

БЖЦ отримує відповідні дані від користувача, датчиків та інших компонентів, вбудованих у ОТ, і від МСС. Дані, надані користувачем, є його PIN-кодом, необхідним для ідентифікації користувача. Дані про стан і положення ОТ надходять від датчиків та інших компонентів. Датчики дають інформацію про тиск повітря в шинах, швидкість ОТ та пробіг тощо.

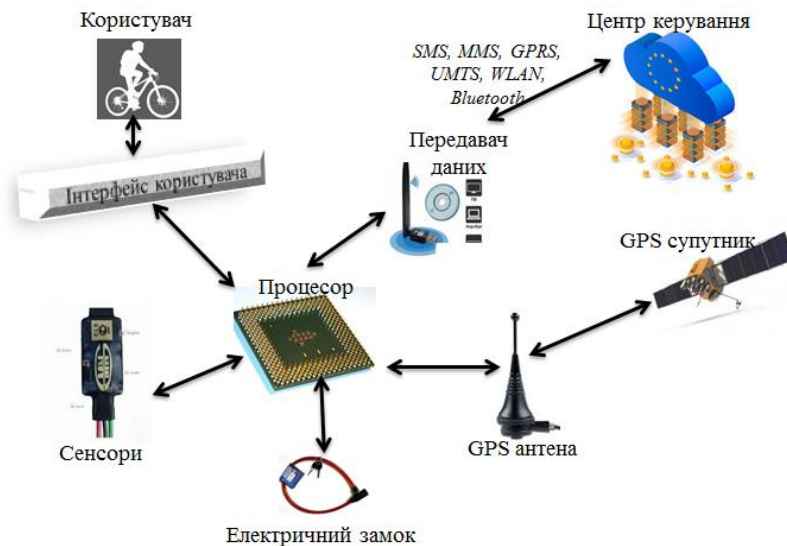


Рисунок 3.3 – Структурна схема БЖЦ особистого транспорту

Положення ОТ визначається антеною глобальної системи позиціонування (GPS). MCC надає ОТ інформацію та інструкції (від БЖЦ) щодо користувача та використання ОТ, наприклад, заблокувати чи розблокувати, скільки з користувача стягуватиметься за використання тощо.

БЖЦ обробляє отримані дані, щоб надати інформацію про оцінку ефективності ОТ, діагностику та прогноз (рис. 3.4). Ця інформація передається до MCC через SMS, службу мультимедійних повідомлень (MMS), загальну пакетну радіопередачу (GPRS), універсальну мобільну телекомунікаційну систему (UMTS), Bluetooth або бездротову локальну мережу (WLAN).

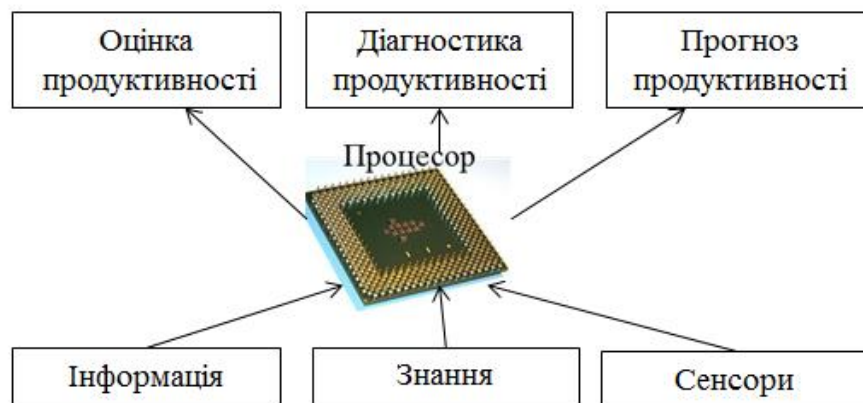


Рисунок 3.4 – Функціональна схема блоку життєвого циклу послуги «Виклик ОТ»

Частина обробленої інформації доступна для користувача. Це інформація про статус ОТ (робочий чи ні), місце розташування ОТ, вартість прокату, пробіг тощо (рис. 3.5). Ця інформація надається користувачеві за допомогою інтерфейсу користувача.

Потужність для руху ОТ забезпечує сам водій ОТ. Для функціонування інших компонентів, що реалізовані на ОТ (освітлення, деякі датчики) БЖЦ потребує електричної енергії. Електричну енергію виробляє динамо, встановлене на ОТ.

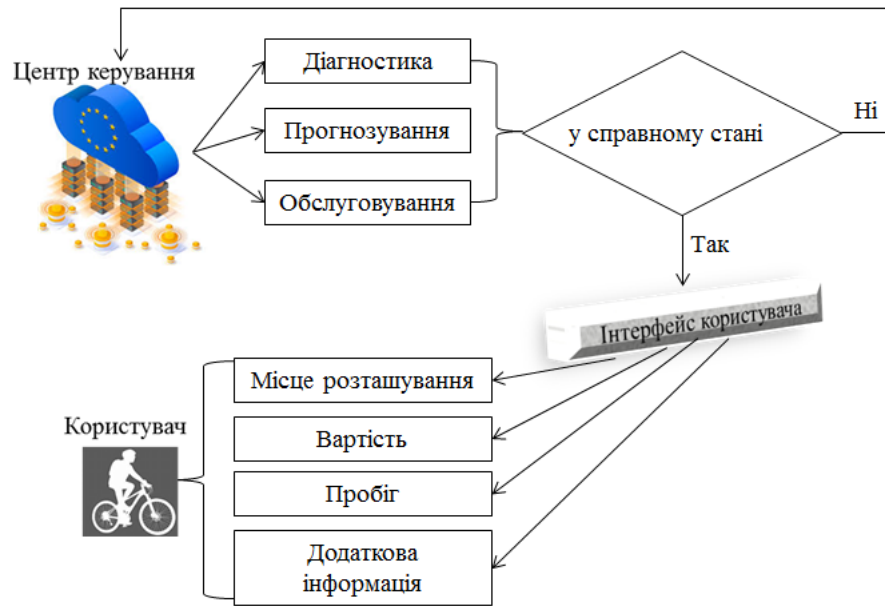


Рисунок 3.5 – Інформація, яка надана БЖЦ

Динамо перетворює частину механічної енергії, яку надає водій ОТ, в електричну. Енергія від динамо використовується для компонентів, а достатня кількість енергії накопичується в батареях. Оскільки динамо є додатковим навантаженням для водія ОТ, БЖЦ вмикає динамо лише тоді, коли йому не потрібна повна потужність для руху, наприклад, їзда з гірки.

4 ЕКСПЕРЕМЕНТАЛЬНЕ МОДЕЛЮВАННЯ

4.1 Специфікація вимог до онлайн-системи оренди ОТ

Кожен тип особистого транспорту має різну вартість оренди на день. Вартість оренди залежить від кількості днів, марки та швидкості руху ОТ. Система повинна мати такі функції:

Оренда: система, обладнана для відповідей на запити *Клієнта* щодо наявності та вартості оренди різних «*Типів*» ОТ на певні дати в майбутньому. Коли клієнт приймає рішення щодо «*Типу*» ОТ та дат, система повинна мати можливість «*Зарезервувати*» або «*Визначити*» тип ОТ для запиту на запитані дати. Клієнту слід надати «номер підтвердження»

Підбирання: система обробляє підборку ОТ. Клієнт заходить і надає підтвердження: номер або ім'я. Система має отримати всю інформацію про бронювання цього клієнта. Потім клієнта просять надати водійські права.

Повернення: система запускає повернення. Система повинна зафіксувати дату та час повернення, і в залежності від цих параметрів розрахувати остаточну суму оренди.

Основна мета полягає в такому: розробити веб-систему, яка дозволить клієнтам онлайн реєструвати та бронювати автомобіль, а компанії – ефективно керувати бізнесом з оренди автомобілів.

Особистий транспорт - це транспортний засіб, яким можна користуватися тимчасово протягом певного періоду за певну плату. Оренда автомобіля допомагає людям пересуватися, навіть якщо вони не мають доступу до свого особистого транспортного засобу або взагалі не є власником транспортного засобу. Особа, яка хоче орендувати автомобіль, повинна спочатку зв'язатися компанією з прокату ОТ. Це можна зробити онлайн. Особа має надати певну інформацію, наприклад; дату оренди та тип ОТ. Особа, яка орендує автомобіль, повинна пред'явити дійсне посвідчення водія.

4.2 Функції користувачів в онлайн-системі

Адміністратор:

- Адміністратор може увійти в систему.
- Перевіряє базу даних автомобіля.
- Створює цінову стратегію.
- Керує платіжною системою.
- Завершує замовлення.
- Скасовує замовлення.

Співробітник:

- Оновлює базу даних.
- Надає клієнту інформацію про автомобіль.
- Надає альтернативи.
- Підтримує контакти.

Менеджер з технічного обслуговування:

- Перевіряє технічне обслуговування.
- Віддає на утримання.
- Надає інформацію адміністратору.
- Оновлює бази даних.

Замовник:

- Клієнт може увійти в систему.
- Відвідає веб-сайт.
- Оформлює замовлення.
- Скасовує замовлення.

Системні вимоги:

Сервер (мінімальні вимоги):

Processor: Intel® Xeon® processor 3500 series HDD: Minimum 500GB Disk Space
RAM: Minimum 16GB OS: Windows 8.1, Linux 6 Database: SQL Server 2014 (SQL14)
Application: XAAMP, phpmyadmin.

Клієнт (мінімальні вимоги): Processor: Intel Dual Core HDD: Minimum 80GB

Disk Space RAM: Minimum 1GB OS: Windows 10, Linux

Обмеження проектування та впровадження

– Програма використовуватиме php, Ajax, javascript, jQuery та css як основні веб-технології.

– Як протоколи зв'язку використовуються протоколи HTTP і FTP. FTP використовується для завантаження веб-програми в реальному домені, і клієнт може отримати доступ до неї через протокол HTTP.

– Кілька типів перевірок роблять цю веб-програму захищеною

– Оскільки система прокату автомобілів є веб-додатком, потрібне підключення до Інтернету

– Система оренди автомобілів використовуватиметься на ПК та працюватиме через Інтернет або інтранет з будь-яким веб-браузер.

4.3 Спеціальні вимоги

Інтерфейс користувача:

– Усі користувачі побачать ту саму сторінку, коли зайдуть на цей веб-сайт. Ця сторінка запитує ім'я користувача та пароль.

– Після автентифікації за допомогою правильного імені користувача та пароля користувач буде перенаправлений на відповідний профіль, де вони можуть займатися різними видами діяльності.

– Інтерфейс користувача має бути простим і узгодженим із загальною термінологією, які зрозумілі користувачам системи. Система матиме простий інтерфейс, сумісний із стандартним інтерфейсом, щоб уникнути необхідності нечастого навчання користувачів.

Апаратні інтерфейси:

– Не потрібні додаткові апаратні інтерфейси.

– Система використовуватиме стандартне апаратне забезпечення та ресурси передачі даних.

– Це включає, але не обмежується, загальне мережеве підключення на

сервері/хостингу сайта, мережевий сервер і засоби керування мережею.

Інтерфейси програми:

ОС: Windows 7, Linux

Веб-браузер: система є веб-додатком; клієнтам потрібен сучасний веб-браузер, наприклад Mozilla Firefox, Internet Explorer, Opera або Chrome. Комп'ютер повинен мати підключення до Інтернету для доступу до системи.

Інтерфейси зв'язку:

– Ця система використовує комунікаційні ресурси, включаючи, але не обмежуючись, HTTP протоколом зв'язку з веб-браузером і веб-сервером і TCP/IP мережевий протокол HTTP.

– Ця програма зв'язуватиметься з базою даних, у якій зберігаються всі бронювання інформації. Користувачі можуть зв'язуватися з сервером через протокол HTTP за допомогою функція, яка називається HTTP Service. Ця функція дозволяє програмі використовувати дані, отримані сервером для виконання запиту користувача.

Функціональні вимоги:

Це послуги, які система повинна надавати, як система повинна реагувати на конкретні вхідні дані та те, як система повинна поводитися в конкретних ситуаціях. Це визначає функціональні можливості програми, які повинні вбудувати в продукт, щоб надати користувачам можливість виконувати свої завдання.

Бронювання:

– Система повинна дозволяти клієнту зареєструватися для бронювання.

– Система повинна дозволяти клієнту переглядати детальний опис конкретного ОТ.

– Система повинна повідомляти про вибір недоступних автомобілів під час бронювання.

– Система має надати опцію розширеного пошуку, щоб обмежити пошук ОТ конкретної категорії пошуку.

– Система повинна дозволяти клієнтам вибирати конкретний ОТ за допомогою іншого категорія під час бронювання.

- Система повинна переглядати список доступних ОТ під час бронювання.
- Система повинна дозволяти клієнтам скасувати бронювання за допомогою бронювання.

Номер підтвердження.

- Система повинна дозволяти працівнику оновлювати інформацію про бронювання.
- Система повинна дозволяти працівнику переглядати бронювання, зроблені клієнтами.
- Система повинна представляти інформацію про засоби захисту та їх щоденну вартість, і просить клієнта прийняти або відхилити умови регулювання під час бронювання.
- Система повинна мати можливість надавати унікальний номер підтвердження бронювання для всіх успішно здійснених бронювань.
- Система повинна мати можливість відображати підсумок резервування для успішно здійснених бронювань.

Ввід:

- Система повинна дозволяти менеджеру входити в систему за допомогою свого імені користувача та пароля.
- Система повинна дозволяти працівнику входити в систему за допомогою свого імені користувача та пароля.
- Система повинна дозволяти менеджеру створювати новий обліковий запис користувача.
- Система повинна дозволяти менеджеру змінювати пароль облікового запису.
- Система повинна дозволяти персоналу змінювати пароль облікового запису.
- Система повинна дозволяти персоналу виходити з системи.
- Система повинна дозволяти менеджеру вийти з системи.

Особистий транспорт:

- Система повинна дозволяти персоналу реєструвати нові ОТ.
- Система повинна дозволяти персоналу вибирати ОТ зі списку.
- Система повинна дозволяти клієнту вибирати ОТ зі списку.

- Система повинна дозволяти персоналу шукати ОТ за певним записом.
 - Система повинна дозволяти персоналу клієнта шукати ОТ за певним записом.
 - Система повинна дозволяти персоналу оновлювати інформацію про ОТ, який потребує модифікації.
 - Система повинна дозволяти персоналу відображати всі списки ОТ.
 - Система повинна дозволяти персоналу відображати всі наявні ОТ.
 - Система повинна дозволяти клієнту відображати всі наявні ОТ.
 - Система повинна дозволяти персоналу відображати всі орендовані ОТ.
 - Система повинна дозволяти персоналу відображати всі ОТ, які не працюють.
- Оренда:
- Система повинна дозволяти персоналу реєструвати клієнтів у списку оренди.
 - Система повинна дозволяти персоналу оновлювати інформацію про орендну плату клієнта в списку орендної плати.
 - Система повинна мати можливість зберігати всі зміни, зроблені в списку оренди клієнта.
 - Система повинна дозволяти персоналу вибирати записи про оренду клієнта за конкретною категорією пошуку.
 - Система повинна дозволяти персоналу шукати дані про орендну плату клієнтів, використовуючи специфіку категорії.
 - Система повинна дозволяти персоналу відображати клієнтів, які орендують ОТ.
 - Система повинна дозволяти персоналу відображати всі записи орендної плати клієнтів.
 - Система повинна надавати роздрукований підсумок для успішної оренди.

4.4 Нефункціональні вимоги

Нефункціональні вимоги не стосується конкретних послуг, які надає система своїм користувачам. Вони можуть стосуватися таких властивостей системи, як

надійність, час відгуку та зайнятість магазину.

Крім того, вони можуть визначати обмеження на реалізацію системи, такі як можливості пристроїв введення/виведення або представлення даних, що використовуються в інтерфейсах з іншими системами.

Нефункціональні вимоги, такі як продуктивність, безпека або доступність, зазвичай визначають або обмежують характеристики системи в цілому.

Зручність використання. Система надає меню довідки та підтримки в усіх інтерфейсах, з якими користувач може взаємодіяти з системою. Користувач може користуватися системою, прочитавши довідку та підтримку.

Безпека. Система надає ім'я користувача та пароль, щоб запобігти несанкціонованому доступу до системи.. Підсистема повинна забезпечувати високий рівень безпеки та цілісності даних, що зберігаються системою.

Продуктивність: Час відповіді системи на кожну інструкцію, виконану користувачем, не повинен перевищувати щонайменше 10 секунд, 50 секунд для дуже складних завдань і 20-25 секунд для менш складних завдань.

Наявність: Система має бути завжди доступною для доступу 24 години, 7 днів на тиждень. Також в у разі будь-якої серйозної несправності системи, система має бути доступною через 1 до 2 робочих днів, щоб бізнес-процес не постраждав.

Обробка помилок:

Помилка має бути значно зведена до мінімуму та має бути надано відповідне повідомлення про помилку, яке допоможе користувачу відновити помилку

Перевірка користувацьких даних має велике значення. Крім того, стандартний час відновлення після помилки має складати від 15 до 20 секунд.

Простота використання. Вважається рівень знань, якими володіють користувачі цієї системи, простий, але якісний інтерфейс користувача має бути розроблений таким чином, щоб зробити його легким для розуміння та потребувати менше навчання.

На рисунках 4.1 – 4.6 наведено діаграми діяльності для оформлення замовлення.

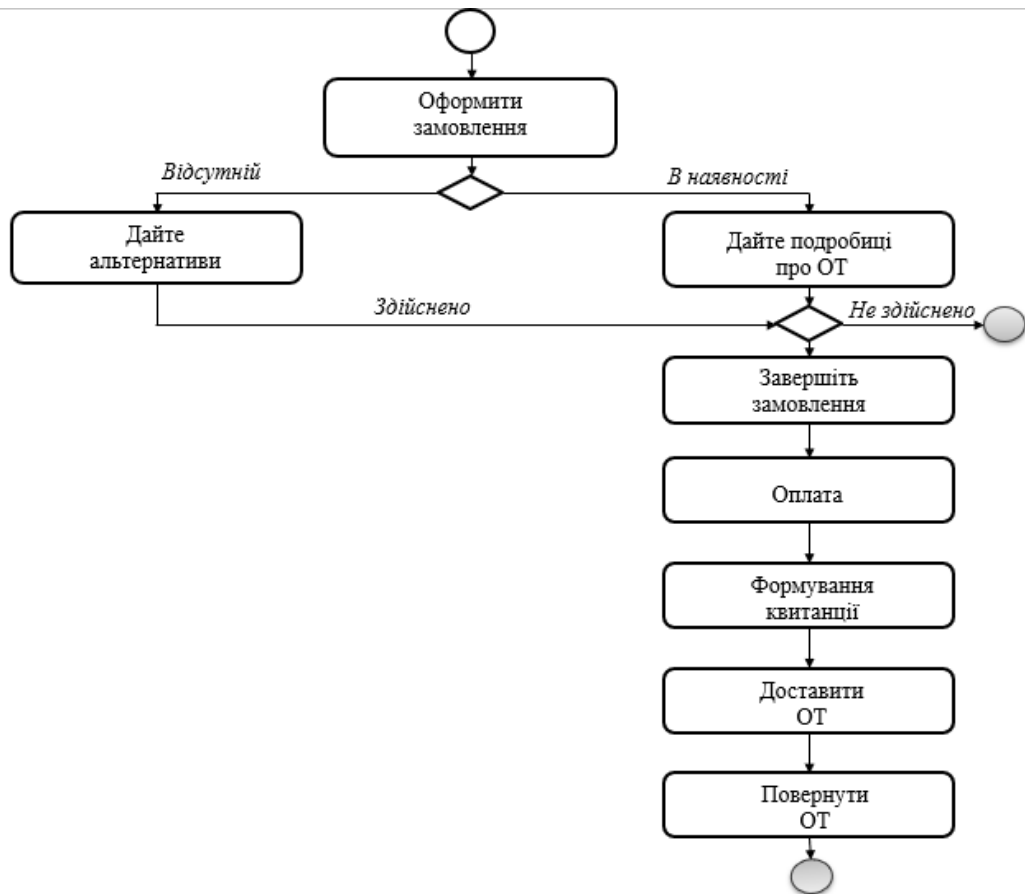


Рисунок 4.1 - Діаграма діяльності системи

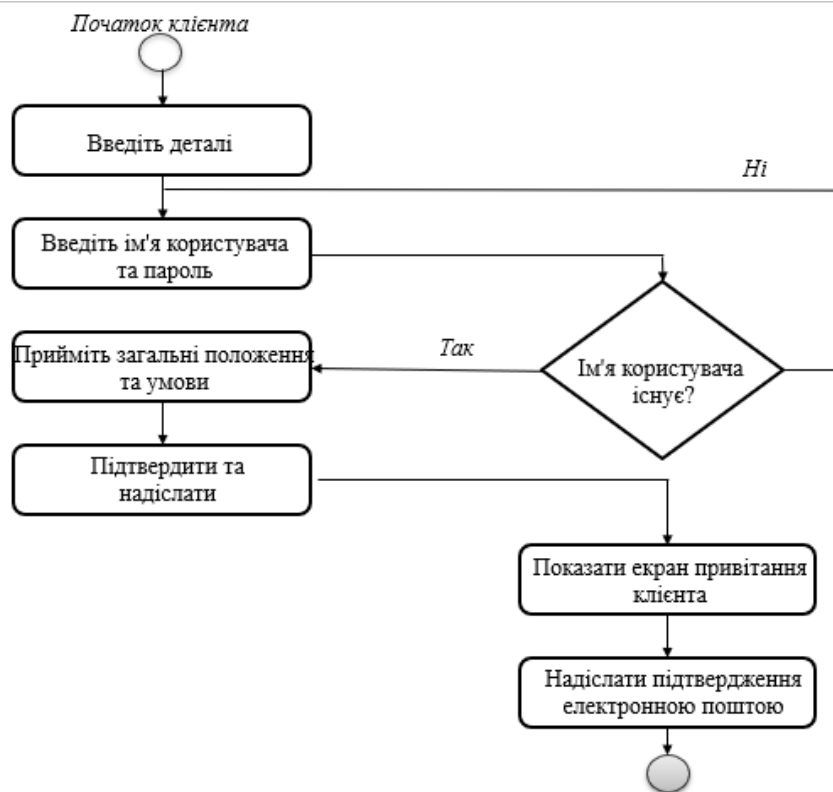


Рисунок 4.2 - Реєстрація учасників

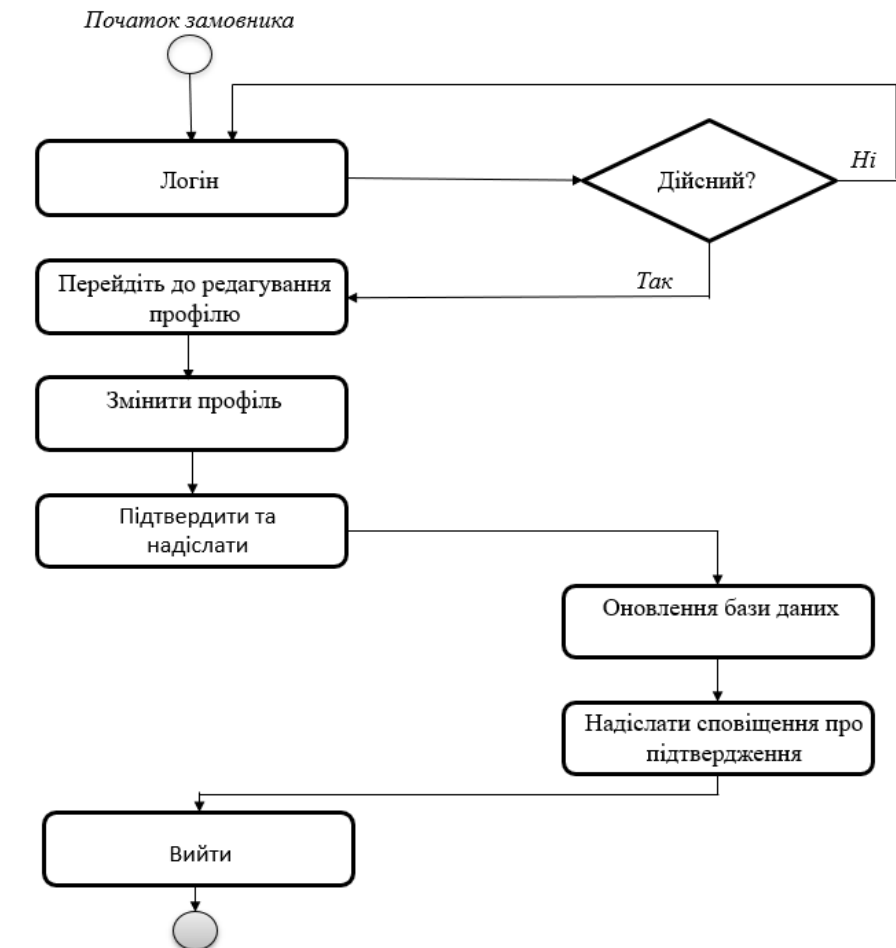


Рисунок 4.3 - Модифікація профілю



Рисунок 4.4 - Відгуки клієнтів

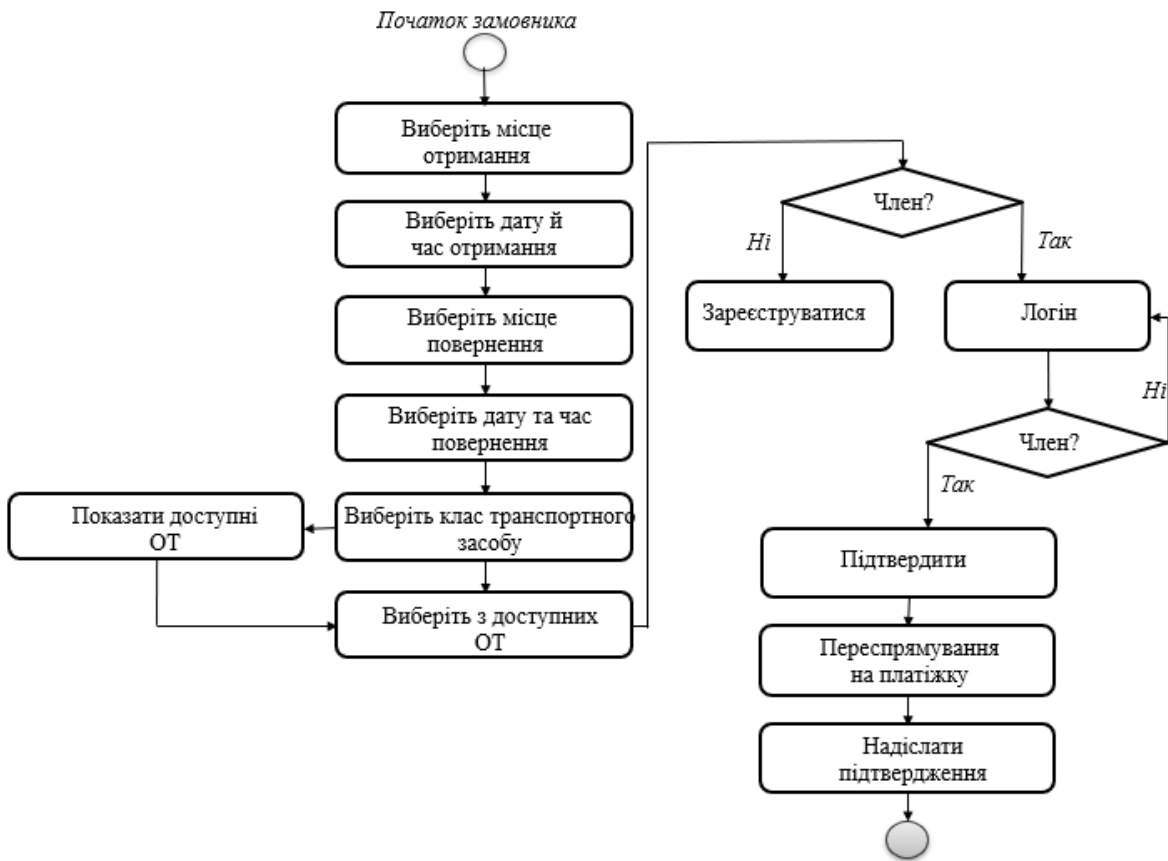


Рисунок 4.5 - Бронювання ОТ

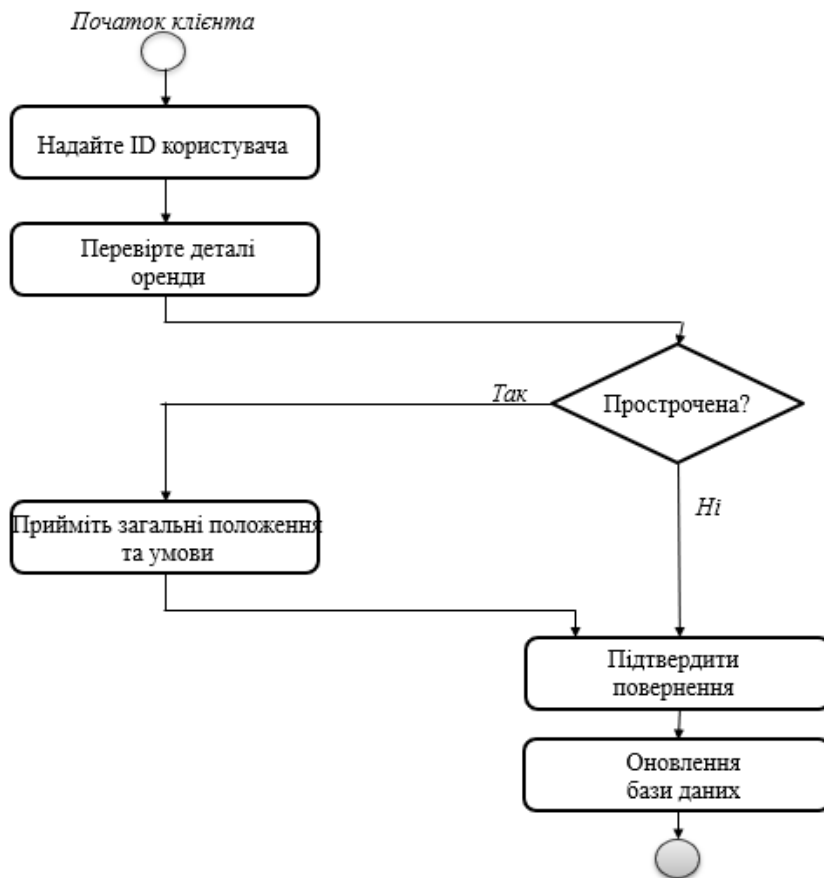


Рисунок 4.6 - Оплата оренди ОТ

Опис актора та варіантів використання показує детальний опис взаємодії між акторами та їхніми варіантами використання. Опис дозволяє мати правильне розуміння того, як актор взаємодіє з системою через їхні випадки використання (рис. 4.7).

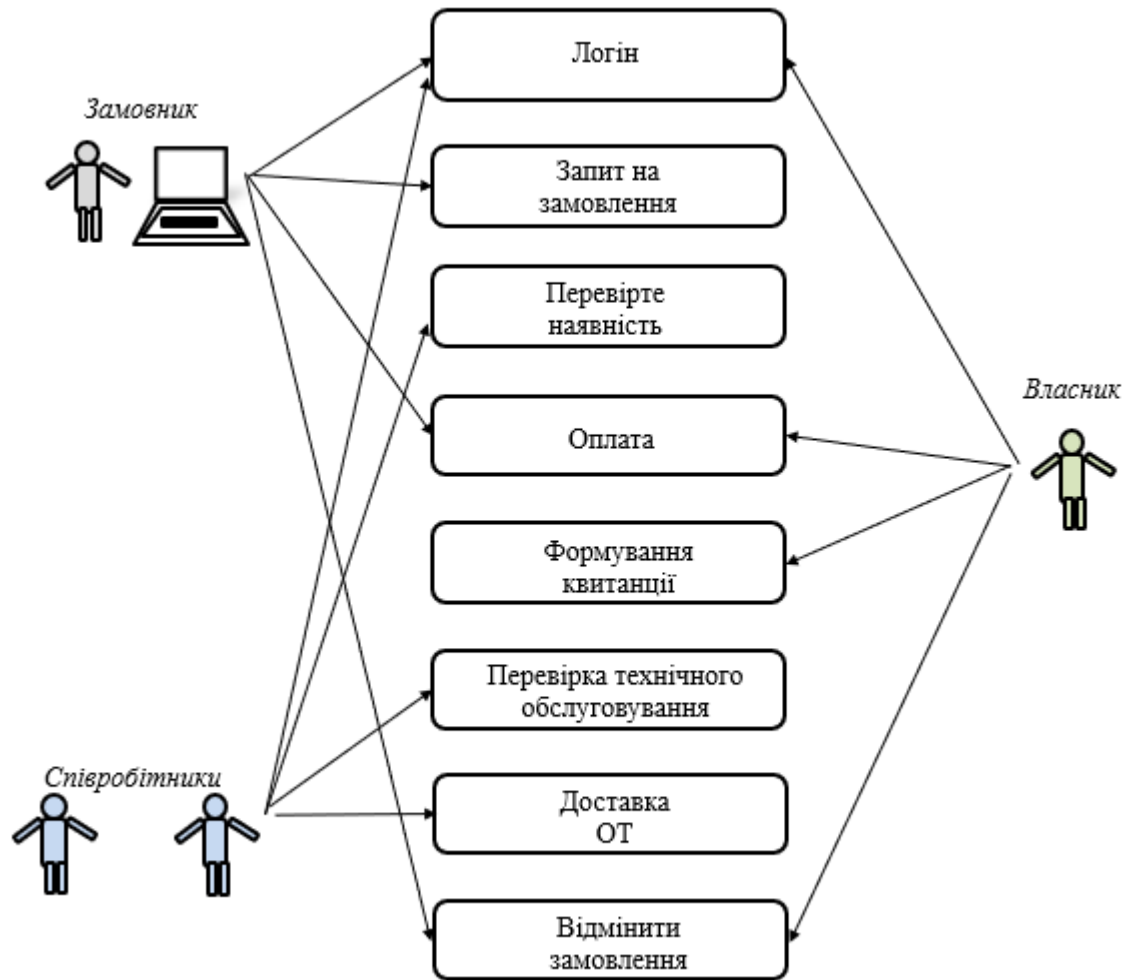


Рисунок 4.7 - Діаграма варіантів використання

Замовник має зареєструватися як учасник. Цей варіант використання описує дії клієнта, щоб зареєструватися онлайн і стати учасником.

Деталі клієнта необхідні під час реєстрації. Дані для входу автоматично надсилаються на адресу клієнта після успішної реєстрації.

Зробити бронювання визначає, що цей варіант використання дозволяє клієнту шукати та бронювати. Незареєстрованого клієнта буде направлено зареєструватися до підтвердження бронювання. Сповіщення автоматично надсилається на адресу

клієнта після виконання завдання.

Повернення автомобіля визначає, що цей варіант використання описує подію, коли клієнт повертає позичений автомобіль, варіант використання розширюється сценарій використання «оренди процесу» від штатного актора.

Надати відгук визначає, що цей варіант використання використовується клієнтом для надання відгуків/коментарів компанії; підтвердження сповіщення буде надіслано клієнту після того, як буде подано відгук.

Переваги послуг онлайн-прокату ОТ

- Це рішення онлайн-прокату ОТ є повністю функціональним і гнучким.
- Він дуже простий у використанні.
- Ця онлайн-система оренди ОТ допомагає в управлінні бек-офісами, оптимізуючи та стандартизація процедур.
- Це економить багато часу, грошей і праці.
- Екологічність: стає моніторинг діяльності транспортного засобу та бізнесу в цілому легко та включає мінімум паперової роботи.
- Додаток діє як офіс, який працює 24/7.
- Підвищує ефективність менеджменту при пропонуванні якісних послуг клієнтів.
- Забезпечує розробку та підтримку спеціальних функцій із програмою.

Раніше будь-яка діяльність, що пов'язана з орендою автомобілів, обмежувалася лише фізичним розташуванням. Незважаючи на те, що фізичне розташування повністю не знищено; природа функцій і те, як ці функції досягаються, були змінені завдяки можливостям Інтернету. Сьогодні клієнти можуть забронювати, орендувати та отримати ОТ онлайн, коли клієнт стане зареєстрованим учасником.

ВИСНОВКИ

У ході магістерської підготовки був проведений аналіз літератури, присвяченої темам організації бізнесу, транспортної інфраструктури, Big Data технологіям, а також розглянута концепція «Мобільність як послуга».

Визначено, що використання засобів інформаційно-комунікаційних технологій у мережі постачальників послуг та їхнього обладнання є важливим для надання конкурентоспроможних послуг.

В результаті розроблено інтелектуальну систему оренди особистого транспорту, яка здатна надавати послуги завдяки можливостям Інтернету.

Запропоновано формалізацію процесу обслуговування ОТ, яка складається із структурної та функціональної схем блоку життєвого циклу послуги «Виклик ОТ».

Розроблено специфікацію вимог до онлайн-системи оренди ОТ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jittrapirom P. et al. *Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges.* – 2017. – С. 15-45
2. Rasouli S., Timmermans H. Applications of theories and models of choice and decision-making under conditions of uncertainty in travel behavior research. *Travel Behaviour and Society.* – 2014. – Т. 1. – №. 3. – С. 79-90.
3. Sochor J., Strömberg H., Karlsson I. C. M. A. Implementing mobility as a service: challenges in integrating user, commercial, and societal perspectives *Transportation research record.* – 2015. – Т. 2536. – №. 1. – С. 1-9..
4. Sochor J. L., Strömberg H., Karlsson M. A. An innovative mobility service to facilitate changes in travel behavior and mode choice. *22nd World Congress on Intelligent Transportation Systems, Bordeaux, October 5-9, 2015.* – 2015.
5. Kamargianni M. et al. A critical review of new mobility services for urban transport. *Transportation Research Procedia.* – 2016. – Т. 14. – С. 3294-3303.
6. Storme T. et al. Impact assessments of new mobility services: *A critical review.* *Sustainability.* – 2021. – Т. 13. – №. 6. – С. 3074.
7. Calderón F., Miller E. J. A literature review of mobility services: definitions, modelling state-of-the-art, and key considerations for a conceptual modelling framework. *Transport Reviews.* – 2020. – Т. 40. – №. 3. – С. 312-332.
8. Overview - Spark 3.3.1 Documentation – Режим доступа: www/ URL: https://spark.apache.org/docs/latest/ – Загл. з екрану.
9. Welcome to the Delta Lake documentation — Delta Lake Documentation – Режим доступа: www/ URL: https://docs.delta.io/latest/index.html – Загл. з екрану.
10. Apache Kafka – Режим доступа: www/ URL: https://kafka.apache.org/documentation/ – Загл. з екрану.
11. Welcome to Apache Cassandra's documentation! | Apache Cassandra Documentation – Режим доступа: www/ URL: https://cassandra.apache.org/doc/latest/ – Загл. з екрану.

12. Welcome to Elastic Docs | Elastic [Електронний ресурс] – Режим доступу: [www/ URL: https://www.elastic.co/guide/index.html](https://www.elastic.co/guide/index.html) – Загл. з екрану.
13. Documentation | Apache Airflow [Електронний ресурс] – Режим доступу: [www/ URL: https://airflow.apache.org/docs/](https://airflow.apache.org/docs/) – Загл. з екрану.
14. Introduction | Tour of Scala | Scala Documentation [Електронний ресурс] – Режим доступу: [www/ URL: https://docs.scala-lang.org/tour/tour-of-scala.html](https://docs.scala-lang.org/tour/tour-of-scala.html) – Загл. з екрану.
15. SQL | SQL Syntax [Електронний ресурс] – Режим доступу: [www/ URL: https://www.sqltutorial.org/sql-syntax/](https://www.sqltutorial.org/sql-syntax/) – Загл. з екрану.
16. JSON | Introducing JSON [Електронний ресурс] – Режим доступу: [www/ URL: https://www.json.org/json-en.html](https://www.json.org/json-en.html) – Загл. з екрану.
17. Н. Аксак , І. Бова. Огляд Big Data та аналітики у медицині та охороні здоров'я середовища. *Традиції та нові наукові стратегії у Центральній та Східній Європі*: Матеріали V Міжнародної науково-практичної конференції (м. Київ, 24–25 червня 2022 р.) / ГО «Інститут інноваційної освіти»; Науково-навчальний центр прикладної інформатики НАН України. – Київ : ГО «Інститут інноваційної освіти», 2022. -С. 125 -127.
18. Аксак Н.Г. Дистанційний моніторинг пацієнта на основі концепції агент-група-роль. *Системні дослідження та інформаційні технології*. – 2018. – № 3. – С. 7–18.
19. Axak N. Development of multi-agent system of neural network diagnostics and remote monitoring of patient. *Eastern-European Journal of Enterprise Technologies*. – 2016. – № 4/9 (82) – Р. 4–11.
20. Аксак Н.Г. Мультиагентная система нейросетевой диагностики и удаленного мониторинга пациента. *Інформаційні технології: проблеми та перспективи*: монографія за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г., 2017. – С. 325–340.
21. Аксак Н.Г. Система віддаленого нейромережного діагностування меланому/ *Інформаційні системи та технології в медицині (ISM-2018)*. Збірник наукових праць. ХНУРЕ.– Харків: «Друкарня Мадрид», 2018.- 116–118.