

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства»

Студента 2м курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Цьомка Олександра
Олександровича

Науковий керівник
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Рзаєва Світлана
Леонідівна

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Криворучко Олена
Володимирівна

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ТЕОРЕТИЧНІ ЗАСАДИ СУПУТНИКОВОГО МОНІТОРИНГУ АВТОТРАНСПОРТУ ЛОГІСТИЧНОГО ПІДПРИЄМСТВА.....	5
1.1. Особливості функціонування логістичного підприємства	5
1.2. Система супутникового моніторингу.....	7
1.3. Висновки до розділу 1.....	9
РОЗДІЛ 2 ПРИНЦИПИ ВЗАЄМОДІЇ АПАРАТНОЇ ЧАСТИНИ GPS СИСТЕМИ	10
2.1. Огляд та налаштування пристрою BITREK.....	10
2.2. Протокол формування даних «Codec 8»	14
2.3. Архітектура REST	17
2.5. Висновки до розділу 2.....	19
РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ.....	20
3.1. Проектування бази даних.....	20
3.2. Розробка серверної частини ІС з використанням Spring Boot	21
3.3. Проектування клієнтської частини ІС з використанням Vue	25
3.4. Візуалізація двовимірної мапи за допомогою Leaflet	27
3.5. Висновки до розділу 3.....	30
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32
ТЕХНІЧНЕ ЗАВДАННЯ	33

					КНТЕУ 121-06-21.МР		
Зм.	Арк.	№ докум.	Підпис	Дата			
Зав. Каф.	Криворучко О.В.		19.10.20	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Керівник	Рзаєва С.Л.		19.10.20		Зміст	2	34
Гарант	Криворучко О.В.		19.10.20		Зміст Факультет інформаційних технологій, 2м курс, 6 група		
Розробив	Цьомка О.О.		19.10.20				

ВСТУП

У транспортному та логістичному бізнесі технологія GPS або GPS сьогодні простягається далі, ніж просто інструмент, який допомагає водіям орієнтуватися на місцевості.

Технологія GPS повністю змінила транспортну галузь. Розробники програмного забезпечення можуть надати вам рішення для відстеження GPS, що включають ефективні для бізнесу інструменти, необхідні вашому бізнесу і потребам клієнтів.

За допомогою комп'ютеру або смартфона ви можете легко визначити розміщення транспортних засобів, ефективно планувати призначення водіїв та керувати витратами. Ці функції ідеально підходять для ведення успішного транспортного або логістичного бізнесу, не роблячи великої кількості дзвінків і не переглядаючи громіздкі звіти для прийняття рішень.

Якщо ви керуєте бізнесом, який має певний парк транспортних засобів, важливо мати інформацію про місцезнаходження будь-якого даного транспортного засобу в будь-який час.

Використання GPS моніторингу дозволяє вам бачити, який автомобіль знаходиться в русі і в якому напрямку. Також ви можете відстежувати швидкість автомобіля, маршрути, запуск і вимикання двигуна, холостий хід і маршрути. Ви можете сказати, чи йде автомобіль до призначеного місця призначення, чи були взяті маршрути і коли автомобіль простоює.

Він також стане в нагоді при пошуку викраденого автомобіля. Такий контроль над парком транспортних засобів є ключовим елементом успіху в транспортно-логістичному бізнесі.

					КНТЕУ 121-06-21.МР			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В.		24.01.20		В	3	34
Керівник		Рзаєва С.Л.		24.01.20				
Гарант		Криворучко О.В.		24.01.20				
Розробив		Цьомка О.О.		24.01.20				
					Вступ	Факультет інформаційних технологій, 2м курс, 6 група		

Мета: аналіз роботи пристроїв супутникового моніторингу та розробка програмного забезпечення для роботи з ними.

Об'єкт дослідження: моніторинг та контроль автотранспорту логістичного підприємства.

Предмет дослідження: розробка системи супутникового моніторингу транспортних засобів.

Завдання:

- опис особливостей функціонування логістичних підприємств;
- аналіз та налаштування пристрою для GPS моніторингу;
- розробка БД моніторингу та контролю автотранспорту;
- аналіз та розробка декодера протоколу «Codec 8»
- розробка серверної частини мовою програмування Java;
- розробка клієнтської частини що має візуалізацію мапи з відмітками об'єктів у вигляді маячку.

					КНТЕУ 121-06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ СУПУТНИКОВОГО МОНІТОРИНГУ АВТОТРАНСПОРТУ ЛОГІСТИЧНОГО ПІДПРИЄМСТВА

1.1. Особливості функціонування логістичного підприємства

Логістична компанія - це підприємство, що надає послуги з транспортування, обробки та зберігання вантажів, сприяючи своїм клієнтам в процесі просування товарів від виробника до споживача.

Логістика, як наука, вперше з'явилася в армії - в інтендантській службі. Оскільки конкурентна боротьба в сучасному бізнесі по гостроті своїй нагадує військові дії, досягнення цієї науки швидко були сприйняті комерсантами.

Сьогодні існує кілька основних розділів логістики:

- закупівельний;
- розподільний;
- збутовий;
- транспортний;
- логістика запасів;
- складський;
- інформаційний;
- комплексний.

Логістична компанія може стати вигідним партнером для тих підприємств, які:

- планують розширення ринків збуту з регіонального на національний або міжнародний рівень;
- хочуть знизити собівартість продукції за рахунок зниження витрат на її доставку;

					КНТЕУ 121 06-21.МР			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В.		25.06.20		P1	5	34
Керівник		Рзаєва С.Л.		25.06.20				
Гарант		Криворучко О.В.		25.06.20				
Розробив		Цьомка О.О.		25.06.20				
					Розділ 1 Теоретичні засади супутникового моніторингу автотранспорту логістичного підприємства	Факультет інформаційних технологій, 2м курс, 6 група		

- хочуть полегшити собі діяльність з проведення експортно-імпортних операцій, митного оформлення;
- яких не задовольняє робота внутрішніх логістичних служб, дорого
- обходиться їх утримання;
- здійснюють перевезення вантажів періодично, не регулярно, разово;
- потребують перевезення небезпечних, великогабаритних вантажів або вантажів, що вимагають особливих умов транспортування;
- перевозять вантажі дрібними партіями і потребують формування збірних партій вантажу.

Послуги, які може надати логістична компанія:

- обробка вантажів (формування партій товару, навантаження, вивантаження, сортування, фасування, упаковка, маркування і так далі);
- перевезення вантажів на будь-які відстані, в будь-які країни, різними видами транспорту;
- відповідальне зберігання товарів на власному складі;
- митне оформлення вантажів;
- послуги дистрибуції товарів;
- страхування вантажів у страхових компаніях;
- надання різних інформаційних послуг;

Такі компанії можуть надавати окремі види послуг, такі, як доставка вантажів, або повний комплекс послуг починаючи від отримання товарів у виробника, закінчуючи постачанням їх кінцевому споживачеві, включаючи оформлення всіх необхідних документів для проведення логістичних операцій.

Транспортна компанія на час проведення робіт за договором з підрядником бере на себе повну відповідальність за вантаж, за забезпечення його збереження за кількістю і якістю, а також зобов'язується надавати клієнту повну інформацію про етапи транспортування, про місцезнаходження, кількість і стан вантажу.

Для початку роботи з логістичною компанією необхідно укласти договір, в якому детально вказуються всі види послуг, що надаються, порядок оформлення

					КНТЕУ 121 06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		6

документів, право власності на вантаж, ступінь відповідальності сторін, порядок оплати за надані послуги та інші умови співпраці.

Розцінки на роботи, що виконуються транспортними компаніями, залежать від кількості і типу послуг, що надаються, дальності відстані перевезення і виду транспорту, яким вони здійснюються, кількості і виду вантажу, особливостей митного оформлення та інших умов. Слід зауважити, що комплексні послуги логістики та послуги, що надаються за багаторічними контрактами обходяться значно дешевше, ніж разові.

1.2. Система супутникового моніторингу

Основна ідея систем GPS полягає у визначенні місця розташування певного об'єкта в режимі реального часу. Системи GPS стеження використовують мережу глобальної навігаційної супутникової системи (GNSS). Кожен з цих супутників на орбіті посилає сигнали на приймач. Потім приймач використовує ці сигнали, щоб обчислити точне місце розташування, за допомогою 4 супутників і відстані до кожного з них, щоб оцінити 4 значення: широту, довготу, висоту над рівнем моря і час. Положення і відстань до супутника визначає положення приймача. Точне положення супутника багаторазово передається на всі приймачі в прямої видимості і зазвичай вимагає відкритого неба. Відстань до супутника визначається точним часом надходження супутникової передачі на приймач. Час, що генерується атомними годинами кожного супутника, передається супутником через задані проміжки часу.

Пристрої GPS моніторингу працюють за тим же принципом, але замість відображення інформації на самому пристрої, мікропроцесор на пристрої вираховує його місце розташування і передає отримані дані на сервер через Інтернет, використовуючи всесвітню мережу стільникового зв'язку GSM. На цьому сервері розміщена платформа, на якій кінцеві користувачі можуть отримати доступ і переглянути поточний і минулий шлях пристрої. Потім всі ці дані передаються

					КНТЕУ 121 06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		7

через Інтернет і відображаються на пристрої кінцевого користувача за допомогою комп'ютера або на смартфоні з допомогою спеціального додатку.

При виборі GPS-трекера потрібно вирішити, слід використовувати активний трекер в режимі реального часу або пасивний трекер. Пасивні трекери не дозволяють користувачам стежити за кожним рухом об'єкта що відслідковується. Спочатку інформація повинна бути отримана, а потім завантажена на комп'ютер. Такі типи трекерів зазвичай більш доступні, так як не вимагають щомісячної плати.

Активні трекери обробляють і доставляють дані в режимі реального часу, використовуючи SIM-карту і GSM-приймач. Диспетчери отримують інформацію з сервера в режимі реального часу, що дозволяє їм уважно стежити за своїми активами або цінними речами.

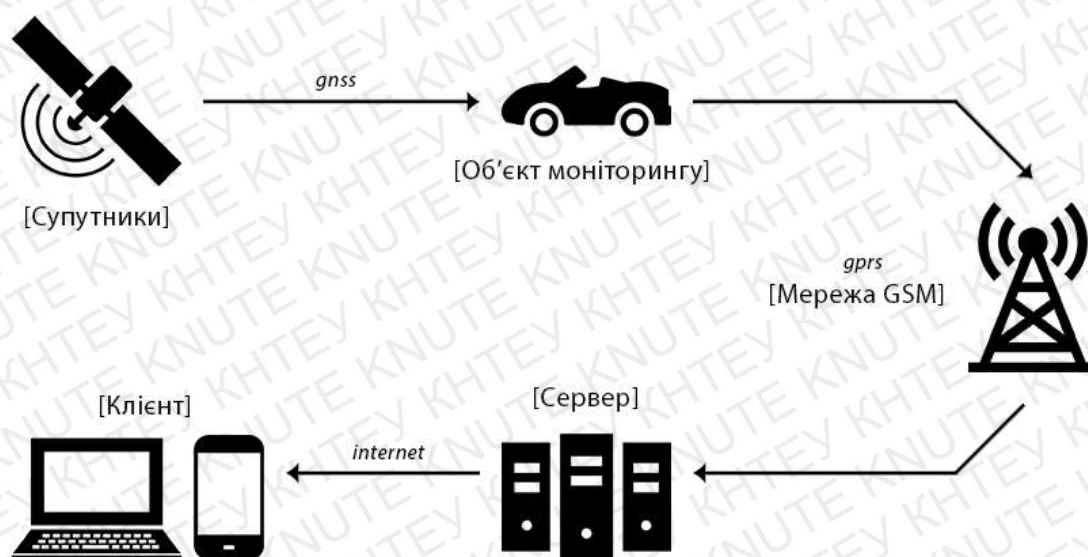


Рис. 1.1. Схема роботи моніторингу що складається з активних трекерів.

У разі, якщо сигнал GPS супутників заблокований, перебуваючи наприклад в приміщенні, де сигнал GPS супутників не доступний, пристрій шукає сигнали стільникової вежі і намагається оцінити приблизне місце розташування.

GPS трекер визначає, де приблизно знаходиться пристрій оцінюючи відстань між вежами стільникового зв'язку. Точність до 100 метрів можна отримати в міських районах, де щільність розташування веж стільникового зв'язку досить

висока. У сільських і малонаселених районах, де на квадратний кілометр припадає менше веж, між базовими станціями можуть спостерігатися кілометри, тому місце розташування визначається неточно, часто показуючи відстань з відхиленням у кілометрах від реального місцезнаходження пристрою.

Зазвичай GPS трекер складається з мікропроцесору, модулю GSM та GPS кожен з яких має свою антену, що можуть бути як внутрішніми так і зовнішніми та батарею. Більш складні пристрої мають модулі пам'яті для тимчасового зберігання даних, шини для підключення до живлення та датчиків що передбачені виробником автотранспорту (наприклад через CAN шину) або не заводських датчиків, якими обладнано транспортний засіб. Наприклад, одними із найважливіших датчиків являються датчик палива, датчик напруги бортової мережі та тощо

1.3. Висновки до розділу 1

В першій частині першого розділу було розглянуто особливості функціонування логістичного підприємства. Описано як саме з'явилася логістика, скільки основних розділів вона налічує на сьогодні, при яких обставинах логістична компанія може стати вигідним партнером для бізнесу та які послуги вона надає.

В другій частині першого розділу було розглянуто принцип роботи системи супутникового моніторингу, описано що таке GPS – трекер, визначено різницю понять активний та пасивний трекер, та чому ці пристрої є обов'язковою частиною моніторингу транспортних засобів.

					КНТЕУ 121 06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		9

ПРИНЦИПИ ВЗАЄМОДІЇ АПАРАТНОЇ ЧАСТИНИ GPS СИСТЕМИ

2.1. Огляд та налаштування пристрою BITREK

На момент проектування та розробки ІС було обрано GPS трекер українського виробництва ВІТРЕК 910 (Рис 2.1 ВІ 910 TREK).



Рис. 2.1. BI 910 TREK

Основні характеристики пристрою:

- Стандарт передачі даних: GSM 900/1800.
- Наявність зовнішніх антен GPRS та GSM.
- Наявність акселерометру.
- Кількість SIM карток: 2.
- Вбудований акумулятор: 1 000 мА.
- Об'єм пам'яті: 2МБ.
- Клас захисту корпусу: IP65.

					КНТЕУ 121 06-21.МР			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В.		07.09.20		Р2	10	34
Керівник		Рзаєва С.Л.		07.09.20				
Гарант		Криворучко О.В.		07.09.20				
Розробив		Цюмка О.О.		07.09.20	Розділ 2 Принципи взаємодії апаратної частини GPS системи	Факультет інформаційних технологій, 2м курс, 6 група		

- Робоча температура: від -30 °С до +80 °С.
- Габарити: 95 x 125 x 36 мм.
- Вага: 100 г.

При відсутності живлення вбудований акумулятор дозволяє пристрою працювати до 12 годин. При втраті з'єднання з мобільною мережею пристрій записує та зберігає 65 тис. записів (декілька тижнів), та автоматично передає накопичені за цей час дані на сервер при першому можливому встановленні з'єднання з мобільною мережею.

Для роботи в GSM мережі, в пристрій повинна бути встановлена як мінімум одна SIM картка. Друга являється необов'язковою та може бути використана як альтернативна (в місцях де відсутнє покриття першої картки, або для роумінгу в інших країнах). Для встановлення карток потрібно відключити від пристрою усі антени та живлення, після чого зняти кришку що закріплена на двох гвинтах (Рис 2.2. Внутрішній вигляд плати пристрою).



Рис 2.2. Внутрішній вигляд плати пристрою

Перед налаштуванням пристрою потрібно підключити антени GPS та GSM у відповідні роз'єми типу SMA. Після чого виконати підключення з'єднувального кабелю до транспортного засобу. В роботі розглянуто лише обов'язкові для роботи пристрою та реалізації моніторингу контакти з'єднувального кабелю.

Таблиця 2.1.

Значення контактів з'єднувального кабелю

№	Цвет	Найменування	Тип сигналу	Призначення контакту
A8	Червоний	+ Vin	Живлення	«+» Бортового живлення (номінальна напруга 12 В або 24 В)
C1	Чорний	GND	Живлення	Загальна маса
C8	Чорний	GND	Живлення	Загальна маса

Існує два способи налаштування пристрою: за допомогою програматора USB-UART та спеціального ПЗ; за допомогою SMS повідомлень з командами, що відправляються на номер SIM карти пристрою. В даній роботі розглянуто налаштування за допомогою SMS команд. Будь яке SMS повідомлення з командами може містити у собі дані для автентифікації при умові, що встановлені параметри логіну та паролю. Існує другий спосіб ідентифікації: пристрій буде ігнорувати усі повідомлення з командами, якщо мобільний номер не збігається з номерами заздалегідь сформованого списку авторизованих номерів. У випадку, якщо список авторизованих номерів порожній, то повідомлення з командами будуть прийматися з будь якого мобільного номеру. SMS повідомлення з командами має наступний вигляд: `<логін> <пароль> <команда1>;<команда2>;<командаN>;` Перелік команд що використовуються в роботі наведений в таблиці 2.2.

Таблиця 2.2.

Перелік доступних команд

Команда	Опис	Наявність відповіді
getstatus	Інформація про стан GSM модулю	Так
cpureset	Перезавантажити пристрій	Ні
getparam	Отримати значення параметру по ідентифікатору	Так
setparam	Задати значення параметру по ідентифікатору	Так
flush	Запит комунікаційних параметрів	Так

Приклади відповіді на команду getstatus:

- Data Link: 1 GPRS: 1 IP: 11.111.111.111 OP: 25501 GSM: 4 Roaming: 0
- Data Link: 0 GPRS: 0 GSM: 2 Roaming: 0, де Data Link – поточний стан з'єднання (0 – підключення до серверу відсутнє, 1 – підключення активне); GPRS – підключення до мережі; IP – адреса пристрою, що видана пристрою; OP – (MCC+MNC) код мобільної мережі оператора SIM картки (на момент виконання роботи, код оператора 25501 належить ПрАТ «ВФ Україна»).

Приклад відповіді на команду flush наведено на Рис 2.3. Значення відповіді розділено комами: перше – imei пристрою, наступне – назва APN, логін доступу GPRS, пароль доступу GPRS, IP адреса серверу, порт серверу, режим роботи (завжди дорівнює нулю).

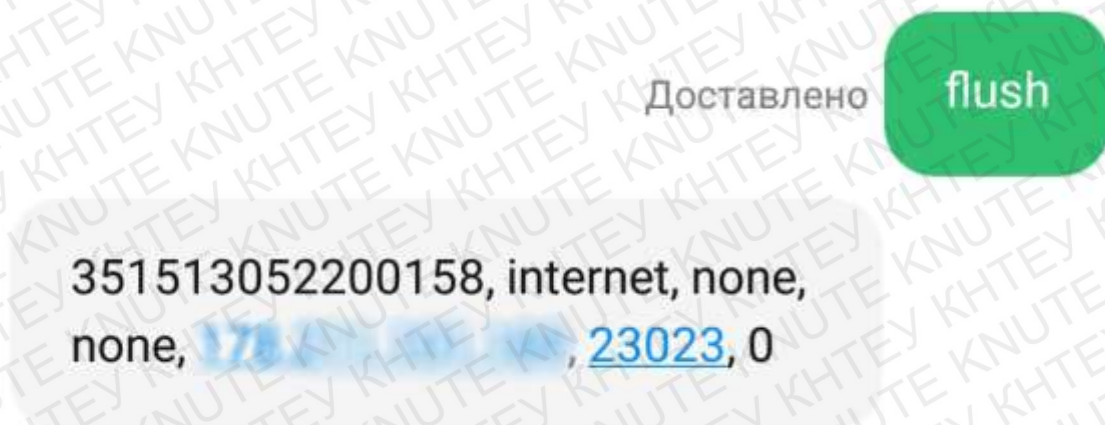


Рис 2.3. Команда flush

Приклад використання команди getparam наведено на Рис 2.4, де 0246 та 0245 – ідентифікатори параметрів.



Рис 2.4. Команда getparam

Приклад використання команди setparam:

- setparam 0246 23003

Ця команда задасть значення 23003 параметру з ідентифікатором 246. При успішному виконанні якої, відповіддю від пристрою буде це саме значення. Ідентифікатори параметрів та їх призначення, що використовуються при налаштуванні трекеру в даній роботі наведено у таблиці 2.3.

Таблиця 2.3.

Ідентифікатори параметрів

ID	Тип	Назва	Призначення
0261-0269	string	Phone	список авторизованих номерів
0245	string	Host	адреса серверу
0246	short int	Port	порт серверу
0020-0059	short int	Allowed Operators List	список кодів дозволених операторів
0242	string	APN	назва точки доступу GPRS
0243	string	Usname	логін доступу GPRS
0244	string	Uspass	пароль доступу GPRS
0270	short int	Send Period	період передачі даних на сервер
0906	short int	Connect Interval	період очікування між спробами з'єднання
0186	char	Device reboot	період перезавантаження в годинах (0-255)

2.2. Протокол формування даних «Codec 8»

Кодек - це пристрій або комп'ютерна програма для кодування або декодування цифрового потоку даних або сигналу. Кодек кодує потік даних або сигнал для передачі та зберігання, можливо, в зашифрованому вигляді, а функція декодера змінює кодування для відтворення або редагування.

					КНТЕУ 121 06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		14

Протокол формування даних під назвою «Codec 8» використовується більшістю трекерів виробництва BITREK. Цей протокол використовується також трекерами литовського виробництва Teltonika, що є провідним європейським виробником обладнання для супутникового моніторингу. Пристрої, що використовують протокол Codec 8, приймають або надсилають данні на сервер за допомогою протоколу передачі даних TCP.

Перед тим, як почати передачу даних на сервер, пристрій відправляє пакет для ініціалізації сесії. Перші два байти пакету містять розмір переданого IMEI пристрою для його ідентифікації. Так як IMEI складається із 15 символів, кожен символ має розмір одного байту, то загалом пакет ініціалізації сесії повинен мати розмір 17 байтів (Рис 2.5. Пакет ініціалізації сесії). Після отримання такого пакету, сервер повинен підтвердити або відкинути з'єднання відправивши два байти що містять одиницю або нуль в десятковій системі числення.

Розмір даних (2 byte)		IMEI (15 byte) ASCII кодування				
0	15	1	1	...	1	1

Рис 2.5. Пакет ініціалізації сесії

Після ініціалізації сесії, пристрій почне надсилати пакети з даними (Рис 2.6. Структура пакету), де:

- 0x00000000 – преамбула;
- Data Field Length – розмір, який обчислюється від «Codec ID» до «Number Of Data 2»;
- Codec ID – ідентифікатор кодеку, в даній роботі він дорівнює 0x08;
- Number Of Data 1 – кількість записів, що надіслані;
- AVL Data – основні дані, що містять передані записи;
- Number Of Data 2 – кількість записів, що надіслані;
- CRC-16 – контрольна сума пакету, що розрахована приладом.

4 byte	4 byte	1 byte	1 byte	X byte	1 byte	4 byte
0x00000000	Data Field Length	Codec ID	Number Of Data 1	AVL Data	Number Of Data 2	CRC-16

Рис 2.6. Структура пакету

Розмір «AVL Data» динамічний, тобто залежить від кількості записів що він містить у собі але його максимальний розмір дорівнює 1280 байтів. Опираючись на це можна відмітити, що максимальний розмір пакету з даними, що надсилає прилад на сервер, не може перевищувати 1295 байтів.

Дані, що передаються на сервер протягом сесії повинні перевірятися на цілісність. Сервер повинен містити метод, що розраховує контрольну суму CRC-16 надісланих даних від «Codec ID» до «Number of Data 2» та звіряти розраховану контрольну суму з контрольною сумою, що надіслано пристроєм в кінці пакету. Дані можна вважати цілісними, якщо контрольні суми збігаються. В іншому випадку сервер повинен відповісти на такий пакет нулем, що має розмір 4 байти.

Основні дані мають вигляд послідовного набору записів (Рис 2.7 Структура записів пакету), де:

- Timestamp - різниця в мілісекундах між поточним часом та північчю, січень, 1970 UTC (час UNIX);
- Priority - визначає пріоритет даних AVL;
- IO Element - додаткова інформація з датчиків;
- Longitude - довгота;
- Latitude - широта;
- Altitude - метри над рівнем моря;
- Angle - градуси від північного полюса;
- Satellites - кількість видимих супутників;
- Speed - швидкість, обчислена із супутників;

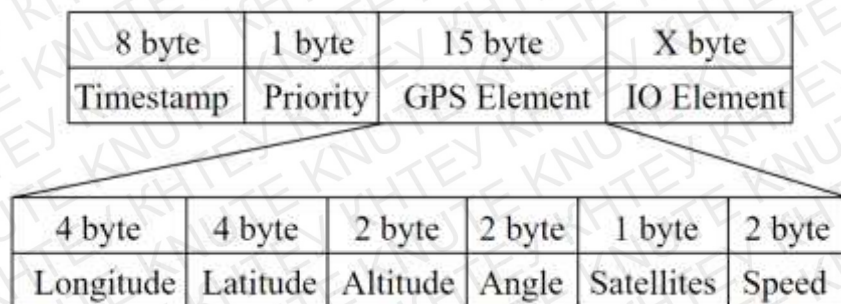


Рис 2.7. Структура записів пакету

Дані датчиків бувають чотирьох видів та відрізняються розміром (в байтах), але мають схожу структуру, що зображена на Рис 2.8, де:

- IO event - при значенні «0» дані створені не за подією;
- IO tnum - кількість даних загалом;
- IO num - кількість n байт даних;
- IO ID – ідентифікатор датчику;
- IO value - значення n байт;

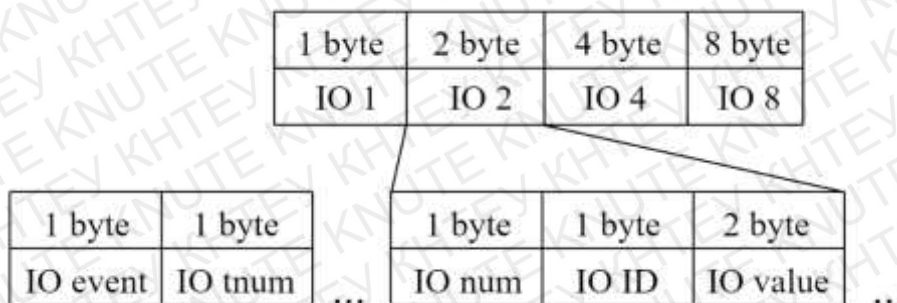


Рис 2.8. Структура даних датчиків

Після того як сервер отримав, перевірів цілісність та декодував дані, він повинен відповісти трекеру кількістю записів розміром 4 байти.

2.3. Архітектура REST

REST (англ. Representational State Transfer - «передача стану уявлення») - архітектурний стиль взаємодії компонентів розподіленого додатка в мережі.

В мережі Інтернет виклик віддаленої процедури може являти собою звичайний HTTP-запит (зазвичай «GET» або «POST»; такий запит називають «REST-запит»), а необхідні дані передаються в якості параметрів запиту (Рис 2.9
 Діаграма архітектури REST додатку).

Одна транзакція RESTful сервісу складається як мінімум з:

- методу запиту (GET)
- шляху запиту (/api/track)
- тіла запиту (params)
- коду відповіді (200 OK)
- тіла відповіді (JSON)

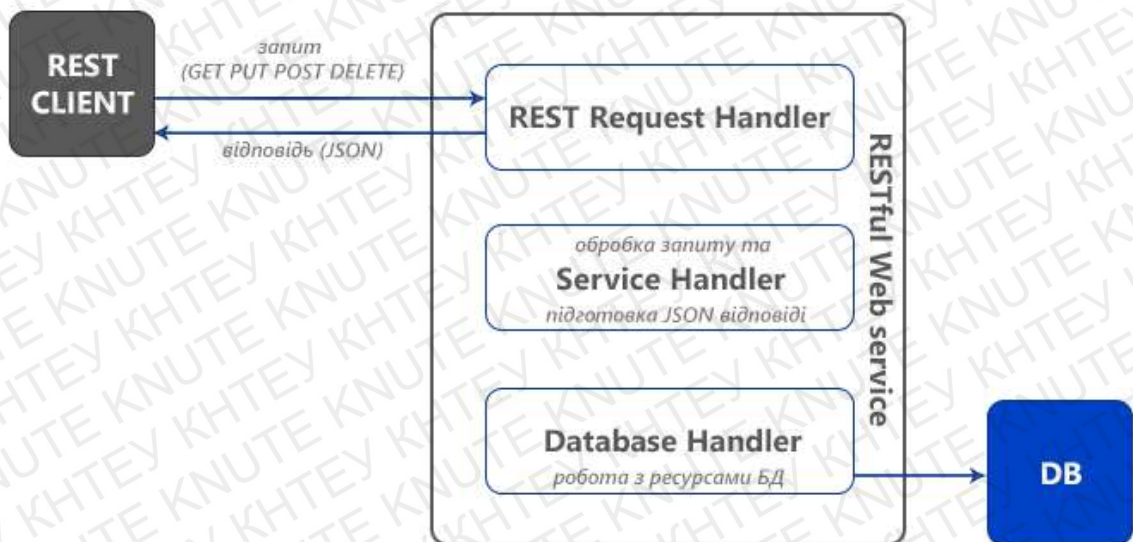


Рис 2.9. Діаграма архітектури REST додатку

Розроблена ІС використовує наступні методи запиту:

- GET – отримання даних об’єкту
- POST – створення об’єкту з переданих даних
- PUT – оновлення існуючого об’єкту
- DELETE – видалення існуючого об’єкту

Для тестування та створення колекцій веб сервісу використовується ПЗ під назвою Postman (Рис 3.2. Приклад GET запиту в Postman). Будь-який розробник або тестувальник відкривши колекцію зможе з легкістю розібратися в роботі сервісу. Тестувальники можуть писати тести і проводити автоматизоване тестування прямо

з Postman. А інструменти для автоматичного документування за описами з створених колекцій заощадають час.

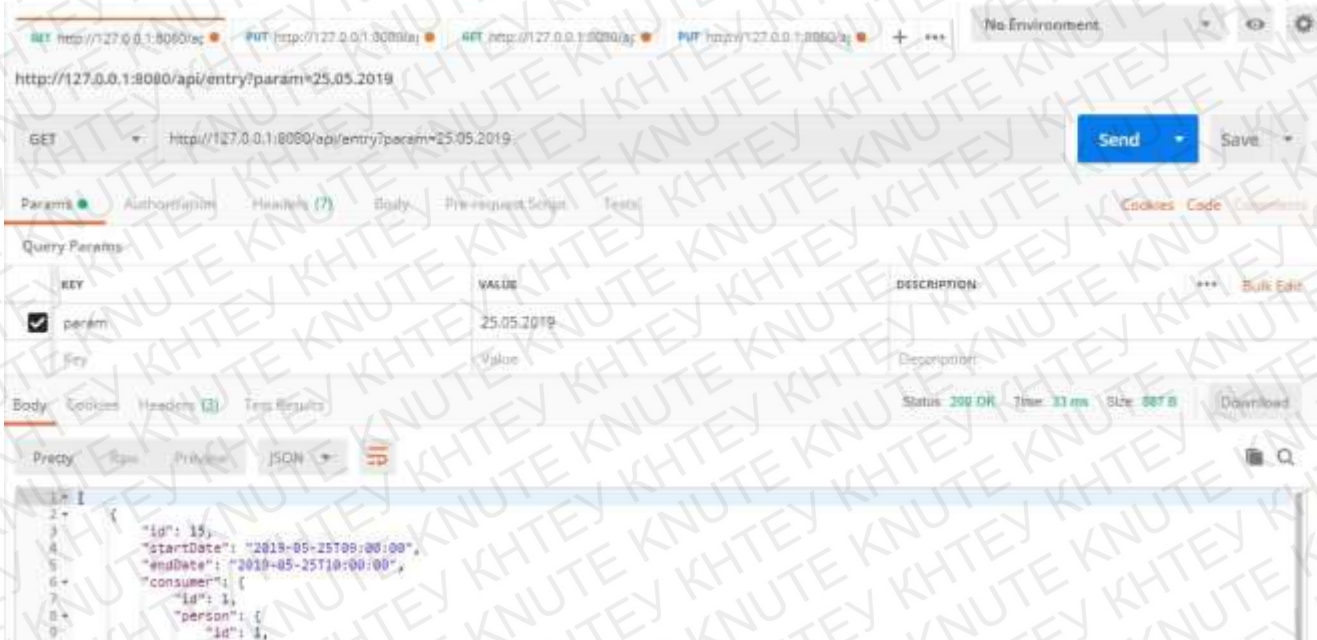


Рис 2.10. Приклад GET запиту в Postman

2.5. Висновки до розділу 2

В цьому розділі було розглянуто пристрій українського виробництва ВІ 910 TREK, його характеристики, підключення та способи налаштування. Розглянуто протокол формування даних під назвою Codec 8, що використовується GPS трекерами виробництва Vitrek, Teltonika та інших. Описано принцип спілкування серверної частини з пристроями моніторингу, а саме структуру пакету ініціалізації сесії та структуру пакетів з даними, що дає можливість написання методу декодування таких пакетів та використовувати його у серверній частині системи.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Проектування бази даних

Для розробки ІС було обрано СУБД MySQL. Створено три таблиці: «user» - для збереження автентифікаційних даних, «device» - для збереження даних про пристрої, «tracks» - для збереження геопозиційних даних.

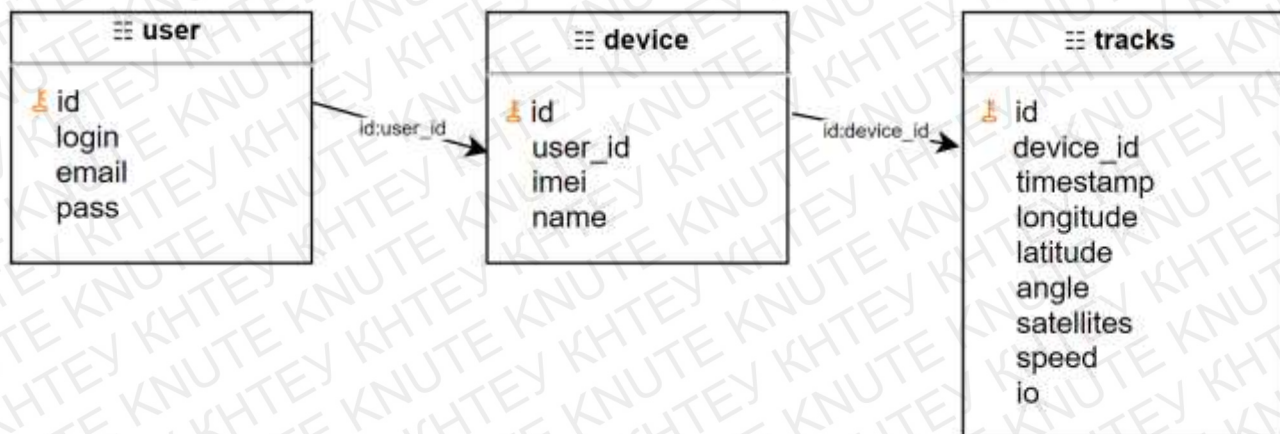


Рис 3.1. Логічна модель БД

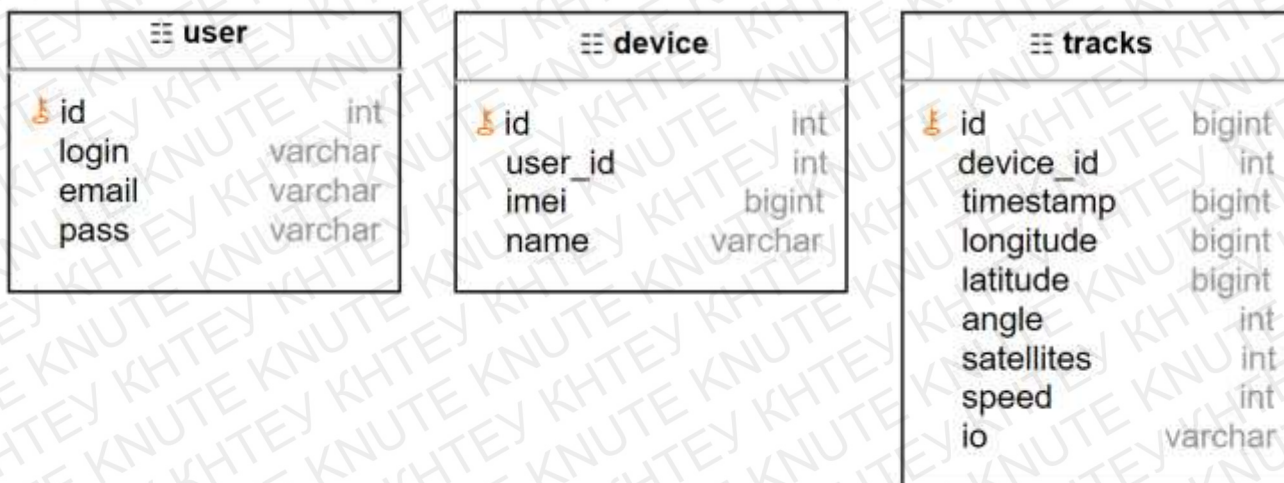


Рис 3.2. Фізична модель БД

					КНТЕУ 121 06-21.МР				
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів	
Зав. Каф.		Криворучко О.В.	19.10.20			РЗ	20	34	
Керівник		Рзаєва С.Л.	19.10.20						
Гарант		Криворучко О.В.	19.10.20			Розділ 3 Проектування та програмна реалізація	Факультет інформаційних технологій, 2м курс, 6 група		
Розробив		Цьомка О.О.	19.10.20						

3.2. Розробка серверної частини ІС з використанням Spring Boot

Серверну частину вирішено розробляти на мові програмування Java разом з Spring Boot та системою залежностей Maven. Spring Boot - платформа для розробки веб сервісів на основі Spring Framework з мінімальною потребою в конфігурації. Вона забезпечує вбудований HTTP сервер (*Tomcat, Jetty*). Spring Boot автоматично налаштовує додаток на основі залежностей, доданих до проекту за допомогою анотації *@EnableAutoConfiguration*. Точкою входу додатку є клас, що містить анотацію *@SpringBootApplication* і основний метод.

Головна перевага Maven - це управління залежностями. Рідко які проекти пишуться без використання сторонніх бібліотек. Ці сторонні бібліотеки теж в свою чергу використовують бібліотеки різних версій. Maven дозволяє керувати такими складними залежностями, що дозволяє вирішувати конфлікти версій і в разі потреби легко переходити на нові версії бібліотек. Залежності підключаються до проекту у файлі «*pom.xml*» (Рис 3.3. Залежності розроблюваної ІС).

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.15</version>
  </dependency>
</dependencies>
```

Рис 3.3. Залежності розроблюваної ІС

Структура проекту складається з пакетів що містять класи, конфігураційного файлу Spring, файлу залежностей Maven та пакету статичних ресурсів, що містить клієнтську частину (Рис 3.4. Структура проекту).

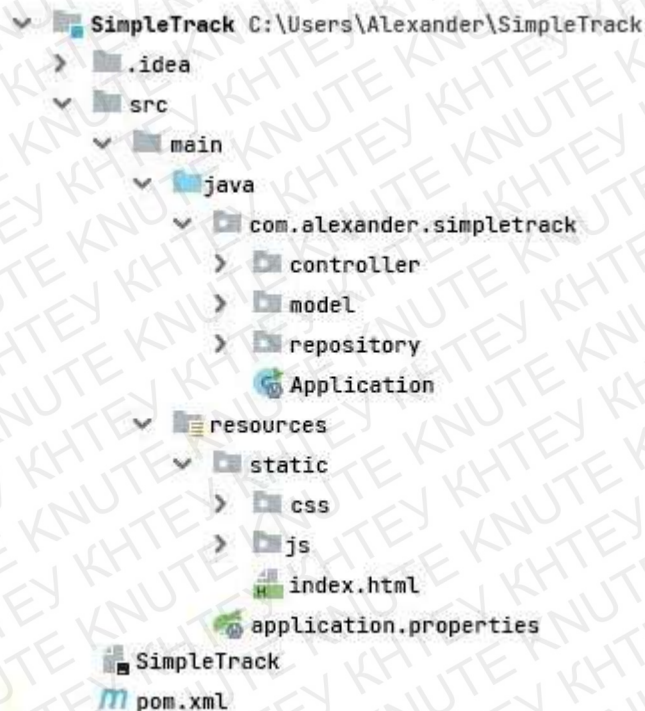


Рис 3.4. Структура проекту

Конфігураційний файл Spring (*application.properties*) використовується для налаштування підключення до бази даних, в якому містяться: адреса з параметрами часового поясу, логін та пароль (Рис 3.5. Конфігурація підключення до БД).

```
spring.datasource.url=jdbc:mysql://localhost:3306/demo?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=Europe/Kiev
spring.datasource.username=root
spring.datasource.password=...

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update

logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type=TRACE
```

Рис 3.5. Конфігурація підключення до БД

JpaRepository передбачає наявність сутностей (Рис 3.6. Фрагмент сутності «Track») у виді класів які використовуються для роботи з таблицею БД. Сутність містить конструктори, геттери та сеттери. Використовуються наступні анотації:

- *@Entity* – оголошення сутності
- *@Table (name = "track")* – дозволяє уточнити деталі таблиці

- `@Column` – оголошення стовпця
- `@Id` – ідентифікує стовпець первинним ключем
- `@GeneratedValue(strategy = GenerationType.IDENTITY)` – конфігурує спосіб інкрементації ключа
- `@NotNull` – не дозволяє приймати значення NULL
- `@Size(max = 70)` – оголошує максимальну кількість символів стовпця

```
@Entity
@Table (name = 'track')
public class Track {
    @Id
    @GeneratedValue (strategy = GenerationType.IDENTITY)
    private long id;
    ...
}
```

Рис 3.6. Фрагмент сутності «Track»

Щоб додати до таблиці зовнішні ключі використовуються спеціальні анотації (Рис 3.7. Оголошення зовнішнього ключа): `@ManyToOne`, `@JoinColumn`, `@OnDelete`.

```
@ManyToOne (fetch = FetchType.EAGER, optional = false)
@JoinColumn (name = 'user_id', nullable = false)
@OnDelete (action = OnDeleteAction.NO_ACTION)
private User user;
```

Рис 3.7. Оголошення зовнішнього ключа

Окрім сутностей для них створено інтерфейси репозиторіїв. `JpaRepository` - це інтерфейс фреймворка Spring Data що надає набір стандартних методів JPA для роботи з БД. Він оголошується анотацією `@Repository`. При відсутності потрібних методів є можливість додати свій за допомогою анотації `@Query`.

Контролери визначають адресу та методи передачі даних. Для оголошення класу контролером існує спеціальна анотація `@RestController`. Анотація

@Autowired дозволяє підключити потрібний репозиторій оголосивши його. API веб сервісу має наступну колекцію:

- /api/location
- /api/ride
- /api/device
- /api/signup
- /api/signin

В даний час JWT користується популярністю для автентифікації та обміну інформацією. Замість того щоб створювати сеанс (автентифікація на основі сеансу), сервер кодує дані в веб-токен JSON і відправляє його клієнту. Клієнт зберігає JWT, а потім кожен запит від клієнта до захищених маршрутів або ресурсів повинен бути прикріплений до цього JWT (зазвичай в заголовку). Сервер перевірить цей JWT і поверне відповідь.

Порівнюючи з перевіркою автентичності на основі сеансу, у якій потрібно зберігати сесії на куки, великою перевагою автентифікації (перевірка на основі маркерів) полягає в тому, що ми зберігаємо маркер на стороні клієнта, тобто у локальному сховищі браузера. Таким чином, нам не потрібно створювати ще один бекенд-проект, що підтримує власні програми, або додатковий модуль автентифікації для користувачів власних додатків.

Існує три важливі частини JWT: Header, Payload, Signature. Разом вони об'єднуються в структуру: *header.payload.signature*.

Клієнт зазвичай звертається до JWT в заголовку авторизації з префіксом Bearer:

Authorization: Bearer [header].[payload].[signature]

					КНТЕУ 121 06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		24

3.3. Проектування клієнтської частини ІС з використанням Vue

Клієнтська частина являє собою веб додаток, реалізований за допомогою CSS, HTML, JS. Порівняння найпопулярніших JS фреймворків наведено у таблиці 3.1.

Таблиця 3.1.

Порівняння JavaScript фреймворків

	Angular	React	Vue
Засновники	Google	Facebook	Колишній працівник Google
Перший випуск	Вересень 2016	Березень 2013	Лютий 2014
Підходить для	Широкомасштабні, багатофункціональні додатки	Сучасні веб розробки, native додатки під Android та iOS	Веб розробки та односторінкові додатки
Популярність	Широко поширений серед розробників	Понад 27 тисяч зірок додано за рік	Більше 40 тисяч зірок на GitHub протягом року
Компанії, які використовують	Google, Forbes, Wix та weather.com	Facebook, Uber, Netflix, Twitter, Reddit, Paypal, Walmart та інші	Alibaba, Baidu, GitLab та інші

Вирішено обрати фреймворк Vue для реалізації клієнтської частини тому, що:

- це найкращий вибір для односторінкових додатків;
- надає даним реактивності;
- набирає популярність серед розробників;
- простіший для освоєння на перших етапах фронтенд розробки.

React дуже схожий по перевагам, але показує трохи гірші результати у тестах на швидкодію.

Об'єктна модель документа (DOM) - це програмний інтерфейс (API) для HTML документів. DOM повністю підтримує об'єктно-орієнтоване уявлення веб-сторінки, роблячи можливим її зміну за допомогою мови JavaScript.

Всі шаблони Vue є дійсним HTML-кодом (Рис 3.8 Фрагмент шаблону Vue). Для роботи Vue компілює шаблони в render-функції віртуального DOM. У поєднанні з системою реактивності, Vue вміє визначати мінімальне число компонентів для повторного рендеру і застосовує мінімальну кількість маніпуляцій до DOM при зміні стану програми.

Щоб оголосити межі дії екземпляру на HTML сторінці треба вказати в його властивості «el» назву id існуючого DOM елементу.

```
<el-table :data='tracker_table' style='width: 100%'>
  <el-table-column type='expand'>
    <template slot-scope='props'>
      <p>Назва: {{ props.row.name }}</p>
      <p>imei: {{ props.row.imei }}</p>
    </template>
  </el-table-column>
</el-table>
```

Рис 3.8. Фрагмент шаблону Vue

Для виводу змінних використовується mustache синтаксис: {{ props.row.name }}. Двокрапка перед DOM атрибутом означає, що він реактивний та буде завантажений зі змінної екземпляру, у цьому випадку фігурні дужки не потрібні.

Змінні або масиви які потребують реактивності потрібно представити у виді об'єктів та розмістити в спеціальній властивості «data» потрібного екземпляру. Vue автоматично створить для них усі потрібні геттери та сеттери. Усі дані, які знаходяться поза властивістю «data», не можуть бути реактивними. Реактивність дозволяє відслідковувати зміни даних та створювати потрібні при змінах методи. Для використання функцій їх потрібно розмістити у властивості «methods».

Так як односторінковий додаток передбачає роботу з даними без потреби перезавантажувати сторінку потрібно якимсь чином додати йому динамічності. Для реалізації асинхронних запитів підключено JS бібліотеку Axios (Рис 3.9

Приклад axios запиту GET). На відміну від метода *fetch()*, при використанні якого для передачі даних в JSON, необхідно виконати метод в два етапи: спочатку зробити фактичний запит, а потім викликати метод *json()* для перетворення отриманих даних з сервера, Axios робить це автоматично.

```

axios
  .get('/api/track', {
    params: {
      param: param_date
    }
  })
  .then(response => {
    data.tracks = response.data;
  });

```

Рис 3.9. Приклад axios запиту GET

3.4. Візуалізація двовимірної мапи за допомогою Leaflet

Leaflet - провідна відкрита бібліотека JavaScript для мобільних інтерактивних карт. Вагою всього близько 39 KB JS-коду, вона володіє всіма можливостями відображення, які коли-небудь були необхідні більшості розробників.

Leaflet розроблений з урахуванням простоти, продуктивності і зручності використання. Він ефективно працює на всіх основних настільних і мобільних платформах, може бути розширений за допомогою безлічі плагінів, має гарний, простий у використанні і добре документований API і простий, читаємий вихідний код, в який приємно внести свій вклад.



Рис 3.10. Перелік компаній що використовують Leaflet

					КНТЕУ 121 06-21.МР	Аркуш
Зм.	Арк.	№ докум.	Підпис	Дата		27

Для того, щоб почати використання Leaflet потрібно зробити декілька кроків. Першим кроком буде підключення CSS та JS файлів до нашого програмного забезпечення на стороні клієнтській стороні (Рис 3.11. Підключення Leaflet за допомогою CDN unpkg).

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
  integrity="sha512-xodZBNTC5n17Xt2atTPUE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOAS6/keqq/sM2MZ19ScR4PsZChSR
  crossorigin=""/>
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
  integrity="sha512-XQoYMqMTK8LvI.XY0GZ3nZ448hOEQiglfqkJs1NOqV44cwnnrBc8PKAc8xxy20w0vlaXaVUearIOBhiXZ5V3ynx
  crossorigin=""></script>
```

Рис 3.11. Підключення Leaflet за допомогою CDN unpkg

Наступним кроком буде створення порожнього блоку div з параметром id, назву якого ми будемо використовувати у JS коді для конфігурації мапи. Далі потрібно обрати постачальника тайлів (плиток) мап.

Плиткова карта (растрова або векторна) - це карта, що відображається в браузері шляхом безшовного з'єднання десятків індивідуально запитуваних файлів зображень або векторних даних через інтернет (Рис 3.12. Приклад завантаження плиткової карти). Це найпопулярніший спосіб відображення та навігації по картах, замінюючи інші методи, такі як WMS, які зазвичай відображають одне велике зображення, кнопками зі стрілками для навігації по довколишніх областях. Google Maps був одним з перших великих картографічних сайтів, які використовували цей метод. Перші плиткові веб-карти використовували растрові зображення плитки, до появи векторних плиток.



Рис 3.12. Приклад завантаження плиткової карти

Є кілька переваг плиткових карт. Кожен раз, коли користувач відкриває карти, більшість плиток все ще залишаються на екрані, в той час як нові плитки з'являються на екрані. Це значно підвищує зручність роботи користувача в порівнянні з витяганням одного зображення карти для всього видового екрану. Крім того, відображення рендерингових зображень, що обслуговуються веб-сервером, вимагає менших обчислювальних витрат, ніж рендеринг зображень в браузері, що є перевагою в порівнянні з такими технологіями, як WFS. У той час як значна кількість плиток знаходяться в растровому форматі (такий як PNG або JPG), кількість постачальників векторних плиток зростає. Перевага векторних плиток полягає в тому, що можна застосовувати до карти свій власний стиль. У цьому випадку фактичні пікселі плиток карти вимальовуються браузером. Перевага векторних плиток полягає в тому, що користувач може вирішити повернути карту без того, щоб всі текстові мітки стали нечитабельним, так як мітки при цьому не повертаються.

Спочатку для системи планувалося використання плиток Google Maps, але після оновлення політики використання їх сервісів було вирішено обрати іншого

постачальника. Так як існує велика кількість постачальників плиток (рис 3.13.) було обрано найкращі варіанти для розробки системи. З восьми варіантів прийнято рішення використати векторні плитки постачальника MapTiler. Плитки цього сервісу мають безліч переваг та, навіть, не поступаються плиткам від Google. Одним із важливих переваг є оновлення та точність даних доріг, будівель та текстових міток на території України.



Рис 3.13. Порівняння постачальників плиток map

Останнім кроком є підключення посилання яку надає постачальник плиток карт до Leaflet шляхом оновлення параметра, котрий посилається на об'єкт раніше створеного блоку div в розмітці клієнтської частини.

3.5. Висновки до розділу 3

В даному розділі описано розробку та структуру серверної частини і бази даних. Проаналізовано найпопулярніші фреймворки для створення клієнтської частини та обрано один із них для розробки системи. Розглянуто принцип роботи візуалізації map, а саме допоміжну бібліотеку Leaflet та роботу плиткових map. Визначено різницю в використанні растрових плиток та векторних. Проаналізовано та обрано постачальника плиток map.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Відповідно до поставлених завдань, можна зробити наступні висновки.

Було розглянуто принцип роботи GPS, трекерів, пристрій українського виробництва BITREK, протокол формування даних «Codec 8», бібліотеку для роботи з мапами Leaflet, визначено основні функціональні вимоги та створено технічне завдання на розробку інформаційної системи. Згідно технічного завдання було розроблено базу даних, серверну частину з використанням мови програмування Java, методу формування даних «Codec 8» та методу передачі даних REST. Окрім серверної частини було спроектовано зрозумілий та зручний інтерфейс для клієнтської частини, яка розроблена з використанням сучасного інструменту Vue та Leaflet.

Щодо подальшого вдосконалення система може бути доповнена підтримкою ще декількох не менш популярних протоколів формування даних для збільшення кількості підтримуваних виробників приладів, підтримкою більшості зовнішніх датчиків, що підключаються до трекерів та розрахунками загальних статистичних даних про стан транспортних засобів.

					КНТЕУ 121 06-21.МР			
					Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Зм.	Арк.	№ докум.	Підпис	Дата		ВП	31	34
Зав. Каф.	Криворучко О.В.		30.10.20			Факультет інформаційних технологій, 2м курс, 6 група		
Керівник	Рзаєва С.Л.		30.10.20					
Гарант	Криворучко О.В.		30.10.20					
Розробив	Цьомка О.О.		30.10.20		Висновки та пропозиції			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Herbert Schildt - Java: The Complete Reference, Ninth Edition. / Herbert Schildt - McGraw-Hill Education; 9 edition, 2014 – 1312 с.
2. Shelley Powers - JavaScript Cookbook: Programming the Web / Shelley Powers - O'Reilly Media; 2 edition, 2015 – 634 с.
3. Paul DuBois - MySQL Cookbook: Solutions for Database Developers and Administrators / Paul DuBois - O'Reilly Media; 3 edition, 2014 – 866 с.

Інтернет ресурси

1. Getting Started - Building a RESTful Web Service [Електронний ресурс]. – Режим доступу : <https://spring.io/guides/gs/rest-service>.
2. Spring Data JPA - Reference Documentation [Електронний ресурс]. – Режим доступу: <https://docs.spring.io/spring-data/jpa/docs/current/reference/html>.
3. Guide - Vue.js [Електронний ресурс]. – Режим доступу: <https://vuejs.org/v2/guide>.
4. Java tutorial – Learn Java [Електронний ресурс]. – Режим доступу: <https://www.javatpoint.com/java-tutorial>.
5. Components - Element [Електронний ресурс]. – Режим доступу: <https://element.eleme.io/#/en-US/component>.

					КНТЕУ 121 06-21.МР			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В.		24.01.20		СД	32	34
Керівник		Рзасва С.Л.		24.01.20				
Гарант		Криворучко О.В.		24.01.20		Список використаних джерел	Факультет інформаційних технологій, 2м курс, 6 група	
Розробив		Цьомка О.О.		24.01.20				

ТЕХНІЧНЕ ЗАВДАННЯ

Розробка інформаційної системи передбачає наступне технічне завдання:

1) Загальні відомості.

Назва інформаційної системи: SimpleTrack.

Терміни початку та кінця робіт: Липень 2020 – Жовтень 2020

Потенційні користувачі:

- Логістичні підприємства
- Транспортні підприємства
- Сільсько-господарські підприємства
- Фізичні особи

Мета: моніторинг транспортних засобів за допомогою gps трекерів.

Призначення: перегляд геолокації об'єктів моніторингу на мапі.

2) Функціональні вимоги.

Інформаційна система повинна надати користувачу:

- Роботу з заздалегідь налаштованими gps трекерами.
- Візуальне відображення 2D мапи з можливістю зміни масштабу, що містить усі стандартні для мапи умовні позначки (назви міст, вулиць, річок і т. п.).
- Відображення останнього місцеположення об'єктів моніторингу у вигляді маячку на мапі.
- Відображення пройденого маршруту об'єкту моніторингу за певною датою.

Інші функціональні вимоги:

- Перевірка цілісності переданих трекером даних.
- Збереження геолокаційних даних об'єктів моніторингу.

					КНТЕУ 121 06-21.МР			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка системи супутникового моніторингу та контролю автотранспорту логістичного підприємства	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В.		28.02.20		ТЗ	33	34
Керівник		Рзаєва С.Л.		28.02.20				
Гарант		Криворучко О.В.		28.02.20				
Розробив		Цьомка О.О.		28.02.20				
					Технічне завдання	Факультет інформаційних технологій, 2м курс, 6 група		

- Кожен користувач повинен мати доступ лише до своїх об'єктів моніторингу.
- Усі поля форм повинні перевірятися в залежності від типу.

3) Вимоги до апаратного забезпечення.

Назва	Клієнт	Сервер
Центральний процесор	2 GHz+	2 x 2 GHz+
Платформа	x86-64, ARM	x86-64
Оперативна пам'ять	2 Gb+	4 Gb+
Вільне місце на диску	350 Mb	50 Gb
Мережева карта	10/100/1000 Mbps	100/1000 Mbps

GPS - трекер повинен мати GSM модуль та, щонайменше, один слот для сім картки.

4) Вимоги до програмного забезпечення.

На серверному обладнанні повинно бути попередньо встановлено наступне ПЗ:

- Операційна система Windows 7+ або Ubuntu 10.04+
- JRE 1.8 202
- MySQL 8.0 (або PostgreSQL)

Клієнтське обладнання повинне мати операційну систему та один із сучасних браузерів: Google Chrome, Mozilla Firefox, Brave.

GPS - трекер повинен працювати на протоколі «Codec 8».