

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

**«Проектування та створення Web- додатку моніторингу
автомобільного трафіку»**

Студента 4 курсу, 13 групи,

спеціальності
122 «Комп'ютерні науки»

Леонтьюка

Микити

Олександровича

підпис студента

Науковий керівник

кандидат фізико-математичних наук,
доцент

Самойленко Ганна

Тимофіївна

підпис керівника

Гарант освітньої програми

кандидат технічних наук, доцент

Демідов Павло

Георгійович

підпис керівника

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Пурський О.І.
« » грудня 2020р.

Завдання на випускний кваліфікаційний проект студенту

Леонтьоку Микиті Олександровичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

«Проектування та створення Web- додатку моніторингу автомобільного трафіку»

Затверджена наказом ректора від «04» грудня 2020 р. № 4111

2. Строк здачі студентом закінченої роботи 29 травня 2021 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: обґрунтування та розробка створення Web-орієнтованого мобільного додатку моніторингу автомобільного трафіку, з урахуванням сучасних тенденцій побудови організаційних та функціональних інформаційних структур даного заходу.

Об'єкт дослідження: процес проектування Web-орієнтованого мобільного додатку

Предмет дослідження: засоби створення Web-орієнтованого мобільного додатку

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.
2	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.
3	Самойленко Г.Т.	15.12.2020 р.	15.12.2020 р.

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 Теоретичні основи в розробці сучасного Web-орієнтованого мобільного додатку

1.1. Поняття та принципи роботи сучасного Web-орієнтованого додатку

1.2. Огляд існуючих засобів проектування

Висновки до розділу

РОЗДІЛ 2. Організація розробки Web-орієнтованого додатку

2.1. Аналіз предметної області - моніторингу автомобільного трафіку

2.2 Аналіз існуючих систем для проектування мобільних Web-орієнтованих додатків

2.3 Переваги вибраної системи

Висновки до розділу

РОЗДІЛ 3. Програмна реалізація розробки Web-орієнтованого мобільного додатку

3.1 Структура мобільного Web—додатку моніторингу автомобільного трафіку

3.2. Розробка користувацького інтерфейсу

3.3 Налаштування додатку з моніторингу

3.4 Тестування.

Висновки до розділу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.10.2020	01.10.2020
2	Розробка та затвердження завдання на випускну кваліфікаційну роботу	15.12.2020	15.12.2020
3	Вступ	03.02.2021	
4	РОЗДІЛ 1. Теоретичні основи в розробці сучасного Web-орієнтованого мобільного додатку.	28.02.2021	
5	РОЗДІЛ 2. Організація розробки Web-орієнтованого додатку.	06.04.2021	
6	Організація розробки Web-орієнтованого додатку	12.05.2021	
7	Висновки	15.05.2021	
8	Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику	20.05.2021	
9	Попередній захист випускної кваліфікаційної роботи	26.05.2021	
11	Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи	27.05.2021	
12	Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі	29.05.2021	
13	Публічний захист випускної кваліфікаційної роботи	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2020р.

9. Керівник випускного кваліфікаційного проекту

Самойленко Г.Т.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Леонтьук М.О.

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slightly textured appearance and is set against a dark background.

Керівник випускної кваліфікаційної роботи (проекту)

30.05.2021 p.

(nidnuc, dama)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми:

(підпис, прізвище, ініціали)

Демідов П.Г.

Завідувач кафедри

(підпис, прізвище, ініціали)

Пурський О.І.

« » 2021 p.

)

АНОТАЦІЯ

Метою цієї роботи є створення системи зчитування даних з сенсорів смартфона та аналізу інформації для подальшого її використання у сфері транспорту.

Система складається з мобільного додатку, за допомогою якого зчитуються показники сенсорів, веб-додатку, для детального аналізу отриманої інформації та веб-серверу. Мобільний додаток реалізовано засобами мови Swift для операційної системи IOS. Веб-додаток створено за допомогою скриптової мови JavaScript. Веб-сервер реалізовано з використанням мов програмування Swift та SQL.

Загальний обсяг роботи: 45 сторінок, 12 таблиць, 31 ілюстрація та 11 використаних джерел.

Ключові слова: транспорт, геоінформаційна система, сенсори смартфона.

ABSTRACT

The purpose of this work is creating a system for reading data from smartphone sensors and analysis this information for future use of transport telematics.

The system consists of a mobile application, which reads the sensor data and a web application for performing detailed analysis of the received information. The mobile application for iOS was created using the Swift programming language. The web application was created using the JavaScript programming language. The web server was implemented using Swift and SQL programming languages.

Number of pages: 45

Number of tables: 12

Number of illustrations: 31

Number of sources: 11

Key words: transport telematics, geographic information system, smartphone sensors.

ЗМІСТ

Перелік умовних позначень та скорочень.....	9
ВСТУП.....	10
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ В РОЗРОБЦІ СУЧАСНОГО WEB-ОРІЄНТОВАНОГО МОБІЛЬНОГО ДОДАТКУ.....	12
1.1. Вибір технологій та їх обґрунтування.....	13
1.2. Мова програмування Swift.....	14
1.3. Мова програмування JavaScript.....	15
1.4. Середовище розробки xCode.....	15
1.5. Фреймворк UIKit.....	15
1.6. Мова програмування SQL.....	16
1.7. СУБД MySQL.....	17
1.8. Веб-фреймворк Vapor.....	18
РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ WEB-ОРІЄНТОВАНОГО ДОДАТКУ.....	19
2.1. Структура програмного забезпечення.....	19
2.2. Опис бази даних.....	20
2.3. Діаграма прецедентів системи.....	28
РОЗДІЛ 3. ПРОГРАМНА ОРГАНІЗАЦІЯ РОЗРОБКИ WEB- ОРІЄНТОВАНОГО ДОДАТКУ.....	29
3.1. Мобільний додаток користувача.....	29
3.2. Розширена версія мобільного додатку.....	33
3.3 Веб -додаток.....	38
ВИСНОВКИ ТА РЕЗУЛЬТАТИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ITS — інтелектуальна транспортна системи

Wi-Fi — wireless fidelity, технологія бездротової локальної мережі

СКБД — система керування базами даних

БД — база даних

ГБД — геоінформаційна база даних

CPU — central processing unit, електронний блок або інтегральна схема, яка виконує машинні інструкції, процесор

API — application programming interface, опис способів, якими одна комп'ютерна програма може взаємодіяти з іншою програмою

LLVM — low level virtual machine, проект програмної інфраструктури для створення компіляторів і супутніх їм утиліт

UML — unified modeling language, це система позначень, яку можна застосовувати для об'єктно-орієнтованого аналізу і проектування

ORM — Object-Relational Mapping, технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування

ВСТУП

Сьогодні однією з невирішених проблем залишається збір, аналіз та обробка даних. Кожна сфера розробляє свої стратегії та методи роботи з інформаційними потоками. Збір інформації для дорожньо-транспортної сфери навіть сьогодні є проблематичними, не зважаючи на те що досить велика кількість сфер вже давно винайшла оптимальні шляхи обробки інформації й протягом тривалого часу не змінює ці методи.

Транспортна сфера являється одним із основних факторів, які впливають на соціально-економічне життя населення, тому дослідження області та покращення якості транспортної системи має велике значення. У цих умовах, транспортна система стала однією з найбільш перспективних і необхідних галузей, яка являє собою систему віддаленого контролю стану автотранспорту. Нині вже існує певна кількість додатків, що працюють за допомогою систем транспортної навігації та їх кількість з кожним роком продовжує зростати.

Аналіз дорожньо-транспортного руху, стану доріг та поведінки водіїв на дорогах викликає значні труднощі, як в теорії так і на практиці. У сучасному світі автотранспортні засоби та дороги оснащують спеціальними сенсорами та пристроями, такими як камери спостереження для відслідковування автомобільного трафіку, різні типи радарів та відеореєстратори. Однак, дані зібрані з цих пристроїв досить важко групувати, передавати на інші пристрої та аналізувати.

Щоб вирішити поставлену проблему необхідно знайти такий спосіб, який допоможе швидко та максимально ефективно зібрати дані про події, що відбуваються, під час руху автотранспорту й зможе оптимальним шляхом підготовки до їхнього аналізу. Одним із методів найефективнішого вирішення проблеми буде створення мобільного додатку, який збирає дані сенсорів мобільного пристрою та конвертує їх до вигляду, в якому їх легко та швидко аналізувати.

Програмне забезпечення допоможе збирати інформацію для декількох ключових цілей:

- Аналіз стану доріг;
- Аналіз дорожнього руху;
- Побудова оптимальних маршрутів;
- Класифікація подій на дорозі;
- Аналіз поведінки водіїв на дорозі;

Такий спосіб обробки даних може бути оптимальним для всіх, без винятку, транспортних засобів, що є значною перевагою серед інших методів аналізу цієї предметної області. Отже, враховуючи мобільність даного методу збору та аналізу даних, можна зробити висновок, що побудована система є найбільш логічним варіантом вирішення задачі, що стоїть перед нами. Таким чином програмний продукт надає можливість максимально оптимізувати процес збору даних для дорожньо-транспортної сфери та, завдяки зручному інтерфейсу, значно полегшити аналіз отриманої інформації.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ В РОЗРОБЦІ СУЧАСНОГО WEB-ОРІЄНТОВАНОГО МОБІЛЬНОГО ДОДАТКУ

1.1 Вибір технологій та їх обґрунтування

Систему було реалізовано за допомогою мов програмування Swift та JavaScript. Серверна частина веб-додатку(API) було реалізовано за допомогою мови програмування Swift та фреймворка Vapor. Графічна частина веб-додатку була реалізована за допомогою мови програмування JavaScript, на сьогодні це найбільш оптимізована та найпопулярніша мова для розробки графічної частини веб-додатків, яка є монополістом у даній галузі. Мобільний додаток було реалізовано за допомогою мови програмування Swift на платформі iOS, яка також є основною для даної галузі, виключення Objective-C, але Swift було запроваджено, як сучасну, оптимізовану заміну Objective-C. Додаток створено на основі фреймворку UIKit. Заміною UIKit, міг би стати фреймворк SwiftUI, але після дослідження було виявлено, що він ще достатньо молодий та не має більшості необхідних інструментів.

Для взаємодії з сенсорами телефону було використано наступні бібліотеки:

- CoreLocation – бібліотека для роботи з GPS
- CoreMotion – бібліотека для роботи з сенсорами телефона (акселерометр, гіроскоп, магнітометр)

1.2 Мова програмування Swift

Багатопарадигмова компільована статична мова програмування з відкритим кодом, що розробляється компанією Apple. Вона побудована на базі LLVM компілятора, та компілюється у машинний код.

Swift була представлена на конференції розробників WWDC 2014.

Ідеї для Swift запозичені із таких мов програмування як Objective-C, Rust, Haskell, Ruby, Python, CLU, та багатьох інших зі списку. Ця мова вже має за плечима роки розвитку, і вона продовжує розвиватися, включаючи в себе все нові і нові можливості.

Компілятор Swift побудований з використанням технологій вільного проекту LLVM. Swift успадковує найкращі елементи мов C і Objective-C, тому синтаксис звичний для знайомих з ними розробників, але водночас відрізняється використанням засобів автоматичного розподілу пам'яті і контролю переповнення змінних і масивів, що значно збільшує надійність і безпеку коду.

Мова пропонує безліч сучасних методів програмування, таких як замикання, узагальнене програмування, лямбда-вирази, кортежі і словникові типи, швидкі операції над колекціями, елементи функційного програмування. Основним застосуванням Swift є розробка користувацьких застосунків для macOS, iOS, tvOS, watchOS з використанням інструментів Cocoa і Cocoa Touch. При цьому Swift надає об'єктну модель, сумісну з Objective-C. Сирцевий код мовою Swift може змішуватися з кодом на C і Objective-C в одному проекті.

Swift – це потужна мова програмування, така ж зрозуміла, захоплююча та цікава, як скриптова мова. Вона підтримує так звані playground-и, які дозволяють програмістам експериментувати з кодом, бачачи результат в режимі реального часу без необхідності компілювати і запускати додаток.

Swift виключає велику кількість поширених програмних помилок за допомогою застосування сучасних програмних паттернів.

1.3 Мова програмування JavaScript

JavaScript – скриптова мова, що зазвичай використовується при створенні сценаріїв поведінки браузера, що вбудовуються у веб-сторінки. Була створена для того, щоб "оживляти веб-сторінки". Програми, які використовуються цією мовою називаються скриптами, тому що їх можна записати прямо в HTML веб-сторінки та запустити автоматично під час завантаження сторінки. Сценарії надаються та виконуються як звичайний текст для запуску їм не потрібна спеціальна підготовка чи компіляція.

JavaScript використовується в якості мови сценаріїв на стороні клієнта. Це означає, що код JavaScript записується на сторінку HTML. Коли користувач вимагає HTML-сторінки з JavaScript у ній, скрипт надсилається до браузера, і браузер повинен вирішувати щось робити.

JavaScript може використовуватися в інших контекстах, ніж веб-браузер. Netscape створив JavaScript на стороні сервера як CGI-мову, яка може робити приблизно так само, як Perl або ASP. Немає жодної причини, через яку JavaScript не можна було б використовувати для написання реальних, складних програм. Однак сайт стосується виключно використання JavaScript у веб-браузерах.

Сьогодні JavaScript може виконуватись не тільки в браузері, але і на сервері або фактично на будь-якому пристрою, який має спеціальну програму під назвою механізм JavaScript. У браузері є вбудований «двиглок», який іноді називають "віртуальною машиною

1.4 Середовище розробки xCode

xCode – це інтегрована середовище розробки(IDE) від компанії Apple. Яка дозволяє створювати програмне забезпечення, в першу чергу, для таких мов як Swift, Objective-C, C, а також розуміє і надає візуальну допомогу при написанні коду таких мов як JavaScript, SQL, Ruby та інших мовах.

xCode має безліч корисних інструментів для аналізу роботи та оптимізації iOS додатків, наприклад, інструменти для аналізу використання оперативної пам'яті та пошуку витоків пам'яті (memory leak), інструменти для аналізу енерговитрат пристрою під час використання вашого додатку, інструменти для аналізу відображення графічного інтерфейсу та безліч інших. Завдяки всім цим інструментам розробник має можливість знайти проблемні ділянки додатку та оптимізувати їх.

1.5 Фреймворк UIKit

UIKit це iOS фреймворк який забезпечує необхідні інфраструктуру для iOS додатків, надає архітектуру для реалізації графічного інтерфейсу, а також основний цикл запуску, необхідний для взаємодії між користувачем, операційною системою та додатком.

UIKit для відображення графічних елементів використовує Auto-Layout. Auto-Layout динамічно обчислює розмір і положення всіх графічних елементів у вашій ієрархії представлень, покладаючись на обмеження, розміщених на цих представленнях. Наприклад, ви можете обмежити кнопку так, щоб вона була горизонтально розташована по центру зображення, і щоб верхній край кнопки завжди залишався на 8 пунктів нижче нижньої частини зображення. Якщо розмір або положення перегляду зображення змінюються, положення кнопки автоматично перераховується.

Такий підхід до проектування на основі обмежень дозволяє створювати інтерфейси користувача, які динамічно реагують як на внутрішні, так і на зовнішні

зміни.

1.6 Мова програмування SQL

В перекладі з англійської мови Structured Query Language – мова структурованих запитів. SQL являє собою декларативну мову програмування, що використовується для формування запитів до реляційних баз даних, а також для оновлення і керування ними.

SQL – одна з формальних мов, тобто засіб, за допомогою якого ви передаєте комп'ютеру інструкції, що називаються програмою. Програмне забезпечення вашої бази даних виконує цю програму, написану на мові SQL. Це значить, що СУБД виконує ті запити, які ви їй передали, і відображає результати їх роботи, у тому числі яке-небудь повідомлення про помилку. Треба сказати, що мови програмування, названі також формальними мовами, відрізняються від мов спілкування, названих неформальними або природними мовами, головним чином тим, що створюються під конкретну мету, повністю позбавлені двозначності, мають вельми обмежений словниковий запас і гнучкість. Таким чином, якщо ви не отримали результату від роботи своєї програми, на який розраховували при її написанні, це відбулося тому, що ваша програма містить яку-небудь помилку (логічну або синтаксичну - в останньому випадку, швидше за все, буде виведено відповідне повідомлення, що описує помилку), а не тому, що комп'ютер неправильно зрозумів ваші інструкції, формалізовані у вигляді програми.

1.7 СУБД MySQL

MySQL, на відміну від SQL, це не мова, а вільна система керування реляційними базами даних. СУБД MySQL є однією з найвідоміших, надійних і швидких з усіх існуючих СУБД. MySQL – компактний багатонитевий сервер баз даних. Характеризується великою швидкістю, стійкістю і простотою використання. MySQL написаний під десятки видів операційних систем. Це iFreeBSD, OpenBSD, MacOS, OS/2, SunOS, Win9x/00/NTiLinux. Сьогодні MySQL особливо поширена на платформах Linux і Windows.

Принцип роботи СУБД MySQL аналогічний принципу роботи будь-якої СУБД, що використовує SQL (Structured Query Language, мова структурованих запитів) як командної мови для створення/видалення баз даних, таблиць, для поповнення таблиць даними, для здійснення вибірки даних.

MySQL, як і будь-яка інша СУБД являє собою програму-сервер, яка знаходиться в пам'яті комп'ютера і обслуговує TCP порт. У випадку з MySQL, номером порту буде число 3306. А клієнтська програма, будь то CGI-додаток на Perl або програмний продукт на C, з'єднується з СУБД з цього порту і посилає йому рядки на SQL. Той у свою чергу їх інтерпретує, виконуючи необхідні дії, і відсилає результати запиту назад клієнтові. Таким способом відбувається спілкування сервера баз даних з клієнтськими програмами.

MySQL має розвинену систему доступу до баз даних. Користувачеві бази даних може бути наданий доступ до всієї бази даних, окремих таблиць і окремих стовпців таблиць. Є розмежування на дії, які може виробляти користувач із записами. Для організації такої, складною (на перший погляд) структури доступу використовується декілька таблиць в системній базі даних. На підставі значень цих таблиць налаштовується політика надання доступу [10].

База даних, яку сервер MySQL використовує для зберігання внутрішньої інформації про користувачів, за замовчуванням має ім'я mysql. У цій базі даних певні таблиці для зберігання інформації користувача облікових записів.

1.8 Веб-фреймворк Vapor

Vapor - це веб-фреймворк з відкритим кодом, який на написано на базі мови програмування Swift. Vapor використовує для роботи з інтернет мережею офіційний фреймворк Swift NIO, також створений компанією Apple. Він може бути використаний для створення API RESTful, веб-додатків та додатків у режимі реального часу за допомогою WebSockets. Також Vapor має ORM, мову шаблонів Leaf та бібліотеки полегшення аутентифікації та авторизації користувачів.

Фреймворк Leaf - шаблонна мова для створення динамічних HTML-сторінок для веб-сайту або для створення електронних листів, що надсилаються з API.

Фреймворк Fluent - швидкий та безпечний у використанні фреймворк ORM, створений для Swift, який забезпечує основу для побудови інтеграції баз даних.

РОЗДІЛ 2. ОРГАНІЗАЦІЯ РОЗРОБКИ WEB-ОРІЄНТОВАНОГО ДОДАТКУ

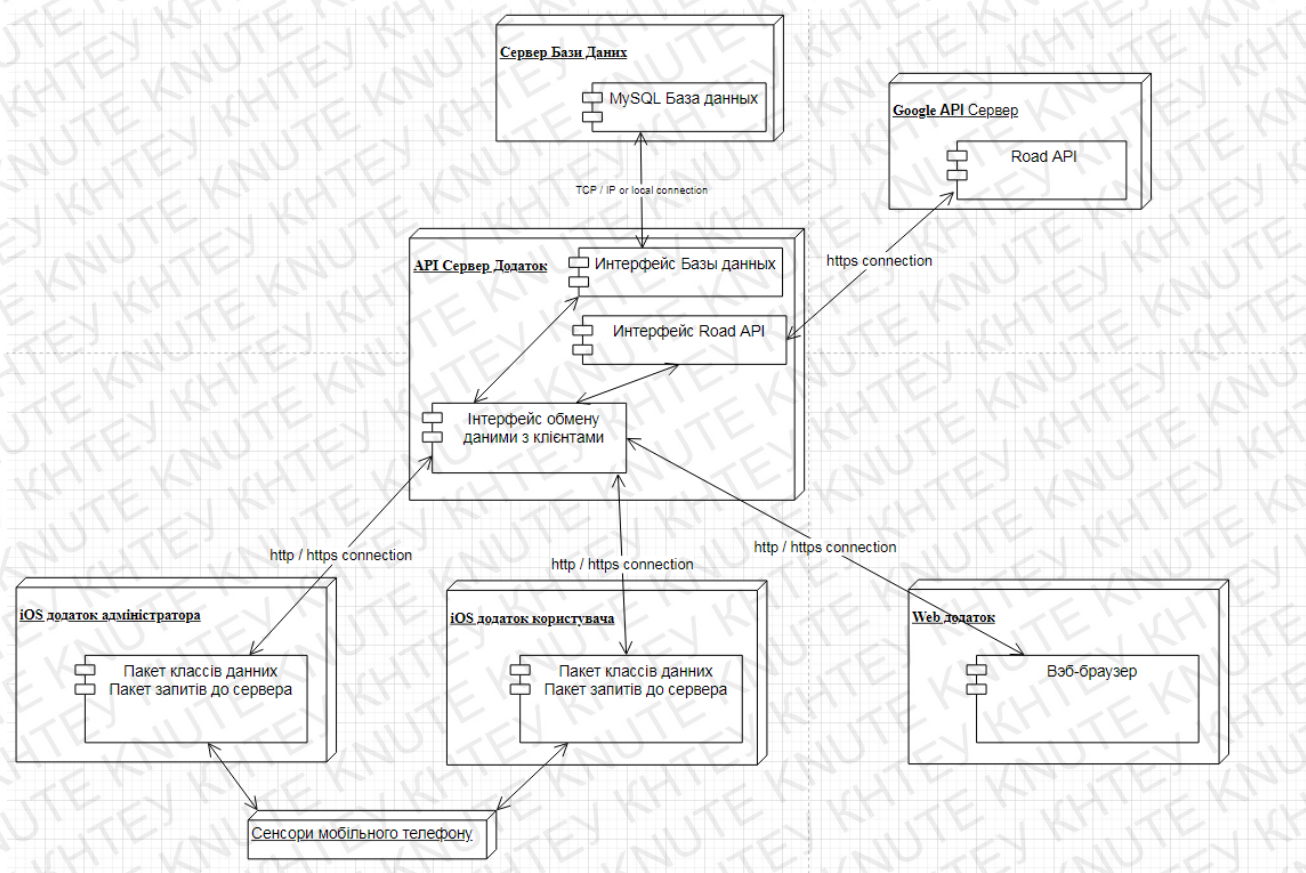
2.1. Структура програмного забезпечення.

Програмний продукт складається з двох мобільних додатків, API серверу, веб-додатку, бази даних та був побудований на клієнт серверній архітектурі з різними рівнями доступу. В додатку є два типи користувачів:

- Адміністратор – має доступ до веб-додатку, розширеної версії мобільного додатку та до всіх даних в БД.
- Користувач – має доступ лише до базової версії мобільного додатку

На рисунку 4.1 подано діаграму розгортання розробленої системи.

Діаграма розгортання системи показує обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Діаграма розгортання в UML нотації моделює фізичне розгортання артефактів на вузлах.



Ілюстрація 2.1 – Діаграма розгортання системи.

2.2. Опис бази даних

Для реалізації поставленої задачі було розроблено базу даних засобами мови програмування MySQL, яка складається з 12 таблиць.

Таблиця “Прискорення” містить інформацію про зчитану інформацію з акселерометра, що включає в себе індекс, та прискорення по трьом осям (таблиця 2.1.).

Таблиця 2.1. Структура таблиці “Прискорення”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Значення осі X	double	Прискорення по осі X
Значення осі Y	double	Прискорення по осі Y
Значення осі Z	double	Прискорення по осі Z

Таблиця “Розташування у просторі” (таблиця 2.2.) містить зчитану інформацію отриману через акселерометр, гіроскоп та магнітометр, орієнтація аерокосмічного апарата щодо інерціальної системи відліку.

Таблиця 2.2. Структура таблиці “Розташування у просторі”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Значення pitch	double	Поперечна вісь
Значення yaw	double	Вертикальна вісь
Значення roll	double	Поздовжня вісь

Таблиця “Вектор Гравітації” (таблиця 2.3.) містить зчитану інформацію з акселерометра та магнітометра, що являє собою нормований вектор сили тяжіння.

Таблиця 2.3. Структура таблиці “Вектор Гравітації”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Значення осі X	double	Сила тяжіння відносно осі X
Значення осі Y	double	Сила тяжіння відносно осі Y
Значення осі Z	double	Сила тяжіння відносно осі Z

Таблиця “Швидкість обертання” (таблиця 2.4.) містить зчитану інформацію з акселерометра та магнітометра, що являє собою нормований вектор сили тяжіння.

Таблиця 2.4. Структура таблиці “Швидкість обертання”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Значення осі X	double	Швидкість обертання по осі X
Значення осі Y	double	Швидкість обертання по осі Y
Значення осі Z	double	Швидкість обертання по осі Z

Таблиця “Локація” (таблиця 2.5.) містить зчитану інформацію з GPS

Таблиця 2.5. Структура таблиці “Локація”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Довгота	double	Довгота
Широта	double	Широта
Похибка місця знаходження	double	Швидкість обертання по осі Z
Курс пересування	double	Напрямок пересування у радіанах
Швидкість	double	Швидкість пересування (км\год)
Дата коли дані були отримані	datetime	Дата коли дані були отримані

Таблиця “Елемент запису сесії” (таблиця 2.6.) містить зчитану з усіх сенсорів телефону.

Таблиця 2.6. Структура таблиці “Елемент запису сесії”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Ідентифікатор сесії	int	Ідентифікатор для зв'язку з таблицею “Сесія”
Ідентифікатор прискорення	int	Ідентифікатор для зв'язку з таблицею “Прискорення”
Ідентифікатор розташування у просторі	int	Ідентифікатор для зв'язку з таблицею “Розташування у просторі”
Ідентифікатор вектору гравітації	int	Ідентифікатор для зв'язку з таблицею “Вектор гравітації”
Ідентифікатор швидкості обертання	int	Ідентифікатор для зв'язку з таблицею “Швидкість обертання”
Ідентифікатор локації	int	Ідентифікатор для зв'язку з таблицею “Локація”
Час коли дані були отримані	datetime	Час коли дані були отримані
Часова мітка відносно сесії запису	double	Час пройдений відносно початку запису

Таблиця “Сесія” (таблиця 2.7.) містить зчитану з усіх сенсорів телефону.
Таблиця 2.7. Структура таблиці “Сесія”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Назва	string	Назва для класифікації подій на даній ділянці
Частота оновлення даних сенсорів	double	Частота з якою дані сенсорів оновлюються
Час коли почалась сесія	datetime	Час коли почалась сесія
Ідентифікатор користувача від якого отримано запис	int	Ідентифікатор для зв'язку з таблицею “Користувач”
Ідентифікатор чи проіндексовані дані	tinyint	Флаг для перевірки чи проаналізовані дані

Таблиця “Користувач” (таблиця 2.8.) містить дані про користувача.
Таблиця 2.8. Структура таблиці “Користувач”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Електронна пошта	double	Електронна пошта, використовується для авторизації користувача
Зашифрований пароль	double	Зашифрований пароль, використовується для авторизації користувача
Роль	double	Роль користувача у системі

Таблиця “Сесія користувача” (таблиця 2.9.) містить дані сесії авторизації.
Таблиця 2.9. Структура таблиці “Сесія користувача”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Значення ідентифікатору	string	Значення ідентифікатору для підтвердження сесії
Ідентифікатор користувача	int	Ідентифікатор для зв'язку з таблицею “Користувач”
Дата до якої дійсна сесія	datetime	Дата після якою сесія вважається недійсною

Таблиця “Вузол графа дороги” (таблиця 2.10.) містить вузол графа для побудови графа доріг.

Таблиця 2.10. Структура таблиці “Вузол графа дороги”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Широта	double	Широта
Довгота	double	Довгота

Таблиця “Ребро графа стану доріг” (таблиця 2.11.) містить ребро графа для побудови графа стану доріг.

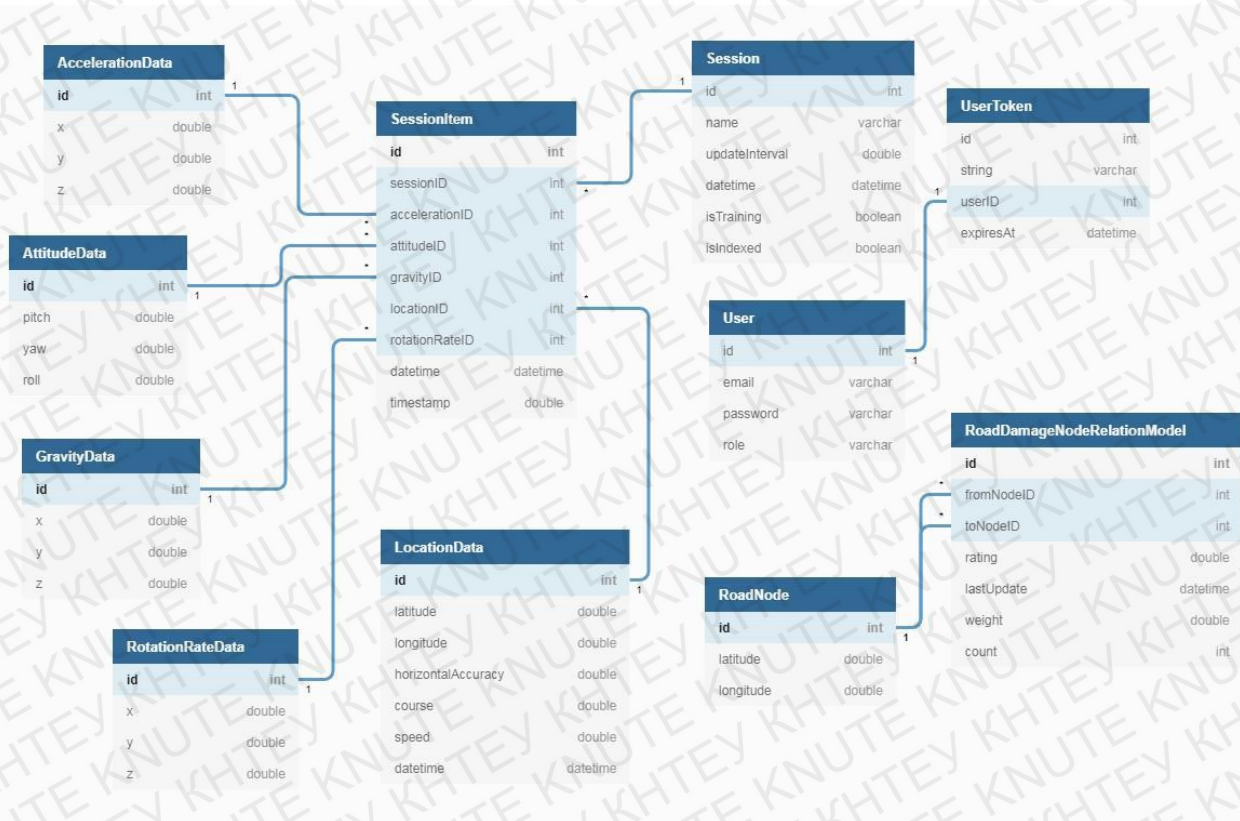
Таблиця 2.11. Структура таблиці “Ребро”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Ідентифікатор першого вузла	double	Ідентифікатор для зв'язку з таблицею “Вузол графа дороги”
Ідентифікатор другого вузла	double	Ідентифікатор для зв'язку з таблицею “Вузол графа дороги”
Оцінка стану дороги	double	Оцінка стану дороги
Дата попереднього оновлення	datetime	Дата попереднього оновлення, потрібно для перерахунку оцінки при наступній обробці
Кількість обробок	int	Кількість обробок, потрібно для перерахунку оцінки при наступній обробці
Вага обробок	double	Вага обробок, потрібно для перерахунку оцінки при наступній обробці

Таблиця 2.12. Структура таблиці “Історія дій в системі”

Назва поля	Тип поля	Призначення
Ідентифікатор	int	Автоінкремент, унікальне поле в таблиці
Ідентифікатор виконавця	double	Ідентифікатор для зв'язку з таблицею “Користувач”
Опис	double	Опис події в системі

На ілюстрації 2.2 подано схему бази даних зі зв'язками.



Таблиця “Історія дій в системі” (таблиця 2.12.) містить історію дій в системі.

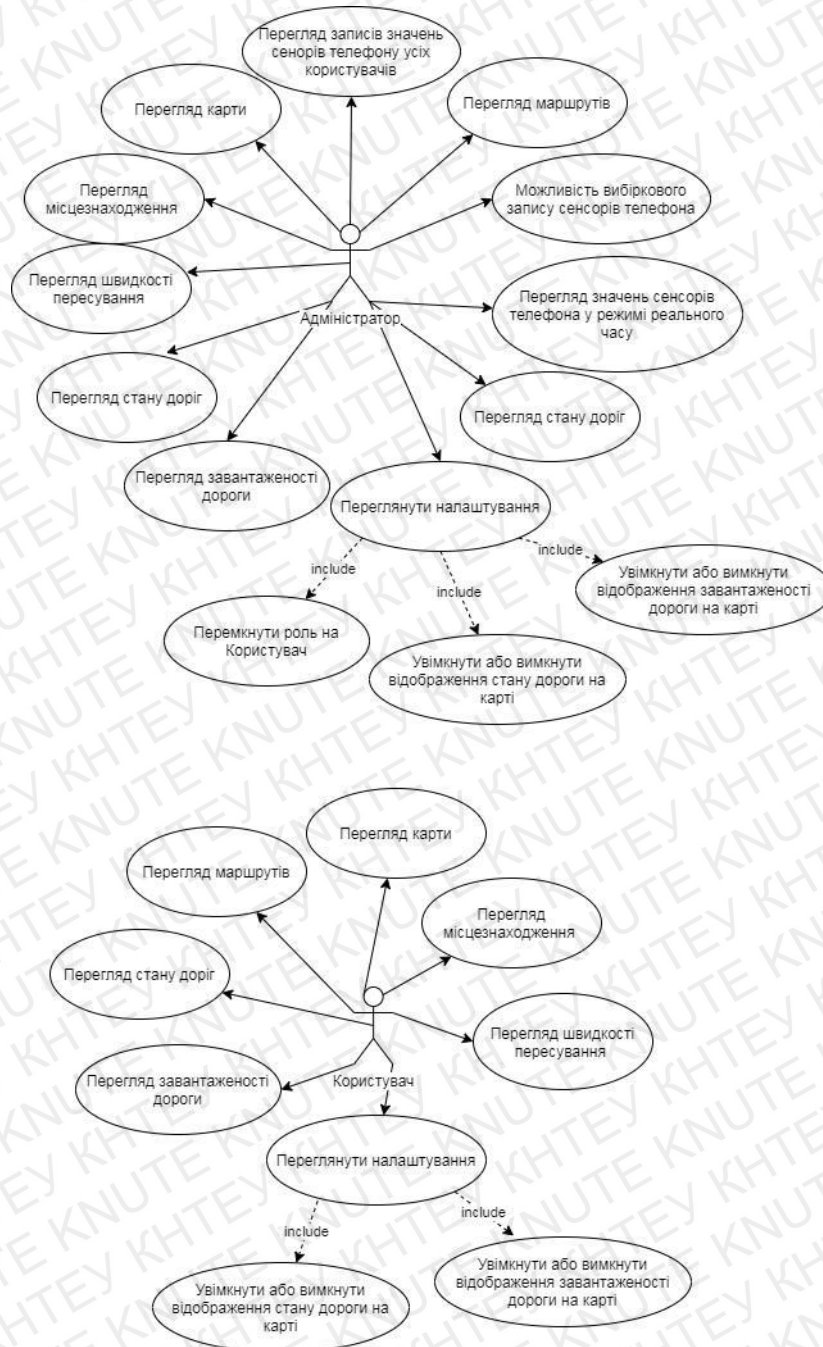
Ілюстрація 2.12 – Схема бази даних.

Схема бази даних представляє собою структура системи, що розробляється. База

даних описана формальною мовою, яка підтримується системою керування баз даних (СКБД).

2.3. Діаграма прецедентів системи

На ілюстрації 2.1 подано діаграму прецедентів системи.

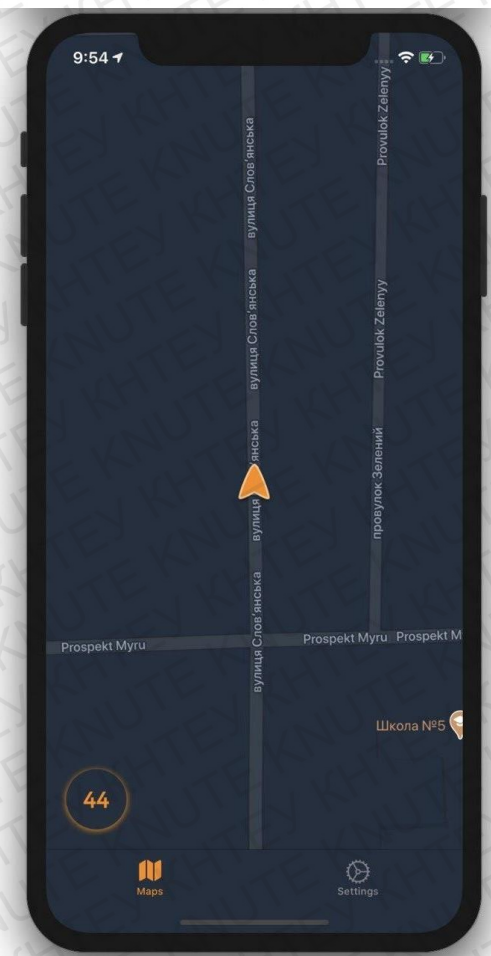
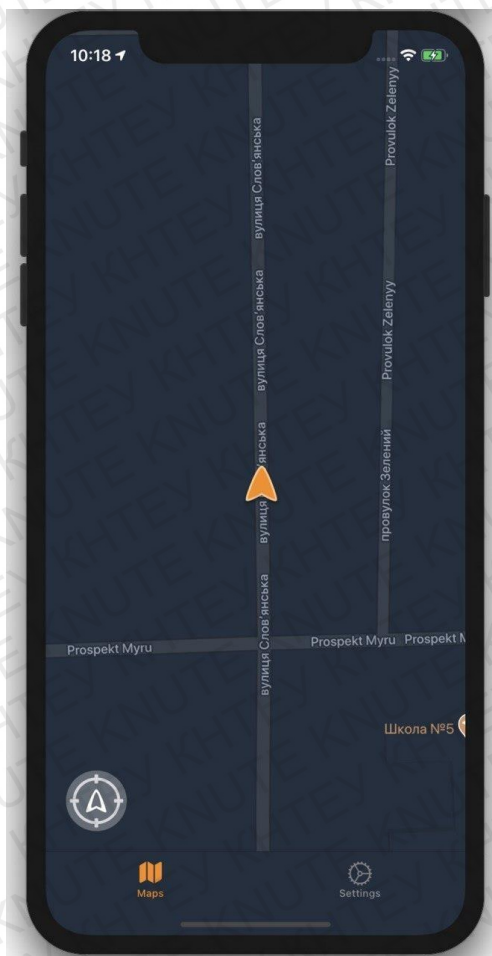


Ілюстрація 2.1 – Діаграма прецедентів системи

РОЗДІЛ 3. ПРОГРАМНА ОРГАНІЗАЦІЯ РОЗРОБКИ WEB- ОРІЄНТОВАНОГО ДОДАТКУ

3.1 Мобільний додаток користувача

При вході до мобільного додатку, додаток запитує дозвіл на використання геопозиції користувача. У випадку, коли користувач відмовиться, додаток не буде коректно працювати. Якщо користувач дозволить використання геоданих, то його зустрічає географічна мапа, яка демонструє його поточне місцезнаходження. Візуалізацію подано на рисунках 3.1 – 3.2.

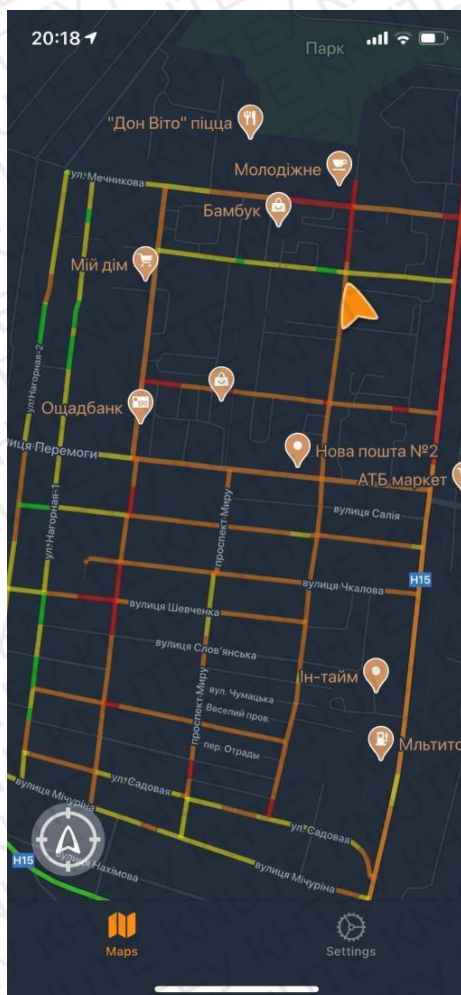


Ілюстрація 3.1 – Мапа у неактивному стані
Ілюстрація 3.2 – Мапа у стані подорожі
У нижньому кутку екрану можна побачити кнопку "Закріпити камеру за користувачем", після натискання на неї, додаток переходить у стан подорожі,

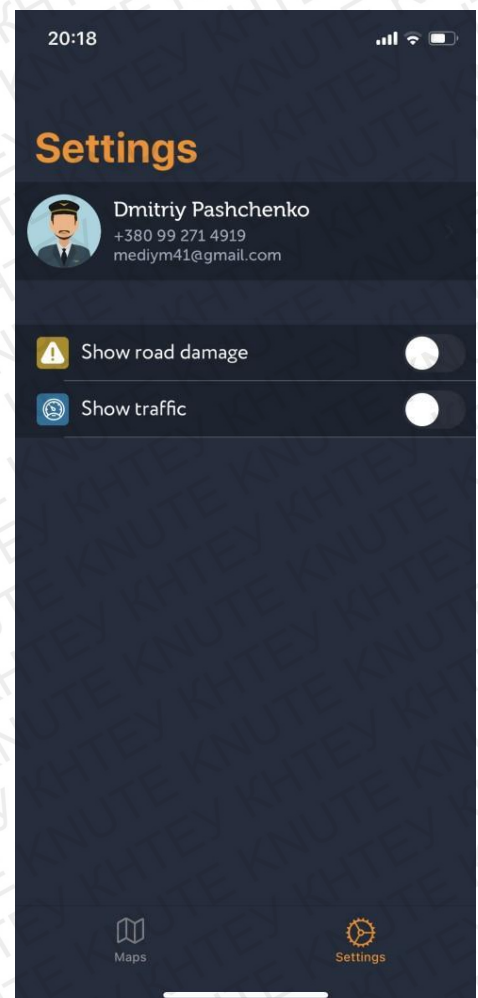
кнопка навігації замінюється спідометром (Ілюстрація 3.2), сліду за користувачем на карті. У цьому стані додаток паралельно починає слідкувати за даними сенсорів і якщо вони проходять фільтрацію, то відправляються на сервер, для подальшого аналізу. Якщо користувач проведе (свайпне) по карті додаток перейде у неактивний стан (Ілюстрація 3.1).

Також користувач може побачити стан деяких відрізків дороги, де вже було зібрано достатньо даних для присвоєння статусу (Ілюстрація 3.3). В залежності від стану дороги, стежку, на екрані, буде виділено відповідним кольором (зелений – дорога в гарному стані, жовтий – дорога нормальна, помаранчевий – стан дороги поганий, червоний – стан дороги дуже поганий)

Друга навігаційна вкладка є Налаштування, після натискання на неї, користувач потрапляє на список усіх можливих налаштувань додатку (Ілюстрація 3.4).



Ілюстрація 3.3 – Стан дороги



Ілюстрація 3.4 -Налаштування у користувача

Користувач має дві опції для перемикання: відображати стан дороги, відображати завантаженість доріг. Якщо перемикач увімкнено, то відповідні елементи будуть відображатись карті (на першій навігацій вкладки Карти).

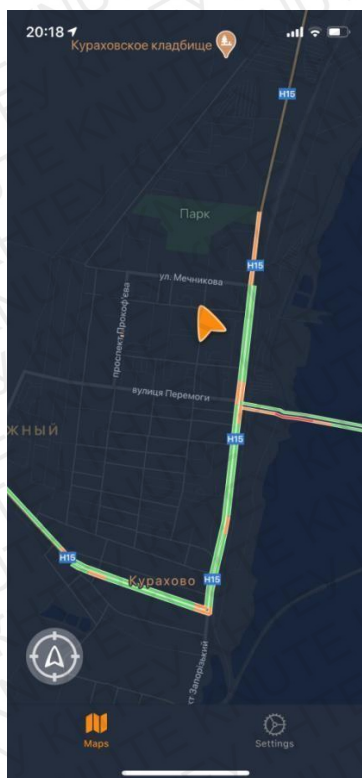
Коли перемикач Відображати увімкненою, то на вкладці Карти можна побачити завантаженість деяких відрізків дороги, де вже було зібрано достатньо даних для присвоєння статусу(Ілюстрація 3.5). В залежності від завантаженості дороги, стежку, на екрані, буде виділено відповідним кольором (зелений – дорога вільна, жовтий – дорога мало завантажена, помаранчевий – дорога середнє завантажена, красний – дорога сильно завантажена)

Також у додатку, як у всіх сучасних навігаторах, є можливість прокласти маршрут до певного місця. Для цього користувачу потрібно здійснити довге натискання на точці на карті, в яку він хоче потрапити. Додаток поставить в цій точці мітку, прокладе до неї маршрут, покаже інформацію про побудований маршрут (приблизна тривалість подорожі, дистанція до точки призначення, та приблизний стан дороги через який він пролягає)(Ілюстрація 3.6).

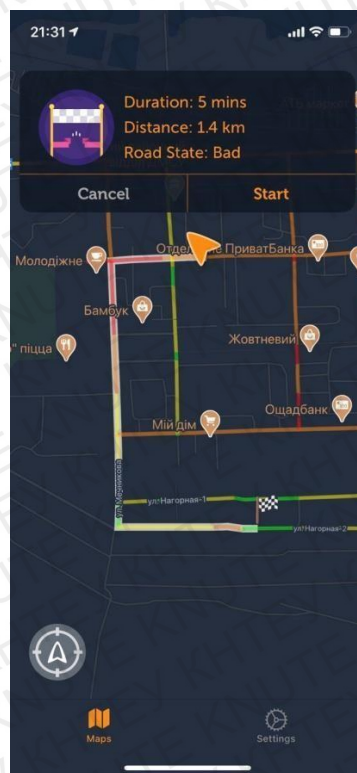
Щоб розпочати подорож, користувачу потрібно натиснути кнопку Start(Старт), та у протилежному випадку, кнопку Cancel(Відмінити).

Якщо користувач натисне кнопку Start(Старт), то додаток перейде у режимі та прибере модальне вікно з інформацією про маршрут(Ілюстрація 3.7).

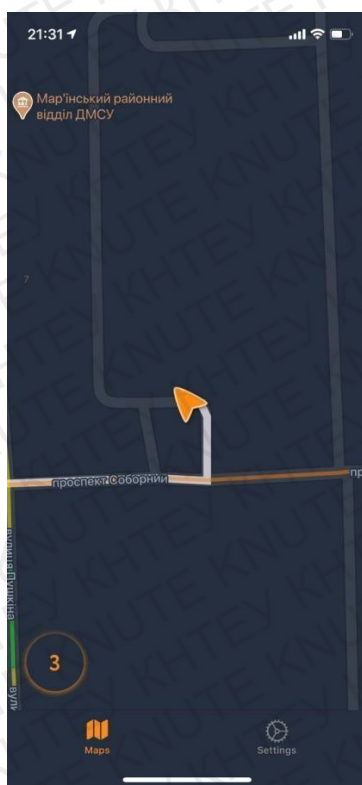
Якщо користувач натисне кнопку Cancel(Відмінити), то додаток прибере усі графічні елементи, відносно маршруту. Щоб прибрати маршрут, користувачу потрібно натиснути на маркер, котрий встановився після прокладки маршруту, після чого з'явиться модальне вікно, у котрому користувачу потрібно підтвердити своє рішення, натиснувши кнопку Stop(Стоп)(Ілюстрація 3.8).



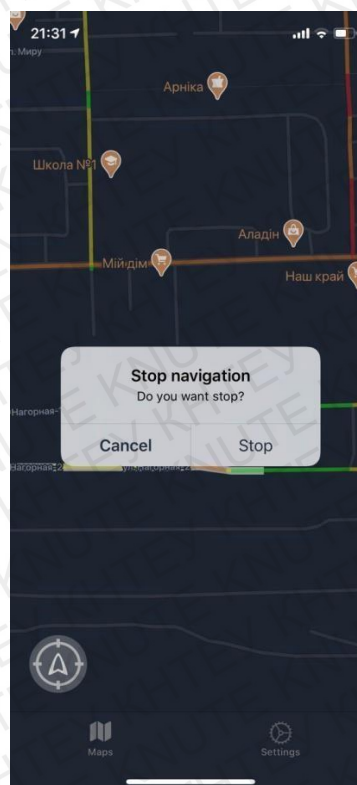
Ілюстрація 3.5 – Завантаженість на дорозі



Ілюстрація 3.6 – Прокладення маршруту та початок подорожі



Ілюстрація 3.7– Режим подорожі після прокладання маршруту

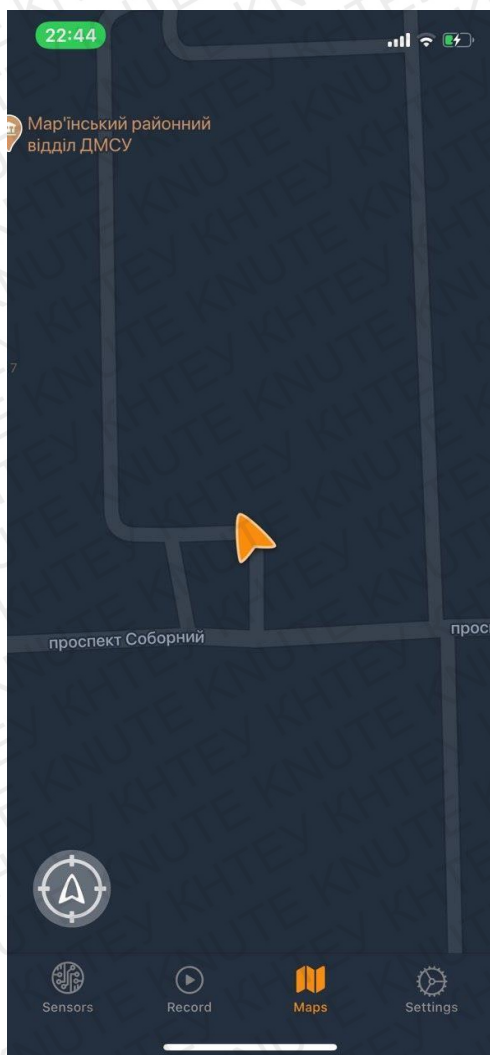


Ілюстрація 3.8 – Модальне вікно для скасування прокладеного маршруту

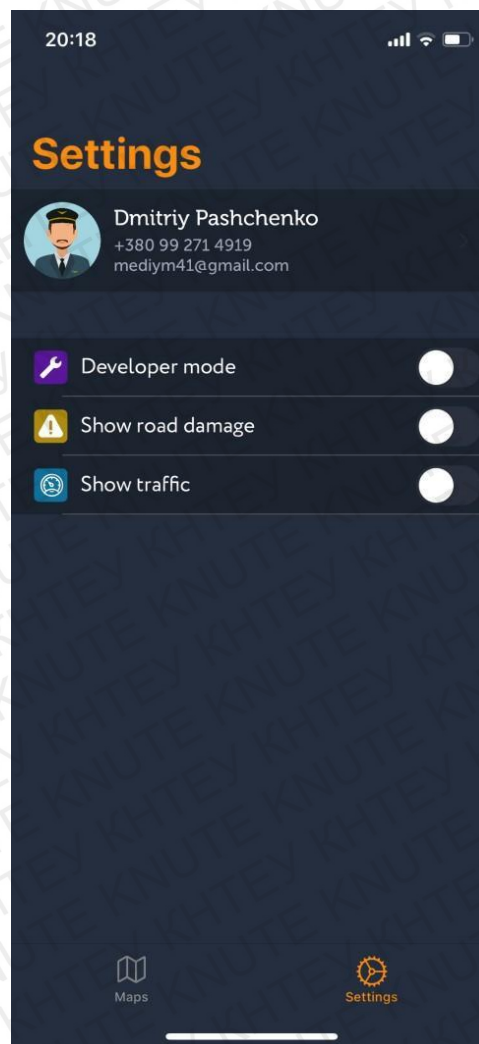
3.2 Розширена версія мобільного додатку

Розширена версія має такий самий інтерфейс та схему взаємодії, як і в базовому додатку. Проте, в ньому присутні декілька додаткових вкладок (Sensors(Сенсори) та Record (Запис). Її подано на рисунку 3.9.

На ілюстрації 3.10 проілюстровано налаштування у адміністратора.



Ілюстрація 3.9 – Розширена версія додатку



Ілюстрація 3.10 – Налаштування у адміністратора

Вкладка Sensors(Сенсори) дозволяє користувачу додатку переглядати значення показників, отриманих від сенсорів(Ілюстрації 3.11-3.14). Тим самим імітувати певні патерни поведінки.



Ілюстрація 3.11 – Значення Attitude



Ілюстрація 3.12 – Значення Rotation

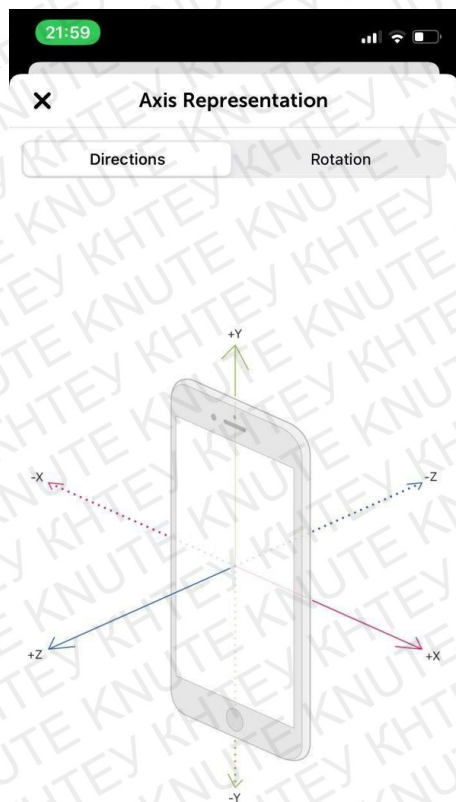


Ілюстрація 3.13 – Значення Gravity

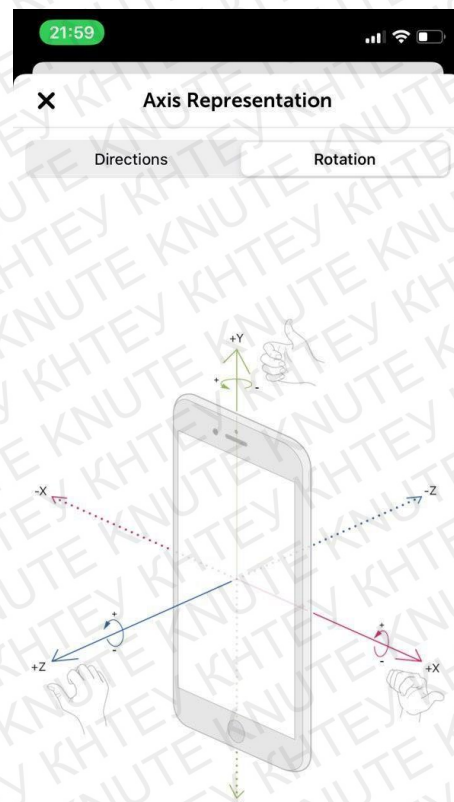


Ілюстрація 3.14 – Значення Acceleration

Для полегшення розуміння розміщення векторів(x , y , z), було додано вкладку Info(Інформація), на котрій можна переглянути візуальну інтерпретація векторів(Ілюстрації 3.15-3.16).



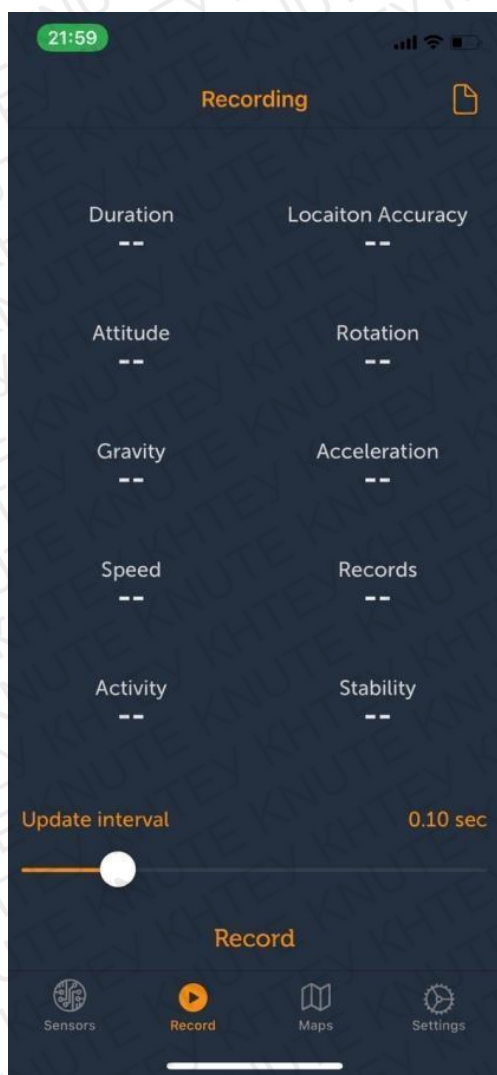
Ілюстрація 3.15 – Візуальна інтерпретація напрямків осей



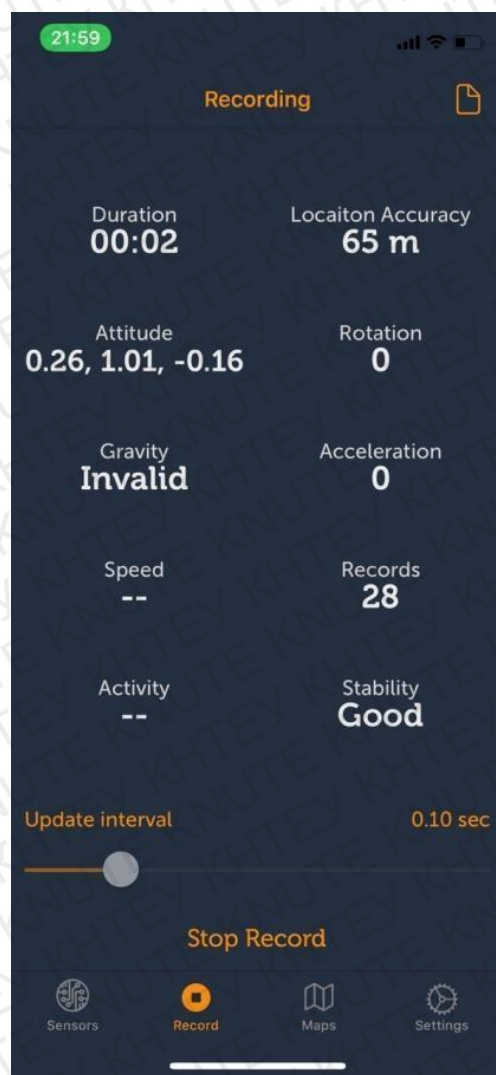
Ілюстрація 3.16 – Візуальна інтерпретація напрямків обертання

Вкладка Records дозволяє користувачу додатку записувати конкретні ділянки, та спостерігати за важливими даними у режимі реального часу. Усе що потрібно це вибрати певну частоту оновлення даних та натиснути кнопку Record(Запис), в наслідок чого, екран заповниться поточними даними.

Для того, щоб зупинити запис потрібно натиснути кнопку Stop(Стоп), після чого додаток покаже модальний діалог у якому потрібно ввести ім'я файлу, до якого буде записано дані (Ілюстрації 3.17 – 3.19).

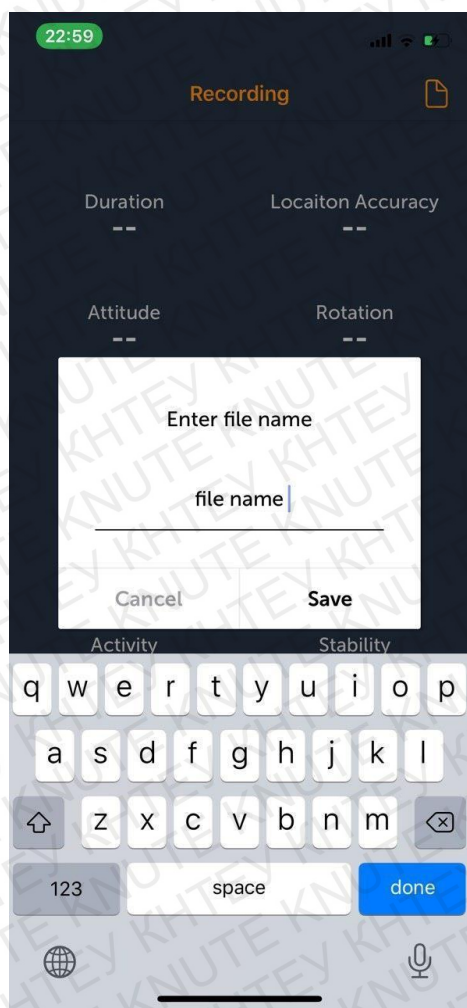


Ілюстрація 3.17 – Неактивний стан



Ілюстрація 3.18 – Активний стан

Після цього файл збережеться локально, щоб його кудись відправити потрібно натиснути кнопку файл, що розташована у верхньому правому куті екрану. З'явиться список усіх записів, у якому буде відображено, у тому числі й тільки що створений файл. Його потрібно обрати та вибрати місце призначення відправлення. Щоб видалити записи потрібно натиснути кнопку Clear(Очистити)(. Ілюстрація 3.20).



Ілюстрація. 3.19 – Збереження запису

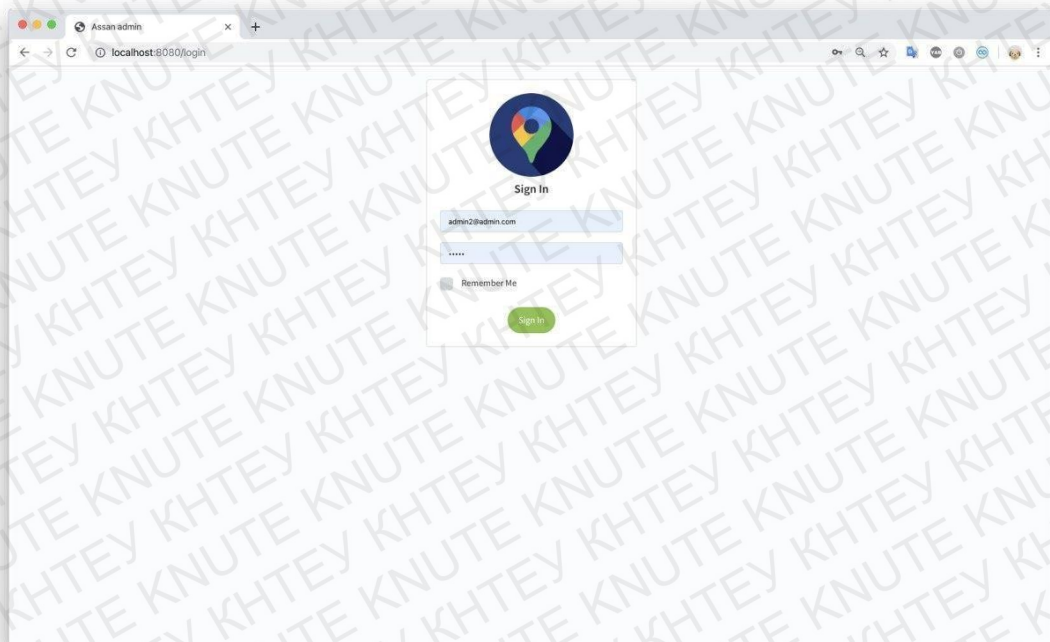


Ілюстрація 3.20 – Список записів які зберігаються локально

3.3 Веб-додаток

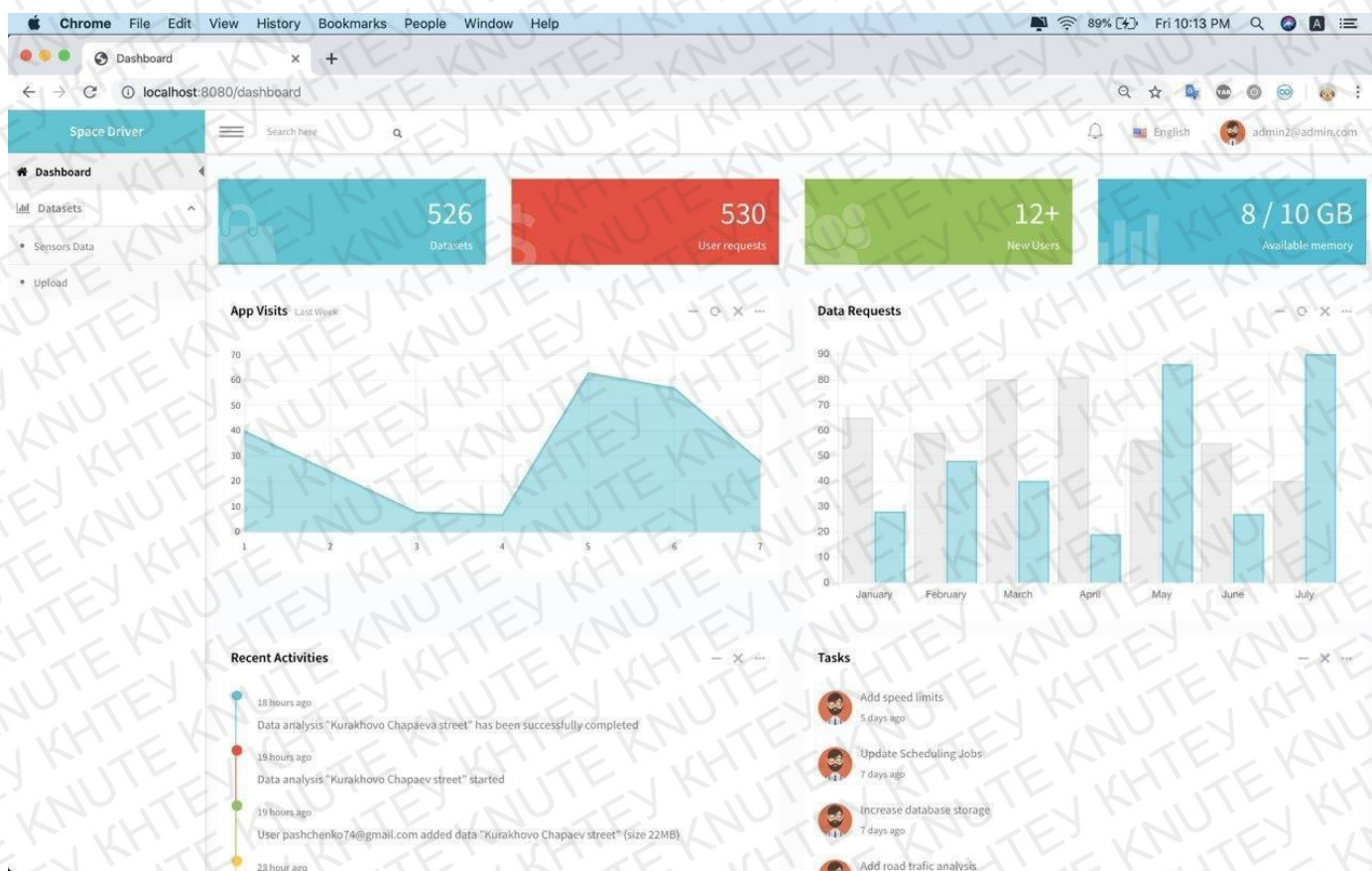
Веб-додаток призначений для перегляду даних, в зручному для аналізу вигляду. Для початку роботи з даними, користувачу потрібно пройти процедуру авторизації, а саме ввести електронну пошту та пароль. Після натискання кнопки Sign In (Авторизуватися), користувач може розпочати роботу з додатком.

Форму авторизації, наведено на ілюстрації 3.21.



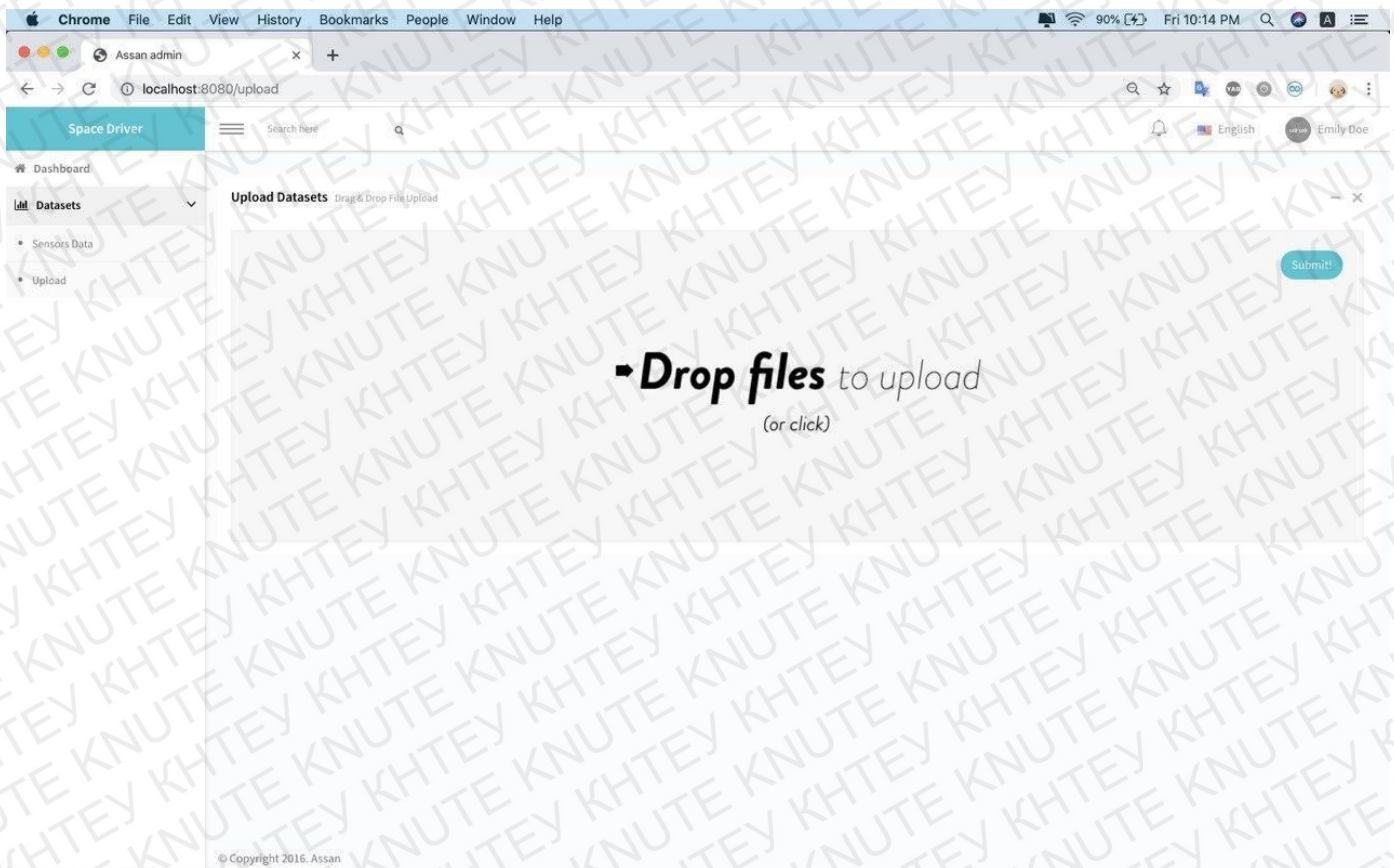
Ілюстрація 3.21 – Форма авторизації.

На ілюстрації 3.22 Наведено вкладку Dashboard(Панель приладів). Вона показує загальний стан систем



Ілюстрація 3.22 – Dashboard.

На ілюстрації 3.23 наведено вкладку Datasets (Набори даних). На вкладці користувач може додати файл з даними. На ілюстрації 3.23 наведено ситуацію, коли ще не додано жодного файлу.



Ілюстрація 3.23 – Datasets.

На ілюстрації 3.24 приведено вкладку Datasets(Набори даних), на якій можна обрати таблиці даних. Інтерфейс має можливість сортування та фільтрації. Натиснувши на будь-який заголовок стовпця, додаток відсортує вибірку за зростанням по відповідному стовпцю. Якщо повторити цю операцію, додаток відсортує по тим же критеріям, але вже за спаданням. Щоб знайти запис с певною назвою, потрібно ввести текст, що потребується, після чого додаток прибере усі записи, котрі не містять введений текст. Якщо записів буде дуже багато, то користувач може скористатись пагінацією або розширити кількість елементів у списку.

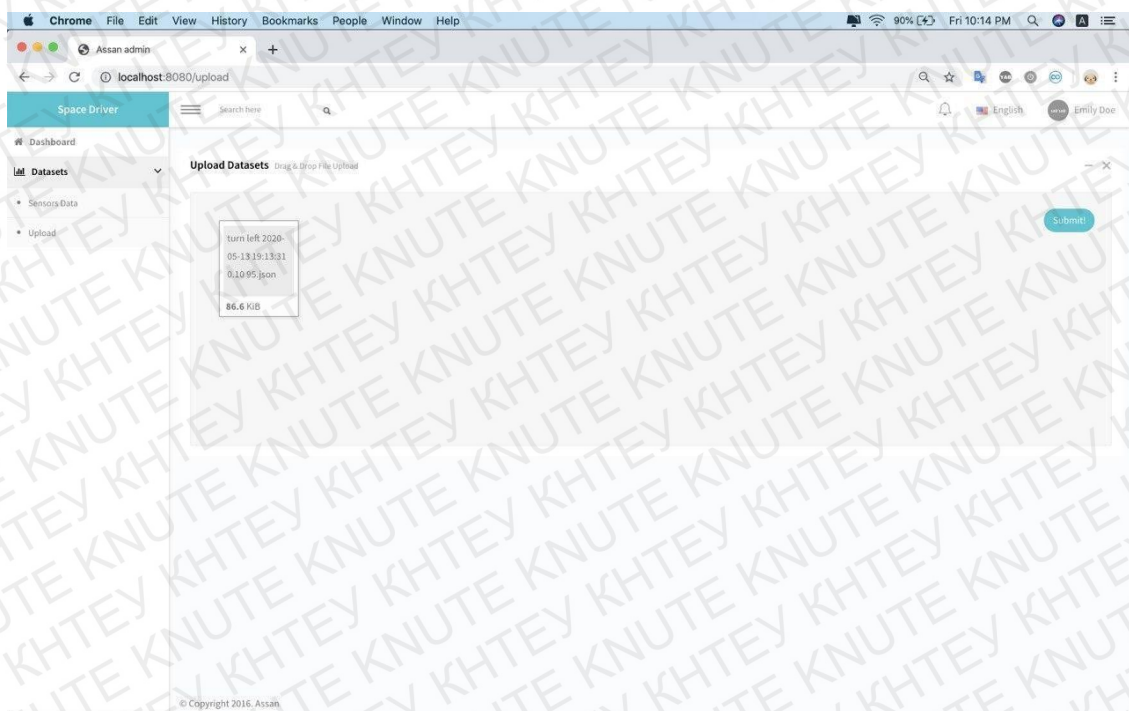
The screenshot shows the 'Datasets' page in the Space Driver application. A sidebar on the left contains 'Dashboard', 'Datasets' (selected), 'Sensors Data', and 'Upload'. The main area displays a 'Data Table' with 16 entries. The table has columns for an index, Name, Interval, Date, and Is Training. The first two entries are '(Updated) 89'. The last entry is '5+'. A search bar is located in the top right of the table area.

#	Name	Interval	Date	Is Training
1	(Updated) 89	0.10	2020-05-13 19:15:01	✓
2	(Updated) 89	0.10	2020-05-13 19:15:01	✓
3	89	0.10	2020-05-13 19:15:01	✓
4	89	0.10	2020-05-13 19:15:01	✓
5	89	0.10	2020-05-13 19:15:01	✓
6	89	0.10	2020-05-13 19:15:01	✓
7	89	0.10	2020-05-13 19:15:01	✓
8	89	0.10	2020-05-13 19:15:01	✓
9	89	0.10	2020-05-13 19:15:01	✓
10	6 plus sb	0.10	2020-05-13 19:15:16	✓
11	5-6 plus stop mega	0.10	2020-05-13 19:10:15	✓
12	7	0.10	2020-05-13 19:30:35	✓
13	2-3 then 4	0.10	2020-05-13 19:08:51	✓
14	5 and speed bump	0.10	2020-05-13 19:10:52	✓
15	6.5	0.10	2020-05-13 19:16:28	✓
16	5+	0.10	2020-05-13 19:14:22	✓

Showing 1 to 16 of 16 entries

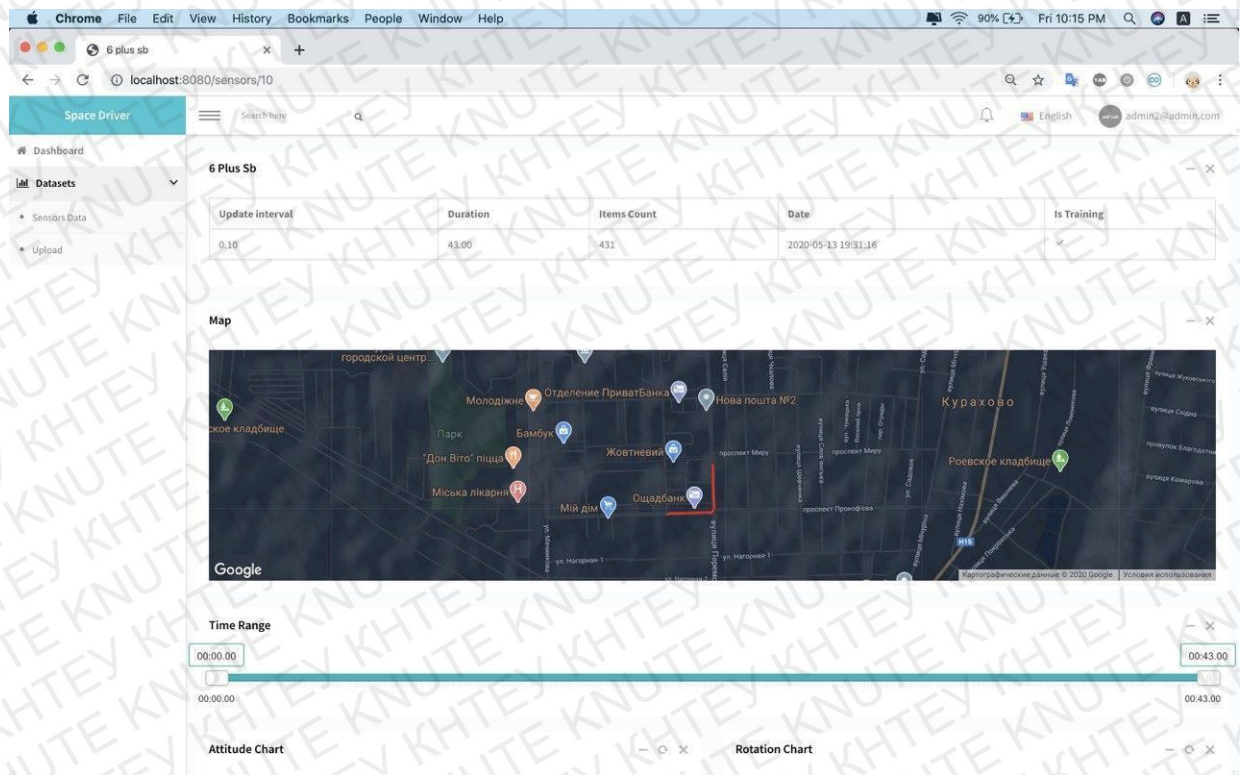
Ілюстрація 3.24 – Datasets (Data Table).

На вкладці Upload можна завантажити зібрані дані вручну. Для цього потрібно перетягнути файл у відповідну зону та відпустити або натиснути кнопку Submit та вибрати потрібний файл у модальному вікні. На рисунку 5.25 показано випадок, коли дані вже завантажено.



Ілюстрація 3.25 – Datasets (Upload Datasets).

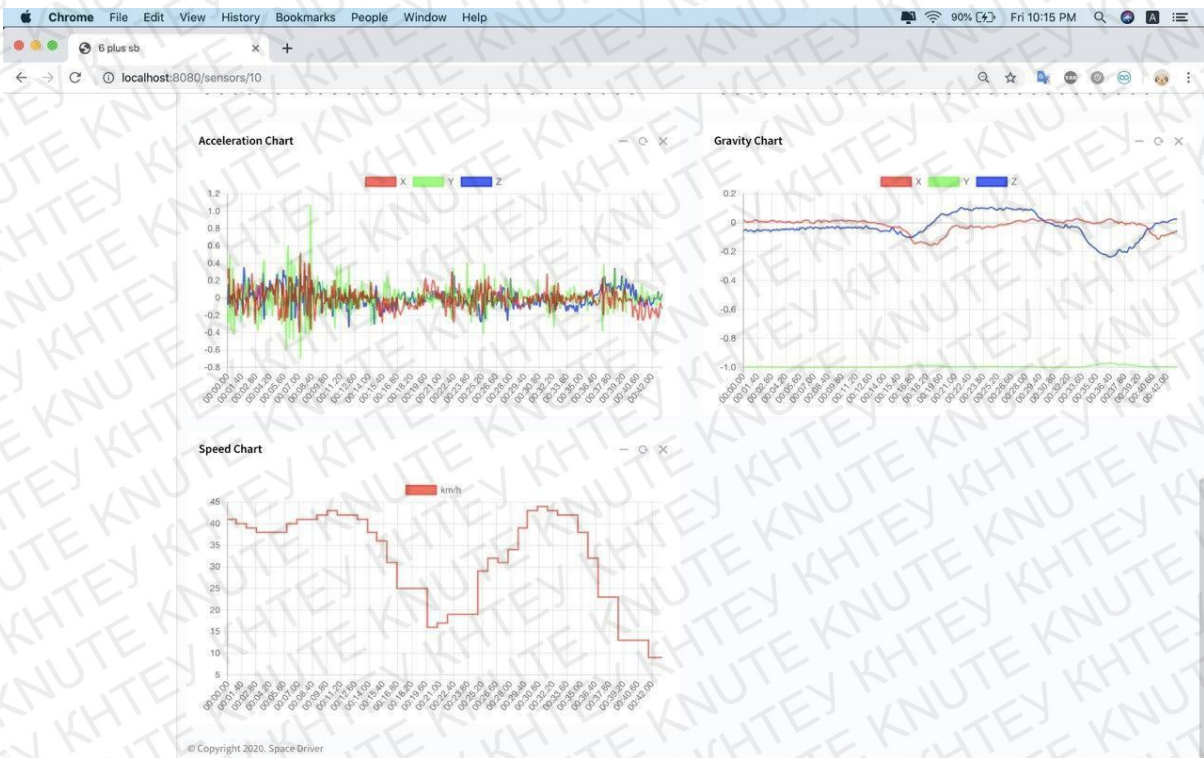
На ілюстрації 3.26 – 3.28 наведено показники у вигляді графіків та діаграм, за конкретною таблицею-записом. Тут можна переглянути більш детально інформацію, отриману від сенсорів, на мобільних пристроях.



Ілюстрація 3.26 – Datasets (Upload Datasets).



Ілюстрація 3.27 – Datasets (Upload Datasets).



Ілюстрація 3.28 – Datasets (Upload Datasets).

ВИСНОВКИ ТА РЕЗУЛЬТАТИ

У даній роботі було вивчено і проаналізовано проблему збору та аналізу даних у сфері транспорту.

Було проведено огляд методів і засобів розробки програмної системи, а саме: мови програмування Swift, JavaScript, а також SQL. Для роботи було обрано середовище розробки xCode. Також використано СУБД MySQL та фреймворк Vapor.

Розглянуто систему для зчитування даних з датчиків смартфонів та аналізу інформації для подальшого використання в галузі транспорту.

Розроблена архітектура програмного продукту, яка складається з бази даних, веб-програми та двох мобільних додатків. Один мобільний додаток призначений для звичайних користувачів системного драйвера, а другий - для системних адміністраторів і має розширені функції.

Отже було виконано всі поставлені задачі, а саме:

1. Розглянуто мобільний додаток клієнта (водія), який виступає у ролі навігатора, та надає доступ до інформації про стан доріг.
2. Розглянуто мобільний додаток адміністратора, який базується на мобільному додатку клієнта, але з розширеними можливостями: відслідковування значень сенсорів у режимі реального часу, з їх візуальною інтерпретацією, та ручний режим запису даних з більшими можливостями.
3. Створено веб-додаток, який складається з форми авторизації (доступно тільки для адміністраторів), головного екрану на якому відображається стан системи, списку записаних даних сенсорів, екрану з детальною інформацією запису даних сенсорів, екран ручного завантаження даних сенсорів.
4. Створено базу даних, яка зберігає інформацію про користувачів системи, дані, отримані від сенсорів, дані аналізу стану доріг, а також дані, що потребуються для авторизації, історія дій в системі.

Проведено тестування програмного забезпечення в умовах міста та отримано ряд результатів, на основі яких побудовано мапу стану дороги. Програмна система успішно пройшла апробацію, а її результати подано у звіті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. StarLine [Електронний ресурс]. – 1988. – Режим доступу до ресурсу: <https://starline.in.ua/>.
2. Пичаи С. Maps JavaScript API Goto Console [Електронний ресурс] / Сундар Пичаи – Режим доступу до ресурсу: <https://developers.google.com/maps/documentation/javascript/infolowings>.
3. Waze [Електронний ресурс]. – 2008. – Режим доступу до ресурсу: <https://www.waze.com/ru>.
4. Усов В. Swift. Основы Разработки приложений под iOS и macOS / Василий Усов., 2018. – 448 с.
5. Флэнаган Д. JavaScript. Карманный Справочник / Дэвид Флэнаган., 2020. – 320 с.
6. xCode [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/xcode/>.
7. UIKit [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.apple.com/documentation/uikit>.
8. Шварц Б. MySQL по максимуму. Оптимизация, репликация, резервное копирование / Бэрн Шварц., 2018. – 864 с.приложений / Ян Ф. Дарвин., 2019.
9. Vapor [Електронний ресурс] // 2016 – Режим доступу до ресурсу: <https://vapor.codes/>.
10. Google Maps Platform Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/maps/documentation>
11. Робин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – СПб.: Питер, 2016. – 768 с