

А.М. Ісаєв, магістрант
О.В. Кузьменко, ст. викладач
І.І. Сугоняк, к.т.н., доц.

Державний університет «Житомирська політехніка»

Дослідження та реалізація системи моніторингу стану автошляхів

Реалізовано програмний комплекс, що складається з мобільного додатка для збору даних і вебдодатка для збереження й відображення даних про пошкодження і стан автомобільних шляхів. Таке програмне забезпечення буде актуальним для використання у логістиці, навігаційних системах, комунальних та державних організаціях і просто у повсякденному житті автомобілістів. Під час проектування й розробки було виявлено та проаналізовано аналоги, що дало змогу виокремити функціональні можливості, якими має володіти система. Було розроблено алгоритм визначення пошкоджень дорожнього покриття, який дозволить у цифровому еквіваленті відобразити ступінь пошкодження та позначити його на карті за допомогою координат системи глобального позиціонування. Вказаний алгоритм використовує сенсори, що доступні у будь-якому сучасному смартфоні, такі як акселерометр, що робить програмне забезпечення широко доступним.

Ключові слова: пошкодження автошляхів; стан автошляхів; мобільний додаток; система глобального позиціонування; алгоритм.

Постановка проблеми. У сучасних реаліях транспортні витрати автомобілістів можуть містити додаткові ресурси на пальне та амортизацію, що безпосередньо пов'язано зі станом доріг. Сукупні витрати при цьому прямо залежать від кількості автомобілів, активізації діяльності громадського транспорту, вантажних автомобілів. Під час аналізу таких витрат потрібно враховувати тенденції розвитку автомобільного руху, в тому числі появу безпілотних автомобілів, доцільність використання яких обґрунтована екологічністю, ефективним використанням пального та економією на заробітній платі водія. Від якості доріг часто залежить кількість ДТП, збитки від яких потрібно відшкодовувати власникам в установленому порядку. Автоматизація визначення пошкодження доріг дасть можливість зменшити вплив цього фактора на транспортні витрати.

Аналіз останніх досліджень і публікацій. Проблемі розробки та впровадженню алгоритмів розподіленого моніторингу стану дорожнього покриття за допомогою сенсорів з мобільних телефонів присвячено низку наукових публікацій. Зокрема, в [13] було використано спектральний аналіз тривимірних сигналів прискорення для отримання надійних міток аномалій дорожнього покриття. Сигнали прискорення використовувалися для оцінки швидкості та зменшення помилок вибірки. Також було запропоновано усунення фактора швидкості для виявлення більш надійних ознак та представлено фреймворк для візуального аналізу передбачень і класифікацій на основі міток. У [14], для отримання даних про дорожнє покриття, автори запропонували метод поєднання даних з мікрофона й акселерометра для розрізнення низьких і високих амплітуд макротекстур на дорогах. Проводилися експерименти спільного використання тривимірних акселерометрів, інтегрованих у мобільні телефони, разом із приймачами GPS для розпізнавання аномалій дорожнього покриття [15–16]. Запропоновано систему збору й передачі інформації про стан доріг за допомогою трекерів у єдину базу даних геоінформаційної системи [17].

Перспективною задачею розробки алгоритмів розподіленого моніторингу стану дорожнього покриття є впровадження машинного навчання, що дає можливість отримувати найбільш релевантну інформацію про мітку, незалежно від специфіки апаратного забезпечення смартфона й особливостей автомобіля. Серед вітчизняних рішень авторами виокремлено технологію Navizor [18].

Метою дослідження є розробка алгоритму розпізнавання вад дорожнього покриття за даними мікроелектромеханічних систем смартфона (англ. Microelectromechanical systems) та реалізація веборієнтованої системи геопозиціонування знайдених дефектів з подальшою можливістю використання отриманих даних у державних, соціальних, приватних сферах.

Викладення основного матеріалу. Мета роботи обумовлює реалізацію алгоритму визначення стану доріг, збереження та відображення даних для їх подальшого застосування. Отримані дані можуть бути застосовані у майбутніх дослідженнях, побудові оптимальних маршрутів у навігаційних системах, інформуванні про ступінь зношеності дорожнього покриття комунальних та державних установ й інших напрямках підтримки прийняття рішень, які використовують стан доріг як критерій. Найкращим варіантом системи є клієнт-серверний проєкт у вигляді мобільного та вебдодатка з алгоритмом визначення стану дорожнього покриття. Для більш чіткого розуміння, як має працювати додаток і які потреби задовольняти, слід проаналізувати аналоги.

Аналоги. На сьогодні існує кілька програм, що аналізують стан доріг і відображають їх на карті. Наприклад, Waze [1], Spothole [2], SmartRoadSense [3] тощо. Розглянемо згадані більш детально.

Waze – навігаційний додаток для мобільних пристроїв, що дозволяє відстежувати ситуацію на дорогах у режимі реального часу, прокладати оптимальні маршрути, дізнаватися про розташування радарів швидкості, отримувати інформацію та попереджати інших користувачів про зміну дорожніх умов, перешкоди, поліцію, спілкуватися з іншими користувачами на карті.

Spothole – це програма з краудфандингом. Вона заохочує громадян повідомляти та відстежувати вибоїни у своєму місті. У свою чергу, Spothole дає змогу відповідним органам вжити заходи та відремонтувати вибоїни, роблячи дороги безпечними для всіх.

SmartRoadSense – це система безперервного моніторингу якості доріг. Система використовує акселерометри та датчики глобального позиціонування (GPS), що є на будь-якому сучасному смартфоні, для виявлення та класифікації пошкодження доріг. Дані збираються та обробляються на пристрої мобільним додатком.

Було розглянуто три системи, які мають одну мету – допомогти покращити якість доріг та проінформувати про їх стан. Проаналізувавши ці аналоги можна зробити висновок, що створена система має зрозумілий інтерфейс, збір даних відбувається майже без втручання людини, наявний аналіз стану доріг (від відмінної до незадовільної), відображає дані на карті в мобільному та вебдодатках.

Стек технологій. Значна частина проекту – це вебдодаток, за основу якого взято PHP-фреймворк Yii2 [4, 5], мову програмування PHP [6] та стек вебтехнологій HTML, CSS, JS. Мобільний додаток реалізовано на мові Java [7] з використанням Android SDK та Android Studio [8], яка є офіційним інтегрованим середовищем розробки для операційної системи Android від Google. Як систему управління базами даних (СУБД) для проекту було обрано MySQL [9] – систему керування реляційними базами даних з відкритим вихідним кодом, яка використовує мову структурованих запитів.

Система працює з геопозицією пристрою та має відображати зібрані дані на мапі. Для цього використовується відповідний JavaScript API [10], який дозволяє налаштовувати карти з власним вмістом і зображеннями для відображення на вебсторінках і мобільних пристроях. Також бібліотеки мап використовуються в обчисленнях алгоритму визначення стану автошляхів.

Алгоритм роботи. Система моніторингу автошляхів має автоматично, під час поїздки на авто, збирати дані про стан доріг. Це можливо, використовуючи смартфон з акселерометром та GPS, тому основний алгоритм роботи системи полягає в збиранні даних із сенсорів смартфона та їх обробці. Розглянемо алгоритм детально.

Акселерометр показує направлене прискорення смартфона уздовж осей Ox , Oy , Oz . Ці значення ми можемо отримати зі смартфона й позначимо їх як ax , ay , az на рисунку 1, *а* та у формулах. Фрагмент програмного коду отримання значень акселерометра наведено нижче.

```
this.ax=this.normalize(data.acceleration.x);
this.ay=this.normalize(data.acceleration.y);
this.az=this.normalize(data.acceleration.z);
```

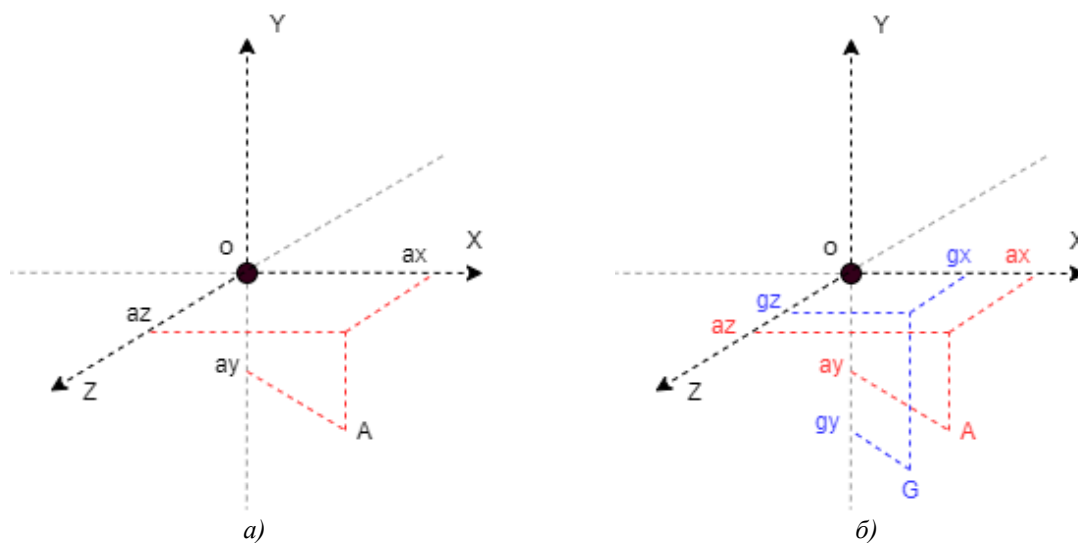


Рис. 1. Ілюстрація отриманих значень від акселерометра та сенсора гравітації

Нерівність на дорозі для смартфона буде прискоренням у сторону дорожнього покриття, тому необхідно визначити напрямок до дороги за допомогою сенсора гравітації. На рисунку 1, *б* та у формулах це позначено як gx , gy , gz . Фрагмент коду отримання значень сенсора гравітації наведено нижче.

```
this.gx=this.normalize(data.accelerationIncludingGravity.x);
this.gy=this.normalize(data.accelerationIncludingGravity.y);
this.gz=this.normalize(data.accelerationIncludingGravity.z);
```

За допомогою значень акселерометра та сенсора гравітації можливо побудувати вектори акселерації (Va) та гравітації (Vg) й обчислити числове значення струсу за допомогою проєкції (Pag) вектора акселерації на вектор гравітації [11]. Схематичне зображення алгоритму наведено на рисунку 2.

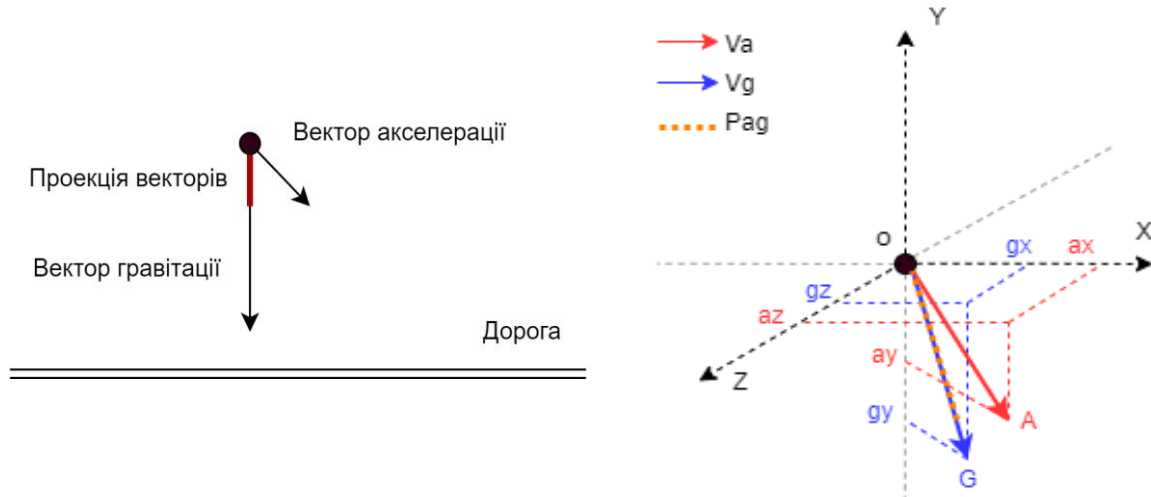


Рис. 2. Схеми роботи алгоритму та визначення проєкції векторів

Для визначення проєкції вектора акселерації на вектор гравітації необхідно добуток вектора акселерації і вектора гравітації поділити на добуток вектора гравітації помноженого на самого себе.

Обчислення наведено у формулі 1, де ax , ay , az – значення акселерометра, а gx , gy , gz – значення сенсора гравітації, Va – вектор акселерації, Vg – вектор гравітації, Pag – проєкція векторів.

$$Pag = \frac{Va \times Vg}{Vg \times Vg} = \frac{ax \times gx + ay \times gy + az \times gz}{gx \times gx + gy \times gy + gz \times gz} \quad (1)$$

У програмному кодї для обчислень проєкції векторів реалізовано клас Vector. Клас і код обчислення наведено нижче.

```
class Vector{
    constructor(x, y, z) {
        this.x=x;
        this.y=y;
        this.z=z;
    }

    project(vector) {
        return this.dot(vector) / vector.dot(vector);
    }

    dot(vector) {
        return ((this.x * vector.x) + (this.y * vector.y) + (this.z *
vector.z));
    }
}
...
var VectorAcce = new Vector(this.ax, this.ay, this.az);
var VectorAcceGravity = new Vector(this.gx, this.gy, this.gz);
this.project=VectorAcce.project(VectorAcceGravity);
```

Дізнавшись числове значення струсу, можна розпочати визначення координат пристрою. Геолокація на смартфоні працює добре, але не бездоганно. Зазвичай між двома отриманими координатами GPS є декілька відібраних записів сили струсу, а відстань між ними може бути до 10 м, а система потребує точніших значень.

За допомогою двох координат, кожна з яких містить значення широти й довготи (позначимо їх як $lat1$, $lng1$ і $lat2$, $lng2$), відміток часу цих координат ($t1$ і $t2$) та відмітки часу сили струсу (час спрацювання акселерометра та сенсора гравітації ta) можна дізнатися координати пошкодження дороги більш точно, використавши інтерполяцію.

Інтерполяція – в обчислювальній математиці – спосіб знаходження проміжних значень величини за наявним набором відомих значень. Схематичне зображення інтерполяції наведено на рисунку 3.

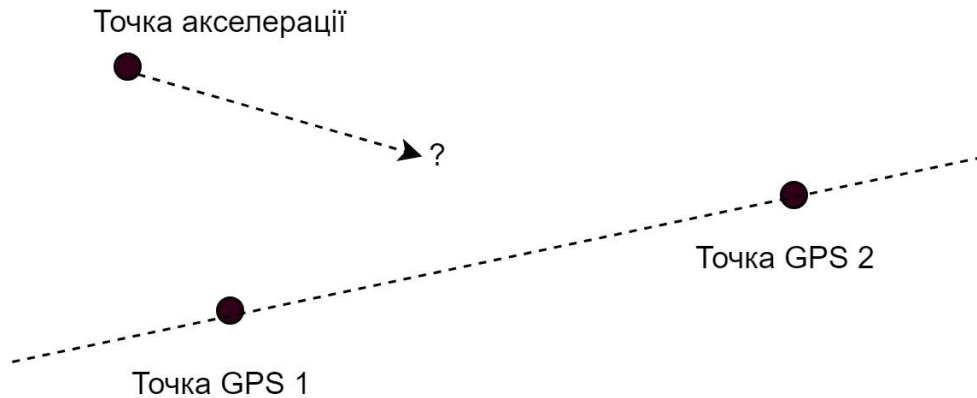


Рис. 3. Схема інтерполяції

Обчислимо середню швидкість (S) між двома точками GPS. Для цього потрібно дистанцію між точками поділити на час руху від однієї до другої, визначивши дистанцію, використовуючи формулу 2, де S – швидкість, D – дистанція, $lat1, lng1, lat2, lng2$ – значення координат точок.

$$D = \sqrt{(lat2 - lat1)^2 + (lng2 - lng1)^2}. \quad (2)$$

Знаючи дистанцію, за допомогою формули 3 визначимо швидкість, де S – швидкість, D – дистанція, а $t1, t2$ – відмітки часу в точках.

$$S = \frac{D}{t2 - t1}. \quad (3)$$

Знаючи швидкість між двома точками, мітки часу акселерації та першої геопозиції, можна дізнатися дистанцію до точки акселерації, використовуючи формулу 4, де Da – дистанція до точки спрацювання сенсорів, $t1$ – час у першій координаті GPS, ta – час спрацювання сенсорів.

$$Da = S \times (ta - t1). \quad (4)$$

Отже, маючи координати першої точки GPS і дистанцію до точки спрацювання сенсорів можна визначити точніші координати вказаної точки. Не вистачає лише напрямку (кута) від першої точки GPS до другої, який можна визначити за допомогою формули 5, де H – кут між точками, $lat1, lng1, lat2, lng2$ – значення координат двох точок GPS.

$$\begin{aligned} y &= \sin(lng2 - lng1) \times \cos(lat2); \\ x &= \cos(lat1) \times \sin(lat2) - \sin(lat1) \times \cos(lat2) \times \cos(lng2 - lng1); \\ H &= \frac{\arctan(y, x) \times 180}{\pi}. \end{aligned} \quad (5)$$

Тепер є можливість за допомогою першої точки геопозиції, дистанції та напрямку до точки спрацювання сенсорів визначити її геопозицію за формулою 6, де $Alat, Alng$ – широта та довгота точки спрацювання сенсорів, $lat1, lng1$ – координати першої точки GPS, H – напрямок (кут), Da – дистанція до точки спрацювання сенсорів.

$$\begin{aligned} Alat &= lat1 + Da \times \cos(H); \\ Alng &= lng1 + Da \times \sin(H). \end{aligned} \quad (6)$$

Останній крок визначення координат спрацювання сенсорів у математичних обрахунках виконано, що дає можливість перейти до програмного коду. У програмному коді використаємо сферичну інтерполяцію [12] та бібліотеку `google.maps.geometry.spherical`, яка працює з картами Google, тому у кодї, наприклад, для визначення дистанції між координатами або напрямку, будуть застосовуватися функції такої бібліотеки. Фрагмент коду, що відповідає формулі 2, наведено нижче.

```
var startLatLng = new google.maps.LatLng(tempFirst.latitude,
tempFirst.longitude);
var endLatLng = new google.maps.LatLng(tempSecond.latitude,
tempSecond.longitude);
var gpsSpeed =
google.maps.geometry.spherical.computeDistanceBetween(startLatLng, endLatLng
) / (tempSecond.timestamp - tempFirst.timestamp);
var heading = google.maps.geometry.spherical.computeHeading(startLatLng,
endLatLng);
var interpolatedDuration = tempAcce[i].timestamp - tempFirst.timestamp;
var interpolatedDistance = gpsSpeed * interpolatedDuration;
var interpolatedLatLng = google.maps.geometry.spherical.computeOffset(
startLatLng, interpolatedDistance, heading);
```

Формули (2–5) повторюються для кожного спрацювання акселерометра між кожними двома точками геопозиції пристрою, що дає можливість визначити координати усіх спрацювань акселерометра.

Визначення координат усіх точок спрацювання акселерометра дасть можливість більш точно розмістити на карті пошкодження доріг, після відправки їх на сервер.

Інтерфейс та робота з системою. Роботу з системою можна починати з вебдодатка. У ньому є можливість завантажити мобільний додаток, зареєструватися у системі, або, якщо вже є обліковий запис, потрапити до особистого кабінету з поїздками користувача чи переглядати зібрані усіма користувачами дані на загальнодоступній карті (рис. 3).

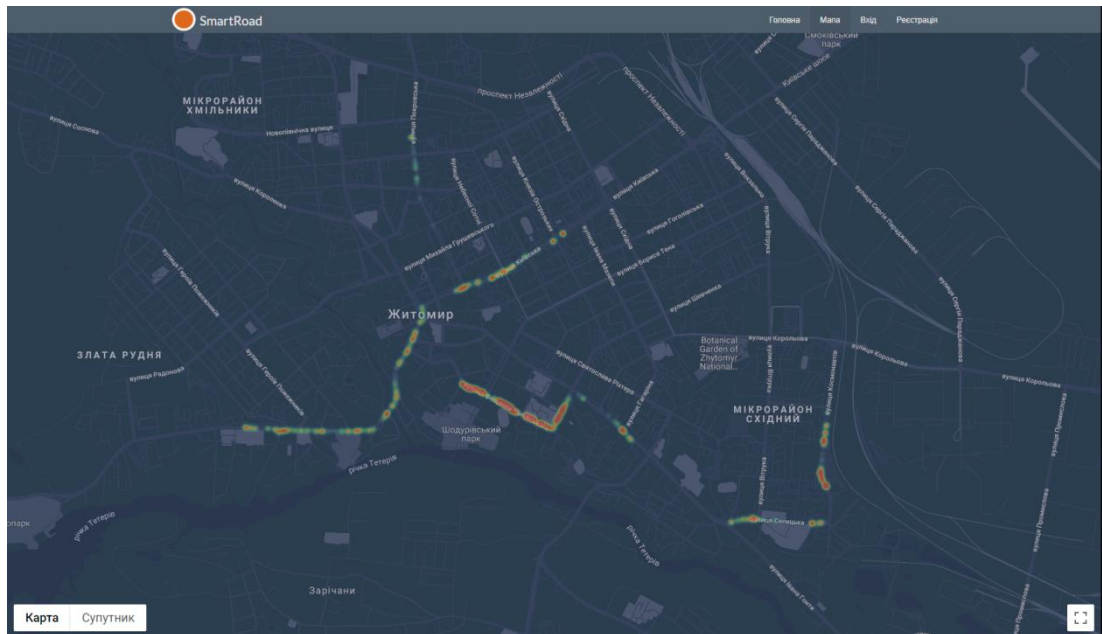
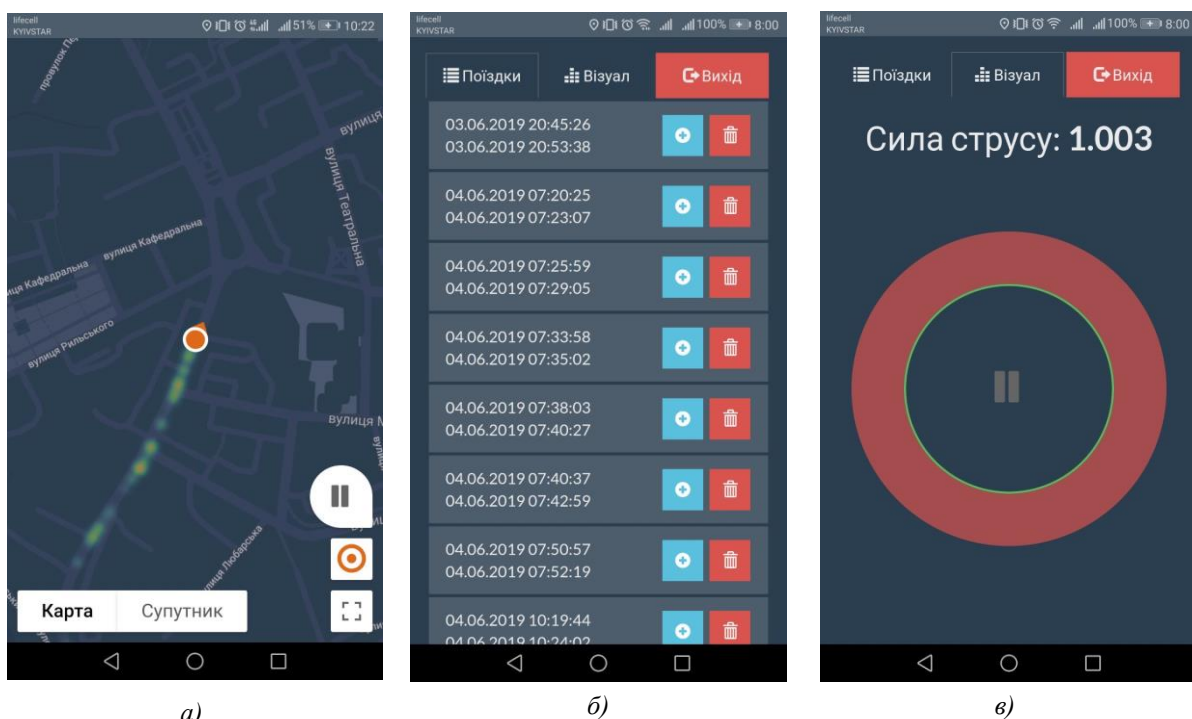


Рис. 3. Візуалізація пошкоджень автошляхів

Перейдемо до інтерфейсу мобільного додатка. Після входу користувач потрапляє на сторінку з мапою (рис. 4, а) та вкладками «Поїздки» зі списком поїздок (рис. 4, б) і «Візуал» (рис. 4, в) для оформлення зовнішнього вигляду додатка.



а)

б)

в)

Рис. 4. Інтерфейс користувача додатка

На мапі видно розміщення користувача, при цьому є можливість переключатися між мапою та супутниковим виглядом, вмикати повноекранний режим та керувати слідуванням за місцем розташування користувача. Також на карті доступна кнопка старту поїздки, яка вмикає алгоритм визначення пошкоджень доріг.

Також кнопка старту доступна на вкладці «Візуал», вона розширюється залежно від сили струсу, яка зазначається над кнопкою.

На сторінці зі списком поїздок користувач може відобразити їх на карті або видалити.

Висновки. Створено систему моніторингу стану автошляхів, що складається з веб- та мобільного додатків. Мобільний додаток отримує дані, вебдодаток зберігає та відображає дані. Такий комплекс є актуальним для використання на різних рівнях: у логістиці, управлінні, комунальних та державних організаціях і в повсякденному житті автомобілістів.

Під час проєктування і розробки проаналізовано аналоги та спроектовано функціональні можливості системи, обрано архітектуру для швидкої розробки системи і її придатності до подальшого вдосконалення. Для безпосередньої розробки використано сучасні набори мов програмування та технологій: PHP, JS, HTML, CSS, Android Java, MySQL, JavaScript Maps API. Реалізовано основний алгоритм системи, який використовує акселерацію та гравітацію пристрою й обчислює рівень пошкодження дороги в скалярному вимірі, а за допомогою геолокації визначає глобальні координати пошкодження дорожнього покриття.

Розроблювальна система вийшла гнучкою для модернізації, легкою в розгортанні та добре витримує навантаження.

Список використаної літератури:

1. Waze [Електронний ресурс]. – Режим доступу : <https://www.waze.com/ru/waze>.
2. Spothole. Pothole Tracking App [Електронний ресурс]. – Режим доступу : <http://www.fillinthepotholes.com/spot-hole-app/>.
3. SmartRoadSense [Електронний ресурс]. – Режим доступу : <http://www.smartroadsense.it>.
4. Documentation Yii2 [Електронний ресурс]. – Режим доступу : <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>.
5. Web Application Development with Yii 2 and PHP. – 2014. – 360 p.
6. Documentation PHP [Електронний ресурс]. – Режим доступу : <https://www.php.net/docs.php>.
7. Java SE8 for the Really Impatient: A Short Course on the Basics (Java Series). – 2017. – 836 p.
8. Android developer [Електронний ресурс]. – Режим доступу : <https://developer.android.com/studio/intro>.
9. About MySQL [Електронний ресурс]. – Режим доступу : <https://www.mysql.com/about/>.
10. Google Maps JS API [Електронний ресурс]. – Режим доступу : <https://developers.google.com/maps/documentation/javascript/tutorial>.
11. Introduction to vector projection [Електронний ресурс]. – Режим доступу : <https://www.khanacademy.org/math/linear-algebra/matrix-transformations/lin-trans-examples/v/introduction-to-projections>.
12. Utility functions for computing geodesic angles, distances and areas. The default radius is Earth's radius of 6378137 meters [Електронний ресурс]. – Режим доступу : <https://developers.google.com/maps/documentation/javascript/reference/geometry#spherical>.
13. Perttunen Mikko Distributed Road Surface Condition Monitoring Using Mobile Phones / Mikko Perttunen, Oleksiy Mazhelis, Fengyu Cong [Електронний ресурс]. – Режим доступу : https://link.springer.com/chapter/10.1007/978-3-642-23641-9_8.
14. The use of vehicle acceleration measurements to estimate road roughness / A.Gonzlez, E.J. O'brien, Y.Y. Li, K.Cashell // Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility. – 2008. – Vol. 46. – P. 483–499.
15. The pothole patrol: using a mobile sensor network for road surface monitoring / J.Eriksson, L.Girod, B.Hull and other // In MobiSys 2008 : Proceeding of the 6th International Conference on Mobile Systems, Applications and Services. – New York, ACM, 2008. – P. 29–39.
16. Mohan P. Nericell: rich monitoring of road and traffic conditions using mobile smartphones / P.Mohan, V.N. Padmanabhan, R. Ramjee // In Proc. ACM SenSys 2008. – New York, ACM, 2008. – P. 323–336.
17. Титаренко В.С. Оптимізація режимів експлуатації несучої системи транспортних засобів на основі використання системи збору інформації про зовнішній стан дорожнього покриття в ІТС / В.С. Титаренко, В.П. Шумляківський, І.А. Балагуц // Вісник ЖДТУ. – 2014. – № 4 (71).
18. Сервіс моніторингу стану доріг Navizor [Електронний ресурс]. – Режим доступу : <http://navizor.com>.

References:

1. Waze, [Online], available at: <https://www.waze.com/ru/waze>
2. Spothole. Pothole Tracking App, [Online], available at: <http://www.fillinthepotholes.com/spot-hole-app/>
3. SmartRoadSense, [Online], available at: <http://www.smartroadsense.it>
4. Documentation Yii2, [Online], available at: <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>
5. *Web Application Development with Yii 2 and PHP* (2014), 360 p.
6. Documentation PHP, [Online], available at: <https://www.php.net/docs.php>

7. *Java SE8 for the Really Impatient: A Short Course on the Basics (Java Series)* (2017), 836 p.
8. Android developer, [Online], available at: <https://developer.android.com/studio/intro>
9. About MySQL, [Online], available at: <https://www.mysql.com/about/>
10. Google Maps JS API, [Online], available at: <https://developers.google.com/maps/documentation/javascript/tutorial>
11. Introduction to vector projection, [Online], available at: <https://www.khanacademy.org/math/linear-algebra/matrix-transformations/lin-trans-examples/v/introduction-to-projections>
12. Utility functions for computing geodesic angles, distances and areas. The default radius is Earth's radius of 6378137 meters, [Online], available at: <https://developers.google.com/maps/documentation/javascript/reference/geometry#spherical>
13. Perttunen, Mikko, Mazhelis, Oleksiy and Cong, Fengyu Distributed, «Road Surface Condition Monitoring Using Mobile Phones», [Online], available at: https://link.springer.com/chapter/10.1007/978-3-642-23641-9_8
14. Gonzalez, A., O'Brien, E.J., Li, Y.Y. and Cashell, K. (2008), «The use of vehicle acceleration measurements to estimate road roughness», *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, Vol. 46, pp. 483–499.
15. Eriksson, J., Girod, L., Hull, B. and other (2008), «The pothole patrol: using a mobile sensor network for road surface monitoring», *In MobiSys 2008*, proceeding of the 6th International Conference on Mobile Systems, Applications, and Services, ACM, New York, pp. 29–39.
16. Mohan, P., Padmanabhan, V.N. and Ramjee, R. (2008), «Nericell: rich monitoring of road and traffic conditions using mobile smartphones», *In Proc. ACM SenSys 2008*, ACM, New York, pp. 323–336.
17. Titarenko, V.Y., Shumliakivskiy, V.P. and Balaguts, I.A. (2014), «Optimization of modes of exploitation of the carrier system of vehicles based on the use of the system of gathering information on the external state of the road surface in ITS», *Visnyk ZSTU*, No. 4 (71).
18. Service for road condition monitoring Navizor, [Online], available at: <http://navizor.com>

Ісаєв Андрій Миколайович – магістрант за освітньою програмою «Інформаційні системи та технології» Державного університету «Житомирська політехніка».

Наукові інтереси:

- інтелектуальний аналіз даних;
- розробка мобільних та вебдодатків для інформаційних систем.

E-mail: isaev.andrii@gmail.com.

Кузьменко Олександр Вікторович – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

Наукові інтереси:

- інтелектуальний аналіз даних;
- розробка вебдодатків та геоінформаційних систем.

E-mail: kuzmenko@ukr.net.

Сугоняк Інна Іванівна – кандидат технічних наук, доцент, завідувач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

Наукові інтереси:

- системний аналіз та теорія оптимальних рішень;
- проектування сховищ даних, інтелектуальний аналіз даних.

E-mail: org_sii@gmail.com.

Стаття надійшла до редакції 17.03.2020.