

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу

Кафедра інженерії програмного забезпечення

Пояснювальна записка до дипломного проекту

магістра
(освітній ступінь)

на тему «Модель екосистеми Інтернету речей системи
«Шкільний автобус»

ХАІ.603.695пз1.121.195–9/19.20В

Виконала: студентка 6 курсу групи №6-95Пз1
Спеціальність 121 Інженерія програмного
забезпечення

(код та найменування)

Освітня програма Хмарні обчислення
та Інтернет речей

(найменування)

Палаус О.М.

(прізвище й ініціали студента)

Керівник: Вдовіченко О. В.

(прізвище й ініціали)

Рецензент: Іванісенко І.М.

(прізвище й ініціали)

Харків – 2020

Міністерство світи і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет програмної інженерії та бізнесу
 (повне найменування)
 Кафедра інженерії програмного забезпечення
 (повне найменування)
 Рівень вищої освіти другий (магістерський)
 Спеціальність 121 – Інженерія програмного забезпечення
 (код та найменування)
 Освітня програма Хмарні обчислення та Інтернет речей
 (найменування)

ЗАТВЕРДЖУЮ
Завідувач кафедри

 (підпис) _____ (ініціали та прізвище)
 “ _____ ” _____ 2020 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТКИ

Палаус Ольги Миколаївни
 (прізвище, ім'я, ПЗ батькові)

1. Тема дипломного проекту: «Модель екосистеми Інтернету речей системи «Шкільний автобус»

керівник дипломного проекту Вдовітченко Олександр Валерійович, к. т. н.
 (прізвище, ім'я, ПЗ батькові, науковий ступінь, вчене звання)

затвержені наказом Університету № _____ від “ _____ ” _____ 2020 року

2. Термін подання студентом проекту 30.11.2020

3. Вихідні дані до проекту формулювання проблемних питань оцінки рівня впровадження технології Інтернету речей та хмарних обчислень як інструментальних засобів системи відстеження транспорту

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) аналіз існуючих методів, моделей та інструментальних засобів екосистеми Інтернету речей «Шкільний автобус», розробка моделі та методу екосистеми Інтернету речей «Шкільний автобус», алгоритми ПЗ, результати проектування ПЗ та перевірки методу.

5. Перелік графічного матеріалу:

РПЗ – стор. 67, рисунків – 35 шт., формул – 11 шт., слайдів презентації – 17 шт.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Вдовітченко О.В., доц. каф. 603		
2	Вдовітченко О.В., доц. каф. 603		
3	Вдовітченко О.В., доц. каф. 603		

7. Дата видачі завдання _____ 2020 р.

8. Нормоконтроль _____ « ____ » _____ 2020 р.
(підпис) (ініціали та прізвище)**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і затвердження теми диплому	01.09.2020	
2	Аналіз методів, моделей та інструментальних засобів екосистеми Інтернету речей «Шкільний автобус»	02.09.2020–25.09. 2020	
3	Розробка моделі та методів екосистеми Інтернету речей «Шкільний автобус»	28.09.2020–15.10.2020	
4	Розробка алгоритмів ПЗ	16.10.2020–26.10.2020	
5	Проектування прототипу ПЗ	27.10.2020–09.11.2020	
6	Перевірка ефективності розробленого метода	10.11.2020–19.11.2020	
7	Оформлення пояснювальної записки до НДР магістра	20.11.2020–25.11.2020	
8	Передзахист дипломної роботи	26.11.2020–30.11.2020	
9	Захист дипломної роботи	01.12.2020–16.12.2020	

Студентка

(підпис)

Палаус О. М.

(прізвище та ініціали)

Керівник проекту

(підпис)

Вдовітченко О. В.

(прізвище та ініціали)

РЕФЕРАТ

Дипломний проект: 67 с., 35 рисунків, 29 посилань.

Саме з Інтернетом речей більшість науковців та винахідників пов'язують четверту промислову революцію (Індустрія 4.0). Переважна більшість експертів погоджуються, що Інтернет речей до 2025р. набуде широкого розповсюдження і принесе користь людству.

Інтернет речей – це сукупність фізичних об'єктів, комплексно з'єднаних між собою за допомогою сучасних Інтернет технологій. Він дає можливість появи нових, більш ефективних методів виробництва та ведення бізнесу.

Промисловий Інтернет речей – це один з найбільш великих сегментів Інтернету речей з точки зору кількості підключених пристроїв і ступеня корисності цих сервісів для виробництва і автоматизації підприємств.

Проте є певна категорія напрямків, як наприклад, освіта, освітній процес, де технологія Інтернету речей ще недостатньо розкрита та впроваджена.

Метою роботи є підвищення ефективності системи "Шкільний автобус" для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, за рахунок розробки моделі системи відстеження шкільного транспорту на основі технологій Інтернету речей та хмарних обчислень

Об'єкт дослідження – системи відстеження транспорту.

Предмет дослідження – технології Інтернету речей та хмарних обчислень як інструментальні засоби системи відстеження транспорту.

Методи дослідження. Досягнення мети роботи базується на використанні теорії системного аналізу, що дозволяє комплексно розглянути всі аспекти, пов'язані з розвитком технологій застосування Інтернету речей та хмарних обчислень, технологій проектування, логістики, геодезичні методи визначення координат об'єктів, методи відстаней між об'єктами за їх географічними координатами.

Практичне значення отриманих результатів. Результати роботи можуть бути використані надалі для впровадження технології Інтернету речей в систему відстеження транспорту для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, з метою підвищення ефективності безпеки життєдіяльності.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ РЕЧЕЙ, ХМАРНІ ОБЧИСЛЕННЯ, СИСТЕМИ СТЕЖЕННЯ ТРАНСПОРТУ, ШКІЛЬНИЙ АВТОБУС

ABSTRACT

Magister of diploma project: 67 pp., 35 figures, 29 references.

It is the Internet of Things that most scientists and inventors associate the fourth industrial revolution with (Industry 4.0). The vast majority of experts agree that the Internet of Things will have become widespread and extremely beneficial to the humanity by 2025.

The Internet of Things is a collection of physical objects that are connected with each other with the help of some modern Internet technologies. It enables new, more efficient methods of production and business to emerge.

The Industrial Internet of Things is one of the largest segments of the Internet of Things in terms of the number of devices and their level of usefulness for manufacturing and business automation.

However, there is a certain category of spheres such as education and studying process, where the Internet of Things technology is not developed and properly implemented yet.

The aim of the work is to increase the efficiency of the "School Bus" system for the transportation of secondary school students who live in rural areas, by developing a model of school transport tracking system based on the Internet of Things technologies and cloud computing.

The object of research – transport tracking systems.

Subject of research – Internet of things and cloud computing technologies as tools of transport tracking system.

Methods of research. A number of certain methods are used to achieve the goal. They are mostly based on the theory of systems analysis, which allows to comprehensively consider all the aspects related to the development of technologies for the effective use of the Internet of Things and cloud computing, design technologies, logistics, geodetic methods for the determination of object coordinates and methods of distances between objects according to their geographical coordinates.

The practical significance of the results obtained. The results of the work can be further used to integrate the Internet of Things technology into the tracking system aimed at the efficient transportation of secondary school students from rural areas in order to improve safety measures.

INTERNET OF THINGS, CLOUD COMPUTING, TRANSPORTATION TRACKING SYSTEMS, SCHOOL BUS

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

IoT, Internet of things – технологія Інтернет речей.

IIoT, Industrial IoT – промисловий Інтернет речей.

GPS, Global Positioning System – система глобального позиціонування.

PaaS, Platform as a Service – «платформа як послуга».

TMS, Transport Management System – системи керування транспортною логістикою.

ПЗ – Програмне забезпечення.

Система моніторингу стану машини – система дистанційного діагностування технічного стану вузлів і агрегатів з постійним обміном даних між машиною і оператором сервісної служби.

Хмарні обчислення – модель забезпечення повсюдного та зручного доступу на вимогу через мережу до спільного пулу обчислювальних ресурсів.

ЗМІСТ

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ	6
ВСТУП.....	8
1 АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ЕКОСИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ «ШКІЛЬНИЙ АВТОБУС»	10
1.1 Потреби суспільства щодо відстеження транспорту різного призначення ..	10
1.2 Основні положення державної програми «Шкільний автобус»	11
1.3 Огляд ринку програмного забезпечення	15
1.3.1 Класичне програмне забезпечення відстеження транспорту	15
1.3.2 Інтернет речей процесу відстеження транспорту	18
1.4 Висновки до розділу 1	24
2 МОДЕЛІ ТА МЕТОДИ ЕКОСИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ «ШКІЛЬНИЙ АВТОБУС».....	26
2.1 Опис моделі екосистеми Інтернету речей «Шкільний автобус»	26
2.2 Методи відстеження місця розташування шкільного автобусу	33
2.3 Висновки до розділу 2.....	41
3 РОЗРОБКА ПРОТОТИПУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЕКОСИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ СИСТЕМИ «ШКІЛЬНИЙ АВТОБУС»	43
3.1 Моделювання програмного забезпечення	43
3.2 Алгоритми роботи з системою «Шкільний автобус».....	49
3.3 Розробка прототипу програмного додатку Інтернету речей для сповіщення користувачів про наближення автобусу	56
3.4 Висновки до розділу 3.....	62
ВИСНОВКИ	64
ПЕРЕЛІК ПОСИЛАНЬ	66
ДОДАТОК А. Машинний лістинг реалізованого програмного забезпечення...	68
ДОДАТОК Б. Зміст json-файлу, який містить координати автобусу.....	72
ДОДАТОК В. Зміст json-файлу, який містить координати користувача.....	73

ВСТУП

Актуальність досліджень. На даний момент все більше вітчизняних підприємств та компаній використовують у своїй діяльності системи моніторингу транспорту. Це автоматизовані системи управління логістикою, які дозволяють систематизувати транспорт і процес перевезення, в більшості випадках за допомогою супутникових сигналів [1].

Основний принцип застосування таких систем – постійне відстеження транспортних засобів в дистанційному режимі, регуляція маршруту і надання звіту про пересування після прибуття в кінцевий пункт.

Для того, щоб здійснити переміщення транспорту по цифровій карті, на нього встановлюється спеціальний навігатор, який відстежує, приймає і передає сигнал. Дані видаються або на карту, або в текстовому поданні.

Супутникова (цифрова) карта, що застосовується для побудови маршруту, здатна вирішувати такі завдання як:

- моніторинг та відстеження місцезнаходження транспортного засобу в режимі реального часу. При цьому можуть передаватися також такі дані, як швидкість пересування, загальна витрата пального на певну кількість кілометрів, а також загальний стан транспортного засобу. Це дозволяє ще більш точно диспетчеру оптимізувати маршрут і знизити витрати на перевезення;

- контроль за графіком прибуття в кінцевий пункт. Маршрут будується за певних точок, проходження яких фіксується. Дані про швидкість дозволяють віддавати водієві вказівки за планом пересування, при якому він зможе досягти кінцевої мети в точний час;

- статистика і аналітика. На основі отриманих даних логісти складають карти маршрутів для майбутніх перевезень, економістом виділяється бюджет на паливо, а також поліпшуються умови перевезень;

- безпеку. Якщо транспортний засіб потрапив в дорожньо–транспортну пригоду, то за допомогою карти можна передати сигнал про допомогу. Також супутникові карти ефективні, якщо стався угон. Транспорт можна відразу ж відстежити.

З розвитком новітніх технологій, наприклад Інтернету речей та хмарних обчислень, системи моніторингу (відстеження) транспорту доцільно модернізувати здатністю самостійно аналізувати, приймати рішення та повідомляти всіх учасників процесу про зміни в діяльності транспорту.

Дослідженням шляхів та напрямків впровадження Інтернету речей та хмарних обчислень присвячено значну кількість робіт, серед яких слід особливо відзначити роботи вчених: Д. Андерсона, Д. Баркера, Д. Еванса, К. Ештона, Н. Карра, К. Пістера, Д. Хелбінга, Е.Шмідта, В. Бикова, Н. Богданової, С. Ваняшина, А. Гребешкова, В. Довгаля, А. Рослякова, А. Сівкова, М. Самсонова, З. Сейдаметової, С. Сейтвелієвої, А. Цоя, І. Шадеркіної, В. Шадеркіної та ін..

Практично не вивченим є питання впровадження сучасних систем відстеження в освітніх закладах. Таким чином, в даний час залишається

актуальним завдання розробка екосистеми Інтернету речей «Шкільний автобус».

Метою роботи є підвищення ефективності системи "Шкільний автобус" для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, за рахунок розробки моделі системи відстеження шкільного транспорту на основі технологій Інтернету речей та хмарних обчислень.

Для досягнення мети потрібно вирішити наступні завдання:

- проаналізувати державну програму «Шкільний автобус» та існуючі проектні рішення для відстеження автобусу;
- проаналізувати існуючі інструментальні засоби технології Інтернету речей;
- розробити модель екосистеми IoT системи «Шкільний автобус»;
- розробити метод IoT для визначення моменту сповіщення користувача про наближення автобусу;
- виконати моделювання програмного забезпечення;
- розробити програмний додаток Інтернету речей для сповіщення користувачів про наближення автобусу.

Об'єкт дослідження – системи відстеження транспорту.

Предмет дослідження – технології Інтернету речей та хмарних обчислень як інструментальні засоби системи відстеження транспорту.

Методи дослідження. Досягнення мети роботи базується на використанні теорії системного аналізу, що дозволяє комплексно розглянути всі аспекти, пов'язані з розвитком технологій застосування Інтернету речей та хмарних обчислень, технологій проектування ПЗ, логістики, геодезичні методи визначення координат об'єктів, методи відстаней між об'єктами за їх географічними координатами.

Наукова новизна отриманих результатів. Вперше розроблено модель екосистеми Інтернету речей для відстеження шкільного транспорту, яка на відмінну від існуючих базується на комплексному використанні Інтернету речей, хмарних обчислень та екосистеми програмного забезпечення. Розроблений метод Iot базується на порівнянні відстаней між об'єктами системи, географічні координати яких були передані їх відповідними пристроями.

Практичне значення отриманих результатів. Результати роботи можуть бути використані надалі для впровадження технології Інтернету речей в систему відстеження транспорту для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, з метою підвищення ефективності безпеки життєдіяльності.

1 АНАЛІЗ МЕТОДІВ, МОДЕЛЕЙ ТА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ЕКОСИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ «ШКІЛЬНИЙ АВТОБУС»

1.1 Потреби суспільства щодо відстеження транспорту різного призначення

Найпопулярнішим сегментом суспільства, що активно використовує системи відстеження транспорту залишаються комерційні структури, одним з головних підрозділів якої є логістика. До них можна віднести компанії–перевізники, логістичні служби виробників і дистриб'юторів товарів, транспортні служби роздрібних мереж, служби кур'єрської доставки та інтернет–магазинів з власними кур'єрами. Адже, від того, наскільки чітко, пунктуально і злагоджено працює ця ланка, залежить уся робота компанії, а також ступінь задоволеності споживача.

Використання спеціальних систем для керування транспортом дозволить компанії грамотно вирішувати поставлені виробничі завдання, а також оптимізувати неухильно зростаючі витрати. Зазвичай такі системи використовують супутниковий моніторинг, що дозволяє підвищити ефективність роботи автотранспорту через максимальний облік стану техніки.

Система GPS–моніторинга складається з двох важливих компонентів – самого GPS–трекера та його програмного забезпечення [2]. GPS –трекер монтується в транспортний засіб, після чого визначає координати за допомогою супутників та передає дані на спеціальний захищений сервер. Програмне забезпечення потрібно для обробки отриманих даних та виведення готових координат на карту, де буде показана локація транспорту, а також маршрут, який він здійснив за певний період часу. Крім визначення маршруту і розташування, можна отримувати інформацію про зупинки, пробіг, паливному витраті (в тому числі про заправках і зливах палива), а також про те, що транспорт відхилився від заданого маршруту.

GPS маячок дозволяє визначити місце розташування певного об'єкта з точністю. Може використовуватися не тільки для автомобілів, але і для маленьких дітей, цінного вантажу і будь–яких об'єктів, які можуть вкрасти. Що стосується транспортних засобів, то такі пристрої є доповненням до основної охоронній системі.

Отже, впровадження системи відстеження транспорту дозволить компанії автоматизувати наступні функції диспетчера:

- припинення нецільового використання транспорту. При наявності GPS–контролю водієві набагато складніше міняти маршрут та використовувати службовий транспорт не за призначенням;

- контроль над ситуацією та прогнозування затримок. Відстежуючи пересування автомобілів в режимі on–line, диспетчер володіє ситуацією та може вчасно реагувати на відхилення від плану, події, попереджати клієнтів про затримку, прогнозувати час доставки;

– скорочення витрат на паливно–мастильні матеріали (ПММ). Збір статистики про пересування з подальшою передачею в облікову систему та аналізом, а також контроль над діями водіїв завжди призводять до скорочення витрат на ПММ на 10–15%, що з лишком покриває вартість послуги GPS–моніторингу;

– інформування про події. Для того, щоб позбавити оператора від необхідності постійно відслідковувати стан транспорту on–line, в модулі моніторингу може бути реалізована можливість генерації повідомлень оператору за подією. Інформація про подію формується на основі встановлених датчиків;

– можливі події для відстеження: перевищення швидкості, в'їзд/виїзд із зони, відхилення від рейсу, відвідування точки рейсу, зміна стану вантажу, торгової точки або рейсу, відкриття/закриття дверей, увімкнення/вимикання двигуна, температура в рефрижераторі і т.д.

Отже, системи відстеження знадобляться підприємствам, які пов'язані з транспортною логістикою, наприклад, підприємствам сільського господарства, морських портів, залізничниці. Всюди де є важливим питання знизити витрати на доставку та переміщення продукції, та збільшити прибуток від виробництва.

На жаль, на даний момент спостерігається низька тенденція впровадження систем відстеження транспорту в соціальних сферах, наприклад для відстеження громадського транспорту, шкільних автобусів та іншого соціального транспорту. Одним з факторів даної тенденції це завелика вартість систем відстеження, великий термін окупності або навіть його відсутність. Адже, впровадження систем відстеження в громадському транспорті в більшій мірі забезпечує зручність користування транспортом для мешканців міст чи селищ, безпеку перевезення школярів та спокій батьків у віддалених районах області.

Аналіз предметної області показав доцільність та корисність використання системи відстеження транспорту в комерційних цілях для збільшення прибутку та зменшення витрат підприємства, проте при пасажирських перевезеннях використання таких систем більше зумовлено питаннями зручності та безпеки, а ніж економічної вигоди для підприємства.

1.2 Основні положення державної програми «Шкільний автобус»

Теперішня система освіти перебуває в стадії реформ. Створення Нової української школи, процеси децентралізації країни та освіти, впливають на якість шкільної освіти. Проте за вимогою суспільства освітні послуги мають залишатись і якісними і доступними для кожного.

Під час створення освітніх округів, мереж опорних шкіл та їх філій підвезення учнів та педагогічних працівників сільської місцевості, що проживають за межею пішохідної доступності до навчальних закладів, до місця навчання (роботи) і додому повинно бути належно організовано і є дуже важливою складовою надання освітніх послуг.

Закон України «Про Освіту», прийнятий у вересні 2017 року регулює багато аспектів надання освітніх послуг, зокрема і про підвезення [6].

Частина четверта статті 13 Закону говорить про те, що особи, які здобувають повну загальну середню освіту, проживають у сільській місцевості і потребують підвезення до закладу освіти і у зворотному напрямку, забезпечуються таким підвезенням за кошти місцевих бюджетів, у тому числі із забезпеченням доступності відповідного транспорту для осіб з порушенням зору, слуху, опорно-рухового апарату та інших маломобільних груп населення.

Частина п'ята статті 13 Закону встановлює, що органи місцевого самоврядування мають право прийняти з урахуванням потреб і пропозицій територіальних громад спільне рішення про організацію здобуття освіти (початкової, базової та профільної середньої) та відповідно забезпечення підвезення здобувачів освіти до закладу освіти чи його філії і у зворотному напрямку.

Відповідно до частини першої статті 56 Закону, особи, які здобувають повну загальну середню освіту в закладах освіти не за місцем проживання, на період навчання забезпечуються гуртожитками або підвезенням. А відповідно до частини другої цієї ж статті, органи місцевого самоврядування забезпечують пільговий проїзд учнів, вихованців, студентів та педагогічних працівників до місця навчання і додому у визначених ними порядку та розмірах за рахунок видатків відповідних місцевих бюджетів.

Частина друга статті 66 Закону говорить про те, що районні, міські ради та ради об'єднаних територіальних громад забезпечують доступність дошкільної, початкової та базової середньої освіти, позашкільної освіти, а також забезпечують та фінансують підвезення учнів і педагогічних працівників до закладів початкової та базової середньої освіти і у зворотному напрямку (у разі потреби – транспортними засобами, пристосованими для перевезення осіб, які пересуваються на кріслах колісних).

Таким чином відповідний орган місцевого самоврядування повинен визначити спосіб, у яких здійснюється підвезення учнів і педагогічних працівників та забезпечити підвезення. Варто також, за необхідності, врахувати й передбачити фінансування для перевезення дітей з інвалідністю.

Залежно від існуючих інфраструктурних умов, матеріально-технічного забезпечення та фінансової спроможності для забезпечення підвезення учнів та педагогічних працівників можуть використовуватися:

- шкільні автобуси (власні або на умовах спонсорської допомоги);
- орендовані транспортні засоби у суб'єктів підприємницької діяльності,
- орендовані транспортні засоби у рад відповідних адміністративно-територіальних одиниць;
- перевезення учнів на громадському транспорті місцевого сполучення (що, практично неможливо за відсутності належних місцевих транспортних сполучень, доріг та відповідних маршрутів).

Шкільний автобус є спеціальним транспортним засобом для регулярних перевезень, що здійснюються за встановленим маршрутом і розкладом, з посадкою і висадкою дітей і педагогічних працівників на передбачених

маршрутом зупинках до місць навчання і додому, а також для організації екскурсійних поїздок, здійснення нерегулярних перевезень учнів та педагогічних працівників до місць проведення позакласних і позашкільних, культурно–масових заходів і спортивних змагань, підвезення до пунктів тестування зовнішнього незалежного оцінювання, можливе забезпечення участі учасників навчально–виховного процесу в заходах (нарадах, семінарах, конференціях) районного чи обласного рівнів [7].

На комерційній основі використання шкільних автобусів забороняється.

Організація перевезень дітей шкільними автобусами здійснюється у відповідності з діючими нормативно–правовими актами України [8] із забезпечення безпеки дорожнього руху, перевезень пасажирів автобусами, зокрема Закону України «Про дорожній рух», Закону України «Про автомобільний транспорт» та Правил надання послуг пасажирського автомобільного транспорту. При цьому балансоутримувач шкільного автобуса, що організовує перевезення учнів і педагогів, а саме керівник загальноосвітнього навчального закладу або керівник органу управління освітою (засновника закладу освіти) має пов'язані із цим додаткові обов'язки. Серед них:

1) визначає належне місце стоянки автобуса, забезпечує його технічне обслуговування та створює необхідні умови для проходження передрейсового технічного огляду;

2) забезпечує проходження водієм автобуса передрейсового та післярейсового медичного огляду;

3) Забезпечує щорічне навчання водія шкільного автобуса з питань безпеки дорожнього руху в обсязі технічного мінімуму.

4) Забезпечує наявність відповідної необхідної документації, в тому числі:

– паспорт маршруту шкільного автобуса;

– графік руху шкільного автобуса;

– накази про призначення відповідального за організацію перевезень, вихователя і супроводжуючих;

– накази про затвердження списків учнів та педагогів, які перевозяться; – наказ про затвердження інструкцій з організації безпечного перевезення учнів та педагогів;

– інструкції для водія про особливості роботи у весняно–літній і осінньо–зимовий періоди; про заходи безпеки під час перевезення учнів і педагогів; по наданню першої медичної допомоги потерпілим у дорожньо–транспортній пригоді;

– інструкцію для водія, вихователя і супроводжуючих по діях в разі дорожньо–транспортної пригоди та ін. можливих загроз;

– журнал обліку інструктажів для водія, вихователя та супроводжуючих;

– журнал обліку інструктажів для учнів;

– журнал передрейсового та післярейсового медичного огляду водія;

– журнал обліку порушень водієм правил дорожнього руху та участі в дорожньо–транспортних пригодах;

- журнал обліку технічного стану шкільного автобуса при випуску на лінію та повернення;
- договір на технічне обслуговування автобуса та ін.

Розглянемо, які варіанти перевезень може обрати школа або її засновник згідно з діючими нормативно–правовими актами України [8]:

а) Використання власного автобуса.

Найпоширенішим варіантом для підвезення є використання власного шкільного автобуса, який перебуває на балансі освітнього закладу або засновника закладу освіти.

Використання власного шкільного автобуса дає ряд переваг. Шкільний автобус є спеціально обладнаний для учнів, постійно їздить по встановленому маршруту саме для потреб підвезення дітей та педагогічних працівників, завжди доступний і окрім регулярних щоденних перевезень до школи та назад може використовуватися також і для підвезення дітей з нагоди позашкільних заходів, урочистих подій, спортивних змагань, для участі в учнівських олімпіадах, здачі ЗНО тощо.

Однак варто зауважити, що утримання шкільного автобуса передбачає цілу низку витрат, які має взяти на себе власник. Серед них як планові витрати, оскільки експлуатація транспортних засобів потребує як поточних і періодичних витрат на обслуговування, передбачає страхових внесків (за транспортний засіб, за водія), витрат на паливо (що може змінюватися протягом року), плановий технічний огляд, мастильні матеріали, заробітна плата водія (в т.ч. і в канікулярний період чи у дні коли рейсів здійснюється менше), так і позапланові витрати (у разі появи несправностей автобуса чи ДТП), підміна водія у разі його хвороби тощо.

б) Орендований транспорт.

Альтернативою шкільним автобусам для підвезення учнів до школи та назад додому є використання орендованих автобусів і у багатьох областях зустрічаються успішні практики їх використання. Для використання орендованого транспорту керівник навчального закладу (директор), як замовник транспортних послуг та перевізник (власник транспортного засобу) визначають маршрут перевезення, складають схему маршруту та розклад руху, укладають письмовий договір про перевезення.

Для здійснення пасажирських перевезень автомобільний перевізник повинен мати ліцензію на здійснення господарської діяльності з надання послуг з перевезення пасажирів і вантажів відповідно до видів робіт, визначених Законом України "Про автомобільний транспорт" , мати ліцензійні картки на кожний автобус, інші документи, передбачені законодавством України. Групи дітей повинні перевозити досвідчені водії транспортних засобів, які мають стаж керування автобусом не менше ніж п'ять років.

Під час організації регулярних автобусних перевезень дітей в сільській місцевості саме власники автобусів зобов'язані проводити комісійне обстеження стану автомобільних доріг, пунктів посадки та висадки дітей із залученням органів Державної автомобільної інспекції та дорожніх організацій; встановлювати на маршрутах спеціальні знаки зупинки із зазначенням часу

проходження автобусів, які здійснюють перевезення дітей, проводити плановий технічний огляд та ремонт автобуса, забезпечувати страхування транспорту та водія. Оскільки водій автобуса надається перевізником, перевізник забезпечує проведення медичних оглядів водіїв, які здійснюють перевезення дітей перед виїздом на рейс, зобов'язаний також проводити передрейсовий контроль технічного та санітарного стану автобуса безпосередньо перед виїздом, проводити інструктаж водія з безпеки дорожнього руху, забезпечити водія необхідними документами та ін. Максимальна кількість дітей та супроводжуючих осіб при перевезенні автобусом не повинна перевищувати кількості місць для сидіння в ньому, передбачену технічною характеристикою транспортного засобу та визначену в реєстраційних документах на цей транспортний засіб.

Отже, аналіз проблемної галузі показав, що незалежно від виду перевезень (власний чи орендований автобус) адміністрація школи або її засновник зацікавлені в безпеці та зручності перевезені осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, спокою батьків за життя дітей під час перевезень, низьку вартість послуг перевезень.

1.3 Огляд ринку програмного забезпечення

1.3.1 Класичне програмне забезпечення відстеження транспорту

Сучасний ринок комп'ютерних програм дає змогу знайти програму майже для будь-яких завдань. Інтернет значно спростив процес пошуку та покупки програм, в ньому можливо знайти та купити будь-що. Для пошуку проектних рішень для реалізації завдань аналогічного призначення був проаналізований саме ринок Інтернет програм.

Проаналізуємо програмний ринок на наявність програм для систем відстеження шкільного транспорту.

Системи відстеження транспорту дуже популярні в логістичних фірмах, компаніях доставки пошти та вантажних перевезень. Всі вони мають дуже високу вартість, що обмежує доступ до них малих фірм та бюджетних організацій, яким було б корисно мати системи стеження за власним спеціалізованим транспортом

Розглянемо існуючу програмні рішення систем відстеження для вантажних перевезень. До них відносять системи керування транспортною логістикою [12].

Система Logist.ua включає функції планування, GPS-моніторингу та керування транспортом підприємства. Вона має модульну структуру – залежно від того, які процеси треба оптимізувати і в яких масштабах, для впровадження вибирається один або кілька модулів. Модуль «Планування» дозволяє проаналізувати усі ввідні дані для розрахунку маршрутів доставки і на цій основі спланувати оптимальні рейси, що враховують усі особливості автомобілів і доріг, графіки роботи клієнтів, пріоритети та інше. Модуль

«Моніторинг» за допомогою системи GPS–навігації дозволяє відстежувати рух транспорту в реальному часі, зіставляти по карті фактичний маршрут кожної машини із запланованим. А також фіксувати всі події, що відбуваються з автомобілем: відхилення від маршруту, затримки в часі, відвідування точок призначення і запізнення. За необхідності диспетчер може вчасно скоригувати дії водіїв. Мобільні додатки Logist.ua дають можливість здійснювати планування і моніторинг рейсів за допомогою мобільних пристроїв.

На рисунку 1.1 показано вікно для відстеження за транспортом.

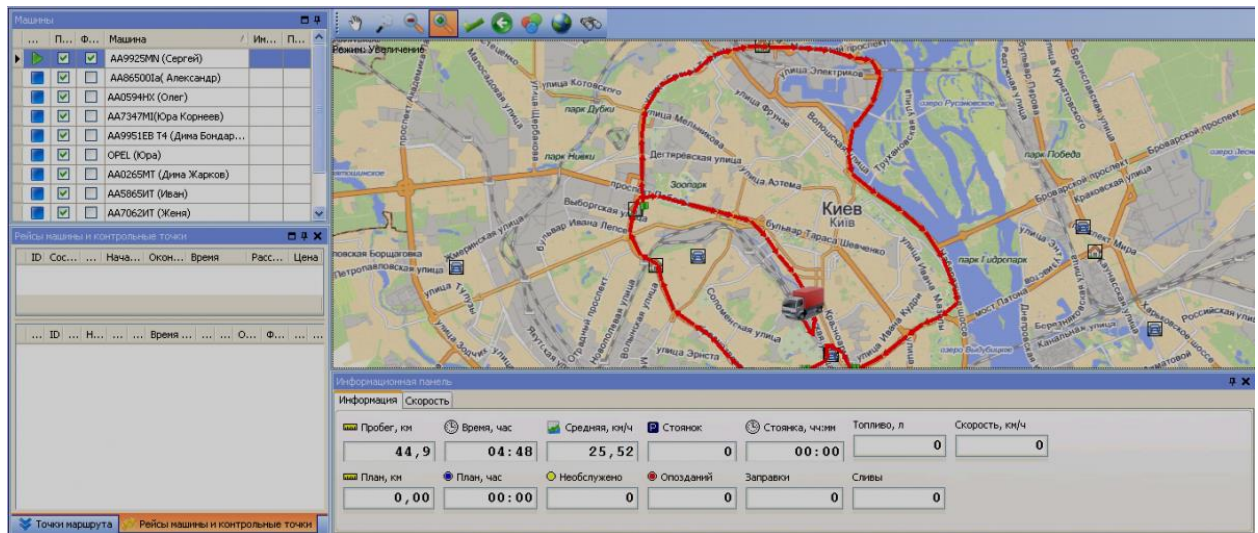


Рисунок 1.1 – Вікно стеження за машиною в модулі моніторингу

Отже, до найпопулярніших функцій модуля GPS моніторингу Logist.UA можна віднести наступні:

- контроль місця розташування автомобілів з нанесенням на карту;
- моніторинг транспорту підприємства та пересування автомобілів в режимі on–line;
- формування історії пересувань по кожному автомобілю (архів);
- статистика по пробігу, швидкості руху, розрахунок середньої витрати палива;
- передача даних в облікову систему підприємства для подальшої аналітики.

Стеження здійснюється програмою Logist.UA на основі GPS–технологій, із застосуванням спеціалізованого трекінгового GPS обладнання або без нього (з використанням LBS). Logist.UA підтримує більшість моделей GPS–трекерів та обладнання: Bitrek, Новітек, Teltonika і аналоги [13].

З мінусів потенційні клієнти називають вартість (не оголошується компанією). На українському ринку система керування транспортною логістикою називається АСТОР: TMS. Існує мобільна версія сервісу. Відомі користувачі: PepsiCo Russia, Metro Cash & Carry Russia, The Coca Cola Company, Groupe Danone, Nestle S.A., «МегаФон», ВАТ «САН ІнБев», «Кедбері Росія та СНД», Global Spirits (ТМ «Хортиця»), «Славутич», Billa.

Розглянемо існуючі програмні рішення систем відстеження для громадського транспорту.

Одним з перших проектних рішень є відомий та популярний інтернет-сервіс пошуку маршрутів громадського транспорту України «EasyWay», вікно даного додатку наведено на рисунку 1.2.

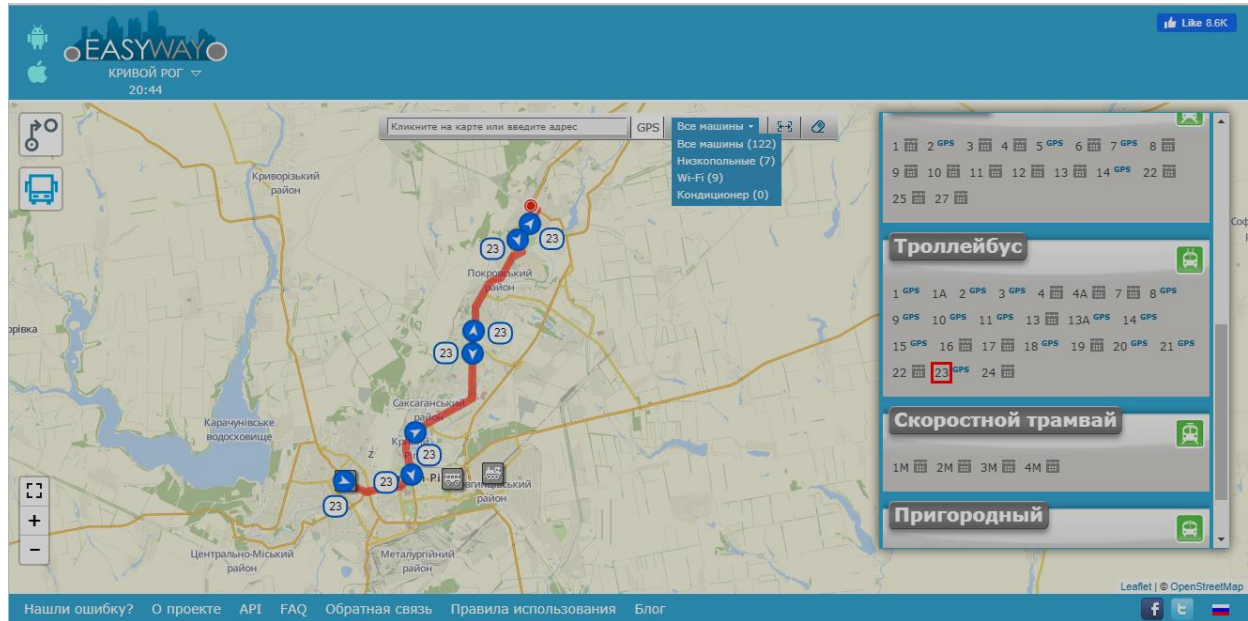


Рисунок 1.2 – Вікно моніторингу громадського транспорту в системі «EasyWay»

Він був створений у 2011 аби допомогти людині зорієнтуватись у незнайомому місті та підказати, який громадський транспорт обрати для свого пересування. Команда EasyWay на постійній основі співпрацює з багатьма міськими радами, безкоштовно надає віджет-систему Сервісу для офіційних міських сайтів та веб-порталів, забезпечуючи жителів корисною та доступною інформацією про громадський транспорт [14].

Для користувачів також доступні безкоштовні мобільні додатки для Android та IOS, для бізнесу надається унікальна послуга API доступу, що дозволяє інтегрувати громадський транспорт у будь-яку інформаційну систему.

Система пошуку маршрутів громадського транспорту EasyWay є соціальним проектом, умовно безкоштовним для використання, натомість пропонується перегляд реклами, яку в додатку можна відключити за певну плату.

Розглянемо мобільні програмні рішення систем відстеження для громадського транспорту.

«Львівський транспорт Онлайн» – це мобільний додаток, завдяки якому можна слідкувати за рухом комунальних автобусів, трамваїв та тролейбусів у режимі реального часу [15]. Також він показуватиме обраний користувачем маршрут, зупинки та розташування транспорту. Таким чином, обравши на карті зупинку, можна побачити, за який час той чи інший транспорт туди прибуде.

Проте додаток працює лише за умови постійного доступу до Інтернету. Вікно даного додатку наведено на рисунку 1.3.

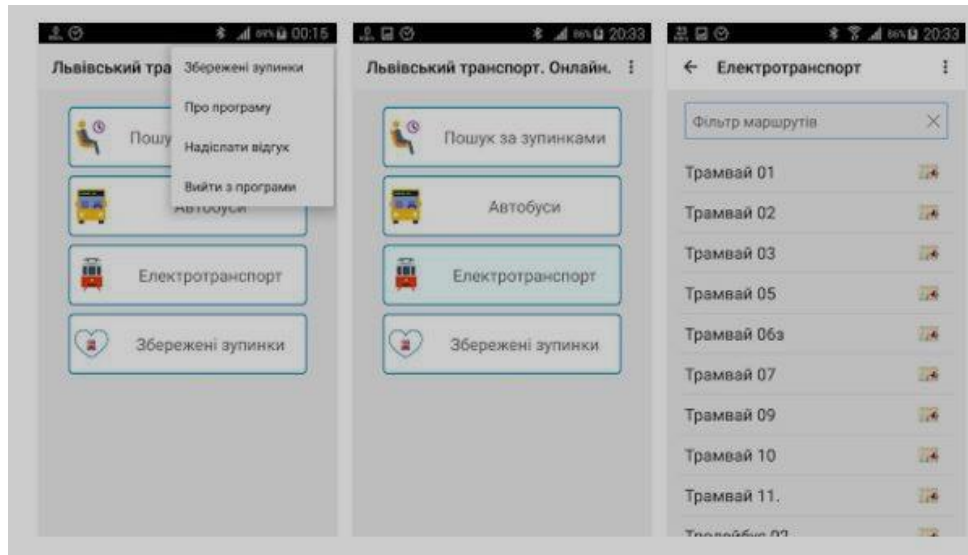


Рисунок 1.3 – Вікно моніторингу громадського транспорту в системі «Львівський транспорт Онлайн»

Функціонал додатку включає в себе наступні можливості: онлайн трекер транспорту у Львові, пошук за наявними зупинками, збереження улюблених зупинок, інформація для електротранспорту та автобусів, відображення транспорту певного маршруту на мапі.

Розробники цього додатку лише надають доступ до інформації із сайту Львівської міської адміністрації [16], через інтерфейс Android аплікації і не несуть відповідальність за достовірність цих даних. Інформація оновлюється онлайн для кожного автобуса, трамвая чи тролейбуса на основі їх GPS розміщення, тому за винятком форс-мажорних обставин (ДТП та ін.), оцінка часу прибуття є точною. Аби полегшити користування міським транспортом для львів'ян, до додатку хочуть долучити і приватних перевізників.

Отже, аналіз ринку програмного забезпечення не виявило систем відстеження для шкільного транспорту. Існуючі проектні рішення дозволяють відстежувати вантажоперевезення, комерційні пасажирські перевезення. Знайдені програмні додатки є або комерційними та дорогими проектами, або соціальні проекти з обмеженими функціональними можливостями.

1.3.2 Інтернет речей процесу відстеження транспорту

Аналіз ринку програмного забезпечення показав відсутність готових рішень для відстеження шкільного транспорту. Для вирішення цієї проблеми можна скористатись проектними рішеннями існуючих систем для відстеження комунального чи промислового транспорту.

Проте одна з найбільших проблем, з якою стикається логістична галузь, – це відстеження власного транспорту. В даному випадку для підвищення

ефективності роботи систем відстеження необхідно впроваджувати новітні технології, які основані на взаємодії декількох пристрів та збереженні великої кількості даних, наприклад Інтернет речей та хмарні обчислення.

Пристрій на основі Інтернету речей може фіксувати безліч параметрів, таких як поведінка водія, час простою і зупинки. Сучасна система стеження за транспортними засобами може використовуватися для контролю палива, контролю навантаження, контролю температури, біометричних даних і так далі.

Інтернет речей по суті робить рішення для відстеження транспортних засобів набагато більш ефективними, економічними і автоматизованими.

Інтернет речей – концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами, за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитування та приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів.

Термін «Інтернет речей», зобов'язаний своєю появою Кевіну Ештону, який в 1997 р, працюючи на компанію Proctor and Gamble, застосував технологію радіочастотної ідентифікації для керування системою поставок [9]. З тих пір Інтернет речей звершив перехід від простих радіочастотних міток до екосистеми і індустрії.

Промисловий Інтернет речей – це один з найбільш великих сегментів Інтернету речей з точки зору кількості підключених пристроїв і ступеня корисності цих сервісів для виробництва і автоматизації підприємств [10]. Цей сегмент традиційно служить операційно–технологічною базою. Сюди входять апаратні і програмні засоби моніторингу фізичних пристроїв. Традиційні завдання інформаційних технологій вирішуються інакше, ніж операційно–технологічні завдання.

Операційні технології (ОТ) зосереджені на оцінці продуктивності, часу безвідмовної роботи, зборі даних і відповідної реакції в режимі реального часу, а також безпеки систем. Інформаційні технології спрямовані на безпеку, групування, сервіси та надання даних. Оскільки Інтернет речей починає займати важливе місце в сфері виробництва і промисловості, світи ІТ і ОТ об'єднуються, особливо в області діагностичного обслуговування тисяч виробничих машин і верстатів, і зможуть забезпечувати безпрецедентним обсягом даних приватні та публічні хмарні інфраструктури.

До характеристик цього сегмента відноситься необхідність надавати операційно–технологічній системі готові рішення в режимі реального часу або майже в режимі реального часу. Це означає, що у всьому, що стосується виробничого цеху, головним параметром для Інтернету речей буде час відгуку. Крім того, важливу роль будуть грати тривалість простою і безпеку. Це має на

увазі потребу в запасі потужності і, ймовірно, в наявності приватних хмарних мереж і сховищ даних. Промисловий Інтернет речей – це один з сегментів на цьому ринку що найбільш швидко розвивається. Важливою особливістю цього напрямку є те, що він спирається на старі технології, тобто на апаратні і програмні засоби, які не можна назвати актуальними.

Приклади застосування Промислового Інтернету Речей.

- профілактичне обслуговування промислового обладнання;
- зростання продуктивності завдяки попиту в реальному часі;
- енергозбереження;
- системи безпеки, такі як вимірювання температури, вимірювання тиску і контроль над витоком газу;
- експертна система для виробничого цеху.

До екосистеми Інтернету речей відносяться усі засоби, сервіси і технології, які використовуються в Інтернеті речей [11].

До них можна віднести:

- розумні датчики/виконавчі механізми: вбудовані системи, операційні системи реального часу, джерела безперебійного живлення, мікро–електромеханічні системи;
- системи зв'язку з датчиками: зона охоплення бездротових персональних мереж становить від 0 см до 100 м. Для обміну даними між датчиками застосовуються низькошвидкісні малопотужні інформаційні канали, які часто побудовані не на протоколі IP;
- локальні обчислювальні мережі: зазвичай це системи обміну даними на основі протоколу IP, наприклад, 802.11 Wi-Fi–мережу для швидкої радіозв'язку, часто це пирингові або зіркоподібні мережі;
- агрегатори, маршрутизатори, шлюзи, пограничні пристрої: постачальники вбудованих систем, самі бюджетні складові (процесори, динамічна оперативна пам'ять і система зберігання даних), виробники модулів, виробники пасивних компонентів, виробники тонких клієнтів, виробники стільникових і бездротових радіосистем, постачальники міжплатформового програмного забезпечення, розробники інфраструктури туманних обчислень, інструментарій для граничної аналітики, безпеку граничних пристроїв, системи управління сертифікатами;
- глобальна обчислювальна мережа: оператори стільникового зв'язку, оператори супутникового зв'язку, оператори малопотужних глобальних мереж. Зазвичай застосовуються транспортні протоколи Інтернету для IoT і мережевих пристроїв;
- хмара: інфраструктура в якості постачальника послуг, платформа в якості постачальника послуг, розробники баз даних, постачальники послуг потокової і пакетної обробки даних, інструменти для аналізу даних, програмне забезпечення в якості постачальника послуг, постачальники озер даних, оператори програмно–визначених мереж / програмно–визначених периметрів, сервіси машинного навчання;
- сервіси аналізу даних: величезні масиви інформації передаються в хмару. Робота з великими обсягами даних і отримання з них користі – це

завдання, що вимагає комплексної обробки подій, аналітики і прийомів машинного навчання;

- безпека: при зведенні всіх елементів архітектури воедино постають питання кібербезпеки. Безпека стосується кожного компонента: від датчиків фізичних величин до ЦПУ і цифрового апаратного забезпечення, систем радіозв'язку і самих протоколів передачі даних. На кожному рівні необхідно забезпечити безпеку, достовірність і цілісність. У цьому ланцюзі не повинно бути слабких ланок, оскільки Інтернет речей стане головною мішенню для атак хакерів в світі.

Архітектура Інтернету речей відрізняється в залежності від реалізації. Тим не менше вона дещо схожа на архітектуру класичних систем. Один із прикладів архітектури показаний на рисунку 1.4.

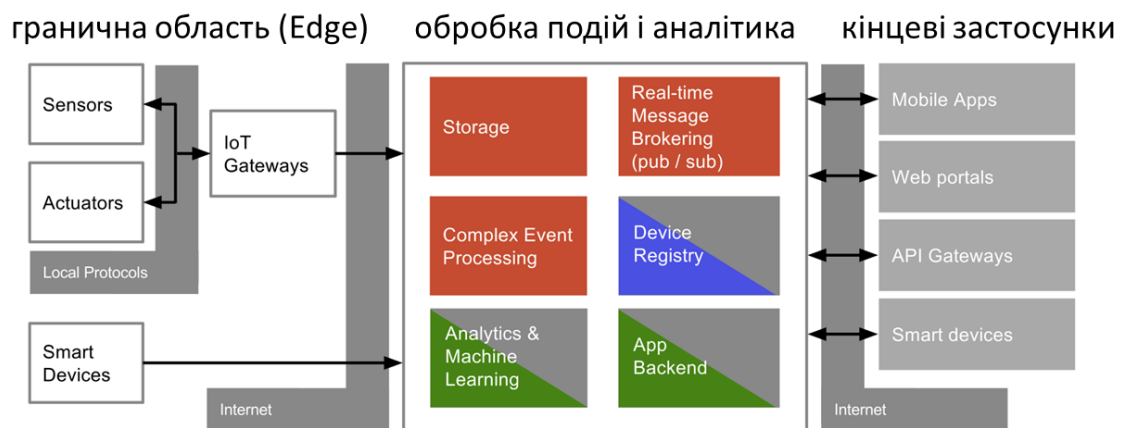


Рисунок 1.4 – Типова архітектура Інтернету речей

Взаємодія з «речами» відбувається через датчики та виконавчі механізми. Ці датчики разом з усією інфраструктурою для інтеграції з рівнем обробки подій через мережу Internet формують так звану граничну область.

Події (дані), що надходять з граничної області зберігаються і обробляються відповідно до задачі та перенаправляються потрібним додаткам. Додатково на цьому рівні відбувається адміністрування та керування пристроями з граничної області. Події обробляються з використанням аналітичних сервісів на основі них проводиться машинне навчання, що дозволяє зробити певні висновки про об'єкт. Цей рівень як правило реалізований з використанням хмарних або туманних обчислень.

Отримання результатів, контроль, віддалене керування та адміністрування системи проводиться через кінцеві програмні додатки з використанням Internet.

Отже, Інтернет речей на даний момент є однією з інноваційною технологією, впровадження якої дозволить підвищити ефективність використання інформаційних технологій.

Для складання екосистеми Інтернету речей системи «Шкільний автобус» ми повинні проаналізувати існуючі інструментальні засоби на можливість реалізації наступних функціональних можливостей системи: збір інформації

про місце розташування шкільного автобусу, збереження часу переміщення та аналіз пройдених зупинок по маршруту, повідомлення клієнтів про прибуття на поточну чи кінцеву зупинки.

Отже, майбутня екосистема повинна мати засоби для збору та передачі GPS–координат автобусу, засоби для збереження та обробки інформації адміністратором системи, засоби для розробки мобільного клієнтського додатку.

Для реалізація технології IoT можна використати такі сучасні хмарні платформи як IBM Bluemix та Microsoft Azure.

IBM Bluemix – публічно–хмарна платформа, розроблена IBM. Підтримує декілька мов програмування і середовищ розробки, а також інструментів в стилі DevOps для побудови, виконання, розгортання та управління додатками в хмарі. Так як Bluemix заснована на Cloud Foundry, то можна користуватися всіма ресурсами зростаючої екосистеми середовищ виконання і служб. Крім додаткових середовищ і служб, Bluemix надає панель управління, яка дозволяє створювати, переглядати і адмініструвати додатки і служби, а також стежити за використанням ресурсів вашого застосування. Крім того, панель управління Bluemix надає можливість керувати організаціями, просторами і доступом користувачів [20].

Bluemix забезпечує доступ до широкого спектру служб, які можна вбудовувати в додатки. Деякі з цих служб відбуваються з Cloud Foundry, інші поставляються IBM і сторонніми постачальниками. В каталог регулярно додаються нові і вдосконалені служби. Ось деякі з часто використовуваних середовищ виконання: Node.js, PHP, Python, Ruby.

Серед підтримуваних мов програмування і платформ програмування – Java, Node.js, Go, PHP, Swift, Python, Ruby Sinatra, Ruby on Rails, крім того, передбачена технологія модульних пакетів збірки (buildpacks), що дозволяє підтримати інші мови, такі як Scala.

Підтримує техніку PaaS – «функція як послуга», реалізовану на базі Apache OpenWhisk. Дана модель хмарних обчислень, при якій користувач отримує доступ для використання інформаційно–технологічну платформу: операційні системи, системну систему управління базами даних, використовуючи програмне забезпечення, засоби розробки та тестування, розміщені в обласному провайдері.

У цій моделі вся інформаційно–технологічна інфраструктура, яка включає в себе обчислювальні мережі, сервери, систему зберігання, цілком керується провайдером, ним же визначається параметри доступних для користувачів віртуальних платформ і набір параметрів цих платформ, створювати їх віртуальні екземпляри, встановлення, розробити, тестувати, експлуатувати на них прикладне програмне забезпечення, при цьому динамічно змінюючи кількість використаних обчислювальних ресурсів.

Розглянемо можливості альтернативної платформи Microsoft Azure.

Microsoft Azure – хмарна платформа компанії Microsoft. Надає можливість розробити, виконати запропоновані та збережені дані на серверах, розташованих у розподілених центрах даних [21].

Microsoft Azure реалізує хмарні моделі платформ як обслуговування (PaaS) та інфраструктури як обслуговування (IaaS). Можливо, використовували як сторонніх, так і сервісних служб Microsoft у своїх моделях ПЗ як послуга (SaaS) [21]. Працездатність платформи Microsoft Azure забезпечує глобальна мережа розподілених центрів даних Microsoft.

Окрім базових функцій операційних систем, Microsoft Azure має і додаткові: виділення ресурсів за вимогою для масштабування, автоматичну синхронну реплікацію даних для підвищення постійної доступності та інше.

Для віртуальних машин доступні образи наступних операційних систем: Windows Server, CoreOS, Ubuntu Server, Red Hat, Clear Linux OS, Debian, SUSE Linux Enterprise Server, Oracle Linux.

Практично всі сервіси Microsoft Azure мають інтерфейс взаємодії API, побудований на основі обмежених можливостей для розподіленої системи REST, що дозволяє розробникам використовувати хмарні сервіси з будь-якою операційною системою, пристрої та платформи.

Окрім того, користувачі можуть створювати та керувати власними сервісами, використовуючи візуальний веб-інтерфейс порталу Azure. Портал може налаштовувати сервіси, редагувати права доступу, відстежувати стан ресурсів і керувати біллінгом.

Для всіх мов програмування, що підтримує Microsoft надаються набори бібліотеки. На даний момент підтримуються наступні мови та програмні платформи: .NET, Java, Node.js, Python, PHP, Go.

Для організації фізичної моделі IoT необхідно обрати певну платформу, наприклад Arduino або Raspberry Pi.

Arduino — торгова марка апаратно-програмних засобів для побудови простих систем автоматики та робототехніки, орієнтована на не професіональних користувачів.

Програмна частина складається з безкоштовної програмної оболонки (IDE) для написання програм, їх компіляції та програмування апаратури. Апаратна частина — набір змонтованих друкованих плат. Повністю відкрита архітектура системи дозволяє вільно копіювати та поповнювати лінійку продукції Arduino.

Arduino може використовуватись як для створення автономних об'єктів автоматики, так і підключатись до програмного забезпечення на комп'ютері через стандартні проводні і безпроводні інтерфейси

Raspberry Pi — одноплатний комп'ютер розміром з банківську картку. Часто використовується як мозок робота, домашній сервер чи просто «комп'ютер». Спочатку проект створювався як освітній, Raspberry Pi відмінно підходить для вивчення основ електроніки.

До вказаних плат можна підключити велику кількість датчиків, як рухів, температур, тиску, передачі даних.

Для розробки мобільного додатку для відстеження шкільного автобусу можна використати середовище розробки Android Studio або Visual Studio.

Android Studio — це інтегроване середовище розробки для роботи з платформою Android, анонсована 16 травня 2013 року на конференції Google I/O.

Android Studio, побудована на програмному забезпечення IntelliJ IDEA от компанії JetBrains, – офіційний засіб розробки Android додатків. Дане середовище розробки доступне для Windows, OS X и Linux. В травні 2017 року, на щорічній конференції Google I/O, Google анонсував підтримки мову Kotlin, який використовується в Android Studio, як офіційної мови програмування для платформи Android в додаток до Java и C++.

Іншим популярним кроссплатформеним середовище розробки додатків для мобільних пристроїв є Visual Studio.

Використовуючи Visual Studio, можна створювати додатки для пристроїв Android, iOS і Windows [22]. При розробці програми доступні інструменти Visual Studio для додавання підключених служб, таких як Office 365, служба додатків Azure і Application Insights. Підтримується створення додатків за допомогою C # і .NET Framework, HTML і JavaScript або C ++. Існує можливість спільного використання коду, рядків, зображень, а в деяких випадках навіть призначеного для інтерфейсу користувача.

Отже, ринок інтегрованих середовищ розробки програмних додатків дозволяє достатні альтернативи для розробки Інтернету речей системи «Шкільний автобус».

1.4 Висновки до розділу 1

Саме з Інтернетом речей більшість науковців та винахідників пов'язують четверту промислову революцію (Індустрія 4.0). Переважна більшість експертів погоджуються, що Інтернет речей до 2025р. набуде широкого розповсюдження і принесе користь людству.

Інтернет речей – це сукупність фізичних об'єктів, комплексно з'єднаних між собою за допомогою сучасних Інтернет технологій. Він дає можливість появи нових, більш ефективних методів виробництва та ведення бізнесу.

Технології, які дозволяють реалізувати Інтернет речей, вирішують чотири основні задачі: ідентифікацію, збирання даних, зберігання даних та обмін інформацією:

- для ідентифікації елементів мережі потрібен простий у виконанні алгоритм ідентифікації, що дозволить передавати дані не перевантажуючи мережу. Для визначення точного місця знаходження об'єкта вже довгий час існує технологія GPS, що використовується у смартфонах, навігаторах та інших гаджетах;

- щоб збирати потрібні дані, слідкуючи за змінами в оточуючому середовищі, об'єкти повинні оснащуватися відповідними сенсорами;

- для обробки та накопичення даних з сенсорів, в об'єкти має бути вбудований комп'ютер (це може бути просто ще один чіп);

- для обміну інформацією між пристроями можуть бути використані як дроти, так і технології бездротових мереж (наприклад, Wi-Fi та Bluetooth).

Аналіз предметної області показав відсутність проектних рішень застосування технології Інтернет речей та хмарних обчислень для відстеження шкільних автобусів, тому розробка екосистеми Інтернет речей системи «Шкільний автобус» є актуальною та доцільною.

Результати роботи можуть бути використані надалі для впровадження технології Інтернету речей в систему відстеження транспорту для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, з метою підвищення ефективності безпеки життєдіяльності.

2 МОДЕЛІ ТА МЕТОДИ ЕКОСИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ «ШКІЛЬНИЙ АВТОБУС»

2.1 Опис моделі екосистеми Інтернету речей «Шкільний автобус»

Аналіз предметної області показав, що в існуючій державній програмі «Шкільний автобус» задіяні наступні учасники:

- адміністрація школи, яка організовує та контролює своєчасне підвезення учнів до навчального закладу;
- батьки учнів, які дають згоду у вигляді письмової заяви на участь в організованому та безпечному перевезення дітей;
- учні, яких необхідно доставляти в навчальний заклад;
- вчителі школи, які мають право або контролювати процес перевезення школярів закріпленого класу або користуватись програмою «Шкільний автобус»;
- водії автобусів, які надають послуги перевезення учнів сільської місцевості.

Система «Шкільний автобус» проектується для підвищення безпеки перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості за рахунок відстеження поточних координат шкільного транспорту, сповіщення користувачів системи про актуальні координати шкільного автобусу та реакція системи на його розташування відносно координат певних користувачі чи об'єктів (школи, зупинок).

Моделі процесів описують взаємодії між усіма учасниками системи, які виконують або активну участь (задання вхідних параметрів), або пасивну (перегляд вихідних параметрів).

Розглянемо формальний опис кожного учасника системи детально.

Адміністрація школи розпочинає процес організованого підвезення учнів до школи, проводить конкурс серед перевізників, визначає умови перевезення, складає маршрут автобусу та розклад руху, контролює процес підвезення учнів та працівників. Проте системою можуть в подальшому користуватись не лише представники адміністрації, але й черговий вчитель, класні керівники:

$$School = \{teacher_i\}, i = 1..n$$

Для створення маршруту перевезення учнів адміністрація школи проводить опитування серед батьків учнів, які безпосередньо дають згоду на підвезення своїх дітей до навчальних закладів, зацікавлені в безпеці та контролі місця розташування дітей в навчальний час та в час транспортування. Тому виділяємо множину батьків:

$$Parents = \{Parents_i\}, i = 1..n$$

Наступним учасником системи є учні навчального закладу. Саме їх місце проживання впливає на систему зупинок. Саме учням важливо отримати інформацію про поточне розташування автобусу на маршруті для своєчасного прибуття на зупинку. Тому виділяємо множину учнів:

$$Pupil = \{pupils_i\}, i = 1..m$$

Головним учасником системи є шкільний автобус, розташування якого важливе для всіх учасників системи:

$$Bus = \{buses_i\}, i = 1..m$$

На вході всі учасники системи отримують відповідно від автобусу його вихідні поточні координати.

Системна модель «Шкільний автобус» – це формальне представлення взаємозв'язків між усіма учасниками системи. Вона зображена на рисунку 2.1.

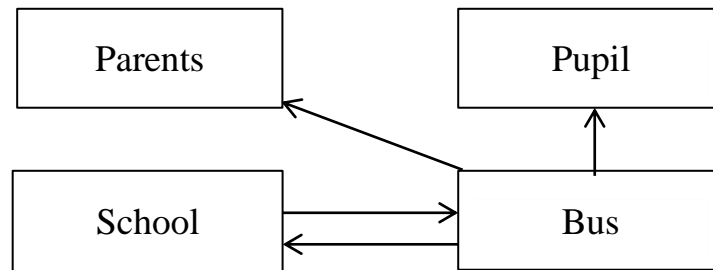


Рисунок 2.1 – Структура системної моделі

Для кожного автобусу можна поставити у відповідність дані про водія, контактний телефон, маршрут рейсу, початкова та кінцева зупинка, орієнтовний час перебування в дорозі. Відповідні, автобус можна представити у вигляді наступного набору:

$$Bus = \langle id, driver, phone, reis, start, finish, time \rangle$$

В цьому набор допускається збіг стартової та фінішної зупинки, у випадку коли автобус починає та закінчує маршрут з шкільного ганку.

Елемент *reis* – це елемент множини

$$reis = \{station_i, time_st_i, x_i, y_i \quad i = 1..n\}$$

який містить перелік всі зупинок на маршруті та відповідний час автобуса прибуття на зупинку, координати їх розташування на карті. В цьому наборі не допускаються збіги.

Модель структури рейсу представимо у вигляді орієнтованого графа:

$$reis = \langle S, R \rangle$$

где *S* – множина зупинок на маршруті;

R – множина бінарних відношень, які характеризують зв'язки по часу між зупинками.

Про кожного учня необхідно знати обов'язковий набір:

$$Pupil = \langle id, station, reis \rangle$$

який обов'язково містить зупинку, на якій дитина буде чекати автобус.

Для кожного батька дуже важлива інформація чи прибув і коли автобус на зупинку саме його дитини, тому елемент *Parents* буде містити наступний набір:

$$Parents = \langle id, id_pupil_i \rangle$$

Як бачимо батьки можуть отримувати інформацію про розташування автобусу для кожної своєї дитини, наприклад якщо колись будуть чекати автобус в різний час та місцях.

Про школу (викладачів) необхідно фіксувати наступний набір

$$School = \langle id, x_school, y_school \rangle,$$

який містить інформацію про місце розташування школи.

Всі приведені моделі можна представити у вигляді орієнтованого графу, який наведено на рисунку 2.2.

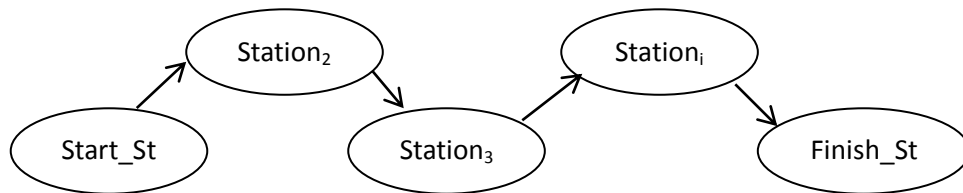


Рисунок 2.2 – Модель графу рейсу

Аналізуючи граф, можна побачити, що шкільний автобус необов'язково може розпочинати маршрут з шкільного ганку.

Взаємозв'язок між учнями, батьками та зупинками можна представити у вигляді дерева, яке представлено на рисунку 2.3.

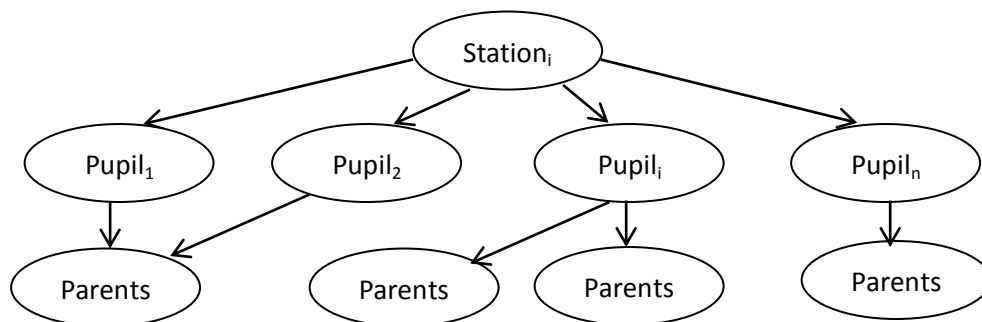


Рисунок 2.3 – Модель графу станція

Аналізуючи граф, можна зробити висновок, що по-перше, на зупинці можуть бути декілька учнів з однієї родини, по-друге, відстежувати автобус для своєї дитини можуть одночасно обидва з батьків.

Отже, нами була описана структура системної моделі, яка складається з активних учасників системи, а саме адміністратора школи, який разом з водієм формує маршрут автобусу, надалі будемо його називати «адміністратор», та пасивних учасників – «користувачів», до яких можна віднести вчителів, учнів та їх батьків.

Для моделювання системи відстеження шкільного автобусу використаємо методологію сімейства IDEF, за допомогою яких можна ефективно відобразити та аналізувати моделі діяльності широкого спектру складних систем в різних напрямках. При цьому глибина дослідження процесів в системі визначається

самостійно розробником, що дозволяє не перенавантажувати створену модель зайвими даними.

В нашому дипломному проекті скористаємось стандартом IDEF0, який відноситься до методології функціонального моделювання та графічної нотації, яка представлена для формалізації та опису бізнес–процесів

На рисунку 2.4 представлений головний вузол системи відстеження місця розташування шкільного автобусу.

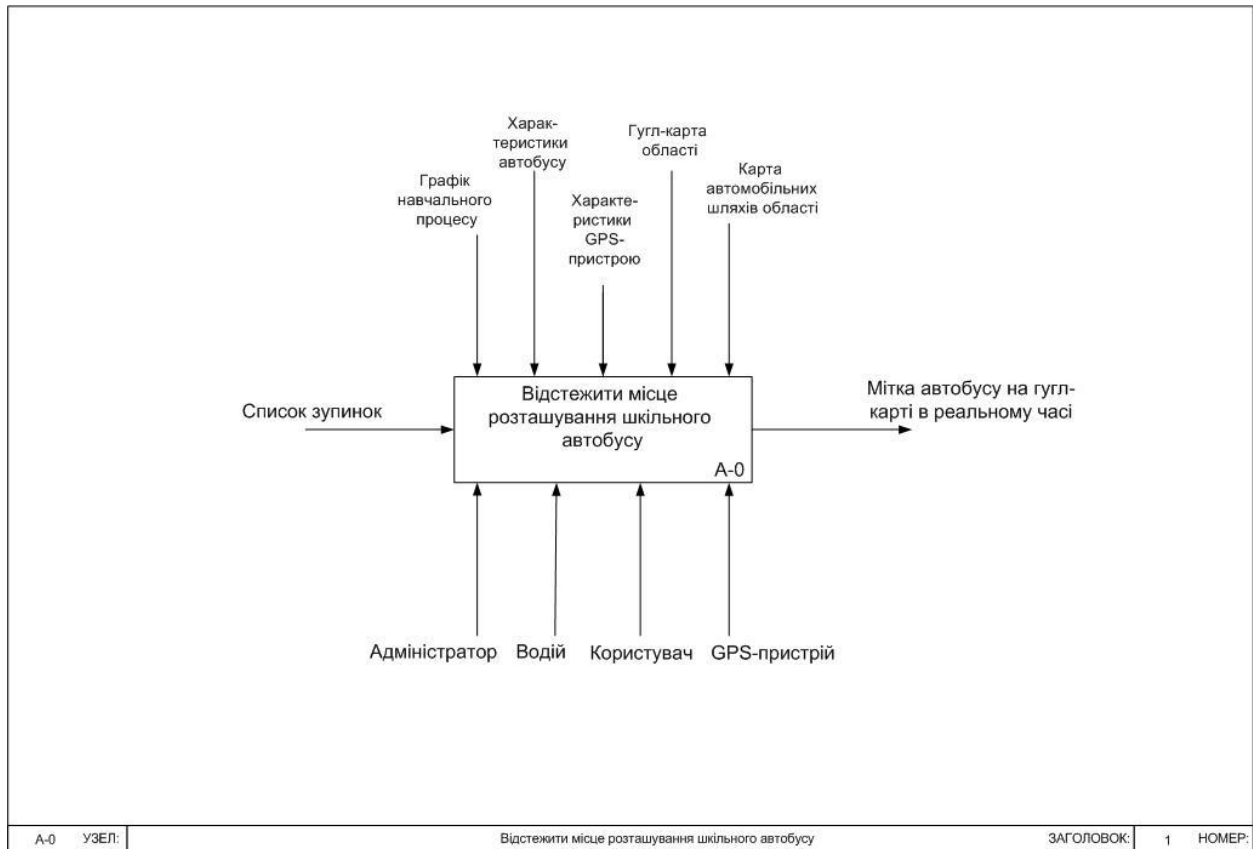


Рисунок 2.4 – Діаграма вузлу «Відстежити місце розташування шкільного автобусу»

Вхідними даними основного вузла є список зупинок, який був складений адміністратором школи на підставі заяв батьків.

На виході очікується мітка поточного розташування автобусу на гул–карті в реальному часі.

Всі дії в даній системі будуть виконуватись враховуючи:

- графік навчального процесу, а саме час початку та кінця навчального процесу;

- характеристики автобусу, саме стан автобусу, його технічні параметри, наприклад швидкість, які будуть впливати на формування розкладу руху автобусу;

- особливості передачі сигналу GPS–пристроєм, який буде встановлений в автобусі чи на смартфоні водія;

- карта автомобільних шляхів області, яка дозволить сформувати маршрут автобусу;
- інтерактивні гугл– карти області для відображення мітки поточного розташування автобусу.

На рисунку 2.5 наведена декомпозиція основного вузла системи, яка складається з чотирьох операцій: «Сформувати розклад руху автобусу», «Сформувати маршруту автобусу на гугл–карті», «Визначити місце розташування автобусу на маршруті», «Відобразити автобус на гугл–карті».

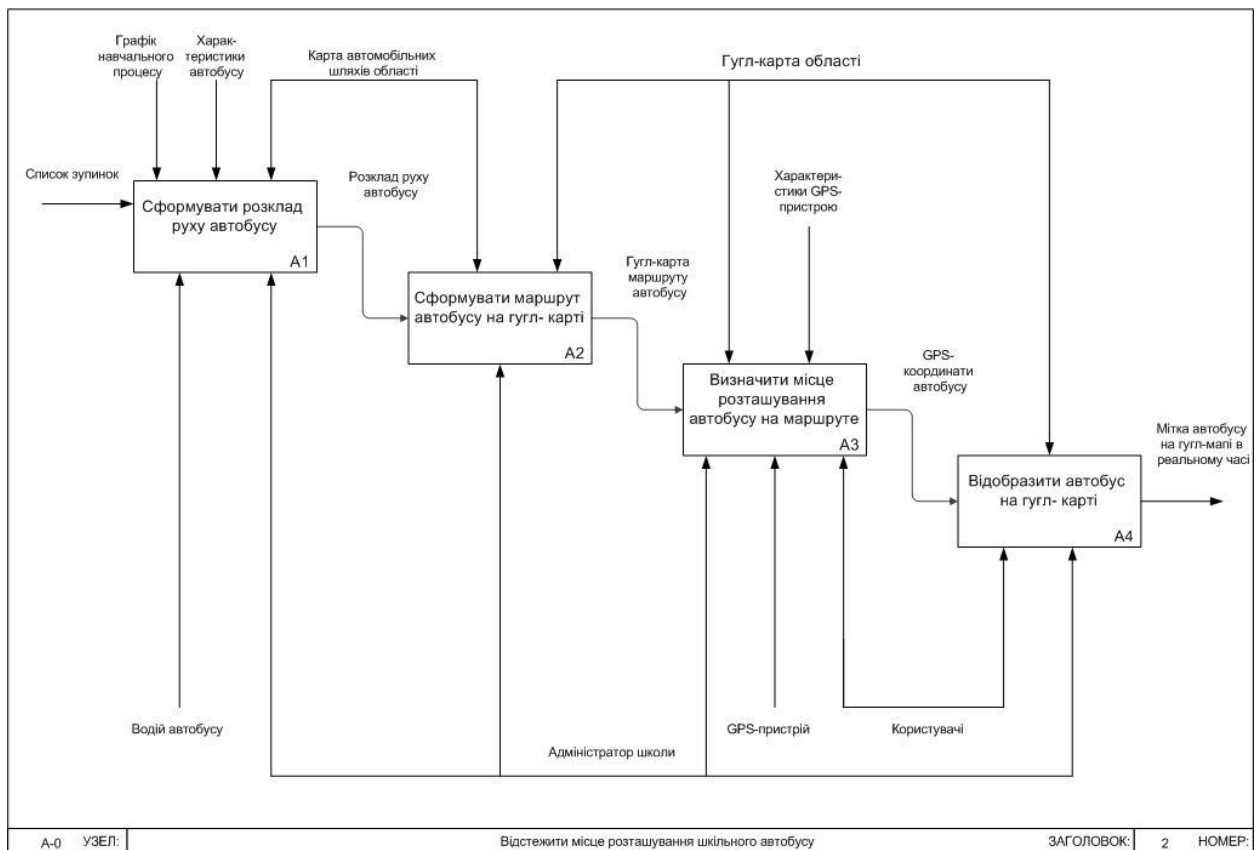


Рисунок 2.5 – Діаграма декомпозиції вузлу «Відстежити місце розташування шкільного автобусу»

Розглянемо детально кожну операцію.

Процес відстеження шкільного автобусу розпочинається з операції «Сформувати розклад руху автобусу». Водій автобусу разом з адміністратором школи аналізує графік навчального процесу, час початку та закінчення навчальних занять, враховуючи карту автомобільних шляхів області та характеристики автобусу (швидкість руху на різних ділянках) складають розклад руху автобусу. Він містить перелік усіх зупинок (разом зі стартовою та фінішною), час прибуття на ці зупинки.

Після створення розкладу руху автобусу починається процес «Сформувати маршруту автобусу на гугл–карті». Згідно складено розкладу адміністратор визначає географічні координати розташування на карті автомобільних шляхів області та створює цей маршрут на гугл–карті області.

Наступна операція «Визначити місце розташування автобусу на маршруті» може бути викликана або адміністратором або користувачами системою, при цьому йде звернення до встановленого на автобусі GPS-пристрою, враховуючи принципи роботи якого визначаються вихідні дані – GPS – координати автобусу.

Отримані GPS – координати автобусу є вхідними даними для останнього процесу «Відобразити автобус на гугл-карті». Цей процес можуть запустити всі користувачі системою, адміністратор школи та, користуючись гугл-картою, побачити мітку автобусу на гугл-карті в реальному часі.

Розглянемо декомпозицію процесу «Визначити місце розташування автобусу на маршруті», яка зображена на рисунку 2.6.

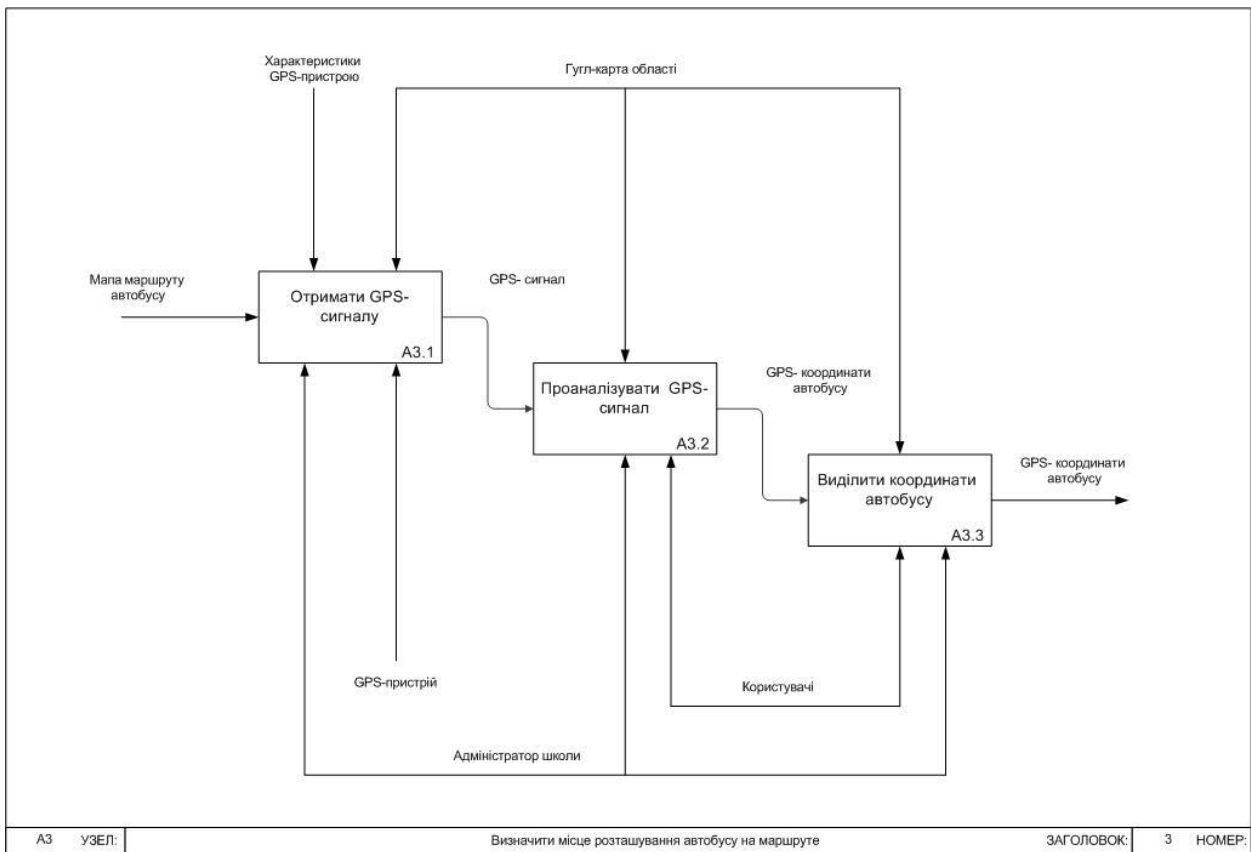


Рисунок 2.6 – Діаграма декомпозиції вузлу «Визначити місце розташування автобусу на маршруті»

Процес визначення місця розташування автобусу на маршруті можна поділити на три етапи: «Отримати GPS-сигнал», «Проаналізувати GPS-сигнал», «Виділити координати автобусу».

Саме на цьому етапі в системі визначення місця розташування автобусу на маршруті необхідно та доцільно використати технології «Інтернет речей» для визначення моменту сповіщення про наближення шкільного автобусу до певної зупинки відносно певного користувача.

Технологія «Інтернет речей» дозволить, використовуючи отримані координати місця знаходження автобусу, географічні координати певної

зупинки та отримані координати певного пристрою користувача (наприклад, смартфона з GPS-навігацією), визначати момент наближення автобусу до користувача з заданим пристроєм навігації.

Враховуючи зроблений формальний опис системи та спроектовану модель процесу відстеження місця розташування шкільного автобусу, розглянемо модель екосистеми Інтернету речей системи «Шкільний автобус», яка зображена на рисунку 2.7.

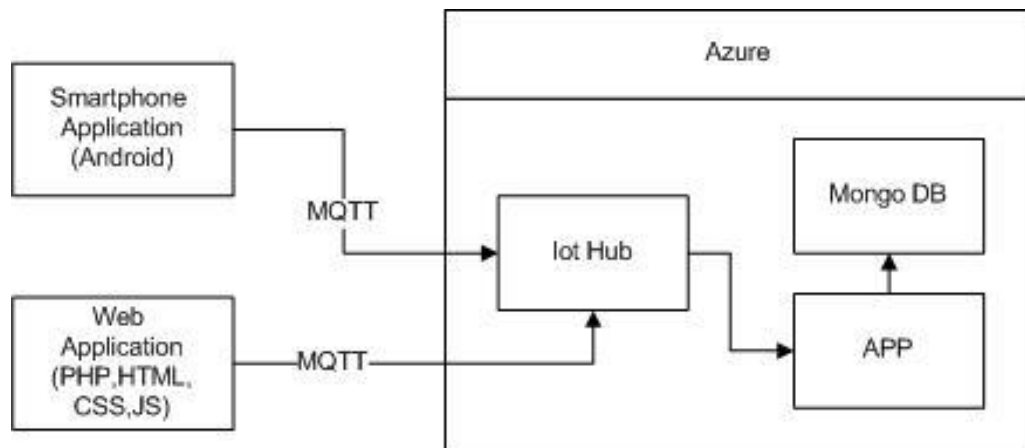


Рисунок 2.7 – Модель екосистеми Інтернету речей системи «Шкільний автобус»

В даній системі будуть взаємодіяти два пристрої, а саме смартфони, водія та користувача системи, які мають GPS-пристрої та працюють на операційній системі Android.

Для оновлення даних, наприклад про координати зупинок, бажана дистанція для відстеження між автобусом та користувачем, буде розроблено вебдодаток, наприклад засобами PHP, HTML, CSS, JS. Цей додаток буде містити функціональні можливості для аналізу GPS-сигналу про місце розташування автобусу, сповіщення користувачів про рух автобусу відносно зупинок.

Для обміну повідомленнями між пристроями та додатком Інтернету речей доцільно скористатись мережевим протоколом MQTT.

Для забезпечення надійний обміном даними з високим рівнем безпеки між додатком Інтернету речей (APP) та GPS-пристроєм, необхідно використати сервіси хмарної платформи Azure. Центр Інтернету речей (IoT Hub) дозволить серверній частині рішення, розміщеного в хмарі, віртуально підключатись до потрібного пристрою, отримані характеристики сигналу зберігати у відповідній базі даних MongoDB.

Отже, на цьому етапі було спроектовано модель екосистеми Інтернету речей системи «Шкільний автобус», яка дозволить реалізувати процес відстеження місця розташування шкільного автобусу.

2.2 Методи відстеження місця розташування шкільного автобусу

Для забезпечення функції сповіщення користувачів екосистеми Інтернету речей системи «Шкільний автобус» про наближення автобусу, що має власний GPS – пристрій, до певного користувача з іншим GPS – пристроєм, наприклад смартфоном, в площині певної зупинки, необхідно розглянути методи визначення координат характерних точок об'єктів нерухомості та визначення відстані між двома об'єктами заданими своїми координатами.

Розглянемо методи визначення координат характерних точок об'єктів нерухомості [23]. В геодезії існує певний набір таких методів:

- геодезичний метод (наприклад, метод триангуляції, полігонометрії, трилатерації, метод прямих, зворотних чи комбінованих зарубок);
- метод супутникових геодезичних вимірювань (визначень);
- фотограметричний метод;
- картометричний метод;
- аналітичний метод.

Вибір методу визначення координат характерних точок земельних меж залежить від точності визначення таких координат, яка встановлена для земельних ділянок певного цільового призначення та певного дозволу використання [24].

Триангуляція – це один з методів створення мережі опорних геодезичних пунктів та сама мережа, створена цим методом. Він полягає в побудові рядів та мереж, трикутників які примикають один до одного та в визначенні положення їх вершин в обраній системі координат. В кожному трикутнику вимірюють всі три кута, а одну з його сторін обчислюють шляхом послідовного рішення попередніх трикутників, починаючи з того з них, у якому одна з його сторін отримана шляхом вимірювань.

Методом полігонометрії шляхом вимірювання на місцевості довжин ліній, які послідовно з'єднують ці пункти та створюють полігонометричний хід, та горизонтальних кутів між ними.

Трилатерація – метод визначення розташування геодезичних пунктів шляхом побудови на місцевості системи суміжних трикутників, в яких вимірюються довжини їх сторін [25].

Метод прямих, зворотних чи комбінованих зарубок – це метод визначення координат певної точки вимірюванням елементів, які пов'язують її положення з вихідними пунктами.

Метод супутникових геодезичних вимірювань [26].

Визначення координат точок земної поверхні за допомогою супутників засноване на радіодальномерних вимірах дальностей від супутників до приймача, встановленого у певній точці. Якщо виміряти дальності R_1 , R_2 і R_3 до трьох супутників (рисунок 2.8), координати яких на даний момент часу відомі, то методом лінійної просторової зарубки можна визначити координати точки розташування приймача P .

Через несинхронність ходу годинника на супутнику і в приймачі певні до супутників відстані будуть відрізнятись від справжніх. Такі помилкові відстані

отримали назву «псевдодальностей». Для виключення цих похибок визначення координат точок з достатньою точністю можливо при одночасному спостереженні не менше 4 супутників.

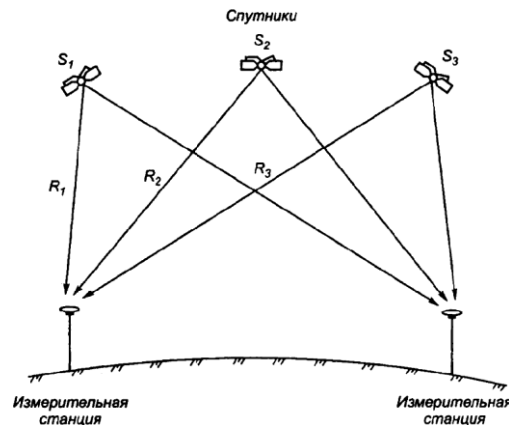


Рисунок 2.8 – Принципова схема супутникової системи позиціонування

Системи супутникового позиціонування працюють в грінвічській просторовій прямокутній системі координат з початком, який збігається з центром мас Землі. При цьому система GPS використовує координати світової геодезичної системи WGS-84 (World Geodetic System, 1984 г.), а ГЛОНАСС – систему координат ПЗ-90 (Параметри Землі, 1990 г.). Обидві координатні системи встановлені незалежно один від одного за результатами високоточних геодезичних і астрономічних спостережень. Оскільки ці координатні системи засновані на різних еліпсоїдах і орієнтовані на різні території, то геодезичні та прямокутні координати одних і тих же точок земної поверхні в цих системах не збігаються.

Більшість сучасних приймачів працюють з супутниками GPS, тому координати вимірюваних точок отримують найчастіше в системі WGS-84. Для переходу до державної або місцевої системи координат використовують передбачену програмами обробки функцію трансформування.

Як зазначалося раніше, визначення відстаней від супутникового приймача до супутника є не що інше, як радіодальномірні вимірювання: приймач приймає електромагнітні коливання з супутника, порівнює їх зі своїми, виробленими власним генератором і в результаті визначає дальність до космічного апарату. Дальності вимірюють двома способами – кодовим і фазовим. У першому випадку порівнюють коди отриманого з супутника сигналу і згенерованого в самому приймачу, а в другому – фази. Найбільш точним є фазові зміни. В GPS всі супутники працюють на одних і тих же частотах, але кожен має свій код. У ГЛОНАСС, навпаки, кожен супутник має свою частоту, але коди у всіх однакові.

Перенесення від супутника до приймача всієї інформації здійснюється за допомогою так званих несучих електромагнітних коливань, випромінюваних на двох частотах L1 і L2. Відповідно до цього на практиці використовують як одночастотні приймачі, що працюють тільки з частотою L1, так і двочастотні,

що використовують обидві частоти. Двочастотні приймачі дають більш високу точність визначення координат.

Способи позиціонування [27] можна розділити на дві групи – абсолютні визначення координат кодовим методом (розрізняють автономний і диференційний методи) і відносні фазові вимірювання (методи «статика» і «кінематика»).

При виконанні абсолютних вимірювань визначаються повні координати точок земної поверхні. Спостереження, що виконуються на одному пункті незалежно від вимірювань на інших станціях, називаються автономними. Автономні спостереження дуже чутливі до всіх джерел похибок, забезпечують точність визначення координат від декількох метрів і використовуються для знаходження наближених координат.

Для підвищення точності абсолютні вимірювання можна виконувати одночасно на двох пунктах: базовій станції P1, розташованій на точці з відомими координатами (зазвичай пункті державної геодезичної мережі), і рухомий станції P2, встановленої над обумовленою точкою (рисунок 2.9).

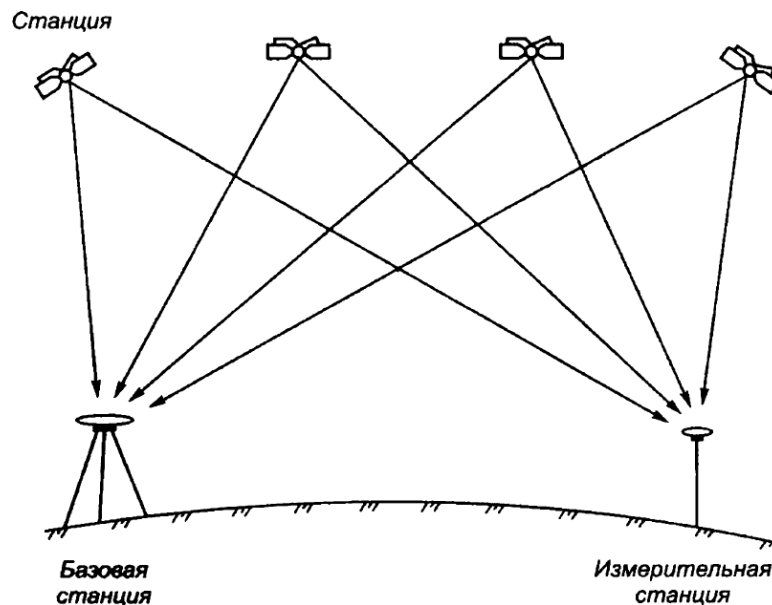


Рисунок 2.9 – Сутність диференційного способу позиціонування

На базовій станції виміряні відстані до супутників порівнюють з обчисленими за координатами і визначають їх різниці. Ці різниці називають диференціальними поправками, а спосіб вимірювання – диференціальним. Диференціальні поправки враховуються в ході обчислень координат рухомий станції після вимірів або при використанні радіомодемів вже в процесі вимірювань. Диференціальний метод заснований на тому міркуванні, що при відносно невеликих відстанях між станціями P1 і P2 (зазвичай не більше 10 км) похибки вимірювань на них практично однакові. При збільшенні відстані між станціями точність падає. Для підвищення точності вимірювань збільшують час спостережень, яке може коливатися від декількох хвилин до декількох годин.

Для вирішення геодезичних завдань, коли необхідно отримувати координати точок з високою точністю, використовують відносні вимірювання, при яких дальності до супутників визначають фазовим методом, і по ним обчислюють збільшення координат або вектора між станціями, на яких встановлені супутникові приймачі.

Обчислення середньої похибки розташування характерних точок, отриманого методом супутникових геодезичних вимірювань, проводиться з використанням програмного забезпечення, за допомогою якого виконується обробка матеріалів супутникових спостережень, а також за базовою формулою [27].

Визначення середньої похибки при цьому методі можна також здійснити за такою формулою:

$$m_p = a + bD, \quad (2.1)$$

де D – відстань між базовим та рухомим приймачами, км.

Фотограметричний метод полягає у визначенні координат межових знаків по знімках, отриманих в результаті дистанційного зондування Землі.

При визначенні місця розташування характерних точок, суміщених з контурами географічних об'єктів, зображених на аерофотознімки, середню квадратичну погрішність місця розташування характерних точок при проведенні кадастрових робіт прийняти вважати рівною:

$$m_p = KxM, \quad (2.2)$$

де M – знаменник масштабу аерознімку (космознімку);

K – коефіцієнт, рівний 0,0005 м.

Картометричний метод полягає у визначенні координат межових знаків за картографічним матеріалом. Вибір масштабу картографічного матеріалу залежить від необхідної точності. Як правило, використовуються карти великого масштабу: 1: 100 – 1: 5000.

На картографічному матеріалі, як правило, відображається квадратна координатна сітка зональної системи плоских прямокутних координат. Сторони квадратів цієї сітки зазвичай виражаються цілим числом кілометрів, тому її називають кілометровою сіткою. Лінії кілометрової сітки, проведені з півночі на південь паралельні осьового меридіану зони (вісь X), а лінії, проведені з заходу на схід – паралельні екватору (вісь Y).

Для визначення плоских прямокутних координат точки на карті знаходять квадрат кілометрової сітки в якому вона знаходиться, а потім опускають з точки перпендикуляри до сторін квадрата. За допомогою масштабу картографічного матеріалу визначають довжини перпендикулярів. Знаючи значення координат ліній квадрата кілометрової сітки обчислюють шукані значення координат точки.

Обчислення середньої похибки розташування характерної точки залежить від картографічного матеріалу, використовуваного при визначенні координат:

1) Використання топографічних планів і карт на паперовому носії.

При визначенні місця розташування характерних точок, суміщених з контурами географічних об'єктів, зображених на карті (плані), середня

квадратична похибка розташування характерної точки при проведенні кадастрових робіт прийняти вважати рівною:

$$m_p = K \times M, \quad (2.3)$$

де M – знаменник масштабу карти (плану);

K – коефіцієнт, рівний 0,0005 м.

2) Використання растрових зображень.

За растровому зображенню визначають координати чотирьох перехресть координатної сітки і порівнюють їх зі справжніми координатами. Фактично визначається якість сканування топографічної карти або плану:

$$m_p = \sqrt{\frac{\sum (dX_i)^2 + \sum (dY_i)^2}{4}} \quad (2.4)$$

Аналітичний метод.

Під аналітичним методом визначення координат розуміється визначення координат характерних точок в результаті розрахунків або за допомогою геоінформаційних систем (наприклад, визначення координат утворених в результаті поділу нових земельних ділянок, межі яких визначені методом проектування в камеральних умовах) [27].

Також даний метод поширюється на випадки, коли характерні точки знову утвореного об'єкта нерухомості приймаються рівними точкам (збігаються з точками), відомості про яких містяться в кадастрі (наприклад, визначення координат нової земельної ділянки, отриманого шляхом об'єднання внесених в кадастр суміжних ділянок).

Величина середньої квадратичної похибки місця знаходження характерних точок при аналітичному методі приймається рівною величині середньоквадратичної похибки місця знаходження характерних точок, використовуваних для обчислень.

При відсутності на момент проведення кадастрових робіт можливості візуального огляду підземних конструктивних елементів будівлі, споруди або об'єкта незавершеного будівництва середня квадратична похибка розташування характерної точки контуру підземного конструктивного елементу будівлі, споруди або об'єкта незавершеного будівництва визначається за такими формулами:

1) при обчисленні координат характерних точок контуру підземного конструктивного елементу будівлі, споруди або об'єкта незавершеного будівництва на підставі отриманих значень координат характерних точок контуру наземних конструктивних елементів, результатів внутрішнього обміру і товщини огорожувальних конструкцій (стін) конструктивних елементів:

$$M_t = \sqrt{m_n^2 + m_n^2 + m_k^2} \quad (2.5)$$

де M_t – середня квадратична похибка розташування характерної точки контуру підземного конструктивного елементу;

m_n – середня квадратична похибка розташування характерної точки контуру наземного конструктивного елементу;

m_n – середня квадратична похибка лінійних (лінійно–кутових) вимірювань параметрів підземних конструктивних елементів;

m_k – середня квадратична похибка передачі координат з наземного на підземний конструктивний елемент будівлі.

2) при обчисленні координат характерних точок контуру підземних конструктивних елементів, місце розташування яких визначено з використанням приладів пошуку (наприклад, трасошукачів, георадарів, трубокабелешукачів, тепловізорів):

$$M_t = \sqrt{m_m^2 + m_p^2} \quad (2.6)$$

де M_t – середня квадратична похибка розташування характерної точки контуру підземного конструктивного елемента;

m_m – середня квадратична похибка розташування характерної точки проєкції підземного конструктивного елемента на поверхню земельної ділянки;

m_{np} – середня квадратична похибка визначення місця розташування підземних конструктивних елементів прибором пошуку.

При цьому величина середньої квадратичної похибки місця характерної точки контуру підземного конструктивного елемента не обмежується значеннями точності визначення координат характерних точок меж земельних ділянок, і може перевищувати зазначені там значення середніх квадратичних похибок для відповідних категорій земель і дозволеного використання земельних ділянок.

Отже, розглянувши всі методи визначення координат характерних точок об'єктів нерухомості, наприклад в нашому випадку – будівлі школи, зупинок транспорту в селищах, а також рухомих об'єктів – автобусу та користувачів системи будемо використовувати метод супутникових геодезичних вимірювань. Він дозволить визначати GPS– координати як сталих об'єктів, так і рухомих. При цьому треба врахувати, що географічні координати одного об'єкту можуть бути представлені в різних форматах. В залежності від того, як представлені хвилини та секунди – як значення від 0 до 60 чи від 0 до 100 (десяткові долі).

Формат координат зазвичай записують наступним чином: DD – градуси, MM – хвилини, SS – секунди, якщо хвилини та секунди представлені як десяткові долі, то записують просто DD.DDDD. Наприклад:

- DD MM SS: 50° 40' 45" с.д., 40 50' 30" п.ш. – градуси, хвилини, секунди;

- DD MM.MM: 50° 40.75' с.д., 40 50.5' п.ш. – градуси, десяткові хвилини;

- DD.DDDDD: 50.67916 с.д., 40.841666 п.ш. – десяткові градуси.

Найпоширенішим способом для збереження координат обирають представлення у вигляді DD.DDDDD.

Для реалізації екосистеми Інтернету речей системи «Шкільний автобус» наступним кроком необхідно розробити метод для визначення моменту сповіщення користувача про наближення автобусу, тобто метод для взаємодії двох Gps– пристроїв, наприклад, смартфона водія автобусу та користувача (учня, його батьків, вчителів). Цей метод буде базуватись на порівнянні відстані

між автобусом та користувачем, між автобусом та найближчою зупинкою, яка ще не була пройдена.

Розглянемо методику розрахунку відстаней між об'єктами за їх географічними координатами [28].

Найкоротшою відстанню між точками заданими своїми географічними координатами є довжина дуги кола, проведеного на сфері по цим двом точкам. Це означення базується на концепції, що форма Землі може бути описана як сфера. Тому в навігації найчастіше використовується для обчислення найкоротшої відстані між точками поверхні Землі рівняння для обчислення відстані на великому колі.

Обчислення відстані саме цим методом більш ефективніше і в багатьох випадках більш точніше, чим обчислення його для спроектованих координат (в прямокутних системах координат), оскільки, по-перше, для цього не потрібно переводити географічні координати в прямокутну систему координат (виконувати проекційні перетворення), та, по-друге, багато проекцій, якщо невірно обрані, можуть привести до значних спотворень довжин в силу особливостей проекційних спотворень.

Існує три способи розрахунку сферичної відстані великого кола.

1. Сферична теорема косинусів.

У випадку маленьких відстаней та невеликої розрядності обчислень (кількість знаків після коми), використання формули (2.7) може призвести до значним помилок пов'язаних з округленням:

$$\Delta\delta = \arccos \{ \sin \phi_1 \sin \phi_2 + \cos \phi_1 \cos \phi_2 \cos \Delta\lambda \} \quad (2.7)$$

де $\phi_1, \lambda_1; \phi_2, \lambda_2$ — широта та довгота двох точок в радіанах;

$\Delta\lambda$ — різниця координат по довготі;

$\Delta\delta$ — кутова різниця.

Для переведення кутової відстані в метричну, потрібно кутову різницю помножити на радіус Землі (6372795 метрів), одиниці кінцевої відстані будуть рівні одиницям, в якій представлений радіус (в цьому випадку – метри).

2. Формула гаверсинусів використовується, щоб уникнути проблем з невеликими відстанями.

$$\Delta\sigma = 2 \arcsin \left\{ \sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos \phi_1 \cos \phi_2 \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right\} \quad (2.8)$$

де $\phi_1, \lambda_1; \phi_2, \lambda_2$ — широта та довгота двох точок в радіанах;

$\Delta\lambda$ — різниця координат по довготі;

$\Delta\delta$ — кутова різниця.

3. Модифікація для антиподів. Попередня формула також схильна до проблеми точок-антиподів, щоб її вирішити використовується наступна її модифікація.

$$\Delta\sigma = \arctan \left\{ \frac{\sqrt{[\cos \phi_2 \sin \Delta\lambda]^2 + [\cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2 \cos \Delta\lambda]^2}}{\sin \phi_1 \sin \phi_2 + \cos \phi_1 \cos \phi_2 \cos \Delta\lambda} \right\} \quad (2.9)$$

де $\phi_1, \lambda_1; \phi_2, \lambda_2$ — широта та довгота двох точок в радіанах;

$\Delta\lambda$ — різниця координат по довготі;

$\Delta\delta$ — кутова різниця.

Отже, для того, щоб виключити похибки для визначення невеликих відстаней між географічними точками надалі будемо використовувати формулу гаверсинусів з модифікацією для антиподів. При обчисленні треба враховувати градусний формат GPS- координат, який треба попередньо перевести в радіальний.

Розглянемо функцію визначення умови для сповіщення користувача про наближення автобусу до певного користувача та найближчої зупинки формула (2.10):

$$f(\text{dist}, b(x_1, y_1), u(x_2, y_2), st(x_3, y_3), \text{stop}) = \begin{cases} 1, & \text{якщо виконується умова відстаней,} \\ 0, & \text{інакше.} \end{cases} \quad (2.10)$$

де dist – стале значення відстані, число від 50 до 500 метрів;
 $b(x_1, y_1)$ – GPS- координати місця розташування автобусу;
 $u(x_2, y_2)$ – GPS- координати місця розташування користувача;
 $st(x_3, y_3)$ – GPS- координати місця розташування найближчої зупинки;
 stop – ознака того, що найближча зупинка вже була здійснена, по замовченню значення логічної змінної рівно false .

Умова відстаней між об'єктами представляє собою систему наступних нерівностей (2.11) :

$$fl = \begin{cases} d(b(x_1, y_1), u(x_2, y_2)) = \text{dist}, \\ d(b(x_1, y_1), st(x_3, y_3)) \leq 2 * \text{dist}, \\ \text{stop} = \text{false}. \end{cases} \quad (2.11)$$

де dist – обране значення відстані, число від 50 до 500 метрів;
 $d(b(x_1, y_1), u(x_2, y_2))$ – відстань між автобусом та користувачем;
 $d(b(x_1, y_1), st(x_3, y_3))$ – відстань між автобусом та зупинкою
 stop – ознака того, що зупинка ще не була пройдена.

Тобто, якщо найближча зупинка не була пройдена, відстань між автобусом та користувачем дорівнює певному значенню dist , а відстань між автобусом та зупинкою менше або дорівнює подвійній відстані dist , то система Інтернету речей сповістить користувача про наближення автобусу до користувача. Для розрахунку відстані між об'єктами необхідно використовувати формулу (2.9) гаверсинусів з модифікацією для антиподів.

Розглянемо методику використання даної функції в екосистемі Інтернету речей системи «Шкільний автобус».

1. За переданими параметрами $b(x_1, y_1)$ та $u(x_2, y_2)$ визначаємо найкоротшу відстань dist_{bu} між автобусом та користувачем, використовуючи формулу (2.9) гаверсинусів з модифікацією для антиподів.

2. За переданими параметрами $b(x_1, y_1)$ та $st(x_3, y_3)$ визначаємо найкоротшу відстань dist_{bst} між автобусом та найближчою зупинкою, використовуючи формулу (2.9) гаверсинусів з модифікацією для антиподів.

3. Якщо виконуються наступні умови ($distbu = dist$) та ($distbst \leq 2 * dist$) та ($stop=false$), то сповістити користувача про наближення автобусу.

4. За бажанням (додатковими настройками системи), можна сповістити користувача про те, що автобус пройшов найближчу до нього зупинку. Для цього повинні одночасно виконуватись наступні три умови ($distbu = dist$) та ($distbst \leq 2 * dist$) та ($stop=true$).

5. За бажанням (додатковими настройками системи), цей алгоритм можна виконати декілька разів, наприклад для $dist=500$, $dist=250$, $dist=125$.

Отже, для реалізації технології Інтернету речей в системі «Шкільний автобус» був розроблений метод визначення моменту оповіщення про наближення автобусу до певного користувача відносно найближчої зупинки.

2.3 Висновки до розділу 2

Для розробки моделі екосистеми Інтернету речей системи «Шкільний автобус» в даному розділі було проаналізовано бізнес– процес «Відстежити місце розташування шкільного автобусу», виявлені учасники та описані особливості системної моделі вказаної системи.

Система «Шкільний автобус» складається з активних учасників системи, а саме адміністратора, який разом з водієм формує маршрут автобусу, надалі будемо його називати «адміністратор», та пасивних учасників – «користувачів», до яких можна віднести вчителів, учнів та їх батьків.

За допомогою методології функціонального моделювання IDEF0 було спроектовано модель процесу відстеження місця розташування шкільного автобусу.

Декомпозиція основного вузла системи складається з чотирьох операцій: «Сформувати розклад руху автобусу», «Сформувати маршруту автобусу на гул–карті», «Визначити місце розташування автобусу на маршруті», «Відобразити автобус на гул–карті».

Саме на етапі визначення місця розташування автобусу доцільно застосувати технології Інтернету речей для сповіщення користувача про наближення до нього автобусу.

В даній системі будуть взаємодіяти два пристрої, а саме смартфони, водія та користувача системи, які мають GPS–пристрої та працюють на операційній системі Android.

Для оновлення даних, наприклад координат зупинок, бажаної дистанції для відстеження між автобусом та користувачем, буде розроблено вебдодаток, наприклад засобами PHP, HTML, CSS, JS. Цей додаток буде містити функціональні можливості для аналізу GPS–сигналу про місце розташування автобусу, сповіщення користувачів про рух автобусу відносно зупинок.

Для обміну повідомленнями між пристроями та додатком Інтернету речей доцільно скористатись мережевим протоколом MQTT.

Для забезпечення надійний обміном даними з високим рівнем безпеки між додатком Інтернету речей (APP) та GPS– пристроєм, необхідно використати хмарної платформи Azure. Центр Інтернету речей (IoT Hub) дозволить

серверній частині рішення, розміщеного в хмарі, віртуально підключатись до потрібного пристрою, отримані характеристики сигналу зберігати у відповідній базі даних MongoDB.

Розглянувши всі методи визначення координат характерних точок об'єктів нерухомості, наприклад в нашому випадку – будівлі школи, зупинок транспорту в селищах, а також рухомих об'єктів – автобусу та користувачів системи був обраний метод супутникових геодезичних вимірювань. Він дозволить визначати GPS– координати як сталих об'єктів, так і рухомих.

Для реалізації екосистеми Інтернету речей системи «Шкільний автобус» був розроблений метод для визначення моменту сповіщення користувача про наближення автобусу. Цей метод базується на порівнянні відстані між автобусом та користувачем, між автобусом та найближчою зупинкою, яка ще не була пройдена.

Оскільки в навігації найчастіше використовується для обчислення найкоротшої відстані між точками поверхні Землі рівняння для обчислення відстані на великому колі, то для виключення похибки для визначення невеликих відстаней між географічними точками будемо використовувати формулу гаверсинусів з модифікацією для антиподів. При обчисленні треба враховувати градусний формат GPS– координат, який треба попередньо перевести в радіальний.

Отже, в цьому розділі було спроектовано модель екосистеми Інтернету речей системи «Шкільний автобус», яка дозволить реалізувати процес відстеження місця розташування шкільного автобусу.

3 РОЗРОБКА ПРОТОТИПУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЕКОСИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ СИСТЕМИ «ШКІЛЬНИЙ АВТОБУС»

3.1 Моделювання програмного забезпечення

Для розробки прототипу програмного забезпечення для екосистеми Інтернету речей системи «Шкільний автобус» проведемо моделювання відповідної системи, використовуючи метод UML.

UML (англ. Unified Modeling Language — уніфікована мова моделювання) – мова графічного опису для об’єктного моделювання в області розробки програмного забезпечення, для моделювання бізнес–процесів, системного проектування та відображення організаційних структур. [29, с.14] Суть методу полягає в проектуванні різних за змістом графічних діаграм, які передають структуру, співвідношення та взаємодію всіх елементів системи.

Діаграма прецедентів або діаграма варіантів використання описує основні дії користувачів системи, вона представлена на рисунку 3.1.

На діаграмі відображені відношення, що існують між акторами і варіантами використання (прецедентами). Діаграма варіантів використання – це граф спеціального вигляду, який є графічною нотацією для представлення конкретних варіантів використання. Основна ідея UseCase–діаграми – надати узагальнений вид системи, що дає можливість замовнику, кінцевому користувачеві і розробнику спільно обговорювати функціональність і поведінку системи.

Аналіз діаграми варіантів використання показує, що користувачам системи, а саме пасивним учасникам (учням, їх батькам, вчителям) доступні лише операції перегляду даних чи запиту на розміщення автобусу. Змінювати параметри системи (розклад та маршруту автобусу) може лише адміністратор школи.

Оскільки головним призначенням системи є визначення поточного місця розташування автобусу, то цей варіант має більш розгорнуту структуру:

- варіант «визначити місце розташування автобусу» включає варіант «переглянути поточне місце розташування автобусу», як остаточний варіант демонстрації результату виконання пошуку автобусу;

- для розширення можливостей наочної демонстрації результатів варіанту «переглянути поточне місце розташування» використовується варіант «відобразити автобус на гул–карті»;

- для «визначення відстані між автобусом та користувачем» необхідно «визначити місце розташування автобуса» та «визначити місце розташування користувача»;

- варіант «сповістити про наближення автобусу» розширити варіант «визначення відстані між автобусом та користувачем» в момент досягнення цієї відстані певного значення.

Саме на цьому етапі системи доцільно використати можливості сучасної технології Інтернету речей.

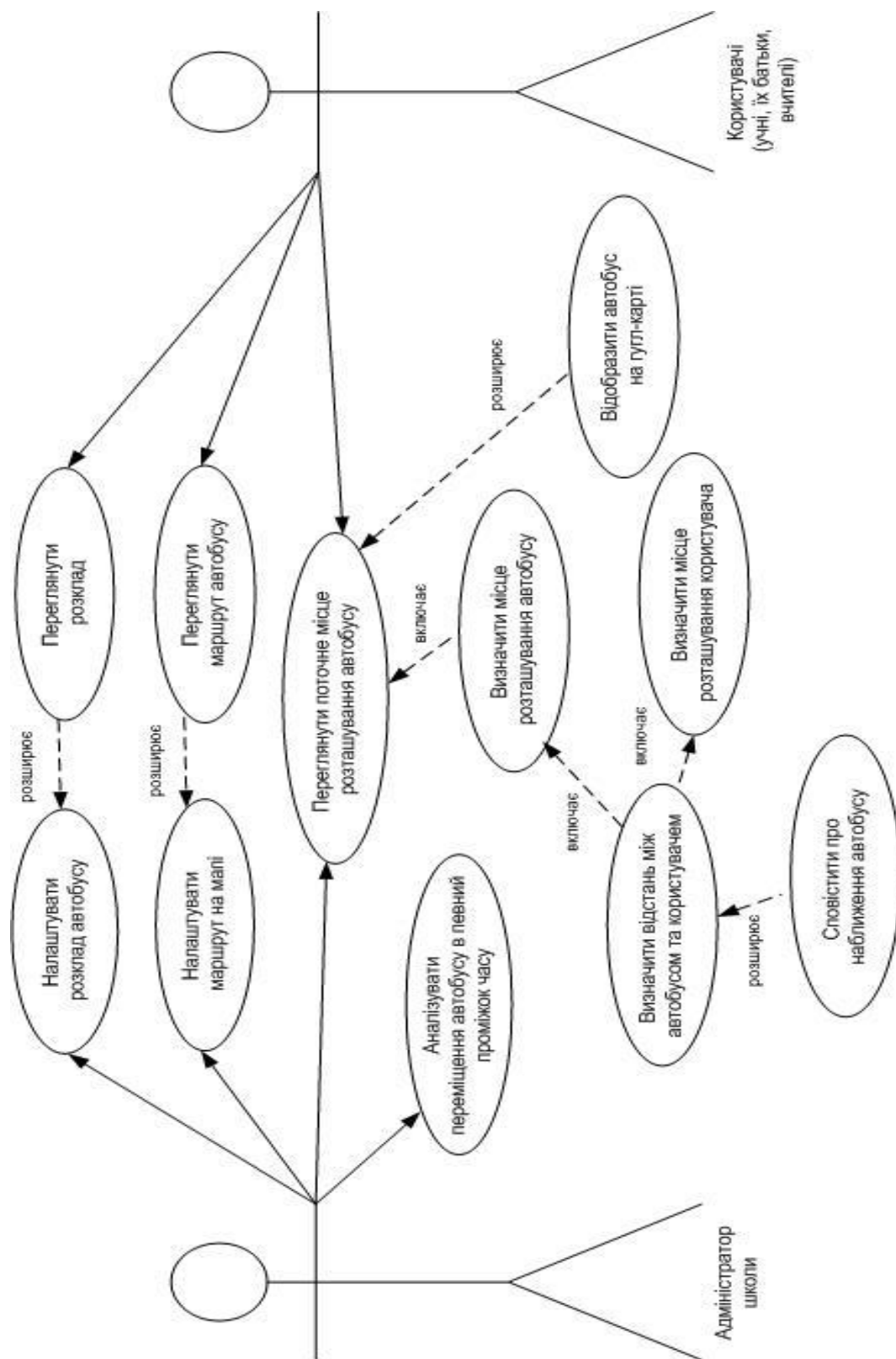


Рисунок 3.1 – Діаграма варіантів використання

Для реалізації технології Інтернету речей в системі «Шкільний автобус» необхідно проаналізувати взаємодію між пристроями (користувача, автобуса та Інтернету речей) за допомогою діаграми діяльності, що зображена на рисунку 3.2.

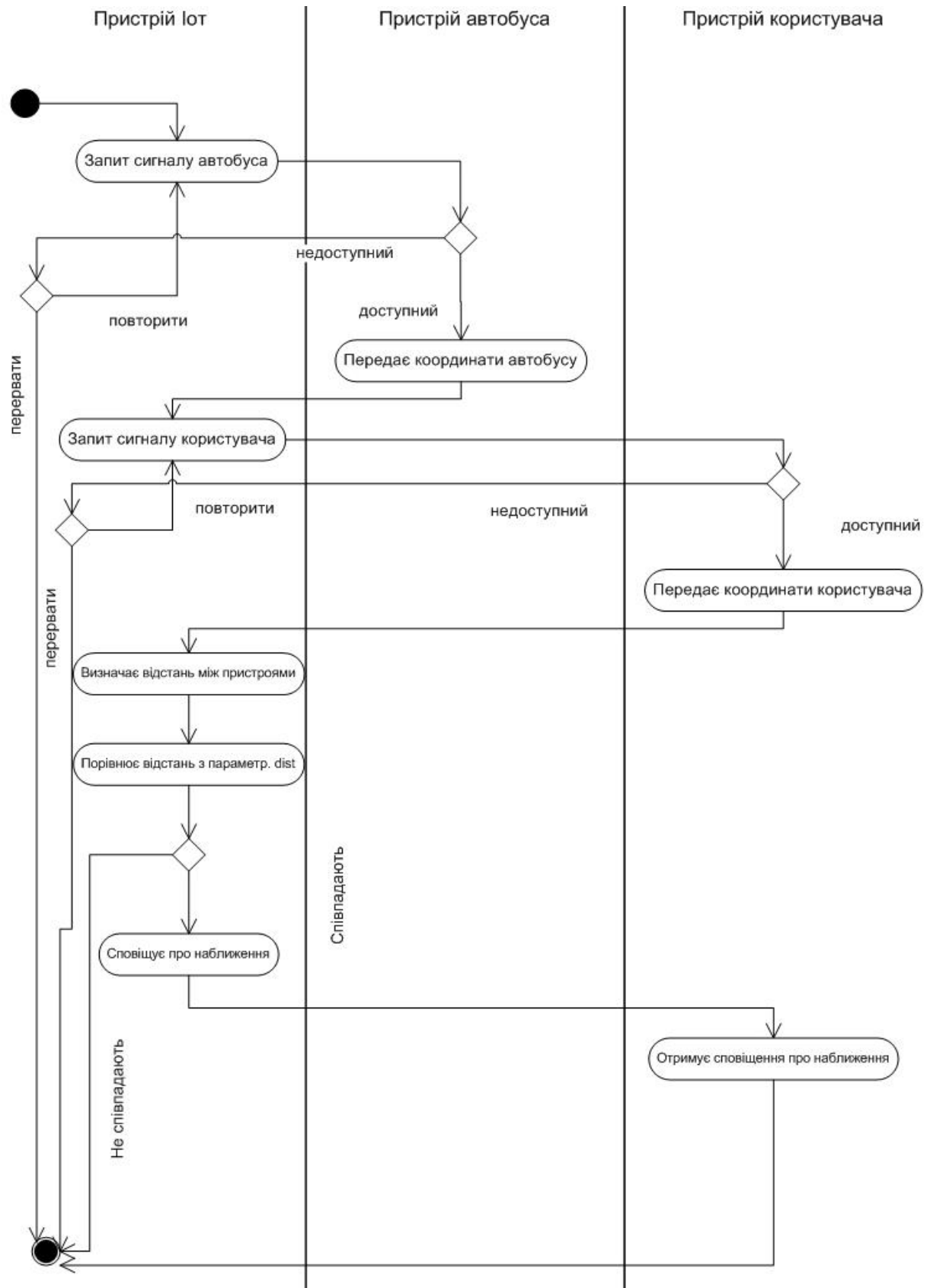


Рисунок 3.2 – Діаграма діяльності пристроїв

Передбачається, що пристрій Іот буде безперервно посилати запит до пристрою автобусу на отримання сигналу автобусу, і від цього залежать наступні кроки:

- якщо сигнал від пристрою автобусу є, то очікується передача географічних координат розміщення автобусу;
- якщо координати автобусу передано, то відбувається запит до пристрою користувача на сигнал і відповідно координат розміщення користувача в цей же час;
- якщо обидва сигнали отримані та передані координати і автобуса, користувача, то відбувається визначення відстані між автобусом і користувачем;
- якщо отримана відстань між автобусом та користувачем менше або дорівнює певному коефіцієнту (параметр, що можна змінювати), то на пристрій користувача відсилається певне повідомлення.

Порівнюючи швидкість руху автобусу (в середньому по селу від 20 до 20 кілометрів на годину) і не значну швидкість користувача (3–4 км/г), можна допустити, що відстань між користувачем та автобусом за секунду скорочуються або збільшується на 10 метрів.

Отже, при організації роботи пристроїв доцільно для своєчасної передачі сповіщень налаштувати передачу сигналів від пристроїв кожні 5 секунд.

Наступним кроком була розроблена діаграма класів системи «Шкільний автобус». Діаграма класів – це статична структурна діаграма, яка описує структуру системи, демонструє класи системи, їх атрибути, методи та залежності між класами. Діаграма класів системи «Шкільний автобус» зображена на рисунку 3.3. Аналіз діаграми класів показує:

- в системі «Шкільний автобус» налічується 11 класів, які поєднанні різними відношеннями, базовими класами є Bus, Reis, Station, All_user;
- від класу All_user унаслідують всі атрибути та методи класи Pupil, Parents, Teacher, Admin_school;
- клас Admin_school має власні методи для оновлення паролю входу в систему для налаштувань;
- класи Reis та Route_bus представлені у відношенні композиції, оскільки кожен рейс має свій власний маршрут зупинок;
- у відповідних відношеннях асоціації перебувають класи Bus та Reis, Station та All_user, Station та Route_bus;
- між класами Bus та Signal_bus, All_user та Signal_user існує відношення асоціації.

Розглянемо кожен клас окремо, а саме його призначення, властивості та методи.

Клас Bus призначений для збереження інформації про характеристики автобусу, що працює на рейсі. Має наступні властивості:

- id_bus, унікальний код для ідентифікації об'єкта, ціле число;
- driver, відомості про водія, текстовий тип;
- phone_dr, телефон водія з GPS пристроєм, довге ціле число;
- gps_bus, відомості про GPS пристрій автобусу, текстовий тип.

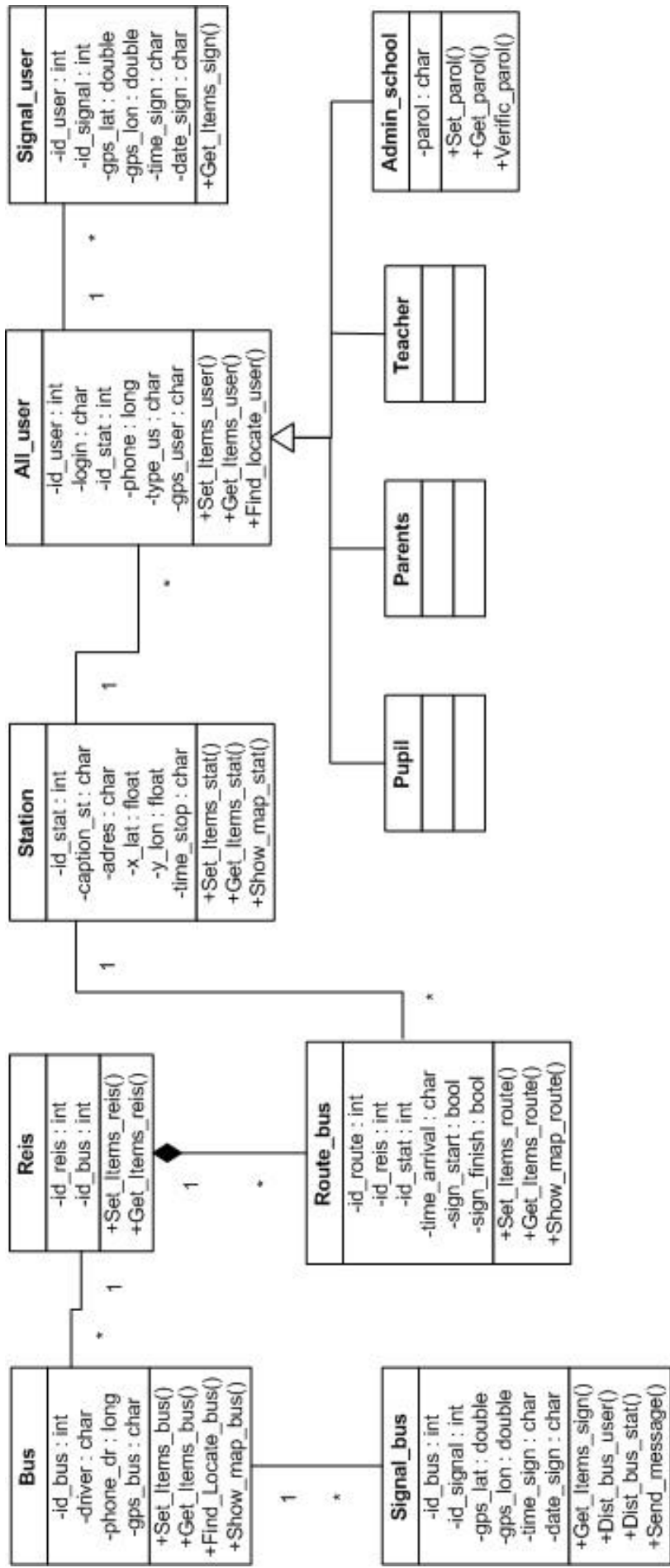


Рисунок 3.3 – Діаграма класів системи «Шкільний автобус»

Для класу Bus визначені наступні методи:

- Set_Items_bus – для оновлення даних про автобус;
- Get_Items_bus – для перегляду даних про автобус;
- Find_Locate_bus – для визначення місця розташування автобусу;
- Show_map_bus – для відображення автобусу на гул–карті.

Клас Signal_Bus призначений для збереження інформації про отриманий сигнал від автобусу. Він має наступні властивості:

- id_bus, код для ідентифікації автобусу, ціле число;
- id_signal, унікальний код для ідентифікації сигналу, ціле число;
- gps_lat, географічна координата широта, дійсне число;
- gps_lon, географічна координата довгота, дійсне число;
- time_sign, час отримання координат, символьним тип;
- date_sign, дата отримання координат, символьним тип.

Для класу Signal_Bus визначені наступні методи:

- Get_Items_sign – для перегляду отриманих параметрів;
- Dist_bus_user – для визначення відстані між користувачем та автобусом;
- Dist_bus_stat – для визначення відстані між автобусом та найближчою зупинкою;
- Send_message – для сповіщення користувача про наближення автобусу.

Клас Station призначений для збереження інформації про зупинки автобусу. Він має наступні властивості:

- id_stat, унікальний код для ідентифікації зупинки, ціле число;
- caption_st, назва зупинки, символьний тип;
- adres, адреса зупинки, символьний тип;
- x_lat, географічна координата широта зупинки, дійсне число;
- y_lon, географічна координата довгота зупинки, дійсне число;
- time_stop, час прибуття автобусу на зупинку, символьний тип.

Для класу Station визначені наступні методи: Set_Items_stat – для оновлення даних про зупинку, Get_Items_stat – для оновлення даних про зупинку, Show_map_stat – для відображення зупинки на гул–карті.

На основі інформації про зупинки створюється маршрут для певного рейсу. Клас Route_bus призначений саме для збереження маршруту автобусу. Він має наступні властивості:

- id_route, унікальний код для ідентифікації маршруту, ціле число;
- id_reis, код для ідентифікації рейсу, ціле число;
- id_stat, код для ідентифікації зупинки, ціле число;
- time_arrival, час прибуття на зупинку за маршрутом, символьний тип;
- sign_start, ознака, що зупинка є стартовою, логічний тип;
- sign_finish, ознака, що зупинка є кінцевою, логічний тип.

Для класу Route_bus визначені наступні методи: Set_Items_route – для оновлення даних про маршрут, Get_Items_route – для оновлення даних про маршрут, Show_map_route – для відображення маршруту на гул–карті.

Клас Reis призначений для збереження інформації про закріплений автобус до певного маршруту. Він має такі властивості, як id_reis (унікальний

код для ідентифікації рейсу, ціле число), `id_bus` (код для ідентифікації автобусу, ціле число). Для класу `Reis` визначені методи:– `Set_Items_reis` для оновлення параметрів рейсу, `Get_Items_reis` для перегляду параметрів рейсу.

Для збереження даних про користувачів системи був розроблений клас `All_user`, який містить наступні властивості:

- `id_user`, унікальний код для ідентифікації рейсу, ціле число;
- `login`, логін користувача в системі, символьний тип;
- `id_stat`, код для ідентифікації зупинки, ціле число;
- `phone`, телефон користувача, довге ціле число;
- `type_us`, тип користувача, символьний тип;
- `gps_user`, відомості про GPS пристрій користувача, текстовий тип.

Для цього класу визначені наступні методи: для оновлення даних про користувача – `Set_Items_user`, для перегляду даних про користувача – `Get_Items_user`, для визначення місця розташування автобусу – `Find_locate_user`.

Оскільки класи `Pupil`, `Parents`, `Teacher`, `Admin_school` унаслідують всі атрибути та методи від класу `All_user`, розглянемо унікальні властивості підкласу `Admin_school`, що призначений для збереження параметрів адміністратора школи, який єдиний має можливість оновлювати всі параметри системи. Клас `Admin_school` має єдину властивість `parol`, в якій зберігається пароль доступу до оновлення всіх параметрів системи, а методи `Set_parol`, `Get_parol`, `Verific_parol` призначені відповідно для оновлення, перегляду та перевірки паролю адміністратора.

Клас `Signal_user` створений для збереження даних про сигнал з `gps` пристрою користувача. Він має наступні властивості:

- `id_user`, код для ідентифікації користувача, ціле число;
- `id_signal`, унікальний код для ідентифікації сигналу, ціле число;
- `gps_lat`, географічна координата широта, дійсне число;
- `gps_lon`, географічна координата довгота, дійсне число;
- `time_sign`, час отримання координат, символьним тип;
- `date_sign`, дата отримання координат, символьним тип.

Для класу `Signal_user` визначений метод `Get_Items_sign` для перегляду отриманих параметрів.

3.2 Алгоритми роботи з системою «Шкільний автобус»

Аналіз процесу відстеження шкільного автобусу показав необхідність спроектувати два алгоритми роботи з системою «Шкільний автобус»: для адміністратора школи, як активного учасника системи, що має право створювати та оновлювати ключові об'єкти системи (розклад руху та маршрут автобусу) та користувачів (вчителів, учнів школи та їх батьків) для перегляду відомостей про ключові об'єкти системи та відстеження автобусу в певний час. Розглянемо алгоритм роботи адміністратора з системою, графічне зображення якого наведено на рисунку 3.4.



Рисунок 3.4 – Алгоритм роботи адміністратора з системою

Для обмеження доступу до режиму адміністратора інших користувачів вхід в цей режим повинен бути організований через пароль, який можна змінити при налаштуванні параметрів адміністратора.

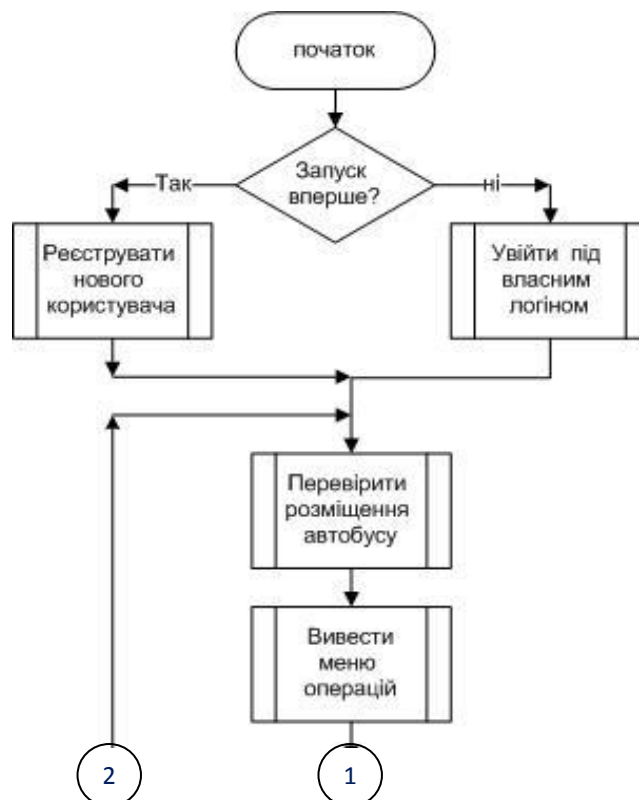
Адміністратору будуть доступні наступні функціональні можливості:

- налаштувати розклад автобусу, тобто створити чи оновити розклад автобусу, з можливістю змінювати маршрут, часу зупинок;
- налаштувати маршрут на мапі, тобто створити чи оновити маршрут на гул-карті згідно оновленого розкладу автобусу;
- переглянути поточне місце розташування автобусу, тобто в режимі реального часу визначити місце розташування автобусу;
- аналізувати переміщення автобусу в певний проміжок часу, тобто мати змогу будувати шлях переміщення автобусу за певний день, за певну годину.

Розглянемо алгоритм роботи користувачів з системою, графічне зображення якого наведено на рисунку 3.5. Користувачам (вчителям, учням школи та їх батькам) будуть доступні наступні функціональні можливості:

- при запуску системи автоматично перевіряти поточне місце розташування автобусу;
- переглянути розклад автобусу, тобто список зупинок з часом прибуття автобусу;
- переглянути маршрут на мапі з підписом зупинок;
- переглянути поточне місце розташування автобусу, тобто в режимі реального часу визначити місце розташування автобусу.

При першому запуску системи користувачі зможуть налаштувати власні параметри: тип користувача та зупинку автобуса, відстеження якої їх цікавить. При потребі вони можуть змінити вказані параметри.



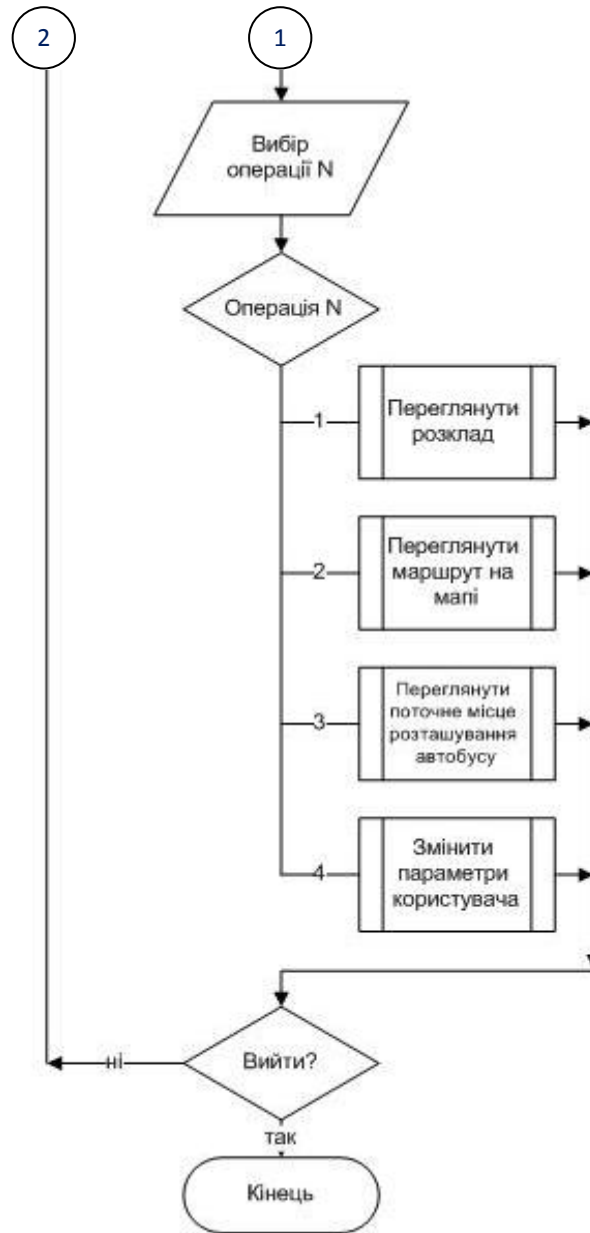


Рисунок 3.5 – Алгоритм роботи користувачів з системою

Для реалізації технології Інтернет речей в процесі відстеження шкільного автобусу був розроблений алгоритм для програмного додатку для сповіщення користувача про наближення автобусу, графічне зображення якого наведено на рисунку 3.6.



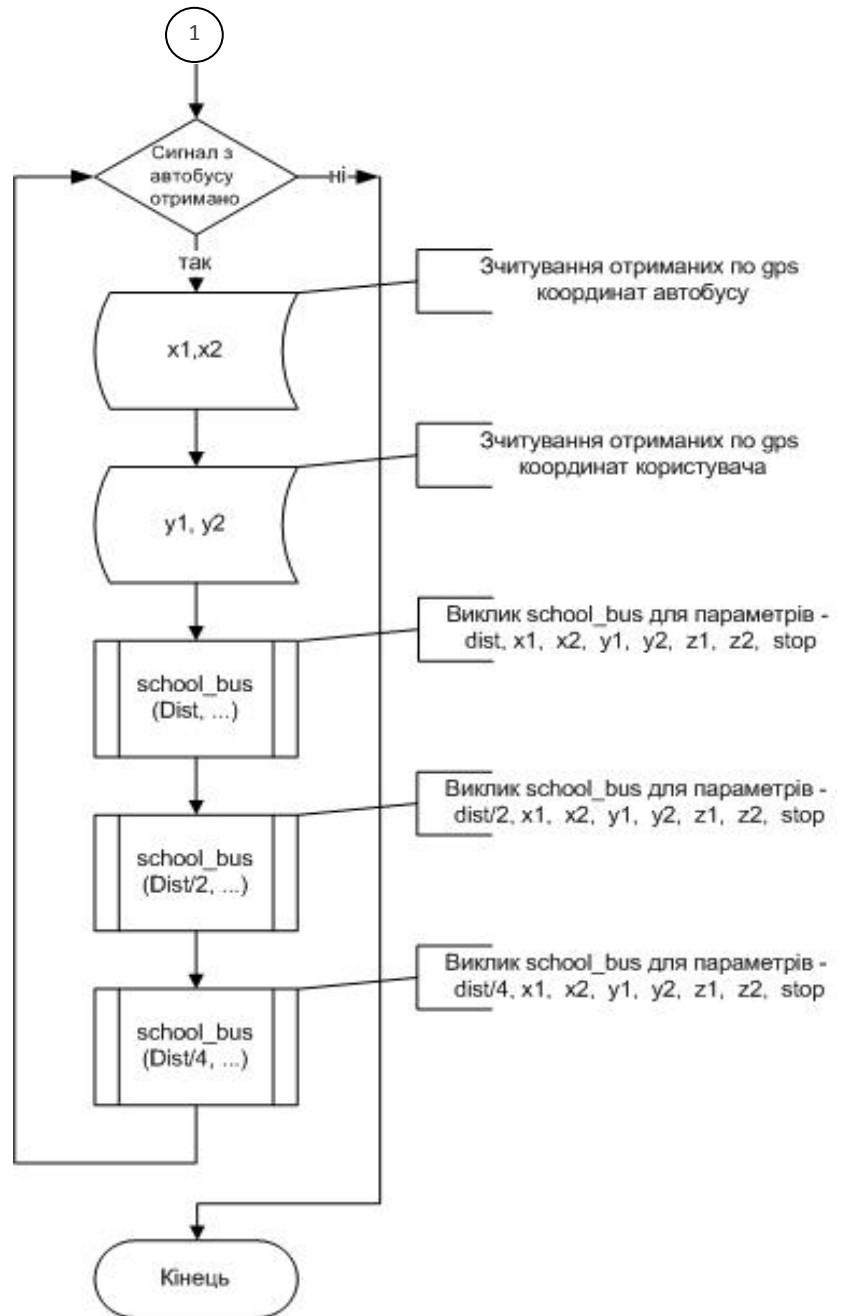


Рисунок 3.6 – Головний алгоритм роботи екосистеми Інтернету речей

Головний алгоритм роботи екосистеми Інтернет речей системи «Шкільний автобус» є компактним за рахунок використання допоміжного алгоритму `school_bus` для сповіщення користувача про наближення автобусу до певного користувача та найближчої зупинки, графічне зображення якого знаходиться на рисунку 3.7.

В наведеному алгоритмі використовується метод, який був розроблений в пункті 2.2 і базується на порівнянні відстаней між автобусом та користувачем, автобусом та найближчою зупинкою

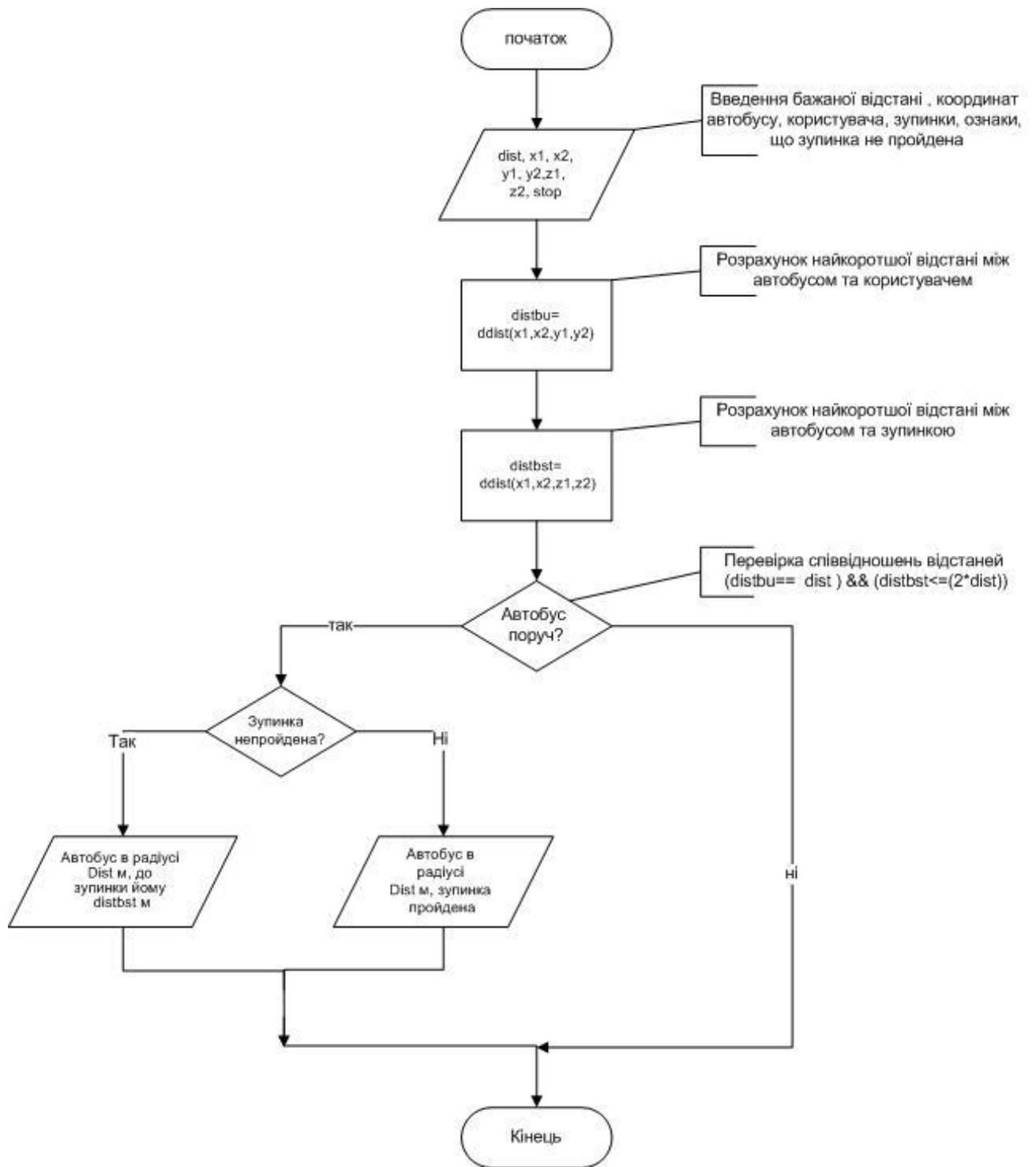


Рисунок 3.7 – Алгоритм сповіщення користувача про наближення автобусу до певного користувача та найближчої зупинки

Для розрахунку відстані між отриманими від GPS-пристроїв географічними координатами двох об'єктів використовується наступний алгоритм, що зображений на рисунку 3.8.

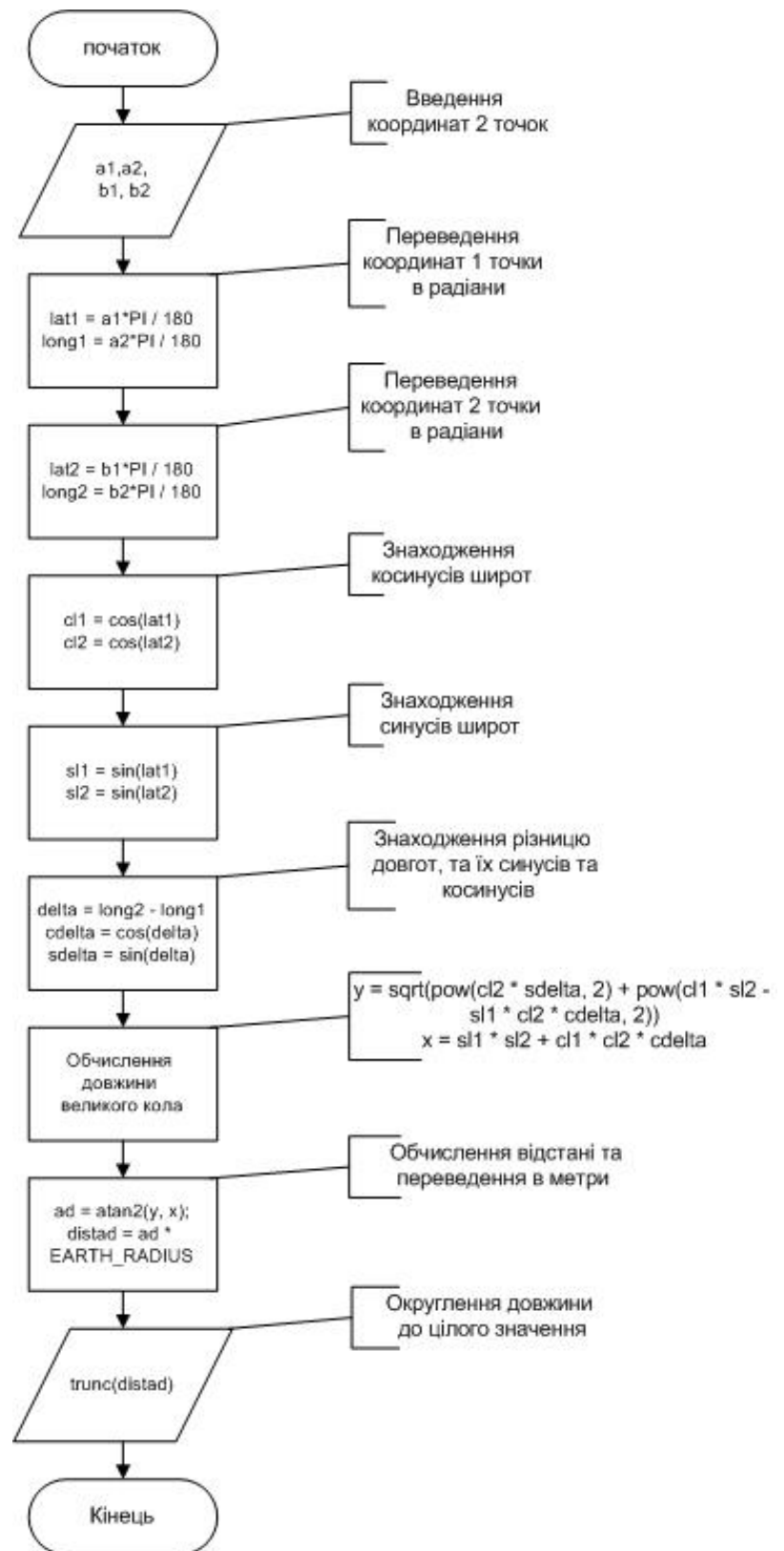


Рисунок 3.8 – Алгоритм розрахунку відстані між двома об'єктами

Отже, розроблені алгоритми демонструють як і особливості роботи користувачів з системою «Шкільний автобус», так і послідовність дій екосистеми Інтернету речей системи для оповіщення користувача про наближення автобусу до нього та найближчої зупинки.

3.3 Розробка прототипу програмного додатку Інтернету речей для сповіщення користувачів про наближення автобусу

Для реалізація технології IoT в системи «Шкільний автобус» використаємо сучасну хмарну платформу Microsoft Azure, яка має всі необхідні інструментальні засоби.

Після реєстрації на відповідному порталі Microsoft Azure необхідно створити новий центр IoT за допомогою сторінки «Центр IoT», яка зображена на рисунку 3.9

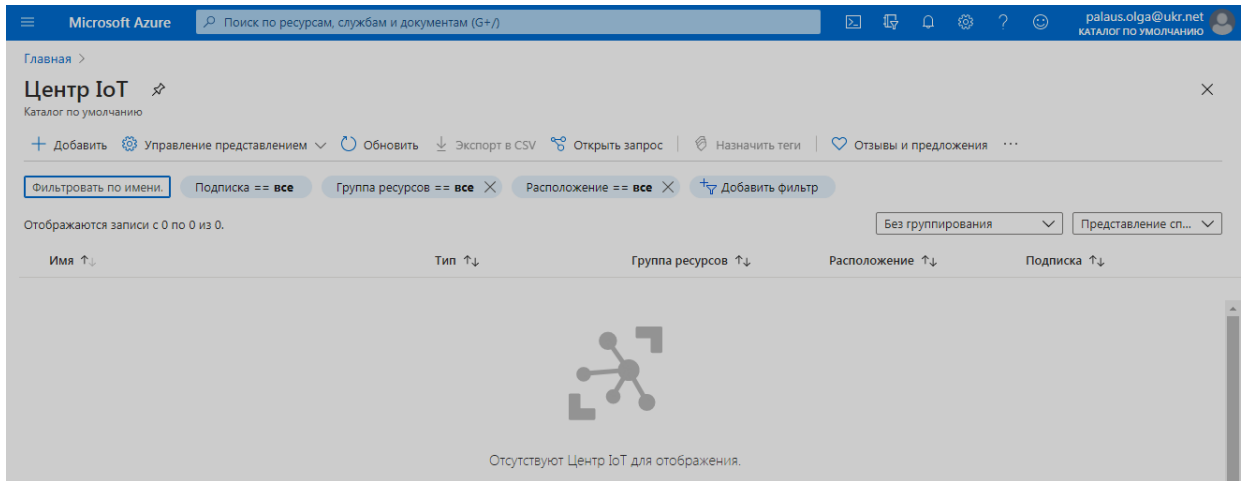


Рисунок 3.9 – Сторінка для створення нового центру IoT

При створенні (рисунок 3.10) задається назва для групи ресурсів, наприклад `groupe_bus`, та назва самого центру Інтернету речей, в нашому випадку – `busiot`.

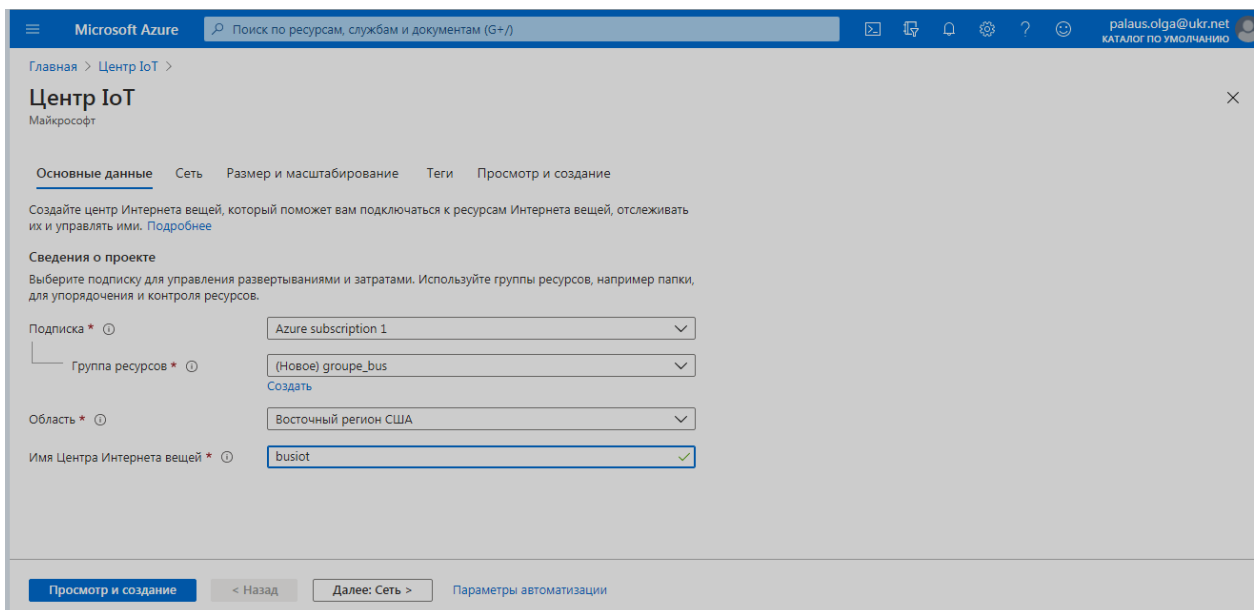


Рисунок 3.10 – Сторінка для задання параметрів нового центру IoT

Після успішного створення центру IoT можна переглянути чи змінити його параметри за допомогою сторінки, що зображено на рисунку 3.11.

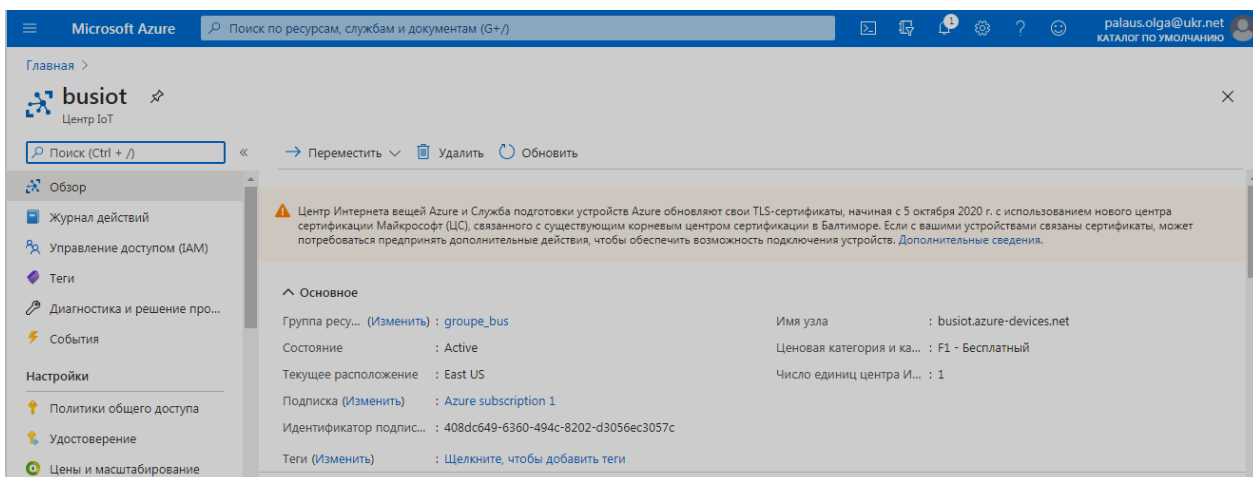


Рисунок 3.11 – Сторінка для оновлення параметрів центру IoT

Наступним етапом необхідно створити та підключити нові пристрої до центру IoT busiot. Це можна зробити вибравши відповідний пункт списку в лівій частині вікна (рисунок 3.12).

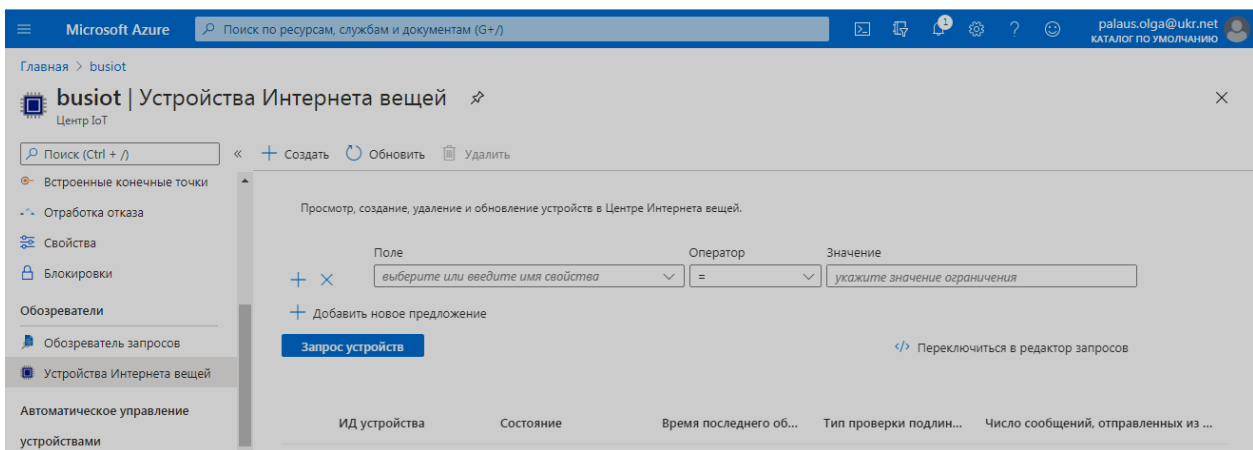


Рисунок 3.12 – Сторінка для створення нового пристрою для центру IoT

На даній сторінці за допомогою кнопки «Створити» можна задати параметри новому пристрою, це зображено на рисунку 3.13.

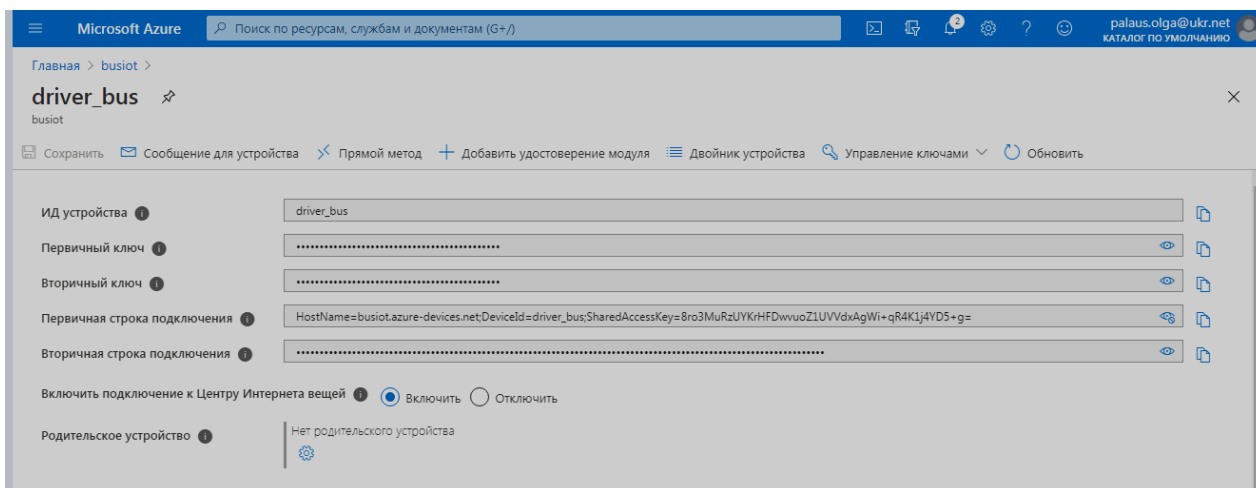


Рисунок 3.13 – Сторінка для задання параметрів нового пристрою

Аналогічним способом створені пристрої для автобусу, учня, його батьків та вчителя (рисунок 3.14).

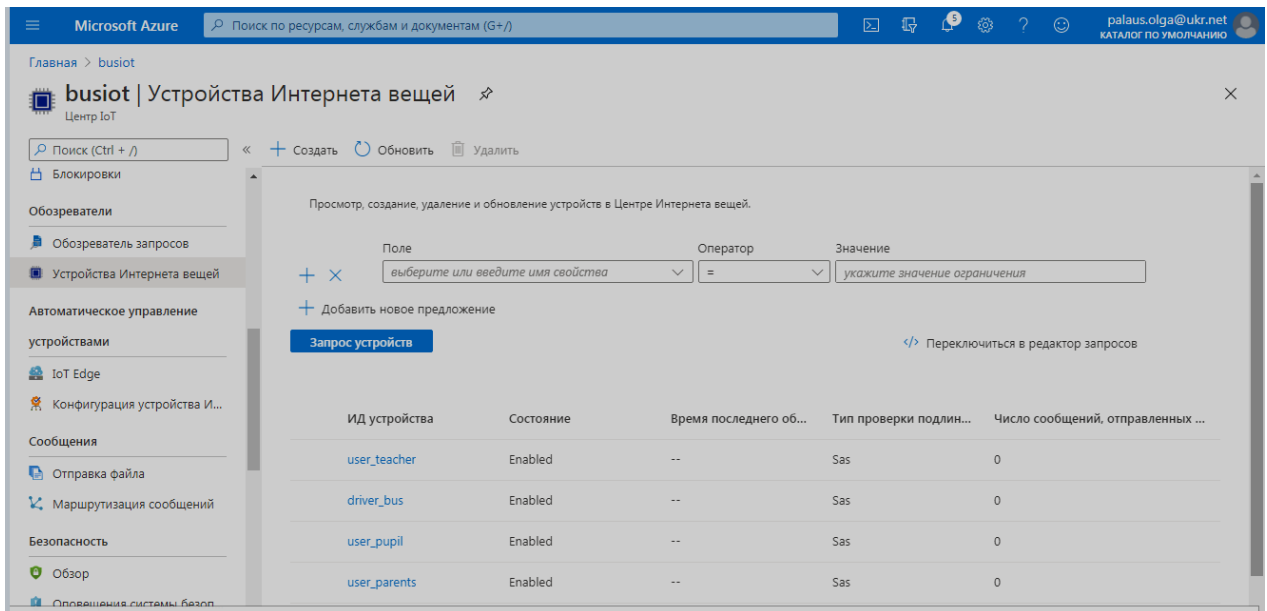


Рисунок 3.14 – Сторінка для усіма пристроями центру IoT

Для реалізації технології Інтернет речей в процесі відстеження шкільного автобусу був розроблений алгоритм для сповіщення користувача про наближення автобусу. Розглянемо його програмну реалізацію.

Оскільки платформа Microsoft Azure підтримує цілий пакет мов програмування по версії середовища виконання (C#, JavaScript, F#, Java, Python), то для розробки програмного додатку для IoT, був обраний C#.

В головній функції додатку для IoT, поки приймається сигнал від пристрою автобусу відбувається наступний блок команд (рисунок 3.15).

```
// перевірка чи пройдена зупинка
distbst = ddist(x1, x2, z1, z2);
if ((distbst <= 20) && (stop==false))
{
    stop = true;
    Console.WriteLine(" Увага! Зупинка пройдена, вона від автобусу в радіусі = {0} метрів ", distbst);
    Console.WriteLine(" ");
}
// перевірка дистанції для сповіщення
school_bus(dist, x1, x2, y1, y2, z1, z2, stop);
school_bus(dist / 2, x1, x2, y1, y2, z1, z2, stop);
school_bus(dist / 4, x1, x2, y1, y2, z1, z2, stop);
```

Рисунок 3.15 – Програмний код аналізу знаходження автобусу відносно найближчої зупинки та користувача

Після отримання від відповідних пристроїв координат місця знаходження автобусу та користувача відбувається визначення відстані між автобусом та найближчою зупинкою. Якщо зупинка в радіусі 20 метрів, то відбувається оновлення ознаки наближення до зупинки stop та відповідне

повідомлення на пристрій користувача. Далі відбувається виклик функції `school_bus` для трьох наборів параметрів, в яких відмінність лише у значенні бажаній відстані між автобусу та користувачем. Програмний код цієї функції наведено на рисунку 3.16.

```
// перевірка чи треба повідомляти про наближення автобусу
3 references
public static void school_bus(int dist, double x1, double x2, double y1, double y2, double z1, double z2, bool stop)
{
    int distbu, distbst;
    distbu = ddist(x1, x2, y1, y2);
    distbst = ddist(x1, x2, z1, z2);
    if (((distbu >= (dist - 5)) && (distbu <= (dist + 5)) && (distbst <= (2 * dist))) && (stop == false))
    {
        Console.WriteLine(" Увага! Автобус від Вас в радіусі = {0} метрів ", dist);
        Console.WriteLine(" при цьому автобус від зупинки в радіусі = {0} метрів ", distbst);
        Console.WriteLine(" ");
    }
    if (((((distbu >= (dist - 5)) && (distbu <= (dist + 5))) && (distbst <= (2 * dist))) && (stop != false))
    {
        Console.WriteLine(" Увага! Автобус від Вас в радіусі = {0} метрів ", dist );
        Console.WriteLine(" при цьому автобус вже пройшов вашу зупинку!");
        Console.WriteLine(" ");
    }
}
}
```

Рисунок 3.16 – Програмний код аналізу необхідності сповіщення користувача про наближення автобусу

В своїй роботі функція `school_bus` базується на порівнянні відстані між різними об'єктами – користувачем та автобусом, автобусом та зупинкою, для обчислення яких викликається функція `ddist`. Програмний код цієї функції наведено на рисунку 3.17.

```
4 references
public static int ddist(double a1, double a2, double b1, double b2)
// знаходження відстані між двома точками
// A1, A2 - широта, довгота першої точки , B1, B2 - широта, довгота другої точки
{
    double lat1, lat2, long1, long2;
    long EARTH_RADIUS = 6372795;
    // перевести координат в радіани
    lat1 = a1 * Math.PI / 180;
    lat2 = b1 * Math.PI / 180;
    long1 = a2 * Math.PI / 180;
    long2 = b2 * Math.PI / 180;
    double c11, c12, s11, s12, delta, cdelta, sdelta, y, x, ad, distad;
    // косінуси та синуси широт та різниці довгот
    c11 = Math.Cos(lat1);
    c12 = Math.Cos(lat2);
    s11 = Math.Sin(lat1);
    s12 = Math.Sin(lat2);
    delta = long2 - long1;
    cdelta = Math.Cos(Math.Abs(delta));
    sdelta = Math.Sin(Math.Abs(delta));
    // обчислюємо довжину великого кола
    y = Math.Sqrt(Math.Pow(c12 * sdelta, 2) + Math.Pow(c11 * s12 - s11 * c12 * cdelta, 2));
    x = s11 * s12 + c11 * c12 * cdelta;
    // обчислюємо відстань та переводимо в метри
    ad = Math.Atan2(y, x);
    distad = ad * EARTH_RADIUS;
    int ss;
    ss = (int)Math.Floor(distad);
    return ss;
}
```

Рисунок 3.17 – Програмний код для визначення відстані між 2 точками

Функція `ddist` згідно формули гаверсинусів визначає найкоротшу відстань між двома точками, заданих своїми географічними координатами.

Протестуємо розроблене програмне забезпечення для наступних ситуацій:

- автобус їде по маршруту до певної зупинки, користувач – учень повільно рухається до зупинки;

- автобус їде по маршруту до певної зупинки, користувачі (батько чи мати) знаходиться статично вдома;

- автобус їде по маршруту до певної зупинки, користувач (учень) чекає його на зупинці.

На рисунку 3.18 наведена система сповіщень, яку отримує користувач, який рухається на зустріч автобусу.

```

Увага! Автобус від Вас в радіусі = 500 метрів
при цьому автобус від зупинки в радіусі = 354 метрів

Увага! Автобус від Вас в радіусі = 250 метрів
при цьому автобус від зупинки в радіусі = 126 метрів

Увага! Зупинка пройдена, вона від автобусу в радіусі = 20 метрів

Увага! Автобус від Вас в радіусі = 125 метрів
при цьому автобус вже пройшов вашу зупинку!

Увага! Автобус від Вас в радіусі = 125 метрів
при цьому автобус вже пройшов вашу зупинку!

Увага! Автобус від Вас в радіусі = 125 метрів
при цьому автобус вже пройшов вашу зупинку!

Увага! Автобус від Вас в радіусі = 250 метрів
при цьому автобус вже пройшов вашу зупинку!

Увага! Автобус від Вас в радіусі = 500 метрів
при цьому автобус вже пройшов вашу зупинку!

```

Рисунок 3.18 – Сповіщення користувача системи, який рухається повільно

Аналізуючи систему сповіщень, можна зробити висновки, що автобус рухається значно швидше ніж користувач. Порівнюючи швидкість руху автобусу (в середньому по селу від 20 до 20 кілометрів на годину) і не значну швидкість користувача (3–4 км/г), можна допустити, що відстань між користувачем та автобусом за секунду скорочуються або збільшується на 10 метрів.

Після того, як були отримані повідомлення про наближення автобусу на відстань 500 м, 250 м до користувача, автобус прибув на зупинку раніше користувача, про що йому теж було повідомлено. Після цього автобус почав віддалятися від зупинки та від користувача, про що свідчать останні повідомлення.

Розглянемо систему сповіщень користувача, який знаходиться вдома і майже не рухається (наприклад, батьки учня, який вже пішов на зупинку). На рисунку 3.19 наведена система сповіщень в цій ситуації.

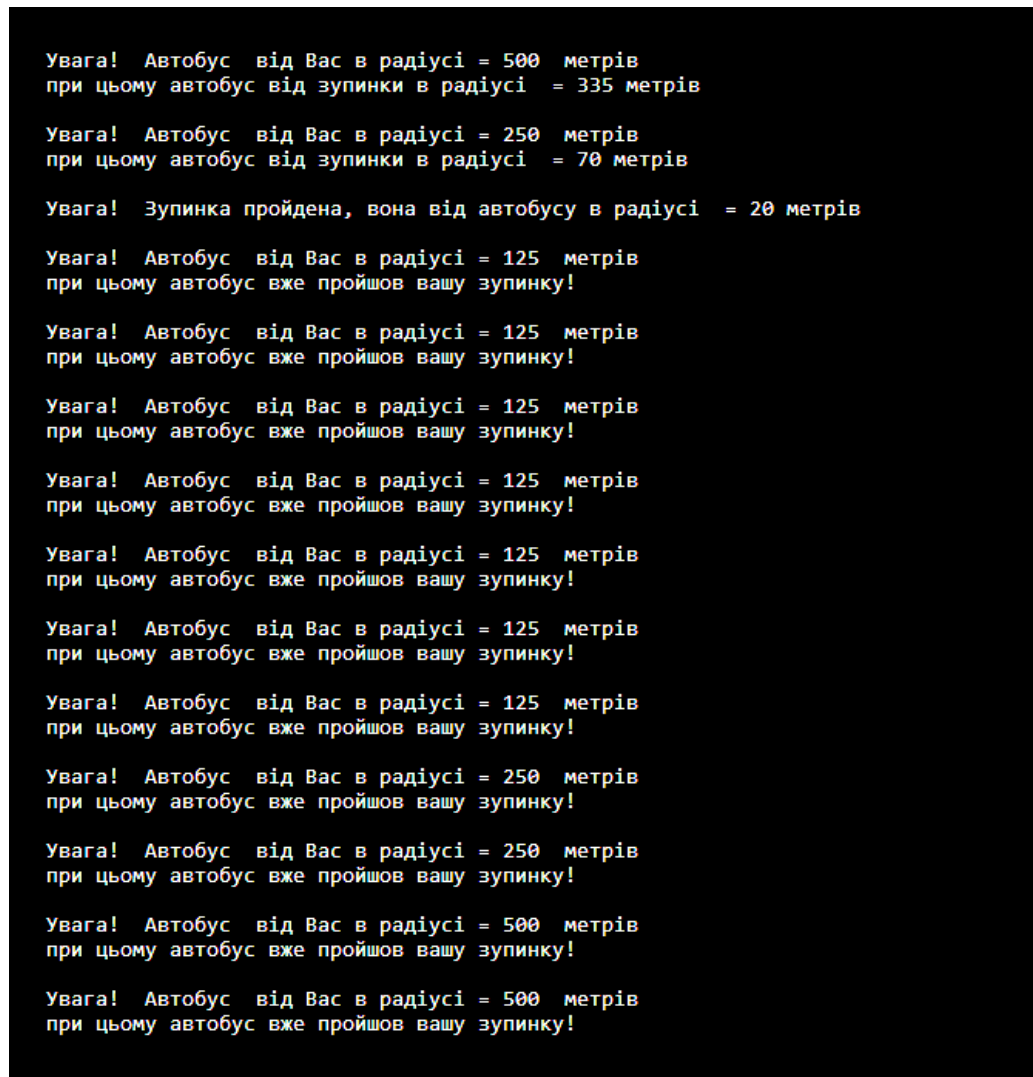


Рисунок 3.19 – Сповіщення користувача системи, який не рухається

Аналізуючи систему сповіщень в другому випадку, можна зробити висновки, що ці повідомлення будуть корисні батькам для контролю своєї дитини, її пересування та місця знаходження.

Розглянемо ситуації коли автобус рухається за маршрутом, а користувач вже чекає його на зупинці (рисунок 3.20).

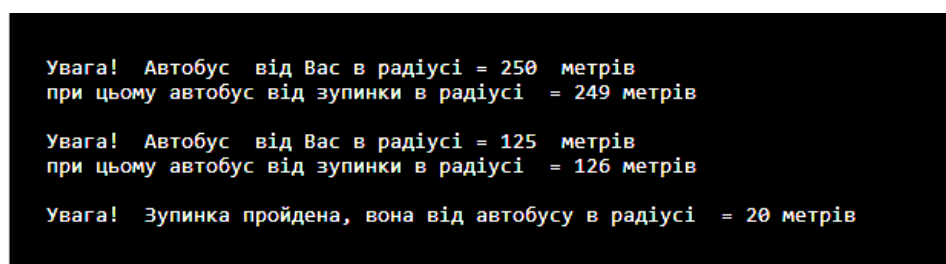


Рисунок 3.20 – Сповіщення користувача, який знаходиться на зупинці

Як бачимо через певні технічні проблеми (пропав сигнал автобусу чи користувача), що можна пояснити як особливостями місцевості (перепади висот над рівнем моря), користувач, який очікує автобус на зупинці отримує лише 2 повідомлення про наближення автобуса, інформацію про прибуття автобусу на зупинку. Подальші повідомлення про рух автобусу відсутні через те, що користувач сів у автобус та рушив разом з ним, тобто відстань між ними дуже мала та не змінюється.

Отже, розроблений прототип програмного забезпечення Інтернету речей для системи «Шкільний автобус» доводить доцільність впровадження технологій Інтернету речей та хмарних обчислень в систему відстеження шкільного транспорту з метою підвищення безпеки перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості.

3.4 Висновки до розділу 3

Для розробки прототипу програмного забезпечення для екосистеми Інтернету речей системи «Шкільний автобус» було проведено моделювання відповідної системи, використовуючи метод UML та були спроектовані:

- діаграма варіантів використання для демонстрації шляхів використання системи різними користувачами;
- діаграма діяльності для демонстрації взаємодії між пристроями в екосистемі Інтернету речей в системі «Шкільний автобус»;
- діаграма класів для демонстрації об'єктів системи, їх атрибутів, методів та залежностей між класами.

Наступним етапом були розроблені алгоритми:

- для роботи адміністратора з системою «Шкільний автобус» для оновлення всіх необхідних параметрів;
- для роботи користувачів з системою «Шкільний автобус» для відстеження автобусу;
- алгоритм роботи екосистеми Інтернет речей системи «Шкільний автобус»;
- для розрахунку відстані між отриманими від GPS–пристроїв географічними координатами двох об'єктів;
- для сповіщення користувача про наближення автобусу до нього та найближчої зупинки.

Для реалізація технології IoT в системі «Шкільний автобус» була використана сучасна хмарна платформу Microsoft Azure, яка має всі необхідні інструментальні засоби та великий пакет мов програмування по версії середовища виконання. Використовуючи мову програмування C# та алгоритми розроблені в пункті 3.2, був створений програмний додаток Інтернету речей для сповіщення користувачів про наближення автобусу.

Розроблений прототип програмного забезпечення екосистеми Інтернету речей для системи «Шкільний автобус» був протестований і для ситуації коли автобус їде по маршруту до певної зупинки, користувач – учень

рухається до зупинки, і коли автобус їде по маршруту до певної зупинки, користувачі (батько чи мати) знаходиться статично вдома. Відповідні координати руху автобусу та користувача були збереженні відповідних текстових файлах, формату який підтримує хмарна платформа Microsoft Azure.

Тестування прототипу програмного забезпечення Інтернету речей для системи «Шкільний автобус» доводить доцільність впровадження технологій Інтернету речей та хмарних обчислень в систему відстеження шкільного транспорту з метою підвищення безпеки перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості.

ВИСНОВКИ

Державна програма «Шкільний автобус» була розроблена для забезпечення виконання Закону України «Про Освіту», а саме статті 13, яка говорить про те, що особи, які здобувають повну загальну середню освіту, проживають у сільській місцевості і потребують підвезення до закладу освіти і у зворотному напрямку, забезпечуються таким підвезенням за кошти місцевих бюджетів, у тому числі із забезпеченням доступності відповідного транспорту для осіб з порушенням зору, слуху, опорно-рухового апарату та інших маломобільних груп населення.

Метою роботи є підвищення ефективності системи "Шкільний автобус" для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості, за рахунок розробки моделі системи відстеження шкільного транспорту на основі технологій Інтернету речей та хмарних обчислень

Саме з Інтернетом речей більшість науковців та винахідників пов'язують четверту промислову революцію (Індустрія 4.0). Промисловий Інтернет речей – це один з найбільш великих сегментів Інтернету речей з точки зору кількості підключених пристроїв і ступеня корисності цих сервісів для виробництва і автоматизації підприємств.

Проте є певна категорія напрямків, як наприклад, освіта, освітній процес, де технологія Інтернету речей ще недостатньо розкрита та впроваджена.

Аналіз предметної області показав відсутність проектних рішень застосування технології Інтернет речей та хмарних обчислень для відстеження шкільних автобусів, тому розробка екосистеми Інтернет речей системи «Шкільний автобус» є актуальною та доцільною.

Для реалізації екосистеми Інтернету речей системи «Шкільний автобус» в даному розділі було розроблено:

- модель процесу відстеження місця розташування шкільного автобусу;
- модель екосистеми Інтернету речей системи «Шкільний автобус»;
- метод для визначення моменту сповіщення користувача про наближення автобусу. Цей метод базується на порівнянні відстані між автобусом та користувачем, між автобусом та найближчою зупинкою, яка ще не була пройдена.

Для розробки прототипу програмного забезпечення для екосистеми Інтернету речей системи «Шкільний автобус» було проведено моделювання відповідної системи, в результаті чого:

- спроектовані діаграма варіантів використання для демонстрації шляхів використання системи різними користувачами, діаграма діяльності для демонстрації взаємодії між пристроями в екосистемі Інтернету речей в системі «Шкільний автобус», діаграма класів для демонстрації об'єктів системи, їх атрибутів, методів та залежностей між класами.

- розроблені алгоритми для роботи адміністратора з системою «Шкільний автобус» для оновлення всіх необхідних параметрів, для роботи

користувачів з системою «Шкільний автобус» для відстеження автобусу, алгоритм роботи екосистеми Інтернет речей системи «Шкільний автобус», для розрахунку відстані між отриманими від GPS–пристроїв географічними координатами двох об'єктів, для сповіщення користувача про наближення автобусу до нього та найближчої зупинки.

Для реалізація технології IoT в системі «Шкільний автобус» була використана сучасна хмарна платформу Microsoft Azure та мова програмування C#.

Працездатність розробленого прототипу програмного забезпечення була перевірена на 3–х варіантах вихідних даних.

Отже, впровадження технології Інтернету речей в систему відстеження транспорту для перевезення осіб, які здобувають повну загальну середню освіту та проживають у сільській місцевості є доцільним та необхідним для підвищення ефективності безпеки життєдіяльності учасників освітнього процесу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Цифрові карти в логістиці. [Електронний ресурс] // acris.com.ua. – 2017. – Режим доступу до ресурсу: <http://acris.com.ua/ua/stati/tsifrovye-karty-v-logistike/>
2. Что такое GPS–мониторинг и зачем нужен? [Електронний ресурс] // 130.com.ua. – 2018. – Режим доступу до ресурсу: <https://130.com.ua/chto-takoe-gps-monitoring/>
3. Кислий В.М., Біловодська О.А., Олефіренко О.М., Смоляник О.М. Логістика: Теорія та практика: Навч.посіб. – К: Центр учбової літератури, 2010. – 360 с.
4. Крикавський Є.В. Логістика. Основи теорії: Підручник. – Львів: Львівська політехніка, 2004. – 416 с.
5. Тридід О. М., Таньков К. М., Логістичний менеджмент: Навчальний посібник. – Інжек, 2005. – 224 с.
6. Закон України «Про Освіту»
7. Постанова Кабінета міністрів УКРАЇНИ «Про затвердження Державної цільової соціальної програми "Шкільний автобус"» № 31 від 16 січня 2003 р., Київ
8. Збірник нормативно–правових актів щодо матеріально–технічного забезпечення галузі освіти. Шкільний автобус / Укл. Молибог В.В. – Вид. 2.– К., 2019.–53 с.
9. Интернет речей: концепція IoT. Що чекати від майбутнього? [Електронний ресурс] // futurum.today – 2019. – Режим доступу: <https://futurum.today/internet-rechei-kontseptsiiia-iot-shcho-chekaty-vid-maibutnoho/>
10. Промышленный Интернет Вещей. [Електронний ресурс] // www.it.ua – 2019.– Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/promyshlennyj-internet-veschej>
11. Интернет речей (IoT) . – [Електронний ресурс] // deps.ua– 2019. – Режим доступу: <https://deps.ua/ua/katalog/iot.html>
12. Системи та програми для логістики керування транспортом. [Електронний ресурс] // msb.aval.ua. – 2019. – Режим доступу до ресурсу: https://msb.aval.ua/business_it/logistic/
13. Logist.UA – TMS Система управління транспортом. [Електронний ресурс] // systemgroup.com.ua. – 2019. – Режим доступу до ресурсу: <https://systemgroup.com.ua/ru/transportnaya-i-pochtovaya-logistika/logistua-tms-sistema-upravleniya-transportom#GPS-monitoring>
14. EasyWay [Електронний ресурс] // www.eway.in.ua – 2019. – Режим доступу до ресурсу: <https://www.eway.in.ua/ua/cities/>
15. Обзор приложений для общественного транспорта. [Електронний ресурс] // hi-tech.ua. – 2017. – Режим доступу до ресурсу: <https://hi-tech.ua/article/poehali-obzor-prilozheniy-dlya-obshhestvennogo-transporta/>

16. Система відстеження транспорту Львова. [Електронний ресурс] // lad.lviv.ua. – 2015. – Режим доступу до ресурсу: <https://lad.lviv.ua>
17. Громова Т. Власть «Индустрии 4.0». [Електронний ресурс] // fastsalftimes.com – 2016. – Режим доступу: <http://fastsalftimes.com/sections/obzor/522.html.1>.
18. Комиссаров А. Технологический ренессанс: Четвертая промышленная революция. – [Електронний ресурс] // www.vedomosti.ru – 2015. – Режим доступу: www.vedomosti.ru/opinion/articles/2015/10/14/612719-promishlennaya-revolyuetsiya.
19. The Internet of Things Will Thrive by 2025. – [Електронний ресурс] // www.pewinternet.org. – 2016 – Режим доступу: <http://www.pewinternet.org/2014/05/14/internet-of-things/>
20. IBM Cloud infrastructure API. – [Електронний ресурс] // [www.ibm.com](https://www.ibm.com/developerworks/rn/library/cl-bluemixfoundry/index.html). – 2019 – Режим доступу: <https://www.ibm.com/developerworks/rn/library/cl-bluemixfoundry/index.html>
21. Mobile Apps. – [Електронний ресурс] // [azure.microsoft.com](https://azure.microsoft.com/en-us/services/app-service/mobile/). – 2019 – Режим доступу: <https://azure.microsoft.com/en-us/services/app-service/mobile/>
22. Кроссплатформенная разработка для мобильных устройств в Visual Studio. – [Електронний ресурс] // [docs.microsoft.com](https://docs.microsoft.com/ru-ru/visualstudio/cross-platform/cross-platform-mobile-development-in-visual-studio?view=vs-2019) – 2019 – Режим доступу: <https://docs.microsoft.com/ru-ru/visualstudio/cross-platform/cross-platform-mobile-development-in-visual-studio?view=vs-2019>
23. Закон України «про Державний земельний кадастр» від 11 липня 2011 року
24. Карабцова З.М. Геодезия. – Владивосток: Издательство Дальневосточного университета, 2002. – 150 с.
25. Бугаевский Л.М. Математическая картография: Учебник для вузов. – М: Златоуст, 1998. – 400с.
26. Слюсар В. И. Цифровые антенные решетки. Решения задач GPS (рус.) // Электроника: наука, технология, бизнес. — 2009. — Вып. 1. — С. 74—78.
27. Горбачёв А. Ю. Математическая модель погрешностей GPS // Авиакосмическое приборостроение. — М.: НАУЧТЕХЛИТИЗДАТ, 2010. — № 5.
28. Серапинас Б.Б. Математическая картография. – Москва : Академия , 2005. –336с.
29. Г. Буч, Д. Рамбо, И. Якобсон. Краткая история UML // Язык UML. Руководство пользователя. – The Unified Modeling Language User Guide. — 2-е. — М.: ДМК Пресс, 2006.– 496 с.

ДОДАТОК А

Машинний лістинг реалізованого програмного забезпечення

```
// проект для визначення моменту сповіщення користувача
// про наближення автобусу до нього та певної зупинки
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Newtonsoft.Json; //підключили бібліотеку для
                        // роботи з Json файлами
namespace ConsoleApp2
{
    class Program
    {
        // функція гаверсинусів для знаходження відстані між двома точками
        // A1, A2 - широта, довгота першої точки ,
        // B1, B2 - широта, довгота другої точки
        public static int ddist(double a1, double a2, double b1, double b2)
        {
            double lat1, lat2, long1, long2;
            long EARTH_RADIUS = 6372795;
            // перевести координат в радіани
            lat1 = a1 * Math.PI / 180;
            lat2 = b1 * Math.PI / 180;
            long1 = a2 * Math.PI / 180;
            long2 = b2 * Math.PI / 180;
            double c11, c12, s11, s12, delta, cdelta, sdelta, y, x, ad, distad;
            // косінуси та синуси широт та різниці довгот
            c11 = Math.Cos(lat1);
            c12 = Math.Cos(lat2);
            s11 = Math.Sin(lat1);
            s12 = Math.Sin(lat2);
            delta = long2 - long1;
            cdelta = Math.Cos(Math.Abs(delta));
            sdelta = Math.Sin(Math.Abs(delta));
            // обчислюємо довжину великого кола
            y = Math.Sqrt(Math.Pow(c12 * sdelta, 2) + Math.Pow(c11 * s12 - s11 *
c12 * cdelta, 2));
            x = s11 * s12 + c11 * c12 * cdelta;
            // обчислюємо відстань та переводимо в метри
            ad = Math.Atan2(y, x);
            distad = ad * EARTH_RADIUS;
        }
    }
}
```

```

        int ss;
        ss = (int)Math.Floor(distad);
        return ss;
    }
    // перевірка чи треба повідомляти про наближення автобусу
    public static void school_bus(int dist, double x1, double x2, double y1,
double y2, double z1, double z2, bool stop)
    {
        int distbu, distbst;
        distbu = ddist(x1, x2, y1, y2);
        distbst = ddist(x1, x2, z1, z2);
        // перевірка як близько автобус,
        // якщо він ще не пройшов зупинку користувача
        if (((distbu >= (dist - 5)) && (distbu <= (dist + 5)) && (distbst <= (2
* dist))) && (stop == false))
        {
            Console.WriteLine(" Увага! Автобус від Вас в радіусі = {0}
метрів ", dist);
            Console.WriteLine(" при цьому автобус від зупинки в радіусі
= {0} метрів ", distbst);
            Console.WriteLine(" ");
        }
        // перевірка як близько автобус,
        // якщо він вже пройшов зупинку користувача
        if (((((distbu >= (dist - 5)) && (distbu <= (dist + 5))) && (distbst <= (2
* dist))) && (stop != false))
        {
            Console.WriteLine(" Увага! Автобус від Вас в радіусі = {0}
метрів ", dist);
            Console.WriteLine(" при цьому автобус вже пройшов вашу
зупинку!");
            Console.WriteLine(" ");
        }
    }
    static void Main(string[] args)
    {
        int i = 0;
        // масив з координатами автобусу
        double[] xybus = new double[400];
        // масив з координатами користувача
        double[] xyuser = new double[400];
        //підключаємо файл з координатами автобусу
        using (StreamReader r = new StreamReader("file-bus11.json"))
        {
            string json = r.ReadToEnd(); // зчитуємо файл повністю

```

```

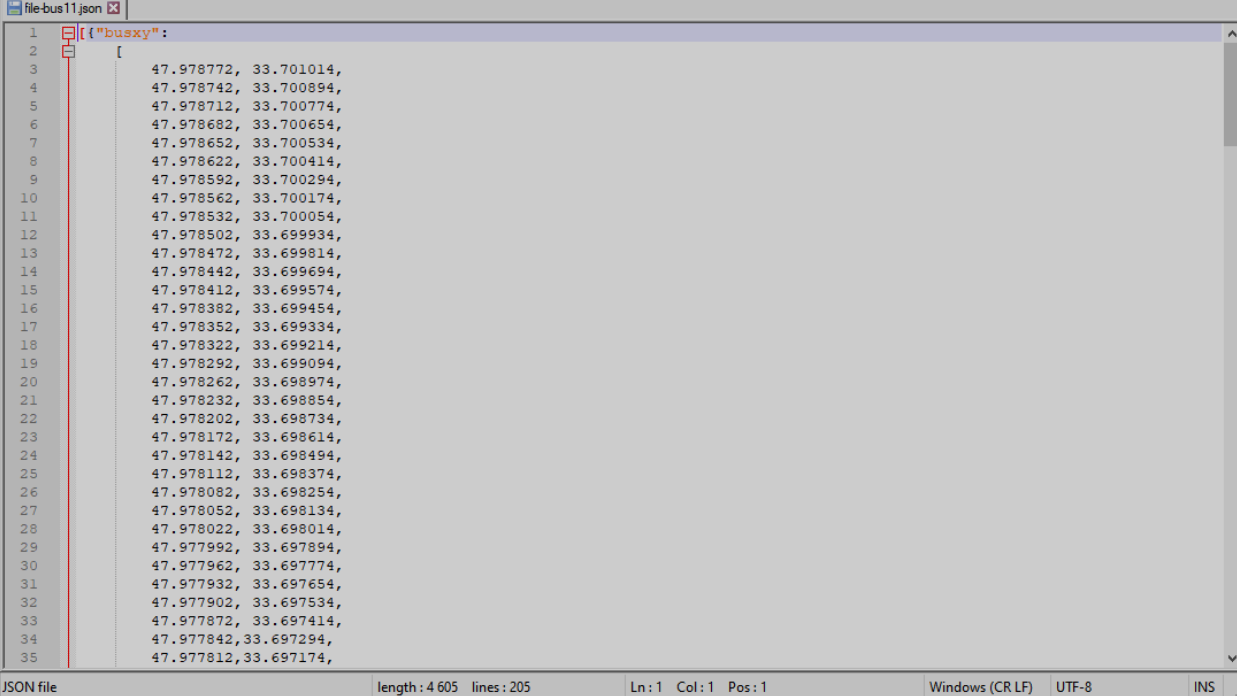
//зміст файлу переводимо в масив об'єктів
dynamic array = JsonConvert.DeserializeObject(json);
//звертаємось до об'єкту як до масиву значень
foreach (var item in array)
{
    for (i = 0; i < 400; i++)
    {
        // зчитуємо властивість busxy в звичайний масив хybus
        хybus[i] = item.busxy[i];
    }
}
}
//підключаємо файл з координатами користувача
using (StreamReader rr = new StreamReader("file-user11.json"))
{
    // зчитуємо файл повністю
    string json = rr.ReadToEnd();
    //зміст файлу переводимо в масив об'єктів
    dynamic array = JsonConvert.DeserializeObject(json);
    //звертаємось до об'єкту як до масиву значень
    foreach (var item in array)
    {
        for (i = 0; i < 100; i++)
        {
            // зчитуємо властивість userxy в звичайний масив хyuser
            хyuser[i] = item.userxy[i];
        }
    }
}
double x1, x2, y1, y2, z1, z2;
int distbst;
const int dist = 500; // дистанція для сповіщення
bool stop = false; //ознака, що зупинка не була пройдена
z1 = 47.977442; //координати зупинки - широта
z2 = 33.695036 ; //координати зупинки - довгота
Console.WriteLine(" "); Console.WriteLine(" ");
for (i=0; i< 400; i=i+2)
{
    x1 = хybus[i]; // передача координат автобусу
    x2 = хybus[i + 1];
    y1 = хyuser[i]; // передача координат user
    y2 = хyuser[i + 1];
    // перевірка чи пройдена зупинка
    distbst = ddist(x1, x2, z1, z2);
    if ((distbst <= 20) && (stop == false))

```

```
    {
        stop = true;
        Console.WriteLine(" Увага! Зупинка пройдена, вона від
автобусу в радіусі = {0} метрів ", distbst);
        Console.WriteLine(" ");
    }
    // перевірка дистанції для сповіщення
    school_bus(dist, x1, x2, y1, y2, z1, z2, stop);
    school_bus(dist / 2, x1, x2, y1, y2, z1, z2, stop);
    school_bus(dist / 4, x1, x2, y1, y2, z1, z2, stop);
}
Console.ReadKey();
}
}
```

ДОДАТОК Б

Зміст json-файлу, який містить координати автобусу



```
1 [{"busxy":
2   [
3     47.978772, 33.701014,
4     47.978742, 33.700894,
5     47.978712, 33.700774,
6     47.978682, 33.700654,
7     47.978652, 33.700534,
8     47.978622, 33.700414,
9     47.978592, 33.700294,
10    47.978562, 33.700174,
11    47.978532, 33.700054,
12    47.978502, 33.699934,
13    47.978472, 33.699814,
14    47.978442, 33.699694,
15    47.978412, 33.699574,
16    47.978382, 33.699454,
17    47.978352, 33.699334,
18    47.978322, 33.699214,
19    47.978292, 33.699094,
20    47.978262, 33.698974,
21    47.978232, 33.698854,
22    47.978202, 33.698734,
23    47.978172, 33.698614,
24    47.978142, 33.698494,
25    47.978112, 33.698374,
26    47.978082, 33.698254,
27    47.978052, 33.698134,
28    47.978022, 33.698014,
29    47.977992, 33.697894,
30    47.977962, 33.697774,
31    47.977932, 33.697654,
32    47.977902, 33.697534,
33    47.977872, 33.697414,
34    47.977842, 33.697294,
35    47.977812, 33.697174,
```

JSON file length: 4605 lines: 205 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

ДОДАТОК В

Зміст json-файлу, який містить координати користувача



```
1 [{"usezxy":  
2 [  
3 47.975766,33.693830,  
4 47.975774,33.693846,  
5 47.975782,33.693862,  
6 47.975790,33.693878,  
7 47.975798,33.693894,  
8 47.975806,33.693910,  
9 47.975814,33.693926,  
10 47.975822,33.693942,  
11 47.975830,33.693958,  
12 47.975838,33.693974,  
13 47.975846,33.693990,  
14 47.975854,33.694006,  
15 47.975862,33.694022,  
16 47.975870,33.694038,  
17 47.975878,33.694054,  
18 47.975886,33.694070,  
19 47.975894,33.694086,  
20 47.975902,33.694102,  
21 47.975910,33.694118,  
22 47.975918,33.694134,  
23 47.975926,33.694150,  
24 47.975934,33.694166,  
25 47.975942,33.694182,  
26 47.975950,33.694198,  
27 47.975958,33.694214,  
28 47.975966,33.694230,  
29 47.975974,33.694246,  
30 47.975982,33.694262,  
31 47.975990,33.694278,  
32 47.975998,33.694294,  
33 47.976006,33.694310,  
34 47.976014,33.694326,  
35 47.976022,33.694342,
```

JSON file length : 4 420 lines : 204 Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 INS