

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «**РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ
ВІДСТЕЖЕННЯ ГЕОЛОКАЦІЇ ТА ГРАФІКУ
РУХУ ГРОМАДСЬКОГО ТРАНСПОРТУ**»

Виконав: студент 2 курсу, групи 8.1228
спеціальності 122 комп'ютерні науки
освітньої програми комп'ютерні науки
(шифр і назва спеціальності)

Є. О. Таращук

(ініціали та прізвище)

Керівник доцент кафедри комп'ютерних наук,
доцент, к.т.н., Решевська К.С.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри програмної інженерії,
доцент, к.т.н., Мухін В.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет Математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Напрямок підготовки 122 комп'ютерні науки

(шифр і назва)

Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерних наук, к.т.н.,
доцент

_____ Борю С. Ю.

(підпис)

« 02 » вересня 2019 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ (СТУДЕНТЦІ)
Таращику Євгену Олександровичу**

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка мобільного додатку відстеження геолокації та графіку руху громадського транспорту"
керівник роботи (проекту) Решевська Катерина Сергіївна, к.т.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 29 » травня 2019 року № 811-с

2. Строк подання студентом роботи 16.12.2019

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розроблений програмний продукт відповідно до теми кваліфікаційної роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація до доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 02.09.2019

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	03.09.2019	
2.	Збір вихідних даних.	05.09.2019	
3.	Обробка методичних та теоретичних джерел.	11.09.2019	
4.	Розробка першого та другого розділу.	14.10.2019	
5.	Розробка третього розділу.	05.11.2019	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	12.12.2019	
7.	Захист кваліфікаційної роботи.	14.01.2020	

Студент _____
(підпис)

Є. О. Таращик _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

К. С. Решевська _____
(ініціали та прізвище)

Нормоконтроль пройдено
Нормоконтролер _____
(підпис)

О. Г. Спиця _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка мобільного додатку відстеження геолокації та графіку руху громадського транспорту»: 75 с., 28 рис., 18 табл., 8 джерел, 1 додаток.

ANDROID, GPRS, GPS, АЛГОРИТМ, БАЗА ДАНИХ, ЛІЧИЛЬНИК, МОБІЛЬНИЙ ДОДАТОК, ПРОГРАМА, СИСТЕМА КОНТРОЛЮ, СТРУКТУРНА СХЕМА, ТРАНСПОРТ.

Предмет дослідження – мобільний додаток, що дозволяє відстежувати рух громадських транспортних засобів.

Об'єкт дослідження – система контролю руху громадського транспорту з можливістю перерозподілу кількості транспортних засобів між маршрутами.

Мета роботи: Дослідження систем віддаленого контролю руху громадського транспорту та створення мобільного додатку для моніторингу громадського транспорту з точки зору оптимальності маршрутів слідування та завантаженості.

Метод дослідження – оглядовий, аналітичний.

Для реалізації поставленої мети було вирішено такі завдання:

- розглянуті принципи та особливості побудови мобільних систем з визначення місцезнаходження;
- розроблена структура мобільного додатку;
- обрані засоби реалізації мобільного додатку;
- створено базу даних;
- розроблено інтерфейс та структуру мобільного додатку;

При виконанні роботи застосовані система управління базами даних (СУБД) MySQL та SQLite, середа розробки програмного забезпечення Android Studio, скриптова мова програмування PHP.

SUMMARY

Master's qualification work "Development of a mobile application for geolocation tracking and public transport traffic schedules": 75 pages, 28 figures, 18 tables, 8 sources, 1 application.

ALGORITHM, ANDROID, CONTROL SYSTEM, DATABASE, GPRS, GPS, METER, MOBILE APP, PROGRAM, STRUCTURAL DIAGRAM, TRANSPORT.

The subject of the study is a mobile application that allows you to track the movement of public vehicles.

The object of study is a public transport traffic control system with the possibility of redistributing the number of vehicles between routes.

Purpose: To investigate remote control systems for public transport and create a mobile application for monitoring public transport in terms of optimization of routes and congestion.

The research method is a survey, analytical.

To accomplish this goal, the following tasks were solved:

- the principles and features of construction of mobile location systems are considered;
- mobile application structure has been developed;
- Selected means of implementation of the mobile application;
- database was created;
- interface and structure of the mobile application have been developed;

The database management system (DBMS) of MySQL and SQLite, the environment of development of Android Studio software, the scripting language of PHP programming are applied at work.

ЗМІСТ

Завдання на кваліфікаційну роботу	2
Реферат	4
Summary	5
Зміст	6
Перелік умовних скорочень.....	7
Вступ.....	8
1 Огляд області розробки, постановка завдань	10
1.1 Застосування транспортних систем позиціонування	10
1.2 Огляд існуючих аналогів.....	12
1.3 Постановка задач роботи	18
2. Розробка структури системи і вибір методу реалізації	20
2.1 Розробка структури системи.....	20
2.2 Вибір засобів реалізації системи	24
2.3 Розробка структури бази даних	31
3 Написання програмного коду та відлагодження програми.....	42
3.1 Блок схема алгоритму мобільного додатка.....	42
3.2 Розробка інтерфейсу додатка.....	44
3.3 Написання скрипта для збору даних	50
3.4 Розробка та відлагодження роботи лічильника пасажирів	52
3.5 Написання мобільного додатка.....	65
Висновки.....	71
Перелік посилань	73
Додаток А Скрипт Transport.php	74

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ТЗ– транспортні засоби

GPS – Global Positioning System – система глобального позиціонування

GPRS – General Packet Radio Service – пакетна радіозв'язок загального користування

БД – база даних

ЕОМ – електронно–обчислювальна машина

ВСТУП

В останні десятиліття у великих містах спостерігається велике зростання числа міського транспорту. Природно, разом із зростанням транспортного парку зростає і навантаження на міські і, в меншій мірі, міжміські дороги і транспортні магістралі. Автомобільні потоки перевантажують транспортну мережу, часом паралізуючи сполучення між різними частинами міста.

В даний час транспортна система вийшла і зайняла одне з пріоритетних місць серед основних чинників, що визначають ефективне функціонування економіки країни, стала найважливішою умовою її подальшого розвитку і входження регіонів у світове господарство. Транспортна система в рамках будь-якого регіону повинна гарантувати необхідні умови для функціонування і розвитку основних галузей виробництва, забезпечувати безперешкодне і рівне для всіх категорій населення транспортне обслуговування і забезпечувати максимально ефективне використання соціально-економічного та виробничого потенціалу. Кожна країна або регіон повинен мати таку транспортну систему, яка повністю задовольняла б попит даної території в транспортних послугах.

Пасажирський транспорт є складовою частиною транспортної інфраструктури як міста, так і регіону. Його злагоджене, стійке і ефективне функціонування є необхідною умовою підвищення якості життя населення і подальшого соціально-економічного розвитку міста, регіону та країни в цілому.

Система супутникового моніторингу транспорту включає наступні компоненти:

– транспортний засіб, обладнаними GPS контролером або трекером, який отримує дані від супутників і передає їх на серверний центр моніторингу за допомогою GSM, CDMA або рідше супутникової і УКХ зв'язку. Останні два актуальні для моніторингу в місцях, де відсутня повноцінне GSM-покриття;

- серверний центр з програмним забезпеченням для прийому, зберігання, обробки і аналізу даних;

- комп'ютер диспетчера, провідного моніторингу автомобілів.

Використання систем супутникового моніторингу підвищує якість і ефективність роботи корпоративного транспорту, і в середньому на 20–25% знижують витрати на паливо і утримання автопарку.

Мета дослідження: Дослідження систем віддаленого контролю руху громадського транспорту та створення мобільного додатку для моніторингу громадського транспорту з точки зору оптимальності маршрутів слідування та завантаженості.

Згідно постановленої мети були поставлені наступні задачі дослідження:

- вивчення підходів до реалізації систем контролю руху транспорту;
- аналіз існуючих аналогів;
- вивчення методики геокодування в інформаційних системах орієнтованих на певну групу користувачів;
- розробка структури системи контролю руху на основі мобільних додатків, серверу та лічильника пасажирів;
- розробка програми для пошуку оптимального маршруту громадського транспорту.

Об'єкт дослідження. Програмне забезпечення контролю руху громадського транспорту.

Предмет дослідження. Система контролю та аналізу руху громадського транспорту, а також програмне забезпечення мобільного додатка та сервера.

Методи дослідження – системний, структурно–функціональний, аналіз, синтез, аналогія, порівняння, статистичний аналіз.

1 ОГЛЯД ОБЛАСТІ РОЗРОБКИ, ПОСТАНОВКА ЗАВДАНЬ

1.1 Застосування транспортних систем позиціонування

Використання мобільних пристроїв значно полегшує керування, оскільки це зручний інструмент та мобільні технології зростають швидко, тим самим полегшуючи електронне управління та контроль.

Географічна інформаційна система (ГІС, геоінформаційна система) – програмно–апаратний комплекс, призначений для збору, управління, аналізу і відображення просторово розподіленої інформації. ГІС – не тільки і не стільки інформаційні системи для географії, скільки інформаційні системи з географічно організованою інформацією. У найпростішому варіанті географічні інформаційні системи – поєднання звичайних баз даних (атрибутивної інформації) з електронними картами, тобто потужними графічними засобами.

Основна ідея ГІС – зв'язок даних на карті і в базі даних. ГІС – це і аналітичні засоби для роботи з будь–координатно–прив'язаною інформацією. В принципі, ГІС можна розглядати як певне розширення концепції баз даних. У цьому сенсі ГІС фактично являє собою новий рівень і спосіб інтеграції та структурування інформації. ГІС пропонує абсолютно новий шлях розвитку картографії. Долаються основні недоліки звичайних карт – їх статичність і обмежена ємність як носія інформації. В останні десятиліття паперові карти через перевантаженість інформацією стають нечитабельними. ГІС ж забезпечує управління візуалізацією інформації. З'являється можливість виводити (на екран, на тверду копію) тільки ті об'єкти або їх безлічі, які цікавлять нас в даний момент. Фактично здійснюється перехід від складних комплексних карт до серії взаємопов'язаних приватних карт. При цьому поліпшується структурованість

інформації, а, отже, підвищується ефективність її обробки і аналізу. У ГІС карта оживає і стає дійсно динамічним об'єктом в сенсі:

- змінності масштабу;
- перетворення картографічних проекцій;
- варіювання об'єктним складом карти;
- можливості опитувати через карту в режимі реального часу численні бази даних;
- зміни способу відображення об'єктів (колір, тип лінії і т.п.), в тому числі і визначення символіки через значення атрибутів, тобто синхронізації візуалізації зі змінами в базах даних;
- легкості внесення будь-яких змін.

Маршрутний громадський транспорт в будь-якій формі своєї організації однозначно вимагає коштів підвищеного контролю витрат палива і контролю за перебігом своєї автотехніки. Важливим питанням залишається оцінка витрат ресурсу як для автобуса великої місткості, так і для стандартного маршрутного таксі, а саме відстеження кількості рейсів за добу, виявлення з'їздів з маршруту, проходження контрольних точок (зупинок), попередня оцінка пасажиропотоку.

Це питання можна назавжди вирішити за допомогою системи GPS-моніторингу автотранспорту, яка буде корисною власнику автопідприємства будь-якої форми власності. Програмне забезпечення системи моніторингу автотранспорту дозволяє створити на векторній карті будь-яку кількість контрольних зон, проходження через які буде фіксуватися системою в автоматичному режимі. Стеження за автобусом в реальному часі дозволить диспетчеру оперативно реагувати на яка складається у маршруті ситуацію.

Відстежити реальне місце розташування автобуса на маршруті, швидкість його руху, кількість і тривалість зупинок, а також залишок палива в баку можна за допомогою GPS-трекера, встановленого на автобусі. Функціональні можливості програми-клієнта системи gprs контроль автотранспорту дозволяє

перетворити типовий маршрут громадського транспорту в географічну зону, що дозволить в автоматичному режимі контролювати всі несанкціоновані відхилення від маршруту або простої, з фіксацією часу моменту настання і тривалості такої події.

Системи супутникового моніторингу транспорту вирішують наступні завдання:

- визначення координат місцезнаходження транспортного засобу а також його напрямки, швидкості руху;
- облік пересування транспортних засобів, та ін.;
- аналіз пройдених маршрутів, швидкісного режиму, витрати палива та ін.;

Система супутникового моніторингу транспорту включає наступні компоненти:

- транспортний засіб, обладнаними GPS контролером або трекером, який отримує дані від супутників і передає їх на серверний центр моніторингу за допомогою GSM, CDMA. Останні два актуальні для моніторингу в місцях, де відсутня повноцінне GSM–покриття;
- серверний центр з програмним забезпеченням для прийому, зберігання, обробки і аналізу даних;
- комп'ютер диспетчера, провідного моніторинг транспортів.

1.2 Огляд існуючих аналогів

Більшість українських громадян користується громадським транспортом. І поки зупинки цього транспорту не оснащуються електронними табло про час його прибуття, часто міським жителям доводиться лише гадати, коли ж вдасться виїхати в пункт призначення. Часом це очікування посилюється похмурої

погодою, або боязню спізнитися. В результаті – пасажери заталкиваються в переповнений транспорт, бо не знають, коли прийде наступний.

Але, ми живемо у вік технологій і кишенькових девайсів, за допомогою яких можна полегшувати своє життя. У цій збірці ми розповімо жителям міст України, де і як в режимі реального часу відслідковувати громадський транспорт онлайн.

В першу чергу, варто знати, що для подібного відстеження транспорту вам знадобиться доступ до інтернету. Також подібні сервіси представлені двох типів – для ПК (комп'ютера), або ті що встановлюються на смартфон / планшет у вигляді додатку які показують місце розташування автобусів, тролейбусів, трамваїв.

З маршрутками справа йде гірше. Їх побачити онлайн можна тільки в тому випадку, якщо вони оснащені GPS-трекером . А власники приватних маршруток, як ви розумієте, такими речами не завжди «заморочуються». Більш того, бувають випадки, що водії маршруток, не бажаючи бути контрольованими, спеціально виводять з ладу трекери. У цьому випадку навіть оснащена маячком машина стане невидимою для програми. У дипломному проекті розглядається кілька додатків 2 з них працюють в Україні та одне додаток використовуються в Європі.

Перший додаток для відстеження транспорту, Easyway, приклад наведено на рис.1.1. було запущено ще більше двох років тому і з тих пір зазнало багато змін. Хоча воно залишається досить «великоваговим» і працює не дуже спритно на бюджетних пристроях, але зручність його використання зросла в рази. Easyway підійде насамперед для того, щоб відстежувати потрібний вам транспорт на маршруті. Ви зможете вийти з дому і дійти до зупинки якраз до часу прибуття.

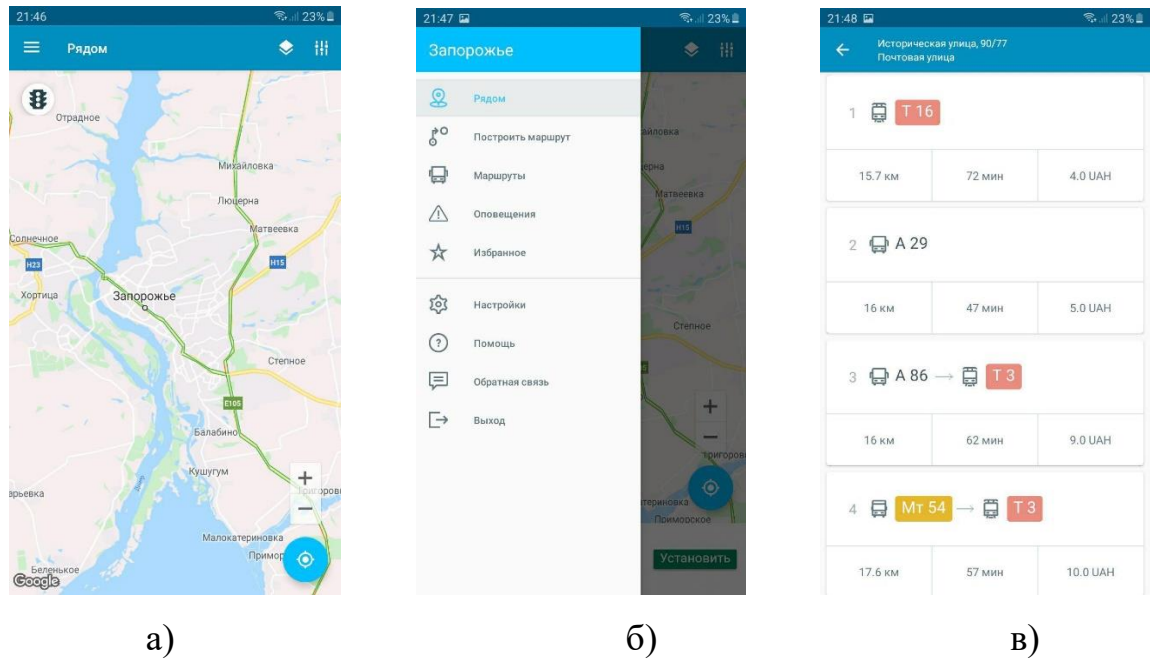


Рисунок 1.1 – Скріншоти мобільного додатка Easuyway:

- а) Зображення карти і головне меню додатка Easuyway;
- б) Зображення меню і функціоналу;
- в) Зображення побудова шляху і вибір маршрути.

Розглянемо його переваги над іншими додатками:

- наявність Offline-режиму роботи;
- показує вартість проїзду;
- показує навантаження на дорогу в реальному часі;
- виводить маршрут транспорту зі списком зупинок і показує які ще номери маршрутів зупиняться там;
- має всі види карт;
- настройка пошуку маршруту (оптимальний за часом, дешевше, швидше), а так можна вибрати тип транспорту;
- можливість створення закладок по місцях, де найчастіше буває користувач.

Розглянемо Easuyway недоліки у порівнянні з іншими додатками:

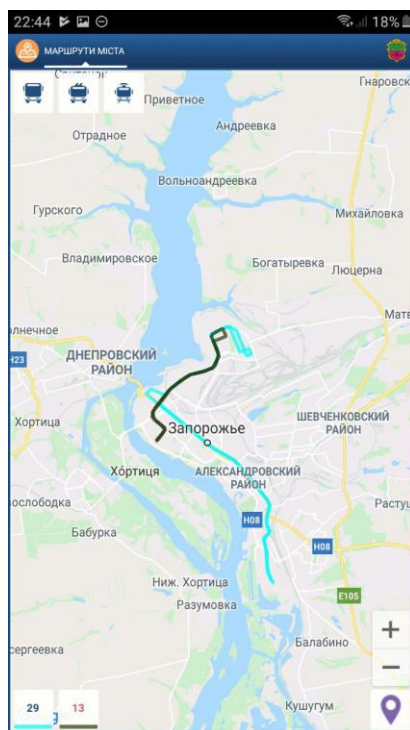
- не відстежує і не виводить кількість людей в транспорті.

– не стабільна робота додатка, періодично відбуваються збої в роботі (вилітає і «глючить»).

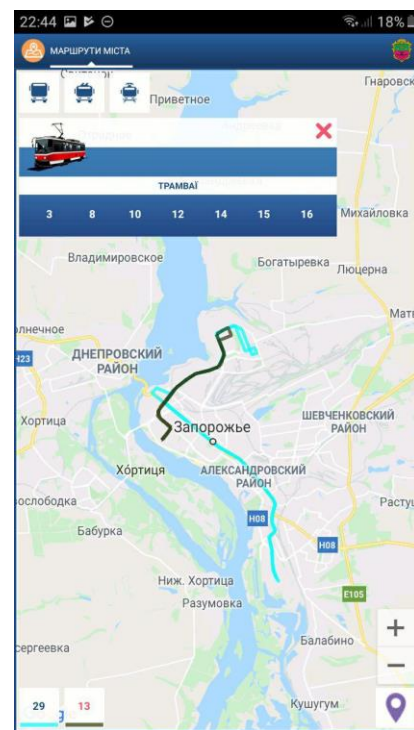
Другий додаток для відстеження транспорту, Dozor City, приклад наведено на рис.1.2

Розглянемо переваги Dozor City над іншими додатками:

- простий інтерфейс;
- немає реклами;



а)



б)

Рисунок 1.2 – Скріншоти мобільного додатка Dozor City:

- а) Зображення карти і головне меню додатка Dozor City;
- б) Зображення побудова шляху і вибір маршрути.

Розглянемо недоліки Dozor City у порівнянні з іншими додатками:

- відсутність Offline-режиму роботи;

– пошук оптимального шляху з А в Б. Працює пошук тільки з назвами зупинок, немає можливості використовувати адресу вулиці або власного місця розташування;

– не відстежує і не виводить кількість людей в транспорті

Третій додаток для відстеження транспорту, Jakdojade, приклад наведено на рис.1.3

Розглянемо переваги додатка Jakdojade над іншими додатками:

– наявність Offline-режиму роботи;

– показує вартість проїзду;

– виводить маршрут транспорту зі списком зупинок і показує, які ще маршрути транспорту зупиняються там;

– є всі види карт;

– налаштування пошуку маршруту (Оптимальний за часом, ціною, типу транспорту);

– закладки по місцях де частіше буває користувач.

– відстежує в реальному часі і показує кількість зупинок;

– сімейний аккаунт;

– підв'язка до оплати транспорту (онлайн) ;

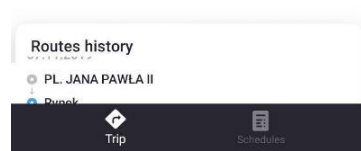
– будільник на транспорт за часом / датою.

Розглянемо його недоліки над іншими додатками:

– не відстежує і не виводить кількість людей в транспорті;

– нав'язлива реклама;

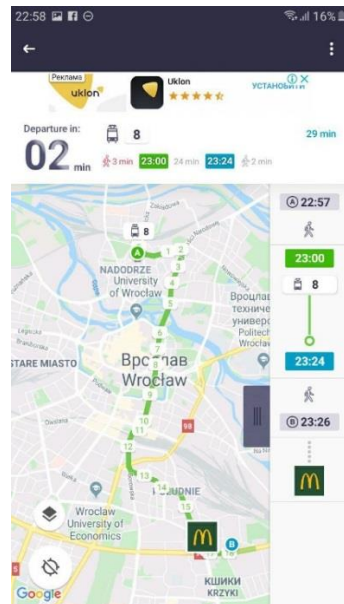
– є платний контент.



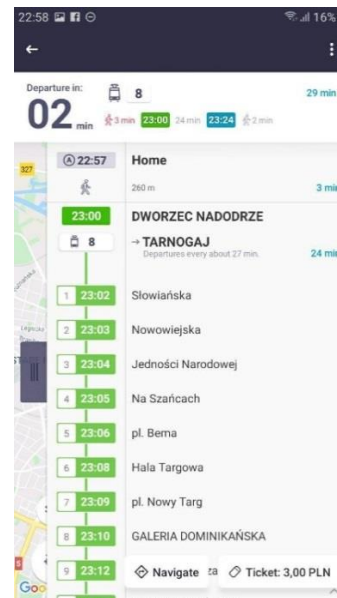
a)



б)



в)



г)

Рисунок 1.3 – Скріншоти мобільного додатка Jakdojade:

а) Головне меню додатка Jakdojade;

б) Вибір маршруту:

в) Зображення побудованого шляху, список зупинок, час відправлення і прибуття;

г) Список зупинок.

Структурований аналіз існуючих аналогів наведено в табл.1.1.

Таблиця 1.1 Порівняння мобільних додатків.

Функціонал	Easyway	Dozor City	Jakdojade
Наявність Offline–режиму роботи	✓	×	✓
Показує вартість проїзду	✓	×	✓
Має кілька зображень карт	✓	×	✓
Пошук оптимального шляху	✓	×	✓
Відстежує в реальному часі	×	×	✓
Виводить кількість пасажирів в транспорті	×	×	×
Планування поїздки	×	×	✓
Показує час відправки транспорту та прибуття	×	×	✓
Можливість створення закладок по місцях	✓	×	✓
Сімейний аккаунт(контроль дитини)	×	×	✓

Проаналізувавши додатки наведені в табл 1.1. Ми отримуємо те, що в система контролю транспортного руху і якість транспортування пасажирів сильно поступається. З цього ми проаналізувавши всі перевага і недоліки створили свій мобільний додаток, так само ми додали деякі доопрацювання і покращання.

1.3 Постановка задач роботи

У роботі пропонується розробка системи контролю руху міського транспорту яка повинна мати такі параметри:

- відстежувати транспорт і показувати їх місце положення;
- підраховувати кількість пасажирів
- показує завантаженості транспортної лінії і при необхідності додавати транспорт на лінію, не заважаючи міському потоку
- онлайн моніторинг транспорту і планування маршруту
- мобільний додаток повинен мати ряд функцій.
- відображати на карті, де знаходиться ваш тролейбус, автобус, трамвай або маршрутка и Показувати напрямок його руху;
- визначати фактичний час прибуття громадського транспорту на кожну зупинку;
- показувати на карті повний перелік міських маршрутів транспорту, як комунальних тролейбусів, так і приватних автобусів;
- показувати транспорт, адаптований для людей з обмеженою мобільністю;
- відображати місце розташування користувача за даними GPS, тому користувач може знайти найближчу зупинку.

Під час проведення досліджень було вирішено розробляти наш додаток для мобільних пристроїв працюють під управлінням платформи android.

2. РОЗРОБКА СТРУКТУРИ СИСТЕМИ І ВИБІР МЕТОДУ РЕАЛІЗАЦІЇ

2.1 Розробка структури системи

Найістотнішою відмінністю багатьох систем супутникового моніторингу, представлених на ринку, є функціональність серверного та клієнтського програмного забезпечення, можливість різнобічно обробляти дані.

Функції серверного центру може виконувати як звичайний комп'ютер з встановленим програмним забезпеченням для простих систем моніторингу, так і розподілена серверна система з використанням декількох серверів, що виконують різні завдання, здатна вести одночасний моніторинг десятків тисяч автомобілів і забезпечувати підключення до серверного центру декількох тисяч користувачів (диспетчерів) одночасно.

Диспетчерське програмне забезпечення для супутникового моніторингу транспортних засобів можна умовно розділити на кілька типів :

- ПЗ, що містить всі компоненти, включаючи карти і базу даних руху об'єктів на єдиному комп'ютері;

- ПЗ, що має клієнтську частину, яка встановлюється на комп'ютери диспетчерів;

- ПЗ, що використовує web-інтерфейс, що дозволяє уникнути установки будь-яких спеціальних компонентів і вести моніторинг з будь-якого комп'ютера, підключеного до Інтернет .

Різновидом останнього варіанту є ПЗ, що використовує трирівневу архітектуру, коли компоненти і функції центру обробки даних розподілені між декількома серверами: бази даних, картографічної підсистеми, телекомунікаційним сервером і сервером додатка, що забезпечує роботу web-інтерфейсу користувача.

Важливу роль в програмному забезпеченні для супутникового моніторингу грає картографічна основа. Чим більш деталізовані і якісні карти використовуються в системі, тим зручніше диспетчерам вести моніторинг і стежити за місцезнаходженням транспортних засобів.

Як правило, в програмах, що мають клієнтську частину, карти встановлюються безпосередньо на комп'ютер користувача. А web-системи використовують онлайн карти, які завдяки Web-GIS сервера завантажуються в міру необхідності, що, безумовно, вимагає високої швидкості інтернет-з'єднання. Web-GIS дозволяє одночасно використовувати такі карти, як Яндекс.Карты, Карты Google, OpenStreetMap, Карты Yahoo!, Карты Bing, Карты Gurtam і інші.

Програмне забезпечення для супутникового моніторингу зазвичай має ряд інтерфейсів. Вхід користувачів в систему моніторингу найчастіше захищений паролем для запобігання несанкціонованого доступу до інформації. У системах існує певна ієрархічна структура, при якій адміністратор системи моніторингу управляє правами доступу різних користувачів до різних об'єктів моніторингу і різних функцій програми.

Найпоширеніші функції, які присутні в більшості систем супутникового моніторингу:

- підключення та налаштування трекерів в системі;
- підключення та налаштування датчиків в системі;
- моніторинг поточного стану транспорту на мапі;
- моніторинг стану приладів і датчиків транспортного засобу;
- перегляд маршруту переміщення і пробігу автомобіля за вибраний інтервал часу;
- створення точок інтересу і геозон на карті;
- контроль переміщення з / в геозон;

- настройка повідомлень, що висилаються системою, коли відбуваються певні події (перевищення швидкості, слив палива і ін.);
- настройка шаблонів звітів, виконання звітів;
- побудова графіків на підставі даних системи;
- управління об'єктами моніторингу через SMS команди або CSD з'єднання;
- створення маршрутів і шляхових точок, контроль дотримання маршруту.

Додаткові функції, які розширюють можливості системи супутникового моніторингу:

- пошук найближчого до заданої точки автомобіля;
- передачу текстових повідомлень водієві транспортного засобу і назад, від водія до диспетчеру;
- забезпечення голосового зв'язку з водієм;
- ведення журналу техобслуговування автомобіля;
- визначення периметра і площі об'єктів на карті;
- web–доступ в систему моніторингу з мобільного телефону або КПК;
- експорт зі звітів у формати, підтримувані іншим ПО (Excel, Pdf, XML, CSV і ін.);
- зміна іконок, що відображають об'єкти на карті;
- передача даних від іншого обладнання встановленого на транспортному засобі (тахограф, датчик рівня палива)

В процесі виконання дипломного проекту була запропонована структура системи моніторингу руху громадського транспорту з обліком пасажирів, що наведена на рис.2.1.

Структура системи включає в себе GPS датчик, який приймає з супутника дані про свої координати і відправляє їх до мікроконтролера, а той в свою чергу комутується з сервером через модуль мобільного інтернету (2G/GPRS –

оскільки пакети даних невеликі та такий спосіб мобільного зв'язку є на сьогоднішній день найдешевшим).

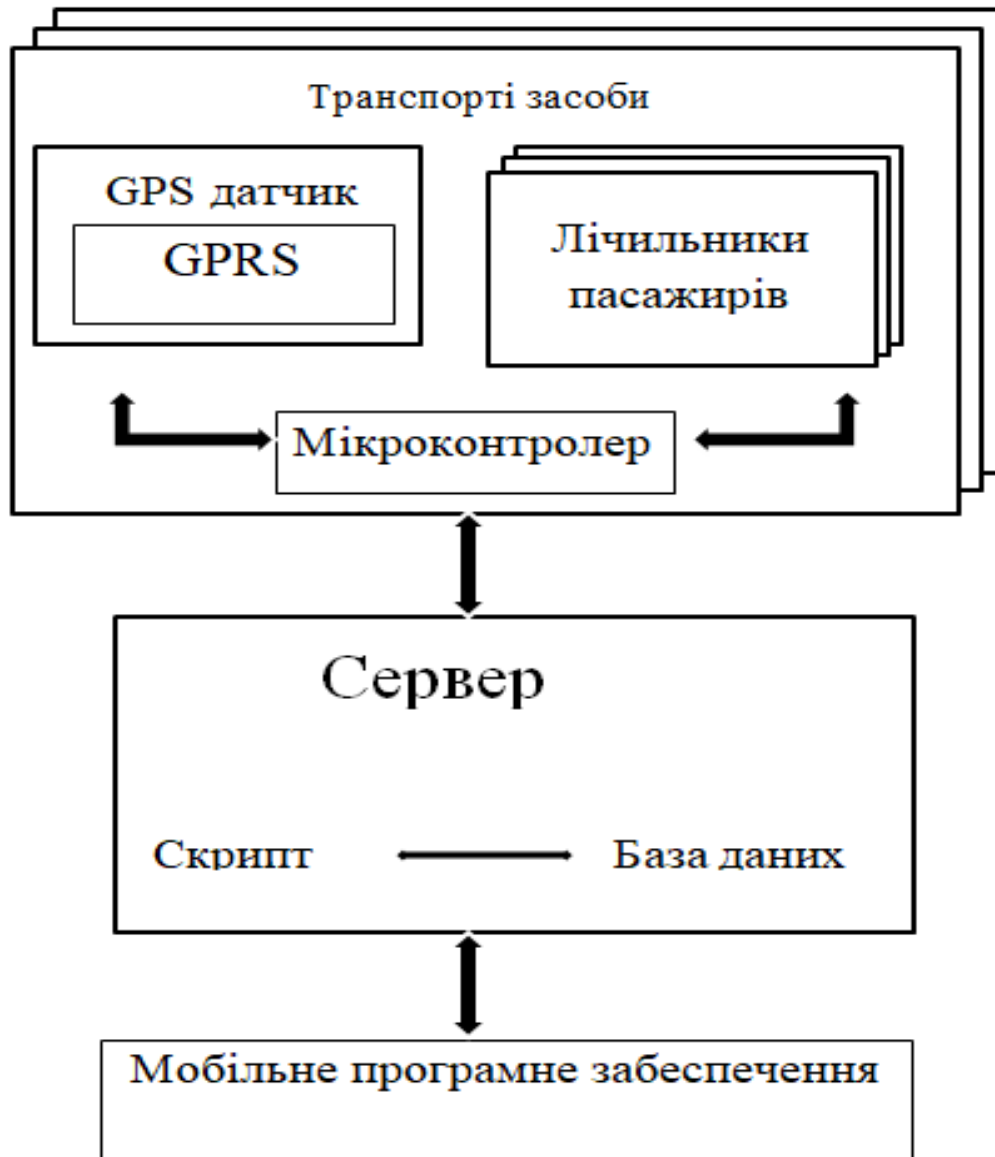


Рисунок 2.1 – Структурна схема системи

Транспортний засіб також буде обладнуватись лічильниками на основі мікроконтролера, який буде рахувати кількість перетинань п'яти ліній з пар світло діоду та фотодіоду. Кількість таких лічильників для кожного транспортного засобу залежить від кількості його дверей. Лічильник не лише проводить підрахунок перетинань, а й визначає кількість перетинань у кожному

напрямку, щоб отриману інформацію по кількості пасажирів які ввійшли та вийшли з транспортного засобу за певний проміжок часу. Інформація щодо кількості пасажирів, що ввійшли та вийшли з транспортного засобу також запакується в загальний пакет для передачі за допомогою GPRS модуля з використанням POST запиту на сервер.

Віддалений сервер після отримання запиту від окремих GPRS модулів окремих транспортних засобів зберігає данні за допомогою скрипта в базі даних.

Користувач заходячи в мобільний додаток після вводу даних, що його цікавлять отримує необхідну інформацію завдяки знову ж таки зв'язку з сервером через мобільний інтернет його мобільного пристрою. Мобільний пристрій формує запит на сервер в якому вказує набір інформації, що цікавить користувача та сервер дістає з бази даних необхідні данні і відправляє їх користувачеві. При цьому мобільний додаток обробляє отримані дані та подає користувачеві у зручній для нього формі.

2.2 Вибір засобів реалізації системи

Для моніторингу руху міського транспорту було вибрано GPS–модуль NEO–6M. Тому що, він справляється зі своїм завданням і має гнучку систему для виконання вимог нашої задачі. Так само він має невеликі габарити що дозволяє скомпонувати все в один корпус. Зовнішнє зображення модуля наведено на рис.2.2. Його характеристики наведені в табл. 2.1.



Рисунок 2.2 – Зовнішній вигляд GPS-модуль NEO-6M

Таблиця 2.1 – Характеристики GPS-модуля NEO-6M

Параметр	Значення
Можливість роботи з програмами	U-Center і т.п
Довжина x Ширина x Висота антени	25x25
Інтерфейси UART (виведений),	SPI, DDC, PC;
Передає координати в форматі	NMEA
Швидкість відновлення	5 Гц
Швидкість порту по промовчати	9600
Теплий старт	1 с
Холодний старт	27 с
Тип антени	зовнішня пасивна керамічна
Точність позиціонування	5 м
Чіп	GPS Ublox NEO-6M
Чутливість	-147 dBm
Тип показань	UART-порт
Довжина x Ширина x Висота	23x30
Робоча температура	-40 ... 85 °C
Робоча напруга	3 ... 5 В
Споживаний струм	45 мА

В якості цільової операційної системи для мобільного додатку було обрано Android оскільки це система є одна з популярних в світі і їй користуються 59,1%.

Також Android має ряд переваг перед іншими системами а саме:

- операційна система Android є повністю відкритою,
- власники пристроїв на базі Android можуть завантажувати, що дозволяють змінювати різні параметри і втручатися в роботу ОС, підвищуючи її функціонал або прибираючи непотрібні функції;
- кастомізації зовнішнього вигляду Android;
- android можуть запускати будь-які програми в фоні і вони будуть залишатися активними до тих пір, поки в ньому не закінчиться вільна пам'ять або процес завантаження не завершиться до кінця.

В якості керуючого елемента лічильника пасажирів обрано Arduino Nano на базі мікроконтролера ATmega328 оскільки він має достатню кількість портів введення / виводу, а так само підходить нам по технічним вимогам для нашого проекту. Також він має невелику вартість і до нього є велика кількість додаткових модулів. Зовнішній вигляд Arduino Nano наведено на рис.2.3, а характеристики наведені в табл. 2.2.



Рисунок 2.3 – Зовнішній вигляд Arduino Nano

Таблиця 2.2 – Характеристики Arduino Nano

Параметр	Значення
Керуючий мікроконтролер	ATmega328
Робоча напруга	5 В
Напруга живлення (рекомендована)	7–12В
Напруга живлення (максимальна)	6–20В
Цифрові входи/виходи	14 (із яких 6 можуть використовуватись в якості ШІМ–виводів)
Максимальний вихідний струм одного виводу	40 мА
Максимальний вихідний струм 3.3V	50 мА
Flash– пам'ять	32 КБ із яких 0,5КБ використовується завантажувачем
SRAM	2 КБ
EEPROM	1 КБ
Тактова частота	16 МГц

Для створення датчику лічильника пасажирів було вибрано наступні елементи: модуль інфрачервоного світлодіода KY–005 (рис.2.4) і модуль інфрачервоного приймача VS1838В (рис.2.5). Обрані елементи нам оптимально підходять оскільки мають невеликий розмір оскільки потрібно компактно розмістити по п'ять комплектів випромінювачів та приймачів в кожному двірному отворі, для підрахунку кількості пасажирів в транспортному засобі.

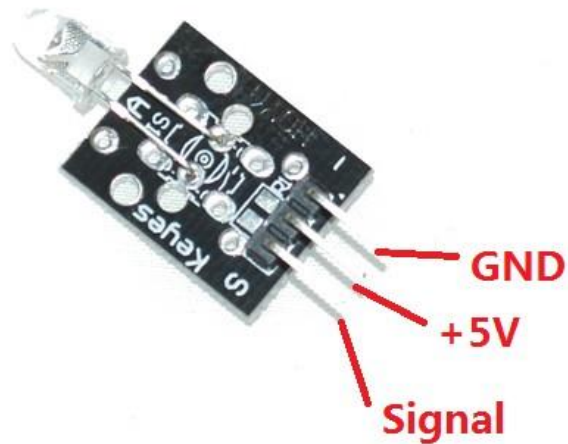


Рисунок 2.4 – Зовнішній вигляд світлодіодного модуля KY–005

Модуль KY–005 дозволяє змонтувати світлодіод інфрачервоного (ІЧ) випромінювання в необхідній точці конструкції приладу. Модуль KY–005 дає можливість налагодити канал передачі даних або зібрати разом з фотоприймачем надійно працюючу фотопара за принципом перекриття світлового потоку. Як приймач інформації переданої по ІЧ каналу застосовують ІЧ фотодіоди. При передачі даних ІЧ канал не сприйнятливий до електромагнітних завад радіо діапазону і не створює перешкод радіоприймачів. Застосування в оптопарах вигідно завдяки слабкій чутливості ІЧ фотоприймача до звичайного освітлення. Модуль ІЧ світлодіода KY–005 полегшує макетування схем і дозволяє зробити простішим вивчення властивостей ІЧ світлодіоди.

Характеристики:

Струм споживання 15 мА

Головний електричний параметр ІЧ світлодіоди – номінальний струм. При складанні системи з застосуванням KY–005 від інженера потрібно застосувати ланцюги харчування, що забезпечують номінальний струм. Ми не наводимо напруга живлення модуля, так цей параметр може змінюватися в широких межах завдяки установці на платі модуля ІЧ світлодіоди різних резисторів або

стабілізаторів струму. Виробник встановлює в модуль різні типи світлодіодів. Напруга на світлодіоді повинно знаходитися в діапазоні 1,2 – 1,5 В.

Деякі типи ІЧ світлодіодів мають світіння частина якого знаходиться у видимій частині спектру.

Підключення КУ–005

Для забезпечення струму 15 мА при харчуванні світлодіода напругою 5 В послідовно з ним слід встановити резистор 240 Ом. На платі передбачені контакти для монтажу резистора. Залежно від обраної лінії управління світлодіодом – харчування модуля або лінія сигналу резистор встановлюється на тому чи іншому посадковому місці плати.

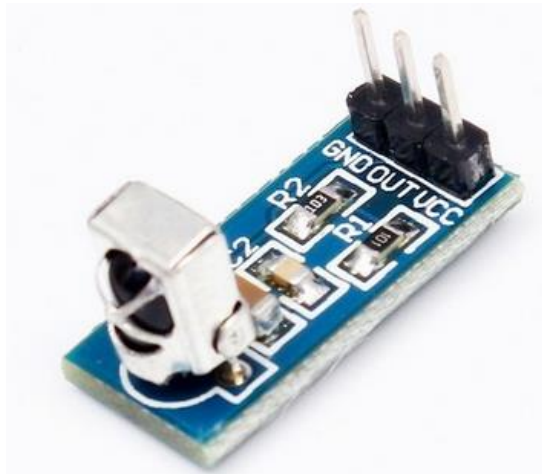


Рисунок 2.5 – Зовнішній вигляд фотоприймального модуля VS1838B

Призначений для прийому сигналу від передавачів інфрачервоного випромінювання. Сприймає елемент модуля – VS1838B. Сигнал модуля орієнтований на обробку мікроконтролером.

Характеристики:

Напруга живлення 2,7 – 5,5 В;

Частота модуляції 38 кГц;

Діапазон температури експлуатації –20 ... 80 ° С.

Для передачі отриманої з лічильника пасажирів та GPS трекера інформації запитів на сервер для обробки за допомогою скриптів використовується модуль Sim800.

Стандартний інтерфейс керування компонента SIM800L надає доступ до сервісів мереж GSM/GPRS 850/900/1800/1900МГц для обміну цифровими даними GPRS. Поставляється з вбудованою антеною, також можна підключити додаткові антени для поліпшення якості сигналу.

Керувати модулем можна за допомогою персонального комп'ютера через перетворювач інтерфейсу USB–UART або безпосередньо через UART модулем мікроконтролера.

Компонент SIM800L має реалізований стек протоколу TCP/IP. Містить мікросхему MT6260SA компанії MediaTek і мікросхему приймача RFMD RF7176.

Зовнішній вигляд модулю SIM800L показаний на рис. 2.6.



Рисунок 2.6 – Модуль GSM/GPRS SIM800L MicroSIM з антенною

Технічні характеристики:

- напруга живлення: від 3.4В до 4.4В;
- рекомендована напруга живлення: 4В;
- струм режиму очікування: 0.7 мА;
- максимальний струм: 500 мА;

- максимальна напруга високого рівня інтерфейсу UART: 2.8 В;
- швидкість UART: 1200–115200 бод;
- робочі діапазони EGSM900, DCS1800, GSM850, PCS1900;
- потужність передачі DCS1800, PCS1900: 1 Вт;
- потужність передачі GSM850, EGSM900: 2 Вт;
- режим мережі: 2G;
- мікрофон: електретний;
- управляється командами AT через UART: (3GPP TS 27.007, 27.005 SIMCOM enhanced AT Commands);
- автоматичне визначення швидкості передачі керуючих AT команд;
- відправка та отримання GPRS даних (TCP/IP, HTTP, и т.д.);
- максимальна швидкість передачі GPRS даних: 85.6 Кбод;
- кодування: CS–1, CS–2, CS–3 и CS–4;
- GSM протокол: 07.10 протокол;
- підтримка неструктурованих даних додаткових послуг USSD;
- підтримує PAP (протокол ідентифікації пароля);
- підтримка годинника реального часу RTC;
- підтримка сімкартки з живленням 3В і 1.8В;
- робоча температура: – 30 до 75 °С;
- розміри: 25x25 мм.

2.3 Розробка структури бази даних

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така

організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків.

Характерна риса файлового підходу – вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних.

База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл–програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, як у випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

СУБД – це складна програмна система накопичення та з наступним маніпулюванням даними, що представляють інтерес для користувача. Кожній прикладній програмі СУБД надає інтерфейс з базою даних та має засоби безпосереднього доступу до неї. Таким чином, СУБД відіграє центральну роль в функціонуванні автоматизованого банку даних.

Найпоширенішою моделлю даних є реляційна модель. Реляційна модель даних являє собою набір двомірних плоских таблиць, що складаються з рядків і стовпців. Первинний документ або лінійний масив являє собою плоску двомірну таблицю. Така таблиця називається відношенням, кожний стовбець– атрибутом, сукупність значень одного типу (стовпця) – доменом, а рядка – кортежем. Таким чином, стовпці таблиці являються традиційними елементами даних, а рядки – записами. Таблиці (відношення) мають імена. Імена також присвоюються і стовпцям таблиці. Кожний кортеж (запис) відношення має ключ. Ключі є прості і складні. Простий ключ–це ключ, який складається з одного атомарного атрибуту, значення якого унікальне (яке не повторюються). Складний ключ складається з двох і більше атрибутів. Для зв'язків відношень друг з другом в базі даних є зовнішні ключі. Атрибут або комбінація атрибута відношення є зовнішнім ключем, якщо він не є основним (первинним) ключем цього відношення, але являється первинним ключем для іншого відношення.

На основі аналізу переліку даних, що будуть зберігатися в базі даних була розроблена структура, що складається з 8 основних таблиць. БД була розроблена таким чином, щоб її структура майже повністю співпадала на сервері та в мобільному додатку за винятком окремих полей однієї з таблиць (таблиця Transport). Розроблена структура БД наведена в таблицях 2.4–2.11, а їх взаємозв'язок на рис.2.7.

Розглянемо більш детально таблицю 2.4 та дані, що там будуть зберігатися.

Стовбець id зберігає унікальний цілочисельний ідентифікатор запису.

Стовбець login зберігає ім'я користувача довжиною до 50 символів (бажано, для запобігання не правильному кодуванню, задавати значення латинськими буквами, можливо з використанням чисел).

Таблиця 2.4 – Users– користувачі

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	id	унікальний ідентифікатор запису	INT	
2	login	Ім'я користувача	VARCHAR(50)	
3	passhash	хеш пароля	CHAR(32)	
4	Accesslvl	рівень доступу	INT	

Стовбець passhash зберігає хеш паролю користувача. Використовується саме хеш, а не пароль у чистому вигляді, оскільки це дає змогу захистити пароль від прямого перегляду.

Паролі користувачів саме хешируються, а не шифруються з наступних причин:

а) трудомісткість. Шифрування може тривати довше, а яке перетворення ми б не вибрали, його доведеться проробляти при кожній перевірці пароля. Однією з вимог до хеш-функцій ж є швидкість виконання;

б) довжина вихідних значень. Результат шифрування має змінну довжину, результат хешування – завжди однаково, а зберігати однорідні за розміром дані в базі даних дуже вже зручно. Не кажучи вже про те, що довжина пароля в зашифрованому вигляді буде давати деяку інформацію про довжину вихідного пароля. Однакова довжина, правда, призводить до можливості виникнення колізій, але про це нижче;

в) управління ключами. Для шифрування потрібно ключ, який теж десь доведеться зберігати і сподіватися, що його ніхто не знайде. У будь-якому випадку, генерація і управління ключами це окрема історія (вони не повинні бути слабкими, їх потрібно регулярно міняти і так далі).

Hash = хеш функція – функція однозначного відображення рядка (будь-якої довжини) на кінцеву множину (рядок заданої довжини).

Саме число (рядок) хеш – результат обчислення хеш-функції над даними.

Існують криптографічні та некриптографічні (класифікуються окремо, до них відносяться, наприклад, контрольні суми) хеш-функції.

Стовбець Accesslvl зберігає рівень доступу користувача. Рівні доступу діляться на:

- адміністратор (код 1), який має можливості редагувати налаштування (у першу чергу додавати зупинки, маршрути та транспортні засоби);
- користувач (код 0), що має можливість лише використовувати додаток для пошуку цікавого йому маршруту.

Таблиця 2.5 – Stations– зупинки

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	Id	унікальний ідентифікатор запису	INT	
2	Name	Назва зупинки	VARCHAR(200)	
3	Latitude	широта	FLOAT	
4	Longitude	довгота	FLOAT	

Розглянемо більш детально таблицю 2.5 та дані, що там будуть зберігатися.

Ця таблиця зберігає дані стосовно зупинок та їх координат.

Стовпець id також унікальний ідентифікатор запису як і у всіх таблицях.

Стовбець Name – зберігає назву запинки громадського транспорту.

Стовбець Latitude – зберігає широту (координату зупинки на карті) та має тип FLOAT оскільки це число з комою в залежності від точності вказання місцевості. Чим більше знаків мічля коми, тем більше точність і менше радіус розбросу по території.

Стовбець Longitude – зберігає довготу (другу координату зупинки на карті)

Наприклад приблизні координати університету: 47.8180, 35.1821.

Таблиця 2.6 – Routes– маршрут

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	id	унікальний ідентифікатор запису	INT	
2	Name	Назва маршруту	VARCHAR(10)	
3	TransType	тип транспорту	INT	TransTypes.id

Розглянемо більш детально таблицю 2.6 та дані, що там будуть зберігатися.

Стовбець Name зберігає назву маршруту (у більшості випадків це буде номер, однак часто може бути номер з літерою чи просто літера) тому для зберігання цих даних були обрано тип VARCHAR, оскільки саме цей тип може зберігати символні значення змінно довжини.

Стовбець TransType зберігає тип транспорту, що ходить по даному маршруту. Це поле є посиланням на таблицю TransTypes у вигляді числового значення ідентифікатора тієї таблиці. Це зроблено для того, щоб зменшити об'єм даних, що зберігається в БД шляхом індексації записів. Назви типів

транспортних засобів будуть часто збігатися на різних маршрутах тому не раціонально їх прописувати в текстовому вигляді.

Можлива ситуація коли назва маршрутів різних типів транспортних засобів будуть однаковим, однак при цьому значення стовпця TransType буде відрізнятися, ну і звісно буде розрізнятися унікальний ідентифікатор id.

Таблиця 2.7 – RoutesStations–відповідність зупинок маршрутам

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	Id	унікальний ідентифікатор запису	INT	
2	RouteLink	Посилання на маршрут	INT	Routes.id
3	StationLink	Посилання на зупинку	INT	Stations.id
4	StationNumb	номер зупинки на шляху прямування	INT	

Розглянемо більш детально таблицю 2.7 та дані, що там будуть зберігатися. Ця таблиця використовується для того щоб можна було побудувати лінію окремого маршруту на карті.

Стовбець RouteLink зберігає посилання на маршрут у таблиці Routes.

Стовбець StationLink зберігає посилання на зупинку у таблиці Stations.

Стовбець StationNumb зберігає номер зупинки StationLink для маршруту StationLink на шляху прямування починаючи з однієї з кінцевих станцій (будь якої обраної адміністратором). У випадку коли маршрут їздить циклічно («по колу») та не має умовно кінцевої станції – задавати можна будь яку зупинку або

зупинку на якій маршрут вїзджає після відпочинку водія (якщо для всіх водіїв ця точка однакова

Таблиця 2.8 – Transport – окремі транспортні засоби

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	Id	унікальний ідентифікатор запису	INT	
2	Number	Державний номер транспорту	VARCHAR(10)	
3	TransType	тип транспорту	INT	TransTypes.id
4	TransMark	модель транспорту	INT	Marks.id
5	IsActive	Знаходження на маршруті	BOOL	
6	RouteNow	Поточний маршрут по якому їздить транспортний засіб	INT	Routes.id
7	EquipId*	Унікальний номер обладнання для передачі даних	LONG	
8	Latitude*	Широта місцезнаходження	FLOAT	
9	Longitude*	Довгота місцезнаходження	FLOAT	
10	PassegersCnt*	Кількість пасажирів на цей момент в салоні	INT	

Розглянемо більш детально таблицю 2.8 та дані, що там будуть зберігатися. В цій таблиці зберігаються дані стосовно окремих автобусів, трамваїв, тролейбусів, маршруток і т.д.

Стовбець Number зберігає державний номер реєстрації автомобіля або номер тролейбуса чи трамвая, що найчастіше написаний спереду та збоку таких транспортних засобів.

Стовбець TransType зберігає посилання та тип транспорту в окремій таблиці (1 – автобус, 2 – тролейбус, 3 – трамвай, 4 – маршрутне таксі).

Стовбець TransMark зберігає посилання на таблицю з маркою транспортного засобу.

Стовбець IsActive зберігає поточний стан транспорту на маршруті (присутній на маршруті чи відсутній).

Стовбець RouteNow зберігає посилання на поточний маршрут на якому зараз знаходиться цей транспортний засіб.

Стовбець EquipId зберігає унікальний апаратний ідентифікаційний номер обладнання, що встановлено на транспортний засіб.

Стовбець Latitude зберігає поточну широту розташування транспортного засобу.

Стовбець Longitude зберігає поточну довготу розташування транспортного засобу.

Стовбець PassengersCnt зберігає поточну кількість пасажирів у транспортному засобі.

Таблиця 2.9 – TransTypes– типи транспортних засобів

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	Id	унікальний ідентифікатор запису	INT	
2	Name	назва типу транспортного засобу (автобус, тролейбус, трамвай)	VARCHAR(50)	

Розглянемо більш детально таблицю 2.9 та дані, що там будуть зберігатися.

В цій таблиці зберігаються типи транспортних засобів з їх назвами які зберігаються у стовбці Name, на цю таблицю посилається таблиця окремих транспортних засобів.

Розглянемо більш детально таблицю 2.10 та дані, що там будуть зберігатися.

В цій таблиці зберігаються типи транспортних засобів з їх назвами які зберігаються у стовбці Name, на цю таблицю посилається таблиця окремих транспортних засобів.

Таблиця 2.10 – Marks– моделі транспортних засобів

№	Назва стовпця	Що зберігається	Тип даних	На які таблиці посилається
1	id	унікальний ідентифікатор запису	INT	
2	MarkName	назва моделі	VARCHAR(100)	

Логічна структура бази даних повинна бути спроектована і описана для конкретної СУБД. Для нашої бази даних була обрана СУБД MySQL Server. Необхідно перейти від ER–моделі до таблиць в базі даних.

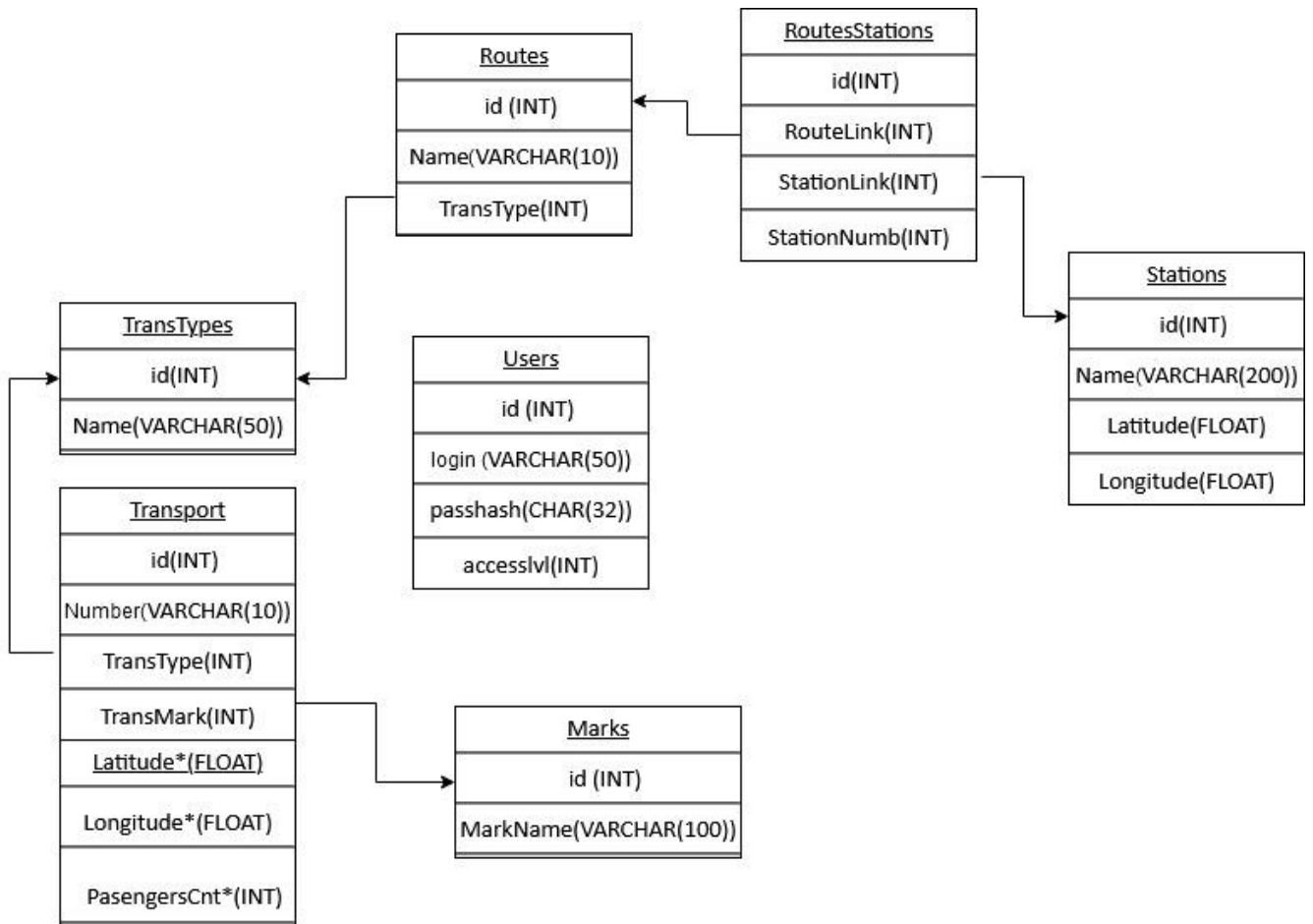


Рисунок 2.7 – Взаємозв'язок таблиц розробленої БД

При проектуванні фізичної моделі бази даних необхідно описати середу і метод зберігання інформації. Для цього необхідно вивчити особливості організації даних обраної СУБД. Для зберігання даних в цій СУБД використовуються таблиці. У них зберігається вся інформація про предметну область. Наша база даних включає кілька взаємопов'язаних таблиць. Об'єкти, які були описані при побудові інфологічної моделі предметної області, в базі даних є таблицями.

3 НАПИСАННЯ ПРОГРАМНОГО КОДУ ТА ВІДЛАГОДЖЕННЯ ПРОГРАМИ

3.1 Блок схема алгоритму мобільного додатка

Розроблений загальний алгоритм роботи програми мобільного додатку наведено на рис.3.1.

Після запуску TransZip, перед користувачем з'являється перше вікно програми з авторизацією. Користувач повинен ввести свій логін та пароль, або якщо він не має їх, пройти швидко реєстрацію. Якщо, логін чи пароль зареєстрованим користувачем був введений не вірно, програма видасть вікно з повідомленням, що поле «Логін» або «Пароль» введено не вірно.

Якщо ж, вхід був виконаний без помилок, користувач побачить на своєму пристрої карту з маршрутами та матиме змогу взаємодіяти з кнопками налаштування, маршрут, планування поїздки, знайти, вихід з аккаунту.

Якщо, натиснута кнопка «Маршрут» користувач має змогу ввести параметри маршруту, обрати початкову та кінцеву точку маршруту (у якості початкової точки може бути використана місцезнаходження користувача), після цього програма отримує координати користувача, шукає та будує обраний маршрут, отримує інформацію з сервера з місцем знаходження транспортних засобів та кількістю пасажирів у них. Користувач має змогу бачити усю отриману інформацію програмою на екрані свого смартфона та у будь який момент змінювати налаштування.

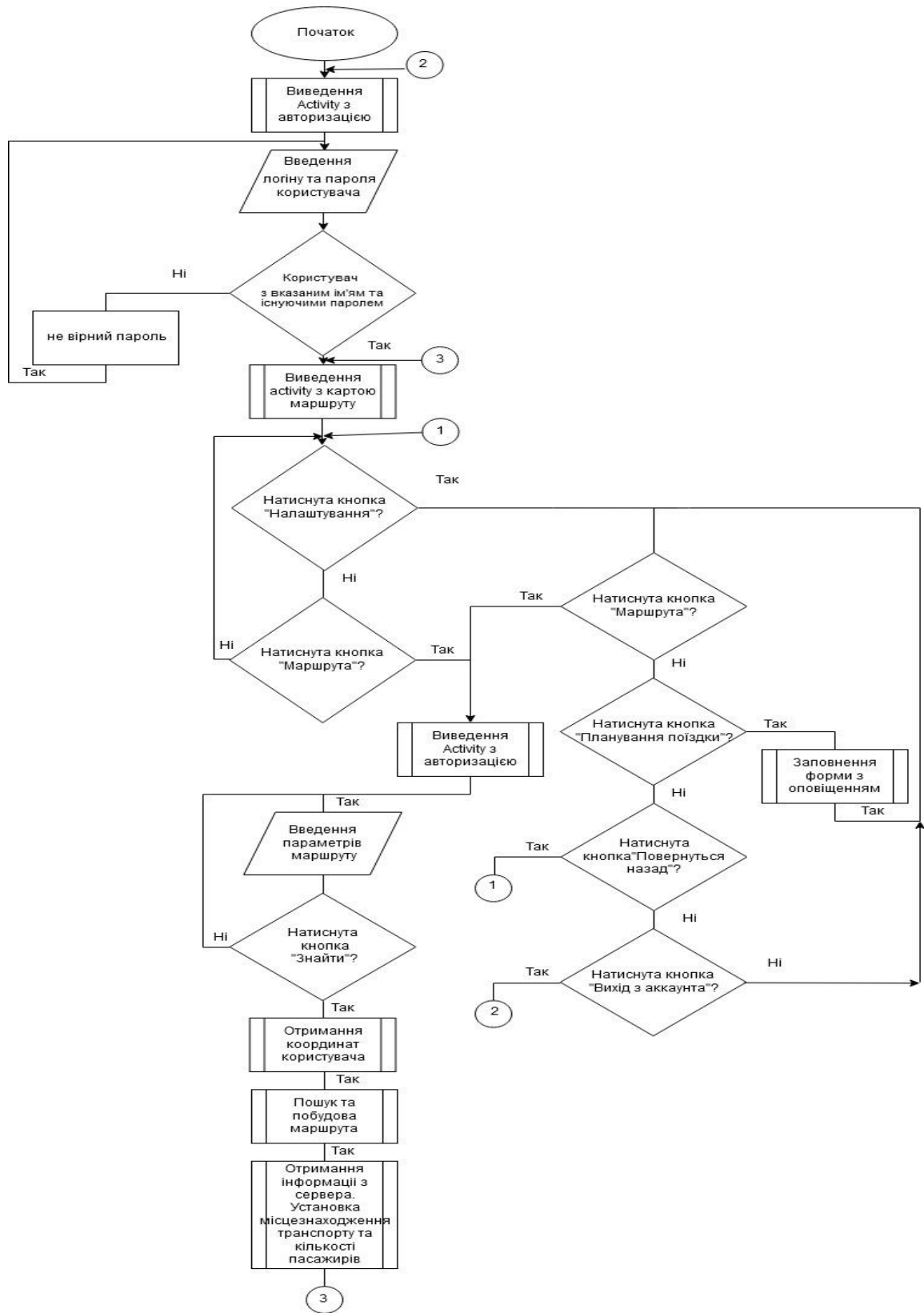


Рисунок 3.1 – Блок схема алгоритму роботи мобільного додатку

Кнопка «Планування поїздки» дозволяє користувачу обрати цікавлячий його маршрут та зробити повідомлення, коли цей маршрут стане менш завантажений пасажиром а також. Для цього, користувач повинен заповнити спеціальну форму у програмі та зберегти її.

3.2 Розробка інтерфейсу додатка

Додатки для Android складаються з компонентів, який система може запускати і керувати так, як їй необхідно. Для цього система повинна бути в змозі запуснути процес для додатка, в якому знаходяться необхідні компоненти, і форматувати потрібні їй об'єкти. Одним з компонентів Android–додатки є діяльність (activity).

Activity є візуальний інтерфейс (окремий екран) для одного виду діяльності, яке користувач може зробити.

Додаток може складатися з одного activity або з декількох. Це залежить від типу програми і його дизайну. Одне activity може викликати інше.

Кожне activity задає вікно для відображення, яке, зазвичай, займає весь екран, але може бути менше і плавати поверх інших вікон. Activity може використовувати додаткові вікна, наприклад, що спливає діалог, який вимагає проміжного відповіді користувача, або вікно, яке відображає користувачам важливу інформацію при виборі елемента, який заслуговує на особливу увагу.

Візуальний інтерфейс будується на основі ієрархії візуальних компонентів – об'єктів, похідних від базового класу View. Android має ряд готових до використання компонентів, включаючи кнопки, текстові поля, смуги прокрутки, меню, прапорці та багато іншого.

На рис.3.1 представлена структура діяльностей проектного додатку. LoginActivity – перше activity, яке відображається під час запуску програми.



Рисунок 3.2 – Зовнішній вигляд розробленої LoginActivity

Цей екран включає в себе наступні елементи інтерфейсу:

- зображення для відображення логотипу програми;
- текстове поле для введення імені користувача;
- текстове поле для введення паролю користувача;
- кнопка входу;
- кнопка реєстрації.

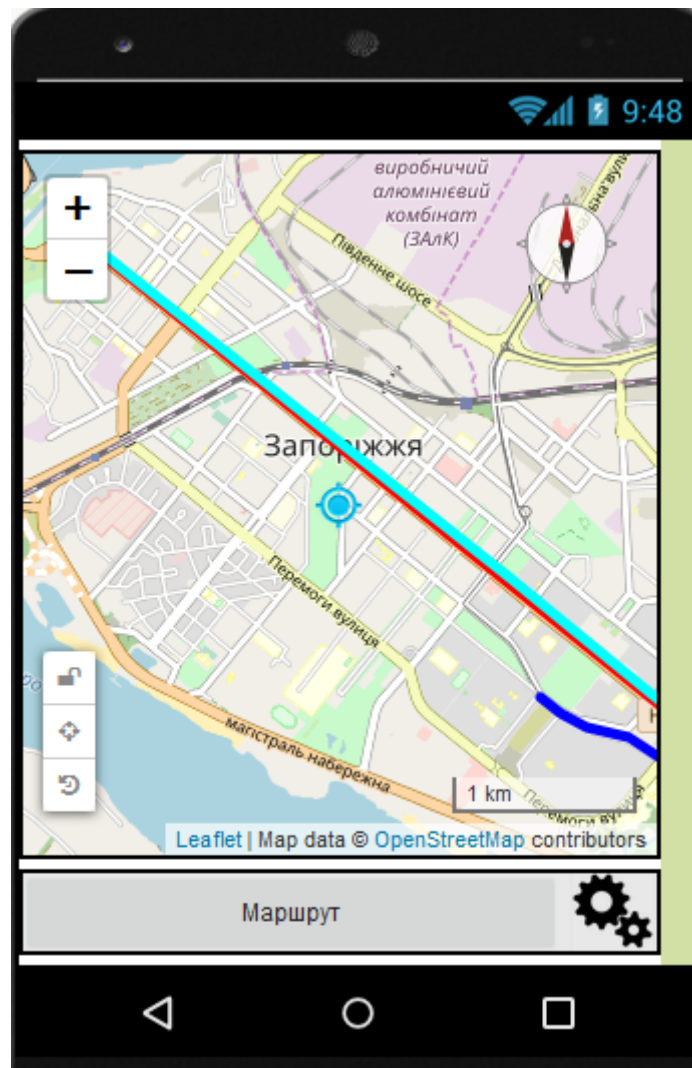


Рисунок 3.3 – Зовнішній вигляд розробленої MapActivity

Наступним екраном є основний екран мобільного додатку, що відображає карту місцевості (міста Запоріжжя) на якій відображаються місцезнаходження транспортних засобів та маршрутів за запитом користувача.

Екран складається з наступних елементів інтерфейсу:

- область висвітлення карти;
- кнопка переходу на екран введення параметрів маршруту;
- кнопка переходу на екран налаштувань.

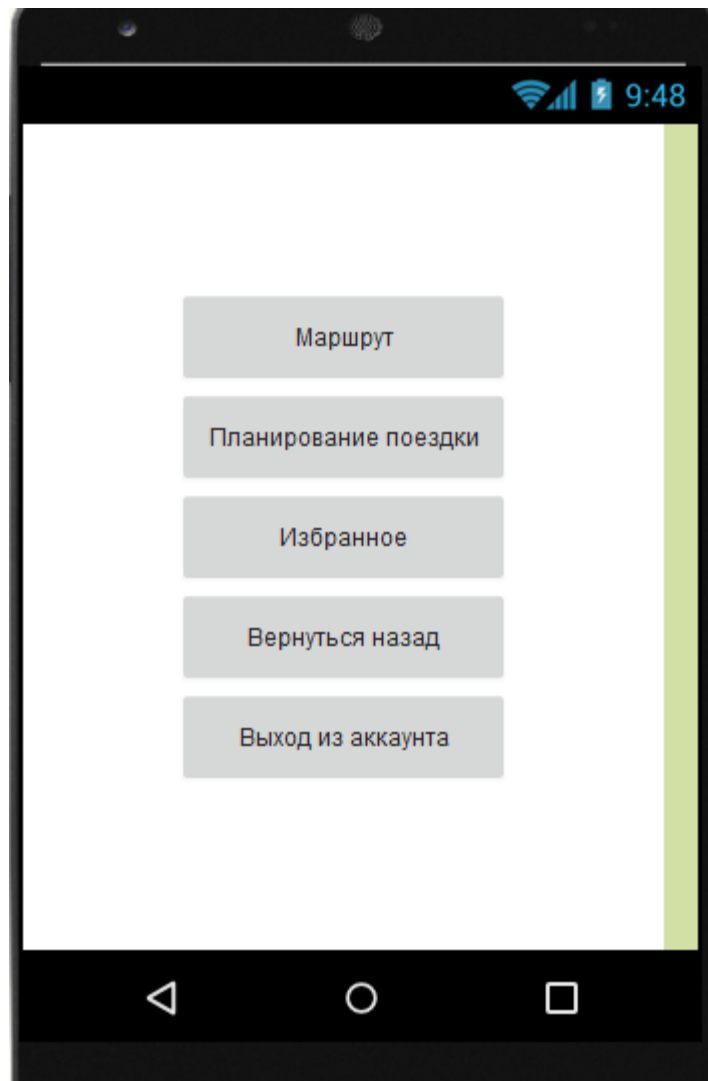


Рисунок 3.4 – Зовнішній вигляд розробленої SetupActivity

Наступним екраном мобільного додатку є екран з налаштуваннями який представляє собою меню виконане за допомогою п'яти кнопок.

Меню включає наступні пункти:

- маршрут (введення параметрів маршруту для пошуку транспортних засобів, що їдуть з початкової точки в кінцеву);
- планування поїздки (нагадування користувачеві вийти в визначений час, щоб встигнути на певний маршрут та доїхати до визначеної точки);
- обране (обрані маршрути по яких часто їздить користувач);
- повернутися назад (кнопка для повернення на основний екран);

– вихід з аккаунта.

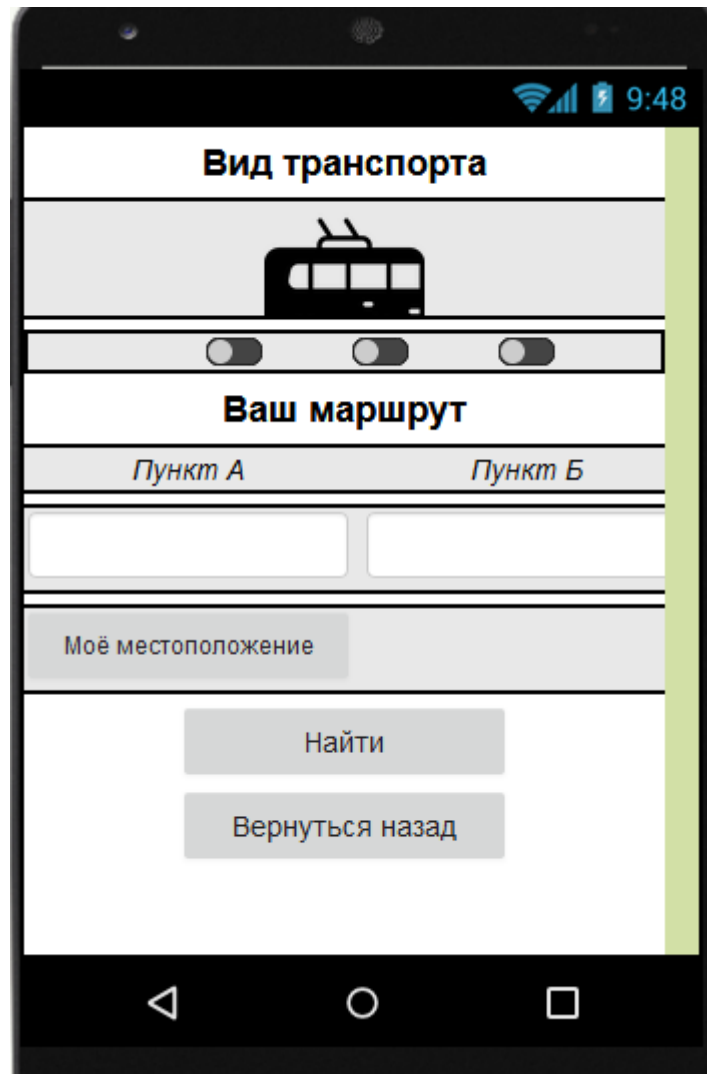


Рисунок 3.5 – Зовнішній вигляд розробленої RouteActivity

Наступним екраном є екран введення параметрів маршруту поїздки. Тут користувач може ввести початкову точку маршруту або замість цього вказати своє місцезнаходження, вказати кінцеву точку маршруту та обрати тип транспорту яким користувач бажає доїхати.

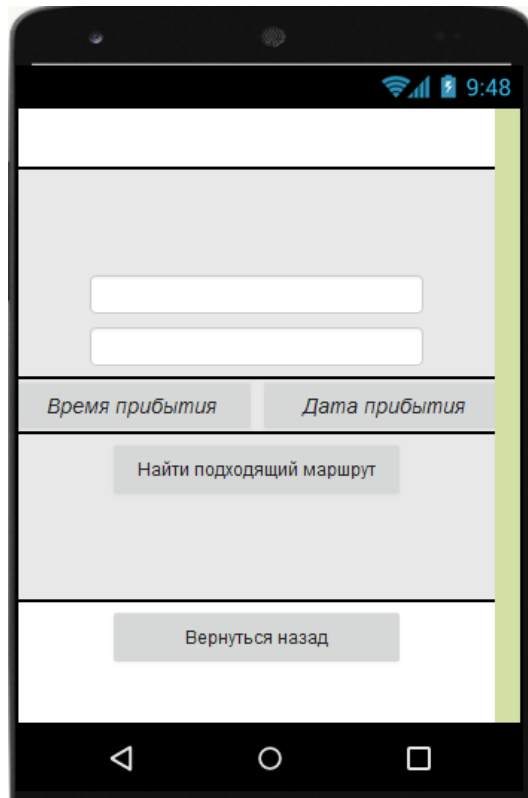


Рисунок 3.6 – Зовнішній вигляд розробленої PlanActivity

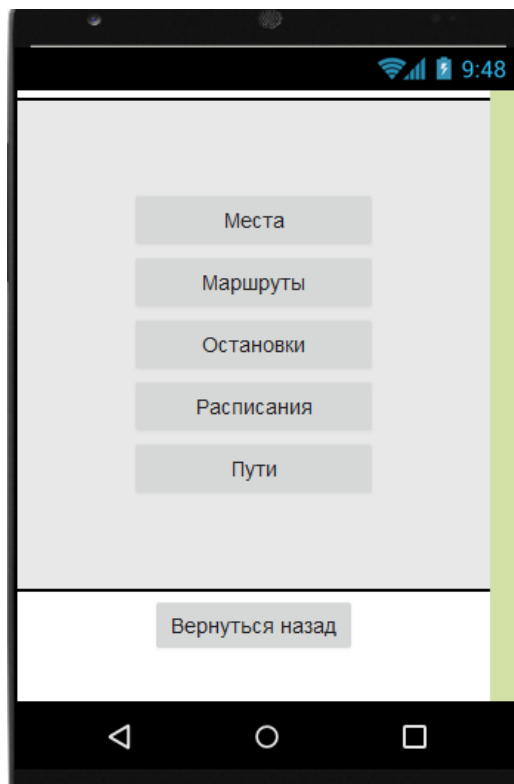


Рисунок 3.7 – Зовнішній вигляд розробленої InfoActivity

3.3 Написання скрипта для збору даних

Перед написанням тексту скрипту обробки даних отриманих з транспортних засобів та запитів на отримання інформації з мобільних пристроїв користувачів першим етапом треба розробити ці запити та скласти відповідні запити до бази даних для збереження та отримання інформації.

Оскільки для з'єднання з інтернетом буде використовуватись GPRS канал, що має невелику пропускну здатність, а також в основі лічильника пасажирів лежить мікроконтролер бажано запити робити не складними з фіксованою довжиною та різними стартовими символами. Розроблені запити, що будуть відправлятися на сервер винесено у табл.3.1

Всі запити відправляються методом “POST”. Основними запитами звісно є запити щодо місцезнаходження та кількості пасажирів у солоні. Однак є додаткові запити – наприклад коли транспортний засіб змінює маршрут за яким він їде, а також у випадках коли транспортний засіб сходить з маршруту з технічних причин чи по завершенні зміни та потім повертається назад.

Також для написання скрипту на сервері та мобільного додатку треба розробити форми запитів з боку мобільного додатку.

Основними даними, які буде запитувати мобільний додаток це дані щодо найближчих маршрутів, що одночасно виконують вимогу кінцевої точки куди користувачеві треба потрапити, запити що до місцезнаходження окремого маршруту, запити щодо кількості пасажирів в окремому транспортному засобі.

Розроблені запити з боку мобільного додатку зведено у табл.. 3.2.

Скрипти для роботи з апаратною частиною та мобільним додатком було прийнято рішення рознести у різні файли відповідно Transport.php та Mobile.php.

Таблиця 3.1 – Опис розробленої структури запитів на сервер з боку апаратної частини, що розміщується в транспортному засобі

№	Запит	Приклад	Опис
1	Id=XXXXXX &Cmd=1 &Latitude=YY.YYYYYYY &Longitude=ZZ.ZZZZZZ	Id=00124 &Cmd=1 &Latitude=47.818068 &Longitude=35.182136	Запит про поточне місцезнаходження транспортного засобу з унікальним ідентифікатором “id”
2	Id=XXXXXX &Cmd=2 &Route="NAME" &Trtype=Y	Id=00124 &Cmd=2 &Route="39" &Trtype=1	Запит про поточний номер маршруту транспортного засобу з унікальним ідентифікатором “id”
3	Id=XXXXXX &Cmd=3 &Pcount=YYY	Id=00124 &Cmd=3 &Pcount=037	Запит про кількість пасажирів, що зараз знаходяться у салоні транспортного засобу
4	Id=XXXXXX &Cmd=4 &Status=0	Id=00124 &Cmd=4 &Status=0	Запит про з'їзд з маршруту
5	Id=XXXXXX &Cmd=4 &Status=1	Id=00124 &Cmd=4 &Status=1	Запит про повернення на маршрут

Таблиця 3.2 – Опис розробленої структури запитів на сервер з боку мобільного додатку

№	Запит	Опис
1	Id=DDDDD &Authcode=FFFFFFFF &Slatitude=YY.YYYYYYY &Slongitude= ZZ.ZZZZZZ &Slatitude=NN.NNNNNN &Slongitude=MM.MMMMMM Trtype=T	Запитати JSON масив маршрутів, що проходять близько координат YY.YYYYYYY та ZZ.ZZZZZZ початкової точки та координат NN.NNNNNN та MM.MMMMMM кінцевої точки з типом транспорту Trtype або за всіма видами якщо Trtype=0
2	Id=DDDDD &Authcode=FFFFFFFF &Mark="MNAME" & Trtype=T	Запитати JSON масив координат всіх транспортних засобів маршруту Mark з типом транспорту Trtype або за всіма видами якщо Trtype=0
3	Id=DDDDD &Authcode=FFFFFFFF &Tid=NNNNN	Запитати JSON масив кількості пасажирів в транспортному засобі та інших його характеристик з ідентифікатором Tid

3.4 Розробка та відлагодження роботи лічильника пасажирів

Ми розглянули існуючі аналоги даної системи та було проаналізовані недоліки та переваги лічильників пасажирів.

Лічильник пасажирів – це пристрій, здатний автоматично підрахувати кількість пасажирів, які увійшли в транспортний засіб та вийшли з транспортного засобу. Ця технологія використовується для визначення кількості пасажирів, динаміки відвідувань, частоти. Роль цих технологій

зосереджена навколо вимірювання поведінки відвідувача, дозволяє визначити і аналізувати поведінкові метрики, розраховувати відсоток конверсії, оцінювати активності

Розглянемо найпопулярніші технології, які на сьогоднішній день використовуються для підрахунку і трекінгу пасажирів:

- інфрачервоний (ІЧ) лічильник;
- термолічильники;
- підрахунок на основі звичайних відеокамер;
- 2D відеосчетчікі;
- 3D відеосчетчікі.

Інфрачервоний (ІЧ) лічильник пасажирів заснований на принципі переривання ІК–променів (рис.3.8). Передавач і приймач кріпляться на стінки проходу один навпроти одного, в результаті чого всі хто заходять в транспорт або виходять з нього перетинають два інфрачервоних променя.

Устаткування даного типу підрозділяється на провідні і бездротові ІК лічильники.

Для роботи провідних лічильників потрібно протягання кабелю до місця установки лічильника. Бездротові лічильники працюють від батарейок, як правило, є автономними, але в них відсутня будь-яка аналітика, і по суті якість підрахунку і аналітики можна порівняти з якістю ручного підрахунку відвідувачів. Переваги та недоліки цього типу лічильників наведені у табл.3.3.



Рисунок 3.8 – Принцип роботи інфрачервоного (ІЧ) лічильника

Таблиця 3.3 – Переваги та недоліки інфрачервоного (ІЧ) лічильника

Переваги	Недоліки
Низька вартість	Обмеження по ширині проходу
Малопомітність	Низька вандалозахищеність
Працюють незалежно від змін температури, вологості, освітленості	Високий ступінь впливу «людського фактора» на рівень похибки

Робота цих термолічильників заснована на принципі зміни температурного фону. Устаткування «сприймає» простір як певний теплової фон, а людей – як «теплові плями» на цьому тлі. Встановлюється на стелі (рис.3.9). Переваги та недоліки цього типу лічильників наведені у табл.3.4.

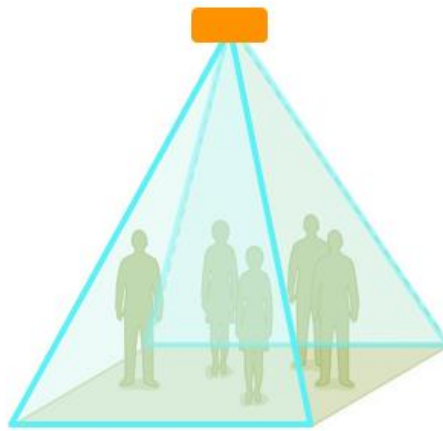


Рисунок 3.9 – Принцип роботи термолічильника

Підрахунок на основі звичайних відеокамер. Підрахунок людей виробляється шляхом обробки відеосигналу – люди сприймаються пристроєм як «світлові плями», що відрізняються від загального світового фону. Відеопотік відправляється на сервер, де і відбувається обробка зображення. Для того, щоб

добитися максимальної точності, камера повинна бути встановлена на стелі над проходом і спрямована вниз (рис.3.10). Переваги та недоліки цього типу лічильників наведені у табл.3.5.

Таблиця 3.4 – Переваги та недоліки термолічильника

Переваги	Недоліки
Висока вандалозахищеність	Примхлива при роботі на входах з вулиць і в ін. Місцях з мінливим тепловим фоном (не підходить для використання в умовах російської погоди)
	Точність підрахунку знижується зі збільшенням трафіку відвідувачів, або з наявністю скупчення людей під лічильником.
	Порівняно висока вартість



Рисунок 3.10 – Принцип роботи підрахунку на основі звичайних відеокамер

Таблиця 3.5 – Порівняльна таблиця підрахунків на основі звичайних відеокамер.

Переваги	Недоліки
<p>Порівняно недорогі витрати на установку (іноді можуть використовувати вже існуючі камери)</p>	<p>Навантаження на інтернет канали та сервера</p> <p>Багатофункціональність не дає переваги, тому що для кожної функції потрібно використовувати окрему камеру.</p>
<p>Можливості використання звичайних відеокамер, на які встановлюється спеціальна прошивка, що дозволяє робити підрахунок</p>	<p>Похибка підрахунку нестабільна за рахунок: Відсутності алгоритмів дозволяють робити підрахунок цільового трафіку (зайшов, але не пройшов, ходить по лінії підрахунку)</p> <p>Принципу підрахунку – якщо фігура в камері більше заданих параметрів – людина не буде пораховано, якщо менше (дитина) – буде врахований як платоспроможний відвідувач</p>
<p>Потенційна багатофункціональність: підрахунок + відеоспостереження + розпізнавання осіб і т.д.</p>	<p>Вплив сонячних променів і щільності трафіку на точність підрахунку</p> <p>На одній входній групі потрібно більше датчиків звичайного: над кожними антискражних воротами по камері</p>
	<p>Немає можливості об'єднувати камери в єдиний прохід (мультисенсорінг)</p>

2D відеолічильник. Технологія відеоаналізу, заснована на обробці відео з кольорового монокулярного сенсора, створена спеціально для цілей підрахунку кількості людей у приміщеннях (рис.3.11). Переваги та недоліки цього типу лічильників наведені у табл.3.6.

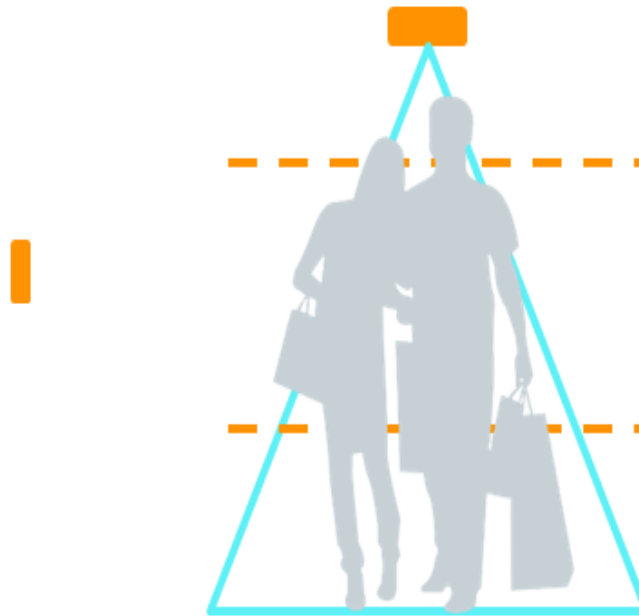


Рисунок 3.11 – Принцип роботи підрахунку на основі 2D відеолічильника

Таблиця 3.6 – Порівняльна таблиця підрахунок на основі 2D відеолічильника

Переваги	Недоліки
Порівняно низька вартість	Вплив сонячних променів на точність підрахунку (не рекомендується установка на входах з вулиці)

Продовження таблиці 3.6

<p>Наявність алгоритмів підрахунку дозволяють підвищити якість підрахунку: підрахунок людей, що йдуть в різних напрямках, виключення з підрахунку охоронців, людей, які увійшли але не пройшли в торгову зону (функціонал доступний лише на пристроях Brickstream)</p>	<p>На точність підрахунку впливає щільність трафіку відвідувачів</p>
<p>Автоматична боротьба з тінями при сонячній засвітке (функціонал доступний лише на пристроях Brickstream)</p>	
<p>Обробка відео відбувається безпосередньо на пристрої (немає навантаження на інтернет трафік)</p>	

3D Відеолічильник. Контроль здійснюється у форматі 3D, технологія здатна відстежити положення, висоту всіх об'єктів в зоні охоплення, напрямки їх руху, форму і розміри. Один пристрій включає в себе всі функції (збирає, зберігає, сповіщає, передає) – немає необхідності придбання додаткових комп'ютерів або пристроїв збору даних (рис.3.12). Переваги та недоліки цього типу лічильників наведені у табл.3.7.

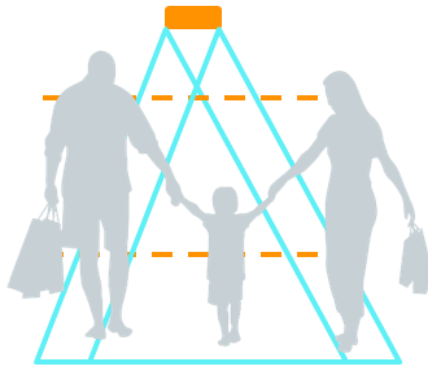


Рисунок 3.12 – Принцип роботи підрахунку на основі 3D відеолічильника

Таблиця 3.7 – Порівняльна таблиця підрахунків на основі 3D відеолічильника

Плюси	Мінуси
Незалежність якості підрахунку від кількості відвідувачів, освітлення, температури і ін. Чинників	Порівняно висока вартість рішення
Широкий функціонал: – окремий підрахунок дорослих і дітей; – виключення персоналу з підрахунку; – виключення увійшли, але не пройшли в торгову зону відвідувачів	
Висока точність підрахунку	
Можливість запису відео	
Один пристрій зберігає, обробляє і передає дані	
Ширина охоплення і висота установки (економія на кількості використовуваних датчиків в одному проході – тільки моделі виробників Xovis і Brickstream)	

Проводячи аналіз розглянутих методів треба зазначити, що оскільки встановлення в транспортних засобах складних комп'ютерних систем дуже незручно із-за їх вартості та складності обслуговування – було прийнято рішення використовувати інфрачервоні лічильники на основі розриву променя при його перетинанні. Для підвищення точності та визначення напрямку руху пасажирів кількість променів робимо рівною п'яти.

На основі отриманих даних було розроблена конструкція пристрою «Лічильник кількості пасажирів» структурна схема з описом елементів наведена на рис.3.13, призначення для відслідкування кількості пасажирів у транспортному засобі. З його допомогою, користувач мобільного додатку матиме змогу отримати інформацію про кількість вільних місць у маршрутному транспортному засобі. А власник матиме змогу отримати повний звіт про кількість перевезених пасажирів за робочий день.

Пристрій складається з наступних електронних компонентів:

- плата Arduino Nano на базі мікроконтролера ATmega328;
- GSM/GPRS модуль SIM800;
- п'яти світлодіодних інфрачервоних модулів KY-005;
- п'яти фотоприймальних модулів для інфрачервоного діапазону VS1838B;
- елемент живлення.

Розроблена конструкція пристрою встановлюється на вході до маршрутного транспортного засобу (на кожні двері) та за допомогою світлодіодних та фотоприймальних модулів фіксує зайшов чи вийшов пасажир. Під час входу або виходу з транспортного засобу пасажир, він розриває зв'язок між світлодіодним та фотоприйомним модулем завдяки чому, програма отримує змінені данні з датчиків та розуміє увійшов чи вийшов пасажир.

Так само ця система працює тільки коли двері в транспорт відкриті, а в іншому випадку вона в режимі очікування. Що дозволяє уникнути проблем при підрахунку пасажирів

Плата управління Лічильник кількості пасажирів призначена для установки і експлуатації на наземному транспорті. Контролер спільно з програмою дозволяє вести облік пасажирів в транспорті. Що дозволяє контролювати кількість людей які знаходяться в громадському транспорті.

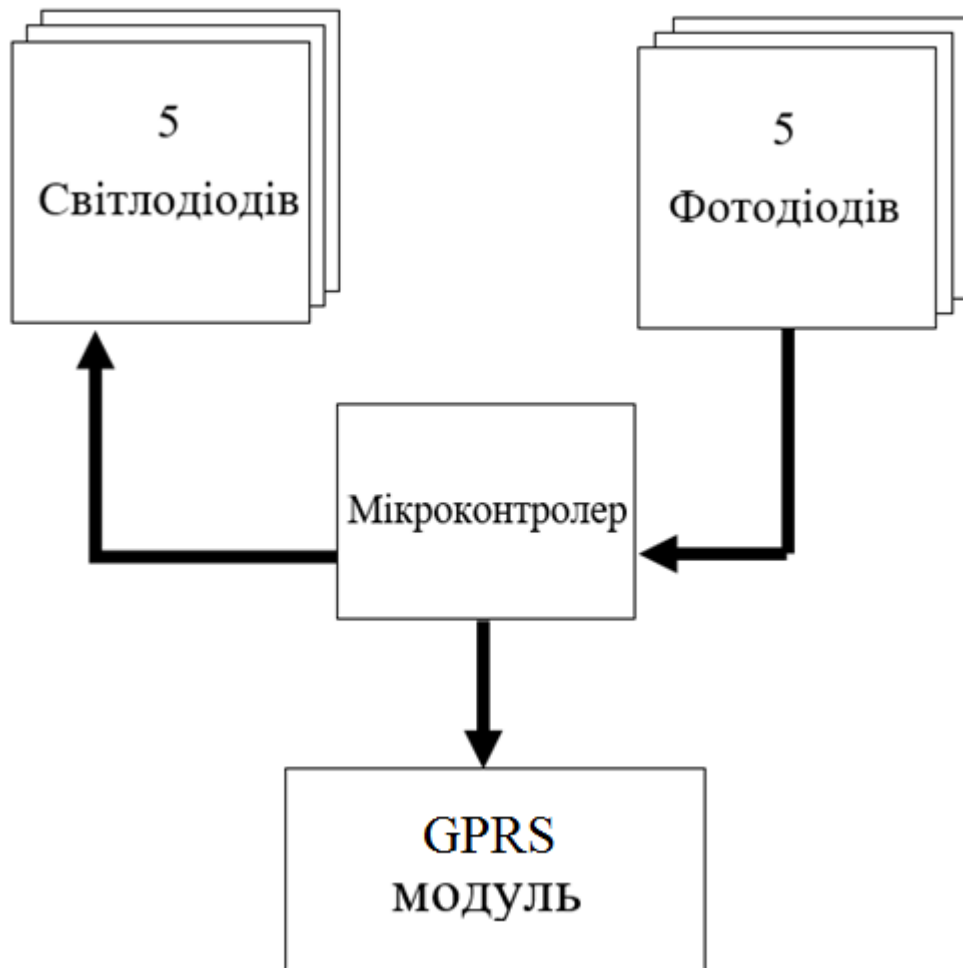


Рисунок.3.13– Структурна схема «Лічильник кількості пасажирів»

Температурний діапазон експлуатації при температурі навколишнього повітря від -40 до $+50$ ° C забезпечує надійну цілорічну роботу в різних кліматичних умовах.

Для з'єднання радіоелементів електричної схеми між собою, в якості базової несучої конструкції вибираємо двосторонню друковану плату виготовлену комбінованим позитивним методом по полуаддитивній технології. З огляду на те, що при проектуванні ПП використовуються ІС, а також високий рівень насиченості ПП навісними елементами по ГОСТ 23751–86 вибираємо четвертий клас точності.

Відповідно до того, що максимальний діаметр виводів навісних елементів, розміщених на платі, дорівнює 0,7 мм, то обираємо товщину плати рівну 1,5 мм.

Як матеріал проекрованої ДПП обираємо склотекстоліт нагрівостійкості вищого сорту, товщиною 1,5 мм, облицьований з двох сторін мідною оксидною фольгою, товщиною 50 мкм

В конструкції даного пристрою застосовуються стандартні ЕРЕ, що мають вологозахисне покриття та низьку інтенсивність відмов, що забезпечує надійну працездатність пристрою протягом гарантованого терміну служби при впливі на нього несприятливих кліматичних факторів. ЕРЕ кріпляться на платі за допомогою пайки.

Номінальні діаметри отворів для кріплення складають 3,4 мм.

Друкована плата була розтрассированна на двох сторонах. Ширина трас з'єднань 0,5 мм. Допуск між провідником і елементом друкованої плати 0,5 мм, допуск між провідником і отвором 0,25 мм, допуск між провідником і краєм друкованої плати 2 мм, допуск відстаней елементів друкованої плати 0,15 мм.

Для перевірки працездатності плати була обрана програма Proteus 8, яка дозволяє проводити аналіз роботи та симуляцію електронних пристроїв на базі мікроконтролерів з можливістю завантаження розробленої керуючої програми (рис. 3.14).

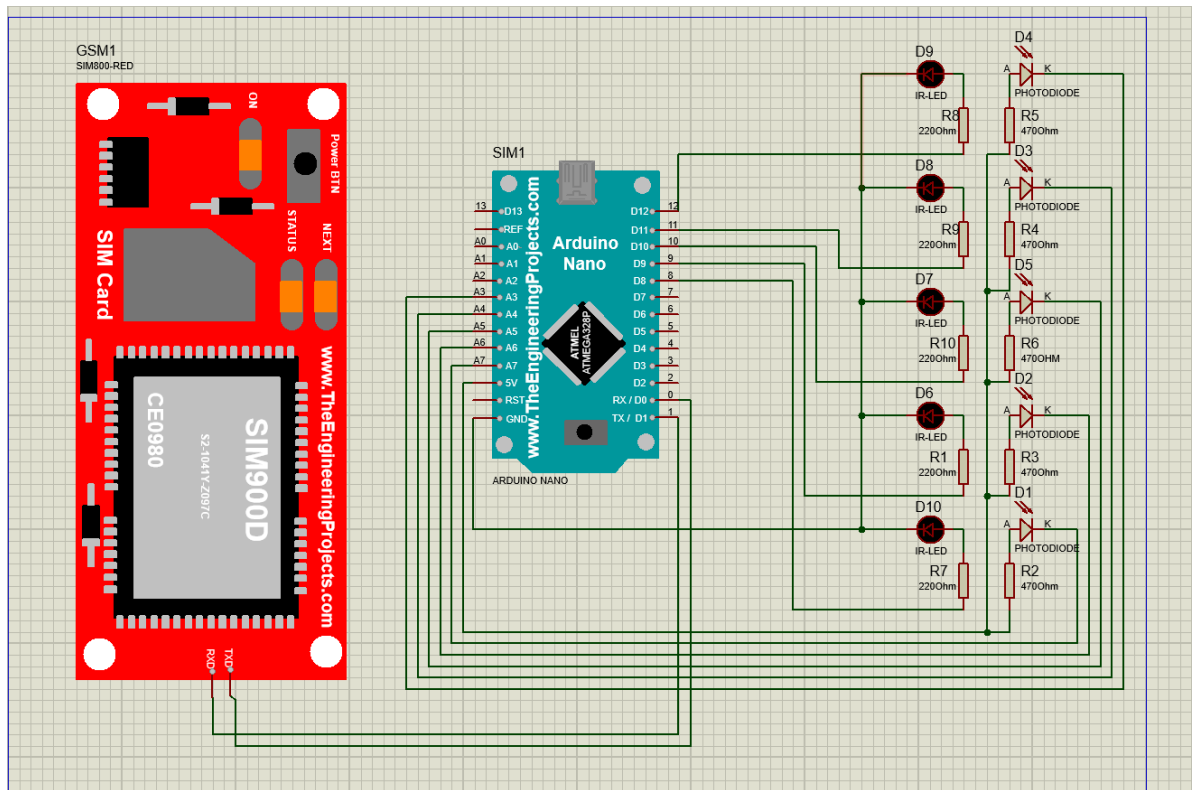


Рисунок.3.14 – Тестування плати керування в Proteus 8.

Плата керування вставляється в корпус виконаний з полістиролу. В якості корпуса обрана базова несуча конструкція, що уявляє собою електричну коробку з габаритами 200x80x75мм.

Для виведення роз'ємів, що служать для приєднання комп'ютеру, а також для роз'ємів живлення, датчиків, в корпусі робляться отвори з розмірами, що відповідають розмірам роз'ємів.

Розробка програми керуючого мікроконтролера проводилася в середовищі розробки Arduino IDE створеної безкоштовно поставляється виробником мікроконтролерів Gravitech (рис.3.15).

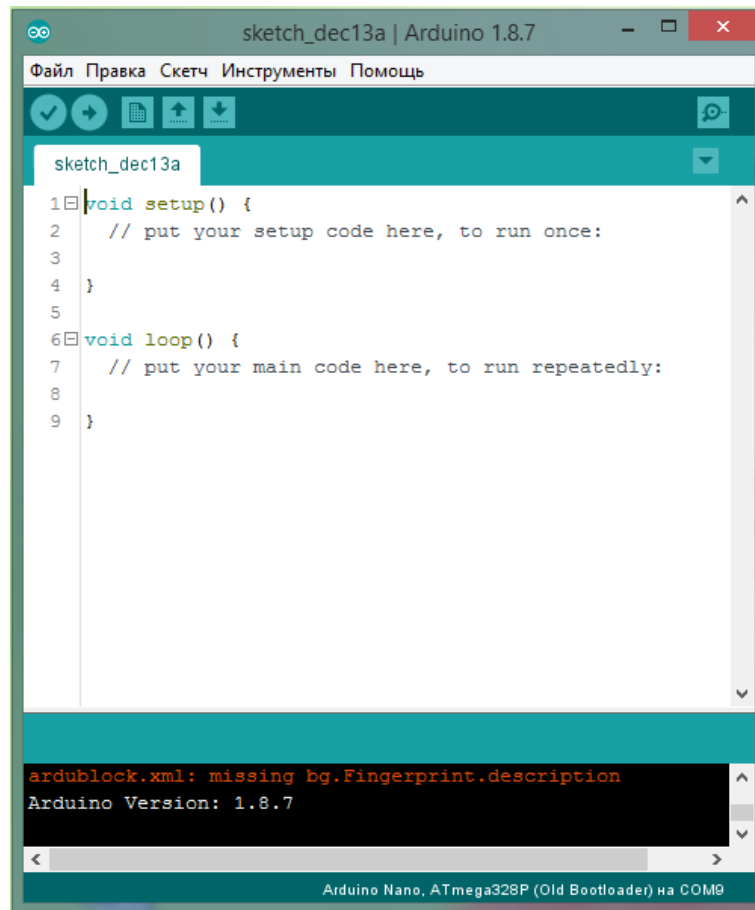


Рис 3.15 – Середовище розробки програм для мікроконтролерів, Arduino IDE

Був створений проект в середовищі розробки структура якого приведена на рис.3.16

Проект включає в себе файли бібліотек з розширенням * .h та *.cpp, Arduino IDE виконує завдання з'єднання файлів підпрограм і бібліотек в єдину «прошивку» для запису коду в пам'ять програм мікросхеми мікроконтролера.


```

26 #include <SoftwareSerial.h>
27 SoftwareSerial mySerial(2, 3);
28 // -- объявляем переменные
29 byte mac[] = {};
30 byte ip[] = {};
31 byte myDns[] = {};
32 byte gateway[] = {};
33 byte subnet[] = {};
34 byte ConTypeRead, setChannelRead, speedRFRead, mySerialSpeedRead, SerialSpeedRead;
35
36 void ReadSetting() {
60
61 void initDevices() {
72
73 void initloadCell() {
78
79 void setup() {
84 void loop() {

```

Рисунок 3.16 – Структура програми керуючого мікроконтролера

3.5 Написання мобільного додатка

Для написання мобільного додатку було використано середовище розробки Android Studio оскільки це спеціальна безкоштовна среда для пристроїв з операційною системою Android (рис.3.17).

Для створення будь якого екрану мобільного додатку розробляється інтерфейс у вигляді XML файлу, однак для спрощення створення екранів в Android Studio присутня палітра компонентів для розміщення елементів перетягуванням (рис.3.18).

Приклад XML коду для компонента ImageView:

```

<ImageView android:layout_width=«fill_parent»
    android:layout_height=«wrap_content»
    android:src="@drawable/droid"
    android:layout_gravity=«center_horizontal|center»
    android:layout_margin=«0px»
/>

```

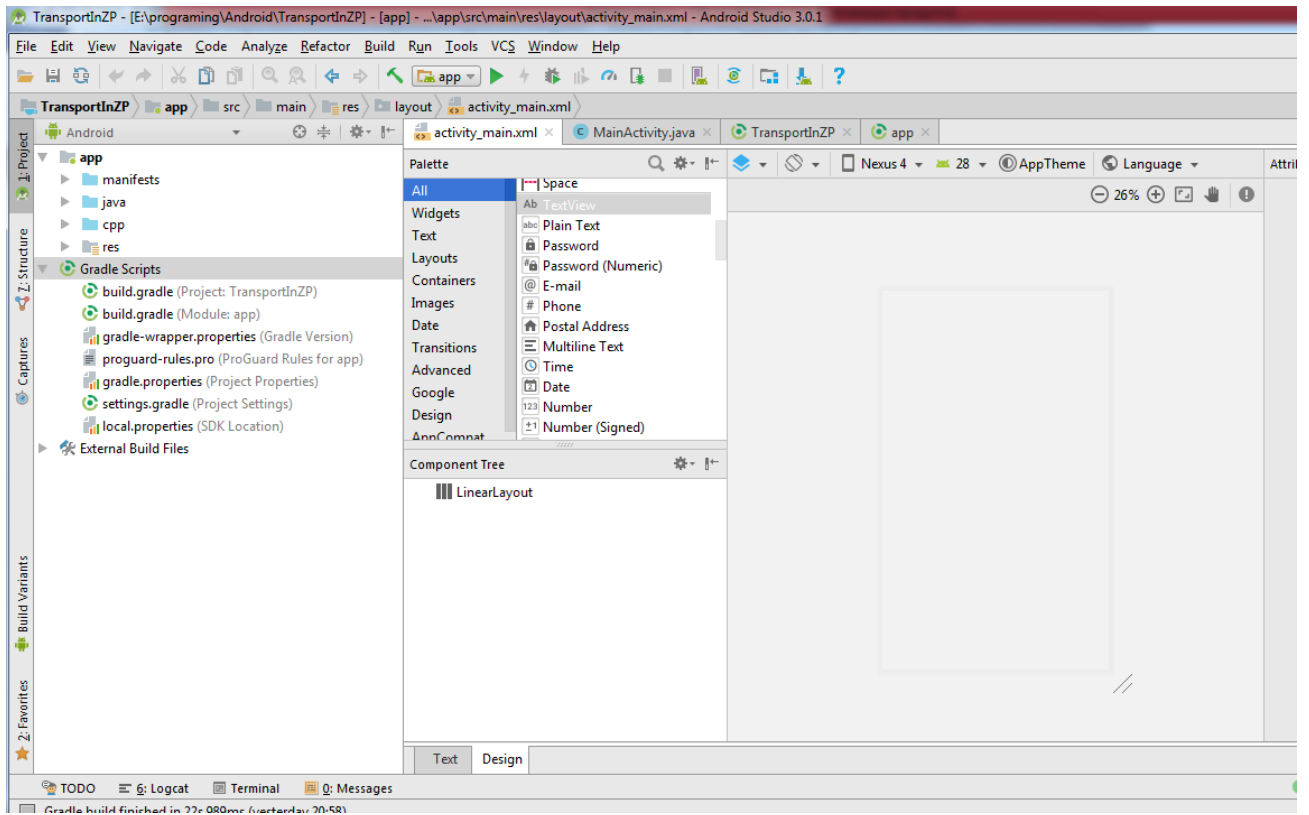


Рисунок 3.17 – Середовище розробки Android Studio

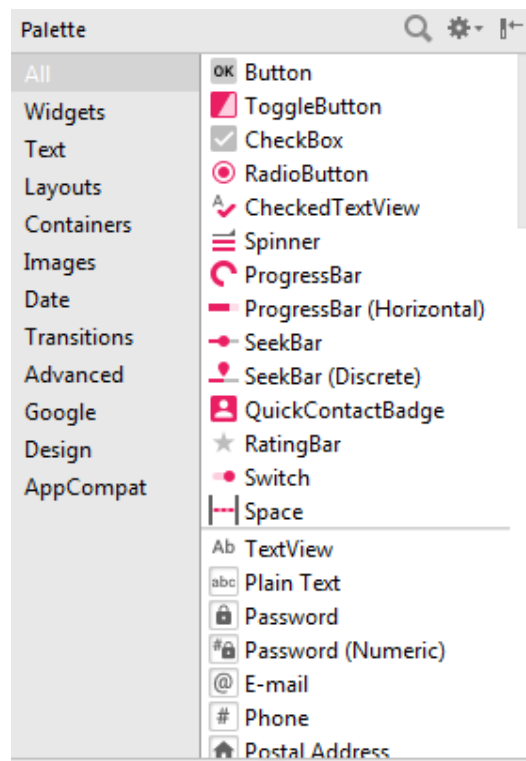


Рисунок 3.18 – Палітра елементів екрану

Основними функціональними елементами мобільного додатку є можливість визначати місцезнаходження користувача, зв'язок з сервером та обробка даних в локальній базі даних.

Для визначення місцезнаходження користувача (мобільного пристрою) використовується об'єкт `LocationManager`.

Основними функціями є наступні:

```
private void showLocation(Location location) {
    if (location == null)
        return;
    if (location.getProvider().equals(LocationManager.GPS_PROVIDER)) {
        tvLocationGPS.setText(formatLocation(location));
    } else if (location.getProvider().equals(
        LocationManager.NETWORK_PROVIDER)) {
        tvLocationNet.setText(formatLocation(location));
    }
}

private String formatLocation(Location location) {
    if (location == null)
        return "";
    return String.format(
        "Coordinates: lat = %1$.4f, lon = %2$.4f, time = %3$tF %3$tT",
        location.getLatitude(), location.getLongitude(), new Date(
            location.getTime()));
}
```

`showLocation` на вхід бере `Location`, визначає його провайдера методом `getProvider` і відображає координати у відповідному текстовому полі яке потім передається на карту.

`formatLocation` на вхід бере `Location`, читає з нього дані і форматує з них рядок. Які дані він бере: `getLatitude` – широта, `getLongitude` – довгота, `getTime` – час визначення.

В методі `onResume` вішаємо слухача за допомогою методу `requestLocationUpdates`. На вхід передаємо:

- тип провайдера: `GPS_PROVIDER` або `NETWORK_PROVIDER`;
- мінімальний час (в мілісекундах) між отриманням даних;
- мінімальна відстань (в метрах). Тобто якщо ваше місце розташування змінилося на вказану кількість метрів, то вам прийдуть нові координати;

– слухач, об'єкт `locationListener`.

```
protected void onResume() {
    super.onResume();
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
        1000 * 10, 10, locationListener);
    locationManager.requestLocationUpdates(
        LocationManager.NETWORK_PROVIDER, 1000 * 10, 10,
        locationListener);
    checkEnabled();
}
```

У `onPause` відключаємо слухача методом `removeUpdates`.

```
protected void onPause() {
    super.onPause();
    locationManager.removeUpdates(locationListener);
}
```

`locationListener` – слухач, реалізує інтерфейс `LocationListener` з методами:

– `onLocationChanged` – нові дані про місцезнаходження, об'єкт `Location`.

Тут ми викликаємо свій метод `showLocation`, який на екрані відобразить дані про місцезнаходження;

– `onProviderDisabled` – вказаний провайдер був відключений користувачем. У цьому методі викликаємо свій метод `checkEnabled`, який на екрані оновить поточні статуси провайдерів;

– `onProviderEnabled` – вказаний провайдер був включений користувачем. Тут також викликаємо `checkEnabled`. Далі методом `getLastKnownLocation` (він може повернути `null`) запитуємо останнім доступне місце розташування від включеного провайдера і відображаємо його. Воно може бути цілком актуальним, якщо ви до цього використовували будь-яка програма з визначенням місця розташування;

– `onStatusChanged` – змінився статус зазначеного провайдера. В поле `status` можуть бути значення `OUT_OF_SERVICE` (дані будуть недоступні довгий час), `TEMPORARILY_UNAVAILABLE` (дані тимчасово недоступні), `AVAILABLE` (все ок, дані доступні). У цьому методі ми просто виводимо новий статус на екран.

Крім того треба прописати дозволи в файлі AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Далі розглянемо роботу з локальною базою даних та відправленням та отриманням даних з сервера.

Для відправки параметрів запитів що наведені в табл.3.2 використовуємо наступні елементи.

Використовуємо HashMap для зберігання параметрів, які повинні бути відправлені на сервер через параметри POST:

```
HashMap<String, String> params;
```

Створюємо StringBuilder, який буде використовуватися для відправки їх на сервер::

```
StringBuilder sbParams = new StringBuilder();
int i = 0;
for (String key : params.keySet()) {
    try {
        if (i != 0){
            sbParams.append("&");
        }
        sbParams.append(key).append("=")
            .append(URLEncoder.encode(params.get(key), "UTF-8"));

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    i++;
}
```

Створюємо HttpURLConnection, відкриваєм з'єднання і відправляємо параметри POST:

```
try{
    String url = "http://www.docsasha.pp.ua/Mobile.php";
    URL urlObj = new URL(url);
    HttpURLConnection conn = (HttpURLConnection) urlObj.openConnection();
    conn.setDoOutput(true);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Accept-Charset", "UTF-8");

    conn.setReadTimeout(10000);
    conn.setConnectTimeout(15000);
```

```

conn.connect();

String paramsString = sbParams.toString();

DataOutputStream wr = new DataOutputStream(conn.getOutputStream());
wr.writeBytes(paramsString);
wr.flush();
wr.close();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

Для отримання результату, який сервер відправляє назад (JSON масив):

```

try {
    InputStream in = new BufferedInputStream(conn.getInputStream());
    BufferedReader reader = new BufferedReader(new InputStreamReader(in));
    StringBuilder result = new StringBuilder();
    String line;
    while ((line = reader.readLine()) != null) {
        result.append(line);
    }

    Log.d("test", "result from server: " + result.toString());

} catch (IOException e) {
    e.printStackTrace();
} finally {
    if (conn != null) {
        conn.disconnect();
    }
}
}

```

Загальна логіка роботи додатка складається в тому, що мобільний додаток отримавши координати початкової та кінцевої точки запитує у сервера які маршрути максимально наближені та проходять через обидві ці точки.

Тобто додаток відправляє запит, а сервер робить запит до бази даних параметрами якого є насамперед зупинки транспорту максимально наближені до вказаних точок, після отримання переліку зупинок сервер в БД знаходить до яких маршрутів належаться ті зупинки та відфільтровує ті, які проходять через начальну та кінцеву точку та задовольняють умові типу транспорту обрану користувачем. Далі сервер відправляє в зворотньому напрямку JSON масив з переліком маршрутів що проходять через ці точки та мобільний додаток промальовує їх користувачеві .

ВИСНОВКИ

У дипломній роботі проводиться дослідження в області міських інформаційних систем на базі мобільних інтерфейсів. Будуть вивчені підходи до реалізації подібних систем, основні методики геокодування в інформаційних системах орієнтованих на певну групу користувачів. Конечним результатом дипломної роботи є розробка програми для пошуку оптимального маршруту громадського транспорту.

На кожний транспортний засіб встановлюється комплексна система обліку пасажиропотоку. В основі системи – датчики з 10 елементів (5 інфрачервоних світло діоди та 5 фотодіодів того ж діапазону). Датчики розпізнають пасажирів які пройшли через них і передають дані по мережі Ethernet або інтерфейсу RS232 на бортовий контролер, встановлений в транспортному засобі. Підрахунок пасажирів відбувається за рахунок аналізу, що дає дуже високу точність і мінімальну похибку.

У свою чергу контролер передає дані про які зайшли і вийшли пасажирів на сервер моніторингу по мережі 2G / 3G. Дані можна записувати і на зовнішній носій.

Можлива інтеграція в систему продажу квитків. Так само, є ряд можливостей а саме:

- підрахунок пасажирів при посадці і висадці з точністю понад 80%;
 - виявлення повернень. Пасажири, що вийшли і швидко повернулися назад, не враховуються двічі;
 - використання на всіх видах транспорту, в тому числі на залізничному.
- Систему обліку пасажирів з такою системою можна використовувати навіть на масових заходах;

– інтеграція в систему продажу квитків. Якщо кількість пасажирів і кількість проданих квитків будуть не відповідати, це буде виявлено моментально;

– Інтеграція в систему ГЛОНАСС / GPS. Можна виявити моменти пікового і нульовий завантаження і співвіднести їх з геопозицією транспорту;

– моніторинг заповнюваності зони. Визначення місцезнаходження пасажирів в режимі реального часу;

– автоматичне завантаження записів на віддалене сховище.

ПЕРЕЛІК ПОСИЛАНЬ

1. Голев П. А. Система навигационного мониторинга транспорта Назва ресурсу: URL: <http://science.kuzstu.ru/wp-content/Events/Conference/RM/2017/RM17/pages/Articles/0301005-.pdf>.

2. Информационные системы и технологии логистики на транспорте Назва ресурсу: URL: <http://elib.psu.by/bitstream/123456789/20600/12/%D0%A2%D0%B5%D0%BC%D0%B0%2012.pdf>.

3. Как работает система мониторинга Назва ресурсу: URL: <https://monitoring-gps.ru/>.

4. Мониторинг транспорта Назва ресурсу: URL: <https://ask-glonass.ru/solutions-monitoring>.

5. Онлайн мониторинг транспорта Назва ресурсу: URL: <https://gurtam.com/ru/wialon/online-monitoring>

6. Принцип работы спутниковых систем Назва ресурсу: URL: http://woodygps.blogspot.com/p/blog-page_91.html (дата звернення: 10.11.2019).

7. Спутниковый мониторинг транспорта Назва ресурсу: URL: [https://ru.wikipedia.org/wiki/Спутниковый мониторинг транспорта](https://ru.wikipedia.org/wiki/Спутниковый_мониторинг_транспорта)

8. Транспортная логистика Назва ресурсу: URL: https://ru.wikipedia.org/wiki/Логистика#Транспортная_логистика

ДОДАТОК А

Скрипт Transport.php

```
<?php
//-----
header("Content-Type: text/html; charset=windows-1251");
require_once 'Settings.php';
//-----
//соединение с БД
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx) // Если дескриптор равен 0 соединение не установлено
{
echo("ERROR_DB_CONNECT_0<br>");
exit();
}
if ($dbcnx>0)
{
//echo("DB_CONNECT_OK<br>");
}
// Код соединения с базой данных
if (!@mysql_select_db($dbname, $dbcnx))
{
echo( "ERROR_DB_CONNECT_1<br>" );
exit();
}
//-----
if (isset($_POST['id'])&&((int)$_POST['id']>0)) {
$TransportId = $_POST['id'];
$Cmd = $_POST['Cmd'];
$Latitude = isset($_POST['Latitude'])? $_POST['Latitude'] : 0.000000;
$Longitude = isset($_POST['Longitude'])? $_POST['Longitude'] : 0.000000;
```

```

$Route = isset($_POST['Route'])? $_POST['Route'] : 0;
$Trtype = isset($_POST['Trtype'])? $_POST['Trtype'] : 0;
$Pcount = isset($_POST['Pcount'])? $_POST['Pcount'] : 0;
$Status = isset($_POST['Status'])? $_POST['Status'] : 0;
if ($Cmd == 1) {
    $query="UPDATE Transport SET Latitude=".$Latitude.", Longitude=".$Longitude." WHERE
EquipId=".$TransportId." );";
    $res = mysql_query($query);
    echo "OK1";
    exit();
}
if ($Cmd == 2) {
    $query="UPDATE Transport SET RouteNow=".$Route." WHERE EquipId=".$TransportId." );";
    $res = mysql_query($query);
    echo "OK2";
    exit();
}
if ($Cmd == 3) {
    $query="UPDATE Transport SET PassengersCnt=".$Pcount." WHERE EquipId=".$TransportId." );";
    $res = mysql_query($query);
    echo "OK3";
    exit();
}
if ($Cmd == 4) {
    $query="UPDATE Transport SET IsActive=".$Status." WHERE EquipId=".$TransportId." );";
    $res = mysql_query($query);
    echo "OK4";
}
else {
    echo "ERR";
    exit();
}
?>

```