

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи мобільного
контролю вантажоперевезень на базі ОС Android”

Виконав здобувач вищої освіти
II курсу, групи КН-22М-2
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Серeda В.Л.
« ____ » _____ 2023 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Середі Владиславу Леонідовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android*

2. Керівник роботи *Якименко Наталія Миколаївна, канд. фіз.-мат. наук, доцент*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 33-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна *1 аркуш*

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

Показники економічної ефективності *1 аркуш*

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Середа В.Л. Дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android. 122 Комп'ютерні науки. Центральнoукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мобільного контролю вантажоперевезень на базі ОС Android.

Метою розробки є дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

Об'єктом дослідження є процес мобільного контролю вантажоперевезень на базі ОС Android.

Предметом дослідження є методи мобільного контролю вантажоперевезень на базі ОС Android.

Методи дослідження базуються на методах розробки мобільних додатків, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на мобільному пристрої під керуванням ОС Android.

Програму розроблено в середовищі Java, Eclipse, Android SDK.

Ключові слова: комп'ютерні науки, вантажоперевезення, ОС Android

ABSTRACT

Sereda V.L. Research and software implementation of a mobile cargo transportation control system based on the Android OS. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the mobile control system of cargo transportation based on the Android OS.

The goal of the development is the research and software implementation of the mobile cargo transportation control system based on the Android OS.

The object of the study is the process of mobile control of cargo transportation based on the Android OS.

The subject of the study is the methods of mobile control of cargo transportation based on the Android OS.

Research methods are based on mobile application development methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the mobile cargo transportation control system based on the Android OS.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a mobile device running the Android OS.

The program was developed in the environment of Java, Eclipse, Android SDK.

Keywords: computer science, trucking, Android OS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	17
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	19
3.1 Опис функціонування системи	19
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми	35
3.4 Розробка діаграми процесів.....	38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	40
4.2 Захист розробленого програмного забезпечення.....	49
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	52
6 НАУКОВА НОВИЗНА	54

					ВКРМ-122.23.0045.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Середа В.Л.</i>					М	1	92
<i>Перев.</i>	<i>Якименко Н.М.</i>							
Н.контр.	<i>Коваленко А.С.</i>					ЦНТУ КН-22М-2		
Затв.	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	55
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	55
7.2 Розрахунок трудомісткості розробки програмної продукції.....	57
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	59
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	63
7.5 Визначення собівартості розробки та ціни програмної продукції.....	68
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	71
7.7 Визначення експлуатаційних витрат.....	72
7.8 Визначення економічної ефективності програмної продукції.....	73
7.9 Висновок.....	75
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	76
8.1 Вступ.....	76
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	77
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	78
8.4 Розробка заходів з умов поліпшення охорони праці.....	80
8.5 Розрахунок занулення.....	80
9 ОСНОВНІ ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІС	–	інформаційна система.
ОС	–	операційна система.
ЦПС	–	цифрова промислова система.
DSSS	–	метод розширення спектра.
LAN	–	локальна мережа.
RFID	–	радіочастотна ідентифікація.
SCADA	–	диспетчерський контроль і накопичення даних.
SSID	–	ідентифікатор мережі.
Wi-Fi	–	бездротова технологія.
WLAN	–	бездротова локальна мережа.

КБПЗ_2023

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Сучасні системи моніторингу автомобілів дозволяють у значній мірі знизити витрати на вантажоперевезення, обслуговування автомобілів, і підвищити ефективність роботи автомобільного парку. Більшість підприємств використовують технології IRNSS/GPS моніторингу автомобілів, та відповідно вантажоперевезень. Система спостереження за автомобілем повною мірою використовує ресурси супутникової системи моніторингу автомобілів. При виборі такої системи спостереження за автомобілем, необхідно визначити основні завдання підприємства, а потім вирішити, яка з існуючих систем IRNSS/GPS моніторингу автомобілів найбільше задовольняє поставленим цілям. Спостереження за автомобілем дозволяє, насамперед, запобігти викраденню, необґрунтований пробіг, так звані «ліві» рейси, витраті палива, все це можливо за допомогою супутникового моніторингу автомобілів. У наш час ціни на популярні марки бензину й дизельне паливо неухильно ростуть і всім автоволодарам стає зрозуміло наскільки актуальна на сьогоднішній день проблема спостереження за автомобілем. Один з основних способів оптимізації витрат – зниження зайвого пробігу й витрат на паливо. Багато які з несумлінних водіїв як і раніше використовують будь-яку можливість для розкрадання палива й «лівого» заробітку. Протидіяти цьому можливо за допомогою системи спостереження за автомобілем, створеної на основі GPS або IRNSS технологій. Так само, можливо контролювати маршрут руху автомобіля, якщо він змінить свій напрямок руху або відхилиться від маршруту, вийшовши за зону контролю, то про це моментально довідається.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем мобільного контролю вантажоперевезень на базі ОС Android.
- Дослідження системи мобільного контролю вантажоперевезень на базі ОС Android.
- Програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

Об'єктом дослідження є процес мобільного контролю вантажоперевезень на базі ОС Android.

Предметом дослідження є методи мобільного контролю вантажоперевезень на базі ОС Android.

Методи дослідження базуються на методах розробки мобільних додатків, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод мобільного контролю вантажоперевезень на базі ОС Android.
- Розроблено вітчизняний продукт мобільного контролю вантажоперевезень на базі ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі мобільного контролю вантажоперевезень на базі ОС Android.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2023

					VKPM-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

GPS-трекер – це спеціальний пристрій, що за допомогою супутників мережі глобального позиціонування відслідковує місце розташування, напрямок руху, швидкість об'єкта, а потім зберігає або пересилає дані своєму власникові.

Сьогодні кожний бажаючий довідатися, де перебуває об'єкт, що цікавить, може зробити це за допомогою глобальної системи позиціонування й такого пристрою, як GPS-трекер. Коштує мініатюрний GPS-трекер відносно недорого, для його покупки яких-небудь спеціальних обмежень або дозволів немає. Об'єктом спостереження в цьому випадку може бути людина, автомобіль, багаж або навіть улюблений домашній вихованець. Пристрій приймає сигнал із супутників і відправляє власникові координати трекера по стандарті GPS, після накладення координат на карту місцевості можна точно довідатися, де перебуває й куди рухається в цей момент об'єкт спостереження, тому числі й вантаж у автомобілі.

1.2 Область застосування

Найбільше масово GPS-трекери сьогодні застосовуються для контролю суспільного, вантажного й легкового транспорту на різних комерційних і авто-транспортних підприємствах. Трекери встановлені на більшості річкових і океанських кораблів, поїздах, автобусах, тролейбусах, електричках метрополітену. Дуже зручно використовувати GPS-трекери для контролю автомобільних вантажоперевезень, щоб відслідковувати рух вантажу й у випадку аварії або надзвичайної події відправити водієві допомогу. GPS трекери дуже корисні керівництву таксі й маршруток для відстеження пересування машин, що

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

знижує ризик викрадення й дозволяє контролювати роботу водіїв і відповідність розкладу.

Всі частіше стали застосовувати GPS трекери в різних комерційних компаніях для контролю менеджерів, регіональних представників, службових водіїв, експедиторів і інших осіб, чия робота пов'язана з використанням службового транспорту, перевезенням вантажів або виїздами до клієнтів.

Деякі догадливі батьки вже стали використовувати трекери для відстеження положення дитини під час прогулянок і відвідування школи.

На жаль, як і будь-яка технічна новинка, GPS-трекери можуть нести не тільки користь, але й шкоду. Із призначення цього приладу будь-якій людині стає зрозуміло, що це ще й відмінний засіб для стеження за конкурентами, друзями, колегами, чоловіками, за ким завгодно. Виявитися жертвою шпигунського трекера дуже легко – він непомітно може бути встановлений в автомобілі, дипломаті, робочій сумці, стільниковому телефоні, ноутбуці або навіть у взутті. Кожний ваш крок: поїздка до знайомих, зустрічі з постачальниками й партнерами, інтимні побачення, відпочинок і розваги – все це легко стане видобутком зловмисника. Складно навіть представити розміри заподіяного збитку, якщо маршрут ваших пересувань у дрібних деталях стане відомий зацікавленим людям. Бізнес, політична кар'єра, репутація, родина – все може звалитися в один момент.

Якщо вас заздалегідь не попереджали про установку GPS трекера у ваш автомобіль, мобільний телефон або іншу річ – самостійно виявити його без застосування спеціальної техніки практично неможливо. Однак, зараз якісні технічні засоби для виявлення й блокування GPS-сигналу вже можна придбати за доступними цінами в багатьох спеціалізованих магазинах.

Для чого потрібні знаходжувачі GPS-сигналу? Якщо ви не знаєте про наявність у вас GPS-трекера, хочете захистити себе від несанкціонованого стеження конкурентів, колег, дружини, шантажистів або задрісників – ви можете сканувати діапазон GPS частот за допомогою спеціального приладу. Як тільки

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

GPS-трекер почне роботу, знаходжувач вас про це сповістить. Деякі моделі дозволяють також визначити місце розташування жучка й знищити його або збити зловмисника зі сліду, причепивши трекер до бродячого собаки. На жаль, така тактика боротьби з GPS-жучками в багатьох випадках неефективна. Справа в тому, що GPS-трекер можуть у будь-який час повторно вам установити, а проводити сканування перед кожним виїздом складно. Крім того, GPS-трекер може бути неактивним і передавати сигнал лише в певні інтервали часу, засікти такий жучок значно складніше.

Коли застосовуються пригнічувачі GPS-сигналу? Це більш універсальні й ефективні пристрої, вони дозволяють заблокувати сигнали в GPS-діапазоні, у якому працюють трекери. У результаті, за допомогою цього пристрою можна легко паралізувати роботу будь-якого GPS-жучка в радіусі до 10-15 метрів. Значним плюсом GPS-пригнічувача є те, що з його допомогою можна на певний час позбутися в тому числі й від санкціонованого стеження, не руйнуючи GPS-трекер. Навіть якщо за вами стежить диспетчер, начальник транспортної служби або директор – можна буде завжди "відлучитися", щоб заїхати додому по особистих справах або пообідати із друзями. Не менш надійно подібний прилад захистить у випадку несанкціонованого стеження з боку керівництва, родичів, колег, конкурентів і т.д. Саме головне – не забувати включити прилад перед важливими поїздками.

Таким чином, виходячи з вищеперерахованого, розробка програмного забезпечення системи мобільного контролю вантажоперевезень на базі ОС Android, є актуальною задачею, яка потребує вирішення у даному магістерському проєкті.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень по профілю теми магістерського проекту

Опис GPS трекінгу від Global House

Супутниковий моніторинг автотранспорту дозволяє визначати місцезнаходження комерційного й приватного транспорту в режимі реального часу й надає необмежені можливості по плануванню й розробці схем більше ефективної роботи, диспетчеризації й охорони автотранспорту.

Озброївшись GPS\IRNSS Трекінгом, зможете:

1. Визначати місцезнаходження, швидкість і маршрут свого автотранспорту в реальному часі.

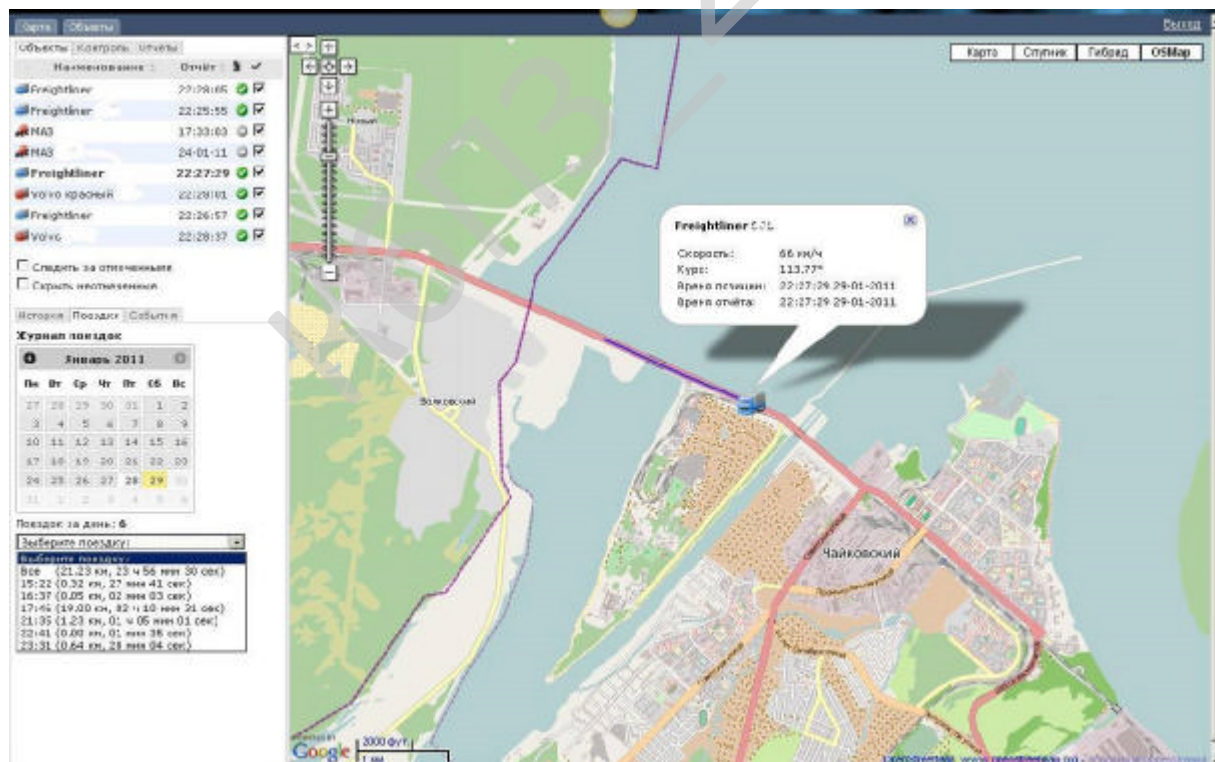


Рисунок 2.1 – Інтерфейс користувача GPS трекінгу від Global House

2. Підвищити продуктивність і доходи автопарку, маючи детальні звіти про роботу транспорту, включаючи інформацію:

- Про маршрут, паркування, зупинки й кілометражі.
- Про годинники роботи двигуна.
- Про необхідність техобслуговування.

3. Супутникове спостереження допоможе істотно скоротити витрату пального, маючи інформацію:

- Про час роботи двигуна на холостих обертах.
- Про перевищення швидкості.
- Про витрату пального.
- Про карту маршруту.

4. Збільшити безпеку водіїв, одержуючи повідомлення на Ваш мобільний телефон і електронну пошту якщо Ваш транспорт:

- Покинув межі ділянки гео-кордони.
- Евакуюють.
- Перевищує швидкість та ін.
- Дистанційно припинити роботу двигуна (опціонально).

Впровадження системи дозволить вам не тільки заощадити до 20-50% палива, але й підвищити ефективність, безпеку використання автопарку, і підвищити бізнес вартість підприємства. Ви завжди будете знати місце розташування машини, і контролювати всі пересування.

Маршрут на карті тепер можна переглядати й за будь-який період у годинниках і по кожній поїзді окремо.

Трекер Плюс

При розробці Трекер Плюс – системи супутникового моніторингу на базі глобальних систем позиціонування GPS і IRNSS відштовхувалися від принципів доступності, функціональності й простоти використання програми спостереження за мобільними телефонами, оснащеними GPS і спеціалізованими трекерами. Система GPS моніторингу місця розташування об'єктів Трекер Плюс

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

дозволяє довідатися, де перебуває телефон, на якому встановлена програма, що здійснює спостереження за телефоном, або де перебуває GPS-трекер, настроєний на наш сервер в Інтернеті. Супутникове спостереження в реальному часі, реалізоване у Трекер Плюс дає можливість не тільки довідатися, де перебуває людина й визначити її місце розташування за допомогою телефону, але й спостерігати за його переміщеннями, зберігаючи й експортуючи його маршрути пересування, сповіщати по Email або SMS про вхід або вихід спостережуваного об'єкта в «зону контролю» і т.д. Області затребуваності даного проекту досить широкі. Це може бути моніторинг транспорту, спостереження за телефоном і його пересуванням і побудова маршрутів з можливим експортом в інші системи. Проект дозволяє стежити за будь-яким видом транспорту, на якому встановлений пристрій GPS tracker або мобільний телефон із програмою Трекер Плюс у будь-якій крапці планети, де є GSM або CDMA стільниковий зв'язок і є впевнений прийом сигналу від GPS або IRNSS супутників.

Одержуючи сигнал із супутників глобальних систем позиціонування GPS або IRNSS, GPS пристрій (мобільний телефон або трекер) визначає своє місце розташування й передає координати по мережах стільникового зв'язку. У випадку відсутності сигналу GSM або CDMA іде збір координат на внутрішню пам'ять пристрою або стільникового телефону. Після того, як з'являється можливість з'єднатися з мережею Інтернет, відбувається вивантаження координат, на основі яких на інтерактивній карті Трекер Плюс відображається місце розташування об'єкта спостереження й будується маршрут його пересування. Існуючий функціонал системи дозволяє відображати маршрут пересування об'єкта спостереження за будь-який проміжок часу, а так само точне місцезнаходження в цей момент. Можливість побудови маршруту пересування за будь-який період часу транспортного засобу сGPS пристроєм надає істотну перевагу перед конкурентами.

Система GPS моніторингу об'єктів Трекер Плюс дозволяє миттєво інформувати по СМС або електронній пошті про перетинання об'єктом

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

спостереження границь певних користувачем «зон контролю», будь то стільниковий телефон (GPS комунікатор) або GPS трекер. Тобто існує можливість визначити місце розташування стільникового телефону в момент перетинання границі контрольної зони. Це дуже зручно для спостереження за дитиною або людиною, якій може знадобитися допомога в будь-який момент. Спостереження за дітьми – це одна з найбільш затребуваних послуг у наш неспокійний час. Кожний батько хоче знати, де перебуває його дитина в цей момент часу, і де вона перебувала учора. Роблячи спостереження за телефоном за допомогою системи GPS моніторингу об'єктів Трекер Плюс, в будь-який момент часу можливо довідатися, де перебуває телефон.

У системі GPS моніторингу Трекер Плюс реалізований адаптивна система економії Інтернет трафіку. Web інтерфейс дозволяє одержати доступ до особистого кабінету Трекер Плюс у будь-якій точці, де є вихід у мережу Інтернет.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для розробки програмного забезпечення мною було використане середовище Eclipse. Eclipse – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation. Написаний в основному на Java, може бути використаний для розробки застосунків на Java і, за допомогою різних плагінів, на інших мовах програмування, включаючи Ada, C, C++, COBOL, Fortran, Perl, PHP, Python, R, Ruby (включно з каркасом Ruby on Rails), Scala, Clojure та Scheme. Середовища розробки зокрема включають Eclipse ADT (Ada Development Toolkit) для Ada, Eclipse CDT для C/C++, Eclipse JDT для Java, Eclipse PDT для PHP.

Eclipse являє собою фреймворк для розробки модульних кросплатформових застосунків із низкою особливостей:

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

- можливість розробки ПЗ на багатьох мовах програмування (рідною є Java);
- крос-платформова;
- модульна, призначена для подальшого розширення незалежним розробниками;
- з відкритим сирцевим кодом;
- розробляється і підтримується фондом Eclipse, куди входять такі постачальники ПЗ, як IBM, Oracle, Borland.

Спочатку проект розроблявся в IBM як корпоративний стандарт IDE, настановлений на розробки на багатьох мовах під платформи IBM. Потім проект було перейменовано на Eclipse і надано для подальшого розвитку спільноті.

Eclipse насамперед повноцінна Java IDE, націлена на групову розробку, має засоби роботи з системами контролю версій (підтримка CVS входить у поставку Eclipse, активно розвиваються кілька варіантів SVN модулів, існує підтримка VSS та інших). З огляду на безкоштовність, у багатьох організаціях Eclipse – корпоративний стандарт для розробки ПЗ на Java.

Друге призначення Eclipse – служити платформою для нових розширень. Такими стали C/C++ Development Tools (CDT), розроблювані інженерами QNX разом із IBM, засоби для підтримки інших мов різних розробників. Безліч розширень доповнює Eclipse менеджерами для роботи з базами даних, серверами застосунків та інших.

З версії 3.0 Eclipse став не монолітною IDE, яка підтримує розширення, а набором розширень. У основі лежать фреймворки OSGi, і SWT/JFace, на основі яких розроблений наступний шар – платформа і засоби розробки повноцінних клієнтських застосунків RCP (Rich Client Platform). Платформа RCP є базою для розробки різних RCP програм як торент-клієнт Azareus чи File Arranger. Наступний шар – платформа Eclipse, що є набором розширень RCP – редактори, панелі, перспективи, модуль CVS і модуль Java Development Tools (JDT).

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Eclipse написана на Java, тому є платформи-незалежним продуктом, крім бібліотеки графічного інтерфейсу SWT, яка розробляється окремо для більшості поширених платформ. Бібліотека SWT використовує графічні засоби платформи (ОС), що забезпечує швидкість і звичний зовнішній вигляд інтерфейсу користувача.

Відповідно до IDC, із Eclipse працюють 2,3 мільйона розробників.

Основою Eclipse є платформа розширеного клієнта. Її складають такі компоненти:

- ядро платформи (завантаження Eclipse, запуск модулів);
- OSGi (стандартне середовище постачання комплектів);
- SWT (стандартний інструментарій віджетів);
- JFace (файлові буфери, робота з текстом, текстові редактори);
- робоче середовище Eclipse (панелі, редактори, проекції, майстри).

GUI в Eclipse написаний з використанням інструментарію SWT. Останній, на відміну від Swing (який лише емулює окремі графічні елементи використовуваної платформи), дійсно використовує графічні компоненти даної системи. Призначений для користувача інтерфейс Eclipse також залежить від проміжного шару GUI, званого JFace, який спрощує побудову призначеного для користувача інтерфейсу, що базується на SWT.

Гнучкість Eclipse забезпечується за рахунок модулів, що підключаються, завдяки чому можлива розробка не тільки на Java, але і на інших мовах, таких як C/C++, Perl, Groovy, Ruby, Python, PHP, Erlang та інших.

Програми для Android є програмами в нестандартному байт-кодi для віртуальної машини Dalvik.

Google пропонує для вільного завантаження інструментарій для розробки (Software Development Kit), який призначений для x86-машин під операційними системами Linux, Mac OS X (10.4.8 або вище), Windows XP, Windows Vista та Windows 7. Для розробки потрібен Java Development Kit 5 або новіший.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Розробку додатків для Android можна вести мовою Java (не нижче Java 1.5). Існує плагін для Eclipse – «Android Development Tools» (ADT), призначений для Eclipse версій 3.3-3.7. Для IntelliJ IDEA також існує плагін, який полегшує розробку Android-додатків. Для середовища розробки NetBeans розроблено плагін, який починаючи з версії Netbeans 7.0 перестав бути експериментальним, проте поки не є офіційним. Крім того існує Motodev Studio for Android, що являє собою комплексне середовище розробки, засноване на базі Eclipse і дозволяє працювати безпосередньо з Google SDK.

Крім того в 2009 році на додаток до ADT був опублікований Android Native Development Kit (NDK), пакет інструментаріїв і бібліотек дозволяє вести розробку додатків на мові C/C++. NDK рекомендується використовувати для розробки ділянок коду, критичних до швидкості.

Доступні бібліотеки:

- Bionic (Бібліотека стандартних функцій, несумісна з libc);
- libc (стандартна системна бібліотека мови C);
- мультимедійні бібліотеки (на базі PacketVideo OpenCORE; підтримують такі формати, як MPEG-4, H.264, MP3, AAC, AMR, JPEG та PNG);
- SGL (рушій двовимірної графіки);
- OpenGL ES 1.0 (рушій тривимірної графіки);
- Surface Manager (забезпечує для додатків доступ до 2D/3D);
- WebKit (готовий рушій для Web-браузера; обробляє HTML, JavaScript);
- FreeType (рушій обробки шрифтів);
- SQLite (проста система керування базами даних, доступна для всіх застосувань);
- SSL (протокол, що забезпечує безпечну передачу даних по мережі).

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

В порівнянні із звичайними додатками Linux, додатки Android підкоряються додатковим правилам:

- Content Providers – обмін даними між додатками.
- Resource Manager – доступ до таких ресурсів, як файли XML, PNG, JPEG.
- Notification Manager – доступ до рядка стану.
- Activity Manager – управління активними додатками.

Для Android був розроблений формат інсталяційних пакетів .apk.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи мобільного контролю вантажоперевезень на базі ОС Android.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ_2023

					VKPM-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

GPS-трекер у транспортному бізнесі приносить більше прибутку, ніж будь-коли, завдяки ефективному використанню ресурсів і аналітики.

З покращенням епідемічної ситуації галузь логістики та транспортування в основному відновила своє колишнє процвітання. Для транспортного сектору обслуговування транспортних засобів є одним із найважчих завдань. Використання технології GPS має велике значення у сфері вантажних перевезень. У сфері транспортування та логістики споріднені компанії також постійно шукають ефективні та інтелектуальні методи управління для вирішення проблем транспортних засобів.

Немає сумніву, що GPS-локатори є найкращим вибором для галузі вантажних перевезень. Встановіть GPS-локатори для транспортних засобів або товарів і отримайте доступ до системи керування GPS. Менеджери запитують про місцезнаходження транспортного засобу та швидкість транспортного засобу через систему управління. Система позиціонування GPS у логістичній галузі також поділяється на дві частини: GPS-локатори, встановлені в транспортних засобах або товарах.

До переваг використання технологічних рішень GPS у сфері вантажних перевезень можна віднести:

1. Керуйте транспортними засобами: ви можете повністю зрозуміти розташування та маршрути всіх транспортних засобів на платформі в режимі реального часу, завершити планування та розгортання вчасно та заощадити ресурси.

2. Оволодійте поведінкою водія за кермом: опануйте інформацію про автомобіль через систему керування, включаючи швидкість руху, час

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

перебування, час роботи тощо, щоб запобігти неадекватній поведінці водіння.

3. Допоможіть зменшити експлуатаційні витрати: переглядайте транспортні засоби, що не використовуються, та транспортні засоби, що використовуються, розумно розподіляйте їх і переконайтеся, що всі транспортні засоби виконують завдання з перевезення вантажів; в той же час, це сприяє інтеграції та коригування транспортних маршрутів для максимізації паливної ефективності та зниження експлуатаційних витрат.

4. Обробка надзвичайних ситуацій у режимі реального часу: зворотний зв'язок інформації про транспортний засіб у режимі реального часу для своєчасного вирішення надзвичайних ситуацій;

5. Інтегрована безпека транспортування: якщо товари, що перевозяться транспортним засобом, є відносно дорогими, GPS-локатор може отримати детальну інформацію про місцезнаходження в будь-який час, зменшити приховану небезпеку крадіжки та включити фактор безпеки.

У сфері транспорту та логістики GPS або технологія GPS сьогодні не просто інструмент, який допомагає водіям орієнтуватися з місця. Завдяки прогресу мобільних технологій і мобільних додатків для логістичних мобільних рішень.

Технологія GPS повністю змінила транспортну галузь. Вони є розробниками мобільних додатків, які можуть надати вам вдосконалені системи відстеження транспортних засобів GPS, які включають ефективні інструменти, необхідні для вашого бізнесу та потреб клієнтів.

За допомогою інтелектуального мобільного пристрою ви можете легко знаходити свої транспортні засоби, ефективно планувати завдання водіїв і керувати витратами.

Ці функції ідеально підходять для успішного ведення транспортного або логістичного бізнесу без здійснення багатьох дзвінків або перегляду громіздких звітів для прийняття рішень.

Технологія GPS-відстеження може творити чудеса для вашого бізнесу, а

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

переваги від неї неперевершені.

Це провідні компанії з розробки мобільних додатків із винятковою командою та досвідом, які пропонують першокласні мобільні рішення для бізнесу, які нададуть вашому логістичному бізнесу найкращий мобільний досвід.

Завдяки творчому підходу та досвіду розробки рішень для мобільних додатків розробники розробляють високоефективне програмне забезпечення для відстеження перевезень, адаптоване до конкретних потреб вашого бізнесу, яке допоможе вам досягти 100% задоволеності клієнтів.

Базове рішення GPS-відстеження для логістичного бізнесу

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

Основне використання програмного забезпечення GPS для відстеження транспортних засобів – це ведення запису про місцезнаходження вашого транспортного засобу на карті. Це дає змогу побачити, який транспортний засіб рухається та в якому напрямку.

Завдяки технології відстеження GPS ви можете контролювати швидкість автомобіля, маршрути, запуск і зупинку двигуна, холостий хід і маршрути.

Ви можете визначити, чи прямує транспортний засіб до призначеного пункту призначення, які маршрути було використано та коли автомобіль відпочиває.

Це також стане в нагоді під час пошуку викраденого автомобіля. Такий контроль парку транспортних засобів є ключовим елементом успіху в транспортно-логістичному бізнесі.

Інновації в розробці додатків GPS-відстеження пропонують кілька переваг для вашого транспортного бізнесу. Від доставки, бек-офісних процедур, продуктивності та засобів зв'язку.

GPS-відстеження дозволяє розробляти логістичні мобільні рішення, які допоможуть вам взяти під контроль і контролювати транспортні засоби та робочу

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

силу.

Сучасне мобільне програмне забезпечення для відстеження транспортних перевезень може надавати інформацію, яка є критично важливою для оцінки витрат, відмінного обслуговування клієнтів і підвищує ефективність.

1. Управління автопарком

Сучасні бізнес-рішення GPS дозволяють мати постійний двосторонній зв'язок із вашим автопарком. Якщо вам потрібно змінити маршрут будь-якого транспортного засобу в парку або призначити нову посадку за тим самим маршрутом, ви можете зробити це на основі інформації, отриманої від програмного забезпечення для відстеження транспортування.

У додатку також буде функція, яка дозволить сповіщати водіїв, які їдуть неправильним шляхом, або застерігати їх від неправомірної поведінки під час виконання службових обов'язків.

Ця функція стосується ефективного планування та організації. Ви можете скасувати доставку або перепрограмувати маршрут, а також прийняти інші обґрунтовані рішення на основі місцезнаходження автомобіля.

Для досягнення підвищеної ефективності та гнучкості при малих бюджетах; все, що вам потрібно зробити, це створити власну систему GPS-стеження.

2. Оптимізація ресурсів

Система GPS допомагає підвищити прозорість між вами та водіями. Завдяки GPS-моніторингу ви можете стежити за будь-якими порушеннями з боку водіїв.

По-друге, ви можете визначити точну причину затримки доставки, якщо така є. Це, безсумнівно, допомогло б зміцнити довіру між вами та водіями, оскільки не було б місця для будь-яких підозр

3. Безпека та продуктивність водія

Водії є хребтом вашого транспортного та логістичного бізнесу. Їхня безпека та благополуччя мають бути найважливішими. Технологія GPS

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

забезпечує безпеку водія, перевіряючи поведінку за кермом.

Постійне стеження утримує їх від перевищення швидкості та зухвалої їзди. За допомогою технології GPS ви можете ідентифікувати водіїв, які демонструють хороші навички водіння, і можете запропонувати відповідні винагороди чи заохочення.

4. Неперевершена ефективність

Система відстеження транспортних засобів GPS є автоматизованою та надає оновлення в реальному часі на вашому віддаленому пристрої. Це допомагає швидко приймати рішення.

Менеджерам і водіям не потрібна паперова тяганина, тому у вас буде більше часу, щоб зосередитися на інших службових обов'язках. Водії їдуть, не турбуючись про пробіг внаслідок дорожніх робіт чи об'їздів на своїх маршрутах.

Таким чином, система GPS є важливим інструментом для отримання найкращої продуктивності та ефективності від кожного вашого персоналу.

Створивши власне спеціальне програмне забезпечення на основі GPS, ви зможете швидко й ефективно впоратися з будь-якою технічною несправністю, негайно призначивши наступний найближчий транспортний засіб.

5. Зменшити адміністративний ресурс

Збір статистичних даних, як-от звітів про пробіг, вимагає залучення працівників у дорозі. Це призведе до величезної кількості робочої сили на дорозі, що зрештою збільшить вартість.

Крім того, це також збільшує кількість співробітників в офісі, щоб завершити роботу, що залишилася. Технологія GPS зменшує це число в цілому, оскільки стає можливим отримувати статистику безпосередньо з автомобіля.

6. Аналіз/оцінка витрат

Ви можете аналізувати витрати на паливо для виконання завдань, пов'язаних із бізнесом, і контролювати споживання палива для особистої вигоди.

За допомогою відстеження транспортних засобів ви також можете визначити старі транспортні засоби, які споживають більше витрат, і позбутися

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

шкідливих звичок водіїв, які можуть призвести до збільшення споживання палива.

7. Мінімальні ризики крадіжки

У транспортно-логістичному бізнесі на кону великі гроші. У будь-якому випадку, якщо викрадений транспортний засіб з коштовними партіями; тоді це призведе до величезних збитків і негативно позначиться на вашій діловій репутації.

Система GPS допомагає надсилати сповіщення у разі будь-якої подібної події. Це допоможе вам вжити відповідних заходів проти винних.

8. Якісне обслуговування клієнтів

Правильне планування допоможе вам розмістити більше поставок на день, що призводить до швидшого обслуговування клієнтів. Це також дозволяє точно інформувати клієнтів про очікуваний час прибуття та у разі будь-яких затримок.

У даній роботі розробляється логістичне мобільне рішення за допомогою розробки додатків GPS-відстеження, створених для досягнення максимальної продуктивності та прибутковості для вашого бізнесу. Наш мобільний додаток для відстеження транспортних засобів пропонує технології майбутнього вже сьогодні.

Технологія GPS є обов'язковою, якщо ви володієте транспортним або логістичним бізнесом. Саме тому, що GPS має безліч переваг, які можуть цілісно змінити ваш бізнес.

Отже, ви повинні виділити значну частину грошей на розгортання технології GPS стандартної якості, оскільки вона завжди матиме користь. Ви можете проконсультуватися з компаніями, які займаються розробкою мобільних додатків, щоб отримати уявлення про вартість розробки програми GPS-відстеження.

Пошук ідеального ідентифікатора розташування для мобільних програм: GPS проти маяків проти Wi-Fi

Технології навігації стали віхою та відіграли важливу роль в успіху

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

смартфонів. GPS, Wi-Fi і Beacon є трьома найбільш використовуваними технологіями. У цьому блозі ми порівняємо всі три технології та обговоримо їхні плюси та мінуси.

Як часто ви шукаєте в Інтернеті найближчі кафе, кінотеатри чи лікарів? Результати більш ніж задовольняють більшість часу, якщо «Місцезнаходження» ввімкнено на вашому пристрої.

Чип GPS, вбудований у ваш смартфон, коли ввімкнено, відстежує ваше місцезнаходження на основі осі координат.

Програми для смартфонів або веб-додатки, такі як Google Maps, показують, де ви точно.

Пристрій GPS потребує безперервного зв'язку з одним або декількома супутниками в чистому небі, щоб забезпечити його роботу.

Це означає, що функціональність GPS зазвичай погіршується в приміщенні або в хмарний вечір.

У наші дні для мобільних додатків потрібне місцезнаходження, щоб покращити персоналізацію та зробити взаємодію з користувачем більш актуальною.

У результаті розробники мобільних додатків використовують різні нові та традиційні технології та методи, щоб переконатися, що «Місцезнаходження» постійно доступне для їхніх додатків, незалежно від того, чи знаходиться пристрій у приміщенні, на вулиці чи зливою, наче завтра не настане.

Порівняно новіша технологія Beacon дуже добре вважається GPS «внутрішніх приміщень» завдяки своїй точності визначення місця розташування до міліметрів.

Прийняття технології Beacon значно нижче порівняно з іншими технологіями визначення місцезнаходження, такими як

GPS, Wi-Fi та RFID, але він швидко набирає популярності, оскільки все більше й більше маркетологів повинні усвідомлювати його цінність для бізнесу.

Wi-Fi вже давно використовується для покращення місцезнаходження

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

GPS. Відносно розташування пристрою, пов'язане з найближчою точкою доступу Wi-Fi, дає змогу відкалібрувати положення навіть у похмурий день, навіть у частково закритому просторі.

Давайте порівняємо технології GPS, Wi-Fi і Beacon і зробимо висновок, яку з них слід включити розробнику мобільних пристроїв у свою програму для найнадійнішого й точного визначення місцезнаходження.

GPS

GPS (або подібна система супутникової навігації як Galileo Європейського Союзу) є основою будь-якої зовнішньої системи визначення місцезнаходження в світі.

GPS забезпечує все: від навігації по автобусній карті (за допомогою програми веб-карт, як-от Garmin) до результатів локального пошуку (за допомогою веб-пошукової системи, як-от)

GPS є, мабуть, найпопулярнішою технологією в цій галузі завдяки численним перевагам.

Це пов'язано з тим, що він надає набагато більше, ніж інформація, така як дата, час, місцезнаходження та маршрути.

Крім того, GPS також надає такі переваги, як аудіодопомога під час надання вказівок користувачеві.

Рішення GPS-відстеження з усіма цими функціями дозволяють користувачам використовувати свої смартфони як пристрої GPS-відстеження.

Однак, як я вже згадував раніше, рішення на основі GPS майже марні, коли перешкода блокує супутниковий зв'язок із пристроєм.

Конструкція може бути будь-якою – від даху до хмар.

Geofencing – це відносно новий метод, який перевіряє дані про місцезнаходження, отримані від GPS-чіпа всередині смартфона, щоб ініціювати дію.

Сьогодні багато програм для iOS і Android включають API геозонування для надсилання сповіщень на основі місцезнаходження.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Поінформованість про місцезнаходження зазвичай налаштовується в програмі за допомогою інтегрованих API карт, як-от API карт Google для Android.

Геозонування працює без будь-яких збоїв, коли визначений віртуальний периметр або геозонована зона знаходяться на відкритому повітрі, як у футбольного клубу, або за гарної погоди.

Однак, як тільки геозона переміщується в закриту територію, як-от кінотеатр, геозона перестає існувати.

Докладніше: Як відстеження GPS може змінити управління автопарком?

Це означає, що якщо ви оновили програму для бронювання фільмів, щоб спрямовувати глядачів до відповідних місць, встановлюючи віртуальні межі для кожного місця, очікуйте помірного падіння рейтингів у App Store.

Переваги:

- GPS дуже доступний, що робить його широко поширеною технологією.
- Необмежений діапазон.
- Для цього потрібна згода користувача.

Недоліки:

- Системи GPS-відстеження споживають багато енергії, тому миттєво розряджають акумулятори мобільних пристроїв.
- Додаткам GPS-відстеження бракує точності в приміщенні, оскільки вони чутливі до будь-яких перешкод.
- Через низьку точність GPS також не є перевіреною технологією для «мікролокації» або дій на основі близькості.

Випадки використання:

- Для ширшого спектру діяльності або «Макророзташування».
- Зовнішнє середовище.

Докладніше: важливі програми для подорожей на основі місцезнаходження для бізнесу

Wi-Fi

Wi-Fi спочатку не був розроблений для визначення місцезнаходження.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Насправді в перші роки його вважали невдалим. Коли Intel зробила його стандартом для всього асортименту ноутбуків, рівень впровадження зріс експоненціально. Освітні кампуси стали прихильниками, а з масовим виробництвом маршрутизаторів Wi-Fi і падінням вартості Wi-Fi став глобальним явищем.

Якщо ви живете в добре заселеному районі, пошук 10-20 гарячих точок на екрані вашого смартфона вже не дивно.

Сьогодні Wi-Fi настільки поширений, що розробники додатків припускають, що користувач завжди буде оточений точками доступу Wi-Fi, тому щоразу, створюючи програму, вони не забувають інтегрувати в неї Wi-Fi.

Знайти розташування смартфона відносно фіксованої точки доступу Wi-Fi так само просто, як додати рядок коду для обчислення відстані між ними.

За допомогою розумного кодування GPS разом із налаштуванням Wi-Fi у приміщенні можна створити карту закритих приміщень, таких як торговий центр.

Якщо ви проходите повз магазин Zara на проході, найближчий Wi-Fi може надіслати вам сповіщення з кодом купону.

Весь механізм можна змусити працювати без програми, якщо на цьому пристрої активне радіо Wi-Fi.

Тепер давайте обговоримо переваги та недоліки технології GPS.

Переваги:

- Висока доступність і доступність.
- Діапазон нескінченний.
- Потрібна згода користувача.
- Його можна використовувати без програми.
- Різні компанії та магазини можуть використовувати його для залучення клієнтів.

Недоліки:

- GPS не може пробити міцні стіни.
- Вимагає багато енергії та розряджає акумулятор менш ніж за 12 годин.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

– Зазнає впливу через атмосферу, багатопрореневість та електромагнітні перешкоди.

– Він має проблеми з конфіденційністю для своїх користувачів, оскільки він не запитує жодних дозволів.

Випадки використання:

– Для доставки звичайного мобільного контенту.

– Внутрішнє середовище.

Технологія маяка

Google Eddystone і iBeacon від Apple є наріжним каменем технології Beacon, що підтримується технологією Bluetooth Low Energy (BLE).

Технологію Beacon можна додати до існуючої програми для Android або iOS за допомогою вищезгаданих профілів Beacon.

Мобільні додатки з увімкненим будь-яким із профілів Beacon підключають ваш пристрій до мініатюрних пристроїв BLE, які зазвичай розміщують у торгових та подібних закритих комерційних приміщеннях.

Маяки доступні у багатьох формах, таких як USB-накопичувачі та монетні осередки. Він надсилає радіохвилі, які можуть легко пробити такі конструкції, як стіни та стовпи

Коли людина заходить у такий простір, як-от сусідній супермаркет із встановленими маяками, пристрій у його кишені фіксує передані сигнали Bluetooth у закодованому вигляді, щоб розшифрувати їх програмою з підтримкою Beacon.

Якщо програма може розшифрувати сигнали, вона обчислює абсолютну позицію користувача пристрою відносно маяка всередині супермаркету та надсилає йому сповіщення про місцезнаходження.

Наприклад, програма може координувати дії з найближчим маяком, щоб надсилати ексклюзивну пропозицію кожного разу, коли людина буває в відділі шампунів на ринку.

Але жодна технологія не є ідеальною. Що, якщо супермаркет за розміром

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

наближається до Walmart, власнику доведеться встановити набагато більше пристроїв, ніж він може собі дозволити.

Радіус дії BLE становить 328 футів. Уявіть собі, скільки пристроїв надто амбітний виконавець встановить у величезному торговому центрі Dubai Mall площею 5,9 мільйона квадратних футів.

Але знову ж таки, не кожен комерційний простір є Dubai Mall. Для невеликих і середніх комерційних або гостинних приміщень,

немає кращого ідентифікатора місцезнаходження, ніж маяк, обладнаний BLE, для вашого додатка, якщо він призначений для вказівки покупцям у торговому просторі.

Тепер давайте обговоримо переваги та недоліки технології GPS

Переваги:

- Висока доступність і доступність.
- Діапазон нескінченний.
- Потрібна згода користувача.

Недоліки:

- GPS не може пробити міцні стіни.
- Вимагає багато енергії та розряджає акумулятор менш ніж за 12 годин.
- Зазнає впливу через атмосферу, багатопроменевість та електромагнітні

перешкоди.

Розробка багатомовних мобільних додатків

Сьогодні програми, як правило, за замовчуванням увімкнули місцезнаходження на основі GPS і Wi-Fi.

Оскільки Beacon є відносно новою технологією, впровадження програми все ще знаходиться на стадії зародження та буде зростати, оскільки все більше і більше роздрібних приміщень охоплюватимуть цю просту, але мобільну технологію додатків для підвищення бізнесу.

На відміну від Wi-Fi і GPS, Beacon не потребує спеціального апаратного забезпечення, оскільки майже кожен смартфон, навіть безглуздий, має радіо

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Bluetooth.

Галузь очікує, що завдяки збільшенню виробництва, збільшенню попиту та підтримці як Apple, так і Google ціни на пристрої Beacons скоро впадуть, ніж пізніше, що змусить ще більше роздрібних торговців і лікарень встановлювати пристрої Beacon у своїх приміщеннях.

Тим часом маркетингологи повинні підготуватися до наступної революції під назвою «роздрібний маркетинг на основі місцезнаходження», укладеного в технологію під назвою BLE.

3.2 Розробка структурної схеми

Гірничодобувна промисловість є вузькоспеціалізованою галуззю, яка значною мірою покладається на велике обладнання для своїх операцій. Незважаючи на те, що видобувні компанії є досить унікальними та мають справу з проблемами, яких не мають інші галузі, завдяки технологіям видобувні компанії можуть легко керувати своєю повсякденною діяльністю. Пристрої відстеження GPS допоможуть вирішити виклики, з якими сьогодні стикається гірничодобувна промисловість, уникаючи при цьому непотрібних витрат і ускладнень.

Гірничодобувна промисловість стикається з багатьма проблемами на різних етапах бізнес-циклу. Логістичні проблеми включають потребу в транспорті для переміщення важкого обладнання та транспортування видобутої продукції, оскільки місця видобутку зазвичай розташовані у віддалених районах.

Віддаленість викликає серйозне занепокоєння. У місцях видобутку та на великих територіях навколо них традиційні засоби радіозв'язку можуть бути недоступні, тобто жодна з стільникових мереж (2G, 3G або 4G LTE) не може бути використана для зв'язку. З іншого боку, супутникова мережа Iridium охоплює всю поверхню землі та забезпечує унікальний засіб передачі даних.

Неефективне управління може швидко призвести до великих і непотрібних витрат. Відстежуючи завантажені транспортні засоби, компанії

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

можуть гарантувати, що час не витрачається на дорогу, і мінімізують дорогий ризик крадіжки. Вантаж повинен постійно контролюватися – від місця шахти до кінцевого пункту призначення. Тільки тоді його можна безпечно доставити.

Ключем до ефективного планування та організації повсякденних операцій є безперервне та без затримок отримання інформації, як-от місцезнаходження транспортних засобів, навіть якщо сигнал GSM недоступний або втрачений. Таким чином, компанія ніколи не втратить свій автопарк і зможе належним чином оновити ланцюг поставок, який включає різні види транспорту. Все це дозволяє досягти кінцевої мети доставки вантажу вчасно.

Також важливо мати точні дані про завантажений вантаж. Інакше існує ризик помилок в обліку і, що більш важливо, частина вантажу може бути вкрадена без відома. У гірничодобувній промисловості, де в постійному обігу перебувають великі обсяги продукції, питання запобігання крадіжкам справді є дуже актуальним.

Супутники GPS

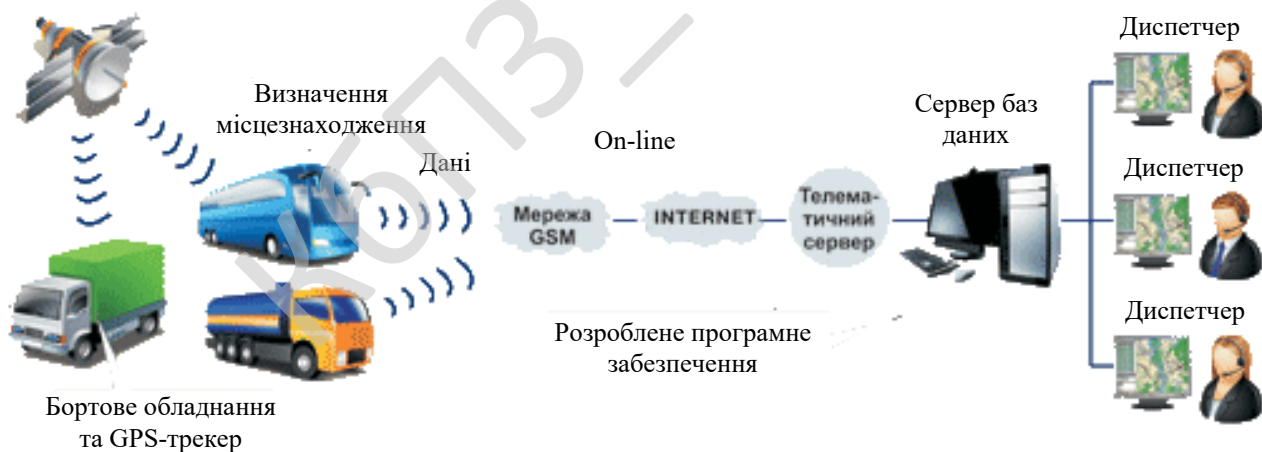


Рисунок 3.1 – Структурна схема системи

Автомобільний GPS-трекер Teltonika FMC650 серії PROFESSIONAL надає найточніші доступні дані відстеження. Маючи цю інформацію у вас під рукою,

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

ви можете миттєво приймати рішення та економити час на доставку вантажу. У результаті ви можете зробити більше пробіжок за той самий час.

Як це працює – коли мережа GSM недоступна, супутниковий модем Iridium Edge® можна використовувати для передачі даних на сервер через супутникову мережу Iridium. Від Північного до Південного полюса, на відміну від стільникового зв'язку, телематичне рішення Iridium Connected® працює в будь-якому куточку світу. Це означає, що інформація доступна безперервно, тому ви можете постійно контролювати свій автопарк.

Точні дані про навантаження можна отримати за допомогою датчиків навантаження, які вимірюють вагу. Цей функціонал допомагає як вести точний облік, так і запобігати крадіжкам. Якщо хтось спробує вкрасти частину вантажу, оператор отримає інформацію про різницю в його вазі.

Для захисту вантажу також важливо встановити геозони автомобіля, тобто вручну позначити віртуальну огорожу або периметр фізичного розташування. Автомобіль можна експлуатувати лише в межах певної зони. У разі його порушення негайно надсилається сповіщення.

Іншим заходом безпеки є включення ідентифікації водія, щоб керувати транспортним засобом могла лише уповноважена особа. Щоб захистити водія, у разі небезпеки можна натиснути кнопку паніки, щоб викликати швидку допомогу. Функція моніторингу палива використовується для зчитування рівня та витрати палива автомобіля. Це дозволяє видобувним компаніям ефективніше використовувати свої ресурси та заощаджувати витрати.

FMC650 – це автомобільний трекер професійного рівня із зовнішніми антенами GNSS і GSM із високим коефіцієнтом посилення. Цей діапазон особливо підходить для обладнання для майнінгу та може забезпечити надійне підключення до мережі. За допомогою трекерів із внутрішніми антенами сигнали GPS і GSM можуть бути легко заблоковані твердими металевими поверхнями автомобіля.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Крім того, модель FMC650 розроблена для комплексних рішень, де один пристрій може виконувати кілька завдань. Такі функції, як дані CAN FMS (J1939), дані CAN палива (J1708), супутниковий модем Iridium Edge®, підключений через RS232, подвійна SIM-карта та можливість ідентифікації драйверів за допомогою iButton або RFID через 1-Wire максимально підвищують ефективність вашого автопарку.

Переваги:

– Ефективна доставка вантажів – це можливо завдяки точним даним відстеження та миттєвому отриманню інформації, навіть коли мережа GSM недоступна, щоб майнінгові компанії могли максимізувати ефективність парку.

– Максимальний захист вантажу – використання геозон, щоб гарантувати, що вантаж не покине вказану територію, і завжди знати точну вагу завдяки датчикам вагового навантаження.

– Спеціальні можливості геозонування – установіть віртуальні межі, щоб отримувати миттєві сповіщення, коли транспортні засоби відхиляються від заздалегідь визначених маршрутів або в'їжджають у заборонені зони, забезпечуючи дотримання протоколів безпеки та ефективне керування маршрутами.

– Детальна оперативна інформація – отримуйте цінну інформацію про моделі використання транспортного засобу, час простою та поведінку водія, щоб оптимізувати роботу та підвищити продуктивність.

– Планування профілактичного технічного обслуговування – відстежуйте стан транспортного засобу та його використання, щоб завчасно планувати технічне обслуговування, скорочуючи час простою та продовжуючи термін служби майнінг-парку.

– Покращені стандарти безпеки – відстежуйте поведінку водія, щоб забезпечити дотримання правил безпеки, зменшуючи ризик нещасних випадків у складних умовах шахт.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Економічне управління автопарком: оптимізуючи маршрути, скорочуючи час простою та проактивно керуючи обслуговуванням транспортних засобів, трекери FMC650 сприяють значній економії коштів, підвищуючи загальну ефективність гірничих робіт.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Функціонально система складається з наступних блоків:

- Блок диспетчерського програмного забезпечення.
- Блок моніторингу.
- Блок контролю.
- Блок аналізу.

Диспетчерське програмне забезпечення системи GPS моніторингу вантажоперевезень – це комплекс програмних модулів, що призначені для моніторингу, контролю, аналізу та оперативного управління транспортними засобами.

Диспетчерське програмне забезпечення системи GPS моніторингу вантажоперевезень надає можливість адмініструвати систему (наприклад, задавати контрольні зони чи формувати запити на отримання даних від абонентського обладнання), дозволяє зберігати отримані дані та формувати необхідні звіти про пробіг, час роботи, простої, перетин заданих районів, швидкісні режими, заправки і зливи, споживання палива, підрахунок пасажирів, спрацьовування та дані датчиків тощо.

У системі GPS моніторингу вантажоперевезень всі дані про місцезнаходження та переміщення транспортних засобів відображаються в зручному для сприйняття вигляді – на електронних картах місцевості.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Функціональні можливості програмного забезпечення GPS моніторингу вантажоперевезень

Моніторинг

– Моніторинг місцезнаходження транспортного засобу і вантажів в режимі реального часу ("on-line").

– Відображення місцезнаходження, швидкості і напрямку руху транспортного засобу на електронній карті.

– Визначення стану транспортного засобу, роботи спеціальних систем і обладнання на основі показань датчиків системи.

Контроль

– Контроль реального пробігу транспортних засобів.

– Контроль палива (заправки, зливи, витрати).

– Контроль початку і закінчення роботи, зупинок транспортного засобу.

– Контроль завантаження і розвантаження транспортного засобу.

– Контроль пасажиропотоку в громадському транспорті.

– Контроль виконання маршрутних завдань.

– Контроль мотогодин спецтехніки.

– Контроль проходження контрольних зон.

Аналіз

– Створення звітів про параметри руху (маршрути, швидкість, пробіг, зупинки, стоянки, стиль водіння і ін.).

– Створення звітів про споживання палива (заправки, зливи, витрати);.

– Створення різних спеціалізованих звітів (стан датчиків спецтехніки, мотогодин, роботи окремих вузлів і агрегатів спецтехніки тощо).

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерського проектування, наведена на рисунку 3.3.

Після початку роботи ПЗ проходить виведення головного вікна ПЗ звідкіля можна переглянути налаштування ПЗ, довідкову та через модуль захисту ПЗ та обробник помилок з використанням карт Google maps та Waze потрапити до обробки даних для отримання GPS даних та формування пакету запиту з відправлення даних використовуючи бібліотеки роботи з TCP/IP протоколом та модулем GPS.

Супутниковий моніторинг автотранспорту дозволяє визначати місцезнаходження комерційного й приватного транспорту в режимі реального часу й надає необмежені можливості по плануванню й розробці схем більше ефективної роботи, диспетчеризації й охорони автотранспорту.

Одержуючи сигнал із супутників глобальних систем позиціонування GPS, смартфон визначає своє місце розташування й передає координати по мережах стільникового зв'язку.

У випадку відсутності сигналу GSM або CDMA іде збір координат на внутрішню пам'ять пристрою або стільникового телефону.

Після того, як з'являється можливість з'єднатися з мережею Інтернет, відбувається вивантаження координат, на основі яких на інтерактивній карті відображається місце розташування об'єкта спостереження й будується маршрут його пересування.

Існуючий функціонал системи дозволяє відображати маршрут пересування об'єкта спостереження за будь-який проміжок часу, а так само точне місцезнаходження в цей момент.

Можливість побудови маршруту пересування за будь-який період часу транспортного засобу з GPS пристроєм надає істотну перевагу перед конкурентами.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

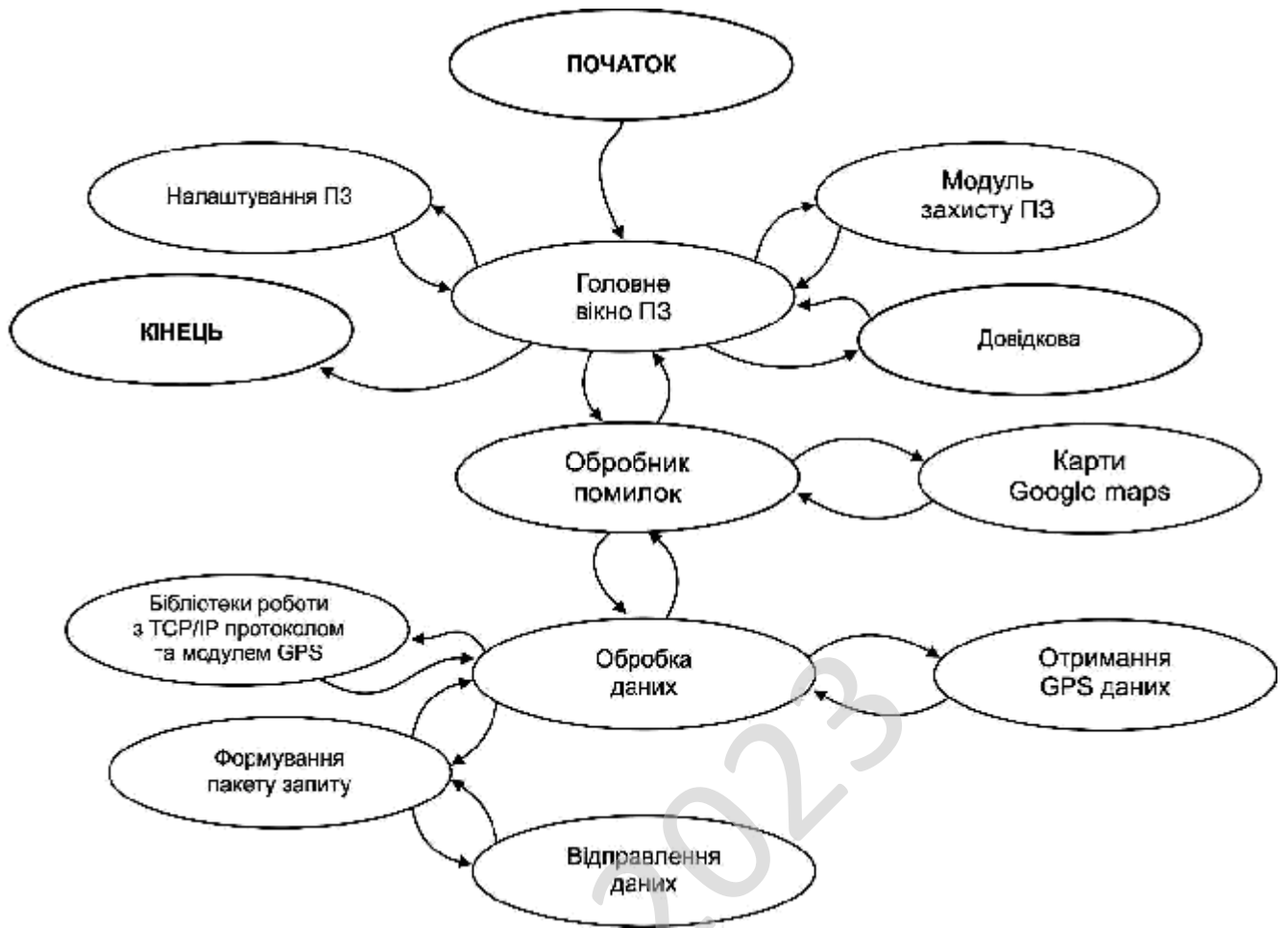


Рисунок 3.3 – Діаграма взаємодії процесів

- Файл розпаковано (запит).
- Є об'єкти стеження (запит).
- Виведення інтерфейсу ПЗ.
- Очікування запиту користувача.
- Запит пакету GPS даних.
- Є запити користувача (запит).
- Обробка дії користувача.
- Налаштування ПЗ (запит).
- Введення нових налаштувань.
- Встановлення нових значень.

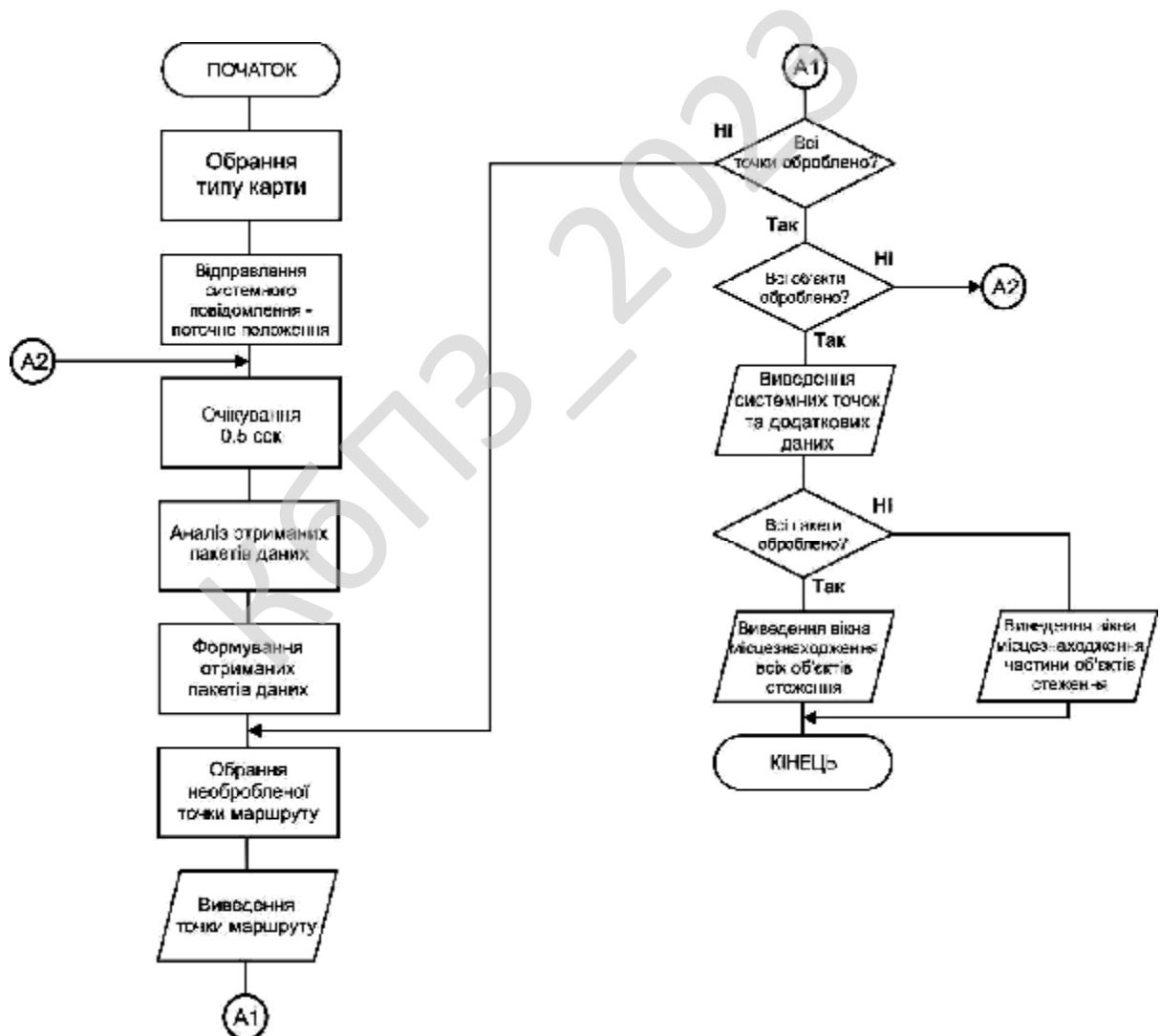


Рисунок 4.2 – Блок-схема підпрограми місцезнаходження об'єктів стеження

- Перегляд знаходження об'єктів стеження (запит).
- Підпрограма місцезнаходження об'єктів стеження – вантажівок.
- Виведення системних повідомлень.
- Сигнал aTerminate (запит).
- Є помилки (запит).
- Виведення повідомлення тіла помилки.
- Звільнення виділених ресурсів ПЗ.

На рисунку 4.2 наведено блок-схему підпрограми місцезнаходження об'єктів стеження – вантажівок. Її робота складається з виконання наступних кроків:

- Обрання типу карти.
- Відправлення системного повідомлення – поточне положення.
- Очікування 0,5 сек.
- Аналіз отриманих пакетів даних.
- Формування отриманих пакетів даних.
- Обрання необробленої точки маршруту.
- Виведення точки маршруту.
- Всі точки оброблено (запит).
- Всі об'єкти оброблено (запит).
- Виведення системних точок та додаткових даних.
- Всі пакети оброблено (запит).
- Виведення вікна місцезнаходження всіх об'єктів стеження.

Розглянемо опис алгоритмів функціонування системи

Розглянемо як було реалізоване створення карти та пошук смартфона у ПЗ.

Щоб додати карту на сторінку, створювався оброблювач події закінчення завантаження сторінки onLoad якій розмішався у тегу head після скрипта завантаження API.

```
<script type="text/javascript">
// Створює оброблювач події .onload
Waze.jquery(function () {
```

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

```
// Створює екземпляр карти й прив'язує його до
// створеного контейнера
    var map = new Waze.Мар(Waze.jquery("#Wazeid")[0]);
// Установлює початкові параметри відображення
// карти: центр карти
// і коефіцієнт масштабування
    map.setcenter(new Waze.Geopoint(37.64, 55.76), 10);
    }
</script>
```

Клас `Waze.Geocoder` дозволяє відправляти запити геокодеру, одержувати інформацію про статус і результат процесу геокодування, а також одержувати результати геокодування у вигляді міток.

Конструктор класу приймає два параметри.

1. `request` – рядок з адресою, яку потрібно геокодувати (для зворотного геокодування – точка з географічними координатами);

2. `options` – параметри:

– `boundedby` – область на карті, у якій здійснюється пошук об'єкта;

– `strictbounds` – прапор, що вказує, що шукати впливає тільки усередині області, заданою опцією `boundedby`;

– `results` – необхідна кількість результатів пошуку (не більш);

– `skip` – вказівка пропустити перші `n` результатів у відповіді;

– `prelang` – мова відповіді.

Щоб визначити координати об'єкта, створювався екземпляр класу `Waze.Geocoder` і передавався йому в якості параметра рядок з адресою об'єкта.

У прикладі нижче проводиться пошук усіх об'єктів з назвою "Кіровоград":

```
var geocoder = new Waze.Geocoder("Кіровоград");
map.addoverlay(geocoder);
```

У результаті виконання прикладу, змінна `geocoder` буде містити результати пошуку у вигляді групи спеціальних міток класу `Waze.Geocoderresult`.

Ці мітки відрізняються від звичайних міток класу `Waze.Placemark` наявністю чотирьох додаткові полів, у яких передаються додаткові відомості про знайдений об'єкт:

– `kind` – вид топоніма (вулиця, будинок, місто і т.д.);

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

- precision – передбачувана точність відповіді;
- text – адреса одним рядком;
- Addressdetails – структуроване подання адреси об'єкта у відповідності зі стандартом hxl.

Клас Waze.Geocoderresult реалізує інтерфейс Waze.Overlay, тому мітки Waze.Geocoderresult можна додавати на карту точно так само, як і звичайні.

Для обробки результатів геокодування необхідно враховувати, що геокодеру потрібен час на обробку запиту й відповідь.

Тому рекомендується завжди використовувати оброблювачі подій для одержання інформації про закінчення геокодування й про його поточний статус.

У класі Waze.Geocoder передбачено дві події: Load (геокодування пройшло без помилок) і Fault (з помилками).

В оброблювачі події Load можна оперувати результатами геокодування як групою.

Груповання об'єктів. Приклад нижче ілюструє обробку подій геокодера реалізованого у роботі.

У випадку успішного геокодування користувачеві карти буде показане повідомлення про кількість знайдених об'єктів, а самий релевантний (перший) з них буде показаний на карті.

```
Waze.Events.observe(geocoder, geocoder.Events.Load, function () {
    if (this.length()) {
        alert("Знайдене :" + this.length());
        map.addoverlay(this.get(0));
        map.panto(this.get(0).getgeopoint())
    }else {
        alert("Нічого не знайдене")
    }
});
Waze.Events.observe(geocoder, geocoder.Events.Fault,
function (error) { alert("Відбулася помилка: " + error.message) });
```

Параметр preflang геокодера дає можливість контролювати мову результатів пошуку.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Цей параметр дозволяє задати мову, на якій буде проводитися видача результатів геокодування. Код мови вказується у відповідності зі стандартом ISO 639.

Список можливих значень:

– uk Україна.

Якщо встановити параметр `preflang` у значення “uk”, то в результаті пошуку адреси «Кропивницький, проспект Університетський 8» буде отриманий результат – «Центральноукраїнський національний технічний університет».

```
var geocoder = new Waze.Geocoder
("Кіровоград, проспект Університетський 8",
{ preflang : "uk" } );
Waze.Events.observe(geocoder, geocoder.Events.Load, function (geocoder) {
    map.addoverlay(geocoder.get(0));
    map.setbounds(geocoder.get(0).getbounds());
});
```

Об'єкт `Waze.Geocoder` підтримує зворотне геокодування, за допомогою якого можна перетворювати координати об'єкта в поштову адресу.

Для зворотного геокодування у параметрі `request` потрібно передати замість адреси геоточки (`Waze.Geopoint`).

```
// Запускає процес геокодування
var geocoder = new Waze.Geocoder
(new Waze.Geopoint(37.588395, 55.762718), {results: 1});
// Оброблювач успішного завершення процесу геокодування
Waze.Events.observe(geocoder, geocoder.Events.Load, function () {
    if (this.length()) {
        map.addoverlay(this.get(0));
        map.panto(this.get(0).getgeopoint())
    }else {
        alert("не знайдене")
    }
});
// Оброблювач невдалого завершення геокодування
Waze.Events.observe(geocoder, geocoder.Events.Fault,
function (geocoder, error) {
    alert("Відбулася помилка: " + error.message)
});
```

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Служба «Waze» має програмний інтерфейс (API) на Javascript, який дозволяє інтегрувати інтерактивні карти у веб-сторінки.

За допомогою Javascript API можливо:

- управляти картою й обробляти події;
- показувати фрагменти карти й супутникових знімків;
- показувати на карті різні об'єкти (мітки, лінії).

Щоб мати можливість користуватися API Waze, необхідно одержати унікальний API ключ.

API стабільно працює з наступними браузерами:

- EDGE.
- Mozilla Firefox.
- Opera 9.x і вище.
- Apple Safari 3.1 і вище.
- Google Chrome.

API Ключ – унікальна цифро-буквенна комбінація, по якій сервіс «Waze» ідентифікує ПЗ, що звертається до API.

Щоб одержати API ключ, слід зареєструватися на Яндексі, у спеціальній формі одержання ключа ознайомитися й прийняти умови користувацької угоди, а також указати програму, на якій буде використовуватися API. Ключ можна закріпити за будь-якою програмою.

Для однієї програми може бути отримано кілька ключів, причому не обов'язково, щоб усі ключі були зареєстровані на один обліковий запис.

В API Waze головним об'єктом є карта. Інші об'єкти лише тим або іншим способом взаємодіють із картою, змінюючи її зовнішній вигляд, поведінку, або доповнюють карту новою функціональністю.

Карта складається з одного або декількох накладених один на одного шарів. Кожний шар являє собою область, набрану з безлічі прозорих (або непрозорих) тайлов.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Один або кілька шарів можуть бути об'єднані в тип карти. Тип карти може мати додаткові параметри (наприклад, мінімальне й максимальне значення коефіцієнта масштабування), тип карти може бути доданий у перемикач типів карт.

Географічна карта земної поверхні має три вбудовані типи, які можна перемикати між собою, наприклад, за допомогою перемикача типів карт:

- звичайна карта місцевості (MAP);
- карта, складена із супутникових знімків земної поверхні (SATELLITE);
- гібридна супутникова карта з дорожньою мережею, границями держав і назвами географічних об'єктів (HYBRID).

У якості шарів можуть використовуватися зображення географічних і інших карт Землі й планет, супутникові знімки, рисунки й схеми, плани будинків і споруджень і просто фотографії.

Взаємодія з картою здійснюється за допомогою елементів керування, які дозволяють користувачеві змінювати більшість параметрів карти.

Об'єкти-оверлеи (мітки) додаються на карту поверх усіх шарів. Об'єкти прикріплюються до певного місця на карті й зберігають цю позицію при зміні масштабу й переміщенні карти.

Для навігації й для вказівки місця розташування об'єктів на карті використовується відповідна система координат.

Для географічних карт – це геодезична система координат WGS-84, де для вказівки положення об'єкта на карту досить указати його довготу й широту.

Для навігації по кресленнях і планам місцевості також може використовуватися декартова система координат.

Можливості карти також можуть бути розширені за допомогою сервісів геокодування, прокладки маршруту й візуалізації Wazeml.

Wazeml (Waze Markup Language) – Xml мови, призначена для опису географічних даних на картах Яндекс.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

За допомогою Wazeml можливо в текстовому виді задати ділянку карти з нанесеними мітками й іншими об'єктами, а потім відобразити його в браузері засобами JS API Waze.

Сервіс геокодування служби Waze и використовує Wazeml для опису знайдених по запиту географічних об'єктів.

Для візуалізації Wazeml документів засобами JS API Waze потрібно вказувати Арі ключ.

Wazeml (рис. 4.3) використовується як формат передачі даних між сервісами Waze і сторонніми програмними засобами.

Дані у форматі Wazeml можуть бути відображені на Waze, у цьому ж форматі повертає інформацію про місце розташування об'єктів сервіс геокодування.

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7624:2014 («Калина»). Одним із основних алгоритмів симетричного блокового шифрування, що використовуються в Україні, є ДСТУ 7624:2014 («Калина»), який визначає сучасний алгоритм симетричного блокового перетворення для забезпечення конфіденційності й цілісності інформації при її обробці та встановлює режими його роботи. В алгоритмі шифрування даних «Калина» використовуються криптографічні перетворення, які відповідають сучасним вимогам до рівня криптостійкості та швидкодії.

Даний стандарт розроблено з урахуванням існуючих та потенційних загроз, подальшого інтенсивного розвитку інформаційних технологій та необхідності активного використання протягом кількох наступних десятиліть.

Стандарт блокового симетричного шифрування ДСТУ 7624:2014 визначає десять різних режимів роботи, що широко поширені відповідно до міжнародних стандартів ISO/IEC 10116:2006.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Це спрямовано на забезпечення широкого застосування ДСТУ 7624:2014, у тому числі для розробки мобільних додатків, що передається комп'ютерними мережами, прозорого шифрування жорстких дисків і змінних носіїв, електронних документів, ключових даних.

Ефективність реалізації систем, засобів та протоколів криптографічного розробки мобільних додатків в інформаційно-телекомунікаційних системах різного призначення може бути забезпечена саме наявністю такої кількості режимів роботи алгоритму.

До блокового шифру «Калина» ставляться такі вимоги: високий рівень криптографічної стійкості з достатнім запасом у разі появи нових атак протягом тривалого часу; висока швидкодія програмної реалізації на сучасних та перспективних платформах; компактність програмної та програмно-апаратної реалізації; можливість ефективної інтеграції декількох алгоритмів в одному засобі криптографічного захисту; прозорість проектування, консервативний підхід до забезпечення стійкості; вища (або однакова) ефективність порівняно з найкращими світовими рішеннями.

Криптографічні алгоритми, які визначаються стандартами ДСТУ 7624:2014 і ДСТУ 7564:2014, є гнучкими, підтримують розмір блоку і довжину ключа від 128 до 512 біт.

Стандарт симетричного блокового шифрування «Калина» є результатом багаторічної плідної співпраці Державної служби спеціального зв'язку та розробки мобільних додатків України та провідних українських вчених.

Даний алгоритм шифрування враховує досвід і результати проведення міжнародних та відкритих національних конкурсів криптографічних алгоритмів.

Алгоритм ДСТУ 7624:2014 забезпечує досить високий рівень криптостійкості порівняно з міжнародним стандартом AES (ISO/IEC 18033-3:2010), оскільки дає можливість застосовувати блок даних і ключ шифрування розміром аж до 512 біт.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що ПЗ працює під керуванням ОС Android. ПЗ використовує вільно розповсюджені карти Google Maps.

Для відстеження руху вантажівок необхідно завантажити ПЗ та обрати необхідну вантажівку, далі все проходить в автоматичному режимі. Розроблене ПЗ відсилає запит на смартфон який знаходиться на вантажівці через мобільну мережу та отримує поточні його значення. Ці значення зберігаються на смартфоні створюючи маршрут руху який у подальшому можна переглянути у карті Google.

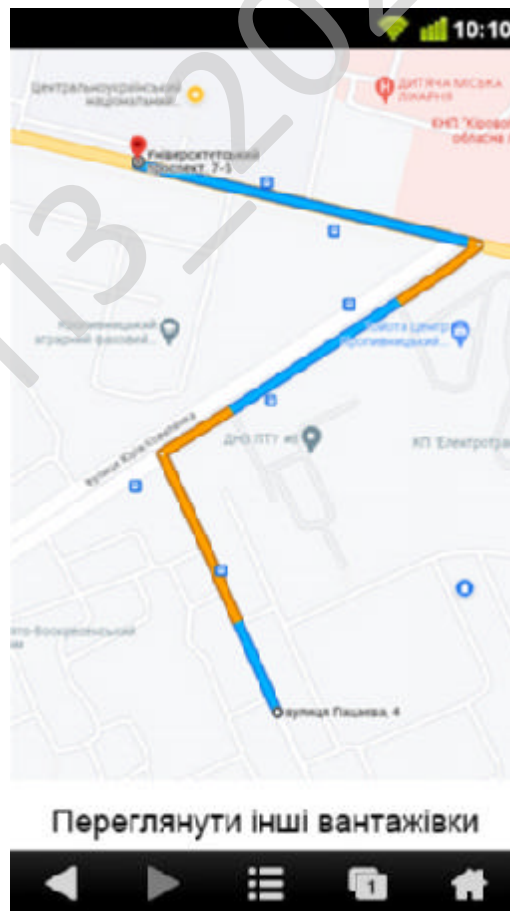


Рисунок 5.1 – Головне вікно програми

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

На рисунку 5.2 зображено форму авторського права, де відображені дані розробника. Було обрано Shareware умову розповсюдження. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (неповнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.



Рисунок 5.2 – Довідка

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мобільного контролю вантажоперевезень на базі ОС Android.

Метою розробки є дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

Об'єктом дослідження є процес мобільного контролю вантажоперевезень на базі ОС Android.

Предметом дослідження є методи мобільного контролю вантажоперевезень на базі ОС Android.

Методи дослідження базуються на методах розробки мобільних додатків, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод мобільного контролю вантажоперевезень на базі ОС Android.

– Розроблено вітчизняний продукт мобільного контролю вантажоперевезень на базі ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 24 днів (один місяць).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	180
3. Запланований термін розробки, днів	Frq	24 (1 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	В
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
9. Мова програмування (1-6)	–	1
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	1
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	3
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	4
17. Складність кінцевого програмного продукту (1-6)	–	5
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	3
20. Вимоги до швидкодії ПП (1-6)	–	3
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	4
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	1
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	18000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	40
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 4,22 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,2^{1,027} = 5,5 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 5,5 \cdot (1 \cdot 1,09 \cdot 1,30 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 1,15 \cdot 1 \cdot 0,87 \cdot 1,10 \cdot 1,22 \cdot 1,12 \cdot 1,10 \cdot 1 \cdot 1 \cdot 1,10) = 12,9 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 12,9^{0,33 + 0,2(1,027 - 1,01)} \cdot 70 = 131 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	70	175	3
Кабельне господарство електромереж	48	50	2400	40
Копіювальний апарат	285	1	285	5
Усього за рік:			3 _ч	221

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{221 \cdot 1}{1,2} = 184,1 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 184,1 / (24 \cdot 8) = 1 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	2	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	13500	13500
Продакт-менеджер	0,5	12500	6250
Інженер-програміст	8,6	12000	103200
Інженер-електронщик	1	12000	12000
Інженер-системотехнік	0,25	12000	3000
Адміністратор мережі	0,5	12000	6000
Системний програміст	0,25	12000	3000
Дизайнер WEB	0,5	13000	6500
Інженер-верстальник	0,25	11700	2925
Бухгалтер-економіст	0,5	12500	6250
Всього за період розробки	$R_{cn} = 13,35$	-	$\Phi_{роб} = 162625$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{162625}{13,35 \cdot 24} = 508 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{y\delta} = R_{cn}^1 S_y \Pi_{пл}, \quad (7.9)$$

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;
 S_y – питома площа на одне робоче місце, m^2 ;
 $C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 200...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно $8 m^2$. З урахуванням цього:

$$B_{y\partial} = 13 \cdot 8 \cdot 20000 = 2080000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 208000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 13 \cdot 3500 = 45500 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом фірми brain від 17.10.23 – джерело <https://brain.com.ua>.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11457
Системний блок		7509
Процесор	AMD Athlon™ 3000G (YD3000C6M2OFH) AM4, 2 ядра, 4 потоки, 3.5 GHz, TDP - 35 Вт, 14nm	-
Системна плата	GIGABYTE A520M H DDR4, 5100 MHz, LAN - 1 Гбит/с, DVI, HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX	-
Відеокарта	Radeon R7 350 2Gb	-
Жорсткий диск	SSD M.2 2280 128GB Silicon Power (SP128GBSS3A55M28) Серія - A55, 128 GB, 3D TLC NAND, M.2, SATA III (6Gb/s)	-
Жорсткий диск	3.5" 500Gb Seagate 500 GB, 7200 об/хв, 16 MB, SATA III, Barracuda 7200.12	-
Оперативна пам'ять	DDR4 8GB 2666 MHz Kingston	-
Корпус	Vinga Apache-500W	-
Кулер	Vinga Q4	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-
інше	Клавіатура, мишка	-
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37 Photo	2970
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1348

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	13	11457	14894,1	163835,1
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2970	297	3267
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	185653,6

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	2080000	-	-
2. Передавальні пристрої	208000	-	-
Всього по групі	2288000	5	114400
Група 4			
3. Обчислювальна техніка	185654	-	-
Всього по групі	185654	50	92827
Група 5, 6			
4. Вимірювальні пристрої	3999	25	999,75
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	45500	25	11375
Всього по групі	146999	-	31874,75
7. Нематеріальні активи	18000	10	1800
Разом	$K_p = 2666153$		$A_p = 250476,75$

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 97500 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 508 \cdot 180 / 180 = 508 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 508 \cdot 10 \cdot 0,01 = 51 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(508+51) = 123 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 508 \cdot 15 \cdot 0,01 = 76 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджів, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Згідно виданих норм приймаємо 0,25 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 160$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 160 \cdot 0,25 \cdot 1 = 40 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості 100 примірників та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 16 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 32 грн./шт.

$$Z_{M2} = 100 \cdot 16 + 32 = 1632 \text{ грн.}$$

Згідно виданих норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (40 + 1632 + 1702) / 180 = 18,7 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 508 \cdot 15 \cdot 0,01 = 76 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 180$ прим.):

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 250477 \cdot 1 / (180 \cdot 12) = 116 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 508 + 51 + 123 + 76 + 18,7 + 76 + 116 = 968,7 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 40%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 40 \cdot 968,7 = 378 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	508
2. Додаткова зарплата виконавців	Z_d	51
3. Відрахування на соціальні потреби	C_{oc}	123
4. Загальногосподарські витрати	Γ_{ocn}	76
5. Витрати на матеріали	Z_m	18,7
6. Освоєння нових операційних систем, мов програмування	O_n	76

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	116
8. Повна собівартість програмного забезпечення	C_n	968,7
9. Плановий прибуток	P_p	378
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1346,7
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	269,3
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	1616

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	1616
Всього капітальних витрат	–	1616

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування системи	Z_p	33550	20130
2. Витрати на електроенергію	$Z_{ел}$	650	542
3. Витрати на амортизацію	$Z_{ам}$	0	404
Всього витрат за рік	I	34200	21076

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин на обслуговування системи зменшилася з 250 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 250 \cdot 100 \cdot 1,1 \cdot 1,22 = 33550 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 150 \cdot 100 \cdot 1,1 \cdot 1,22 = 20130 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел} \quad (7.24)$$

$$Z_{ел\ баз} = 0,15 \cdot 1140 \cdot 3,8 = 650 \text{ грн.}$$

$$Z_{ел\ нов} = 0,15 \cdot 950 \cdot 3,8 = 542 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	1616	–	404
Всього відрахувань	-	–	1616	–	404

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (Ц_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (1346,7 - 968,7) \cdot 180 - (0,05 \cdot 2288000 + 0,5 \cdot 185654 + 0,25 \cdot 49499 + 0,2 \cdot 97500 + 0,1 \cdot 18000) \cdot 1/12 = 47968 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(Ц_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p – балансова вартість основних фондів розробника.

$$T_s = \frac{2666153}{(1346,7 - 968,7) \cdot 180 \cdot 12 / 1} = 3,3 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	180
2. Повна собівартість розробленої програми	Грн.	968,7
3. Ціна розробленої програми	Грн.	1346,7
4. Плановий прибуток від реалізації розробленої програми	Грн.	378
5. Рентабельність програмної продукції	%	40
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2666153
7. Загальний прибуток від реалізації програмної продукції	Грн.	68040
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	47968
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	3,3
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1616
11. Величина економічного ефекту у користувача програмної продукції	Грн.	12720
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де: I_{δ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

K_{δ} , K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (34200 - 21076) - 0,25 \cdot 1616 = 12720 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\delta}}{I_{\delta} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{1616}{34200 - 21076} = 0,1 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м’язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машин (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

- інтелектуальні навантаження;
- монотоність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно Правил улаштування електроустановок (ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За пожежо-вибухонебезпеку приміщення відноситься до класу В. В таблиці 8.1 наведена загальна характеристика приміщення щодо вибухопожеженобезпеки та важкістю робіт.

Таблиця 8.1 – Загальна характеристика приміщення щодо вибухопожеженобезпеки та важкістю робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1 -2002
В – пожежонебезпечне клас П – II	Звичайне без ознак хімічного забруднення та нормальної вологості і за санітарними нормами	1а.....до 139 Вт/м ² 1б.....до 140-174 Вт/м ² Клас умов праці – оптимальний

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії І-а. Людиною в цьому випадку виділяється до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [1].

За результатами виміру освітленості величина освітленості від системи загального штучного освітлення дорівнює 310 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [2] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [4, 10, 11].

8.5 Розрахунок занулення

Розрахунок занулення.

Розрахунок занулення складається з трьох частин:

1. Розрахунок на відключаючу спроможність.
2. Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус.
3. Розрахунок робочого і повторного заземлювачів.

Початкові дані:

1. Потужність електроприладів, які підлягають зануленню: $P = 7$ кВт.
2. Довжина магістрального кабеля: $L_M = 12$ м.
3. Довжина розгалуження (від розподільного щита до електроприладів) : $l = 25$ м.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва вцілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок занулення, рекомендовано заходи з охорони праці.

Список використаних джерел інформації

1. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>
6. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>
7. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

3.3.6.042-99. – Режим доступу до ресурсу:

<https://zakon.rada.gov.ua/rada/show/va042282-99>

8. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

9. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://cpo.stu.cn.ua>

10. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

КБПЗ – 2023

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи мобільного контролю вантажоперевезень на базі ОС Android.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів мобільного контролю вантажоперевезень на базі ОС Android.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем мобільного контролю вантажоперевезень на базі ОС Android.
- Досліджена система мобільного контролю вантажоперевезень на базі ОС Android.
- На основі отриманих результатів досліджень створена програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання мобільного контролю вантажоперевезень на базі ОС Android.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Java, Eclipse, Android SDK. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Android.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7624:2014.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 12720 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Серета В.Л. Дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. M. Holmes He. Creating Apps with React Native. Apress Media. 2022. 445 p.
3. Maurício Aniche. Effective Software Testing. Manning Publications. 2021. 372 p
4. Priscila Heller. Automating Workflows with GitHub Actions. Packt Publishing. 2021. 216 p.
5. JJ Geewax. API Design Patterns. Manning Publications Co. 2021. 481 p.
6. Prateek Prasad. App Design Apprentice. Razeware LLC. 2020. 272 p.
7. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.
8. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.
9. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.
10. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.
11. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
12. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

13. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.
14. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
15. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
16. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.
17. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.
18. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143
19. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.
20. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.
21. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

22. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

23. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

24. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

25. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

26. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

27. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

28. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

29. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

30. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

31. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

32. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

33. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

34. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

35. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

36. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

37. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

38. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

39. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнотраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

40. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

41. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

відновлення та зміцнення поверхонь деталей. Центральнуукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

42. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

43. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

44. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

45. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

46. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSН-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

47. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA – 2017.

48. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи

управління, навігації та зв'язку. – Випуск 4 (44). – Полтава: ПолтНТУ. – 2017. – С. 112-115.

49. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів .Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

50. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

51. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). – Харків: ХУПС. – 2016. – С.96-100.

52. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". – Випуск 8 (145). – Х.: ХУПС – 2016. – С. 77-80.

					ВКРМ-122.23.0045.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-122.23.0045.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Серета В.Л.				<i>Дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android</i>		
Перевірів	Якименко Н.М.						
Н. Контр.	Коваленко А.С.				Літ.	Аркуш	Аркушів
Затв.	Смірнов О.А.				М	1	6
					ЦНТУ КН-22М-2		

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи мобільного контролю вантажоперевезень на базі ОС Android.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 33-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи мобільного контролю вантажоперевезень на базі ОС Android.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-122.23.0045.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи мобільного контролю вантажоперевезень на базі ОС Android;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-122.23.0045.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на мобільному пристрої під керуванням ОС Android і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Android.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Java, Eclipse, Android SDK.

					ВКРМ-122.23.0045.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий розрахунок занулення.

					ВКРМ-122.23.0045.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 92 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 19.12.2023 р.

					ВКРМ-122.23.0045.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Якименко Н.М.

*Дослідження та програмна реалізація
системи мобільного контролю вантажоперевезень на базі ОС Android*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 39

Літера: РП

Кропивницький – 2023 року

РОБОТА З БАЗОЮ ДАНИХ

```

package com.android.Cargo_Control_GPS;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

/**
 * Цей клас дозволяє відкривати, створювати і оновлювати базу даних
 * Тип запитів - SQL
 */

public class DatabaseHelper extends SQLiteOpenHelper {
    private SQLiteDatabase dbCargo_Control_GPS;
    DatabaseHelper(Context context) {
        super(context, Cargo_Control_GPS.DATABASE_NAME, null,
            Cargo_Control_GPS.DATABASE_VERSION);
    }
    private void doDatabase(SQLiteDatabase db,int fromVersion) {
        if (fromVersion < 6) {
            db.execSQL("CREATE TABLE Trips (id integer primary key AUTOINCREMENT,start_date
            timestamp,stop_date timestamp);");
            db.execSQL("CREATE TABLE TripsLogs (id integer primary key AUTOINCREMENT,tripId
            integer,stamp timestamp,position text,longitude double,latitude double,altitude
            double);");
        }
        if (fromVersion < 7) {
            db.execSQL("ALTER TABLE Trips ADD trip_meter float;");
            db.execSQL("ALTER TABLE Trips ADD trip_duration long;");
        }
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        doDatabase(db,0);
    }
    @Override
    // оновлення
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        if (oldVersion < 6) {
            db.execSQL("DROP TABLE IF EXISTS Trips");
            db.execSQL("DROP TABLE IF EXISTS TripsLogs");
        }
        if (oldVersion > newVersion) {
            onCreate(db);
        } else {
            doDatabase(db, oldVersion);
        }
    }
    public void startDatabase() {
        dbCargo_Control_GPS = getWritableDatabase();
    }

    public int startAnewTrip() {
        int current_trip;
        Cursor lst = dbCargo_Control_GPS.rawQuery("select max(id) from Trips",null);
        if (lst.moveToFirst()) {
            current_trip = lst.getInt(0) + 1;
        } else {
            current_trip = 1;
        }
        dbCargo_Control_GPS.execSQL("INSERT INTO Trips (id,start_date) VALUES (" +
        current_trip +",current_timestamp);");
        return current_trip;
    }
    public void deleteTrip(long delete_id) {
        dbCargo_Control_GPS.execSQL("delete from tripslogs where tripid = " +
        delete_id);
    }

```

```

        dbCargo_Control_GPS.execSQL("delete from trips where id = " + delete_id);
    }
// видалення
    public void deleteAllTrip() {
        dbCargo_Control_GPS.execSQL("delete from tripslogs");
        dbCargo_Control_GPS.execSQL("delete from trips");
    }
    public Cursor getAllTripId() {
        return dbCargo_Control_GPS.rawQuery("select id from trips",null);
    }
    public Cursor getTripId(long tripId) {
        return dbCargo_Control_GPS.rawQuery("select id from trips where id = "+
tripId,null);
    }
    public Cursor getTripData(long id) {
        return dbCargo_Control_GPS.rawQuery("select
latitude,longitude,altitude,strftime('%Y-%m-%dT%H:%M:%SZ',stamp) as stamp from
tripslogs where tripid = "+ id,null);
    }
// лічильник
    public int getAllTripTrkCount() {
        Cursor l = dbCargo_Control_GPS.rawQuery("select count(*) from trips t join
tripslogs l on (t.id=l.tripId)",null);
        l.moveToPosition(0);
        return l.getInt(0);
    }
    public int getTripTrkCount(long tripId) {
        Cursor l = dbCargo_Control_GPS.rawQuery("select count(*) from tripslogs
where tripId ="+ tripId,null);
        l.moveToPosition(0);
        return l.getInt(0);
    }
    Нова точка
    public void newPointDB(double longitude,double latitude,double altitude,long
gps_time,int current_trip) {
        dbCargo_Control_GPS.execSQL("INSERT INTO TripsLogs
(tripId,stamp,position,longitude,latitude,altitude) VALUES ("+ current_trip
+",datetime("+gps_time+"/1000,
'unixepoch'),current_timestamp,"+longitude+", "+latitude+", "+altitude+");");
    }
// стоп
    public void stopTrip(long last_timestamp,long first_timestamp,float
total_distance,int current_trip) {
        dbCargo_Control_GPS.execSQL("UPDATE Trips set stop_date =
current_timestamp,trip_duration = "+ ( last_timestamp - first_timestamp
)+",trip_meter = "+total_distance+" where id = " + current_trip);
    }
    public Cursor getListing() {
        return dbCargo_Control_GPS.rawQuery("select t.id as _id,t.id||' -
'||start_date as cmt,count(tl.id)||' positions,'||coalesce(trip_meter,'?')||'
meters'||','||coalesce((trip_duration/1000),'?')||' seconds' as descr from Trips
t left outer join TripsLogs tl on (t.id=tl.tripId) group by t.id,start_date
order by start_date desc",null);
    }
    public void end() {          dbCargo_Control_GPS.close();          } // завершення
}

```

ОБРОБКА ФАЙЛІВ

```

package com.android.Cargo_Control_GPS;
import java.io.IOException;
import android.app.ProgressDialog;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.util.Log;

class ExportThread extends Thread {
    private ProgressDialog mProgress;
    private Cargo_Control_GPS aThis;
    private long tripId = -1;
    private boolean email = false;
    public ExportThread(Cargo_Control_GPS a, long t, ProgressDialog m, boolean b)
{
    super();
    mProgress = m;
    tripId = t;
    aThis = a;
    email = b;
}

public void run() {
    try {
        String fname;
        Cursor ltrip;
        int cntProgress;

        if (tripId == -1) {
            fname = "/sdcard/all-Cargo_Control_GPS.gpx";
            ltrip = aThis.mCargo_Control_GPSDB.getAllTripId();
            cntProgress = aThis.mCargo_Control_GPSDB.getAllTripTrkCount();
        } else {
            fname = "/sdcard/" + tripId + "-Cargo_Control_GPS.gpx";
            ltrip = aThis.mCargo_Control_GPSDB.getTripId(tripId);
        }
        cntProgress = aThis.mCargo_Control_GPSDB.getTripTrkCount(tripId);
    }
    GpxFileWriter fw = new GpxFileWriter(fname, false);
    int cnt = ltrip.getCount();
    mProgress.setProgress(0);
    mProgress.setMax(cntProgress);

    for(int i=0; i<cnt; i++) {
        if (!isInterrupted()) {
            ltrip.moveToPosition(i);
            long id = ltrip.getLong(0);
            fw.startTrk("Trip "+id);
            Cursor l = aThis.mCargo_Control_GPSDB.getTripData(id);
            fw.fromCursorString(l, mProgress, this);
            fw.stopTrk();
        } else {
            Log.w("Cargo_Control_GPS", "Interrupted");
        }
    }

    fw.close();
    if (email && !isInterrupted()) {
Intent i= new Intent(android.content.Intent.ACTION_SEND);
        i.setType("image/jpeg");

i.putExtra(android.content.Intent.EXTRA_SUBJECT, "The gpx file");

i.putExtra(android.content.Intent.EXTRA_STREAM,

```

```
        Uri.parse("file://" + fname));  
  
        aThis.startActivity(Intent.createChooser(i, "Відправлення даних gps..."));  
    }  
    } catch (IOException e) {  
        e.printStackTrace();  
  
        Log.w("Cargo_Control_GPS", "Unable to save");  
    }  
    mProgress.dismiss();  
  
    mProgress.setProgress(0);  
}  
  
}
```

КБПЗ_2023

ФОРМУВАННЯ ОТРИМАНИХ ДАНИХ GPS

```

package com.android.Cargo_Control_GPS;
import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.NoSuchElementException;
import android.app.ProgressDialog;
import android.database.Cursor;
public class GpxFileWriter extends FileWriter {
    protected SimpleDateFormat formatter;
    private boolean startT = false;

public GpxFileWriter(String string, boolean start_trk) throws IOException {

super(string);
// формат дати
formatter = new SimpleDateFormat("yyyy-MM-dd'T'hh:mm:ss'Z'");
// використання xml
write("<?xml version=\"1.0\" encoding=\"UTF-8\"?><gpx version=\"1.1\"
creator=\"Cargo_Control_GPS\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-
instance\" xmlns=\"http://www.topografix.com/GPX/1/1\"";
    startT = start_trk;
    if (startT) {
        startTrk("default");
    }
}

// запис
public GpxFileWriter(String string) throws IOException {
    this(string,true);
}
public void startTrk(String name) throws IOException {
    write("<trk><name>"+ name + "</name> <trkseg>");
}

public void stopTrk() throws IOException {
    write("</trkseg></trk>");
}

public void write(float lat,float lon,float alt,String timestamp) throws
IOException {
    write("<trkpt lat=\""+lat+"\"
lon=\""+lon+"\"><ele>"+alt+"</ele><time>"+timestamp+"</time></trkpt>\n");
}

public void write(float lat,float lon,float alt,long timestamp) throws
IOException {
    Date t = new Date(timestamp);
    write(lat,lon,alt,formatter.format(t));
}

public void fromCursorTimeStamp(Cursor l,ProgressDialog pDlg,Thread
current) throws IOException,NoSuchElementException {
/*
longitude, latitude, altitude, gps time, current trip
широта, довгота, висота, GPS час, поточний час поїздки
*/

    int lat_pos = l.getColumnIndex("latitude");
    int lon_pos = l.getColumnIndex("longitude");
    int alt_pos = l.getColumnIndex("altitude");
    int stamp_pos = l.getColumnIndex("stamp");
    int cnt = l.getCount();
    for(int z=0;z<cnt;z++){
        if (!current.isInterrupted()) {
            pDlg.incrementProgressBy(1);
            l.moveToPosition(z);
write(l.getFloat(lat_pos),l.getFloat(lon_pos),l.getFloat(alt_pos),l.getLong(stam
p_pos));
        }
}
}
}

```

```
    }  
}  
  
public void fromCursorString(Cursor l,ProgressDialog pDlg,Thread current) throws  
IOException,NoSuchElementException {  
    int lat_pos = l.getColumnIndex("latitude");  
    int lon_pos = l.getColumnIndex("longitude");  
    int alt_pos = l.getColumnIndex("altitude");  
    int stamp_pos = l.getColumnIndex("stamp");  
    int cnt = l.getCount();  
    for(int z=0;z<cnt;z++){  
        if (!current.isInterrupted()) {  
            pDlg.incrementProgressBy(1);  
            l.moveToPosition(z);  
  
            write(l.getFloat(lat_pos),l.getFloat(lon_pos),l.getFloat(alt_pos),l.getStr  
ing(stamp_pos));  
        }  
    }  
}  
  
public void close() throws IOException {  
    if (startT) {  
        stopTrk();  
    }  
    write("</gpx>");  
    super.close();  
}  
}
```

K6П3_2023

ОСНОВНА ПРОГРАМА CARGO_CONTROL_GPS

```

package com.android.Cargo_Control_GPS;
import java.util.Date;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.database.Cursor;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.PowerManager;
import android.util.Log;
import android.view.ContextMenu;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.ContextMenu.ContextMenuInfo;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.AdapterContextMenuInfo;
public class Cargo_Control_GPS extends Activity implements LocationListener {
// обрання файлу базу даних
    static final String DATABASE_NAME = "Cargo_Control_GPS.db";
    static final int DATABASE_VERSION = 7;
    public DatabaseHelper mCargo_Control_GPSDB;
// об'яви
    private TextView gpsstateTextView;
    protected LocationManager myLocationManager = null;
    private int current_trip = 0;
    private int current_point = 0;
    private ListView listview_trace;
    private SimpleCursorAdapter sc_adapter;
    private Location last_loc;
    private float total_distance;
    private long first_timestamp = 0;
    private long last_timestamp = 0;
    private ExportThread mExportThread;
    private static final int MENU_COPY = 1;
    private static final int MENU_START = 2;
    private static final int MENU_STOP = 3;
    private static final int MENU_DELETE_ALL = 4;
    private static final int MENU_EXPORT_ALL = 5;
    private static final int MENU_EXPORT_SEND_ALL = 6;
    private MenuItem start_menu;
    private MenuItem stop_menu;
    private MenuItem delete_all_menu;
    private MenuItem export_all_menu;
    private MenuItem export_send_all_menu;

    private ProgressDialog mProgressDialog;
    private AlertDialog mDeleteDialog;

    protected PowerManager.WakeLock wlatp;
// Викликається при першому створенні.
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_PROGRESS);
        setContentView(R.layout.main);
        Log.w("Cargo_Control_GPS", "1");
        listview_trace = (ListView) findViewById(R.id.traceView);

```

```

        registerContextMenu(listview_trace);
        gpsstateTextView = (TextView) findViewById(R.id.gpsstate);
        gpsstateTextView.setText(R.string.tips_not_running);
        mCargo_Control_GPSDB = new DatabaseHelper(this);
        mCargo_Control_GPSDB.startDatabase();
        PowerManager pm = (PowerManager)
getSystemService(Context.POWER_SERVICE);
        wlatp = pm.newWakeLock(PowerManager.FULL_WAKE_LOCK, "Cargo_Control_GPS tag");
        fillList();
    }
    // Створення контекстного меню
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo
        menuInfo)
    {
        menu.setHeaderTitle(R.string.c_menu_title);
        menu.setHeaderIcon(R.drawable.satellite);
        menu.add(Menu.NONE, 0, 0, R.string.c_menu_send);
        menu.add(Menu.NONE, 1, 1, R.string.c_menu_save);
        menu.add(Menu.NONE, 2, 2, R.string.c_menu_delete);
    }

    // виклик вибраного розділу меню
    public boolean onOptionsItemSelected(MenuItem item) {
        AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
        long id = sc_adapter.getItemId(info.position);
        switch(item.getItemId()) {
            case 0: // Відсилка даних
                exportTrip(id, true);
                break;
            case 1: // Збереження даних
                exportTrip(id, false);
                break;
            case 2: // Видалення даних
                exportTrip(id, false);
                deleteTrip(id);
                fillList();
                break;
        }
        return true;
    }
    @Override
    // при паузі
    protected void onPause() {
        super.onPause();
        Log.w("Cargo_Control_GPS", "onpause");
    }
    @Override
    // продовження обробки
    protected void onResume() {
        super.onResume();
        Log.w("Cargo_Control_GPS", "onresume");
    }
    // завершення обробки
    public boolean isFinishing() {
        return false;
    }
    @Override
    // при несподіваному завершенні - Stop
    protected void onStop() {
        super.onStop();
        Log.w("Cargo_Control_GPS", "onstop");
    }

    @Override
    // при початку роботи
    protected void onStart() {
        super.onStart();
        Log.w("Cargo_Control_GPS", "onstart");
    }
}

```

```

@Override
protected void onDestroy() {
    super.onDestroy();
    if (wlatp.isHeld()) {
        wlatp.release();
    }
    myLocationManager =
        (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    myLocationManager.removeUpdates(this);
    mCargo_Control_GPSDB.end();
    Log.w("Cargo_Control_GPS", "ondestroy");
}
@Override
// при перезавантажені
protected void onRestart() {
    super.onRestart();
    Log.w("Cargo_Control_GPS", "onrestart");
}
/**
Запуск запису нового маршруту
*/
private void startAnewTrip() {
    current_trip = mCargo_Control_GPSDB.startAnewTrip();
    current_point = 0;
    total_distance = 0;
    last_loc = null;
    first_timestamp = 0;
    last_timestamp = 0;
}
// створення нової точки на мапі
/*
longitude, latitude, altitude, gps time, current trip
широта, довгота, висота, GPS час, поточний час поїздки
*/
private void newPointDB(double longitude, double latitude, double
                        altitude, long gps_time)
{
    mCargo_Control_GPSDB.newPointDB(longitude, latitude, altitude, gps_time,
    current_trip);
    current_point++;
}

private void stopTrip(){
    myLocationManager =
(LocationManager) getSystemService(Context.LOCATION_SERVICE);
    myLocationManager.removeUpdates(this);
    mCargo_Control_GPSDB.stopTrip(last_timestamp, first_timestamp,
                                total_distance, current_trip);

    current_trip = 0;
    current_point = 0;
}
private void fillList(){
    gpsstateTextView.setText(R.string.tips_not_running);
    Cursor lst = mCargo_Control_GPSDB.getListing();
    startManagingCursor(lst);
    Log.w("Cargo_Control_GPS", "4");
    listview_trace.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
    sc_adapter = new SimpleCursorAdapter(this,
        android.R.layout.simple_list_item_2, lst, new
        String[] {"cmt", "descr"},
        new int[] {android.R.id.text1, android.R.id.text2});
    listview_trace.setAdapter(sc_adapter);
}

private void setupForGPSAutoRefreshing() {
    final long MINIMUM_DISTANCECHANGE_FOR_UPDATE = 0;
// метри
    final long MINIMUM_TIME_BETWEEN_UPDATE = 1000;
// мілісекунди

```

```

        myLocationManager=(LocationManager)getSystemService(
            Context.LOCATION_SERVICE);
        Log.w("Cargo_Control_GPS","40 gps");

        myLocationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
            MINIMUM_TIME_BETWEEN_UPDATE, MINIMUM_DISTANCECHANGE_FOR_UPDATE, this);
        Log.w("Cargo_Control_GPS","50 gps");
    }

    public void onProviderDisabled(String provider) {
        Log.w("Cargo_Control_GPS","OnProviderDisabled " + provider);
        gpsstateTextView.setText(R.string.gps_inactive);
    }

    public void onProviderEnabled(String provider) {
        Log.w("Cargo_Control_GPS","OnProviderEnabled " + provider);
        gpsstateTextView.setText("OnProviderEnabled " + provider);
    }

    public void onStatusChanged(String provider, int status,
        Bundle extras)
    {
        Log.w("Cargo_Control_GPS","onStatusChanged "+provider+": "+status);
        gpsstateTextView.setText("onStatusChanged
            "+provider+": "+status);
    }

    private String formatMilliseconds(long s) {
        String ret = new String();
        if (s < 0) {
            ret += s+"ms";
        } else if (s < 60000) {
            ret += (s/1000)+"s";
        } else if (s < 1000 * 3600) {
            long min = s/(1000*60);
            long sec = (s- (min*60*1000))/ 1000;
            ret += min+"m,";
            ret += sec+"s";
        } else {
            long min = s/(1000*60);
            long hour = min/60;
            min -= hour*60;
            long sec = (s- (min*60*1000))/ 1000;
            ret += hour+"h,";
            ret += min+"m,";
            ret += sec+"s";
        }
        return ret;
    }

    @Override
    public void onLocationChanged(Location location) {
        Log.w("Cargo_Control_GPS","136 write");
        Date d = new Date(location.getTime());
        if (last_loc != null) {
            total_distance += location.distanceTo(last_loc);
        } else {
            first_timestamp = location.getTime();
        }
        last_timestamp = location.getTime();
        last_loc = new Location(location);

        /*
        Опис
        longitude, latitude, altitude, gps time, current trip
        широта, довгота, висота, GPS час, поточний час поїздки
        */
        gpsstateTextView.setText("Altitude : " + location.getAltitude()
            + "\nLongitude : " + location.getLongitude()
            + "\nLatitude : " +location.getLatitude()
            + "\nSpeed : "+ (location.hasSpeed() ? location.getSpeed() + "m/s" : "?"))
    }

```

```

+ "\nTime : "+ d.toLocaleString() + " GMT"
+ "\nTotal distance : " + total_distance
+ "\nAverage Speed: "+ ((last_timestamp - first_timestamp) > 0 ?
(total_distance*1000/(last_timestamp-first_timestamp))+ "m/s":""))
+ "\nDuration : " + formatMilliseconds(last_timestamp - first_timestamp));
newPointDB(location.getLongitude(),location.getLatitude(),location.getAltitude()
,location.getTime());
        Log.w("Cargo_Control_GPS","145 gps");
    }

    private void startAction() {
        if (!wlatp.isHeld()) {
            wlatp.acquire();
        }
        gpsstateTextView.setText(R.string.starting);
        startAnewTrip();
        setupForGPSAutoRefreshing();
    }

    private void stopAction()
    {
        stopTrip();
        fillList();
        if (wlatp.isHeld()) {
            wlatp.release();
        }
    }
    // експортувати маршрут
    private void exportTrip(long id,boolean email)
    {
        showDialog(DIALOG_EXPORT_PROGRESS);
        mExportThread = new ExportThread(this,id,mProgressDialog,email);
        mExportThread.start();
    }

    private static final int DIALOG_DELETE_VALID = 1;
    private static final int DIALOG_COPYRIGHT = 2;
    private static final int DIALOG_DELETE_ALL_VALID = 3;
    private static final int DIALOG_EXPORT_PROGRESS = 4;
    protected long delete_id;

    protected Dialog onCreateDialog(int id) {
        switch (id) {
            case DIALOG_DELETE_VALID:
                mDeleteDialog = new AlertDialog.Builder(Cargo_Control_GPS.this)
                    .setIcon(R.drawable.satellite)
                    .setTitle("Delete trip "+delete_id)
                    .setMessage(R.string.sure_delete)
                    .setPositiveButton(R.string.yes_delete, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                    mCargo_Control_GPSDB.deleteTrip(delete_id);
                    Toast.makeText(Cargo_Control_GPS.this, "Delete "+ delete_id,
                    Toast.LENGTH_LONG).show();
                    fillList();
                }
            })
                    .setNegativeButton(R.string.dont_delete, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                    Toast.makeText(Cargo_Control_GPS.this, "Видалити"+delete_id,
                    Toast.LENGTH_LONG).show();
                }
            })
                    .create();
                return mDeleteDialog;
            case DIALOG_EXPORT_PROGRESS:
                mProgressDialog = new ProgressDialog(Cargo_Control_GPS.this);

```

```

        mProgressDialog.setIcon(R.drawable.satellite);
        mProgressDialog.setTitle("Експортування...");
        mProgressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
        mProgressDialog.setMax(100);
        mProgressDialog.setButton(getText(android.R.string.cancel), new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {
                mExportThread.interrupt();
            }
        });
        mProgressDialog.setProgress(0);
        return mProgressDialog;
    case DIALOG_DELETE_ALL_VALID:
        return new AlertDialog.Builder(Cargo_Control_GPS.this)
            .setIcon(R.drawable.satellite)
            .setTitle("Delete all trip")
            .setMessage(R.string.sure_delete)
            .setPositiveButton(R.string.yes_delete, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                    mCargo_Control_GPSDB.deleteAllTrip();
                    Toast.makeText(Cargo_Control_GPS.this, "Delete all trips",
Toast.LENGTH_LONG).show();
                    fillList();
                }
            })
            .setNegativeButton(R.string.dont_delete, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                    Toast.makeText(Cargo_Control_GPS.this, "", Toast.LENGTH_LONG).show();
                }
            })
            .create();
    // авторське право
    case DIALOG_COPYRIGHT:
        return new AlertDialog.Builder(Cargo_Control_GPS.this)
            .setIcon(R.drawable.satellite)
            .setTitle("Авторські дані")
            .setMessage("Автор: Дядя Ірина Станіславівна\n Тип ліцензії: Free")
            .setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int whichButton) {
                }
            })
            .create();
    }
    return null;
}

private void deleteTrip(long id) {
    delete_id = id;
    if (mDeleteDialog != null) {
        mDeleteDialog.setTitle("Delete trip "+delete_id);
    }
    showDialog(DIALOG_DELETE_VALID);
    Log.w("Cargo_Control_GPS", "del 1 " + id);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    menu.add(Menu.NONE, MENU_COPY, 0,
R.string.menu_copy).setIcon(android.R.drawable.ic_menu_info_details);
    start_menu = menu.add(Menu.NONE, MENU_START, 0,
R.string.menu_start).setIcon(android.R.drawable.ic_media_play).setEnabled(true);
    stop_menu = menu.add(Menu.NONE, MENU_STOP, 0,
R.string.menu_stop).setIcon(android.R.drawable.ic_media_pause).setEnabled(false)
;
    delete_all_menu = menu.add(Menu.NONE, MENU_DELETE_ALL, 0,
R.string.menu_delete_all).setIcon(android.R.drawable.ic_menu_delete).setEnabled(
true);
}

```

```

export_all_menu = menu.add(Menu.NONE, MENU_EXPORT_ALL, 0,
R.string.menu_export_all).setIcon(android.R.drawable.ic_menu_save).setEnabled(true);
export_send_all_menu = menu.add(Menu.NONE, MENU_EXPORT_SEND_ALL, 0,
R.string.menu_export_send_all).setIcon(android.R.drawable.ic_menu_send).setEnabled(true);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case MENU_COPY:
            showDialog(DIALOG_COPYRIGHT);
            return true;
        case MENU_START: // Старт
            stop_menu.setEnabled(true);
            start_menu.setEnabled(false);
            delete_all_menu.setEnabled(false);
            export_all_menu.setEnabled(false);
            export_send_all_menu.setEnabled(false);
            listview_trace.setEnabled(false);
            gpsstateTextView.setLines(15);
            unregisterForContextMenu(listview_trace);
            startAction();
            return true;
        case MENU_STOP: // стоп
            stopAction();
            registerForContextMenu(listview_trace);
            start_menu.setEnabled(true);
            stop_menu.setEnabled(false);
            listview_trace.setEnabled(true);
            delete_all_menu.setEnabled(true);
            export_all_menu.setEnabled(true);
            export_send_all_menu.setEnabled(true);
            gpsstateTextView.setLines(1);
            return true;
        case MENU_DELETE_ALL: // очистити все
            showDialog(DIALOG_DELETE_ALL_VALID);
            Log.w("Cargo_Control_GPS","del all");
            return true;
        case MENU_EXPORT_ALL: // Експортувати все
            exportTrip(-1, false);
            return true;
        case MENU_EXPORT_SEND_ALL:
            exportTrip(-1, true);
            return true;
    }
    return false;
}
}
}

```

ФАЙЛ РЕАЛІЗАЦІЇ ОСНОВНОГО ФУНКЦІОНАЛУ

```

package nl.sogeti.android.gpstracker.viewer;
import nl.sogeti.android.gpstracker.R;
import nl.sogeti.android.gpstracker.actions.DescribeTrack;
import nl.sogeti.android.gpstracker.actions.Statistics;
import nl.sogeti.android.gpstracker.actions.tasks.GpxParser;
import nl.sogeti.android.gpstracker.actions.utils.ProgressListener;
import nl.sogeti.android.gpstracker.adapter.BreadcrumbsAdapter;
import nl.sogeti.android.gpstracker.adapter.SectionedListAdapter;
import nl.sogeti.android.gpstracker.breadcrumbs.BreadcrumbsService;
import nl.sogeti.android.gpstracker.breadcrumbs.BreadcrumbsService.LocalBinder;
import nl.sogeti.android.gpstracker.db.DatabaseHelper;
import nl.sogeti.android.gpstracker.db.GPSTracking;
import nl.sogeti.android.gpstracker.db.GPSTracking.Tracks;
import nl.sogeti.android.gpstracker.util.Constants;
import nl.sogeti.android.gpstracker.util.Pair;
import nl.sogeti.android.gpstracker.viewer.map.CommonLoggerMap;
import nl.sogeti.android.gpstracker.viewer.map.LoggerMap;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.app.Dialog;
import android.app.ListActivity;
import android.app.SearchManager;
import android.content.ActivityNotFoundException;
import android.content.BroadcastReceiver;
import android.content.ComponentName;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.ServiceConnection;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.os.IBinder;
import android.util.Log;
import android.view.ContextMenu;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.AdapterView;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.SimpleCursorAdapter;
import android.widget.TextView;
/**
Показати список всіх треків, також подвоюється для показу результатів пошуку
*/
public class TrackList extends ListActivity implements ProgressListener
{
    private static final String TAG = "OGT.TrackList";
    private static final int MENU_DETELE = Menu.FIRST + 0;
    private static final int MENU_SHARE = Menu.FIRST + 1;
    private static final int MENU_RENAME = Menu.FIRST + 2;
    private static final int MENU_STATS = Menu.FIRST + 3;
    private static final int MENU_SEARCH = Menu.FIRST + 4;
    private static final int MENU_VACUUM = Menu.FIRST + 5;
    private static final int MENU_PICKER = Menu.FIRST + 6;
    private static final int MENU_BREADCRUMBS = Menu.FIRST + 7;
    private static final int DIALOG_FILENAME = Menu.FIRST + 22;

```

```

private static final int DIALOG_RENAME = Menu.FIRST + 23;
private static final int DIALOG_DELETE = Menu.FIRST + 24;
private static final int DIALOG_VACUUM = Menu.FIRST + 25;
private static final int DIALOG_IMPORT = Menu.FIRST + 26;
private static final int DIALOG_INSTALL = Menu.FIRST + 27;
protected static final int DIALOG_ERROR = Menu.FIRST + 28;
private static final int PICKER_OI = Menu.FIRST + 29;
private static final int DESCRIBE = Menu.FIRST + 30;
private BreadcrumbsAdapter mBreadcrumbAdapter;
private EditText mTrackNameView;
private Uri mDialogTrackUri;
private String mDialogCurrentName = "";
private String mErrorDialogMessage;
private Exception mErrorDialogException;
private Runnable mImportAction;
private String mImportTrackName;
private String mErrorTask;
/**
 * реалізація роботи у фоновому режимі
 */
private ProgressListener mExportListener;
private int mPausePosition;
private BreadcrumbsService mService;
boolean mBound = false;
@Override
// створення
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    getWindow().requestFeature(Window.FEATURE_PROGRESS);
    this setContentView(R.layout.tracklist);
    displayIntent(getIntent());
    ListView listView = getListView();
    listView.setItemsCanFocus(true);
    // Додавання до контекстного меню
    registerForContextMenu(listView);
    if (savedInstanceState != null)
    {
        listView.setSelection(savedInstanceState.getInt("POSITION"));
    }
    IntentFilter filter = new IntentFilter();
    filter.addAction(BreadcrumbsService.NOTIFY_DATA_SET_CHANGED);
    registerReceiver(mReceiver, filter);
    Intent service = new Intent(this, BreadcrumbsService.class);
    startService(service);
}
@Override
// старт
protected void onStart()
{
    super.onStart();
    Intent intent = new Intent(this, BreadcrumbsService.class);
    bindService(intent, mConnection, Context.BIND_AUTO_CREATE);
}
@Override
// продовження
protected void onResume()
{
    if (mPausePosition != 0)
    {
        listView.setSelection(mPausePosition);
    }
    super.onResume();
}
@Override
// стоп
protected void onPause()
{
    mPausePosition = listView.getFirstVisiblePosition();
}

```

```

        super.onPause();
    }
    @Override
    protected void onStop()
    {
        if (mBound)
        {
            unbindService(mConnection);
            mBound = false;
            mService = null;
        }
        super.onStop();
    }
    @Override
    // знищення
    protected void onDestroy()
    {
        if (isFinishing())
        {
            Intent service = new Intent(this, BreadcrumbsService.class);
            stopService(service);
        }
        unregisterReceiver(mReceiver);
        super.onDestroy();
    }
    @Override
    // новий ідентифікатор
    public void onNewIntent(Intent newIntent)
    {
        displayIntent(newIntent);
    }
    @Override
    protected void onRestoreInstanceState(Bundle state)
    {
        super.onRestoreInstanceState(state);
        mDialogTrackUri = state.getParcelable("URI");
        mDialogCurrentName = state.getString("NAME");
        mDialogCurrentName = mDialogCurrentName != null ? mDialogCurrentName : "";
        getListView().setSelection(state.getInt("POSITION"));
    }
    @Override
    protected void onSaveInstanceState(Bundle outState)
    {
        super.onSaveInstanceState(outState);
        outState.putParcelable("URI", mDialogTrackUri);
        outState.putString("NAME", mDialogCurrentName);
        outState.putInt("POSITION", getListView().getFirstVisiblePosition());
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        boolean result = super.onCreateOptionsMenu(menu);
        menu.add(ContextMenu.NONE, MENU_SEARCH, ContextMenu.NONE,
        android.R.string.search_go).setIcon(android.R.drawable.ic_search_category_default).setAlphabeticShortcut(SearchManager.MENU_KEY);
        menu.add(ContextMenu.NONE, MENU_VACUUM, ContextMenu.NONE,
        R.string.menu_vacuum).setIcon(android.R.drawable.ic_menu_crop);
        menu.add(ContextMenu.NONE, MENU_PICKER, ContextMenu.NONE,
        R.string.menu_picker).setIcon(android.R.drawable.ic_menu_add);
        menu.add(ContextMenu.NONE, MENU_BREADCRUMBS, ContextMenu.NONE,
        R.string.dialog_breadcrumbsconnect).setIcon(android.R.drawable.ic_menu_revert);
        return result;
    }
    @Override
    // Вибрана опція меню
    public boolean onOptionsItemSelected(MenuItem item)
    {
        boolean handled = false;
        switch (item.getItemId())

```

```

{
    case MENU_SEARCH:
        onSearchRequested();
        handled = true;
        break;
    case MENU_VACUUM:
        showDialog(DIALOG_VACUUM);
        break;
    case MENU_PICKER:
        try
        {
Intent intent = new Intent("org.openintents.action.PICK_FILE");
intent.putExtra("org.openintents.extra.TITLE",
getString(R.string.dialog_import_picker));
intent.putExtra("org.openintents.extra.BUTTON_TEXT",
getString(R.string.menu_picker));
startActivityForResult(intent, PICKER_OI);
        }
        catch (ActivityNotFoundException e)
        {
            showDialog(DIALOG_INSTALL);
        }
        break;
    case MENU_BREADCRUMBS:
        mService.removeAuthentication();
        mService.clearAllCache();
        mService.collectBreadcrumbsOAuthToken();
        break;
    default:
        handled = super.onOptionsItemSelected(item);
}
return handled;
}
@Override
protected void onItemClick(ListView listView, View view, int position,
long id)
{
    super.onItemClick(listView, view, position, id);
    Object item = listView.getItemAtPosition(position);
    if (item instanceof String)
    {
        if (Constants.BREADCRUMBS_CONNECT.equals(item))
        {
            mService.collectBreadcrumbsOAuthToken();
        }
    }
    else if (item instanceof Pair< ?, ? >)
    {
        @SuppressWarnings("unchecked")
final Pair<Integer, Integer> track = (Pair<Integer, Integer>) item;
if (track.first == Constants.BREADCRUMBS_TRACK_ITEM_VIEW_TYPE)
    {
        TextView tv = (TextView) view.findViewById(R.id.listitem_name);
mImportTrackName = tv.getText().toString();
mImportAction = new Runnable()
        {
            @Override
            public void run()
            {
mService.startDownloadTask(TrackList.this, TrackList.this, track);
            }
        };
        showDialog(DIALOG_IMPORT);
    }
}
else
{
    Intent intent = new Intent();
    Uri trackUri = ContentUris.withAppendedId(Tracks.CONTENT_URI, id);

```

```

        intent.setData(trackUri);
        ComponentName caller = this.getCallingActivity();
        if (caller != null)
        {
            setResult(RESULT_OK, intent);
            finish();
        }
        else
        {
            intent.setClass(this, CommonLoggerMap.class);
            startActivity(intent);
        }
    }
}

@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo)
{
    if (menuInfo instanceof AdapterView.AdapterContextMenuInfo)
    {
        AdapterView.AdapterContextMenuInfo itemInfo =
        (AdapterView.AdapterContextMenuInfo) menuInfo;
        TextView textView = (TextView)
        itemInfo.targetView.findViewById(R.id.listitem_name);
        if (textView != null)
        {
            menu.setHeaderTitle(textView.getText());
        }
        Object listItem = getListAdapter().getItem(itemInfo.position);
        if (listItem instanceof Cursor)
        {
            menu.add(0, MENU_STATS, 0, R.string.menu_statistics);
            menu.add(0, MENU_SHARE, 0, R.string.menu_shareTrack);
            menu.add(0, MENU_RENAME, 0, R.string.menu_renameTrack);
            menu.add(0, MENU_DETELE, 0, R.string.menu_deleteTrack);
        }
    }
}

@Override
public boolean onContextItemSelected(Menu.Item item)
{
    boolean handled = false;
    AdapterView.AdapterContextMenuInfo info;
    try
    {
        info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    }
    catch (ClassCastException e)
    {
        Log.e(TAG, "Bad menuInfo", e);
        return handled;
    }
    Object listItem = getListAdapter().getItem(info.position);
    if (listItem instanceof Cursor)
    {
        Cursor cursor = (Cursor) listItem;
        mDialogTrackUri = ContentUris.withAppendedId(Tracks.CONTENT_URI,
        cursor.getLong(0));
        mDialogCurrentName = cursor.getString(1);
        mDialogCurrentName = mDialogCurrentName != null ? mDialogCurrentName :
        "";

        switch (item.getItemId())
        {
            case MENU_DETELE:
            {
                showDialog(DIALOG_DELETE);
                handled = true;
                break;
            }
        }
    }
}

```

```

        case MENU_SHARE:
        {
            Intent actionIntent = new Intent(Intent.ACTION_RUN);
            actionIntent.setDataAndType(mDialogTrackUri,
Tracks.CONTENT_ITEM_TYPE);
            actionIntent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
            startActivity(Intent.createChooser(actionIntent,
getString(R.string.share_track)));
            handled = true;
            break;
        }
        case MENU_RENAME:
        {
            showDialog(DIALOG_RENAME);
            handled = true;
            break;
        }
        case MENU_STATS:
        {
            Intent actionIntent = new Intent(this, Statistics.class);
            actionIntent.setData(mDialogTrackUri);
            startActivity(actionIntent);
            handled = true;
            break;
        }
        default:
            handled = super.onContextItemSelected(item);
            break;
    }
}
return handled;
}
@Override
// створення розділу діалогу
protected Dialog onCreateDialog(int id)
{
    Dialog dialog = null;
    Builder builder = null;
    switch (id)
    {
        case DIALOG_RENAME:
            LayoutInflater factory = LayoutInflater.from(this);
            View view = factory.inflate(R.layout.namedialog, null);
            mTrackNameView = (EditText) view.findViewById(R.id.nameField);
            builder = new
AlertDialog.Builder(this).setTitle(R.string.dialog_routename_title).setMessage(R
.string.dialog_routename_message).setIcon(android.R.drawable.ic_dialog_alert)
                .setPositiveButton(R.string.btn_okay,
mRenameOnClickListener).setNegativeButton(R.string.btn_cancel,
null).setView(view);
            dialog = builder.create();
            return dialog;
        case DIALOG_DELETE:
            builder = new
AlertDialog.Builder(TrackList.this).setTitle(R.string.dialog_delete_title).setIc
on(android.R.drawable.ic_dialog_alert).setNegativeButton(android.R.string.cancel
, null)
                .setPositiveButton(android.R.string.ok,
mDeleteOnClickListener);
            dialog = builder.create();
            String messageFormat =
this.getResources().getString(R.string.dialog_delete_message);
            String message = String.format(messageFormat, "");
            ((AlertDialog) dialog).setMessage(message);
            return dialog;
        case DIALOG_VACUUM:
            builder = new
AlertDialog.Builder(TrackList.this).setTitle(R.string.dialog_vacuum_title).setMe

```

```

message(R.string.dialog_vacuum_message).setIcon(android.R.drawable.ic_dialog_alert
)
        .setNegativeButton(android.R.string.cancel,
null).setPositiveButton(android.R.string.ok, mVacuumOnClickListener);
        dialog = builder.create();
        return dialog;
    case DIALOG_IMPORT:
        builder = new
AlertDialog.Builder(TrackList.this).setTitle(R.string.dialog_import_title).setMe
ssage(getString(R.string.dialog_import_message, mImportTrackName))

.setIcon(android.R.drawable.ic_dialog_alert).setNegativeButton(android.R.string.
cancel, null).setPositiveButton(android.R.string.ok, mImportOnClickListener);
        dialog = builder.create();
        return dialog;
    case DIALOG_INSTALL:
        builder = new AlertDialog.Builder(this);

builder.setTitle(R.string.dialog_noopicker).setMessage(R.string.dialog_noopick
er_message).setIcon(android.R.drawable.ic_dialog_alert)
        .setPositiveButton(R.string.btn_install,
mOiPickerDialogListener).setNegativeButton(R.string.btn_cancel, null);
        dialog = builder.create();
        return dialog;
    case DIALOG_ERROR:
        builder = new AlertDialog.Builder(this);

builder.setIcon(android.R.drawable.ic_dialog_alert).setTitle(android.R.string.di
alog_alert_title).setMessage(mErrorDialogMessage).setNeutralButton(android.R.str
ing.cancel, null);
        dialog = builder.create();
        return dialog;
    default:
        return super.onCreateDialog(id);
    }
}
@Override
protected void onPrepareDialog(int id, Dialog dialog)
{
    super.onPrepareDialog(id, dialog);
    AlertDialog alert;
    String message;
    switch (id)
    {
        case DIALOG_RENAME:
            mTrackNameView.setText(mDialogCurrentName);
            mTrackNameView.setSelection(0, mDialogCurrentName.length());
            break;
        case DIALOG_DELETE:
            alert = (AlertDialog) dialog;
            String messageFormat =
this.getResources().getString(R.string.dialog_delete_message);
            message = String.format(messageFormat, mDialogCurrentName);
            alert.setMessage(message);
            break;
        case DIALOG_ERROR:
            alert = (AlertDialog) dialog;
            message = "Failed task:\n" + mErrorTask;
            message += "\n\n";
            message += "Reason:\n" + mErrorDialogMessage;
            if (mErrorDialogException != null)
            {
                message += " (" + mErrorDialogException.getMessage() + ") ";
            }
            alert.setMessage(message);
            break;
        case DIALOG_IMPORT:
            alert = (AlertDialog) dialog;

```

```

        alert.setMessage(getString(R.string.dialog_import_message,
            mImportTrackName));
        break;
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode,
    Intent data)
{
    if (resultCode != RESULT_CANCELED)
    {
        switch (requestCode)
        {
            case PICKER_OI:
                new GpxParser(TrackList.this, TrackList.this).execute(data.getData());
                break;
            case DESCRIBE:
                Uri trackUri = data.getData();
                String name;
                if (data.getExtras() != null && data.getExtras().containsKey(Constants.NAME))
                {
                    name = data.getExtras().getString(Constants.NAME);
                }
                else
                {
                    name = "shareToGobreadcrumbs";
                }
                mService.startUploadTask(TrackList.this, mExportListener, trackUri, name);
                break;
            default:
                super.onActivityResult(requestCode, resultCode, data);
                break;
        }
    }
    else
    {
        if (requestCode == DESCRIBE)
        {
            mBreadcrumbAdapter.notifyDataSetChanged();
        }
    }
}
private void displayIntent(Intent intent)
{
    final String queryAction = intent.getAction();
    final String orderBy = Tracks.CREATION_TIME + " DESC";
    Cursor tracksCursor = null;
    if (Intent.ACTION_SEARCH.equals(queryAction))
    {
        // пошук запитів для треків
        tracksCursor = doSearchWithIntent(intent);
    }
    else if (Intent.ACTION_VIEW.equals(queryAction))
    {
        final Uri uri = intent.getData();
        if ("content".equals(uri.getScheme()) &&
            GPSTracking.AUTHORITY.equals(uri.getAuthority()))
        {
            // щоб подивитися в одному треку, необхідно задати його після LoggerMap
            Intent notificationIntent = new Intent(this, LoggerMap.class);
            notificationIntent.setData(uri);
            startActivity(notificationIntent);
            finish();
        }
        else if (uri.getScheme().equals("file") ||
            uri.getScheme().equals("content"))
        {
            mImportTrackName = uri.getLastPathSegment();
            // отримання ім'я файлу

```

```

        mImportAction = new Runnable()
        {
            @Override
            public void run()
            {
                new GpxParser(TrackList.this, TrackList.this).execute(uri);
            }
        };
        showDialog(DIALOG_IMPORT);
        tracksCursor = managedQuery(Tracks.CONTENT_URI, new String[] {
Tracks._ID, Tracks.NAME, Tracks.CREATION_TIME }, null, null, orderby);
    }
    else
    {
        Log.e(TAG, "У доступі відмовлено " + uri);
    }
}
else
{
    tracksCursor = managedQuery(Tracks.CONTENT_URI, new String[] { Tracks._ID,
Tracks.NAME, Tracks.CREATION_TIME }, null, null, orderby);
}
displayCursor(tracksCursor);
}
private void displayCursor(Cursor tracksCursor)
{
    SectionedListAdapter sectionedAdapter = new SectionedListAdapter(this);
    String[] fromColumns = new String[] { Tracks.NAME, Tracks.CREATION_TIME,
Tracks._ID };
    int[] toItems = new int[] { R.id.listitem_name, R.id.listitem_from,
R.id.bcSyncedCheckBox };
    SimpleCursorAdapter trackAdapter = new SimpleCursorAdapter(this,
R.layout.trackitem, tracksCursor, fromColumns, toItems);
    mBreadcrumbAdapter = new BreadcrumbsAdapter(this, mService);
    sectionedAdapter.addSection("Local", trackAdapter);
    // Розширення
    trackAdapter.setViewBinder(new SimpleCursorAdapter.ViewBinder()
    {
        public boolean setViewValue(View view, final Cursor cursor, int
columnIndex)
        {
            if (columnIndex == 0)
            {
                final long trackId = cursor.getLong(0);
                final String trackName = cursor.getString(1);
                // Вивести повідомлення якщо ми в онлайні
                final CheckBox checkbox = (CheckBox) view;
                final ProgressBar progressbar = (ProgressBar) ((View)
view.getParent()).findViewById(R.id.bcExportProgress);
                if (mService != null && mService.isAuthorized())
                {
                    checkbox.setVisibility(View.VISIBLE);
                    boolean isOnline = mService.isLocalTrackSynced(trackId);
                    checkbox.setEnabled(!isOnline);
                    boolean isSynced = mService.isLocalTrackSynced(trackId);
                    checkbox.setOnCheckedChangeListener(null);
                    checkbox.setChecked(isSynced);
                    checkbox.setOnCheckedChangeListener(new
OnCheckedChangeListener()
                    {
                        @Override
                        public void onCheckedChanged(CompoundButton
buttonView, boolean isChecked)
                        {
                            if (isChecked)
                            {
                                Intent namingIntent = new Intent(TrackList.this, DescribeTrack.class);
                                namingIntent.setData(ContentUris.withAppendedId(Tracks.CONTENT_URI, trackId));

```

```

namingIntent.putExtra(Constants.NAME, trackName);
mExportListener = new ProgressListener()
{
@Override
public void setIndeterminate(boolean indeterminate)
{

progressbar.setIndeterminate(indeterminate);
}
@Override
public void started()
{

checkbox.setVisibility(View.INVISIBLE);

progressbar.setVisibility(View.VISIBLE);
}
@Override
public void finished(Uri result)
{

checkbox.setVisibility(View.VISIBLE);

progressbar.setVisibility(View.INVISIBLE);
progressbar.setIndeterminate(false);
}
@Override
public void setProgress(int value)
{
progressbar.setProgress(value);
}

@Override
public void showError(String task, String errorMessage, Exception exception)
{
TrackList.this.showError(task, errorMessage, exception);
}
};
startActivityForResult(namingIntent, DESCRIBE);
}
});
}
else
{
checkbox.setVisibility(View.INVISIBLE);
checkbox.setOnCheckedChangeListener(null);
}
return true;
}
return false;
}
});
setListAdapter(sectionedAdapter);
}
private Cursor doSearchWithIntent(final Intent queryIntent)
{
final String queryString = queryIntent.getStringExtra(SearchManager.QUERY);
Cursor cursor = managedQuery(Tracks.CONTENT_URI, new String[] { Tracks._ID,
Tracks.NAME, Tracks.CREATION_TIME }, "name LIKE ?", new String[] { "%" +
queryString + "%" }, null);
return cursor;
}
/*
Інтерфейс ProgressListener і користувальницького інтерфейсу
*/
@Override
public void setIndeterminate(boolean indeterminate)
{
setProgressBarIndeterminate(indeterminate);
}

```

```

    }
    @Override
    public void started()
    {
        setProgressBarVisibility(true);
        setProgress(Window.PROGRESS_START);
    }
    @Override
    public void finished(Uri result)
    {
        setProgressBarVisibility(false);
        setProgressBarIndeterminate(false);
    }
    @Override
    public void showError(String task, String errorDialogMessage, Exception
errorDialogException)
    {
        mErrorTask = task;
        mErrorDialogMessage = errorDialogMessage;
        mErrorDialogException = errorDialogException;
        Log.e(TAG, errorDialogMessage, errorDialogException);
        if (!isFinishing())
        {
            showDialog(DIALOG_ERROR);
        }
        setProgressBarVisibility(false);
        setProgressBarIndeterminate(false);
    }
    private ServiceConnection mConnection = new ServiceConnection()
    {
        @Override
public void onServiceConnected(ComponentName className, IBinder service)
        {
            LocalBinder binder = (LocalBinder) service;
            mService = binder.getService();
            mBound = true;
            mBreadcrumbAdapter.setService(mService);
        }
        @Override
public void onServiceDisconnected(ComponentName arg0)
        {
            mBound = false;
            mService = null;
        }
    };
    private OnClickListener mDeleteOnClickListener = new
DialogInterface.OnClickListener()
    {
        @Override
public void onClick(DialogInterface dialog, int which)
        {
            getContentResolver().delete(mDialogTrackUri, null, null);
        }
    };
    private OnClickListener mRenameOnClickListener = new
DialogInterface.OnClickListener()
    {
        @Override
public void onClick(DialogInterface dialog, int which)
        {
            Log.d( TAG, "Context item selected: "+mDialogUri+" with name
"+mDialogCurrentName );
            String trackName = mTrackNameView.getText().toString();
            ContentValues values = new ContentValues();
            values.put(Tracks.NAME, trackName);
            TrackList.this.getContentResolver().update(mDialogTrackUri, values,
null, null);
        }
    };
};

```

```
private OnClickListener mVacuumOnClickListener = new
DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        DatabaseHelper helper = new DatabaseHelper(TrackList.this);
        helper.vacuum();
    }
};

private OnClickListener mImportOnClickListener = new
DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        mImportAction.run();
    }
};

private final DialogInterface.OnClickListener mOiPickerDialogListener = new
DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        try
        {
            startActivity(oiAboutIntent);
        }
    }
};

private BroadcastReceiver mReceiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        if
(BreadcrumbsService.NOTIFY_DATA_SET_CHANGED.equals(intent.getAction()))
        {
            mBreadcrumbAdapter.updateItemList();
        }
    }
};
}
```

ΦOPMA ABTOPA

```

package nl.sogeti.android.gpstracker.viewer;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import nl.sogeti.android.gpstracker.R;
import android.app.Activity;
import android.content.Context;
import android.content.pm.PackageManager.NameNotFoundException;
import android.os.Bundle;
import android.util.Log;
import android.webkit.WebView;
import android.widget.TextView;

public class About extends Activity
{
    private static final String TAG = "OGT.About";
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.about);
        fillContentFields();
    }
    private void fillContentFields()
    {
        TextView version = (TextView) findViewById(R.id.version);
        try
        {
            version.setText(getPackageManager().getPackageInfo(getPackageName(),
0).versionName);
        }
        catch (NameNotFoundException e)
        {
            version.setText("");
        }
        WebView license = (WebView) findViewById(R.id.license_body);
        license.loadUrl("file:///android_asset/license_short.html");
        WebView contributions = (WebView) findViewById(R.id.contribution_body);
        contributions.loadUrl("file:///android_asset/contributions.html");
        WebView notice = (WebView) findViewById(R.id.notices_body);
        notice.loadUrl("file:///android_asset/notices.html");
    }
    public static String readRawTextFile(Context ctx, int resId)
    {
        InputStream inputStream = ctx.getResources().openRawResource(resId);
        InputStreamReader inputreader = new InputStreamReader(inputStream);
        BufferedReader buffreader = new BufferedReader(inputreader);
        String line;
        StringBuilder text = new StringBuilder();
        try
        {
            while ((line = buffreader.readLine()) != null)
            {
                text.append(line);
                text.append('\n');
            }
        }
        return text.toString();    }}

```

МОНИТОРИНГ GPS

```

package nl.sogeti.android.gpstracker.viewer;
import nl.sogeti.android.gpstracker.R;
import nl.sogeti.android.gpstracker.actions.DescribeTrack;
import nl.sogeti.android.gpstracker.actions.Statistics;
import nl.sogeti.android.gpstracker.actions.tasks.GpxParser;
import nl.sogeti.android.gpstracker.actions.utils.ProgressListener;
import nl.sogeti.android.gpstracker.adapter.BreadcrumbsAdapter;
import nl.sogeti.android.gpstracker.adapter.SectionedListAdapter;
import nl.sogeti.android.gpstracker.breadcrumbs.BreadcrumbsService;
import nl.sogeti.android.gpstracker.breadcrumbs.BreadcrumbsService.LocalBinder;
import nl.sogeti.android.gpstracker.db.DatabaseHelper;
import nl.sogeti.android.gpstracker.db.GPSTracking;
import nl.sogeti.android.gpstracker.db.GPSTracking.Tracks;
import nl.sogeti.android.gpstracker.util.Constants;
import nl.sogeti.android.gpstracker.util.Pair;
import nl.sogeti.android.gpstracker.viewer.map.CommonLoggerMap;
import nl.sogeti.android.gpstracker.viewer.map.LoggerMap;
import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.app.Dialog;
import android.app.ListActivity;
import android.app.SearchManager;
import android.content.ActivityNotFoundException;
import android.content.BroadcastReceiver;
import android.content.ComponentName;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.ServiceConnection;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.os.IBinder;
import android.util.Log;
import android.view.ContextMenu;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.AdapterView;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.SimpleCursorAdapter;
import android.widget.TextView;
public class TrackList extends ListActivity implements ProgressListener
{
    private static final String TAG = "OGT.TrackList";
    private static final int MENU_DETELE = Menu.FIRST + 0;
    private static final int MENU_SHARE = Menu.FIRST + 1;
    private static final int MENU_RENAME = Menu.FIRST + 2;
    private static final int MENU_STATS = Menu.FIRST + 3;
    private static final int MENU_SEARCH = Menu.FIRST + 4;
    private static final int MENU_VACUUM = Menu.FIRST + 5;
    private static final int MENU_PICKER = Menu.FIRST + 6;
    private static final int MENU_BREADCRUMBS = Menu.FIRST + 7;
    public static final int DIALOG_FILENAME = Menu.FIRST + 22;
    private static final int DIALOG_RENAME = Menu.FIRST + 23;
    private static final int DIALOG_DELETE = Menu.FIRST + 24;
    private static final int DIALOG_VACUUM = Menu.FIRST + 25;

```

```

private static final int DIALOG_IMPORT = Menu.FIRST + 26;
private static final int DIALOG_INSTALL = Menu.FIRST + 27;
protected static final int DIALOG_ERROR = Menu.FIRST + 28;
private static final int PICKER_OI = Menu.FIRST + 29;
private static final int DESCRIBE = Menu.FIRST + 30;
private BreadcrumbsAdapter mBreadcrumbAdapter;
private EditText mTrackNameView;
private Uri mDialogTrackUri;
private String mDialogCurrentName = "";
private String mErrorDialogMessage;
private Exception mErrorDialogException;
private Runnable mImportAction;
private String mImportTrackName;
private String mErrorTask;
private ProgressListener mExportListener;
private int mPausePosition;
private BreadcrumbsService mService;
boolean mBound = false;
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    getWindow().requestFeature(Window.FEATURE_PROGRESS);
    this setContentView(R.layout.tracklist);
    displayIntent(getIntent());
    ListView listView = getListView();
    listView.setItemsCanFocus(true);
// Додовання до контекстного меню
    registerForContextMenu(listView);
    if (savedInstanceState != null)
    {
        listView.setSelection(savedInstanceState.getInt("POSITION"));
    }
    IntentFilter filter = new IntentFilter();
    filter.addAction(BreadcrumbsService.NOTIFY_DATA_SET_CHANGED);
    registerReceiver(mReceiver, filter);
    Intent service = new Intent(this, BreadcrumbsService.class);
    startService(service);
}
@Override
protected void onStart()
{
    super.onStart();
    Intent intent = new Intent(this, BreadcrumbsService.class);
    bindService(intent, mConnection, Context.BIND_AUTO_CREATE);
}
@Override
protected void onResume()
{
    if (mPausePosition != 0)
    {
        listView.setSelection(mPausePosition);
    }
    super.onResume();
}
@Override
protected void onPause()
{
    mPausePosition = listView.getFirstVisiblePosition();
    super.onPause();
}
@Override
protected void onStop()
{
    if (mBound)
    {
        unbindService(mConnection);
        mBound = false;
        mService = null;
    }
}

```

```

    }
    super.onStop();
}
@Override
protected void onDestroy()
{
    if (isFinishing())
    {
        Intent service = new Intent(this, BreadcrumbsService.class);
        stopService(service);
    }
    unregisterReceiver(mReceiver);
    super.onDestroy();
}
@Override
public void onNewIntent(Intent newIntent)
{
    displayIntent(newIntent);
}
@Override
protected void onRestoreInstanceState(Bundle state)
{
    super.onRestoreInstanceState(state);
    mDialogTrackUri = state.getParcelable("URI");
    mDialogCurrentName = state.getString("NAME");
    mDialogCurrentName = mDialogCurrentName != null ? mDialogCurrentName : "";
    getListView().setSelection(state.getInt("POSITION"));
}
@Override
protected void onSaveInstanceState(Bundle outState)
{
    super.onSaveInstanceState(outState);
    outState.putParcelable("URI", mDialogTrackUri);
    outState.putString("NAME", mDialogCurrentName);
    outState.putInt("POSITION", getListView().getFirstVisiblePosition());
}
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    boolean result = super.onCreateOptionsMenu(menu);
    menu.add(ContextMenu.NONE, MENU_SEARCH, ContextMenu.NONE,
android.R.string.search_go).setIcon(android.R.drawable.ic_search_category_defaul
t).setAlphabeticShortcut(SearchManager.MENU_KEY);
    menu.add(ContextMenu.NONE, MENU_VACUUM, ContextMenu.NONE,
R.string.menu_vacuum).setIcon(android.R.drawable.ic_menu_crop);
    menu.add(ContextMenu.NONE, MENU_PICKER, ContextMenu.NONE,
R.string.menu_picker).setIcon(android.R.drawable.ic_menu_add);
    menu.add(ContextMenu.NONE, MENU_BREADCRUMBS, ContextMenu.NONE,
R.string.dialog_breadcrumbsconnect).setIcon(android.R.drawable.ic_menu_revert);
    return result;
}
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    boolean handled = false;
    switch (item.getItemId())
    {
        case MENU_SEARCH:
            onSearchRequested();
            handled = true;
            break;
        case MENU_VACUUM:
            showDialog(DIALOG_VACUUM);
            break;
        case MENU_PICKER:
            try
            {
                Intent intent = new Intent("org.openintents.action.PICK_FILE");

```

```

        intent.putExtra("org.openintents.extra.TITLE",
getString(R.string.dialog_import_picker));
        intent.putExtra("org.openintents.extra.BUTTON_TEXT",
getString(R.string.menu_picker));
        startActivityForResult(intent, PICKER_OI);
    }
    catch (ActivityNotFoundException e)
    {
        showDialog(DIALOG_INSTALL);
    }
    break;
case MENU_BREADCRUMBS:
    mService.removeAuthentication();
    mService.clearAllCache();
    mService.collectBreadcrumbsOAuthToken();
    break;
default:
    handled = super.onOptionsItemSelected(item);
}
return handled;
}
@Override
protected void onItemClick(ListView listView, View view, int position,
long id)
{
    super.onItemClick(listView, view, position, id);
    Object item = listView.getItemAtPosition(position);
    if (item instanceof String)
    {
        if (Constants.BREADCRUMBS_CONNECT.equals(item))
        {
            mService.collectBreadcrumbsOAuthToken();
        }
    }
    else if (item instanceof Pair<?, ?>)
    {
        @SuppressWarnings("unchecked")
        final Pair<Integer, Integer> track = (Pair<Integer, Integer>) item;
        if (track.first == Constants.BREADCRUMBS_TRACK_ITEM_VIEW_TYPE)
        {
            TextView tv = (TextView) view.findViewById(R.id.listitem_name);
            mImportTrackName = tv.getText().toString();
            mImportAction = new Runnable()
            {
                @Override
                public void run()
                {
                    mService.startDownloadTask(TrackList.this, TrackList.this, track);
                }
            };
            showDialog(DIALOG_IMPORT);
        }
    }
    else
    {
        Intent intent = new Intent();
        Uri trackUri = ContentUris.withAppendedId(Tracks.CONTENT_URI, id);
        intent.setData(trackUri);
        ComponentName caller = this.getCallingActivity();
        if (caller != null)
        {
            setResult(RESULT_OK, intent);
            finish();
        }
        else
        {
            intent.setClass(this, CommonLoggerMap.class);
            startActivity(intent);
        }
    }
}

```

```

    }
}
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo)
{
    if (menuInfo instanceof AdapterView.AdapterContextMenuInfo)
    {
        AdapterView.AdapterContextMenuInfo itemInfo =
(AdapterView.AdapterContextMenuInfo) menuInfo;
        TextView textView = (TextView)
itemInfo.targetView.findViewById(R.id.listitem_name);
        if (textView != null)
        {
            menu.setHeaderTitle(textView.getText());
        }
        Object listItem = getListAdapter().getItem(itemInfo.position);
        if (listItem instanceof Cursor)
        {
            menu.add(0, MENU_STATS, 0, R.string.menu_statistics);
            menu.add(0, MENU_SHARE, 0, R.string.menu_shareTrack);
            menu.add(0, MENU_RENAME, 0, R.string.menu_renameTrack);
            menu.add(0, MENU_DETELE, 0, R.string.menu_deleteTrack);
        }
    }
}
@Override
public boolean onContextItemSelected(Menu.Item item)
{
    boolean handled = false;
    AdapterView.AdapterContextMenuInfo info;
    try
    {
        info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    }
    catch (ClassCastException e)
    {
        Log.e(TAG, "Bad menuInfo", e);
        return handled;
    }
    Object listItem = getListAdapter().getItem(info.position);
    if (listItem instanceof Cursor)
    {
        Cursor cursor = (Cursor) listItem;
        mDialogTrackUri = ContentUris.withAppendedId(Tracks.CONTENT_URI,
cursor.getLong(0));
        mDialogCurrentName = cursor.getString(1);
        mDialogCurrentName = mDialogCurrentName != null ? mDialogCurrentName : "";
        switch (item.getItemId())
        {
            case MENU_DETELE:
            {
                showDialog(DIALOG_DELETE);
                handled = true;
                break;
            }
            case MENU_SHARE:
            {
                Intent actionIntent = new Intent(Intent.ACTION_RUN);
                actionIntent.setDataAndType(mDialogTrackUri, Tracks.CONTENT_ITEM_TYPE);
                actionIntent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
                startActivity(Intent.createChooser(actionIntent,
getString(R.string.share_track)));
                handled = true;
                break;
            }
            case MENU_RENAME:
            {
                showDialog(DIALOG_RENAME);

```

```

        handled = true;
        break;
    }
    case MENU_STATS:
    {
        Intent actionIntent = new Intent(this, Statistics.class);
        actionIntent.setData(mDialogTrackUri);
        startActivity(actionIntent);
        handled = true;
        break;
    }
    default:
        handled = super.onContextItemSelected(item);
        break;
    }
}
return handled;
}
@Override
protected Dialog onCreateDialog(int id)
{
    Dialog dialog = null;
    Builder builder = null;
    switch (id)
    {
    case DIALOG_RENAME:
        LayoutInflater factory = LayoutInflater.from(this);
        View view = factory.inflate(R.layout.namedialog, null);
        mTrackNameView = (EditText) view.findViewById(R.id.nameField);
        builder = new
        AlertDialog.Builder(this).setTitle(R.string.dialog_routename_title).setMessage(R
        .string.dialog_routename_message).setIcon(android.R.drawable.ic_dialog_alert).se
        tPositiveButton(R.string.btn_okay,
        mRenameOnClickListener).setNegativeButton(R.string.btn_cancel,
        null).setView(view);
        dialog = builder.create();
        return dialog;
    case DIALOG_DELETE:
        builder = new
        AlertDialog.Builder(TrackList.this).setTitle(R.string.dialog_delete_title).setIc
        on(android.R.drawable.ic_dialog_alert).setNegativeButton(android.R.string.cancel
        , null).setPositiveButton(android.R.string.ok, mDeleteOnClickListener);
        dialog = builder.create();
        String messageFormat =
        this.getResources().getString(R.string.dialog_delete_message);
        String message = String.format(messageFormat, "");
        ((AlertDialog) dialog).setMessage(message);
        return dialog;
    case DIALOG_VACUUM:
        builder = new
        AlertDialog.Builder(TrackList.this).setTitle(R.string.dialog_vacuum_title).setMe
        ssage(R.string.dialog_vacuum_message).setIcon(android.R.drawable.ic_dialog_alert
        ).setNegativeButton(android.R.string.cancel,
        null).setPositiveButton(android.R.string.ok, mVacuumOnClickListener);
        dialog = builder.create();
        return dialog;
    case DIALOG_IMPORT:
        builder = new
        AlertDialog.Builder(TrackList.this).setTitle(R.string.dialog_import_title).setMe
        ssage(getString(R.string.dialog_import_message, mImportTrackName))

        .setIcon(android.R.drawable.ic_dialog_alert).setNegativeButton(android.R.string.
        cancel, null).setPositiveButton(android.R.string.ok, mImportOnClickListener);
        dialog = builder.create();
        return dialog;
    case DIALOG_INSTALL:
        builder = new AlertDialog.Builder(this);

```

```

builder.setTitle(R.string.dialog_noopicker).setMessage(R.string.dialog_noopicker_message).setIcon(android.R.drawable.ic_dialog_alert)
    .setPositiveButton(R.string.btn_install,
mOipickerDialogListener).setNegativeButton(R.string.btn_cancel, null);
    dialog = builder.create();
    return dialog;
case DIALOG_ERROR:
    builder = new AlertDialog.Builder(this);

builder.setIcon(android.R.drawable.ic_dialog_alert).setTitle(android.R.string.dialog_alert_title).setMessage(mErrorMessage).setNeutralButton(android.R.string.cancel, null);
    dialog = builder.create();
    return dialog;
default:
    return super.onCreateDialog(id);
}
}
@Override
protected void onPrepareDialog(int id, Dialog dialog)
{
    super.onPrepareDialog(id, dialog);
    AlertDialog alert;
    String message;
    switch (id)
    {
        case DIALOG_RENAME:
            mTrackNameView.setText(mDialogCurrentName);
            mTrackNameView.setSelection(0, mDialogCurrentName.length());
            break;
        case DIALOG_DELETE:
            alert = (AlertDialog) dialog;
            String messageFormat =
this.getResources().getString(R.string.dialog_delete_message);
            message = String.format(messageFormat, mDialogCurrentName);
            alert.setMessage(message);
            break;
        case DIALOG_ERROR:
            alert = (AlertDialog) dialog;
            message = "Failed task:\n" + mErrorTask;
            message += "\n\n";
            message += "Reason:\n" + mErrorMessage;
            if (mErrorDialogException != null)
            {
                message += " (" + mErrorDialogException.getMessage() + ") ";
            }
            alert.setMessage(message);
            break;
        case DIALOG_IMPORT:
            alert = (AlertDialog) dialog;
            alert.setMessage(getString(R.string.dialog_import_message,
mImportTrackName));
            break;
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if (resultCode != RESULT_CANCELED)
    {
        switch (requestCode)
        {
            case PICKER_OI:
                new GpxParser(TrackList.this,
TrackList.this).execute(data.getData());
                break;
            case DESCRIBE:
                Uri trackUri = data.getData();

```

```

        String name;
        if (data.getExtras() != null &&
data.getExtras().containsKey(Constants.NAME))
        {
            name = data.getExtras().getString(Constants.NAME);
        }
        else
        {
            name = "shareToGobreadcrumbs";
        }
        mService.startUploadTask(TrackList.this, mExportListener,
trackUri, name);
        break;
        default:
            super.onActivityResult(requestCode, resultCode, data);
            break;
    }
}
else
{
    if (requestCode == DESCRIBE)
    {
        mBreadcrumbAdapter.notifyDataSetChanged();
    }
}
}
private void displayIntent(Intent intent)
{
    final String queryAction = intent.getAction();
    final String orderBy = Tracks.CREATION_TIME + " DESC";
    Cursor tracksCursor = null;
    if (Intent.ACTION_SEARCH.equals(queryAction))
    {
        tracksCursor = doSearchWithIntent(intent);
    }
    else if (Intent.ACTION_VIEW.equals(queryAction))
    {
        final Uri uri = intent.getData();
        if ("content".equals(uri.getScheme()) &&
GPSTracking.AUTHORITY.equals(uri.getAuthority()))
        {
            // Got to VIEW a single track, instead hand it of to the LoggerMap
            Intent notificationIntent = new Intent(this, LoggerMap.class);
            notificationIntent.setData(uri);
            startActivity(notificationIntent);
            finish();
        }
        else if (uri.getScheme().equals("file") ||
uri.getScheme().equals("content"))
        {
            mImportTrackName = uri.getLastPathSegment();
            mImportAction = new Runnable()
            {
                @Override
                public void run()
                {
                    new GpxParser(TrackList.this, TrackList.this).execute(uri);
                }
            };
            showDialog(DIALOG_IMPORT);
            tracksCursor = managedQuery(Tracks.CONTENT_URI, new String[] {
Tracks._ID, Tracks.NAME, Tracks.CREATION_TIME }, null, null, orderBy);
        }
        else
        {
            Log.e(TAG, "Unable to VIEW " + uri);
        }
    }
}
else

```

```

    {
// формування списку
tracksCursor = managedQuery(Tracks.CONTENT_URI, new String[] { Tracks._ID,
Tracks.NAME, Tracks.CREATION_TIME }, null, null, orderby);
    }
    displayCursor(tracksCursor);
}
private void displayCursor(Cursor tracksCursor)
{
SectionedListAdapter sectionedAdapter = new SectionedListAdapter(this);
String[] fromColumns = new String[]
{ Tracks.NAME, Tracks.CREATION_TIME, Tracks._ID };
int[] toItems = new int[] { R.id.listitem_name, R.id.listitem_from,
R.id.bcSyncedCheckBox };
SimpleCursorAdapter trackAdapter = new SimpleCursorAdapter(this,
R.layout.trackitem, tracksCursor, fromColumns, toItems);
mBreadcrumbAdapter = new BreadcrumbsAdapter(this, mService);
sectionedAdapter.addSection("Local", trackAdapter);
sectionedAdapter.addSection("www.gobreadcrumbs.com", mBreadcrumbAdapter);
trackAdapter.setViewBinder(new SimpleCursorAdapter.ViewBinder()
{
public boolean setViewValue(View view, final Cursor cursor, int
columnIndex)
{
    if (columnIndex == 0)
    {
        final long trackId = cursor.getLong(0);
        final String trackName = cursor.getString(1);
        final CheckBox checkbox = (CheckBox) view;
        final ProgressBar progressbar = (ProgressBar) ((View)
view.getParent()).findViewById(R.id.bcExportProgress);
        if (mService != null && mService.isAuthorized())
        {
            checkbox.setVisibility(View.VISIBLE);
boolean isOnline = mService.isLocalTrackSynced(trackId);
checkbox.setEnabled(!isOnline);
boolean isSynced = mService.isLocalTrackSynced(trackId);
checkbox.setOnCheckedChangeListener(null);
checkbox.setChecked(isSynced);
checkbox.setOnCheckedChangeListener(new
OnCheckedChangeListener()
{
@Override
public void onCheckedChanged(CompoundButton
buttonView, boolean isChecked)
{
            if (isChecked)
            {
// Почати запис траси
Intent namingIntent = new Intent(TrackList.this, DescribeTrack.class);
namingIntent.setData(ContentUris.withAppendedId(Tracks.CONTENT_URI, trackId));
namingIntent.putExtra(Constants.NAME, trackName);
mExportListener = new ProgressListener()
{
@Override
public void setIndeterminate(boolean indeterminate)
{
progressbar.setIndeterminate(indeterminate);
}
@Override
public void started()
{
checkbox.setVisibility(View.INVISIBLE);
progressbar.setVisibility(View.VISIBLE);
}
@Override
public void finished(Uri result)

```



```

    }
    setProgressBarVisibility(false);
    setProgressBarIndeterminate(false);
}
private ServiceConnection mConnection = new ServiceConnection()
{
@Override
public void onServiceConnected(ComponentName className, IBinder service)
{
    LocalBinder binder = (LocalBinder) service;
    mService = binder.getService();
    mBound = true;
    mBreadcrumbAdapter.setService(mService);
}
@Override
public void onServiceDisconnected(ComponentName arg0)
{
    mBound = false;
    mService = null;
}
};

private OnClickListener mDeleteOnClickListener = new
DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which)
{
    getContentResolver().delete(mDialogTrackUri, null, null);
}
};

private OnClickListener mRenameOnClickListener = new
DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which)
{
Log.d( TAG, "Context item selected: "+mDialogUri+" with name
"+mDialogCurrentName );
String trackName = mTrackNameView.getText().toString();
ContentValues values = new ContentValues();
values.put(Tracks.NAME, trackName);
TrackList.this.getContentResolver().update(mDialogTrackUri, values,
null, null);
}
};

private OnClickListener mVacuumOnClickListener = new
DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which)
{
    DatabaseHelper helper = new DatabaseHelper(TrackList.this);
    helper.vacuum();
}
};

private OnClickListener mImportOnClickListener = new
DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which)
{
    mImportAction.run();
}
};

private final DialogInterface.OnClickListener mOiPickerDialogListener = new
DialogInterface.OnClickListener()
{
@Override
public void onClick(DialogInterface dialog, int which)

```

```
        {
Intent oiAboutIntent = new Intent(Intent.ACTION_VIEW, oiDownload);
        try
        {
            startActivity(oiAboutIntent);
        }
        catch (ActivityNotFoundException e)
        {
            oiAboutIntent = new Intent(Intent.ACTION_VIEW, oiDownload);
            startActivity(oiAboutIntent);
        }
    }
};
private BroadcastReceiver mReceiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
If (BreadcrumbsService.NOTIFY_DATA_SET_CHANGED.equals(intent.getAction()))
        {
            mBreadcrumbAdapter.updateItemList();
        }
    }
};
}
```

K6П3_2023