

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем і автоматики
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління
(повна назва кафедри)

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

«Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Ч. 1. Мобільна підсистема і підсистема управління базою даних»

Виконав: студент 2-го курсу, групи
2АКІТ-20м
спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

Кирилюк Артем

(ім'я та прізвище)

Керівник: к.т.н., доцент каф. КСУ

Марія Юхимчук

(ім'я та прізвище)

« ____ » _____ 2021 р.

Опонент: к.т.н., доцент каф. АІТ

Юрій Іванов

(ім'я та прізвище)

« ____ » _____ 2021 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.

Володимир Дубовой

(ім'я та прізвище)

« ____ » _____ 2021 р.

Вінниця ВНТУ – 2021 рік

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра комп'ютерних систем управління
Рівень вищої освіти II-й (магістерський)
Галузь знань – 15 Автоматизація та приладобудування
Спеціальність – 151 Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

д.т.н., проф.

Володимир Дубовой

«01» 10 2021 року

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Кирилюку Артему Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи магістерської кваліфікаційної роботи: Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Ч. 1. Мобільна підсистема і підсистема управління базою даних

керівник магістерської кваліфікаційної роботи доцент кафедри КСУ
Юхимчук Марія Сергіївна, к.т.н., доцент

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

затверджені наказом вищого навчального закладу від “24” 09 2021 року
№ 277

2. Строк подання студентом магістерської кваліфікаційної роботи
10.12.2021 року

3. Вихідні дані до магістерської кваліфікаційної роботи експлуатаційні дані з об'єкту дослідження, клієнт-серверна архітектура.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)огляд предметної області; розробка математичного апарату; розробка програмного забезпечення та експериментальні дослідження.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) архітектура системи; робочий процес системи; приклад вихідних даних; інтерфейс користувача розробленої системи; лістинг програмного забезпечення.

6. Консультанти розділів роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	професор кафедри КСУ, професор, д.т.н. Ковтун В.В.		
Економічний розділ	доцент кафедри ЕПВМ, доцент, к.е.н. Кавецький В.В.		

7. Дата видачі завдання “_01_”_10_2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Огляд предметної області	04.09.2021р.	
2	Розробка математичного апарату	22.09.2021р.	
3	Розробка програмного забезпечення та експериментальні дослідження	3.11.2021р.	
4	Оформлення пояснювальної записки і графічного матеріалу	08.12.2021р.	
5	Попередній захист роботи	16.12. 2021 р.	
6	Остаточний захист роботи	22.12. 2021 р.	

Студент

(підпис)

Артем Кирилюк

Керівник роботи

(підпис)

Марія Юхимчук

АНОТАЦІЯ

Магістерська робота присвячена вирішенню однієї з найактуальніших логістичних проблем – проблеми пошуку паркувальних місць. У першій частині розглядаються існуючі рішення та висвітлюються переваги та недоліки кожного рішення. Основним недоліком існуючої системи пошуку кінцевих точок логістичного процесу, тобто системи пошуку кінцевих точок логістичного процесу, є відсутність більш детальної інформації про саме місце розташування, тобто поточний вільний простір чи ні. Користувач буде знати стан стоянки та може спланувати найкращий маршрут. Це також дозволить компаніям контролювати свої місця для паркування.

Створена покращена система пошуку кінцевих точок логістичного процесу, системні функції та режими використання. Розглядається технологія, яка буде використана для створення системи знаходження кінця логістичного процесу. Система створена як веб-сторінка, написана за допомогою фреймворку Angular. Для відображення карти використовується API Карт Google. Системна база даних базується на базі даних Google Firebase реального часу. Система також використовує Firebase Authentication, службу аутентифікації від Google. Показано, як налаштувати правила читання та запису в базі даних і як увійти в систему.

ABSTRACT

The master's thesis is devoted to solving one of the most pressing logistics problems - the problem of finding parking spaces. The first part discusses the existing solutions and highlights the advantages and disadvantages of each solution. The main disadvantage of the existing logistics process endpoint search system, ie the logistics process endpoint search system, is the lack of more detailed information about the location, ie the current free space or not. The user will know the status of the parking lot and can plan the best route. It will also allow companies to control their parking spaces. An improved system for finding the end points of the logistics process, system functions and modes of use.

The technology that will be used to create a system for finding the end of the logistics process is considered. The system is designed as a web page written using the Angular framework. The Google Maps API is used to display the map. The system database is based on the Google Firebase real-time database. The system also uses Firebase Authentication, Google's authentication service. Shows how to configure read and write rules in the database and how to log in..

ЗМІСТ

ВСТУП.....	4
1 ІСНУЮЧІ СИСТЕМИ ПОШУКУ КІНЦЕВИХ ТОЧОК ЛОГІСТИЧНОГО ПРОЦЕСУ.....	7
1.1 Особливості систем пошуку кінцевих точок логістичного процесу	7
1.2 Система пошуку кінцевих точок логістичного процесу “UNIP – паркуйся без проблем”	8
1.3 Система пошуку кінцевих точок логістичного процесу “Parking UA”	11
1.4 Системи пошуку кінцевих точок логістичного процесу “Parkopedia Парковки”	14
1.5 Порівняння проаналізованих систем пошуку кінцевих точок логістичного процесу	18
2 ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ СИСТЕМИ ПОШУКУ КІНЦЕВИХ ТОЧОК ЛОГІСТИЧНОГО ПРОЦЕСУ	20
2.1 Аналіз технологій для розробки системи пошуку кінцевих точок логістичного процесу	20
2.2 Налаштування бази даних та автентифікація.....	27
2.3 Огляд давачів руху для облаштування парковок	32
3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ПОШУКУ КІНЦЕВИХ ТОЧОК ЛОГІСТИЧНОГО ПРОЦЕСУ.....	36
3.1 Вимоги та опис системи кінцевих точок логістичного процесу	36
3.2 Структура системи пошуку кінцевих точок логістичного процесу.....	38
3.3 Розробка інтерфейсу системи кінцевих точок логістичного процесу	39
3.3.1 Корпоративний режим	40
3.3.2 Користувацький режим	46
3.4 Управління авторизованими користувачами та базою даних	48
3.5 Тестування системи пошуку кінцевих точок логістичного процесу	51
4 ЕКОНОМІЧНА ЧАСТИНА	54
4.2 Визначення рівня конкурентоспроможності розробки	58

	3
4.3 Розрахунок витрат на проведення науково-дослідної роботи	62
4.3.1 Витрати на оплату праці	62
4.3.2 Відрахування на соціальні заходи	65
4.3.3 Сировина та матеріали	66
4.3.4 Розрахунок витрат на комплектуючі	67
4.3.5 Спецустаткування для наукових (експериментальних) робіт	67
4.3.6 Програмне забезпечення для наукових (експериментальних) робіт	68
4.3.7 Амортизація обладнання, програмних засобів та приміщень	69
4.3.8 Паливо та енергія для науково-виробничих цілей	70
4.3.9 Службові відрядження	71
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації	72
4.3.11 Інші витрати	72
4.3.12 Накладні (загальновиробничі) витрати	73
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	74
Висновки до розділу	80
ВИСНОВКИ	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	82
ДОДАТКИ	87
Додаток А	88
Додаток Б	91
Додаток В	108

ВСТУП

Кількість автомобілів у великих містах з кожним роком збільшується, і зростає потреба у пошуку місць, де можна припаркувати автомобілі. Повсякденні люди їздять у відрядження, і кожному потрібно знайти відповідну парковку, де можна на деякий час припаркувати автомобіль. Тепер у кожного є потужний смартфон, а у водіїв почали з'являтися системи пошуку кінця логістичного процесу. Ці системи допомагають водіям знайти місця для паркування у великих містах. Водій може переглянути всі місця для паркування, відзначені на карті, куди йому потрібно поїхати, забронювати вдале місце та прокласти маршрут, щоб водієві не потрібно було турбуватися про те, де припаркувати автомобіль. Сьогодні кожен може завантажити мобільний додаток або знайти веб-сайт, який допоможе знайти місце для паркування автомобіля. Щоб скористатися кінцевим пошуком логістичного процесу, необхідно зареєструватися, ввести інформацію про себе, номер машини. Деякі системи пошуку кінцевих логістичних точок можуть сплачувати плату за паркування за допомогою смартфонів. Сплачайте, прикріпивши кредитну картку, і оплачайте через впроваджений платіжний сервіс, або оплачайте через мобільний рахунок користувача. На даний момент це дуже зручно для користувачів. Немає необхідності шукати паркомат, не потрібно чекати паркувальника, не потрібно платити до кабінки, сплатити плату за паркування можна не виходячи з дому. Крім того, для кожної розробленої системи кінцевих точок логістичного процесу створюється дуже багата позначка паркування, яка може містити багато інформації про паркування, детальні характеристики розташування, місця для інвалідів та різноманітні послуги, які надає паркінг. Провідні компанії та представники бізнесу можуть зацікавитися системою пошуку кінцевих точок логістики, яка значно розширює спектр мережевих послуг користувачів. Використовуючи ці

системи, ви зможете збирати статистичні дані, аналізувати їх і покращувати послуги, що надаються користувачам, виявляти проблеми та прогнозувати обсяг трафіку, щоб збалансувати навантаження на паркування для забезпечення ефективного використання.

Функцією нової системи пошуку кінцевих точок логістики є онлайн-моніторинг паркування та співпраця з підприємствами чи власниками паркінгів, що дозволить користувачам заздалегідь знати його статус, вільний чи зайнятий, чи знаходиться він на технічному обслуговуванні чи технічній роботі. Власники автомобілів можуть налаштувати власні місця для паркування, встановити інший статус або тарифи, і користувачі отримають найновішу інформацію про паркування. Тому тема магістерської роботи — розробка вдосконаленої системи знаходження кінця логістичного процесу — надзвичайно актуальна. Нещодавно запропоновані вдосконалення необхідно впровадити якомога швидше, щоб полегшити кінцевим користувачам пошук місць для паркування та полегшити керування ними власниками паркінгів, а також для подальшого розвитку системи пошуку кінцевих точок логістичного процесу.

Метою магістерської роботи є розробка системи пошуку паркувальних місць. Розвивайте здатність переглядати стан паркування та інформацію про паркування в Інтернеті, а також розвивайте здатність підприємців створювати та контролювати власні паркінги.

Об'єктом дослідження є процес пошуку місць для паркування. Предмет дослідження – методи та засоби пошуку місць для паркування.

Отримані результати можуть бути використані для подальших досліджень системи пошуку паркування та розробки нових ідей, які можуть бути реалізовані на основі системи пошуку паркування.

Практична цінність. За допомогою Angular framework був розроблений програмний додаток для пошуку паркувальних місць з режимом користувача

та режимом підприємства з можливістю відстеження стану парковки через телефон. Оглянута детальна інформація по використанню системою у кожному режимі. Розглянуто налаштування та контроль авторизації користувачів.

Інноваційністю роботи є розробка додатку з використанням систем Angular Framework, Google Maps API та бази даних Firebase Realtime Database. Додаток дозволяє проводити пошук паркувальних місць та переглядати стан самих парковок. Окрім цього доступний корпоративний режим який дозволяє створити свою парковку, змінювати про неї інформацію.

Апробація. Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2022)»:

- Артем Іванович Кирилюк, Владислав Олександрович Бізер «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства», ВНПІК САМН «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2021. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2022/schedConf/presentations>.

1 ІСНУЮЧІ СИСТЕМИ ПОШУКУ КІНЦЕВИХ ТОЧОК ЛОГІСТИЧНОГО ПРОЦЕСУ

1.1 Особливості систем пошуку кінцевих точок логістичного процесу

Різні системи пошуку кінцевих точок, які використовуються в процесі створення логістики, щоб допомогти користувачам цих систем легко знайти місця для паркування своїх автомобілів. Користувачеві потрібно лише завантажити додаток на свій смартфон, а потім виконати досить просту реєстрацію в додатку. Після реєстрації інформація користувача буде повністю безпечною. Система пошуку кінцевої точки логістики використовується разом з картою, тому користувачі зможуть знайти безкоштовне місце для паркування в додатку на карті та перейти до цього місця зі свого поточного місцезнаходження. Користувач може вибрати потрібне місце на карті, або ввести адресу в спеціальне поле пошуку, або скористатися різними фільтрами, щоб вибрати відповідне місце паркування, щоб на карті можна було знайти всі найближчі паркувальні місця. Після вибору на карті відобразяться всі вибрані користувачем місця для паркування в межах певного діапазону. Система пошуку кінцевих точок логістики містить детальну інформацію про більшість паркінгів і паркінгів. Ви можете переглянути інформацію, таку як тарифи, години роботи, кількість кінцевих точок логістичного процесу, розміри, якість обслуговування тощо. Більшість систем пошуку кінцевих логістичних точок можуть сплачувати плату за паркування за допомогою смартфонів, а користувачі можуть підключати кредитні картки для оплати безготівкової плати за паркування.

Поки що існує кілька реалізацій систем пошуку кінцевих точок логістики для Android та iOS. Кожен може завантажити мобільні додатки на свій смартфон. Далі я розгляну кілька мобільних додатків, а саме

«UNIP-Parking без проблем» [1], «Parking UA» [2], «Parkopedia parking» [3].
Усі ці мобільні додатки можна завантажити в Google Play.

1.2 Система пошуку кінцевих точок логістичного процесу “UNIP – паркуйся без проблем”

UNIP [1] — це мобільний додаток, який дозволяє знаходити, вибирати маршрути та оплачувати паркувальне місце поблизу обраного вами пункту призначення. За допомогою цієї програми ви можете отримати більше інформації про всі стоянки міста. Отримайте більше інформації про тарифи, кількість місць, розміри та інші послуги. Додаток має пошукові фільтри. Оплатити можна по телефону.

Після завантаження програми та її запуску користувачеві необхідно зареєструватися. Спочатку потрібно виконати стандартну реєстрацію за номером телефону та короткого коду в текстовому повідомленні, а потім зареєструвати номер автомобіля. Можна ввести нестандартний реєстраційний номер автомобіля. Після успішної реєстрації користувач буде перенаправлено на карту, на якій буде показано різні місця, де можна припаркуватися. Маркер на карті виглядає як коло з цифрою в середині, яка є кількістю паркувальних місць або місць для паркування в цій точці. Якщо підтягнути карту досить близько, маркери на карті зміняться з кола на прямокутник, і це буде вказувати ціну паркування за одну годину, або це може бути безкоштовний напис, що вказує, що ця стоянка або парковка безкоштовна. Є два типи стоянок: на відкритих територіях — це звичайна прямокутна позначка на карті чи закрита територія, а також — прямокутна позначка з кришкою. Користувач може написати зручну для нього адресу у відповідному полі внизу екрана, і він побачить список паркувальних місць або стоянку у відповідному місці, введеному користувачем. Або ви можете натиснути на

будь-який маркер на карті, це місце буде виділено, відкриється інформація про цю стоянку чи стоянку, користувач може переглянути детальну інформацію про це місце, встановити нагадування про певний час і маршрут до цього місця з вашого місцезнаходження чи парку. Якщо користувач уже на місці.

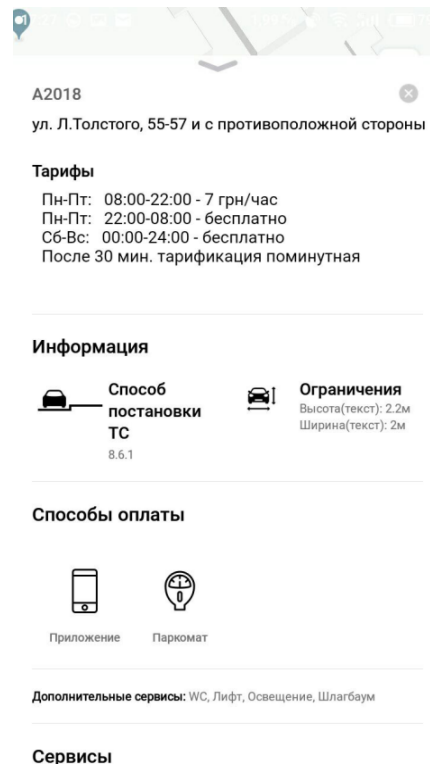


Рисунок 1.1. Деталі вибраного паркінгу або паркувального місця на карті в мобільному додатку UNIP [1]

Користувач може переглянути детальну інформацію про обрану стоянку або стоянку (Рисунок 1.1.). Відкривши детальну інформацію, користувач побачить адресу стоянки, інформацію про тариф, різну інформацію про паркувальне місце, наприклад, обмеження по ширині та висоті, спосіб паркування транспортного засобу тощо, способи оплати паркування, паркування чи різні послуги. надається користувачеві паркуванням, а користувач також може додати цю парковку до вибраного.

Користувач може вибрати зручний час для паркування свого автомобіля. Вам потрібно натиснути дзвіночок внизу екрана на вибраному паркінгу або стоянці, вибрати відповідну дату і час, і програмі потрібно буде надіслати користувачеві повідомлення або нагадування. Ви також можете прикріпити маршрут до вибраної стоянки. Користувачу необхідно натиснути кнопку «Навігація», яка має більше інформації про відстань від місця розташування користувача до стоянки (Рисунок 1.2.). Додаток згенерує маршрут і надасть можливість відкрити або завантажити навігаційну програму для маршруту та навігації користувача до вибраної стоянки.

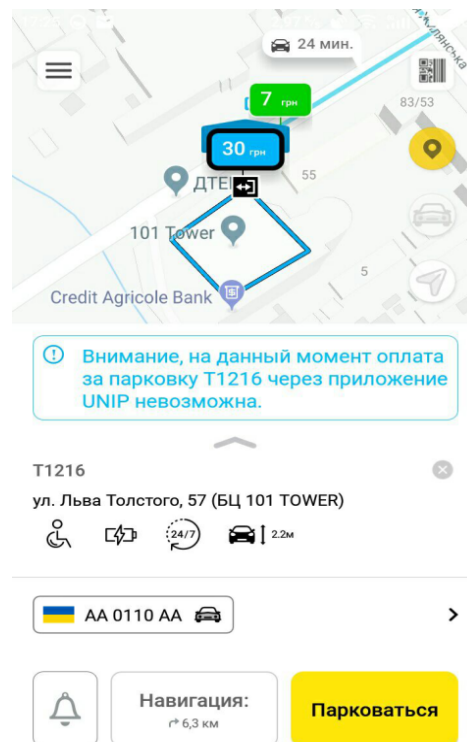


Рисунок 1.2. Можливості паркування в мобільному додатку UNIP [1]

Якщо користувач вже перебуває на потрібному парковці, він може використовувати додаток для оплати паркування. Йому потрібно натиснути кнопку «припаркувати» внизу екрана. Користувачам потрібно вибрати свій транспортний засіб зі списку і можуть зареєструвати кілька транспортних

засобів. Якщо плата за паркування сплачена, користувач повинен вибрати кількість годин або точно вибрати конкретну дату і час, додаток розрахує суму платежу на основі поточного тарифу, а потім натисне кнопку «Почати». Якщо на рахунку недостатньо коштів, програма надає додатковий рахунок. Після початку паркування сума паркування буде списана з рахунку користувача і розпочнеться час до закінчення паркування. Якщо паркування безкоштовне, запуститься лічильник часу паркування, користувач може легко підняти автомобіль і натиснути «Зупинити» внизу програми, і користувач побачить повідомлення про закінчення паркування та час паркування. Користувач може відкрити меню програми. За допомогою меню користувачі можуть поповнити свій рахунок, змінити персональні дані, записані в додатку, додати інший автомобіль, відредагувати або видалити будь-який автомобіль, переглянути вибрану стоянку чи стоянку, придбати абонементи, переглянути повідомлення, переглянути історію паркування або конфігурацію додаток.

1.3 Система пошуку кінцевих точок логістичного процесу “Parking UA”

Parking UA [2] — мобільний додаток, створений на основі цифрового гаманця Masterpass. Програму необхідно зареєструвати, перш ніж функції програми стануть доступними. Додаток визначить місто та відобразить найкращі варіанти на карті. Для оплати паркування необхідно прикріпити кредитну картку. За допомогою функції «Розумні гроші» ви також можете використовувати мобільний рахунок «Київстар» для оплати паркування. Як і в попередньому додатку, користувачам необхідно зареєструватися після завантаження. Необхідно виконати стандартну реєстрацію за номером телефону та короткого коду в повідомленні. Після успішної реєстрації користувач буде перенаправлено на карту, на якій буде показано різні місця,

де можна припаркуватися. Маркер на карті виглядає як коло з цифрою в середині, яка є кількістю паркувальних місць або місць для паркування в цій точці.

Користувач може натиснути на будь-який маркер на карті, місце розташування буде виділено, і відкриється інформація про паркування (Рисунок 1.3.). У додатку буде показана коротка інформація, а саме час, коли можна залишити транспортний засіб, адреса стоянки та погодинна ціна. Користувач може натиснути кнопку «ЗНАЙТИ ПАРКИ ПОРУЧ» і перейти до списку найближчих вулиць. Ви можете вибрати будь-яку стоянку зі списку або ввести потрібну вулицю. Ви також можете вибрати фільтр як фільтр для всіх паркінгів, або працювати цілодобово. Натисніть на потрібну вулицю, і програма відкриє інформацію про паркінг (Рисунок 1.4). Він міститиме ту саму інформацію, що й на рисунку 1.3. Відобразатиметься лише відстань від місця розташування користувача до стоянки. Натисніть кнопку «ПАРКОВАТЬ», і програма відкриє вікно, як показано на рисунку 1.3.

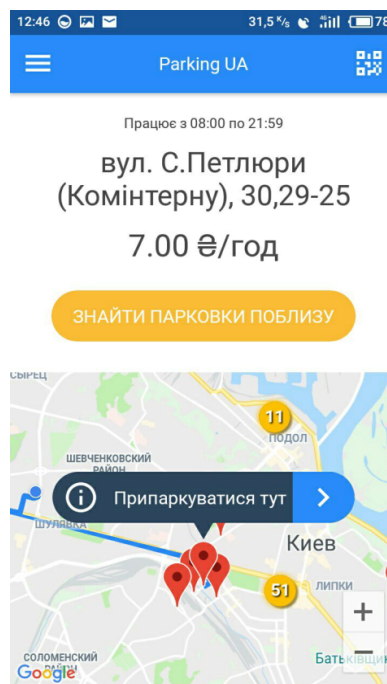


Рисунок 1.3. Інформація про вибране паркувальне місце або паркувальне місце на карті в мобільному додатку parking UA [2]

Щоб припаркувати свій автомобіль, користувачеві необхідно натиснути позначку на карті, щоб перейти до вікна, показаного на рисунку 1.3, а потім потрібно натиснути кнопку «Припаркувати тут», програма відкриє вікно, де ви зможете вибрати автомобіль, кредитна картка та години паркування. Засоби (Рисунок 1.4). Якщо користувач не додав автомобіль, як у попередньому додатку, необхідно ввести реєстраційний номер автомобіля та вибрати марку. Якщо користувач не додав платіжну картку, вам потрібно буде додати свою картку. Користувач також може натиснути «припаркувати» під час перегляду інформації про паркування. У цьому випадку програма також відкриє вікно, в якому потрібно вибрати автомобіль і кредитну картку, як і в попередньому випадку.

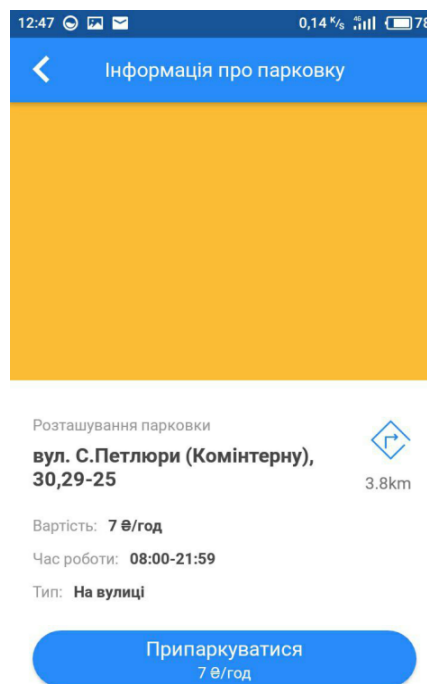


Рисунок 1.4. Інформація про вибране паркувальне місце або паркувальне місце на карті в мобільному додатку «Parking UA»[2]

Користувач може відкрити меню програми. За допомогою меню користувачі можуть переглядати, додавати, видаляти та редагувати інформацію про свої автомобілі та кредитні картки. Ви можете переглянути історію їх стоянки, де і коли припаркувався автомобіль. Перейдіть до налаштувань персональних даних і налаштувань програми. Зв'яжіться зі службою підтримки програми або вийдіть зі свого облікового запису.

1.4 Системи пошуку кінцевих точок логістичного процесу “Parkopedia Парковки”

Parkopedia [3] — мобільний додаток з дуже великою базою даних парковок. Надайте інформацію про паркування в 75 країнах. У мобільному додатку є пошук по вулицях, інформація про кількість вільних місць, різні фільтри пошуку. Цей мобільний додаток більше схожий на довідник з паркування, він дозволяє знайти місце, прокласти до нього маршрут тощо. Кожен користувач може додати місця для паркування в будь-який час. Після запуску програми відкриється початкове вікно. Користувачам не потрібно реєструватися для використання програми. У ньому будуть практично всі функції програми. Додаток буде використовуватися як загальний посібник, ви зможете переглянути будь-яке паркувальне місце, інформацію про нього, прокласти маршрут, додати у вибране. Реєстрація в програмі дозволяє переглядати історію паркування та резервувати місце. Допоможіть програмі знайти стоянки, яких немає на карті. Для пошуку місць для паркування користувачеві необхідно ввести назву міста, місце розташування або індекс у вікні пошуку, і користувач побачить список варіантів, що підходять для запиту, і просто клацніть на потрібну програму, щоб відкрити карту з паркуванням. позначки. Користувач також може натиснути на «справжнє місцезнаходження», і програма відстежить місцезнаходження та відкриє

карту з позначками паркування, де знаходиться користувач. Користувач може переглянути вибрану стоянку, натиснувши «Вибране». Він побачить список вибраних паркінгів, користувач може видалити будь-яку стоянку з вибраного або натиснути будь-яку, а додаток відкриє карту з вибраним паркінгом. Натиснувши «Останні результати», програма відкриє карту з останнім пошуком паркування та відобразить статус на момент запиту.

Користувач може натиснути три кнопки в нижній частині вікна. Перший – мій обліковий запис. При натисканні цієї кнопки відкриється вікно з інформацією про профіль користувача. Це якщо він зареєстрований, якщо ні, він може зареєструватися в додатку, а потім переглянути інформацію свого профілю. Другий — «Мoje місце». При натисканні цієї кнопки відкриється вікно з історією паркування. Остання кнопка — «Таблиця лідерів». Натиснувши на нього, ви зможете переглянути верхній список помічників програми. Це людина, яка активно допомагає розробці програми. Коли користувач знаходиться на карті з різними маркерами, він може натиснути на них, щоб переглянути коротку інформацію про паркінг (рис. 1.5.). Користувачеві буде відображатися адреса паркування, погодинна оплата, кількість паркувальних місць і відстань від місцезнаходження користувача до вибраної стоянки. Користувач може прокручувати вліво або вправо, щоб вибрати інше місце для паркування поблизу. У нижній частині екрана користувачі можуть натиснути три кнопки.

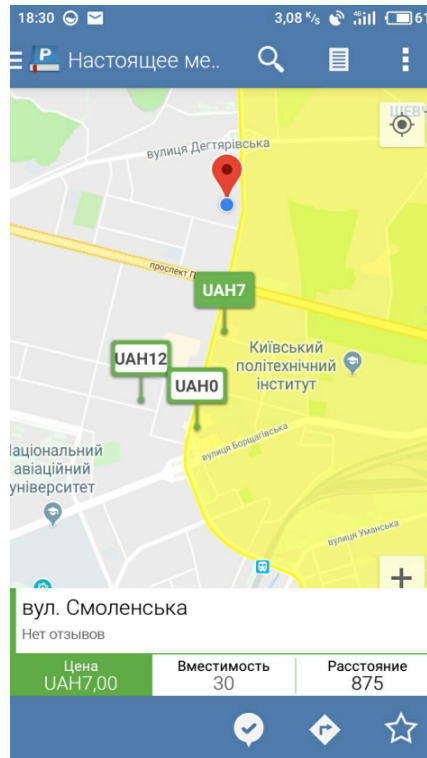


Рисунок 1.5. Коротка інформація про паркування в мобільному додатку «Parkopedia Parking»[3]

Коли користувач перебуває на цій стоянці, він може помітити це, натиснувши першу кнопку нижче, щоб ви могли вибрати свій автомобіль. Щоб прокласти маршрут до цієї стоянки, вам потрібно натиснути другу кнопку нижче, а потім додаток запропонує відкрити додаток для навігації. Щоб додати це місце для паркування до вибраного, користувач повинен натиснути останню кнопку, після чого воно буде додано до списку вибраних паркувальних місць. Користувач може натиснути на повідомлення про паркування, щоб переглянути детальну інформацію (Рисунок 1.6.). Ви також можете прокрутити ліворуч або праворуч, щоб побачити інші стоянки поблизу.

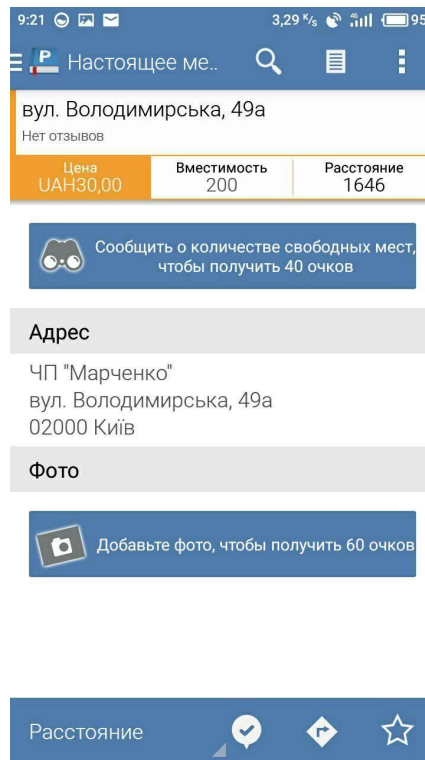


Рисунок 1.6. Детальна інформація про паркування в мобільному додатку «Parkopedia Parking»[3]

У верхній частині екрана користувачі можуть торкнутися значка збільшувального скла, щоб відкрити пошук і знайти потрібне місце для паркування, або торкнутися значка літери, щоб відкрити список паркування, або торкнутися значка програми, щоб відкрити меню програми. У меню, користувачі можуть переглянути свою особисту інформацію, торкнутися Ім'я користувача, перевірити, де його автомобіль припарковано, і натиснути «Моя бронювання». У меню також є фільтр, ви можете знайти найкращий вибір для паркування через різні фільтри. Ви можете вибрати тип паркування, наприклад: загальна парковка, приватна парковка, вулична парковка. Ви можете вибрати тип або варіант на стоянці: фільтр висоти, безкоштовний, можна оплатити картою, є спеціальні місця для інвалідів, можна заряджати електромобілі тощо.

1.5 Порівняння проаналізованих систем пошуку кінцевих точок логістичного процесу

Переглянувши дані кінцевих точок пошукової системи логістичного процесу, можна провести порівняння, виділити всі переваги та недоліки кожної системи та зробити висновки про те, які зручні та корисні, а які непотрібні та небажані для користувачів. Необхідні (Таблиця 1.1). Ви також можете порівняти кількість завантажень та оглядів цих програм. Всі ці програми можна завантажити з Game Market. Тут ви можете знайти інформацію про кількість завантажень і переглянути коментарі. Мобільний додаток «UNIP»[1] має понад 10 000 завантажень, «Parking UA»[2] також має понад 10 000, а «Parkopedia Parking»[3] — понад мільйон завантажень. Мобільний додаток «UNIP»[1] має 158 відгуків з оцінкою 2,4, мобільний додаток «Parking UA»[2] має 353 відгуки з оцінкою 3,5, мобільний додаток «Parkopedia Parking»[3] Номер – 9321, а оцінка – 4,6.

Таблиця 1.1 - Порівняння систем пошуку кінцевих точок логістичного процесу

Функціонал	“UNIP”	“Parking UA”	“Parkopedia Парковки”
Обов’язкова реєстрація	+	+	-
Реєстрування автомобіля	+	+	-
Інформативні маркери на мапі	+	-	+
Зручна мапа для користувача	+	-	+
Пошук парковок	+	+	+

Продовження таблиці 1.1

Навігація до парковки	+	+	+
Детальна інформація про парковку	+	-	+
Фільтри для пошуку	-	-	+
Оплата кредитною карткою через додаток	+	+	-
Додавати парковки до обраних	-	+	+

Кінцеві точки логістичного процесу забезпечують подібний набір функцій. Основне завдання, яке вони виконують, – знайти відповідну парковку для користувачів. Деякі програми були реалізовані добре, а деякі потребують покращення. Мобільний додаток «Parkopedia Parking» більше схожий на довідник, а не на пошукову систему, інші додатки краще реалізовані, і в них є можливість оплачувати паркування через додаток. Зараз це дуже зручно для користувачів. Якщо порівнювати ці додатки за дизайном та зручністю використання, то мобільний додаток «UNIP-Безпроблемна парковка» є найкращим, але в ньому відсутні зручні фільтри для пошуку паркувальних місць, наприклад, додаток «Parkopedia Parking». У кожному з цих додатків добре реалізовані певні функції, але кожній програмі чогось не вистачає. У додатку «UNIP-Parking без проблем» відсутні фільтри в пошуку, у додатку «Parking UA» відсутні фільтри та інформаційні маркери на карті, додаток «Parkopedia Parking» оплачує через додаток та більше інформації про паркування. Тому, дивлячись на огляд існуючих рішень, можна сказати, що існуючій системі пошуку кінця логістичного процесу бракує більш детальної інформації про саму локацію, тобто про те, чи є приміщення в даний момент. Це дозволить користувачам заздалегідь знати стан паркувальних місць і планувати маршрути.

2 ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ СИСТЕМИ ПОШУКУ КІНЦЕВИХ ТОЧОК ЛОГІСТИЧНОГО ПРОЦЕСУ

2.1 Аналіз технологій для розробки системи пошуку кінцевих точок логістичного процесу

Система реалізована у вигляді веб-сторінки, яка буде написана з використанням фреймворку Angular [4]. Фреймворк Angular — це веб-система для написання односторінкових додатків. Він спрямований на вирішення багатьох проблем при розробці односторінкових додатків, на спрощення розробки та тестування таких додатків, а також на створення основи архітектури модель-представлення-контролер (MVC) (Рисунок 2.1). На стороні клієнта також є компоненти, які зазвичай використовуються в багатьох Інтернет-додатках.



Рисунок 2.1. Модель MVC

Фреймворк Angular [4] працює, спочатку читаючи сторінку мови гіпертекстової розмітки (HTML), яка містить додаткові спеціальні атрибути HTML. Angular інтерпретує ці атрибути як інструкції, прив'язуючи вхідну або вихідну частину сторінки до моделі, представленої стандартними змінними JavaScript. Значення цих змінних JavaScript можна встановити вручну в кодї, або їх можна отримати зі статичних або динамічних ресурсів JSON. Angular реалізує рішення MVC для розділення компонентів презентації, даних і логіки. Використовуючи залежний вхід, Angular надає традиційні серверні послуги для клієнтських веб-додатків, таких як контролери, що залежать від перегляду. Таким чином, більшість навантаження на сервер можна зменшити. Виберіть Google Maps API [5], щоб використовувати карту в системі. Google Maps — це безкоштовний картографічний сервіс, який надається Google, а також набір додатків на основі цього сервісу та інших технологій Google. Сервіс являє собою карту світу та супутникові зображення, які надають користувачам панорамні види вулиць, онлайн-аналіз трафіку та можливості маршрутизації (автомобілем, пішки, велосипедом або громадським транспортом). Використання Google Maps API [5] дуже зручно і має багато функцій. Будуть використані навігація, відстеження, маркери використання та інформаційні вікна.

Щоб використовувати базу даних, я вирішив вибрати базу даних реального часу Firebase від Google [6]. База даних Firebase Realtime [6] дозволяє створювати синдикувані програми, які забезпечують безпечний доступ до бази даних безпосередньо з клієнтського коду. Дані зберігаються локально, і навіть автономні події в режимі реального часу продовжують працювати, даючи кінцевому користувачеві чітку відповідь. Коли пристрій повторно підключається, база даних реального часу синхронізує зміни локальних даних з даними сервера оновлення, які відбулися під час

відключення клієнта, і автоматично вирішує будь-які конфлікти. Онлайн-бази даних забезпечують гнучку мову на основі виразів, яка називається правилами безпеки в реальному часі Firebase, щоб визначити структуру ваших даних і те, коли ви можете читати або записувати дані.

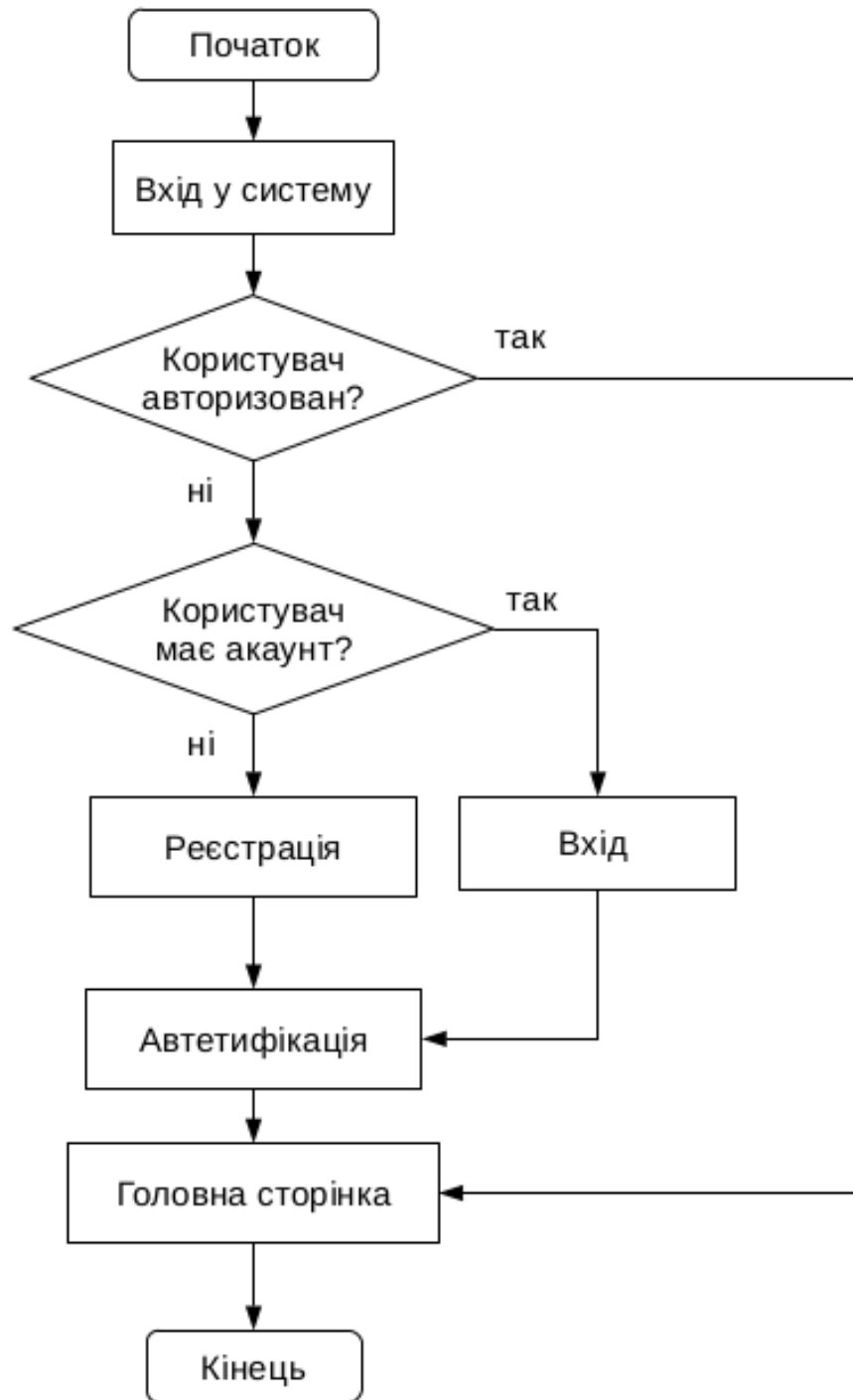


Рисунок 2.2. Блок-схема автетифікації Firebase [7]

Під час інтеграції з Firebase Authentication [7] ви можете визначити, хто має доступ до даних і як вони можуть отримати до них доступ. Онлайн-база даних є різновидом бази даних NoSQL, у порівнянні з реляційною базою даних, вона має різні оптимізації та функції. API бази даних реального часу [6] розроблено, щоб дозволити вам виконувати операції, які можна виконувати швидко. Це дозволяє створити чудовий досвід у реальному часі, обслуговуючи мільйони користувачів без шкоди для швидкості реагування. Тому важливо розглянути, як користувачам потрібно отримати доступ до ваших даних, а потім структурувати їх відповідно. Вирішили вибрати Firebase Authentication [7] від Google для аутентифікації в системі. Firebase Authentication [7] надає сервіс, простий у використанні SDK і готову бібліотеку інтерфейсу для аутентифікації користувачів у програмі. Він підтримує аутентифікацію за допомогою паролів, телефонних номерів, популярних постачальників облікових даних (таких як Google, Facebook і Twitter). Аутентифікація Firebase [7] тісно інтегрована з іншими службами Firebase і використовує галузеві стандарти, такі як OAuth 2.0 і OpenID Connect, тому її можна легко інтегрувати у ваш власний сервер. Блок-схема авторизації показана на рисунку 2.2.

Щоб увійти в програму, ми спочатку отримуємо облікові дані користувача. Ці облікові дані можуть бути адресою електронної пошти та паролем користувача, або це може бути маркер OAuth від федеративного постачальника облікових даних. Потім передайте ці дані в SDK для автентифікації Firebase. Після цього служба Google перевірить цю інформацію та поверне відповідь клієнту. Після успішного входу ви отримаєте доступ до основної інформації про користувача та зможете контролювати доступ користувачів до даних, що зберігаються в інших продуктах Firebase. Ви також можете використовувати наданий маркер

аутентифікації, щоб перевірити ідентичність користувача в службі сервера. За замовчуванням автентифіковані користувачі можуть читати та записувати дані в бази даних Firebase Realtime і Cloud Storage. Ви можете контролювати доступ цих користувачів, змінюючи правила бази даних реального часу Firebase і правила безпеки хмарного сховища. Мобільна версія системи паркувальних місць буде реалізована з використанням Ionic framework [8]. Ionic — це платформа для розробки додатків для веб-розробників, яка дозволяє використовувати єдину базу коду та відкриті веб-стандарти для створення гібридних кросплатформних мобільних додатків, веб-додатків і настільних додатків.

Ionic спеціально використовує Cordova як рівень API для взаємодії з рідними API, але тепер підтримує як Cordova, так і Capacitor, який є новим багатоплатформенним механізмом, вбудованим в Ionic. Додатки, які потрібно створити та розгорнути для роботи на різних платформах, таких як рідна iOS, Android, настільний комп'ютер та Інтернет, як прогресивні веб-додатки — усі вони використовують єдину кодову базу. Він реалізований на основі веб-стандартів. Платформа Ionic побудована на потужних стандартизованих веб-технологіях: HTML, CSS і JavaScript, використовуючи сучасні веб-API, такі як користувацькі елементи та Shadow DOM. Через це компоненти Ionic мають стабільний API і не є непостійними для одного постачальника платформи. Він має гарний дизайн, чистий, простий і практичний. Ionic Framework розроблено для ідеального запуску та відображення на всіх нестандартних платформах. Фреймворк Ionic дуже простий у створенні, тому створення програм Ionic є веселим, легким у освоєнні та доступним майже всім, хто має навички веб-розробки.

Ionic має вбудовану бібліотеку стандартних елементів, які можна використовувати подібно до елементів Bootstrap: картки, кнопки, перемикачі, сегменти, спливаючі вікна, поля введення, списки, сітки рядків і стовпців

тощо. За замовчуванням ці елементи змінюються, щоб програма виглядала як рідна для iOS та Android, але ви можете змінити їх зовнішній вигляд, якщо потрібно. Ionic також надає багато плагінів, які дозволяють використовувати апаратне забезпечення вашого смартфона (Ionic Native / Cordova). Але не забудьте переконатися, що ваша платформа активно підтримує вибрані плагіни. У більшості випадків такі плагіни працюють добре, але іноді вони можуть викликати помилки під час компіляції проекту або конфліктувати з плагінами для входу, які широко використовуються через Facebook або Firebase Analytics. Такі проблеми зазвичай вирішуються очищенням і повторною збіркою проекту або оновленням плагіна (за умови, що є нова версія, яка вирішує проблему). Ви також можете замінити плагіни альтернативними плагінами або, в деяких випадках, додати параметри, які дозволять вам замінити в `config/xml/`. Деякі плагіни вимагають точних налаштувань, таких як глибоке посилання, для належної роботи – модулі легко під'єднати, але важко працювати належним чином. Щоб увійти за допомогою Facebook на консолі розробника Facebook, вам потрібно додати `base64`, який використовується для підписання програми. Однак ви можете показати користувачам підказку, що ви можете вставити текст або посилання. Важливо використовувати тільки дійсні посилання HTML. Також використовуються такі плагіни: Поділитися в Twitter; Копіювати в буфер обміну; Виберіть зображення з галереї або скористайтесь камерою; Обрізати зображення; Блокування обертання екрана; Викличте редактор електронної пошти в браузері програми та локалізуйте або глобалізуйте.

Ionic також має велику перевагу - швидкість розвитку. Оскільки він заснований на Angular, ви можете запустити проект на Ionic у браузері та подивитися, як програма виглядає під час розробки. Щоб побачити такий попередній перегляд, не потрібно встановлювати додаток на смартфон або симулятор. Це значно економить час на зміну інтерфейсу користувача. Коли

ви маєте справу з функціями, які потрібно перевірити на смартфоні (наприклад, фотографування), потрібно всього кілька хвилин, щоб зібрати програму та встановити її на смартфон. На Android ви можете встановити програму безпосередньо з командного рядка, тоді як на iOS її потрібно відкрити в Xcode.

2.2 Налаштування бази даних та автентифікація

Як зазначалося вище, для автентифікації в системі було вирішено вибрати Firebase Authentication [7] від Google, а база даних у системі використовує Firebase Realtime Database [6]. У парях ці два API можуть задовольнити всі вимоги сервера. Як показано на рисунку 2.3. Тільки за допомогою адреси електронної пошти та пароля можна включити постачальника авторизації в системі. При відправці даних у Firebase Authentication [7] для авторизації, тобто надсилання електронної мережі та пароля, користувач отримає електронний лист із підтвердженням (Рисунок 2.4).

Провайдер	Состояние
✉ Адрес электронной почты и пароль	Включен
☎ Телефон	Отключено
🌐 Google	Отключено
🎮 Play Игры	Отключено
🎮 Game Center Beta	Отключено
📘 Facebook	Отключено
🐦 Twitter	Отключено
🐙 GitHub	Отключено
🅈 Yahoo	Отключено
🟨 Microsoft	Отключено
🍏 Apple Beta	Отключено
👤 Анонимный вход	Отключено

Рисунок 2.3. Параметры авторизации аутентификации Firebase [7]

Щоб підтвердити, натисніть створене посилання. Після цього користувач увійде в систему. Аутентифікація Firebase [7] дозволяє увійти в систему іншими способами. Увійти в систему можна по телефону. Вам потрібно надіслати номер телефону, і система відповість повідомленням, що містить код для підтвердження номера телефону.

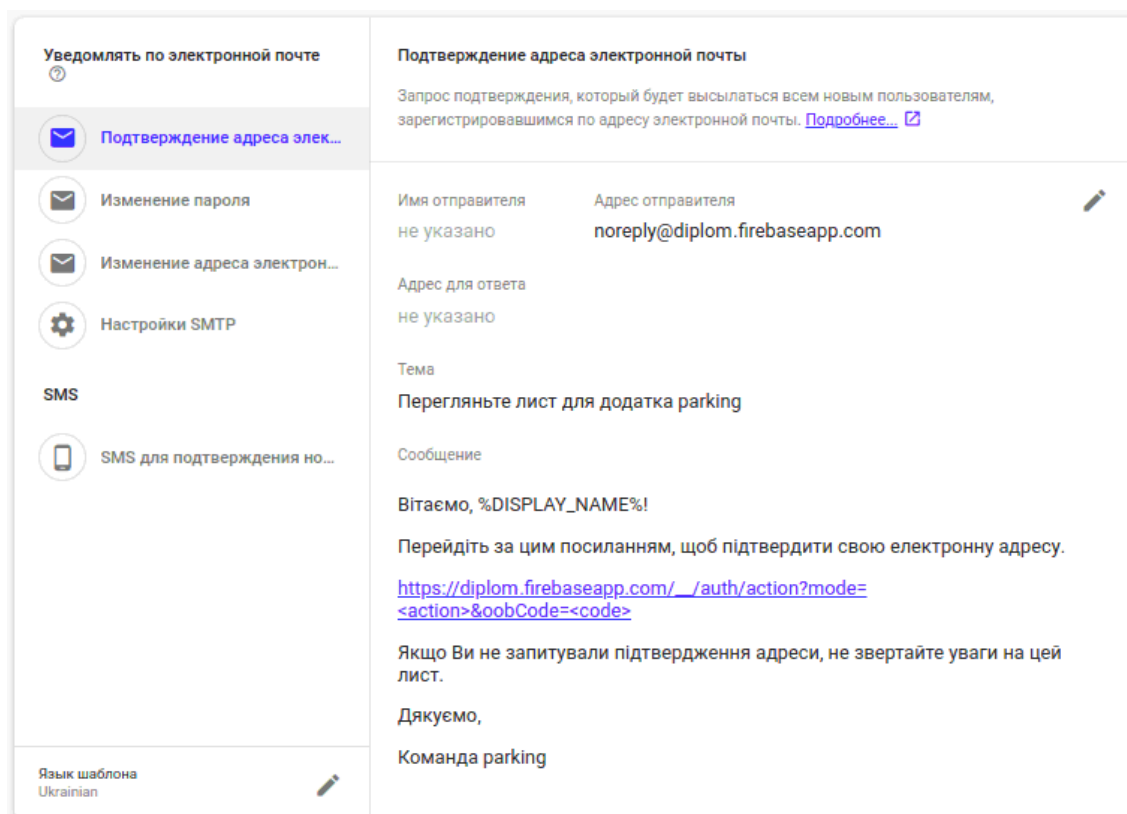


Рисунок 2.4. Шаблон підтвердження електронної пошти для автентифікації Firebase [7]

Система також може використовувати інші системи для входу, такі як Google, Play Games, Game Center, Facebook, Twitter, GitHub, Yahoo, Microsoft, Apple. Якщо ця функція ввімкнена в Firebase Authentication [7], користувач повинен надати цей метод авторизації. Після успішної авторизації користувач буде авторизований в системі та надано дозвіл на використання системи. Також в сервісі є можливість пройти авторизацію без підтвердження в гостьовому режимі. Якщо дати користувачеві таку можливість, то він зможе авторизуватися без авторизації. Авторизовані користувачі служби Firebase Authentication [7] дозволяють змінити пароль або адресу електронної пошти. Користувач отримає електронний лист із підтвердженням нового листа. Після підтвердження пароль буде змінено або буде встановлено нову адресу електронної пошти.


```
{
  "rules": {
    ".read": true,
    "parkingLastId" : {
      ".write": "auth !== null"
    },
    "parkings" : {
      ".write": "auth !== null"
    },
    "companies": {
      "$uid": {
        ".write": "$uid === auth.uid"
      }
    },
    "users": {
      "$uid": {
        ".write": "$uid === auth.uid"
      }
    }
  }
}
```

Рисунок 2.3. Параметри авторизації автентифікації Firebase [7]

Щоб підтвердити, натисніть створене посилання. Після цього користувач увійде в систему. Аутентифікація Firebase [7] дозволяє увійти в систему іншими способами. Увійти в систему можна по телефону. Вам потрібно надіслати номер телефону, і система відповість повідомленням, що містить код для підтвердження номера телефону.

Для використання системи необхідно налаштувати базу даних, як згадувалося раніше, база даних буде використовувати базу даних Firebase Realtime [6]. Він працює в режимі реального часу, тобто при зміні будь-яких даних або додаванні нових даних всі користувачі отримують нові дані, і користувачі навіть не будуть сумніватися, що зможуть використовувати останню версію даних в базі даних. Система налаштувала підписку на отримання нових даних. Після отримання даних система негайно оновлює весь вміст, пов'язаний з ними. У цьому полягала основна перевага даної послуги. Для редагування, видалення або створення нових даних у базі даних

потрібно вказати правила для цих ситуацій у налаштуваннях бази даних (Рисунок 2.5.).

На рисунку 2.5. Показує правила використання бази даних. Правило записується як об'єкт. Щоб налаштувати читання даних з бази даних, потрібно вказати правила для поля «.read». Якщо це false, користувач не зможе переглянути ці дані, і йому буде відмовлено в доступі. Якщо це правда, користувач може прочитати дані з бази даних, і йому буде дозволено доступ до них. Те ж саме використовується для запису, редагування та видалення. Вам потрібно написати правила для поля ".write", якщо помилковий доступ буде заблоковано, якщо буде надано true-доступ і користувачі зможуть виконувати ці операції. Щоб налаштувати лише для читання чи запису для авторизованих користувачів, тоді для результату «.read» або «.write» ви повинні призначити результат: «auth!===null» відповідно. Firebase перевірить, чи авторизований користувач, і замінить результат у правилі, як показано нижче. Якщо користувач авторизований, результат є істинним, і користувач матиме права доступу на читання або запис. Якщо користувач не авторизований, результат хибний, і користувачу буде відмовлено в доступі читання або запису. Щоб налаштувати доступ до персональних даних користувача та заблокувати інших, вираз «\$ uid === auth.uid» необхідно записати в базу даних, щоб отримати результат «.read» або «.write». Коли користувачі записують персональні дані в базу даних, ці поля будуть позначені як унікальний ідентифікатор "uid". Перш ніж читати або редагувати ці дані, Firebase порівнює "uid", записаний в базу даних, з "uid" авторизованого користувача. У разі успіху буде надано доступ, і користувач прочитає або змінить дані в базі даних, якщо результат невдалий, унікальний ідентифікатор буде іншим, і користувач не зможе читати або редагувати ці дані.

2.3 Огляд давачів руху для облаштування парковок

Датчик руху — це сигнальний пристрій, який виявляє рух об'єкта і використовується для моніторингу навколишнього середовища або автоматичного запуску необхідних дій у відповідь на рух об'єкта. Детектор руху - пристрій або дія системи охоронного телебачення, що формує тривожний сигнал при виявленні руху в полі зору камери [9]. Більш чутливий датчик руху також називають датчиком присутності. Датчики руху дуже поширені, і аналітики очікують, що їх використання збільшиться ще на 13-14%. Датчики руху та присутності широко використовуються самостійно або в складі систем безпеки для виявлення вторгнення сторонніх осіб, а також для автоматизованого освітлення та кондиціонування повітря (опалення та кондиціонування) у квартирах, житлових будинках та комерційній нерухомості.

Робота датчика руху заснована на аналізі різних типів хвиль (акустичних, світлових або радіохвиль) з навколишнього середовища. За типом використовуваної хвилі датчики руху поділяються на: інфрачервоний, УЗД, Фотоелектричні, з використанням звичайного світла, мікрохвильова піч, Томографія з використанням радіохвиль. Залежно від того, чи сам датчик ініціює ці хвилі та аналізує їх після відбиття, чи приймає хвилі лише із зовнішнього світу, датчики поділяються на: позитивний; Пасивний; Комбінація, коли одна частина датчика посилає хвилі, а інша частина відокремлюється від неї для їх прийому. Більшість існуючих датчиків руху є комбінацією цих стандартів, і датчики одного типу хвиль зазвичай використовують механізм для їх створення та обробки. Найбільш поширеним: Пасивний інфрачервоний датчик (PIR), в принципі, найбільш

легкодоступний і широко використовуваний датчик руху, на інфрачервоні датчики припадає близько 50% світових датчиків руху; Активні ультразвукові, мікрохвильові та томографічні датчики; Комбінація фотоелектричних та інфрачервоних датчиків.

Кожен механізм має свої помилки, і час від часу допускаються помилкові спрацьовування. Щоб зменшити ймовірність помилкових тривог, датчики іноді поєднують дві технології в одному пристрої (наприклад, інфрачервону та ультразвукову). Однак це, в свою чергу, збільшує вразливість датчика, оскільки він стає менш чутливим і, отже, може не працювати, навіть якщо повинен. Роль інфрачервоних датчиків заснована на аналізі теплового (інфрачервоного) випромінювання. Пасивні інфрачервоні датчики (PIR) не випромінюють випромінювання, а лише аналізують вхідні теплові промені [10]. У середині датчика є два чутливі компоненти, які можуть визначити рівень інфрачервоного випромінювання. Перед кожним знаходиться лінза Френеля, яка фокусує інфрачервоні промені, що падають на датчик на ньому. Найпростіший датчик призначений для поділу навколишнього простору на дві лінзи, кожна з яких проектує теплове випромінювання зі своєї зони відповідальності на свої чутливі компоненти. У нормальних умовах випромінювання двох частин датчика приблизно однаково. Коли з'являється гарячий об'єкт, він спочатку потрапляє тільки в одну частину датчика, тому показання двох чутливих елементів починають відрізнятися, і датчик робить висновок, що є рух.

У реальних ситуаціях датчик з двома лінзами буде занадто грубим, тому на практиці датчик оснащений не парою лінз, а десятками їх. Їх легко розгледіти на поверхні — стільниковій структурі напівпрозорих вікон, за якими знаходяться чутливі елементи. З метою економії місця та матеріалів сенсор розроблений таким чином, що всі лінзи фокусують вхідне випромінювання лише на два чутливі елементи. Тому навколишній простір

поділяється на зони відповідальності між парами кадрів, кожен постріл здатний зафіксувати рух у своїй зоні. Ультразвукові датчики засновані на аналізі звукових хвиль, які перевищують поріг сприйняття людиною. Спеціальний елемент всередині датчика періодично випромінює ультразвуковий промінь. Потім датчик переходить в режим прийому і чекає повернення відбитої хвилі, а потім аналізує її. Якщо стан зони покриття датчика залишається незмінним, відображення хвиль, що випромінюються щоразу, однакові; але якщо рух починається, хвиля змінюється (ефект Доплера), і датчик робить висновок, що ситуація змінилася. Коли зміна перевищує встановлений поріг чутливості, спрацьовує датчик. Як генератор ультразвуку в датчику зазвичай використовуються кварцові або керамічні п'єзоелектричні елементи або спеціальні плівки, які вібрують під дією електростатичного поля. Томографія (радіохвилі) та мікрохвильові датчики [11] працюють так само, як ультразвук, але аналізують відбиття радіохвиль, а не відбиття звукових хвиль. Оскільки радіохвилі можуть проходити через неметалеві бар'єри, такі як стіни та дерев'яні меблі, радіохвильові датчики підходять для моніторингу простору від таких перешкод. Радіохвильові датчики дуже дорогі, тому їх часто використовують для моніторингу великих комерційних площ, таких як склади.

Принцип роботи фотоелектричного датчика заснований на перериванні контрольного променя в тінь його роботи. Як правило, датчик складається з двох частин, одна з яких випромінює світло, а інша приймає світло. У приймальній частині є фотодетектор, а під впливом падаючого світла є електричний струм. Коли промінь блокується будь-яким предметом, світло перестає падати на приймач і спрацьовує датчик. Відомий приклад використання таких датчиків — у поворотних дверях метрополітену ці датчики відключаються для пасажирів, коли вони проходять крізь промінь, не сплачуючи за проїзд. Невидиме інфрачервоне випромінювання часто

використовується у фотоелектричних датчиках. Розглянемо систему пошуку кінцевих точок логістичного процесу. Нова система матиме більш детальну інформацію про поточний стан паркування, тобто конкретне місце на стоянці, щоб кожен користувач міг бачити місце розташування на карті та знати, чи безкоштовне воно онлайн. Система буде містити два режими: режим користувача та режим підприємства. У режимі користувача відобразатиметься карта паркування. Користувачі можуть переглядати інформацію про паркування. У режимі підприємства можна використовувати паркінги, коригувати тарифи, редагувати або створювати нові паркінги.

Отже, розглянута технологія, яка буде використана для створення системи знаходження кінця логістичного процесу. Система буде створена як веб-сторінка, створена за допомогою фреймворку Angular. Для відображення карти буде використовуватися API Карт Google. Системна база даних базуватиметься на базі даних Google Firebase у реальному часі. Система також використовуватиме службу аутентифікації від Google-Firebase Authentication. Покажіть, як налаштувати правила читання та запису в базі даних і як увійти в систему. Для мобільної версії системи було вирішено використовувати фреймворк Ionic. Було переглянуто тип датчика руху та обрано відповідний тип датчика, який буде встановлено на стоянці.

3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ПОШУКУ КІНЦЕВИХ ТОЧОК ЛОГІСТИЧНОГО ПРОЦЕСУ

3.1 Вимоги та опис системи кінцевих точок логістичного процесу

Для демонстрації цієї роботи було вирішено розробити програмний продукт для системи пошуку паркувальних місць. Розроблена програма має бути простою, зрозумілою у використанні та забезпечувати продуктивність для наступних завдань: 1. Знайдіть місце для паркування. 2. Пошук фільтрів. 3. Створення та редагування нових паркувальних місць. 4. Сформулюйте плату за паркування. Для виконання цього завдання виберіть платформу JavaScript, щоб використовувати Angular Framework 7 для написання програм, для мобільної версії використовуйте Ionic Framework, що дозволяє створювати мобільні додатки для операційних систем Android та iOS. Виберіть базу даних Google Firebase Realtime Database як базу даних. Система також використовуватиме службу аутентифікації від Google-Firebase Authentication. Вони взаємодіють разом, ви можете налаштувати базу даних і керувати користувачами, а конфігурація користувача дозволяє налаштувати правила використання бази даних. Для коректної роботи програми визначаються наступні мінімальні характеристики: Пристрої під керуванням IOS версії 12.4.4; Пристрої під управлінням Android версії 5.0; Стабільне підключення до Інтернету; Google Chrome версії 54; Mozilla Firefox версії 71; Safari версії 5.1.7.; 2 ГБ ОЗУ на мобільному пристрої Частота процесора 2,2 ГГц.

Існуючі системи забезпечують подібний набір функцій. Основне завдання, яке вони виконують, - знайти підходяще місце для паркування для користувача. Кожна система має свою унікальність. Деякі люди можуть оплатити паркування за допомогою кредитної картки або програми. Деякі більше схожі на каталоги з різною інформацією про різні стоянки. Деякі

мають різні фільтри для пошуку відповідних місць для паркування. Деякі мобільні додатки виглядають красиво і стильно і користуються великою популярністю у користувачів. Основна ідея нової системи пошуку кінцевих точок логістичного процесу полягає в тому, щоб додати більш детальну інформацію про поточний стан паркінгу, тобто конкретне місце на стоянці, щоб кожен користувач міг бачити місце розташування на карті, і знати, чи це безкоштовно. Система буде містити два режими: режим користувача та режим підприємства.

Режим користувача дозволить користувачам реєструватися в системі та переглядати всі місця для паркування на карті, користувачі бачитимуть безкоштовні місця для паркування на карті та зможуть переглядати інформацію про них. На карті будуть маркери трьох кольорів: зелений, червоний та оранжевий. Кожен колір позначки інформує користувача про поточний стан паркування. Зелений знак означає, що паркування безкоштовне і паркувати можна, червоний знак означає, що паркуватися не можна з технічних причин, а оранжевий знак означає, що місце для паркування зайняте. Власнику паркування потрібна бізнес-модель, щоб ви могли додати свою парковку в систему, додати тариф, групу, і користувач побачить усі додані паркування на карті. У корпоративній моделі буде розроблено можливість додавати місця для паркування, тобто вибрати відповідне місце на карті, записати назву, основну інформацію, додати до місця розташування та вибрати плату за паркування. Ви можете редагувати створену стоянку, змінювати її розташування, плату та назву. Уміння створювати простір вимагає від них об'єднання паркінгів у групу. У деяких місцях можна вибрати стандарт зарядки, який буде використовуватися на всіх паркінгах. Ви також можете вибрати плату за саму автостоянку, щоб вибрати інший тариф для паркування. Можливість додавати та редагувати тарифи.

При створенні тарифу потрібно написати назву на погодинній ціні. Після цього ви можете вибрати тариф на місці або на стоянці.

Інформаційні маркери також корисні для компаній, ви можете в будь-який момент перевірити маркери на карті, можете передати стоянку на технічне обслуговування, і маркери стануть червоними. Нова система в кінці логістичного процесу реалізує веб-сторінку. Для веб-сторінок є два режими роботи: режим користувача та режим підприємства. Власникам автостоянок потрібна корпоративна модель. Щоб додати та налаштувати свою стоянку, вам потрібно зареєструватися, а потім ви зможете додати розташування цих автостоянок на карту. Крім того, ви можете встановити різні тарифи для кожного паркувального місця, встановити плату за паркування, встановити години роботи паркінгу, встановити тарифи на вихідні та налаштувати різні послуги, які можуть надавати паркування. Кожна стоянка має датчик, відображений на карті, з позначкою, яка може бути трьох кольорів: якщо позначка червона, це означає, що парковка зайнята, якщо вона зелена, ви можете паркуватися безкоштовно, якщо вона помаранчева, ви не можете тимчасово зупинитися для виконання технічних робіт.

Користувачеві потрібен режим користувача. Щоб користуватися системою, користувачеві необхідно зареєструватися. Після цього користувач може побачити різні маркери одного кольору на карті, і натиснути на нього, щоб побачити інформацію про паркінг. Виберіть правильний і прокладіть маршрут до стоянки.

3.2 Структура системи пошуку кінцевих точок логістичного процесу

У процесі розробки було визначено два основних етапи: розробка моделі користувача та розробка моделі підприємства. Проектне рішення складається з трьох модулів: Увійти Карта середовище Модуль входу

необхідний для входу та реєстрації нових користувачів. Він реалізує функції реєстрації, перевірки та відправки на сервер для перевірки даних та отримання відповідей. Модуль Мар є основним модулем програми. Він обробляє основні процеси в системі. Завантажте Google Maps і позначте на ньому. Перед початком роботи модуль відправляє в базу даних запит на отримання даних користувача, а відповідно до відповіді надсилає запит на отримання всіх паркувальних місць і пов'язаної інформації. Після отримання всіх необхідних даних починається процес відображення даних в Google Maps і підключення до необхідних сервісів для використання карти. Цей модуль виконує процес редагування та створення нового паркінгу або тарифу. Після того, як користувач створить новий тариф або додасть новий тариф, відредагує існуючу стоянку або налаштує тариф по-іншому, модуль надсилає відредаговані дані до бази даних. База даних перевіряє, чи має користувач право редагувати або створювати, а потім нові дані були збережені. Після оновлення даних база даних надішле відредаговані дані всім користувачам, а головний модуль негайно оновить усі дані. Модуль налаштування займається редагуванням і створенням паркінгів і тарифів. Модуль «Карта» має функцію редагування. Коли користувач вирішує відредагувати дані, модуль «Карта» викликає функцію в модулі «Налаштування». Також можливе створення нових паркінгів або тарифів. У цьому модулі відображається вікно, в якому користувач може заповнити дані, після заповнення всіх необхідних полів інформація передається в модуль карти. Після передачі даних модуль Мар продовжує обробляти дані та надсилати їх до бази даних.

3.3 Розробка інтерфейсу системи кінцевих точок логістичного процесу

Основою взаємодії користувача з системою є інтерфейс. Першим кроком є створення вікна реєстрації (Рисунок 3.1.), а потім використання

головного меню системи (Рисунок 3.3.). При реєстрації користувач може вибрати саме той спосіб реєстрації, який йому потрібен. Доступні два режими: режим користувача та режим підприємства.

3.3.1 Корпоративний режим

Для реєстрації в корпоративному режимі при реєстрації необхідно вибрати відповідний режим, зареєструватися на сайті та ввести адресу електронної пошти, пароль та інформацію про компанію (Рисунок 3.1.).

The image shows a registration form with two tabs at the top: 'Режим користувача' (User mode) and 'Режим підприємства' (Corporate mode). The 'Режим підприємства' tab is selected. Below the tabs are four input fields: 'Введіть ваше ім'я', 'Введіть e-mail', 'Введіть пароль', and 'Введіть пароль ще раз'. Each password field has a small eye icon to the right, indicating a password visibility toggle. At the bottom of the form is a button labeled 'Створити' (Create).

Рисунок 3.1. Зареєструйтеся на сайті

Після реєстрації на електронну пошту буде надіслано підтвердження. Перевірка на місці також може бути проведена під час реєстрації. Якщо ви неправильно ввели адресу електронної пошти або пароль, ви побачите повідомлення про помилку та виділене поле. Користувачу необхідно ввести правильні дані. Якщо дані введені неправильно, реєстрація не працюватиме.

Якщо користувач вже зареєстрований, йому достатньо увійти в систему. Вам потрібно ввести свою електронну адресу та пароль, а потім натиснути «Увійти». Система сама розпізнає, в якому режимі перебуває зареєстрований

користувач, і перенаправляє у відповідний режим. На блок-схемі моделі підприємства можна побачити можливості реалізації та кроки цієї моделі (Рисунок 3.2.). Після успішної авторизації користувач буде перенаправлено в головне меню (Рисунок 3.3.), де відобразиться карта, що містить створені користувачем місця для паркування. Ви можете натиснути на будь-яку позначку та відредагувати її тариф або інформацію (Рисунок 3.4.). Відкриється вікно з інформацією про стоянку, а саме: назва стоянки та її ідентифікаційний номер, місцезнаходження стоянки та поточна плата.

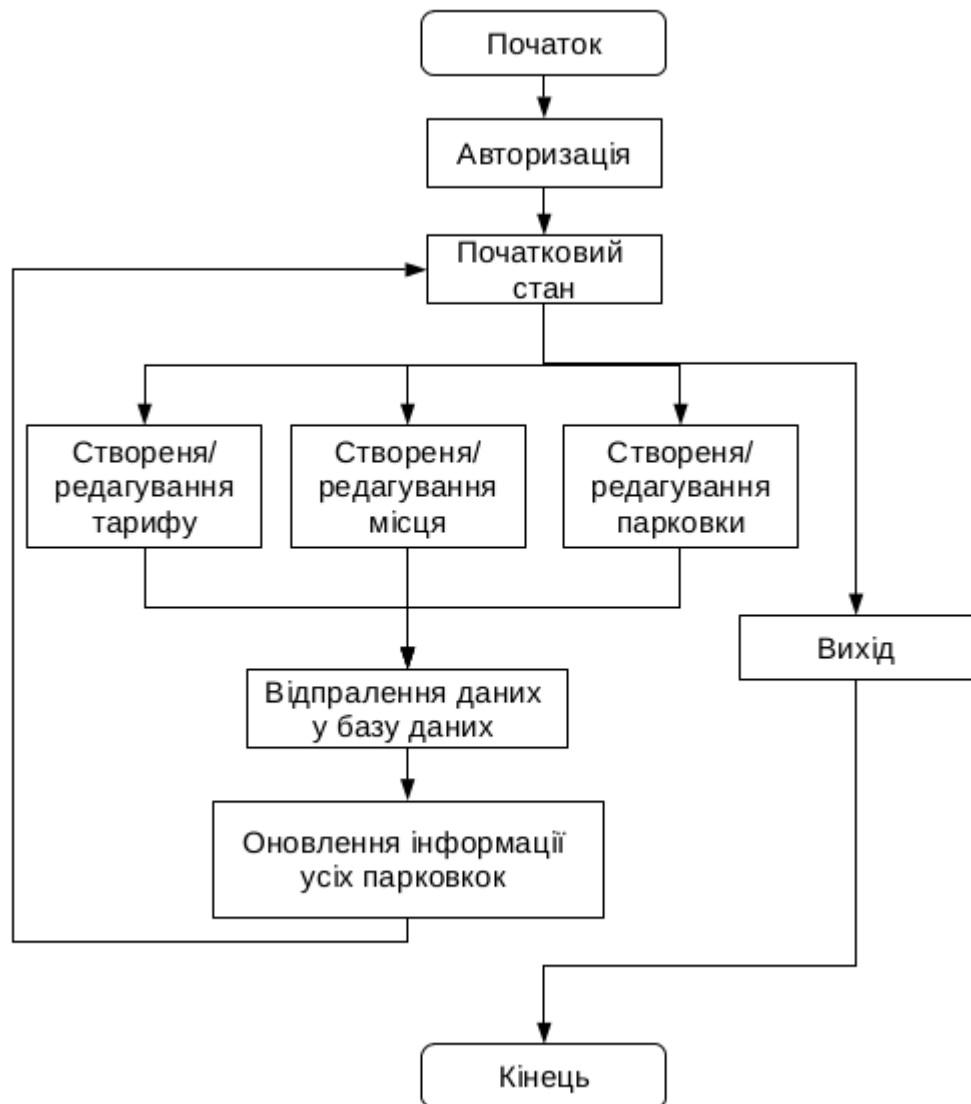
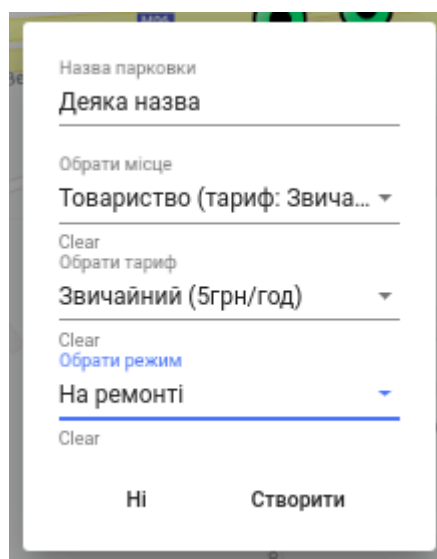


Рисунок 3.2. Блок-схема моделі підприємства

паркування вільне, червона позначка вказує на те, що стоянка зайнята, а помаранчева позначка вказує на те, що паркінг на технічному обслуговуванні. Щоб додати новий маркер, вам потрібно клацнути на карті, карта буде збільшена, вам потрібно натиснути, щоб додати нове місце, і з'явиться вікно для підтвердження створення нового розташування. Ви можете скасувати створення нового паркувального місця, натиснувши «Ні». Створення автостоянки буде скасовано, або користувач може натиснути кнопку «Так», щоб створити нове місце, після чого користувач відкриє нове вікно (Рисунок 3.5.), вам потрібно записати назву автостоянки. автостоянку та виберіть створене місце, щоб додати нове. Паркінг приєднано до наявного розташування. Також потрібно вибрати тариф, а потім при створенні паркінгу він буде прив'язаний до місця розташування та тарифу, який можна переглянути, натиснувши позначку. Нарешті, вам потрібно вибрати режим. Ви можете вибрати «Безкоштовно», тоді після створення автостоянки вона буде доступна всім користувачам відразу, ви можете вибрати «Ремонт», а потім після створення маркера на карті він стане оранжевим і користувачів не буде можна використовувати, але це буде показано на карті. Ви також можете скасувати створення, натиснувши кнопку «Ні».



Назва парковки
Деяка назва

Обрати місце
Товариство (тариф: Звича... ▾

Clear
Обрати тариф
Звичайний (5грн/год) ▾

Clear
Обрати режим
На ремонті ▾

Clear

Ні Створити

Рисунок 3.5. Вікно створення паркування

У лівій частині карти ви можете редагувати назву компанії, додати нове місцеположення або відредагувати існуюче місцеположення, додати нове місцеположення або змінити існуючий тариф. Щоб відредагувати назву компанії, потрібно натиснути кнопку «Редагувати назву компанії» зліва на екрані. Відкриється вікно редагування назви, а в поле введення тексту буде записана стара назва. Ви можете редагувати і підтвердити або скасувати редагування назви компанії.

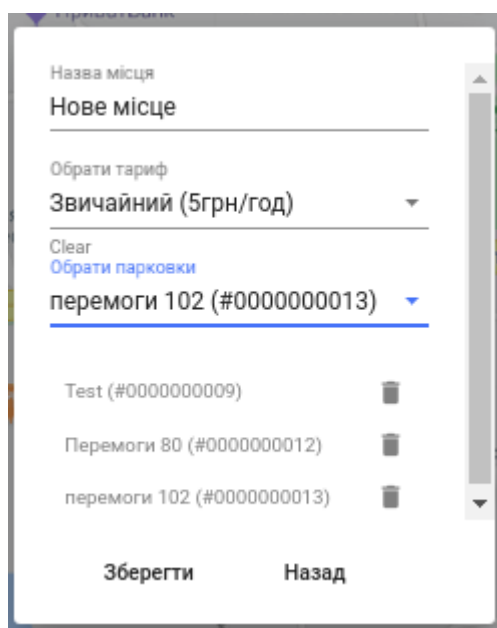


Рисунок 3.6. Вікно створення місця

Якщо ви хочете створити нове місце, вам потрібно натиснути кнопку «Додати місце» в лівій частині карти. Перед користувачем відкриється вікно для створення місця (Рисунок 3.6.). У цьому вікні потрібно написати назву нового місця, вибрати тариф, який буде створено, а потім після його створення на всіх паркінгах, приєднаних до цього місця, буде вибрано тариф місця, крім тих паркувальних місць ділянок, на яких уже вибрано кілька паркувальних місць. Підвищити тариф. Зі списку паркувальних місць ви можете вибрати потрібне місце для паркування, яке буде прикріплено до

цього місця. Після створення нового місця вся інформація про паркування для всіх користувачів буде оновлена негайно, інформація про місце розташування, інформація про новий тариф буде оновлена, і це стосується паркувальних місць, приєднаних до створеного нового місця. Якщо ви хочете відредагувати місце, вам потрібно вибрати місце зі списку в лівій частині карти. Відкриється вікно підтвердження, де ви можете відредагувати місце або видалити його. Якщо вибрати «Редагувати розташування», відкриється те саме вікно, що й у момент його створення (Рисунок 3.6.). Усі дані будуть записані у відповідні поля. Після успішного редагування нова інформація про всіх користувачів буде негайно оновлена.

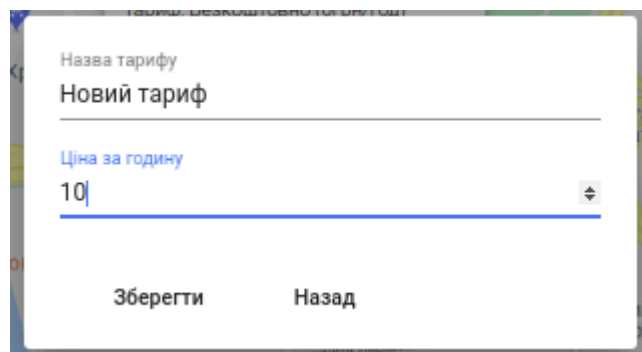


Рисунок 3.7. Вікно для створення тарифу

Якщо ви хочете створити новий тариф, вам потрібно натиснути кнопку «Додати тариф» з лівого боку картки. Перед користувачем відкриється вікно створення тарифів (Рисунок 3.7.). У цьому вікні потрібно записати назву нового тарифу та ціну на наступну годину. Після створення ви можете використовувати нові тарифи під час створення або редагування нових локацій та створення або редагування паркінгів. Щоб відредагувати існуючий тариф, потрібно вибрати потрібний тариф зі списку зліва на карті, після чого відкриється вікно створення нового тарифу. Після успішного редагування вся нова інформація буде оновлена для всіх користувачів.

3.3.2 Користувацький режим

Для реєстрації в режимі користувача при реєстрації необхідно вибрати відповідний режим. Для корпоративного режиму необхідно зареєструватися на сайті, ввести свою електронну пошту, пароль та інформацію про себе (Рисунок 3.1.). Після реєстрації на електронну пошту буде надіслано підтвердження. Якщо ви неправильно ввели адресу електронної пошти або пароль, з'являться помилки та виділені поля (Рисунок 3.2.). Користувачеві необхідно ввести правильні дані. Якщо дані введені неправильно, реєстрація не працюватиме. Після того, як користувач успішно зареєструвався на цьому сайті, він може увійти в систему, при вході йому необхідно ввести адресу електронної пошти та пароль та натиснути «Увійти».

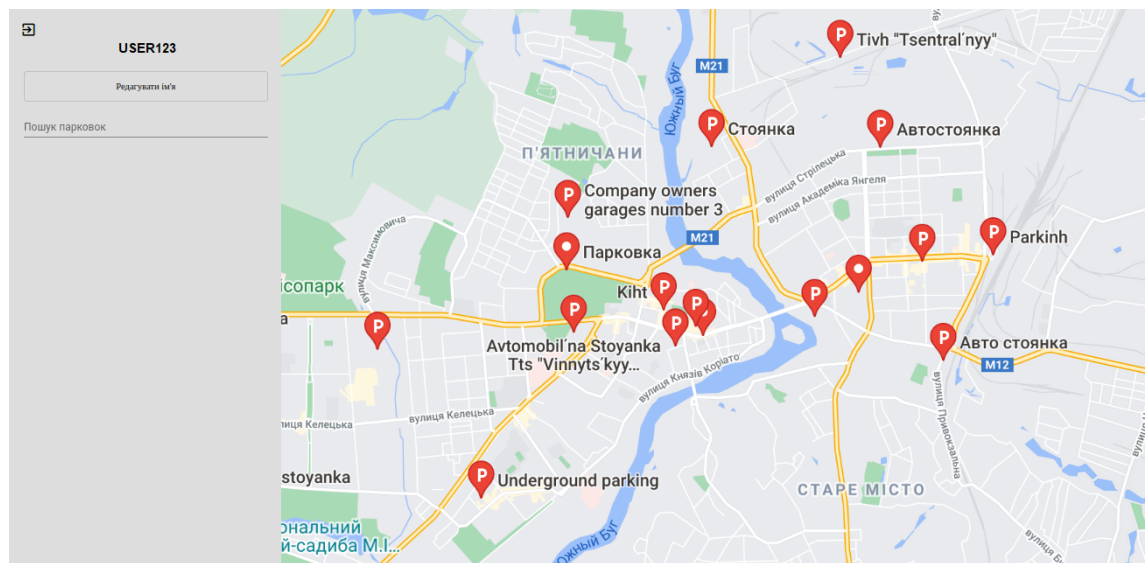


Рисунок 3.8. Головне меню в режимі користувача

Система перевірить, чи зареєстрований користувач. Після успішної авторизації система перенаправить користувача в головне меню режиму користувача (Рисунок 3.8.). На блок-схемі в режимі користувача можна побачити можливості та кроки в цьому режимі (Рисунок 3.10.).

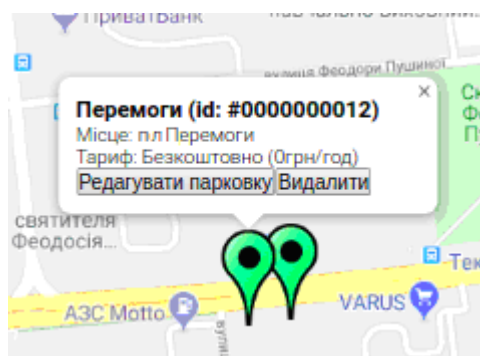


Рисунок 3.9. Інформація про паркування

У головному меню відобразиться карта, яка містить усі паркувальні місця, створені в системі. Ви можете натиснути на будь-яку позначку та переглянути інформацію про цю стоянку (Рисунок 3.9). Відкриється вікно з інформацією про стоянку, а саме: назва стоянки, місцезнаходження стоянки, поточна плата та стан паркування. У цьому вікні також є кнопка «припаркувати», якщо натиснути на неї, користувачеві буде запропоновано прокласти маршрут від місця розташування користувача до паркінгу, стан паркування стане зайнятим, і всі користувачі отримають оновлену інформацію. . Ви можете змінити назву в лівій частині карти. Щоб відредагувати ім'я, натисніть кнопку «Редагувати ім'я» в лівій частині екрана. Відкриється вікно редагування імені, і в поле введення тексту буде записано старе ім'я. Ви можете підтвердити або скасувати редагування імені. Під кнопкою знаходиться поле для пошуку місць для паркування. У цьому полі ви можете ввести адресу, і користувачі побачать список паркувальних місць біля адреси.



Рисунок 3.10. Алгоритм для користувацького режиму

3.4 Управління авторизованими користувачами та базою даних

Налаштування авторизації реалізовані за допомогою служби аутентифікації Firebase [7], а база даних реалізована за допомогою служби баз даних реального часу Firebase [6]. Щоб користуватися цими сервісами, потрібно зареєструватися в Google Developers. Після реєстрації розробники зможуть отримати доступ до всіх сервісів, які надає Google.

Идентификатор	Поставщики	Время создания	Последний вход	Уникальный идентификатор пользователя ↑
botanic-plus@urk.net	✉	4 сен. 2021 г.	4 сен. 2021 г.	16Mzxeo1VbeNC0XZwhAgKkGYU...
alexander@gmail.co...	✉	4 сен. 2021 г.	8 сен. 2021 г.	yZjGRBkXQTaffqpFm7dVov3ZtCy2

Количество строк на странице: 50 1-2 из 2

Рисунок 3.11. Авторизованный користувач

Щоб керувати користувачами, перейдіть на свою сторінку та виберіть вкладку Користувачі в розділі «Автентифікація». Відобразиться список всіх авторизованих користувачів (Рисунок 3.11). Інформація про кожного користувача, метод авторизації та унікальний ідентифікатор. Тут можна вручну додавати нових користувачів, видаляти, скидати паролі або вимкнути авторизованих користувачів. Для налаштування та редагування бази даних необхідно вибрати створену базу даних у розділі «База даних». Щоб налаштувати правила та доступ до редагування або видалення даних, перейдіть на вкладку «Правила» (Рисунок 2.4). Там ви можете налаштувати права доступу для користувачів, детальну інформацію про налаштування прав доступу можна знайти в розділі 2. Щоб переглянути та відредагувати дані в базі даних, перейдіть на вкладку Дані. Відобразяться дані бази даних (Рисунок 3.12). Дані відображаються у вигляді об'єкта з полями. Поле `ParkingLastId` – це лічильник, який використовується для визначення нового паркінгу. Після створення нового паркінгу лічильник збільшиться на одиницю. Є дані про всі паркомісця на замських паркувальних місцях.

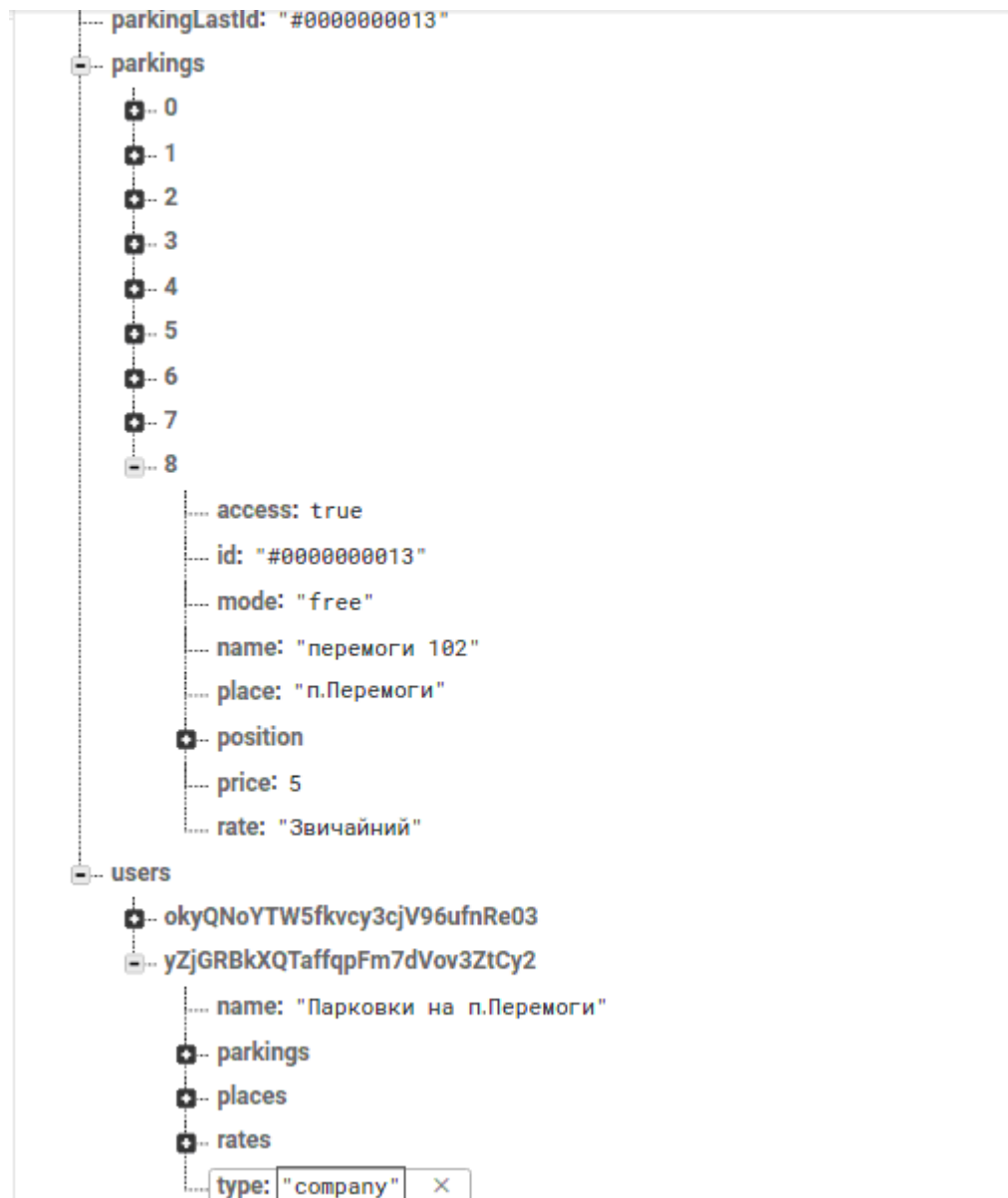


Рисунок 3.12. Фрагменти даних з бази даних

На стоянці міститься така інформація: Умови паркування; дозвіл на паркування; назва стоянки; Координати; Тариф; Де розташована автостоянка; Ціна однієї години. База даних зберігає інформацію про всіх користувачів, і вона знаходиться в папці користувача. У папці ви можете побачити список авторизованих ідентифікаторів користувачів, кожен ідентифікатор зберігає інформацію про кожного користувача. Список інформації про користувача в режимі зареєстрованого підприємства: Назва

компанії; Парковки для всіх користувачів; Створити місце; Встановлювати тарифи; тип користувача. Список інформації про користувача, зареєстрованої в режимі користувача: Ім'я ідентифікатор зайнятого паркувального місця; тип користувача.

3.5 Тестування системи пошуку кінцевих точок логістичного процесу

Структура та інтерфейс системи розглянуті вище. Тест системи буде представлений у вигляді таблиці. Тест виконується мобільною версією та веб-сторінкою впровадженої системи. Тестовими даними будуть запуск системи, час обробки авторизації та реєстрації, час обробки завантаження головного меню, пошук паркування, створення нового тарифу та паркування, редагування тарифу та паркування. Результати тесту наведені в таблиці. Використовуйте комп'ютер із такими характеристиками, щоб перевірити веб-сторінки: Операційна система: Ubuntu 16.04 LTS x64; Процесор: Intel Core i5-5200 CPU 2,20GHz * 4; Пам'ять: DDR4 12 Гб; Графіка: Intel HD Graphics 5500. Функція мобільного телефону: Операційна система: Android 6.0; Процесор: MediaTek MT6753 + GPU ARM Mali-T720 1,3GHz * 8; Пам'ять: 3 Гб.

Це видно з таблиці 3.1. Для роботи на мобільних пристроях потрібно більше часу. Це залежить від особливостей самої технології. Якщо подивитися на характеристики мобільного пристрою тестової системи, то можна сказати, що параметри пристрою невисокі. Якщо тестувати на більш продуктивному пристрої, то час обробки системи буде значно меншим. Якщо ви подивитесь на час виконання веб-сторінки, час виконання однаковий або може відрізнятися, але різниця буде не дуже великою. Найголовніше, щоб комп'ютер відповідав мінімальним характеристикам системи.

Тому в цьому розділі розглядається посібник із використання системи пошуку кінцевих точок логістики. Система передбачає два режими: режим користувача та режим підприємства. Перегляньте детальну інформацію про використання системи в кожному режимі. У режимі користувача користувачі можуть отримати доступ до всіх парковок, зареєстрованих в системі, і шукати місця для паркування, змінюючи їх адреси та назви. У режимі підприємства користувачі мають можливість створювати нові паркінги, налаштовувати та редагувати їх. Ви також можете створювати, видаляти та редагувати місцеположення, де можна прикріплювати місця для паркування, а також можливість створювати, видаляти та редагувати тарифи. Після виконання будь-якої операції інформація всіх користувачів системи буде оновлена. Налаштування та елементи керування авторизацією користувачів переглянуто. Ви можете видалити їх із системи, створити нові та скинути паролі користувачів. Переглядайте базу даних, вмійте налаштовувати правила та права доступу для кожного користувача, редагувати, видаляти або додавати нові дані до бази даних. Проведено тестування мобільного додатка та веб-сторінки, за результатами яких зроблено висновки та пояснення.

Таблиця 3.1 - Тестування можливостей системи

№	Об'єкт тестування	Час виконання у мобільній версії	Час виконання веб-сторінкою
1	Авторизація користувача	~0.5012	~0.4511
2	Реєстрація користувача	~0.9108	~0.8925
3	Запуск системи	~3.1707	~2.8601
4	Створення нової парковки	~0.8487	~0.7992
5	Створення нового тарифу	~0.5578	~0.4135

Продовження таблиці 3.1

6	Редагування парковки	~0.8159	~0.7811
7	Редагування тарифу	~0.5091	~0.3973
8	Видалення парковки	~0.3328	~0.2981
9	Видалення тарифу	~0.3295	~0.2877
10	Оновлення даних	~1.5108	~1.3905
11	Пошук парковки	~0.8108	~0.8007

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних.» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [45].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ІПБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	4	3
2. Ринкові переваги (наявність аналогів)	4	4	4
3. Ринкові переваги (ціна продукту)	1	1	1
4. Ринкові переваги (технічні властивості)	2	3	2
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	3	4
8. Практична здійсненність (наявність фахівців)	4	3	3
9. Практична здійсненність (наявність фінансів)	3	3	4

Продовження таблиці 4.2

10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	4	3	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	35	34	35
Середньоарифметична сума балів $СБ_c$	34,7		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [45].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів $СБ_c$, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних.» становить 34,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою [45]:

$$q_i = \frac{P_i}{P_{\text{базі}}} \quad (4.1)$$

де q_i – одиничний параметричний індекс, розрахований за i -м параметром;

P_i – значення i -го параметра виробу;

$P_{\text{базі}}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Швидкість авторизації користувача	с	3	1,5	2	0,25
Швидкість запуску системи	с	25	17	1,47	0,15
Швидкість оновлення інформації	мс	30	25	1,2	0,15
Швидкість пошуку	мс	9	7	1,26	0,2
Зрозумілість інтерфейсу	бал	6	8	1,33	0,25
Експлуатаційні витрати на підтримку автоматизованої системи	грн	400	400	1	0,45
Вартість мобільного застосунку для користувача	грн	700	600	0,86	0,55

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [45]:

$$I_{\text{НП}} = \prod_{i=1}^n q_i, \quad (4.2)$$

де $I_{\text{НП}}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i -м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{\text{НП}} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [45]:

$$I_{\text{ТП}} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (4.3)$$

де $I_{\text{ТП}}$ – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i -го параметра;

α_i – вагомість i -го параметричного показника, $\sum_{i=1}^n \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{mn} = 2 \cdot 0,25 + 1,47 \cdot 0,15 + 1,2 \cdot 0,15 + 1,26 \cdot 0,2 + 1,33 \cdot 0,25 = 1,49.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [45]:

$$I_{EП} = \sum_{i=1}^m q_i \cdot \beta_i, \quad (4.4)$$

де $I_{EП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i -го виду;

$$\beta_i – \text{частка } i\text{-го економічного параметра, } \sum_{i=1}^m \beta_i = 1;$$

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці .

$$I_{EП} = 1 \cdot 0,45 + 0,86 \cdot 0,55 = 0,92.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [45]:

$$K_{ИТ} = I_{НП} \cdot \frac{I_{ТП}}{I_{EП}}, \quad (4.5)$$

$$K_{ИТ} = 1 \cdot 1,49 / 0,92 = 1,61.$$

Інтегральний показник конкурентоспроможності $K_{ИТ} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [45]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 12450,00 \cdot 21 / 21 = 12450,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12450,00	592,86	21	12450,00
Інженер-програміст	11630,00	553,81	20	11076,19
Інженер-аналітик АСУ	11630,00	553,81	11	6091,90
Консультант (менеджер логістичної діяльності підприємства)	11500,00	547,62	4	2190,48
Всього				31808,57

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.8)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=2379,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [45];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_l = 2379,00 \cdot 1,10 \cdot 1,65 / (21 \cdot 8) = 25,70 \text{ грн.}$$

$$Z_{pl} = 25,70 \cdot 8,00 = 205,61 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка обчислювального обладнання	8,00	2	1,10	25,70	205,61
Підготовка робочого місця розробника автоматизованої системи управління	10,00	3	1,35	31,54	315,43
Інсталяція програмного забезпечення розробки (моделювання) систем управління	6,00	4	1,50	35,05	210,29
Підготовка бази даних	10,00	3	1,35	31,54	315,43

Продовження таблиці 4.6

Підготовка мобільної підсистеми	3,50	5	1,70	39,72	139,02
Формування інтерфейсного каскаду	4,00	5	1,70	39,72	158,88
Налагодження системи	6,00	6	2,00	46,73	280,38
Інтеграція модулів	1,50	4	1,50	35,05	52,57
Всього					1677,62

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.9)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{дод}} = (31808,57 + 1677,62) \cdot 10 / 100\% = 3348,62 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.10)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (31808,57 + 1677,62 + 3348,62) \cdot 22 / 100\% = 8103,66 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_{\beta j} \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_j \quad , \quad (4.11)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{\beta j}$ – вартість відходів j -го найменування, грн/кг.

$$M_1 = 2,00 \cdot 97,00 \cdot 1,1 - 0,000 \cdot 0,00 = 213,40 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір офісний	97,00	2,00	-	-	213,40
Папір для заміток	42,00	2,00	-	-	92,40
Начиння канцелярське	120,00	4,00	-	-	528,00
Органайзер офісний	100,00	4,00	-	-	440,00
Картридж для принтера	325,00	3,00	-	-	1072,50
Диск оптичний	12,10	2,00	-	-	26,62
FLASH-пам'ять	300,00	1,00	-	-	330,00
Всього					2702,92

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_{ϕ}), які використовують при проведенні НДР на тему «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i \quad (4.12)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{np.i}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10...1,12$);

k – кількість найменувань устаткування.

$$B_{спец} = 16200,00 \cdot 1 \cdot 1,1 = 17820,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Сервер бази даних на основі ПЕОМ Everest 2030	1	16200,00	17820,00
Всього			17820,00

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прог} = \sum_{i=1}^k C_{инрг} \cdot C_{прог.i} \cdot K_i, \quad (4.13)$$

де $C_{инрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прог.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10...1,12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прз}} = 5500,00 \cdot 1 \cdot 1,1 = 6050,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows	1	5500,00	6050,00
Прикладний пакет Microsoft Office	1	5100,00	5610,00
Система імітаційного комп'ютерного моделювання	1	8300,00	9130,00
Програмне забезпечення серверної системи	1	5200,00	5720,00
Всього			26510,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{г}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.14)$$

де $Ц_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{г}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (24800,00 \cdot 1) / (2 \cdot 12) = 1033,33 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер ПЕОМ Everglade E7	24800,00	2	1	1033,33
Обчислювальний комплекс обробки даних	26400,00	2	1	1100,00
Робоче місце розробника (офісні меблі та приладдя)	10300,00	5	1	171,67
Пристрій виводу інформації	6400,00	4	1	133,33
Оргтехніка	10200,00	4	1	212,50
Приміщення	310000,00	25	1	1033,33
Смартфон РОСО Х2	5100,00	3	1	141,67
Всього				3825,83

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.15)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 4,50$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,25 \cdot 160,0 \cdot 4,50 \cdot 0,95 / 0,97 = 180,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер Everglade E7 ПЕОМ	0,25	160,0	180,00
Обчислювальний комплекс обробки даних	0,36	160,0	259,20
Робоче місце розробника (офісні меблі та приладдя)	0,25	92,0	103,50
Пристрій виводу інформації	0,40	10,0	18,00
Оргтехніка	0,56	7,0	17,64
Сервер бази даних на основі ПЕОМ Everest 2030	0,10	92,0	41,40
Всього			619,74

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та

приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.16)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 22\%$.

$$B_{cv} = (31808,57 + 1677,62) \cdot 22 / 100\% = 7366,96 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.17)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 40\%$.

$$B_{cn} = (31808,57 + 1677,62) \cdot 40 / 100\% = 13394,48 \text{ грн.}$$

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_{\text{o}} + Z_{\text{p}}) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (4.18)$$

де $H_{\text{ів}}$ – норма нарахування за статтею «Інші витрати», прийmemo $H_{\text{ів}} = 55\%$.

$$I_{\text{в}} = (31808,57 + 1677,62) \cdot 55 / 100\% = 18417,41 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (Z_{\text{o}} + Z_{\text{p}}) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.19)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{\text{нзв}} = 115\%$.

$$B_{\text{нзв}} = (31808,57 + 1677,62) \cdot 115 / 100\% = 38509,12 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема

управління базою даних» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_n + M + K'_e + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сп}} + I_e + B_{\text{нзв}}. \quad (4.20)$$

$$\begin{aligned} B_{\text{заг}} &= 31808,57 + 1677,62 + 3348,62 + 8103,658283 + 2702,92 + 0,00 + 17820,00 + \\ &26510,00 + 3825,83 + 619,74 + 7366,96 + 13394,48 + 18417,41 + 38509,12 = \\ &= 174104,93 \text{ грн.} \end{aligned}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.21)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,95$.

$$ZB = 174104,93 / 0,95 = 183268,34 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження

діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	400	600	800	1000

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 4300 осіб;

C_o – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 600,00 грн;

$\pm \Delta C_o$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 90,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [Козловський, Лесько, Кавецький]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.22)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 35\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (90,00 \cdot 4300,00 + 690,00 \cdot 400) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 157933,23 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (90,00 \cdot 4300,00 + 690,00 \cdot 1000) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 256552,17 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (90,00 \cdot 4300,00 + 690,00 \cdot 1800) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 388044,09 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (90,00 \cdot 4300,00 + 690,00 \cdot 2800) \cdot 0,83 \cdot 0,35 \cdot (1 - 0,18/100\%) = 552408,99 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (4.23)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 157933,23/(1+0,05)^1 + 256552,17/(1+0,05)^2 + 388044,09/(1+0,05)^3 + \\ &+ 552408,99/(1+0,05)^4 = 150412,60 + 232700,38 + 335207,07 + 454468,24 = \\ &= 1172788,30 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot ЗВ, \quad (4.24)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 183268,34 грн.

$$PV = k_{инв} \cdot ЗВ, = 2 \cdot 183268,34 = 366536,69 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = PPP - PV \quad (4.25)$$

де PPP – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 1172788,30 грн;

PV – теперішня вартість початкових інвестицій, 366536,69 грн.

$$E_{abc} = PPP - PV = 1172788,30 - 366536,69 = 806251,61 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = T_{жс} \sqrt[1 + \frac{E_{abc}}{PV}] - 1, \quad (4.26)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, 806251,61 грн;

PV – теперішня вартість початкових інвестицій, 366536,69 грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_g = T_{жс} \sqrt[1 + \frac{E_{abc}}{PV}] - 1 = (1 + 806251,61/366536,69)^{1/4} - 1 = 0,34.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min}

:

$$\tau_{min} = d + f, \quad (4.27)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,06.

$\tau_{min} = 0,1 + 0,06 = 0,16 < 0,34$ свідчить про те, що внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_e}, \quad (4.28)$$

де E_e – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,34 = 2,96 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Мобільна підсистема і підсистема управління базою даних» становить 34,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,61 рази.

Також термін окупності становить 2,96 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Ч. 1. Мобільна підсистема і підсистема управління базою даних».

ВИСНОВКИ

Магістерська робота присвячена вирішенню однієї з найактуальніших логістичних проблем – проблеми пошуку паркувальних місць. У першій частині розглядаються існуючі рішення та висвітлюються переваги та недоліки кожного рішення. Основним недоліком існуючої системи пошуку кінцевих точок логістичного процесу, тобто системи пошуку кінцевих точок логістичного процесу, є відсутність більш детальної інформації про саме місце розташування, тобто поточний вільний простір чи ні. Користувач буде знати стан стоянки та може спланувати найкращий маршрут. Це також дозволить компаніям контролювати свої місця для паркування.

У другій частині розглядається покращена система пошуку кінцевих точок логістичного процесу, системні функції та режими використання. Розглядається технологія, яка буде використана для створення системи знаходження кінця логістичного процесу. Система створена як веб-сторінка, написана за допомогою фреймворку Angular. Для відображення карти використовується API Карт Google. Системна база даних базується на базі даних Google Firebase реального часу. Система також використовує Firebase Authentication, службу аутентифікації від Google. Показано, як налаштувати правила читання та запису в базі даних і як увійти в систему.

В третій частині оглядається матеріал із використання пошукової системи для пошуку кінця логістичного процесу. Система передбачає два режими: режим користувача та режим підприємства. Перегляньте детальну інформацію про використання системи в кожному режимі. Після будь-якої операції інформація всіх користувачів системи буде оновлюватися. Розглянемо налаштування та контроль авторизації користувачів. Можливість видалення з системи, створення нових і скидання паролів користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. UNIP – паркуйся без проблем [Електронний ресурс] // <https://play.google.com> – 2021. – Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=biz.unip.unip>
2. Parking UA [Електронний ресурс] // <https://play.google.com> – 2021. – Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=com.parkingapp.android>
3. Parkopedia Парковки [Електронний ресурс] // <https://play.google.com> – 2021. – Режим доступу до ресурсу: <https://play.google.com/store/apps/details?id=com.parkopedia&hl=ua>
4. Angular (web framework) [Електронний ресурс] // <https://en.wikipedia.org/> – 2021. – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework)) Ошибка! Недопустимый объект гиперссылки.
5. Карты Google [Електронний ресурс] // <https://en.wikipedia.org/> – 2021. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%82%D1%8B_Google
6. Firebase Realtime Database [Електронний ресурс] // <https://firebase.google.com/> – 2021. – Режим доступу до ресурсу: <https://firebase.google.com/docs/database>
7. Firebase Authentication [Електронний ресурс] // <https://firebase.google.com/> – 2021. – Режим доступу до ресурсу: <https://firebase.google.com/docs/auth>

8. Ionic Framework [Електронний ресурс] // <https://ionicframework.com/> – 2021. – Режим доступу до ресурсу: <https://ionicframework.com/docs/intro>
9. Датчики руху [Електронний ресурс] // <https://en.wikipedia.org/> – 2021. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/D0%94%D0%B0%D1%82%D1%87%D0%B8%D0%BA_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F
10. Інфочервоний датчик руху [Електронний ресурс] // <https://learn.adafruit.com> – 2021. – Режим доступу до ресурсу: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>
11. Радіохвильовий датчик руху [Електронний ресурс] // <https://simplisafe.com> – 2021. – Режим доступу до ресурсу: <https://simplisafe.com/blog/motion-detector-guide>
12. Ahmed N. Abdalla, Muhammad Rauf, Azhar Fakharuddin, Xiao Yao - Public transport monitoring with route and dispatch management system, 2011
13. Dimil Josea , Sanath Prasadb , V. G. Sridhar - Intelligent Vehicle Monitoring Using Global Positioning System and Cloud Computing, 2015
14. What is a REST API? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mulesoft.com/resources/api/what-is-rest-api-design>.
15. Raspberry Pi [Електронний ресурс] – Режим доступу до ресурсу: <https://www.raspberrypi.org/>.
16. Debian [Електронний ресурс] – Режим доступу до ресурсу: <https://www.debian.org/>.
17. NEO - 6 u - blox 6 GPS Modules Data Sheet [Електронний ресурс] – Режим доступу до ресурсу:

https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf.

18. Open Source Computer Vision Library [Електронний ресурс] – Режим доступу до ресурсу: <https://opencv.org/>.

19. What is HTTPS? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.instantssl.com/ssl-certificate-products/https.html>.

20. eWiFi [Електронний ресурс] – Режим доступу до ресурсу: <http://ewifi.com.ua/>.

21. В наземному громадському транспорті з'явився безкоштовний Wi-Fi [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://kpt.kiev.ua/news/details/5435-В-наземному-громадському-транспорті-з-явився-безкоштовний-Wi-Fi-/>

22. Кодекс законів про працю України [Електронний ресурс] – Режим доступу до ресурсу: <http://zakon.rada.gov.ua/laws/show/322-08>.

23. Parks M. The History of Public Bus Transportation [Електронний ресурс] / Madeline Parks. – 2014. – Режим доступу до ресурсу: <https://gogocharters.com/blog/history-of-public-bus-transportation/>

24. Inauguration of the first electric streetcar in the world [Електронний ресурс] – Режим доступу до ресурсу: https://www.siemens.com/history/en/news/1075_electrical-streetcar.html

25. Thompson L. Public Transportation in the U. S.: History and Current Status / Louis S Thompson. – 2008.

26. O'Toole R. Urban Transit [Електронний ресурс] / Randal O'Toole. – 2017. – Режим доступу до ресурсу: <https://www.downsizinggovernment.org/transportation/urban-transit>

27. Simpson B. URBAN PUBLIC TRANSPORT TODAY / Barry Simpson., 1994.

28. Iles R. Public Transport in Developing Countries / Richard Iles.. – 2005 с.
29. Vilalta C. Fear of crime in public transport: Research in Mexico City / Carlos Vilalta. // *Crime Prevention and Community Safety*. – 2011. – №13. – С. 171–186.
30. Горбунов А. Особенности раскрытия и расследования карманных краж, совершаемых в общественном транспорте / А. Горбунов, К. Калюжный. // *Общество и право*. – 2014.
31. *Mobilities: New Perspectives on Transport and Society*, 2016.
32. Horan E. Sustainable Urban Development and Liveability. How can Melbourne Retain its title as the World's Most Liveable City and Strive for Sustainability at the Same Time? / E. Horan, J. Craven. // *European Journal of Sustainable Development*. – 2014. – №3.
33. Exploring Bus Rapid Transit passenger travel behaviour using big data / S.Tao, J. Corcoran, I. Mateo-Babiano, D. Rohde. // *Applied Geography*. – №53. – С. 90–104.
34. Using Big Data Analytics for Improved Public Transport. // *UN Global Pulse*. – 2017. – №25.
35. Batty M. Big data, smart cities and city planning / Michael Batty. // *Dialogues in Human Geography*. – 2013.
36. Daduna J. Computer-Aided Transit Scheduling / Joachim Daduna., 2012.
37. Chicca F. Everyday Lifestyles and Sustainability / F. Chicca, B. Vale, R. Vale.. – 238 с.
38. Delbosc A. Modelling the causes and impacts of personal safety perceptions on public transport ridership / Alexa Delbosc. // *Transport Policy*. – №24. – С. 302–309.

39. Управление транспортом на основе математического моделирования / П.Устич, А. Иванова, В. Мышков, В. Садчиков. // Российские железные дороги. – 2008. – №7. – С. 39–43.
40. Иванов В. Автоматизированное управление транспортом / В. Иванов., 1996.
41. Когнитивное управление транспортом. // Государственный советник. – 2015. – №2. – С. 43–52.
42. Mallig N. mobiTopp – A Modular Agent-based Travel Demand Modelling Framework / Nicolai Mallig. // Procedia Computer Science. – 2013. – №19. – С. 854–859.
43. Powerful, Scaled Charging For Your EV Bus Fleet [Электронный ресурс] – Режим доступа до ресурсу: <https://www.proterra.com/technology/chargers/>.
44. Mokhtarian B. The Broader Connection between Public Transportation, Energy Conservation, and Greenhouse Gas Reduction / B. Mokhtarian, L. Mokhtarian, P. Mokhtarian. – 2007.
45. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
46. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А

ЗАТВЕРДЖУЮ

Завідувач кафедру КСУ

д.т.н., проф. Володимир Дубовой

«___»_____ 202_ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«Розробка серверного застосунку з мобільною складовою для логістичного

супроводження діяльності підприємства. Ч. 1. Мобільна підсистема і

підсистема управління базою даних»

08-01.МКР.001.00.000 ТЗ

Керівник: к.т.н., доцент Марія Юхимчук

«_____» _____ 2021 р.

Виконав: студент 2 курсу, групи 2АКІТ-20м

Артем Кирилюк

«_____» _____ 2021 р.

Вінниця 2021

1. Назва та галузь застосування

Розробка серверного застосунку з мобільною складовою для логістичного супроводження діяльності підприємства. Ч. 1. Мобільна підсистема і підсистема управління базою даних.

Розроблена архітектура, модель і система для підвищення ефективності логістичного супроводження діяльності підприємства.

Розробка призначена для використання в галузях інформаційних технологій.

2. Підстави для розробки

Розробку системи здійснювати на підставі наказу по університету № _____ від _____ та завдання до магістерської кваліфікаційної роботи складеного та затвердженого кафедрою КСУ.

3. Мета та призначення розробки

Метою дипломної дисертації є підвищення ефективності логістичного супроводження діяльності підприємства. Для досягнення поставленої мети необхідно розробити архітектуру, модель і саму відповідну систему.

4. Джерела розробки

1. CO-ResNet: Optimized ResNet model for COVID-19 diagnosis from X-ray images / S. Bharati et al. International Journal of Hybrid Intelligent Systems. 2021. Vol. 17. P. 71–85.
2. Nibali A., He Z., Wollersheim D. Pulmonary Nodule Classification with Deep Residual Networks // International Journal of Computer Assisted Radiology and Surgery. 2017. Vol. 12. P. 1799–1808.
3. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent // Proceedings of COMPSTAT2010. 2010. P. 177–186.
4. He S., Wu Q.H., Saunders J.R. A Group Search Optimizer for Neural Network Training // Computational Science and Its Applications - ICCSA 2006 Lecture Notes in Computer Science. 2006. P. 934–943.

5. Rozložník M. Solution Approaches for Saddle-Point Problems // Nečas Center Series Saddle-Point Problems and Their Iterative Solution. 2018. P. 33–39.

5. Показники призначення

Вихідними даними для розробки є результати аналізу об'єкта дослідження.

Результатом роботи методу є ефективності логістичного супроводження діяльності підприємства.

6. Стадії розробки

1. Розділ 1 «Огляд предметної області» має бути виконаний до «04» Вересня 2021 р.

2. Розділ 2 «Розробка математичного апарату» має бути виконана до «22» Вересня 2021 р.

3. Розділ 3 «Розробка програмного забезпечення та експериментальні дослідження» до «3» Листопада 2021 р.

7. Порядок контролю та приймання

1. Рубіжний контроль. Провести до «15» Грудня 2021 р.

2. Попередній захист магістерської кваліфікаційної роботи.
Провести до «16» Грудня 2021 р.

3. Захист магістерської кваліфікаційної роботи.
Провести «22» Грудня 2021 р.

Додаток Б

Фрагменти програмного коду

MapComponent.ts

```
import { Component, OnInit, ViewChild, OnDestroy } from "@angular/core";
import { MatDialogConfig, MatDialog } from "@angular/material";
import { ConfirmComponent } from "../confirm/confirm.component";
import { Router } from "@angular/router";
import { AuthService } from "../auth.service";
import { AngularFireStore } from "@angular/fire/firestore";
import { Observable, Subject, from } from "rxjs";
import { DatabaseService } from "../database.service";
import { Item } from "../item";
import { SubSink } from "subsink";
import { IPlace, IParking, IRate } from "../interfaces/parking.interface";
```

```
@Component({
  selector: "app-map",
  templateUrl: "../map.component.html",
  styleUrls: ["../map.component.scss"]
})
export class MapComponent implements OnInit, OnDestroy {
  companyName;
  places: IPlace[] = [];
  parkings: IParking[] = [];
  rates: IRate[] = [];
  idLastParking: string;
  markerZoom = 17;
  isClearInterval = false;
  parkingsAll: IParking[] = [];
  subs = new SubSink();
  icons = {
    free: 'http://maps.google.com/mapfiles/kml/paddle/grn-circle.png',
    busy: 'http://maps.google.com/mapfiles/kml/paddle/red-circle.png',
    repair: 'http://maps.google.com/mapfiles/kml/paddle/orange-circle.png'
  };
  modes = {
    free: 'Парковка вільна',
    busy: 'Парковка зайнята',
    repair: 'Ведуться технічні роботи'
  };
};
```

```
@ViewChild("gmap") gmapElement: any;
```

```
map: google.maps.Map;
options;
markers = [];
mainMarker;
timerId;
isLoadingData = false;
isUser = false;
```

```
constructor(
  public dialog: MatDialog,
  private router: Router,
  private authService: AuthService,
  private dbService: DatabaseService
```

```

) {}

ngOnInit() {
  this.timerId = setInterval(() => {
    if (this.authService.uid) {
      this.startSetting();
    }
  }, 2000);
}

startSetting() {
  clearInterval(this.timerId);

  this.subs.add(
    this.dbService
      .getItem("/users/" + this.authService.uid + "/type")
      .subscribe(action => {
        this.isUser = action.payload.val() === "user";
        this.getCompanyName();
        this.getAllParkings();
        if (!this.isUser) {
          this.getRates();
          this.getPlaces();
          this.getParkings();
          this.getParkingLastId();
        }
      })
  );

  this.isLoadingData = true;

  this.options = {
    center: { lat: 36.890257, lng: 30.707417 },
    zoom: 12
  };
  this.map = new google.maps.Map(
    this.gmapElement.nativeElement,
    this.options
  );

  navigator.geolocation.getCurrentPosition(
    position => {
      const pos = {
        lat: position.coords.latitude,
        lng: position.coords.longitude
      };

      this.map.setCenter(pos);
    },
    error => {
      console.log(error);
    }
  );

  if (!this.isUser) {
    this.map.addListener("click", e => {
      if (this.map.getZoom() < this.markerZoom) {
        this.map.setZoom(this.markerZoom);
        this.map.setCenter({
          lat: e.latLng.lat(),

```

```

lng: e.latLng.lng()
});
} else {
this.addMarker(e, this.map);
}
});
}
}

getAllParkings() {
this.subs.add(
this.dbService.getItem("/parkings").subscribe((action: any) => {
const parkings = action.payload.val();
if (parkings) {
this.parkingsAll = parkings.filter(parking => parking);
if (this.isUser) {
this.setMarkersUsers();
}
}
})
);
}

getCompanyName() {
this.subs.add(
this.dbService
.getItem("/users/" + this.authService.uid + "/name")
.subscribe(action => {
this.companyName = action.payload.val();
})
);
}

getRates() {
this.subs.add(
this.dbService
.getItem("/users/" + this.authService.uid + "/rates")
.subscribe((action: any) => {
const rates = action.payload.val();
if (rates) {
this.rates = rates.filter(rate => rate);
}
})
);
}

getPlaces() {
this.subs.add(
this.dbService
.getItem("/users/" + this.authService.uid + "/places")
.subscribe((action: any) => {
const places = action.payload.val();
if (places) {
this.places = places.filter(place => place);
}
})
);
}

getParkings() {

```



```

this.subs.add(
this.dbService
.getItem("/users/" + this.authService.uid + "/parkings")
.subscribe((action: any) => {
const parkings = action.payload.val();
if (parkings) {
this.parkings = parkings.filter(parking => parking);
this.setMarkers();
}
})
);
}

getParkingLastId() {
this.subs.add(
this.dbService.getItem("/parkingLastId").subscribe((action: any) => {
const lastId: string = action.payload.val();
if (lastId) {
let id = parseInt(lastId.substr(1), 10);
id += 1;
const idToString = id.toString();
const countSymbol = idToString.length;
this.idLastParking = "#";
for (let index = 0; index < 10 - idToString.length; index++) {
this.idLastParking += "0";
}
this.idLastParking += idToString;
console.log(this.idLastParking);
}
})
);
}

removePlace(event) {
const removePlace: IPlace = event.value;
const findIndex = this.places.findIndex(
place => place.name === removePlace.name
);
this.places.splice(findIndex, 1);
// this.dbService.updateItem("/users/" + this.authService.uid, {
// places: this.places
// });
}

removeRate(event) {
const removeRate: IPlace = event.value;
const findIndex = this.rates.findIndex(
rate => rate.name === removeRate.name
);
this.rates.splice(findIndex, 1);
// this.dbService.updateItem("/users/" + this.authService.uid, {
// rates: this.rates
// });
}

editPlace(event) {
const dialogConfig = new MatDialogConfig();
dialogConfig.data = {
isEditPlace: true
};
};

```

```

const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

dialogRef.afterClosed().subscribe((result: IParking) => {
  console.log(result);
});
}

editRate(event) {
  const dialogConfig = new MatDialogConfig();
  dialogConfig.data = {
    isEditRate: true
  };

  const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

  dialogRef.afterClosed().subscribe((result: IParking) => {
    console.log(result);
  });
}

editParking(parking) {
  console.log(parking);
  const dialogConfig = new MatDialogConfig();
  dialogConfig.data = {
    rates: this.rates,
    places: this.places,
    parking: parking,
    isEditParking: true
  };

  const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

  dialogRef.afterClosed().subscribe((parking: IParking) => {
    if (parking) {
      console.log(parking);
      let findIndex = this.parkings.findIndex(p => p.id === parking.id);
      this.parkings[findIndex] = parking;
      findIndex = this.parkingsAll.findIndex(p => p.id === parking.id);
      this.parkingsAll[findIndex] = parking;
      this.updateParkingsDB();
    }
  });
}

removeParking(parking) {
  let findIndex = this.parkings.findIndex(p => p.id === parking.id);
  this.parkings.splice(findIndex, 1);
  findIndex = this.parkingsAll.findIndex(p => p.id === parking.id);
  this.parkingsAll.splice(findIndex, 1);
  this.updateParkingsDB();
}

setMarkersUsers() {
  this.markers.forEach(marker => {
    marker.setMap(null);
  });

  this.parkingsAll.forEach((parking: IParking, index) => {
    console.log(parking);
  });
}

```

```

const place = parking.place ? parking.place : "Без місця";
const contentString =
`<div id="content">` +
`<h3>${parking.name}</h3>` +
`<p>Місце: ${place}</p>` +
`<p>Тариф: ${parking.rate} (${parking.price} грн/год)</p>` +
`<p>${this.modes[parking.mode]}</p>` +
`<button ${
!parking.access ? "disabled" : ""
} id="info-window${index}" class="btn">` +
`<span class="btn-text">Припаркуватися</span>` +
`</button>`;
"</div>";
const infoWindow = new google.maps.InfoWindow();
const marker = new google.maps.Marker({
position: parking.position,
icon: this.icons[parking.mode]
});
google.maps.event.addListener(infoWindow, "domready", () => {
const infowindow = document.getElementById("info-window" + index);
infowindow.addEventListener("click", () => {
this.editParking(parking);
});
});
marker.setMap(this.map);
this.markers.push(marker);
marker.addListener("click", () => {
infoWindow.close();
infoWindow.setContent(contentString);
infoWindow.open(this.map, marker);
infoWindow.setZIndex(10);
});
});
}

setMarkers() {
this.markers.forEach(marker => {
marker.setMap(null);
});
}

this.parkings.forEach((parking: IParking, index) => {
const place = parking.place ? parking.place : "Без місця";
const contentString =
`<div id="content">` +
`<h3>${parking.name} <span>(id: ${parking.id})</span></h3>` +
`<p>Місце: ${place}</p>` +
`<p>Тариф: ${parking.rate} (${parking.price} грн/год)</p>` +
`<button id="info-window-edit${index}" class="btn">` +
`<span class="btn-text">Редагувати парковку</span>` +
`</button>` +
`<button id="info-window-remove${index}" class="btn">` +
`<span class="btn-text">Видалити</span>` +
`</button>`;
"</div>";
const infoWindow = new google.maps.InfoWindow();
const marker = new google.maps.Marker({
position: parking.position,
icon: this.icons[parking.mode]
});
});
google.maps.event.addListener(infoWindow, "domready", () => {

```

```

const infowindow = document.getElementById("info-window-edit" + index);
infowindow.addEventListener("click", () => {
  this.editParking(parking);
});
});
google.maps.event.addListener(infoWindow, "domready", () => {
const infowindow = document.getElementById(
"info-window-remove" + index
);
infowindow.addEventListener("click", () => {
  this.removeParking(parking);
});
});
marker.setMap(this.map);
this.markers.push(marker);
marker.addListener("click", () => {
  infoWindow.close();
  infoWindow.setContent(contentString);
  infoWindow.open(this.map, marker);
  infoWindow.setZIndex(10);
});
});
}

```

```

editCompanyName() {
const dialogConfig = new MatDialogConfig();
dialogConfig.data = {
  companyName: this.companyName
};

```

```

const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

```

```

dialogRef.afterClosed().subscribe((companyName: boolean) => {
  if (companyName) {
    this.dbService.updateItem("/users/" + this.authService.uid, {
      name: companyName
    });
  }
});
}

```

```

addRate() {
const dialogConfig = new MatDialogConfig();
dialogConfig.data = {
  isRate: true
};

```

```

const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

```

```

dialogRef.afterClosed().subscribe((rate: { price; name }) => {
  console.log(rate);
  if (rate) {
    this.rates.push(rate);
    this.dbService.updateItem("/users/" + this.authService.uid, {
      rates: this.rates
    });
  }
});
}

```

```

addPlace() {
const dialogConfig = new MatDialogConfig();
dialogConfig.data = {
isPlace: true,
rates: this.rates,
parkings: this.parkings
};

const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

dialogRef
.afterClosed()
.subscribe((place: { name: string; rate: string; parking: any }) => {
if (place) {
console.log(place);
const findRate = this.rates.find(
(rate: IRate) => rate.name === place.rate
);
this.updateParkings(place, findRate);

this.places.push({ name: place.name, rate: place.rate });
this.dbService.updateItem("/users/" + this.authService.uid, {
places: this.places
});
}
});
}

addMarker(e, map) {
const pos = {
lat: e.latLng.lat(),
lng: e.latLng.lng()
};

const marker = new google.maps.Marker({
position: pos
// title: "Hello World!"
});

marker.setMap(map);
map.setCenter(pos);
map.setOptions({ zoom: this.markerZoom });

const dialogConfig = new MatDialogConfig();
dialogConfig.data = {
rates: this.rates,
places: this.places,
position: pos
};

const dialogRef = this.dialog.open(ConfirmComponent, dialogConfig);

dialogRef
.afterClosed()
.subscribe((parking: { rate: string; name: string; place: string }) => {
if (parking) {
this.markers.push(marker);
this.setParking(parking, pos);
console.log(parking);
} else {

```

```

marker.setMap(null);
}
});
}

updateParkings(
place: { name: string; rate: string; parking: object },
rate: IRate
) {
console.log(place);
for (const parkingId in place.parking) {
console.log(parkingId);
if (place.parking.hasOwnProperty(parkingId) && place.parking[parkingId]) {
console.log(parkingId);
const findParking = this.parkings.find(
parking => parking.id === parkingId
);
const findParkingAll = this.parkingsAll.find(
parking => parking.id === parkingId
);
console.log(findParking);
console.log(findParkingAll);

findParking.place = place.name;
findParkingAll.place = place.name;

if (rate) {
findParking.rate = rate.name;
findParkingAll.rate = rate.name;
findParking.price = rate.price;
findParkingAll.price = rate.price;
}
}
}
this.updateParkingsDB();
}

exit() {
this.authService.exit();
}

updateParkingsDB() {
this.dbService.updateItem("/users/" + this.authService.uid, {
parkings: this.parkings
});
this.dbService.updateItem("/", { parkings: this.parkingsAll });
}

setParking(parking: { rate: string; name: string; place: string }, position) {
const newParking: IParking = {
...parking,
id: this.idLastParking,
position
};

if (parking.rate) {
const findRate = this.rates.find(
(rate: IRate) => rate.name === parking.rate
);
newParking.rate = findRate ? findRate.name : "Free";
}
}

```

```

newParking.price = findRate ? findRate.price : 0;
} else if (parking.place) {
const findPlace = this.places.find(
(place: IPlace) => place.name === parking.place
);
const findRate = findPlace
? this.rates.find((rate: IRate) => rate.name === findPlace.rate)
: null;
newParking.rate = findRate ? findRate.name : "Free";
newParking.price = findRate ? findRate.price : 0;
} else {
newParking.rate = "Free";
newParking.price = 0;
}
this.parkings.push(newParking);
this.setGlobalParkings(newParking);
this.dbService.updateItem("/users/" + this.authService.uid, {
parkings: this.parkings
});
this.dbService.updateItem("/", { parkingLastId: this.idLastParking });
}

setGlobalParkings(parking: IParking) {
const newParking = {
...parking,
access: true
};
this.parkingsAll.push(newParking);
this.dbService.updateItem("/", { parkings: this.parkingsAll });
}

ngOnDestroy(): void {
this.subs.unsubscribe();
}
}

```

MapComponent.html

```

<div class="container">
  <div class="menu" *ngIf="isLoadingData && !isGuest">
    <mat-icon class="exit-icon" (click)="exit()">exit_to_app</mat-icon>
    <p class="company-name">
      <span class="name-text">{{ companyName }}</span>
    </p>
    <button (click)="editCompanyName()" class="field-menu" mat-stroked-button>
      <span class="btn-text">
        {{ isUser ? "Редагувати ім'я" : "Редагувати назву компанії" }}</span>
      </button>

    <div class="field-menu" *ngIf="!isUser">
      <button (click)="addPlace()" class="btn" mat-stroked-button>
        <span class="btn-text">Додати місце</span>
      </button>
      <button (click)="addRate()" class="btn" mat-stroked-button>
        <span class="btn-text">Додати тариф</span>
      </button>
    </div>

```

```

<mat-form-field class="field-menu" *ngIf="!isUser">
  <mat-label>Редагувати місце</mat-label>
  <mat-select (selectionChange)="editPlace($event)">
    <mat-option *ngFor="let place of places" [value]="place">
      {{ place.name }}
    </mat-option>
  </mat-select>
</mat-form-field>

```

```

<mat-form-field class="field-menu" *ngIf="!isUser">
  <mat-label>Редагувати тариф</mat-label>
  <mat-select (selectionChange)="editRate($event)">
    <mat-option *ngFor="let rate of rates" [value]="rate">
      {{ rate.name }} ({{ rate.price }})грн/год
    </mat-option>
  </mat-select>
</mat-form-field>

```

```

<mat-form-field class="field-menu" *ngIf="isUser">
  <mat-label>Пошук парковок</mat-label>
  <input matInput placeholder="Введіть адресу">
</mat-form-field>
</div>

```

```

<div class="map">
  <div #gmap id="gmap" style="height: 100%;"></div>
</div>
</div>

```

Database.sevice.ts

```

import { Injectable } from "@angular/core";
import { AngularFireDatabase, QueryFn } from '@angular/fire/database';
import { Item } from './item';
import { Observable } from 'rxjs';
import { AuthService } from './auth.service';

```

```
@Injectable()
```

```
export class DatabaseService {
```

```
  item: Observable<any>; // single object
```

```
  constructor(private db: AngularFireDatabase, auth: AuthService) {
```

```
    // this.item = db.object('item').valueChanges();
```

```
    // this.item.subscribe(res => {
```

```
      // console.log(res);
```

```
    // });
```

```
  }
```

```
  getItemsList(query: QueryFn) {
```

```
    // this.item = db.object('item').valueChanges();
```

```
    // return this.items;
```

```
  }
```

```
  // Return a single observable item
```

```
  getItem(path: string) {
```

```
    return this.db.object(path).snapshotChanges();
```

```
  }
```

```
  createItem(path, item): void {
```

```
    // this.items.push(item).catch(error => this.handleError(error));
```



```

    this.db.object(path).set(item).catch(error => this.handleError(error));
  }

  // Update an existing item
  updateItem(path: string, value: any): void {
    this.db.object(path).update(value).catch(error => this.handleError(error));
  }

  // Deletes a single item
  deleteItem(path: string): void {
    this.db.object(path).remove().catch(error => this.handleError(error));
  }

  // Default error handling for all actions
  private handleError(error) {
    console.log(error);
  }
}

```

Login.ts

```

import { Component, OnInit } from '@angular/core';
import { FormControl, Validators, AbstractControl, ValidatorFn } from '@angular/forms';
import { AngularFireStore } from '@angular/fire/firestore';
import { AngularFireAuth } from '@angular/fire/auth';
import { AuthService } from 'src/app/auth.service';
import { Observable } from 'rxjs';
import { Router } from '@angular/router';
import { DatabaseService } from '../database.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
export class LoginComponent implements OnInit {

  nameFormControl: FormControl;
  emailFormControl: FormControl;
  passwordFormControl: FormControl;
  passwordAgainFormControl: FormControl;

  hide = true;
  hideAgainPass = true;
  isSignUp = false;
  isError = false;
  textError = "";
  isUserMode = true;

  constructor(
    private authService: AuthService,
    private router: Router,
    private db: DatabaseService
  ) {}

  ngOnInit() {

    this.nameFormControl = new FormControl("", [
      Validators.required
    ]);
  }

```

```

this.emailFormControl = new FormControl("", [
  Validators.required,
  Validators.email
]);
this.passwordFormControl = new FormControl("", [
  Validators.required,
  Validators.minLength(6),
]);
this.passwordAgainFormControl = new FormControl("", [
  Validators.required,
  Validators.minLength(6),
], );
}

getErrorMessage(control) {
  switch (control) {
    case 'email':
      return this.emailFormControl.hasError('required') ? 'Заповніть поле' :
        this.emailFormControl.hasError('email') ? 'E-mail не валідний' :
        "";
    case 'pass':
      return this.passwordFormControl.hasError('required') ? 'Заповніть поле' :
        this.passwordFormControl.hasError('minlength') ? 'Пароль повинен бути більше 6 символів' :
        "";
    case 'passagain':
      return this.passwordAgainFormControl.hasError('required') ? 'Заповніть поле' :
        this.passwordAgainFormControl.hasError('minlength') ? 'Пароль повинен бути більше 6 символів' :
        "";
    case 'name':
      return this.nameFormControl.hasError('required') ? 'Заповніть поле' : "";
    default:
      break;
  }
}

clickSignUp() {
  this.isSignUp = true;
  this.isError = false;
  this.textError = "";
  this.emailFormControl.setValue("");
  this.emailFormControl.markAsUntouched();
  this.passwordFormControl.setValue("");
  this.passwordFormControl.markAsUntouched();
}

clickSignIn() {
  this.authService.signIn(this.emailFormControl.value, this.passwordFormControl.value);
  this.authService.succesAuth.subscribe(result => {
    this.router.navigateByUrl('/map');
  });
  this.authService.failAuth.subscribe(result => {
    this.isError = true;
    this.textError = 'E-mail або пароль не вірні';
  });
}

clickGuestIn() {
  this.authService.guestIn();
  this.router.navigateByUrl('/map');
}

```

```

clickCreate() {
  this.authService.create(this.emailFormControl.value, this.passwordFormControl.value,
this.nameFormControl.value);
  this.authService.succesAuth.subscribe(result => {
    let value = {}
    if (this.isUserMode) {
      value = {
        name: this.nameFormControl.value,
        currentParkingID: "",
        parkings: [],
        type: 'user'
      }
    } else {
      value = {
        name: this.nameFormControl.value,
        parkings: [],
        places: [],
        rates: [],
        type: 'company'
      }
    }
  });

  this.db.createItem('/users/' + this.authService.uid, value);
  this.router.navigateByUrl('/map');

});
this.authService.failAuth.subscribe((error: any) => {
  this.isError = true;
  this.textError = error.message;
});
}

changeMode(userMode) {
  this.isUserMode = userMode;
}

}

```

Login.html

```

<div class="container" *ngIf="emailFormControl">
<div class="container signin-view" *ngIf="!isSignUp; else createFields">
  <mat-form-field>
    <input
      matInput
      placeholder="Введіть e-mail"
      [formControl]="emailFormControl"
    />
    <mat-error *ngIf="emailFormControl.invalid">{{
      getErrorMessage("email")
    }}</mat-error>
  </mat-form-field>

  <mat-form-field>
    <input
      matInput
      placeholder="Введіть пароль"
      [formControl]="passwordFormControl"
      [type]="hide ? 'password' : 'text'"
    />
  </mat-form-field>

```

```

<mat-error *ngIf="passwordFormControl.invalid">{{
  getErrorMessage("pass")
}}</mat-error>
<button
  mat-icon-button
  matSuffix
  (click)="hide = !hide"
  [attr.aria-label]="Hide password"
  [attr.aria-pressed]="hide"
>
  <mat-icon>{{ hide ? "visibility_off" : "visibility" }}</mat-icon>
</button>
</mat-form-field>
</div>

<ng-template #createFields>
<div class="container signup-view">
<div class="login-btn">
<button
  [disabled]="isUserMode"
  mat-button
  class="btn mode-btn"
  (click)="changeMode(true)"
>
  Режим користувача
</button>
<button
  [disabled]="!isUserMode"
  mat-button
  class="btn mode-btn"
  (click)="changeMode(false)"
>
  Режим підприємства
</button>
</div>
<mat-form-field>
<input
  matInput
  placeholder="{{ isUserMode ? 'Введіть ваше ім'я' : 'Введіть назву вашої компанії'}}"
  [formControl]="nameFormControl"
/>
<mat-error *ngIf="emailFormControl.invalid">{{
  getErrorMessage("name")
}}</mat-error>
</mat-form-field>

<mat-form-field>
<input
  matInput
  placeholder="Введіть e-mail"
  [formControl]="emailFormControl"
/>
<mat-error *ngIf="emailFormControl.invalid">{{
  getErrorMessage("email")
}}</mat-error>
</mat-form-field>

<mat-form-field>
<input
  matInput

```

```

placeholder="Введіть пароль"
[formControl]="passwordFormControl"
[type]="hide ? 'password' : 'text'"
/>
<mat-error *ngIf="passwordFormControl.invalid">{{
  getErrorMessage("pass")
}}</mat-error>
<button
  mat-icon-button
  matSuffix
  (click)="hide = !hide"
  [attr.aria-label]="Hide password"
  [attr.aria-pressed]="hide"
>
  <mat-icon>{{ hide ? "visibility_off" : "visibility" }}</mat-icon>
</button>
</mat-form-field>

<mat-form-field>
  <input
    matInput
    placeholder="Введіть пароль ще раз"
    [formControl]="passwordAgainFormControl"
    [type]="hideAgainPass ? 'password' : 'text'"
  />
  <mat-error *ngIf="passwordAgainFormControl.invalid">{{
    getErrorMessage("passagain")
  }}</mat-error>
  <button
    mat-icon-button
    matSuffix
    (click)="hideAgainPass = !hideAgainPass"
    [attr.aria-label]="Hide password"
    [attr.aria-pressed]="hide"
  >
    <mat-icon>{{
      hideAgainPass ? "visibility_off" : "visibility"
    }}</mat-icon>
  </button>
</mat-form-field>
</div>
<mat-error
  class="error"
  *ngIf="
    passwordAgainFormControl.valid &&
    passwordAgainFormControl.value !== passwordFormControl.value
  "
>
  Паролі не співпадають
</mat-error>
</ng-template>
<mat-error class="error" *ngIf="isError">
  {{ textError }}
</mat-error>

<div class="login-btn">
  <button
    *ngIf="!isSignUp; else createBtn"
    [disabled]="passwordFormControl.invalid || emailFormControl.invalid"
    mat-button

```

```
class="btn"
(click)="clickSignIn()"
>
  Увійти
</button>
<ng-template #createBtn>
  <button
  [disabled]="
    passwordFormControl.invalid ||
    emailFormControl.invalid ||
    passwordAgainFormControl.value !== passwordFormControl.value
  "
  mat-button
  class="btn"
  (click)="clickCreate()"
  >
    Створити
  </button>
</ng-template>

<button *ngIf="!isSignUp" mat-button class="btn" (click)="clickSignUp()">
  Зареєструватися
</button>
</div>
</div>
```

Додаток В
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

**«Розробка серверного застосунку з мобільною складовою для
логістичного супроводження діяльності підприємства. Ч. 1. Мобільна
підсистема і підсистема управління базою даних»**

Перелік ілюстративних матеріалів:

1. Можливості паркування в мобільному додатку UNIP
2. Інформація про вибране паркувальне місце або паркувальне місце на карті в мобільному додатку parking UA
3. Коротка інформація про паркування в мобільному додатку «Parkopedia Parking»
4. Порівняння систем пошуку кінцевих точок логістичного процесу
5. Модель MVC
6. Блок-схема автентифікації Firebase
7. Шаблон підтвердження електронної пошти для автентифікації Firebase
8. Блок-схема роботи додатку в корпоративному режимі
9. Головне меню в корпоративному режимі
10. Інформація про паркування
11. Алгоритм для користувацького режиму
12. Головне меню в режимі юзера
13. Фрагменти даних з бази даних
14. Тестування можливостей системи

Виконав: студент 2-го курсу, групи 2АКІТ-20м
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

_____ **Кирилюк Артем** _____

(ім'я та прізвище)

Керівник: к.т.н., доцент каф. КСУ

_____ **Марія Юхимчук** _____

(ім'я та прізвище)

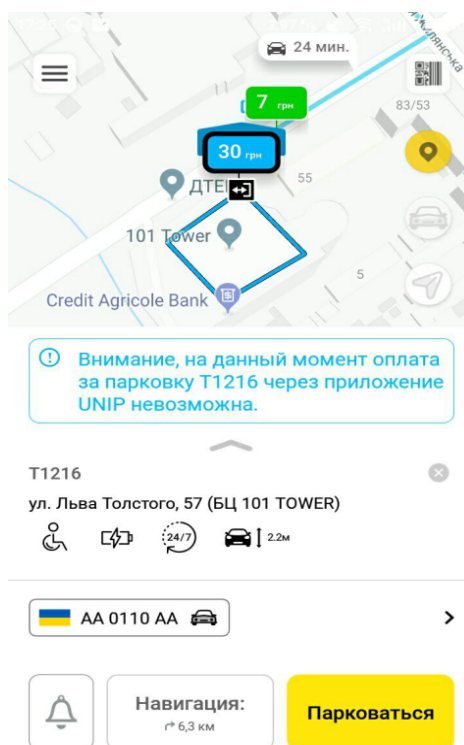
« ____ » _____ 2021 р.

Опонент: к.т.н., доцент АІТ

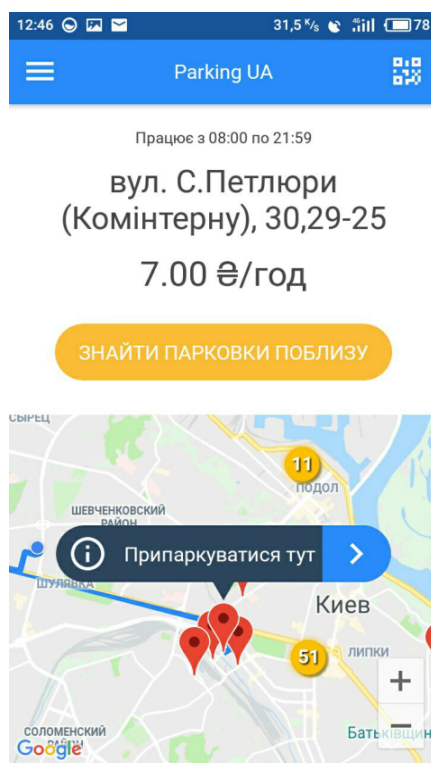
_____ **Юрій Іванов** _____

(ім'я та прізвище)

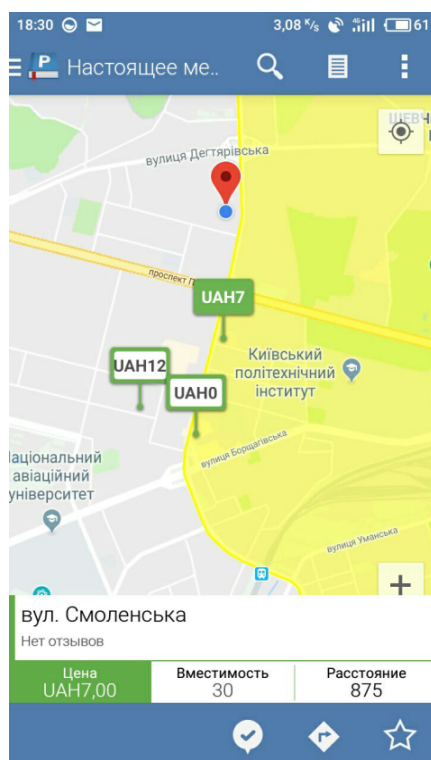
« ____ » _____ 2021 р.



Можливості паркування в мобільному додатку UNIP



Інформація про вибране паркувальне місце або паркувальне місце на карті в мобільному додатку parking UA



Коротка інформація про паркування в мобільному додатку «Parkopedia Parking»

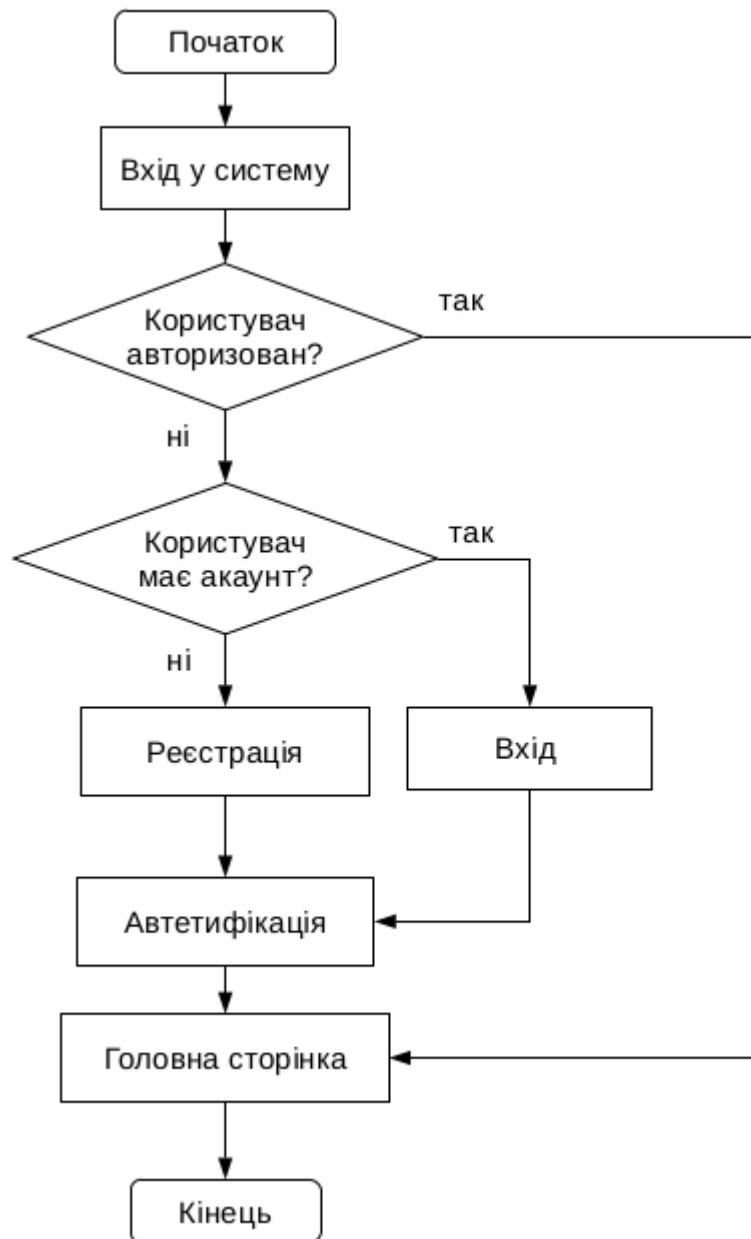
Порівняння систем пошуку кінцевих точок логістичного процесу

Функціонал	“UNIP”	“Parking UA”	“Parkopedia Парковки”
Обов’язкова реєстрація	+	+	-
Реєстрування автомобіля	+	+	-
Інформативні маркери на мапі	+	-	+
Зручна мапа для користувача	+	-	+
Пошук парковок	+	+	+
Навігація до парковки	+	+	+
Детальна інформація про парковку	+	-	+
Фільтри для пошуку	-	-	+
Оплата кредитною <u>картою</u> через додаток	+	+	-
Додавати парковки до обраних	-	+	+













Порівняння систем пошуку кінцевих точок логістичного процесу










Модель MVC



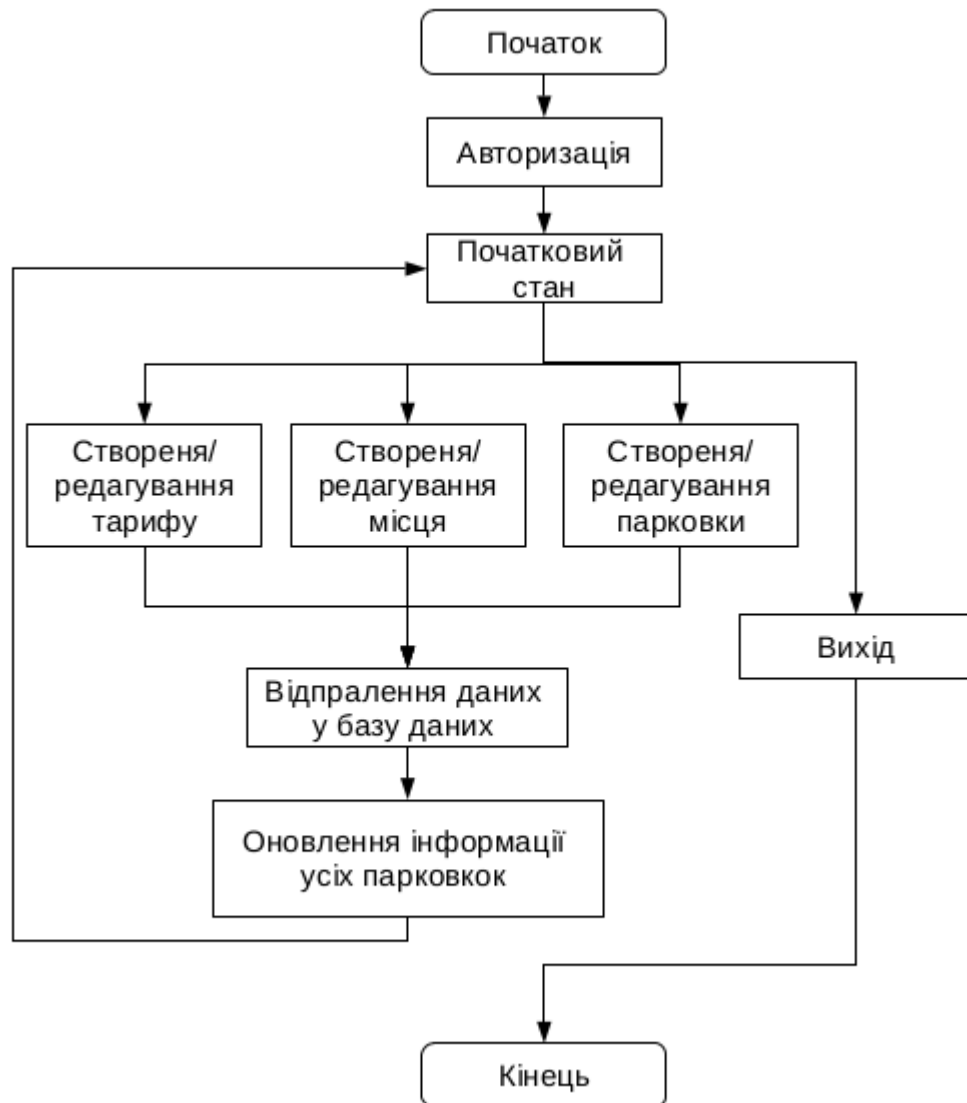
Блок-схема автентифікації Firebase

Провайдер	Состояние
 Адрес электронной почты и пароль	Включен
 Телефон	Отключено
 Google	Отключено
 Play Игры	Отключено
 Game Center Beta	Отключено
 Facebook	Отключено
 Twitter	Отключено
 GitHub	Отключено
 Yahoo	Отключено
 Microsoft	Отключено
 Apple Beta	Отключено
 Анонимный вход	Отключено

Параметри авторизації автентифікації Firebase

<p>Уведомлять по электронной почте</p>	<p>Подтверждение адреса электронной почты</p> <p>Запрос подтверждения, который будет высылаться всем новым пользователям, зарегистрировавшимся по адресу электронной почты. Подробнее...</p>
<ul style="list-style-type: none">  Подтверждение адреса элек...  Изменение пароля  Изменение адреса электрон...  Настройки SMTP SMS  SMS для подтверждения но... 	<p>Имя отправителя Адрес отправителя </p> <p>не указано noreply@diplom.firebaseio.com</p> <p>Адрес для ответа</p> <p>не указано</p> <p>Тема</p> <p>Перегляньте лист для додатка parking</p> <p>Сообщение</p> <p>Вітаємо, %DISPLAY_NAME%!</p> <p>Перейдіть за цим посиланням, щоб підтвердити свою електронну адресу.</p> <p><a href="https://diplom.firebaseio.com/_/auth/action?mode=<action>&oobCode=<code>">https://diplom.firebaseio.com/_/auth/action?mode=<action>&oobCode=<code></p> <p>Якщо Ви не запитували підтвердження адреси, не звертайте уваги на цей лист.</p> <p>Дякуємо,</p> <p>Команда parking</p>
<p>Язык шаблона Ukrainian </p>	

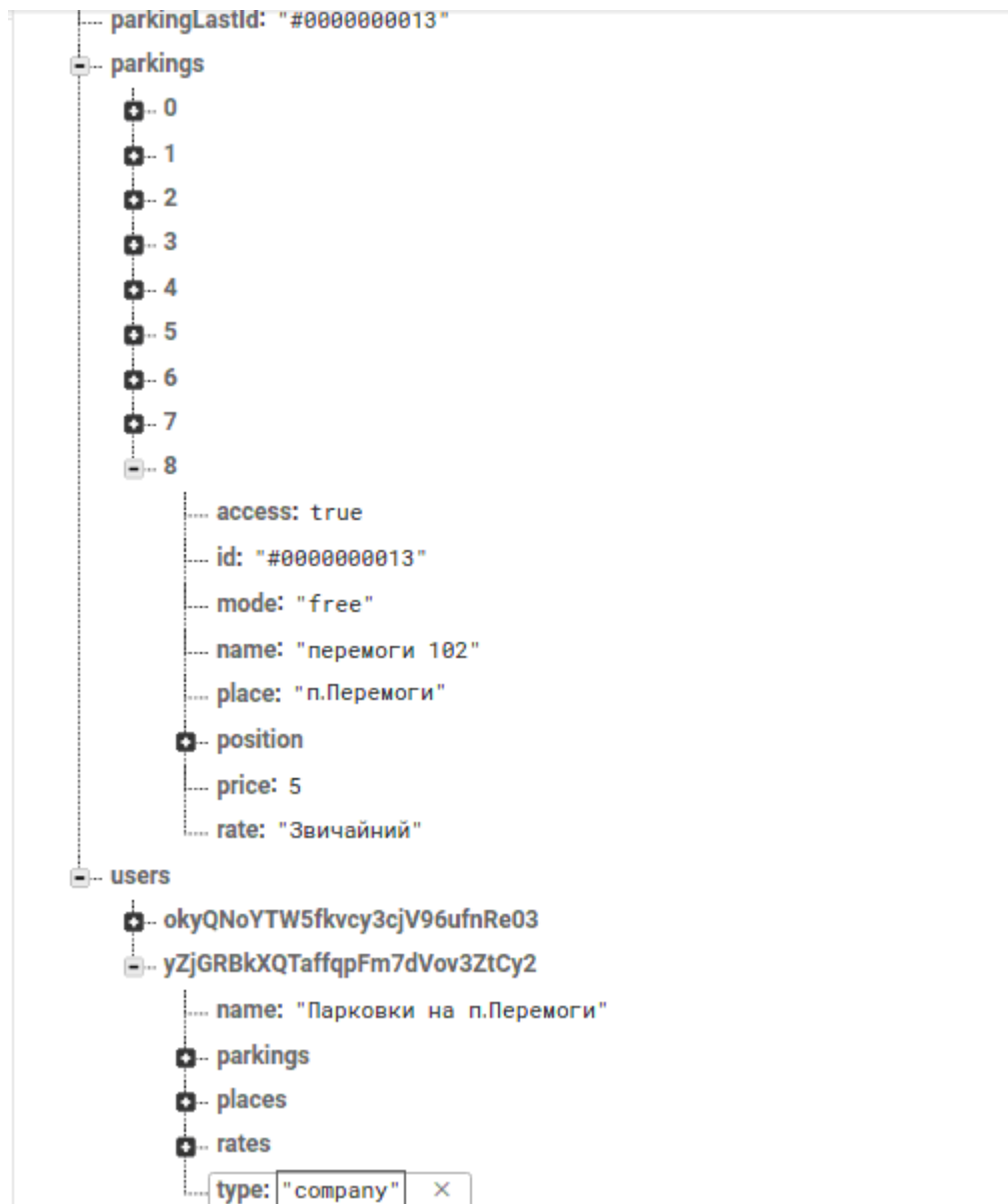
Шаблон підтвердження електронної пошти для автентифікації Firebase



Блок-схема роботи додатку в корпоративному режимі



Алгоритм для користувацького режиму



Фрагменти даних з бази даних

Тестування можливостей системи

№	Об'єкт тестування	Час виконання у мобільній версії	Час виконання веб-сторінкою
1	Авторизація користувача	~0.5012	~0.4511
2	Реєстрація користувача	~0.9108	~0.8925
3	Запуск системи	~3.1707	~2.8601
4	Створення нової парковки	~0.8487	~0.7992
5	Створення нового тарифу	~0.5578	~0.4135
6	Редагування парковки	~0.8159	~0.7811
7	Редагування тарифу	~0.5091	~0.3973
8	Видалення парковки	~0.3328	~0.2981
9	Видалення тарифу	~0.3295	~0.2877
10	Оновлення даних	~1.5108	~1.3905
11	Пошук парковки	~0.8108	~0.8007

Тестування можливостей системи