

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ДОДАТКУ ДЛЯ ПОШУКУ ЗАГУБЛЕНИХ ТВАРИН НА
БАЗІ REACT NATIVE ТА NODE.JS
DEVELOPMENT OF AN APPLICATION FOR FINDING LOST ANIMALS
BASED ON REACT NATIVE AND NODE.JS

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗ-42
Ковальчук Віталій Анатолійович


(підпис)

Керівник:
асистент
Потейчук Михайло Іванович


(підпис)

Кваліфікаційну роботу
допущено до захисту
« 4 » 06 2024 р.
к.т.н., доцент
Гарант освітньої програми:
Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М.П. М.П.
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Ковальчуку Віталію Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка додатку для пошуку загублених тварин на базі React Native та Node.js

Керівник роботи: асистент, Потейчук Михайло Іванович

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, оцінка ергономічних та надійнісних параметрів проектованої системи, функціонально-структурна схема роботи об'єкта проектування, розробка моделі бази даних об'єкта проектування.

5. Перелік графічного матеріалу: 15 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Потейчук М. І.		
Специфікації вимог до розробленої системи	Потейчук М. І.		
Розробка об'єкта проєктування	Потейчук М. І.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		4,5%	
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/П	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	
4	Розробка функціонально-структурної схеми роботи об'єкту проєктування та проєктування бази даних	17.04.2024 р.	
5	Практична реалізація об'єкту проєктування та розробка бази даних	18.05.2024 р.	
6	Тестування та налагодження об'єкту проєктування	01.06.2024 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	05.06.2024 р.	

Здобувач вищої освіти

 (підпис)

Керівник кваліфікаційної роботи

 (підпис)

Ковальчук В. А.

(прізвище, ініціали)

Потейчук М. І.

(прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: Ковальчук Віталій Анатолійович

Тема: «Розробка додатку для пошуку загублених тварин на базі React native та node.js».

Коротка характеристика кваліфікаційної роботи та прийнятих рішень:

Кваліфікаційна робота бакалавра складається з трьох розділів, в яких проаналізовані проблематика та поставлені завдання за темою роботи, здійснено теоретичне дослідження та практичну реалізацію онлайн-сервісу для планування заходів та участі у них.

Самостійні розробки і пропозиції автора: Автор самостійно виконав розробку онлайн-сервісу для планування заходів та участі у них

Практичне значення роботи полягає у створенні дієвого інструмента, який може бути використаний у реальних умовах для поліпшення процесу пошуку загублених тварин, зниження ризиків їх тривалого перебування без догляду, а також для зменшення емоційного навантаження на їх власників.

Недоліки: Істотних недоліків в роботі не виявлено.

Загальний висновок: Кваліфікаційна робота бакалавра здобувача освіти Ковальчука Віталія Анатолійовича присвячена розробці сервісу для пошуку загублених тварин. У роботі проаналізовано проблематику та поставлені завдання, виконано теоретичне дослідження та практичну реалізацію додатку, а також проведено тестування розробленої системи.

Автор самостійно виконав розробку додатку, створивши зручний інструмент для пошуку загублених тварин.

Кваліфікаційна робота здобувача освіти Ковальчука Віталія Анатолійовича рекомендується до захисту екзаменаційній комісії та заслуговує позитивної оцінки.

Рецензент кваліфікаційної роботи: Здолбіцька Ніна Василівна, к.т.н., доцент кафедри комп'ютерних наук

(прізвище, ім'я, по-батькові, посада)



(підпис)

« 7 » червня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач вищої освіти Ковальчук Віталій Анатолійович
(прізвище, ім'я, по батькові)
група ІІІЗ-42

Тема кваліфікаційної роботи:

Розробка додатку для пошуку загублених тварин на базі React Native та node.js

Керівник Потейчук Михайло Іванович

Актуальність теми

Щороку в Україні та світі губиться велика кількість домашніх тварин, що призводить до значних емоційних страждань власників і потенційно небезпечних ситуацій для самих тварин. Створення мобільного додатку, який би допомагав у пошуку та поверненні загублених тварин, може значно полегшити цей процес та зменшити кількість таких випадків.

Об'єкт дослідження

Процес пошуку та повернення загублених тварин за допомогою мобільних технологій

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи:

Автор аналізує різні аналоги, визначає їх особливості і обґрунтовує своє рішення щодо розробки власного додатку. Розроблений додаток з використанням бази React Native та node.js у відповідності до теми.

Зауваження та недоліки: Суттєвих недоліків в роботі не виявлено.

Загальний висновок:

Кваліфікаційна робота бакалавра виконана на високому рівні. Робота виконана відповідно до вимог, логічно й послідовно описує хід виконання поставленого завдання. Пояснювальна записка відповідає стандартам до її оформлення. Робота може бути оцінена на високу оцінку

Керівник


(підпис)

(Потейчук М. І.)

(прізвище, ім'я, по батькові)

« 7 » 06 2024 р.

АНОТАЦІЯ

Ковальчук В. А. Розробка додатку для пошуку загублених тварин на базі React Native та node.js. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки додатків з використанням технологій React Native та node.js і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено додаток для пошуку загублених тварин на базі React Native та node.js і здійснено його тестування. У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки можуть бути застосовані на практиці для пошуку загублених домашніх тварин.

Ключові слова: React Native, node.js, сервер, клієнт, веб-додаток.

ABSTRACT

Kovalchuk V. A. Development of an application for finding lost animals based on React Native and node.js. Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three chapters, conclusions and proposals, a list of used sources and annexes.

In the first chapter, an analysis of the current state of the problem of application development using React Native and node.js technologies were carried out and the task was formulated. The second section defines the requirements for the developed system, as well as means, methods and development algorithms. In the third chapter, an application for searching for lost animals based on React Native and node.js was developed and tested. The conclusions summarize the information presented in the previous parts. The results of the development can be applied in practice to find lost pets.

Keywords: React Native, node.js, server, client, web application.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ СЕРВІСІВ ПОШУКУ ЗАГУБЛЕНИХ ТВАРИН І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	7
1.1 Аналіз сучасного стану проблеми	7
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	12
Висновки до розділу 1.....	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного сервісу для пошуку загублених тварин та проектування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	18
Висновки до розділу 2.....	21
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ ДЛЯ ПОШУКУ ЗАГУБЛЕНИХ ТВАРИН..	22
3.1 Практична реалізація об'єкта проектування.....	22
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	30
3.3 Розгортання та публікація розробленого сервісу для пошуку загублених тварин	34
Висновки до розділу 3.....	36
ВИСНОВКИ	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40

ВСТУП

Актуальність теми полягає в тому, що щороку в Україні та світі губиться велика кількість домашніх тварин, що призводить до значних емоційних страждань власників і потенційно небезпечних ситуацій для самих тварин. Створення мобільного додатку, який би допомагав у пошуку та поверненні загублених тварин, може значно полегшити цей процес та зменшити кількість таких випадків. Використання сучасних технологій, таких як React Native для розробки крос-платформних мобільних додатків та Node.js для серверної частини, дозволяє створити ефективний та швидкодіючий інструмент.

Метою роботи є розробка функціонального мобільного додатку для пошуку загублених тварин, який би включав можливості реєстрації користувачів, реєстрації тварин, ведення бази даних зниклих тварин.

Реалізація мети вимагає вирішення наступних завдань:

- аналіз існуючих рішень у сфері пошуку загублених тварин;
- обґрунтування засобів, методів та алгоритмів розробки додатку для пошуку загублених тварин;
- проектування архітектури додатку та бази даних;
- розробка інтерфейсу користувача за допомогою React Native;
- написання серверного коду на Node.js;
- проведення комплексного тестування додатку.

Об'єктом дослідження є процес пошуку та повернення загублених тварин за допомогою мобільних технологій.

Предметом дослідження є методи та інструменти розробки мобільних додатків, зокрема використання React Native та Node.js.

Практичне значення розробки полягає у створенні дієвого інструмента, який може бути використаний у реальних умовах для поліпшення процесу пошуку загублених тварин, зниження ризиків їх тривалого перебування без догляду, а також для зменшення емоційного навантаження на їх власників.

РОЗДІЛ 1

АНАЛІЗ СЕРВІСІВ ПОШУКУ ЗАГУБЛЕНИХ ТВАРИН І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Існує багато застосунків та сервісів для пошуку загублених тварин, які використовують різноманітні технології від соціальних мереж до GPS-стеження. Розглянемо особливості деяких з них.

PawBoost [1] – це додаток, який допомагає возз'єднати загублених тварин з їхніми власниками, надаючи можливість легко публікувати та поширювати оголошення у вашій місцевості. Він використовує мережу любителів тварин та надає оновлення щодо кількості людей, які переглядають ваше оголошення (рис. 1.1).

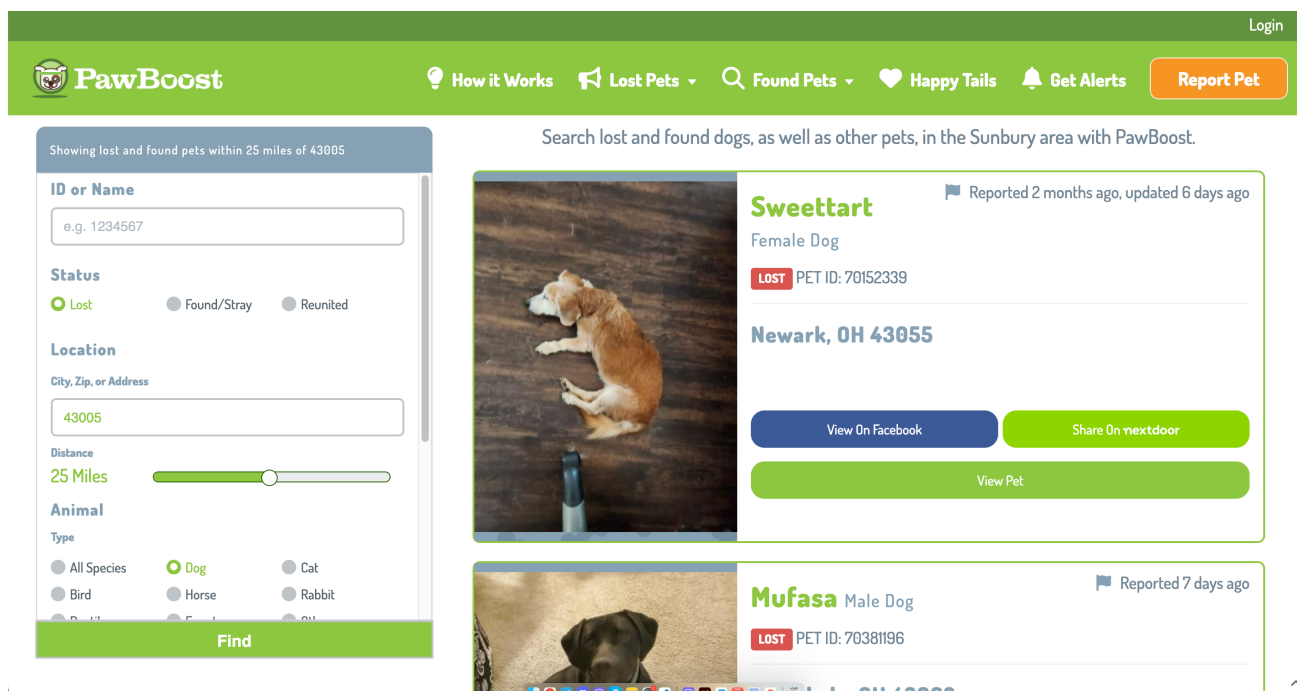


Рисунок 1.1 – Сервіс PawBoost для пошуку загублених тварин [1]

Petco Love Lost [2] – це сервіс, що пропонує безкоштовну та обширну базу даних загублених і знайдених тварин. Ви можете шукати свою тварину або повідомити про її зникнення, сервіс охоплює широку географічну область.

Цей сервіс заслуговує увагу тим, що використовує технологію комп'ютерного зору для пошуку загублених собак. Ви можете завантажити фотографію вашої тварини, і якщо вона загубиться, додаток може порівняти її обличчя з зображеннями знайдених тварин, завантаженими іншими користувачами (рис. 1.2). Додаток інтегрований з мережею організацій, що опікуються тваринами, що збільшує шанси знайти тварину.

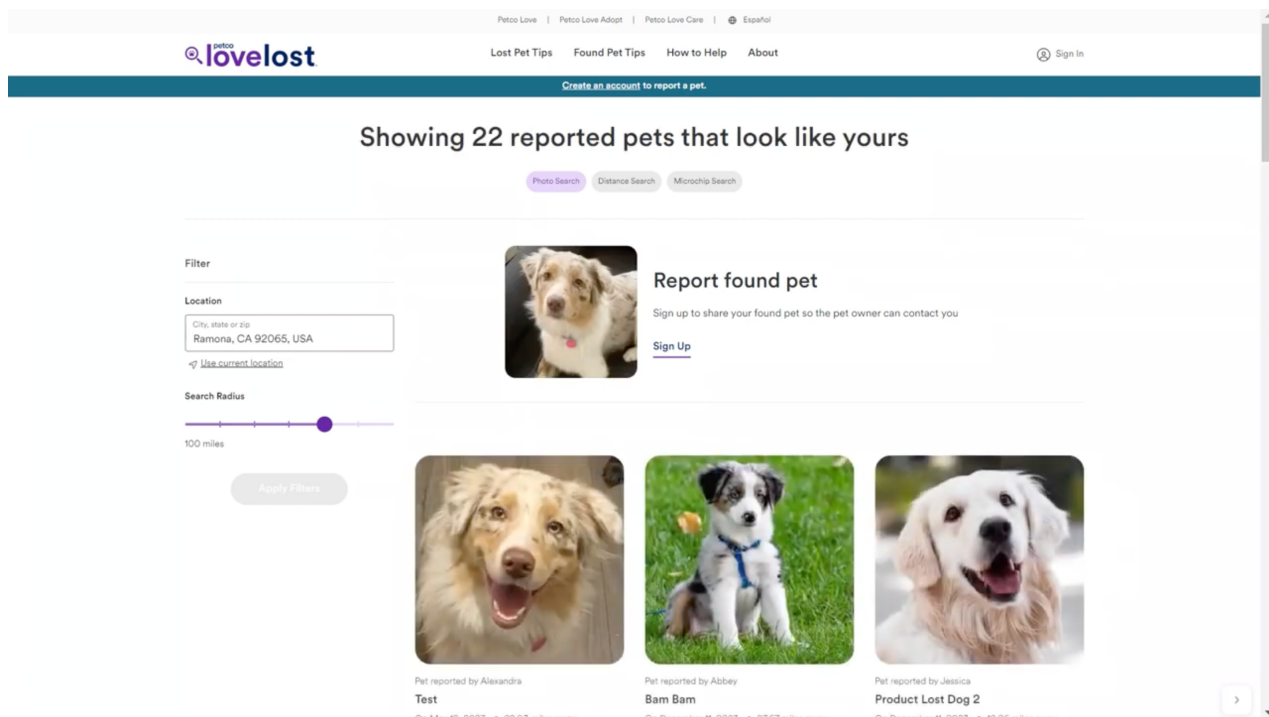


Рисунок 1.2 – Результат розпізнавання і пошуку схожих собак за допомогою сервісу Petco Love Lost [2]

Whistle Pet GPS Tracker [3] – це рішення, що включає мобільний додаток і пристрій, який кріпиться до нашийника вашої тварини та відстежує її місцезнаходження за допомогою GPS, дозволяючи отримувати оновлення в реальному часі та встановлювати «Whistle Zone» для сповіщень, якщо ваша тварина покине визначену зону (рис. 1.3).

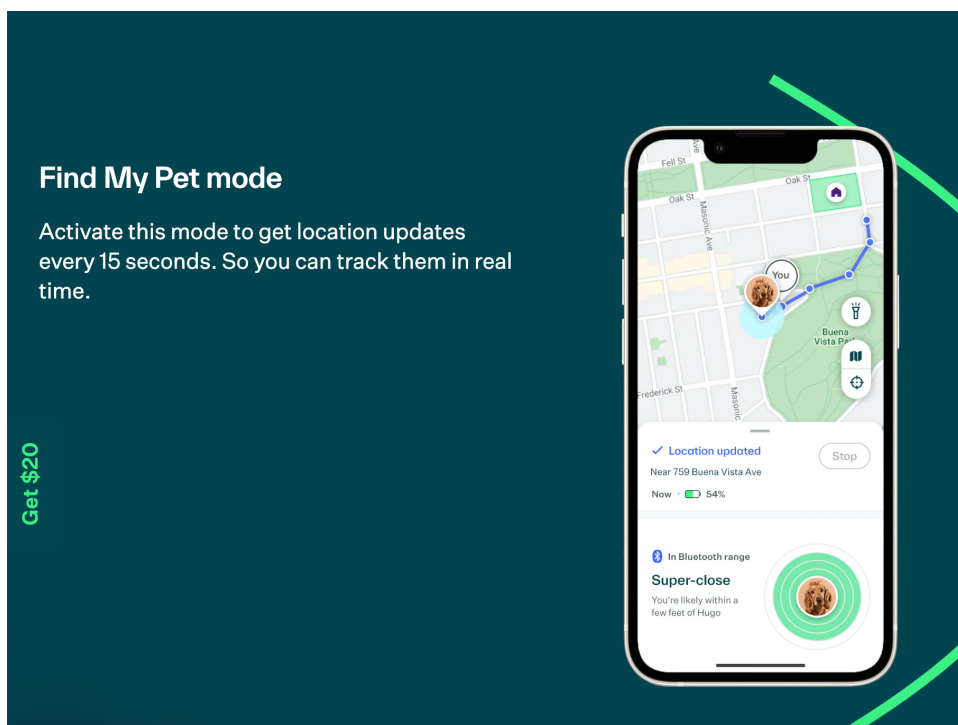


Рисунок 1.3 – Мобільний додаток Whistle Pet GPS Tracker [3]

Pet FBI (Pets Found by Internet) [4] є некомерційною організацією, яка надає обширну онлайн базу даних, де власники можуть повідомити про загублену тварину або шукати її по всій території США та Канади. Ця організація також надає ресурси та поради щодо віднайдення втрачених тварин (рис. 1.4).

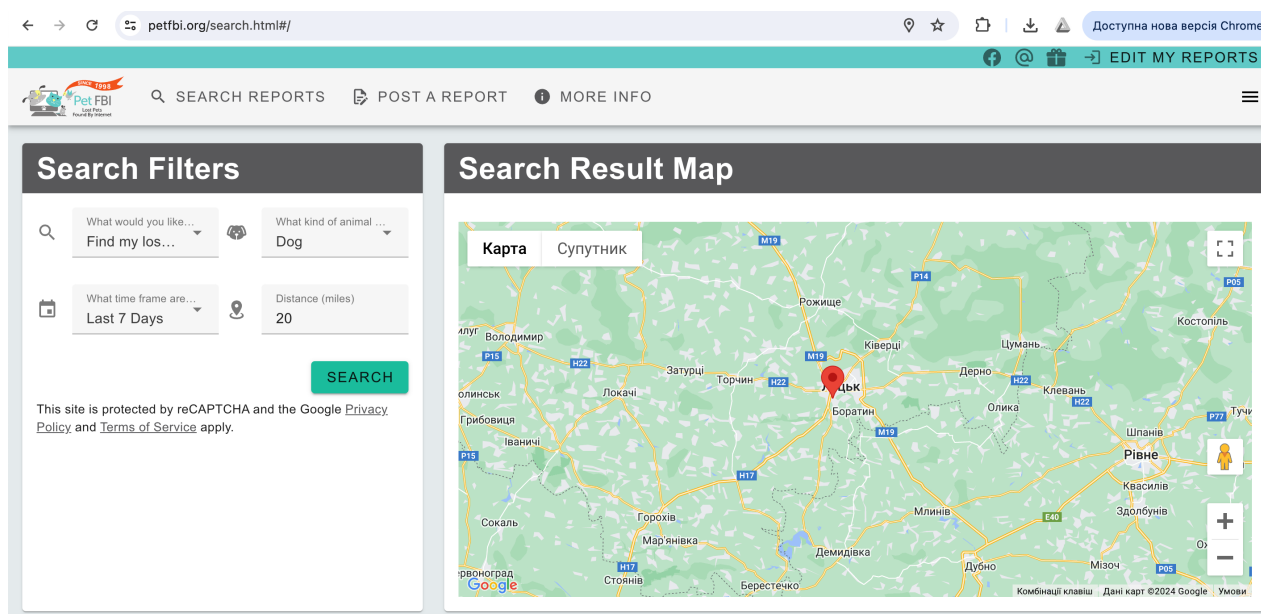


Рисунок 1.4 – Сервіс Pet FBI [4]

Nextdoor Pet Directory [5] – це функція спільноти в додатку Nextdoor, де сусіди можуть ділитися інформацією про своїх тварин. Якщо ваша тварина загубиться, ви можете швидко сповістити про це мешканців вашого району, використовуючи цей довідник (рис. 1.5).

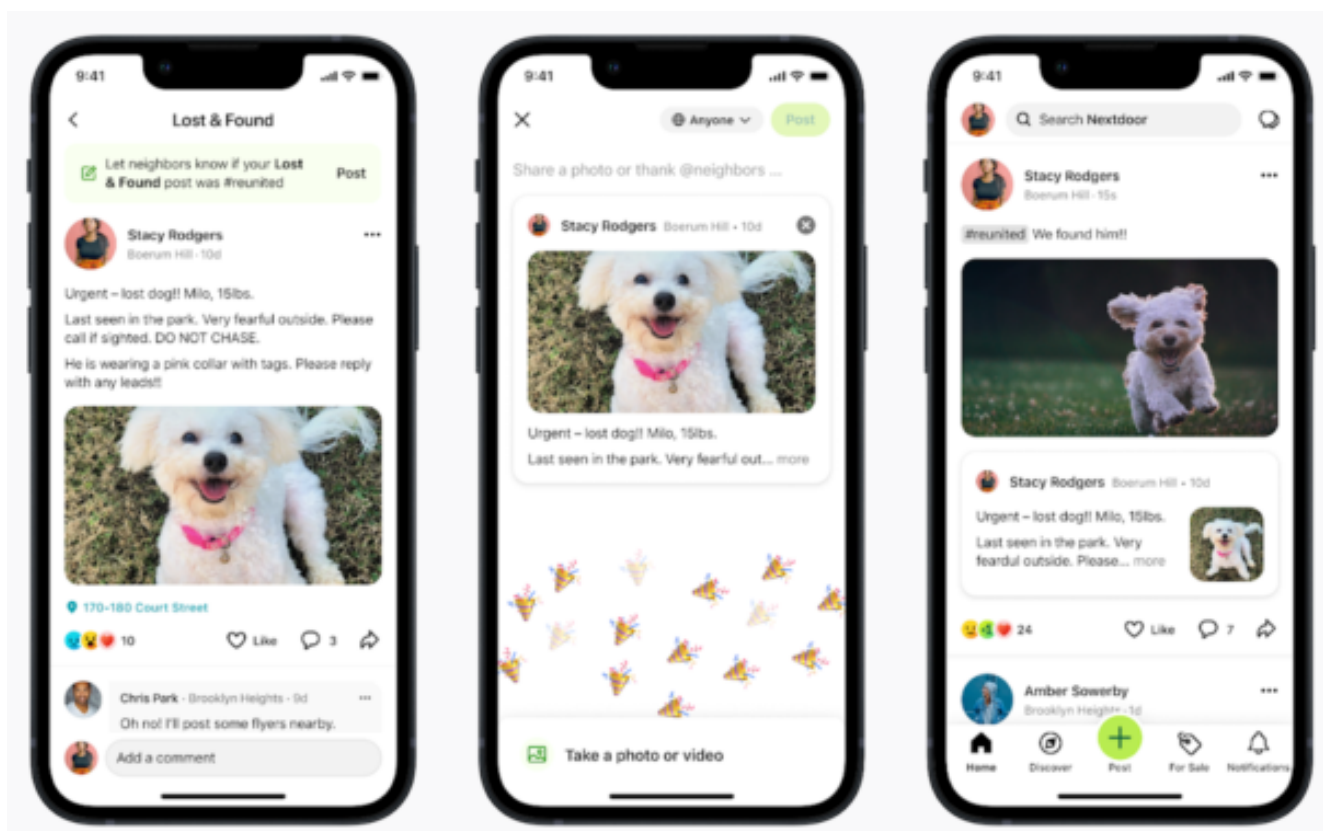


Рисунок 1.5 – Nextdoor Pet Directory [5]

Ці інструменти пропонують різні методи пошуку загублених домашніх тварин, від традиційних методів пошуку до сучасних цифрових рішень, підвищуючи ваші шанси на щасливе возз'єднання з вашим вихованцем.

Недоліком всіх цих засобів для України є їх низька популярність в нашому регіоні, пов'язана, зокрема, з відсутністю української мови інтерфейсу. Ці сервіси переважно орієнтуються на ринок США.

В таблиці 1.1 наведена порівняльна характеристика вищезгаданих сервісів за наступними критеріями: технологія, особливості, платформа, на якій працює сервіс, основні переваги і недоліки.

Таблиця 1.1 – Порівняльна характеристика сервісів для пошуку загублених тварин

Інструмент	Технологія	Особливості	Платформа	Вартість	Переваги	Недоліки
Paw Boost	Мережа користувачів	Поширення оголошень у місцевості, статистика переглядів	Веб, iOS, Android	Безкоштовно	Велика мережа користувачів, швидке поширення	Обмежена функціональність без преміум-акаунта
Petco Love Lost	База даних, розпізнавання облич	Обширна база даних загублених і знайдених тварин, пошук за допомогою фотографії, інтеграція з організаціями	Веб, iOS, Android	Безкоштовна	Велика база даних, легке використання, висока точність завдяки розпізнаванню облич	Технологія розпізнавання покриває лише собак, планується додавання котів
Whistle Pet GPS Tracker	GPS-стеження	Моніторинг місцезнаходження в реальному часі, налаштування зон сповіщень	GPS-пристрій	Платна	Точне відстеження місцезнаходження	Висока вартість, потребує носіння пристрою
Pet FBI	Онлайн база даних	Велика база даних з загубленими тваринами, поради та ресурси для пошуку	Веб	Безкоштовна	Доступність у всій країні, корисні ресурси	Обмежена діяльність поза США та Канадою
Next door Pet Directory	Соц. мережа	Ділення інформацією про тварин у спільноті, швидке сповіщення сусідів	Веб, iOS, Android	Безкоштовна	Легке сповіщення сусідів, інтеграція з соцмережами	Ефективність залежить від активності спільноти

В Україні популярним методом пошуку загублених тварин є соціальні мережі, зокрема тематичні групи у Facebook. Одним з таких прикладів є група «Загублені та Знайдені Тварини. Луцьк та Волинська обл.», що станом на квітень 2024 року налічує понад 30,5 тис. учасників (рис. 1.6) [6].

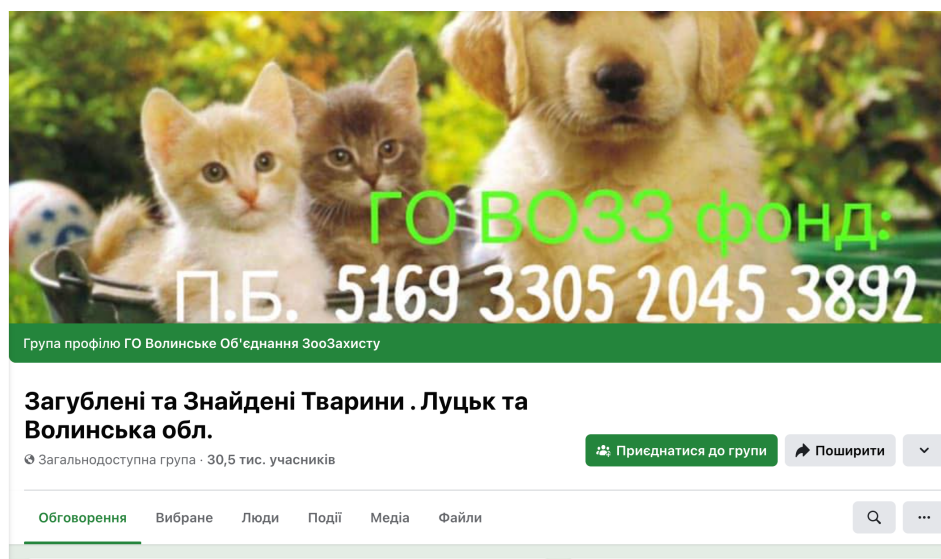


Рисунок 1.6 – Спільнота Facebook з пошуку загублених тварин [6]

Однією з головних переваг соціальних мереж є їх широка аудиторія. Проте, використання соціальних мереж має і свої недоліки. Основним з них є те, що інформація може бути розсіяною і неструктурованою, що ускладнює її пошук. Крім того, на соціальних платформах часто зустрічаються фейкові новини та шахрайські повідомлення, що може відволікати від реальних оголошень. Ще одним недоліком є те, що пости можуть швидко загубитися в потоці новин, особливо у великих спільнотах, де оновлення відбуваються дуже часто.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

У відповідності до теми кваліфікаційної роботи бакалавра, а також після дослідження сучасного стану проблеми та проведеного аналізу існуючих додатків і сервісів для пошуку загублених тварин було поставлено наступні завдання на кваліфікаційну роботу бакалавра:

- обґрунтування засобів, методів та алгоритмів розробки додатку для пошуку загублених тварин;
- розробка архітектури додатку, включно з фронтенд (React Native) та бекенд (Node.js) компонентами, а також структурою бази даних;

- створення інтуїтивно зрозумілого та зручного інтерфейсу користувача за допомогою React Native;
- написання серверного коду на Node.js;
- проведення комплексного тестування додатку для забезпечення його стабільності та надійності.

Висновки до розділу 1

У результаті проведеного аналізу було виявлено, що існує значна кількість додатків та сервісів для пошуку загублених тварин, які використовують різноманітні технології та підходи. Наприклад, такі додатки як PawBoost та Petco Love Lost пропонують різні методи взаємодії користувачів з інформацією про зниклих тварин, від поширення оголошень через мережу користувачів до використання технологій розпізнавання облич. Інші сервіси, як Pet FBI, надають обширні бази даних для зберігання та пошуку інформації про зниклих та знайдених тварин.

Окрім цього, пристрої типу Whistle Pet GPS Tracker забезпечують точне відстеження місцезнаходження тварин у реальному часі, що значно збільшує шанси на їхнє швидке повернення додому. Водночас, такі сервіси як Nextdoor Pet Directory активно використовують соціальні мережі для швидкого поширення інформації серед сусідів та мешканців певної території. В Україні популярним засобом пошуку загублених тварин є соціальні мережі.

Всі ці інструменти мають свої унікальні переваги та недоліки, що слід враховувати при розробці нового додатку для пошуку загублених тварин.

Таким чином, було визначено основні завдання для кваліфікаційної роботи бакалавра, серед яких розробка архітектури додатку на основі технологій React Native та Node.js, створення зручного інтерфейсу користувача, а також проведення комплексного тестування для забезпечення стабільної та надійної роботи додатку.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного сервісу для пошуку загублених тварин та проектування програмного забезпечення

Програмний продукт, розроблений в рамках цієї кваліфікаційної роботи, повинен відповідати низці вимог. Додаток буде створено за допомогою фреймворку React Native, що дозволить забезпечити крос-платформну сумісність з різними мобільними операційними системами. Node.js використовуватиметься для розробки серверної частини, що забезпечуватиме високу продуктивність обробки даних та взаємодію з базою даних.

База даних повинна містити не тільки основну інформацію про клієнтів та їхні записи, але й додаткові дані, які можуть бути корисними для покращення користувацького досвіду. Важливо забезпечити, щоб додаток міг ефективно обробляти запити в реальному часі та мав інтуїтивно зрозумілий інтерфейс, який би відповідав сучасним стандартам дизайну.

До розроблюваного додатку ставляться наступні функціональні вимоги:

- програмний продукт має надавати користувачам можливість реєстрації та входу, щоб забезпечити персоналізацію та безпеку даних;
- користувачі повинні мати можливість реєструвати зникнення тварин, включаючи детальну інформацію про тварину та обставини зникнення;
- додаток повинен містити функцію пошуку, що дозволяє користувачам знаходити записи про зниклих тварин за допомогою різних критеріїв;
- має бути передбачено можливість спілкування між користувачами через систему повідомлень або коментарів.

Нефункціональні вимоги наступні:

- продукт має бути інтуїтивно зрозумілим і легким у використанні, забезпечуючи високу швидкість завантаження та швидкість реакції додатку на дії користувача;

- безпека даних є критичною, відповідно система повинна включати надійні механізми аутентифікації та шифрування;
- додаток повинен бути масштабованим, щоб витримувати зростання числа користувачів та даних;
- важливою є підтримка крос-платформності, оскільки додаток має функціонувати на різних мобільних операційних системах.

На рисунку 2.1 представлено діаграму варіантів використання додатку, що описує його основні функціональні можливості.

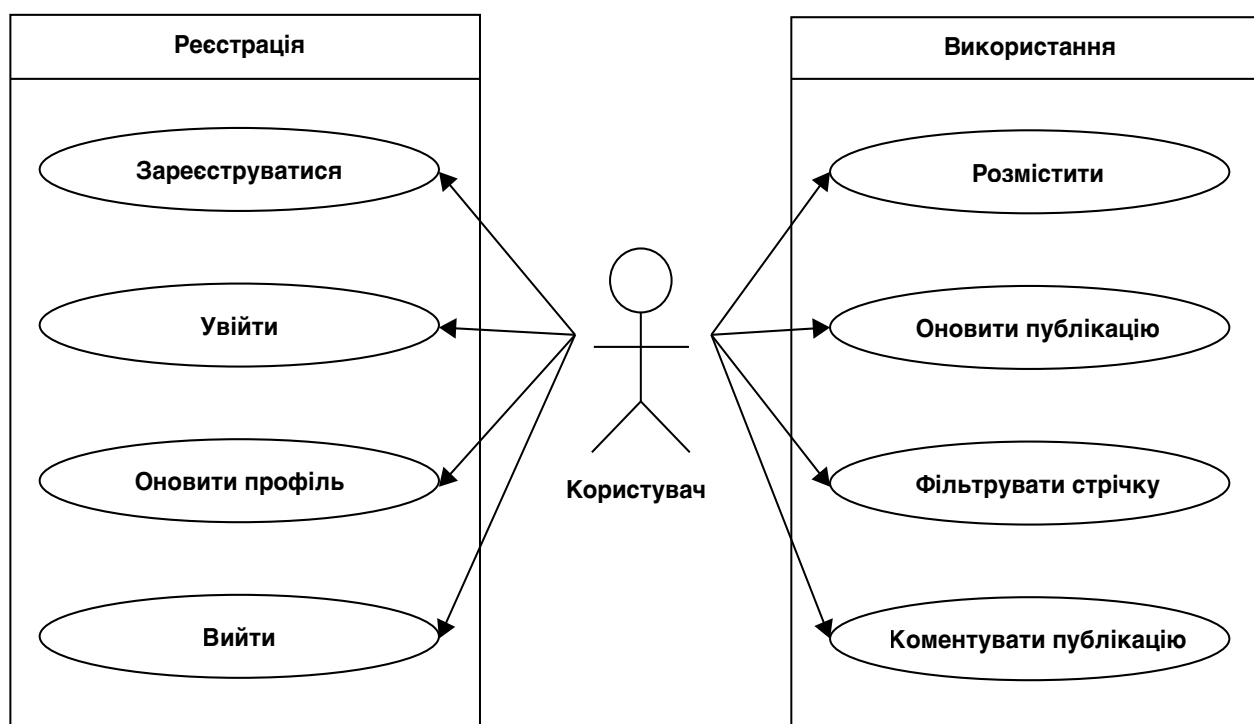


Рисунок 2.1 – Діаграма варіантів використання

Користувач може реєструватися, здійснювати вхід, оновлювати профіль і виходити з профілю. Також користувач, який створив профіль і здійснив вхід має можливість розміщувати публікації, оновлювати їх, здійснювати пошук і фільтрування, а також залишати коментарі.

Для підтримки запропонованих операцій системи використовується проста та чітко визначена структура даних, яка складається з трьох основних сутностей:

«animals», «users» та «files» – тварини, користувачі та файли відповідно (рис. 2.2).

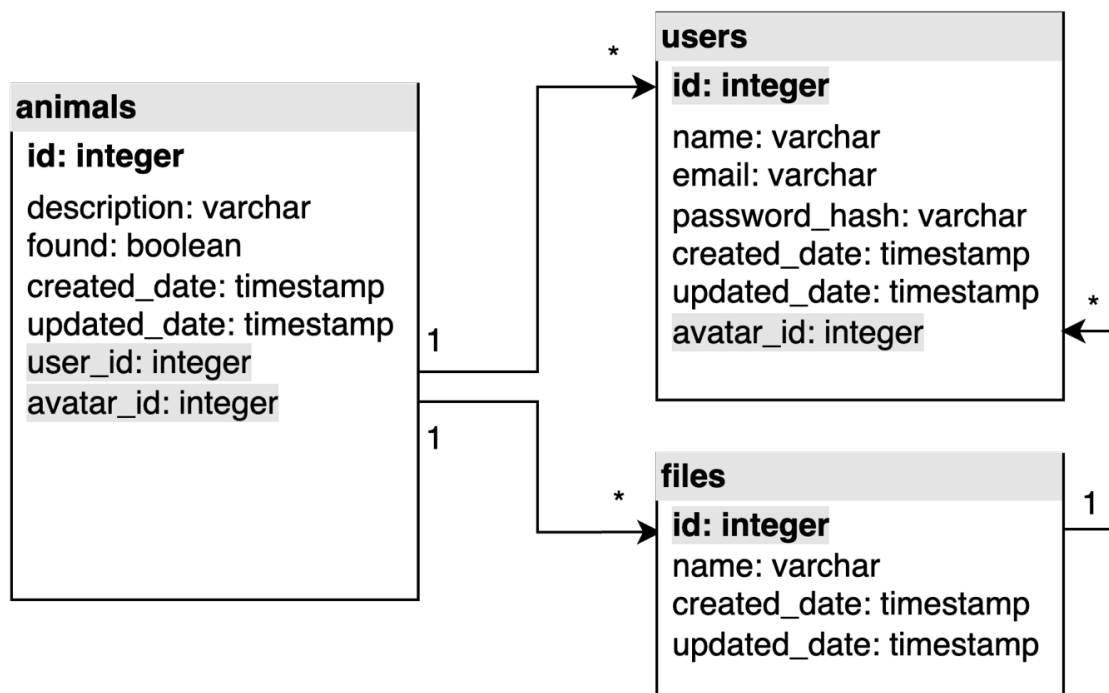


Рисунок 2.2 – Модель сутностей бази даних

«Animals» – зосереджують інформацію про загублених або знайдених тварин. Фактично, це сам пост.

«Users» – це реєстрація всіх користувачів системи. Ці користувачі можуть бути авторами постів, авторами коментарів або просто підписниками.

«Files» – це коментарі, пов’язані з постами.

Вся реляційна структура основних сутностей підтримується в СУБД PostgreSQL, яка також буде опублікована на хмарному сервері.

Було обрано набір нереляційної інформації, яка зазнала мало або взагалі не зазнала еволюції життєвого циклу даних, що зберігаються в системі на основі документів, тобто в нереляційній СУБД.

Згідно з рисунком 2.3, дані, оброблені в рамках цієї операції, надсилаються користувачам як стратегія для покращення контактів, що в деяких випадках може бути вирішальним для процесу пошуку.

notifications
_id: integer content: string user_send: string id_animal: integer name_user: string avatar: string createDate: string

Рисунок 2.3 – Модель документів бази даних MongoDB

У цьому випадку інформація, пов'язана з повідомленнями постів «notifications», зберігається в базі даних MongoDB. Деякі дані з цієї моделі, очевидно, дублюються в обох моделях, однак, враховуючи нереляційний характер моделі Mongo, це є не проблемою, а є стратегією продуктивності в процесі пошуку цих даних. Вся інформація в цій базі даних надсилається через асинхронну комунікацію користувачам, залученим до процесу публікації. Дескриптор схеми JSON показаний у лістингу 2.1.

Лістинг 2.1 – Дескриптор схеми JSON для документування моделі MongoDB

```
{
  "type": "object",
  "title": "notification",
  "properties": {
    "content": {
      "type": "string",
      "description": "зміст повідомлення"
    },
    "user_sent": {
      "type": "string",
      "description": "ідентифікація користувача-автора"
    },
    "id_animal": {
      "type": "integer",
      "description": "ідентифікація публікації"
    },
    "name_user": {
```

```

    "type": "string",
    "description": "ім'я користувача"
  },
  "avatar": {
    "type": "string",
    "description": "зображення користувача"
  },
  "createDate": {
    "type": "string",
    "description": "дата публікації"
  },
  "updateDate": {
    "type": "string",
    "description": "дата оновлення"
  }
},
"required": [
  "user_sent",
  "id_animal",
  "name_user",
  "createAt",
  "updateAt"
]
}

```

Кінець лістингу 2.1

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для розробки цієї системи були використані інструменти, процеси та технології підтримки, що більш детально описані далі.

Для розробки пропонованого додатку передбачається використання технології Node.js та фреймворку React Native.

Node.js [7] – це серверний фреймворк, заснований на мові JavaScript, який дозволяє реалізацію високопродуктивних мікросервісів. Завдяки асинхронній моделі обробки подій, Node.js може ефективно обробляти велику кількість

одночасних з'єднань, що робить його ідеальним для створення масштабованих веб-додатків і API.

React Native [8] – це фреймворк для розробки мобільних застосунків, створений компанією Facebook. Він дозволяє розробляти додатки для платформ iOS та Android, використовуючи мову програмування JavaScript та бібліотеку React. React Native є популярним вибором для створення мобільних застосунків завдяки своїй зручності, ефективності та потужним можливостям.

Для редагування вихідного коду використовувався Visual Studio Code, який забезпечує більшу продуктивність та організацію [9].

Visual Studio Code – це відкрите середовище розробки, розроблене та підтримуване Microsoft. Воно підтримує безліч мов програмування та має розширену систему плагінів, яка дозволяє налаштовувати IDE під конкретні потреби розробника. VS Code забезпечує зручні інструменти для написання, налагодження та управління версіями коду.

Для генерації APK (Android Package) додатку використовуватиметься Android Studio [10]. Android Studio – це офіційне інтегроване середовище розробки (IDE) для створення додатків під операційну систему Android. Воно базується на IntelliJ IDEA, що забезпечує розробникам потужні інструменти для написання, налагодження та тестування коду. Android Studio підтримує різні етапи життєвого циклу розробки додатків, включаючи створення візуальних інтерфейсів користувача, управління ресурсами, а також інтеграцію з системами контролю версій.

З іншого боку, за допомогою Express.js [11] буде реалізовано системний API. Це популярний фреймворк Node.js, який, крім того, що є дуже швидким, легким та гнучким, є архітектурою Model-View-Controller (MVC) та пропонує потужну колекцію ресурсів для розробки бекенду як мобільних, так і веб-додатків [12].

Docker [13] – це відкрита платформа, яка значно спрощує створення, розгортання та адміністрування ізольованих середовищ. Використовуючи Docker, розробники можуть упаковувати додаток чи середовище у контейнер, що

містить усі необхідні залежності та налаштування. Це забезпечує консистентність середовища розробки та продакшн середовища, знижуючи ризики виникнення проблем, пов'язаних з конфігурацією системи.

Також створено базу даних для зберігання інформації про клієнтів та здійснені записи для чого використовувалися MongoDB та PostgreSQL.

MongoDB [14] – це масштабована та гнучка документоорієнтована база даних, яка дозволяє зберігати дані у форматі BSON (Binary JSON). Вона забезпечує високу продуктивність і можливість легкої зміни схеми даних, що робить її зручною для розробки сучасних веб-додатків, де структура даних може часто змінюватися.

PostgreSQL [15] – це система управління об'єктно-реляційними базами даних, яка розроблена як проект з відкритим кодом. Вона відома своєю надійністю, масштабованістю та підтримкою складних SQL-запитів, транзакцій та розширень. PostgreSQL використовується у багатьох великих компаніях та проектах завдяки своїй стабільності та багатим функціональним можливостям.

Redis [16] – це відкрите сховище структур даних з ключем-значенням у пам'яті, яке забезпечує швидкий доступ до даних. Використовується для кешування, управління сесіями, черг повідомлень та інших завдань, що вимагають високої продуктивності і низької затримки.

Buddy [17] – це інструмент для підтримки процесу безперервної інтеграції (CI) та безперервного розгортання (CD) під час розробки. Buddy автоматизує процеси тестування, збірки та розгортання коду, що допомагає забезпечити стабільність та швидке впровадження змін у виробниче середовище.

Організація процесу розробки передбачає використання Agile-методологій з метою гнучкості та ефективності управління змінами, що можуть виникати в процесі реалізації проекту.

Систематичне тестування та оновлення програмного забезпечення є критичними для забезпечення його надійності та безпеки. Реалізація проекту також включатиме регулярні оновлення на основі відгуків користувачів та аналізу їх поведінки в додатку.

Висновки до розділу 2

У розділі 2 було проведено детальний аналіз вимог до розроблюваного сервісу для пошуку загублених тварин, а також спроектовано архітектуру програмного забезпечення.

Зокрема, було визначено функціональні та нефункціональні вимоги.

Функціональні вимоги включають можливість реєстрації та входу користувачів, створення і пошук постів про загублених тварин, а також взаємодію між користувачами через коментарі та повідомлення.

Нефункціональні вимоги зосереджені на забезпеченні безпеки даних, швидкодії додатку та його масштабованості.

Проектування системи передбачає використання технологій React Native для фронтенд-розробки, Node.js для бекенду та PostgreSQL для зберігання основних даних.

Було також враховано використання нереляційної бази даних MongoDB для зберігання коментарів та Redis для створення черг обробки повідомлень. Вибір цих технологій забезпечує високий рівень продуктивності, гнучкість і зручність у подальшому розвитку системи.

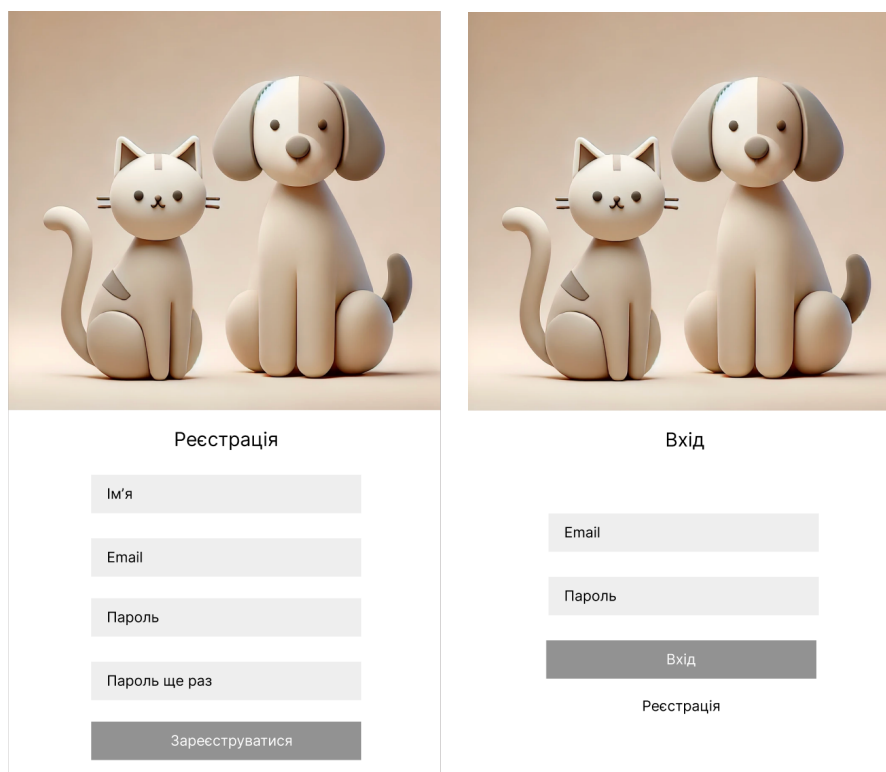
Крім того, використання Agile-методологій дозволить забезпечити ефективне управління проектом та швидке реагування на зміни у вимогах користувачів.

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ ДЛЯ ПОШУКУ ЗАГУБЛЕНИХ ТВАРИН

3.1 Практична реалізація об'єкта проектування

Після першого встановлення додатку користувач повинен зареєструватися, ввівши свої основні дані. Після цього він отримує доступ до системи і може увійти, використовуючи зареєстрований електронний лист та пароль. Відповідно за допомогою React Native було створено екрани для входу в систему та реєстрації користувача (рис. 3.1).



Реєстрація

Ім'я

Email

Пароль

Пароль ще раз

Зареєструватися

Вхід

Email

Пароль

Вхід

Реєстрація

Рисунок 3.1 – Екран входу та реєстрації в системі

Для прикладу фрагмент коду для входу, створений за допомогою React Native, показаний у лістингу 3.1.

Лістинг 3.1 – Фрагмент коду форми для входу в систему

```
import ...  
export default function Login({ navigation }) {
```

```

const passwordInputRef = useRef();
const dispatch = useDispatch();
const loading = useSelector(state => state.auth.loading);
const [userEmail, setUserEmail] = useState('');
const [userPassword, setUserPassword] = useState('');
function handleSignIn() {
    // Відправка запиту на вхід користувача
    dispatch(signInRequest(userEmail, userPassword));
}
return (
    <Background>
        <Container>
            <Form>
                { /* Поле для введення електронної пошти */ }
                <FormInput
                    keyboardType="email-address"
                    placeholder="Введіть ваш email"
                    returnKeyType="next"
                    onSubmitEditing={() =>
passwordInputRef.current.focus() }
                    value={userEmail}
                    onChangeText={setUserEmail}
                />
                { /* Поле для введення пароля */ }
                <FormInput { /* ... */ } />
                { /* Кнопка для відправки форми */ }
                <SubmitButton loading={loading}
onPress={handleSignIn}>Увійти</SubmitButton>
            </Form>
            { /* Посилання на реєстрацію нового користувача */ }
            <SignLink onPress={() =>
navigation.navigate('SignUp')}>
                <SignLinkText>Створити новий акаунт</SignLinkText>
            </SignLink>
        </Container>
    </Background>); }

```

Кінець лістингу 3.1

Після входу в систему користувачу доступна навігаційна панель у верхній частині додатку з вкладками «Дописи» і «Мій профіль». Натиснувши на «Мій профіль» користувач може бачити інформацію про себе, а також вносити в неї зміни (рис. 3.2).

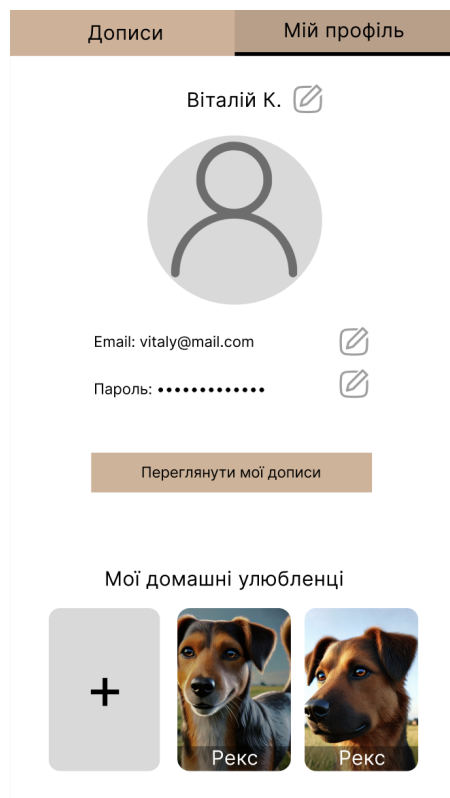


Рисунок 3.2 – Інформація про профіль з можливістю редагування

У цій функціональній опції користувач може змінювати свої основні дані, такі як пароль і відображуване ім'я. Тут користувач також може додати або змінити фото профілю. У цьому меню також знаходиться опція виходу з додатку, після чого користувачу потрібно буде знову увійти для доступу до функціональності додатку при наступному вході в систему.

При натисканні на «Дописи» в навігаційній панелі користувачу відображається стрічка з дописами інших користувачів, які вже використовують додаток. На цьому екрані доступні всі дописи про загублених тварин, які ще не були знайдені, або тварини, які були помічені без власника (рис. 3.3).

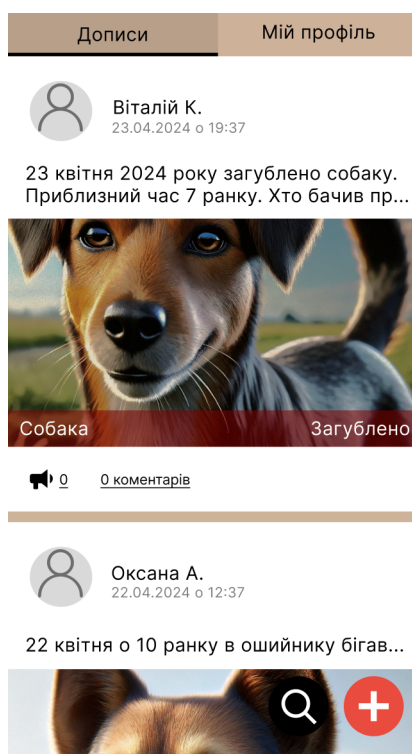


Рисунок 3.3 – Стрічка з дописами користувачів про загублених тварин

Для створення нового допису користувач повинен натиснути на «+» у нижній правій частині екрана (рис. 3.4).

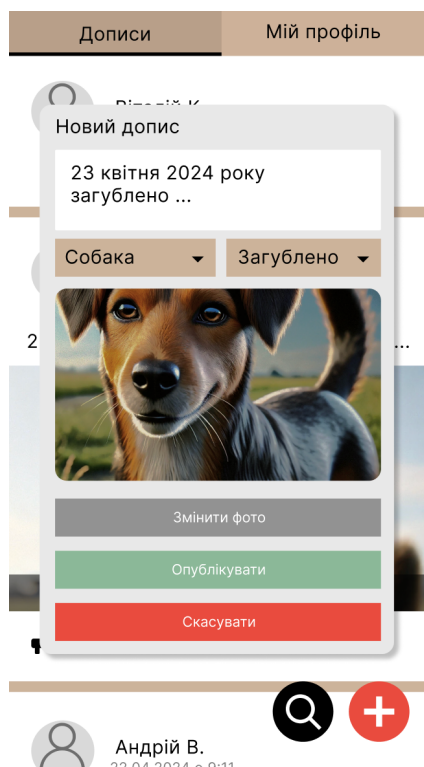


Рисунок 3.4 – Створення нового допису

При створенні нового допису користувач має можливість додати текст, фотографію, вид тварини (кіт, собака), категорію (загублено, знайдено). Вид тварини і категорія будуть доступні як параметри пошуку.

Користувачі можуть взаємодіяти з цими дописами, залишаючи коментарі. Для кожного коментаря фіксується його автор і час публікації. Побачити всі коментарі можна за допомогою смуги прокручування (рис 3.5).

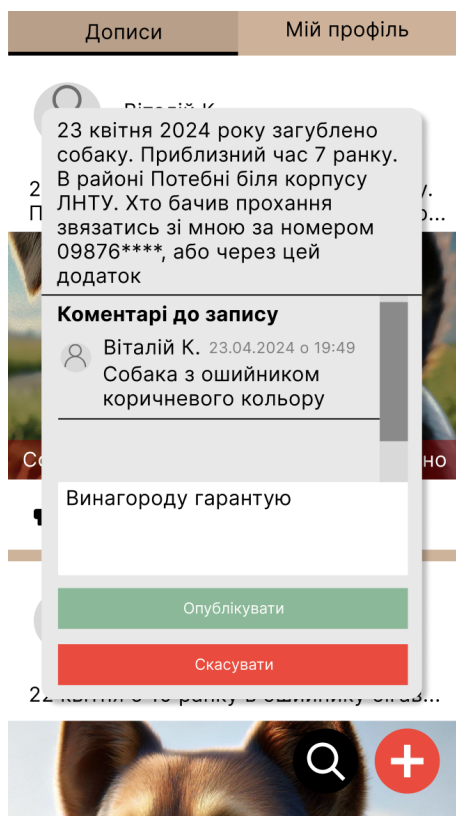


Рисунок 3.5 – Коментування допису

Для швидкої взаємодії ці коментарі також надсилаються електронною поштою всім, хто взаємодівав з певним дописом, що забезпечує сповіщення всіх зацікавлених осіб.

Коментарі зберігаються та відображаються у вигляді списків через API (Application Programming Interface), що був розроблений для цього проекту, використовуючи методи HTTP POST та GET відповідно.

Для надсилання електронних листів (лістинг 3.2) було створено чергу обробки в фоновому режимі.

Лістинг 3.2 – Фрагмент коду для надсилання сповіщень через email

```
import Mail from '../..lib/Mail';
...
class MailMessage {
  // Повернення ключа для поштової служби
  get key() {
    return 'MailMessage';
  }
  // Обробка даних для надсилання поштового повідомлення
  async handle({ data }) {
    const { user_sender, notification } = data;
    // Відправка електронної пошти з використанням контексту даних
    await Mail.sendMail({
      to: `${user_sender.name} <${user_sender.email}>`,
      subject: 'Зниклі Тварини - Новий Контакт',
      template: 'contact',
      context: {
        content: notification.content,
        user_sender: notification.name_user.name,
        date: format(
          parseISO(notification.createdAt), "'день' dd 'місяць' MM
M', o' H:mm'h'", { locale: ua }
        ), },
      });
  }
}
export default new MessageMail();
```

Кінець лістингу 3.2

Ця черга використовує вбудовану базу даних у пам'яті Redis, нереляційну базу даних, яка гарантує, що додаток не буде піддаватися можливим збоям або вузьким місцям під час виконання процесу.

Щодо коментарів, прикріплених до публікацій, вони зберігаються в розподіленій і нереляційній базі даних MongoDB. Такий вибір було зроблено через високу продуктивність при зберіганні нереляційних даних з мінімальними

змінami в життєвому циклі даних. Структура заснована на документно-орієнтованому записі, що полегшує засвоєння інформації розробниками та аналітиками, забезпечуючи кращу продуктивність під час гнучких етапів розробки програмного забезпечення (рис. 3.6).

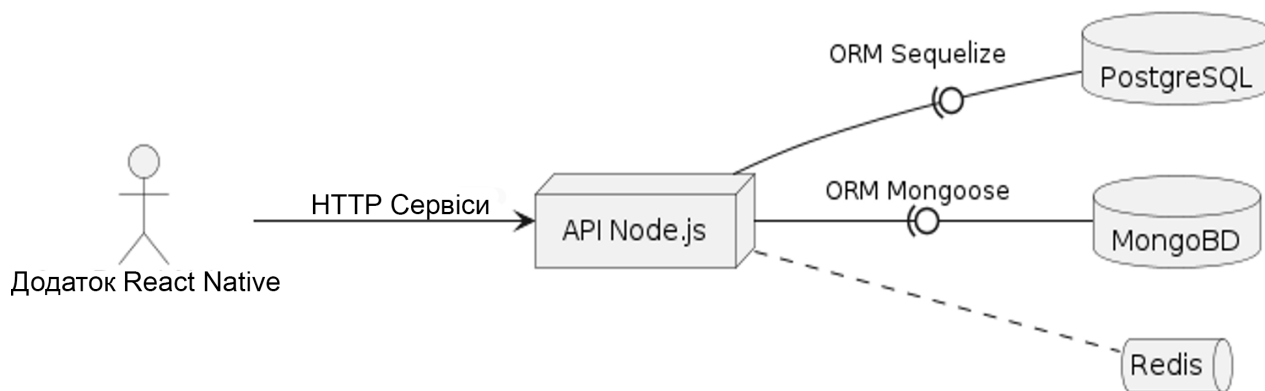


Рисунок 3.6 – Схема розгортання

На головному екрані, при натисканні кнопки «Пошук», є опції для фільтрації дописів або перегляду всіх доступних публікацій. Обидва методи, пов'язані з цими фільтрами і відповідальні за виклик API, засновані на методі HTTP GET.

Крім того, з цього екрана доступна опція створення нового посту, де користувач може описати загублену тварину або знайдену. Також користувач може зробити нове фото або додати вже існуюче зображення, збережене на пристрої, для полегшення ідентифікації тварини.

Метод, відповідальний за цю взаємодію між додатком та веб-сервісом, заснований на методі HTTP POST. Під час цього процесу дані зберігаються в реляційній СУБД PostgreSQL, одній з найбільш просунутих і популярних систем управління базами даних з відкритим вихідним кодом.

При фільтрації дописів користувач може змінювати статус допису з «загублений» на «знайдений». Після цієї операції пост більше не буде відображатися іншим користувачам; він залишається в базі даних, але

функціонально невидимий. Метод, відповідальний за цю функцію, базується на методі HTTP PUT, і лише автор посту має дозвіл на зміну його статусу.

Для всіх основних операцій системи оброблені дані користувачів або постів зберігаються постійно і безпечно в реляційній базі даних PostgreSQL.

Екрани, розроблені в цьому проекті, були створені з урахуванням найкращого користувацького досвіду, щоб задовольнити найбільші потреби, подолати труднощі та відповідати бажанням користувачів. Візуальна ідентичність екранів базується на простих малюнках, які легко зрозуміти та використовувати.

Усі ці представлені функції були розроблені виключно з метою забезпечення користувача комфортною взаємодією з інтерфейсом. Кольори та розташування іконок і інформації були обрані таким чином, щоб забезпечити чистий інтерфейс, уникнувши візуального перевантаження, з використанням яскравих і привабливих кольорів, а також сучасних компонентів, базованих на стандартах, що використовуються в найпопулярніших соціальних мережах.

Як зазначено в п. 2.2 розробка API була виконана з використанням відкритого фреймворку Express.js, що базується на мові JavaScript та Node.js. Node.js дозволяє виконувати мову JavaScript на стороні сервера, яка зазвичай використовується на стороні клієнта. Всі маршрути додатку, крім реєстрації та входу користувача, захищені і можуть бути доступні тільки за допомогою JWT токена автентифікації (JSON Web Token) [18], який буде доступний після автентифікації користувача.

Для збереження даних використовувався ORM (Object Relational Mapping) Sequelize [19] для зв'язку з PostgreSQL. Цей ORM дозволив майже повністю усунути використання SQL запитів, а також полегшив майбутню заміну бази даних, оскільки дозволяє створювати і виконувати запити незалежно від використовуваної реляційної бази даних.

Для комунікації з MongoDB також використовувався ORM Mongoose [20], який є абстракцією нативного драйвера MongoDB. Для створення черг електронних листів використовувалася нереляційна база даних пам'яті Redis, яка

зберігає всю необхідну інформацію для виконання цього завдання. Ця операція виконується у фоновому режимі, щоб не заважати іншим процесам системи. Для відстеження можливих помилок при виконанні цих черг використовувався Sentry.io [21], який є системою відстеження помилок з відкритим вихідним кодом, що підтримує широкий спектр нативних мов і фреймворків для серверів, браузерів, настільних і мобільних пристроїв. Після впровадження всі помилки легко відстежуються через портал, запропонований функціональністю.

Для спрощення впровадження цих численних інструментів використовувався Docker, який є програмною платформою, що спрощує процес створення, запуску, управління і розповсюдження додатків. Він дозволив створити автономне і ізольоване середовище для запуску та розгортання додатків. Ці середовища називаються контейнерами. Docker швидкий і кросплатформений, що дозволяє створювати контейнери на будь-якій системі, і після налаштування більше не потрібно перевстановлювати його залежності навіть при зміні комп'ютерів. Docker спростив перенесення локального API в хмару, де було потрібно лише передати використовувані залежності.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Процес тестування системи пошуку загублених тварин, розробленої на базі React Native та Node.js, включає кілька етапів для забезпечення надійності та функціональності додатку.

Модульне тестування спрямоване на перевірку окремих модулів і компонентів системи для забезпечення їх коректної роботи. Використовувалися інструменти Jest [22] та Mocha [23] для тестування компонентів React Native, а також серверної частини на Node.js. Кожна функція, метод і компонент тестувалися окремо за допомогою мок-даних для перевірки їх поведінки в різних умовах.

Інтеграційне тестування було проведено для перевірки взаємодії між різними модулями та компонентами системи. Використовувалися Supertest [24]

для тестування API та Enzyme [25] для тестування інтеграції компонентів React. Це включало тестування API-запитів до серверної частини, перевірку обробки запитів та відповідей, а також взаємодії між клієнтською та серверною частинами.

Системне тестування перевіряло всю систему як єдине ціле. Було проведено комплексні тести на повну функціональність додатку, включаючи реєстрацію користувачів, створення та редагування постів, взаємодію з коментарями, а також роботу з базою даних. Це дозволило оцінити, як система працює в реальних умовах, забезпечуючи стабільність та надійність її роботи.

Тестування користувацького інтерфейсу оцінювало зручність використання та відповідність інтерфейсу вимогам користувачів. Проводилися юзабіліті-тести з реальними користувачами, збиралися їхні відгуки, і на основі цього вносилися необхідні корективи у дизайн інтерфейсу.

Для автоматизації тестування системи пошуку загублених тварин використовувався фреймворк Cypress [26] для фронтенду і Jest з Supertest [24] для бекенду. У лістингу 3.3 наведено код для автоматизації тестування функцій реєстрації нового користувача.

Лістинг 3.3 – Тест для реєстрації нового користувача

```
describe('User Registration', () => {
  it('should register a new user', () => {
    cy.visit('/register');
    cy.get('input[name="name"]').type('Test User');
    cy.get('input[name="email"]')
      .type('testuser@example.com');
    cy.get('input[name="password"]')
      .type('password123');
    cy.get('button[type="submit"]')
      .click();
    cy.contains('Реєстрація успішна');
  });
});
```

Кінець лістингу 3.3

У лістингу 3.4 наведено код для автоматизації тестування функцій створення нового посту.

Лістинг 3.4 – Тест для створення нового посту

```
describe('Create Post', () => {
  before(() => {
    // Логін користувача
    cy.visit('/login');
    cy.get('input[name="email"]').type('testuser1@example.com');
    cy.get('input[name="password"]').type('password1234');
    cy.get('button[type="submit"]').click();
  });

  it('створюється пост про загублену тварину , () => {
    cy.visit('/create-post');
    cy.get('input[name="animalName"]').type('MyBuddy');
    cy.get('textarea[name="description"]').type('Lost near the
park');

    cy.get('input[type="file"]').attachFile('mybuddy.jpg');
    cy.get('button[type="submit"]').click();
    cy.contains('Пост створено успішно');
  });
});
```

Кінець лістингу 3.4

Код для автоматизації тестування реєстрації користувача для бекенду з Jest та Supertest наведено в лістингу 3.5.

Лістинг 3.5 – Тест для реєстрації користувача

```
const request = require('supertest');
const app = require('../app'); // наш Express додаток
describe('Реєстрація користувача', () => {
  it('реєстрація нового користувача', async () => {
    const response = await request(app)
      .post('/api/register')
      .send({
```

```

        name: 'Тестовий користувач 1',
        email: 'testuser1@example.com',
        password: 'password1234'
    });
    expect(response.statusCode).toBe(201);
    expect(response.body.message).toBe('Реєстрація успішна');
  });
});

```

Кінець лістингу 3.5

Код для автоматизації тестування для створення допису з Jest та Supertest наведено в лістингу 3.6.

Лістинг 3.6 – Тест для створення допису

```

const request = require('supertest');
const app = require('../app');
let token;
beforeAll(async () => {
  // Логін користувача та отримання JWT токена
  const response = await request(app)
    .post('/api/login')
    .send({
      email: 'testuser1@example.com',
      password: 'password123'
    });
  token = response.body.token;
});
describe('Create Post', () => {
  it('створюється новий пост про загублену тварину', async () => {
    const response = await request(app)
      .post('/api/posts')
      .set('Authorization', `Bearer ${token}`)
      .send({
        animalName: 'MyBuddy',
        description: 'Загублено біля парку',
        image: 'mybuddy.jpg'
      });
    expect(response.statusCode).toBe(201);
  });
});

```

```
expect(response.body.message)
    .toBe('Пост створено успішно');
});
});
```

Кінець лістингу 3.6

Ці приклади демонструють основні кроки для автоматизації тестування системи пошуку загублених тварин, що допоможе забезпечити якість та стабільність роботи додатку.

3.3 Розгортання та публікація розробленого сервісу для пошуку загублених тварин

Для використання додатка, створеного на React Native і Node.js, необхідно виконати кілька ключових кроків: налаштування середовища, встановлення залежностей, запуску серверної та клієнтської частин, а також забезпечення їх взаємодії.

Node.js та npm встановлені згідно з інструкціями на офіційному сайті Node.js [7].

Для налаштування та запуску бекенд коду на Node.js потрібно перейти до відповідної директорії та встановити залежності командою «npm install». Створюється файл .env для налаштування змінних середовища, додається туди налаштування для підключення до бази даних та секретні ключі (лістинг 3.7).

Лістинг 3.7 – Файл налаштування змінних середовища

```
PORT=3000
DB_HOST=localhost
DB_USER=username
DB_PASS=password
DB_NAME=database_name
JWT_SECRET=your_jwt_secret
```

Кінець лістингу 3.7

Після цього потрібно запустити сервер командою «npm start».

Для налаштування та запуску фронтенд коду на React Native потрібно перейти до відповідної директорії та встановити залежності командою «npm install». Створюється файл .env і додається URL-адреса бекенд сервера (лістинг 3.8).

Лістинг 3.8 – Параметр URL для бекенд сервера

```
API_URL=http://localhost:3000
```

Кінець лістингу 3.8

Додаток запускається командою «react-native run-android» або «react-native run-ios» для React Native CLI.

Потрібно переконатися, що фронтенд правильно налаштований для взаємодії з бекенд API, перевіривши, що всі запити використовують правильний базовий URL. Виконавши тестові запити з фронтенду до бекенд сервера можна перевірити правильність взаємодії.

Для деплою бекенд коду на Node.js потрібно обрати платформу для хостингу, таку як Heroku, AWS або DigitalOcean, і виконати деплой, дотримуючись інструкцій платформи. Наприклад, для Heroku створюється додаток шляхом здійснення «push» на Heroku, і налаштування змінних середовищ (лістинг 3.9).

Лістинг 3.9 – Розгортання на Heroku

```
heroku create  
git push heroku master  
heroku config:set JWT_SECRET=your_jwt_secret  
DB_HOST=your_db_host ...
```

Кінець лістингу 3.9

Для деплою фронтенд коду на React Native потрібно згенерувати APK/IPA файли для мобільних додатків і опублікувати додаток у Google Play Store та Apple App Store, дотримуючись інструкцій відповідних платформ.

Нарешті, потрібно налаштувати моніторинг бекенд сервера за допомогою таких інструментів, як New Relic або Sentry, для відстеження продуктивності та помилок. Можна використовувати Crashlytics для відстеження помилок у мобільному додатку в реальному часі. Це забезпечить стабільну роботу додатка у продуктивному середовищі.

Висновки до розділу 3

Розробка додатку для пошуку загублених тварин включала кілька етапів, спрямованих на забезпечення його функціональності, зручності використання та надійності. Спочатку користувачам пропонується реєстрація та вхід до системи, після чого вони отримують доступ до стрічки з постами про загублених тварин, де можуть залишати коментарі та взаємодіяти з іншими користувачами. Додаток використовує методи HTTP POST та GET для обробки запитів, а також підтримує асинхронну обробку електронних листів за допомогою черг обробки, створених у Redis.

Особлива увага була приділена вибору технологій для зберігання та обробки даних. Реляційна база даних PostgreSQL використовується для основних операцій, тоді як MongoDB застосовується для зберігання коментарів завдяки її високій продуктивності та гнучкості. Створені екрани додатку були розроблені з урахуванням найкращого користувацького досвіду, що включає простий і зрозумілий інтерфейс. Використання технології Docker забезпечує стабільне та ізольоване середовище для запуску додатку, що полегшує його розгортання та масштабування.

Процес тестування включав різні типи тестів, такі як модульне, інтеграційне, системне, а також тестування інтерфейсу. Це дозволило гарантувати високу якість та стабільність роботи додатку, забезпечуючи

користувачам найкращий досвід. Використання інструментів автоматизації тестування, таких як Cypress та Jest з Supertest, сприяло ефективному та швидкому проведенню тестів.

ВИСНОВКИ

В ході написання кваліфікаційної роботи бакалавра було виконано всебічний аналіз існуючих сервісів для пошуку загублених тварин, що дозволило визначити ключові функціональні та нефункціональні вимоги до розроблюваного додатку. Серед досліджених сервісів було виділено такі як PawBoost, Petco Love Lost, Whistle Pet GPS Tracker, Pet FBI та Nextdoor Pet Directory, кожен з яких має свої переваги та недоліки, що були враховані при проектуванні власного рішення.

Обґрунтовано засоби, методи та алгоритми розробки додатку для пошуку загублених тварин. Використано технології React Native та Node.js через їхню здатність забезпечити швидку розробку та високу продуктивність. Реалізовано методи реєстрації користувачів, створення оголошень про зникнення тварин та пошуку.

Розроблена архітектура додатку включає фронтенд компоненти на базі React Native, бекенд компоненти з використанням Node.js та структуру бази даних. Реалізовано крос-платформну сумісність додатку з мобільними операційними системами. Для зберігання даних використовується PostgreSQL та MongoDB.

Створено інтуїтивно зрозумілий та зручний інтерфейс користувача за допомогою React Native. Розроблено прості та зрозумілі екрани, що забезпечують комфортну взаємодію користувача з додатком. Впроваджено сучасні стандарти дизайну та оптимізовані схеми розміщення елементів.

Написано серверний код на Node.js, включаючи розробку API на фреймворку Express.js. Реалізовано асинхронну модель обробки подій для обробки великої кількості одночасних з'єднань. Використано ORM Sequelize для взаємодії з PostgreSQL та Mongoose для MongoDB. Впроваджено JWT токен для безпеки автентифікації.

Проведено комплексне тестування додатку для забезпечення стабільності та надійності. Виконано модульне, інтеграційне та системне тестування за

допомогою Jest, Mocha та Cypress. Виявлено та виправлено помилки на різних етапах розробки, що забезпечило високу якість кінцевого продукту.

Таким чином, реалізований додаток для пошуку загублених тварин є надійним, зручним у використанні та відповідає сучасним вимогам і очікуванням користувачів, що значно підвищує шанси на успішне повернення загублених домашніх улюбленців до їх власників.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PawBoost. Lost and Found Pets Database. URL: <https://www.pawboost.com/> (дата звернення: 05.02.2024).
2. Report and search lost and found pets Petco Love Lost. URL: <https://petcolove.org/lost/> (дата звернення: 05.02.2024).
3. Whistle GPS pet tracker and activity monitor for pets. URL: <https://www.whistle.com/> (дата звернення: 05.02.2024).
4. Pet FBI – Pets Found by Internet. URL: <https://petfbi.org/> (дата звернення: 05.02.2024).
5. Nextdoor pet directory. URL: <https://nextdoor.com/pages/nextdoor-pet-directory-dallas-tx/> (дата звернення: 05.02.2024).
6. Facebook. Загублені та Знайдені Тварини. Луцьк та Волинська обл. URL: <https://www.facebook.com/groups/1126808634017634> (дата звернення: 06.02.2024).
7. Node.js – Run JavaScript everywhere. URL: <https://nodejs.org/en> (дата звернення: 10.03.2024).
8. React Native. Learn once, write anywhere. URL: <https://reactnative.dev/> (дата звернення: 10.03.2024).
9. Visual Studio Code. Code editing redefined. URL: <https://code.visualstudio.com/> (дата звернення: 10.03.2024).
10. Android Studio & App Tools. Android Developers. URL: <https://developer.android.com/studio> (дата звернення: 10.03.2024).
11. Express – Node.js web application framework. URL: <https://expressjs.com/> (дата звернення: 10.03.2024).
12. Tsonev K. Build a complete MVC website with Express. Code Envato Tuts+. URL: <https://code.tutsplus.com/build-a-complete-mvc-website-with-expressjs--net-34168t> (дата звернення: 26.03.2024).
13. Docker: Accelerated Container Application Development. URL: <https://www.docker.com/> (дата звернення: 26.03.2024).

14. MongoDB: the developer data platform. URL: <https://www.mongodb.com/> (дата звернення: 26.03.2024).
15. PostgreSQL. Official website. URL: <https://www.postgresql.org/> (дата звернення: 26.03.2024).
16. Redis – the real-time data platform. URL: <https://redis.io/> (дата звернення: 26.03.2024).
17. Buddy: the DevOps Automation platform. URL: <https://buddy.works/> (дата звернення: 12.04.2024).
18. JSON Web Tokens. URL: <https://jwt.io/> (дата звернення: 12.04.2024).
19. Sequelize. Feature-rich ORM for Modern TypeScript & JavaScript. URL: <https://sequelize.org/> (дата звернення: 12.04.2024).
20. Mongoose ODM. URL: <https://mongoosejs.com/> (дата звернення: 22.04.2024).
21. Application performance monitoring & Error tracking software. URL: <https://sentry.io/welcome/> (дата звернення: 23.04.2024).
22. Jest. URL: <https://jestjs.io/uk/> (дата звернення: 07.05.2024).
23. Mocha – the fun, simple, flexible JavaScript test framework. URL: <https://mochajs.org/> (дата звернення: 07.05.2024).
24. Supertest. URL: <https://www.npmjs.com/package/supertest> (дата звернення: 07.05.2024).
25. Introduction Enzyme. URL: <https://enzymejs.github.io/enzyme/> (дата звернення: 19.05.2024).
26. Testing frameworks for Javascript, Write, Run, Debug. Cypress. URL: <https://www.cypress.io/> (дата звернення: 23.05.2024).