

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем імені проф. І.В.Ельперіна

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»
Директор інституту(декан факультету)

Андрій Форсюк
(ім'я та прізвище)

(підпис)

«04» лютого 2026р.

«До захисту допущено»
Завідувач кафедри

Сергій Грибков
(ім'я та прізвище)

(підпис)

«04» лютого 2026р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

зі спеціальності 122 «Комп'ютерні науки»
(код та назва спеціальності)

освітньо-професійної програми Комп'ютерні науки

на тему: Розроблення мобільного додатку для підтримки догляду за домашніми тваринами

Виконав: здобувач 5 курсу, групи ЗКН 5-2

Максименко Данило Юрійович
(прізвище, ім'я, по батькові повністю)

(підпис)

Керівник Мазуренко Ольга Олександрівна
(прізвище, ім'я та по батькові повністю)

(підпис)

Консультанти _____
(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

Рецензент _____
(ім'я та прізвище)

(підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ – 2026 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем імені проф. І.В.Ельперіна

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

“ 05 ” листопада **2025 року**

З А В Д А Н Н Я**НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА**

Максименко Данило Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення мобільного додатку для підтримки догляду за домашніми тваринами

керівник роботи Мазуренко Ольга Олександрівна

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

затвержені наказом закладу вищої освіти від 05 листопада 2025 р. № 907-кс

2. Строк подання здобувачем роботи 04.02.2026 р.

3. Вихідні дані до роботи

Технічне завдання на розробку мобільного застосунку, Кросплатформний фреймворк Flutter та мова програмування Dart, Хмарна NoSQL база даних Firebase Cloud Firestore.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз предметної області, Проектування архітектури, Обґрунтування технологій, Розробка програмного забезпечення, Тестування та впровадження

5. Перелік графічного матеріалу

Макети інтерфейсу користувача мобільного застосунку, Алгоритм роботи інтелектуального модуля моніторингу стану тварини, Схема логічної структури бази даних

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Мазуренко О. О., старший викладач НУХТ	05.11.2025	15.11.2025
2	Мазуренко О. О., старший викладач НУХТ	21.01.2026	23.01.2026
3	Мазуренко О. О., старший викладач НУХТ	25.01.2026	30.01.2026

7. Дата видачі завдання 05.11.2025 р.**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та існуючих мобільних рішень для догляду за тваринами	07.11.2025-20.11.2025	виконав
2	Розробка функціональних моделей системи	21.11.2025-28.11.2025	виконав
3	Проектування логічної структури бази даних Firestore	29.11.2025-07.12.2025	виконав
4	Розробка інтерфейсу користувача у FlutterFlow	11.12.2025-15.12.2025	виконав
5	Реалізація програмної логіки на мові Dart та інтеграція з Firebase	16.12.2025-20.01.2026	виконав
6	Тестування додатка та оформлення пояснювальної записки	21.01.2026-05.02.2026	виконав
7	Розробка відеопрезентації	5.02.2026	виконав

Здобувач

(підпис)

Максименко Данило

(прізвище та ініціали)

Керівник роботи

(підпис)

Мазуренко Ольга

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційної роботи студента 5-го курсу, заочної ф.з.о.

Максименка Данила Юрійовича, виконаної на тему: «Розроблення мобільного додатку для підтримки догляду за домашніми тваринами»

Київ: кафедра інформаційних технологій, штучного інтелекту і кібербезпеки, 2026 р.

Кваліфікаційна робота присвячена проблемі автоматизації процесів догляду за домашніми тваринами. Наразі сфера PetTech охоплює значний сектор цифрової економіки, що обумовлено зростанням відповідальності власників за здоров'я вихованців. Власники тварин стикаються з викликами, пов'язаними з керуванням великою кількістю інформації про вакцинацію, харчування та медичні огляди. Впровадження сучасних мобільних систем дозволяє оптимізувати ці процеси та забезпечити якісний контроль стану тварин.

Містить у складі три розділи, що логічно пов'язані між собою структурою та ієрархією вимог.

У першому розділі представлено загальну характеристику предметної області, а саме аналіз існуючих мобільних рішень для власників тварин задля оптимізації роботи системи як для користувача, так і для зберігання даних про вихованців.

До другого розділу входить постановка конкретних проблем та шляхів їх вирішення, представлення функціональних вимог до мобільного додатку. Також тут продемонстровано логічну структуру моделі бази даних у Firebase та зв'язки між її елементами.

Третій розділ включає в себе безпосередньо проектування інтерфейсів у середовищі Figma та розробку компонентів мобільного додатку на базі платформи FlutterFlow. Особливу увагу зосереджено на реалізації системи нагадувань та зручності взаємодії користувача з програмою.

Ключові слова: МОБІЛЬНИЙ ЗАСТОСУНОК, ВЕТЕРИНАРНИЙ ДОГЛЯД, БІОМЕТРИЧНІ ПОКАЗНИКИ, NOSQL АРХІТЕКТУРА, КРОСПЛАТФОРМНА РОЗРОБКА, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС

ABSTRACT

Qualification bachelor's thesis of a 5th-year student

Educational and Scientific Institute of Automation and Computer Systems

Maksymenko Danylo Yuriyovych, on the topic: “Development of a mobile application to support pet care”

Kyiv: Department of Information Technology, Artificial Intelligence, and Cybersecurity, 2026

This bachelor's degree project is dedicated to the problem of automating pet care processes. Currently, PetTech covers a significant sector of the digital economy, which is due to the growing responsibility of owners for the health of their pets. Pet owners face challenges related to managing large amounts of information about vaccinations, nutrition, and medical examinations. The introduction of modern mobile systems allows these processes to be optimized and ensures high-quality monitoring of the animals' condition.

The project consists of three sections that are logically linked by the structure and hierarchy of requirements.

The first section provides a general description of the subject area, namely an analysis of existing mobile solutions for pet owners to optimize the system for both users and pet data storage.

The second section covers specific problems and ways to solve them, as well as functional requirements for the mobile app. It also shows the logical structure of the database model in Firebase and the relationships between its elements.

The third section covers the actual design of interfaces in the Figma environment and the development of mobile application components based on the FlutterFlow platform. Particular attention is paid to the implementation of a reminder system and the convenience of user interaction with the program.

Keywords: MOBILE APPLICATION, VETERINARY CARE, BIOMETRIC INDICATORS, NOSQL ARCHITECTURE, CROSS-PLATFORM DEVELOPMENT, USER INTERFACE

ПЕРЕЛІК УМОВИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

БД – База Даних

МД – Мобільний Додаток

ІТ – Інформаційні Технології

ПЗ – Програмне Забезпечення

ПК – Персональний Комп'ютер

ОС – Операційна Система

UML – Уніфікована мова моделювання

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТОВ «БОДО УКРАЇНА» ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ	11
1.1 Загальна характеристика предметної області догляду за домашніми тваринами	11
1.2 Аналіз організаційних процесів догляду за домашніми тваринами в сучасних умовах	13
1.3 Аналіз поточного стану цифрової підтримки догляду за тваринами	15
1.4 Функціональне моделювання та аналіз існуючих процесів	17
1.5 Огляд існуючих рішень для підтримки догляду за тваринами	25
1.6 Обґрунтування доцільності розроблення мобільного додатку для підтримки догляду за домашніми тваринами	28
1.7 Постановка задач	36
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ МОБІЛЬНОГО ДОДАТКУ	39
2.1 Аналіз і специфікація вимог до інформаційної системи	39
2.2 Постановка та алгоритм розв'язання задачі	43
2.3. Моделювання інформаційної системи	45
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ	50
3.1. Інформаційне забезпечення мобільного додатку	50
3.2 Алгоритмізація та реалізація комплексу задач автоматизації	53
3.3. Інструкція користувача	60
3.4 Технічне та системне забезпечення розробки	64
ВИСНОВКИ	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	73

ДОДАТКИ

8

76

ДОДАТОК А

76

ДОДАТОК Б

79

ВСТУП

Сучасний світ неможливо уявити без використання мобільних додатків. Оскільки кількість власників домашніх улюбленців постійно зростає, сектор догляду за тваринами (PetTech) потребує впровадження нових ІТ для моніторингу стану цієї галузі. Здатність ефективно контролювати свою медичну історію, раціон і вакцинацію є важливим компонентом успіху власника. З цієї причини розробка сучасного МД для підтримки догляду за тваринами є надзвичайно важливим завданням.

Оскільки вони вимагають багато часу та схильні до людських помилок, традиційні методи контролю, такі як паперові журнали та пам'ять, стають все менш ефективними. Крім того, запит на кросплатформні рішення, які дозволяють отримувати доступ до даних з будь-якого пристрою, зростає. Створення таких програмних засобів за допомогою платформ із низьким кодуванням дозволяє значно прискорити процес розробки та забезпечити високу якість інтерфейсу.

Метою кваліфікаційної роботи є розробка МД для підтримки догляду за домашніми тваринами, який забезпечить можливість ефективного моніторингу здоров'я та життєдіяльності вихованців. Для досягнення цієї мети необхідно вирішити наступні завдання:

- Провести аналіз існуючих МД у сфері PetTech, визначити їх переваги та недоліки.
- Визначити вимоги до функціоналу та побудувати логічну модель БД.
- Розробити архітектуру та користувацький інтерфейс у середовищі FlutterFlow.

У процесі роботи над буде використано різноманітні методи дослідження, включаючи аналіз літературних джерел, вивчення досвіду існуючих систем, моделювання через UML, проектування інтерфейсів та тестування готового ПЗ. Особлива увага буде приділена використанню хмарних технологій Firebase для

забезпечення надійності збереження даних. Особлива увага буде приділена використанню сучасних технологій і методик для забезпечення високої якості та надійності системи.

Очікувані результати розробки дозволять підвищити якість догляду за тваринами завдяки систематизації медичних даних та автоматизацію нагадувань. Розроблений МД дозволить власникам швидко реагувати на зміни у стані здоров'я вихованців та забезпечить зручний доступ до всієї необхідної інформації в одному місці.

Розробка інтегрованого рішення, яке враховує особливості поведінкових сценаріїв тварин і використовує low-code технології для оптимізації архітектури ПЗ, є науковою новизною роботи. Проект має практичне значення, оскільки він може бути реалізований для широкого кола користувачів, включаючи власників домашніх тварин, щоб покращити культуру відповідальності за своїх вихованців.

Таким чином, розробка МД для підтримки догляду за тваринами є важливим кроком на шляху до створення персоналізованих цифрових асистентів у сфері PetTech.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТОВ «БОДО УКРАЇНА» ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ

1.1 Загальна характеристика предметної області догляду за домашніми тваринами

Догляд за домашніми тваринами є важливою складовою повсякденного життя їхніх власників і охоплює сукупність дій, спрямованих на забезпечення належних умов утримання, підтримання здоров'я та добробуту тварини. На відміну від разових заходів, таких як лікування у ветеринарній клініці, догляд має постійний характер і здійснюється протягом усього життя тварини. Він включає як щоденні рутинні дії, так і періодичні процедури, що виконуються з певною регулярністю.

До основних складових догляду за домашніми тваринами належить годування, яке передбачає не лише вибір відповідного корму, але й дотримання режиму харчування, контроль порцій та доступу до питної води. Особливості годування залежать від виду тварини, її віку, фізіологічного стану, рівня активності та наявності індивідуальних потреб або обмежень.

Важливе місце у догляді займають профілактичні ветеринарні заходи, зокрема вакцинація, дегельмінтизація, обробка від паразитів та планові огляди. Такі процедури проводяться за визначеним графіком і потребують своєчасного виконання.

Не менш значущою складовою є гігієнічний догляд, який включає підтримання чистоти тварини та місця її утримання: догляд за шерстю, кігтями, вухами, очима, а також регулярне прибирання мисок, підстилок або лотків.

Окрему роль відіграє фізична активність та поведінкові потреби тварини. Прогулянки, ігри, тренування та інші форми активності сприяють підтриманню фізичного здоров'я, зниженню рівня стресу та формуванню стабільної поведінки.

Крім того, у процесі догляду важливим є спостереження за станом здоров'я тварини, яке полягає у відстеженні змін у поведінці, апетиті, рівні активності, зовнішньому вигляді та загальному самопочутті.

Процес догляду за домашніми тваринами передбачає участь кількох сторін. Центральне місце у ньому займає власник тварини, який безпосередньо здійснює більшість щоденних дій, приймає рішення щодо харчування, режиму, звернення до фахівців та виконання рекомендацій. Саме від організованості та уважності власника значною мірою залежить якість догляду.

Важливими учасниками процесу є ветеринарні клініки та спеціалісти, які здійснюють медичне обслуговування тварин, проводять профілактичні та лікувальні заходи, а також надають консультації щодо догляду, харчування та умов утримання.

До процесу догляду також залучені зоомагазини та постачальники товарів для тварин, які забезпечують власників кормами, медикаментами, засобами гігієни та аксесуарами.

Окрему групу учасників становлять сервіси вигулу, грумінгу та тимчасового догляду, які частково перебирають на себе виконання певних процедур або забезпечують догляд у ситуаціях, коли власник не має змоги зробити це самостійно.

Сфера догляду за домашніми тваринами характеризується значною кількістю різнорідних дій, що виконуються з різною періодичністю. Частина з них здійснюється щодня, інші — раз на тиждень, місяць або рік. При цьому вимоги до догляду можуть змінюватися залежно від віку тварини, стану її здоров'я та зовнішніх умов. Інформація, пов'язана з доглядом, часто зберігається у різних формах: у ветеринарних паспортах, записах власника, рекомендаціях лікаря або пам'яті людини, що ускладнює її систематизацію.

Регулярність виконання процедур є характерною особливістю догляду за домашніми тваринами. Багато дій потребують чіткого дотримання графіка, оскільки їх несвоєчасне виконання може призвести до небажаних наслідків. У зв'язку з цим у процесі догляду важливу роль відіграє своєчасне виконання запланованих дій та контроль за їх повторюваністю, що особливо актуально в умовах сучасного темпу життя власників тварин.

1.2 Аналіз організаційних процесів догляду за домашніми тваринами в сучасних умовах

Процес догляду за домашніми тваринами в сучасних умовах не є ізольованою або однотипною діяльністю. Він формується як сукупність взаємопов'язаних дій, що виконуються у різний час, з різною частотою та за участю кількох сторін. Організація догляду залежить від виду тварини, умов її утримання, способу життя власника, а також доступності зовнішніх сервісів. У зв'язку з цим процес догляду доцільно розглядати не як окремі дії, а як послідовність і взаємодію процесів, що мають власну логіку виконання.

Аналіз організаційних процесів дозволяє виділити основні етапи догляду, визначити їх взаємозалежність та зафіксувати роль учасників у кожному з них. Такий підхід є необхідним для подальшого формалізованого подання процесу в межах функціонального моделювання.

Узагальнено процес догляду за домашніми тваринами можна подати як послідовність повторюваних і періодичних дій, які виконуються протягом усього періоду утримання тварини. Ці дії умовно поділяються на кілька груп залежно від їх призначення та частоти виконання.

До щоденних процесів належать дії, пов'язані з базовими потребами тварини: годування, забезпечення доступу до води, прогулянки або активні ігри, візуальний огляд стану тварини, підтримання чистоти місця утримання. Такі процеси виконуються регулярно та зазвичай інтегровані у повсякденний розпорядок власника.

Періодичні процеси охоплюють дії, що виконуються з визначеною, але меншою частотою. До них належать гігієнічні процедури (купання, догляд за шерстю, підстригання кігтів), профілактичні обробки від паразитів, контроль маси тіла, оновлення запасів корму та засобів догляду. Частота виконання таких процесів залежить від індивідуальних особливостей тварини та рекомендацій спеціалістів.

Окрему групу становлять планові ветеринарні процеси, зокрема вакцинація, профілактичні огляди, діагностичні обстеження та контрольні

візити до ветеринарної клініки. Ці процеси зазвичай прив'язані до конкретних дат або часових інтервалів і потребують попередньої підготовки, зокрема запису до клініки та врахування рекомендацій лікаря.

Крім того, у процесі догляду можливі ситуаційні або непередбачувані процеси, пов'язані зі змінами стану здоров'я тварини, травмами або поведінковими порушеннями. У таких випадках стандартний розпорядок догляду коригується відповідно до нових обставин.

Між зазначеними групами процесів існує тісний взаємозв'язок. Результати щоденних спостережень можуть впливати на рішення щодо звернення до ветеринарного спеціаліста, а рекомендації ветеринара — змінювати режим годування, активності або гігієнічного догляду. Таким чином, процес догляду характеризується циклічністю, повторюваністю та залежністю окремих дій одна від одної.

У сучасних умовах саме власник тварини виконує центральну організаційну роль у процесі догляду. Він є ініціатором більшості дій, координує взаємодію з іншими учасниками та відповідає за дотримання режиму утримання тварини. На практиці власник поєднує кілька ролей одночасно: виконавця щоденних процедур, контролера виконання періодичних заходів і посередника у взаємодії з ветеринарними та сервісними організаціями.

Власник самостійно визначає час і порядок виконання щоденних дій, планує періодичні процедури, фіксує рекомендації ветеринарного лікаря та приймає рішення щодо коригування режиму догляду. У разі залучення сторонніх сервісів (вигул, грумінг, перетримка) саме власник організовує передачу інформації про потреби тварини та контролює виконання домовленостей.

Організація догляду значною мірою залежить від особистих чинників власника, таких як графік роботи, рівень поінформованості, досвід утримання тварин та наявність часу. У повсякденній практиці значна частина інформації щодо догляду зберігається у неформалізованому вигляді: у пам'яті власника, паперових записах, повідомленнях або документах з ветеринарної клініки.

Відсутність єдиного підходу до фіксації та впорядкування цієї інформації ускладнює контроль за виконанням процедур, особливо у випадках, коли догляд включає велику кількість періодичних дій.

Таким чином, власник виступає ключовою ланкою, через яку проходять усі основні процеси догляду: від щоденного обслуговування тварини до взаємодії з зовнішніми учасниками. Його дії, рішення та організаційні можливості безпосередньо впливають на послідовність і своєчасність виконання процесів догляду, що робить роль власника визначальною у загальній схемі функціонування цієї предметної області.

1.3 Аналіз поточного стану цифрової підтримки догляду за тваринами

У сучасних умовах цифрові засоби поступово проникають у сферу догляду за домашніми тваринами, однак їх використання має фрагментарний характер і значною мірою залежить від особистих звичок власника. На практиці не існує єдиного усталеного підходу до ведення обліку процедур догляду, що призводить до застосування різних, часто не пов'язаних між собою інструментів. Аналіз поточного стану цифрової підтримки показує, що власники тварин поєднують як цифрові, так і недиджиталізовані способи фіксації інформації.

Поширеним способом організації інформації залишаються паперові носії. До них належать ветеринарні паспорти, блокноти, окремі записи на аркушах або друковані рекомендації ветеринарного лікаря. У таких матеріалах зазвичай фіксуються дати вакцинацій, обробок, призначення лікаря або результати обстежень. Паперові записи зручні для одноразового використання, однак вони зберігаються розрізнено, можуть втрачатися або залишатися недоступними у потрібний момент, наприклад, під час візиту до іншої клініки.

Іншим поширеним інструментом є календарі, як паперові, так і електронні. Власники часто використовують календар для позначення дат вакцинацій, планових візитів до ветеринара або необхідності повторних процедур. У деяких випадках застосовуються стандартні календарні застосунки

на смартфоні з базовими нагадуваннями. Водночас такі записи зазвичай не містять детальної інформації про саму процедуру, її результати або пов'язані рекомендації, що обмежує їх використання як повноцінного інструменту обліку догляду.

Значна частина інформації, пов'язаної з доглядом за тваринами, зберігається в пам'яті власника. Це стосується режиму годування, частоти прогулянок, останніх гігієнічних процедур, реакцій тварини на корм або ліки. Такий спосіб обліку є найбільш неформалізованим і повністю залежить від уваги, завантаженості та досвіду людини. За умови зміни розпорядку дня, стресових ситуацій або тривалого періоду між процедурами інформація може частково втрачатися або спотворюватися.

Окрему групу становлять мобільні застосунки та цифрові сервіси, які власники використовують для підтримки окремих аспектів догляду. Це можуть бути застосунки для ведення календаря щеплень, трекери активності, нагадування про годування або нотатники для збереження інформації. Часто такі інструменти орієнтовані на одну функцію і не охоплюють увесь процес догляду. У результаті власник змушений використовувати кілька різних застосунків одночасно або поєднувати їх із паперовими записами та власною пам'яттю.

Аналіз показує, що поточний стан цифрової підтримки догляду за тваринами характеризується розрізненістю джерел інформації та відсутністю єдиного підходу до її структурування. Дані про тварину можуть зберігатися у різних форматах і місцях: частково у ветеринарному паспорті, частково в календарі, частково в мобільному застосунку або у вигляді неформальних нотаток. Це ускладнює формування цілісного уявлення про стан догляду, особливо у випадках, коли необхідно швидко відтворити історію процедур або передати інформацію іншому учаснику процесу.

Таким чином, у межах поточного стану цифрової підтримки догляду за домашніми тваринами спостерігається використання різних інструментів без чіткої інтеграції між ними. Це створює підґрунтя для подальшого аналізу

існуючих процесів догляду в їх фактичному вигляді та формалізованого подання у вигляді функціональної моделі AS-IS, що буде розглянуто у наступних підрозділах.

1.4 Функціональне моделювання та аналіз існуючих процесів

Здійснено функціональне моделювання та аналіз існуючих процесів.

Функція (A-0): Організувати та виконувати догляд за домашньою твариною в умовах повсякденного життя власника.

Input (входи):

- Потреби тварини (харчування, вода, активність, гігієна, здоров'я).
- Поточний стан тварини (самопочуття, апетит, активність, зовнішні прояви).
- Інформація від зовнішніх учасників (рекомендації ветеринара, записи на процедури, інструкції до препаратів/кормів).
- Наявні ресурси (корм, ліки, гігієнічні засоби, час власника).

Control (керування):

- Режим дня та можливості власника (графік роботи/сім'ї).
- Ветеринарні правила/графіки (дати вакцинацій, інтервали обробок).
- Інструкції до препаратів і кормів (частота, дозування, умови).
- Умови утримання (квартира/будинки, наявність виходу, сезонність).

Output (виходи):

- Виконані дії догляду (годування, прогулянки, гігієна, обробки, візити).
- Поточний стан тварини та зміни (спостереження власника).
- Фрагменти обліку/фіксації (відмітки в календарі, записи в блокноті, записи в застосунках, ветпаспорт, фото/файли).
- Взаємодія з сервісами (запис/оплата/результат послуги).

Mechanism (механізми):

- Власник (ключовий виконавець і координатор).
- Ветеринарна клініка/лікар.
- Зоомагазини/постачальники.
- Сервіси виходу/грумінгу/перетримки.

- Інструменти обліку: пам'ять, паперові нотатки, календар, месенджери, розрізнені застосунки.

Опис AS-IS (як відбувається зараз):

реальному житті догляд відбувається як поєднання рутини (щоденні дії) і подій, прив'язаних до дат (вакцинації, обробки, огляди), а також ситуативних рішень при зміні стану тварини. Власник одночасно: виконує щоденні дії, пам'ятає/планує майбутні процедури, зберігає документи і рекомендації, контактує з клініками та сервісами. Інформація про догляд існує у кількох формах: частина — у ветпаспорті, частина — у календарі, частина — у нотатках або застосунках, частина — лише в пам'яті власника.

Мета: розкласти контекстну функцію на логічні підпроцеси (що потім зручно переноситься в “блоки” AS-IS у 1.4.1).

Функція (A0): *Виконання догляду через набір підпроцесів.*

Основні підфункції AS-IS (A1–A6):

- **A1. Виконувати щоденні потреби тварини** (годування, вода, прогулянка/ігри, базова чистота).
- **A2. Виконувати гігієнічні процедури** (шерсть, кігті, вуха/очі, місце утримання).
- **A3. Забезпечувати профілактику і ветеринарні події** (вакцинації, обробки, огляди, аналізи).
- **A4. Спостерігати стан здоров'я та реагувати на зміни** (симптоми, рішення про звернення).
- **A5. Забезпечувати ресурси для догляду** (покупки корму/препаратів/засобів, контроль запасів).
- **A6. Фіксувати інформацію та координувати взаємодію з сервісами** (записи, документи, календар, грумінг/вигул/клініка).

Input: потреби тварини, стан тварини, рекомендації/призначення, ресурси.

Control: графіки процедур, режим власника, інструкції, умови утримання.

Output: виконані дії, оновлений стан тварини, записи/документи, взаємодії із

сервісами.

Mechanism: власник + зовнішні учасники + інструменти обліку.

Опис AS-IS: Реальний догляд зазвичай не йде “ідеальною послідовністю”. Власник робить щоденну рутину майже автоматично, а періодичні дії часто “вбудовані” в життя як окремі події (наприклад, згадати про обробку, коли з’явився сезон кліщів; згадати про вакцинацію, коли знаходиться ветпаспорт або коли ветеринар нагадав). Фіксація інформації може бути або мінімальною (тільки ветпаспорт), або розкиданою по різних місцях (календар + нотатки + фото призначень).

AS-IS модель 3 (A1). Щоденний цикл догляду

Мета: максимально чітко описати щоденну рутину як “цикл”, де видно входи/виходи, що буде основою для AS-IS блоків і потоків.

Функція (A1): *Виконати щоденні дії догляду та зафіксувати базові спостереження.*

Input:

- Поточні потреби (їжа, вода, активність, туалет/чистота).
- Стан тварини в конкретний день (апетит, настрій, активність).
- Наявні ресурси (корм, вода, наповнювач, аксесуари для прогулянки).

Control:

- Режим дня власника і тварини (час годування/вигулу).
- Обмеження/дієта (якщо є рекомендації або особливості).
- Домашні правила (де можна/не можна, безпека).

Output:

- Виконані дії (погодували, дали воду, вигуляли/пограли, прибрали).
- Набір спостережень (норма/не норма: “з’їв менше”, “млявий”, “пив більше”, “не хотів гуляти”).
- Неформальні відмітки (або відсутність відміток).

Mechanism:

- Власник (основний виконавець).
- Побутові засоби (миски, корм, поводок, лоток).

- Інструменти фіксації (часто — пам'ять; інколи — нотатки/застосунок).

Детальний опис AS-IS (покроково, як це виглядає в житті):

1. **План на день** найчастіше існує не як документ, а як звичка (“годуємо зранку/ввечері”).
2. **Годування і вода:** власник орієнтується на звичну порцію, іноді коригує “на око” (за апетитом/залишками).
3. **Активність:** прогулянка/ігри відбуваються за можливістю власника; у квартирних умовах активність може компенсуватися іграми.
4. **Чистота:** прибирання лотка/місця відбувається регулярно, але частота залежить від режиму й навантаження.
5. **Спостереження:** власник помічає відхилення (апетит, стілець, поведінка), але фіксує їх у кращому випадку у вигляді короткої нотатки/повідомлення/фото; часто — просто “тримає в голові”.
6. **Передача інформації:** якщо є кілька людей у сім'ї, інформація може передаватися усно (“я вже годувала/не годувала”), або через месенджер.

AS-IS модель 4 (A3). Планові профілактичні та ветеринарні події

Мета: показати календарні/регламентні дії (вакцинації, обробки, огляди) як ланцюг “планування → виконання → збереження результатів”.

Функція (A3): *Організувати та виконати планові профілактичні/ветеринарні процедури.*

Input:

- Ветпаспорт, попередні дати вакцинацій/обробок (якщо збережені).
- Рекомендації ветеринара, результати аналізів, призначення.
- Наближення “потрібної дати” (через календар, пам'ять або нагадування клініки).

Control:

- Медичні графіки (інтервали, ревакцинації).
- Правила підготовки до процедур (умови, обмеження).
- Доступність клініки/лікаря, робочий час, запис.

Output:

- Проведена процедура (вакцинація/обробка/огляд).
- Документ/підтвердження (запис у ветпаспорті, чек, рекомендації).
- Наступна контрольна дата (часто повідомляється усно або в документі).

Mechanism:

- Власник (організатор і той, хто приводить тварину/дає препарат).
- Ветеринар/клініка.
- Засоби: препарат/вакцина, транспорт.
- Інструменти фіксації: ветпаспорт, календар, фото призначень, месенджер.

Детальний опис AS-IS (типовий сценарій):

1. **Згадування про подію** відбувається одним із способів: власник сам пам'ятає; бачить запис у календарі; знаходить ветпаспорт; отримує нагадування/повідомлення від клініки (якщо клініка таке практикує).
2. **Підготовка:** з'ясувати, що саме робити (який препарат/яка процедура), перевірити наявність препарату або записатися в клініку.
3. **Виконання:** або в клініці, або вдома (наприклад, частина обробок).
4. **Отримання результатів/рекомендацій:** лікар може дати паперові рекомендації, надіслати в месенджері, або продиктувати.
5. **Фіксація:** найчастіше — відмітка в ветпаспорті; додатково власник може зробити фото призначень, додати нотатку, створити подію в календарі на майбутню дату.
6. **Збереження історії:** зберігається “як вийде”: паспорт + окремі фото + записи; єдиного формату зазвичай немає.

AS-IS модель 5 (A4 + A6). Реагування на зміни стану + облік і комунікації

Мета: зафіксувати, як у реальності відбувається “нестандартний” догляд: симптоми → рішення → звернення → виконання призначень → фіксація інформації, і як при цьому ведеться облік (часто розрізнено).

Функція (A4/A6): *Виявити зміни стану, організувати звернення/догляд та зафіксувати інформацію доступними засобами.*

Input:

- Спостереження власника (симптоми, зміни поведінки).
- Попередня історія (якщо власник її пам'ятає або має записи/фото).
- Контакти клініки/ветеринара, поради з минулого.

Control:

- Невідкладність ситуації (терміново/можна чекати).
- Доступність клінік/час власника.
- Рекомендації ветеринара, схеми лікування, дозування.

Output:

- Прийняте рішення (домашнє спостереження / консультація / візит).
- Проведені дії (візит, лікування, зміна раціону/режиму).
- “Сліди” інформації: повідомлення в месенджері, фото аналізів, записи, чеки.
- Повторні дати (контрольний огляд, повторний аналіз, курс ліків).

Mechanism:

- Власник (спостерігач, виконавець призначень).
- Ветеринар/клініка (діагностика, призначення).
- Канали зв'язку: телефон, месенджери, сайт/запис.
- Засоби фіксації: камера телефону, нотатки, календар, папка з документами.

Детальний опис AS-IS (типова реальна “гілка подій”):

1. **Виявлення змін:** власник помічає симптом (не їсть, млявий, чухається, блювання тощо).
2. **Початкове рішення:** або спостерігати, або одразу звернутися. Часто рішення базується на досвіді власника та доступності ветеринара.
3. **Комунікація:** консультація телефоном/месенджером, або запис на прийом. Інформація передається словами, інколи фото/відео.

4. **Отримання призначень:** може бути паперовий лист, фото, повідомлення в чаті, усні інструкції.
5. **Виконання:** власник організовує ліки, режим, спостереження, інколи діє за таймером/нагадуванням телефону, інколи — “по відчуттю” і пам’яті.
6. **Фіксація перебігу:** якщо власник веде записи — це можуть бути короткі нотатки (температура, реакція на ліки), або просто запам’ятовування “стало краще/гірше”.
7. **Контрольні події:** повторний прийом/аналіз. Їх дата може бути записана у різний спосіб: в календар, у чат, на папір, або залишитися лише як усна домовленість.

Проблема 1. Фрагментарність обліку інформації про догляд за твариною

Де проявляється в моделях:

- **A1** — щоденні дії та спостереження фіксуються неформально або не фіксуються взагалі;
- **A3** — результати профілактичних і ветеринарних процедур зберігаються у різних формах (ветпаспорт, фото, повідомлення);
- **A4/A6** — інформація про симптоми, призначення та перебіг лікування розміщується у різних каналах (месенджери, паперові записи, пам’ять).

Наслідки для процесу: Інформація про стан здоров’я та історію догляду за твариною розподіляється між кількома носіями та не формує єдиного цілісного уявлення. Відтворення повної картини догляду потребує додаткового часу або може бути ускладнене через втрату або неповноту даних.

Проблема 2. Залежність виконання процедур від пам’яті та суб’єктивної уважності власника

Де проявляється в моделях:

- **A1** — щоденні дії виконуються на основі звички без формального підтвердження виконання;
- **A3** — згадування про профілактичні процедури часто базується на пам’яті або випадковому виявленні інформації;

- **A4** — спостереження за симптомами та їх динамікою зберігаються переважно у пам'яті власника.

Наслідки для процесу: Процес догляду стає вразливим до людського фактора, зокрема зайнятості, стресу або зміни режиму дня. Частина інформації може бути втрачена або спотворена, що ускладнює контроль за регулярністю процедур та відстеження змін у стані тварини.

Проблема 3. Відсутність єдиного хронологічного представлення подій догляду

Де проявляється в моделях:

- **A3** — події вакцинацій, обробок та оглядів зберігаються як окремі записи без загальної часової структури;
- **A4/A6** — інформація про лікування та повторні візити не завжди пов'язана з попередніми подіями догляду.

Наслідки для процесу: Власнику складно швидко відтворити послідовність виконаних процедур або встановити зв'язок між попередніми подіями та поточним станом тварини. Це ускладнює аналіз історії догляду та передачу інформації ветеринарному спеціалісту.

Проблема 4. Ускладнена координація між власником та зовнішніми учасниками процесу

Де проявляється в моделях:

- **A3** — передача рекомендацій ветеринара здійснюється у різних формах (усно, паперово, через месенджери);
- **A6** — взаємодія з сервісами виходу, грумінгу або перетримки не має стандартизованого формату обміну інформацією.

Наслідки для процесу: Інформація про потреби тварини, призначення або обмеження може передаватися неповно або втрачатися між етапами взаємодії. Це ускладнює контроль виконання рекомендацій та повторне використання інформації у майбутньому.

Проблема 5. Відсутність системного контролю регулярності процедур

Де проявляється в моделях:

- **A1** — щоденні дії не мають формалізованого підтвердження виконання;
- **A3** — періодичні процедури залежать від нагадувань календаря або пам'яті;
- **A4** — виконання курсів лікування контролюється неформально.

Наслідки для процесу: Контроль повторюваності процедур здійснюється нерівномірно, що ускладнює підтримання стабільного режиму догляду. У деяких випадках власник не має швидкого способу перевірити, чи була виконана конкретна дія і коли саме.

Проблема 6. Відсутність інтеграції між різними інструментами обліку

Де проявляється в моделях:

- **A1** — поєднання пам'яті, паперових записів та окремих цифрових засобів;
- **A3** — використання календаря окремо від документів і призначень;
- **A6** — збереження файлів, фото та повідомлень у різних середовищах.

Наслідки для процесу: Для отримання повної інформації про догляд за твариною власнику необхідно звертатися до кількох джерел, що збільшує часові витрати та підвищує ймовірність пропуску важливих даних.

Проблема 7. Складність ведення історії догляду для кількох тварин або кількох відповідальних осіб

Де проявляється в моделях:

- **A1** — щоденні дії виконуються різними членами родини без єдиного обліку;
- **A4/A6** — інформація передається усно або через месенджери без фіксації в загальній системі.

Наслідки для процесу: Зростає ризик дублювання або пропуску дій, особливо у випадках, коли догляд здійснюють кілька осіб або коли у власника є більше однієї тварини. Відсутність централізованого обліку ускладнює узгодження дій між учасниками.

1.5 Огляд існуючих рішень для підтримки догляду за тваринами

У межах огляду ринку були розглянуті поширені типи цифрових рішень, які власники домашніх тварин використовують для організації догляду: застосунки-щоденники та трекари процедур, застосунки ветеринарних сервісів, а також сервіси “догляд на замовлення” (вигул/перетримка). Виявлено, що більшість рішень орієнтовані на окремі підзадачі (наприклад, нагадування про процедури або бронювання послуг) і рідко охоплюють увесь процес догляду як єдину послідовність дій.

Одним із відомих прикладів застосунків-трекерів є 11pets: Pet Care. За відкритими описами, у ньому передбачені нагадування про ліки та події, можливість зберігати записи про вакцинації, відстежувати показники на кшталт ваги та зберігати документи, а також ділитися даними про тварину.

Попри широкий набір базових функцій, такі рішення часто залишаються “персональним щоденником” власника: інформація ведеться в межах одного застосунку, але взаємодія з іншими учасниками догляду (ветклініка, грумінг, вигул) зазвичай зводиться до ручного обміну повідомленнями або файлами. Крім того, власник змушений самотійно підтримувати актуальність даних і структуру записів, що не завжди узгоджується з реальною побутовою поведінкою, описаною в AS-IS моделях.

Окрему групу становлять застосунки, пов’язані з ветеринарними організаціями та записом на прийом. Прикладом є PetDesk, який позиціонується як рішення для взаємодії власника з постачальниками послуг: у відкритих описах згадуються запити на запис, нагадування, збереження даних про візити/історію та запити на поповнення препаратів.

На ринку також представлені рішення, зосереджені на окремих аспектах безпеки та активності тварини, наприклад GPS-трекери з аналітикою активності. До таких належить Tractive, який у відкритих описах наголошує на відстеженні місця розташування, “віртуальних меж”, моніторингу активності/сну та сповіщеннях про зміни.

Однак рішення цього класу переважно працюють у зв'язці з фізичним пристроєм і підпискою та не призначені для ведення повної історії догляду (вакцинації, призначення, процедури, документи). Тобто вони корисні як додатковий інструмент для конкретної потреби, але не замінюють комплексну підтримку процесу догляду.

Окремо слід виділити сервіси вигулу та перетримки, які вирішують задачу залучення виконавців догляду на час відсутності власника. Наприклад, Rover орієнтований на бронювання послуг догляду та комунікацію із виконавцями; у відкритих матеріалах згадуються бронювання, повідомлення, фотозвіти, GPS-маршрут прогулянки, а також елементи звітності під час вигулу.

Разом з тим сервіси такого типу не є інструментами управління всім циклом догляду: вони допомагають організувати виконання окремої послуги, але не замінюють систематизацію медичних даних, планування профілактики, фіксацію симптомів та ведення історії процедур.

Таким чином, огляд показує наявність різних класів рішень: трекари догляду, ветеринарні платформи, GPS/активність-сервіси та маркетплейси послуг. Кожен клас частково покриває потреби власників, однак у реальному використанні власники часто поєднують кілька інструментів одночасно, а частина інформації продовжує зберігатися поза застосунками (у календарі, нотатках, документах або пам'яті), що відповідає опису поточного стану (розділ 1.3).

Застосунки типу трекерів догляду (наприклад, 11pets) надають найближчий до “журналу” набір можливостей: нагадування, записи про події та збереження документів. Однак у практичному процесі догляду, описаному в AS-IS моделях, значна частина дій не обмежується лише фіксацією подій: присутня координація із зовнішніми учасниками, передача рекомендацій, а також щоденні спостереження, які часто залишаються поза системною структурою або вимагають дисципліни постійного ручного внесення даних. Це означає, що навіть за наявності трекера користувачі можуть продовжувати паралельно користуватися календарем, нотатками та пам'яттю

Застосунки, орієнтовані на взаємодію з ветеринарними установами (наприкладі PetDesk), розв'язують іншу частину задач: спрощують контакт з провайдером послуг, підтримують нагадування та організацію візитів. Водночас вони не завжди покривають повний обсяг домашніх процесів догляду (A1) і можуть зосереджуватися на “подіях клініки”, залишаючи поза увагою рутину, спостереження та багато періодичних процедур, які власник виконує самостійно.

Рішення класу GPS/активність-моніторингу (наприкладі Tractive) підсилюють контроль безпеки та поведінкових показників, надаючи власнику дані про локацію та активність, а також сповіщення про нетипові зміни. Проте цей клас рішень не відповідає задачі ведення історії догляду та планування медичних/профілактичних подій (A3) і не замінює облік процедур та документів.

Сервіси виходу/перетримки (наприкладі Rover) забезпечують організацію виконання окремих послуг і дають власнику звітність щодо конкретної послуги (наприклад, маршрут прогулянки, повідомлення, фотооновлення). Однак вони не є інструментами управління повним циклом догляду, оскільки не призначені для збереження та аналізу медичних даних, графіків профілактики або комплексної структури щоденного догляду.

Отже, розглянуті рішення відрізняються фокусом і здебільшого реалізують підтримку окремих блоків процесу (A1 або A3 або частково A6), тоді як у реальній практиці догляд вимагає одночасної роботи з рутиною, плановими подіями, реагуванням на симптоми та накопиченням історії. Саме ця багатокomпонентність і пояснює, чому наявні рішення повністю або частково не закривають потреби, зафіксовані у функціональних моделях AS-IS.

1.6 Обґрунтування доцільності розроблення мобільного додатку для підтримки догляду за домашніми тваринами

Проведений аналіз процесів догляду за домашніми тваринами та їх поточного стану показує, що основні труднощі пов'язані не з відсутністю

окремих інструментів, а з організацією догляду в цілому. Власник тварини змушений поєднувати щоденні рутинні дії, планові ветеринарні процедури, ситуативні рішення у разі погіршення стану тварини та облік великої кількості інформації. При цьому дані про догляд зазвичай зберігаються у різних формах і не утворюють єдиного цілісного простору.

Для підтримки таких процесів можуть використовуватися різні підходи: паперові записи, стандартні календарі, веб-сервіси або готові мобільні застосунки. Однак аналіз показує, що саме мобільний додаток є найбільш доцільною формою реалізації рішення. Це пояснюється тим, що догляд за твариною відбувається у повсякденних умовах і не прив'язаний до конкретного місця чи часу. Більшість дій виконуються вдома, на прогулянці або під час поїздки, і саме мобільний телефон у таких ситуаціях є найбільш доступним інструментом.

На відміну від веб-рішень, які потребують окремого доступу через браузер і стабільного з'єднання з мережею, мобільний додаток дозволяє швидко переглянути необхідну інформацію, зафіксувати подію або отримати нагадування у будь-який момент. Особливо це важливо для процедур, що мають часову прив'язку, зокрема вакцинацій, обробок, прийому препаратів або контрольних візитів до ветеринарного лікаря. Використання стандартних календарів або нотаток у таких випадках не завжди забезпечує достатній зв'язок між подією та контекстом догляду.

Огляд існуючих мобільних додатків і сервісів показав, що більшість з них орієнтовані на розв'язання окремих завдань. Частина застосунків виконує роль цифрового щоденника, інші зосереджені на взаємодії з ветеринарними клініками, контролі активності або організації послуг вихову й перетримки. У реальному житті власники часто використовують кілька таких інструментів одночасно, доповнюючи їх паперовими записами або власною пам'яттю. Це свідчить про те, що навіть за наявності готових рішень процес догляду залишається розрізненим.

Можна припустити, що з окремими недоліками готових рішень власник може змиритися. Проте аналіз AS-IS моделей показує, що ці недоліки мають системний характер і безпосередньо пов'язані з організацією процесу догляду. Готові застосунки зазвичай мають фіксовану логіку роботи та обмежені можливості адаптації до реального сценарію дій власника, який включає поєднання рутини, планових подій і непередбачуваних ситуацій. У таких умовах власник змушений підлаштовувати власний процес догляду під можливості інструмента, а не навпаки.

Самостійна розробка мобільного додатку дозволяє побудувати рішення, яке виходить безпосередньо з виявлених процесів та проблем, а не з функціональних обмежень уже існуючих сервісів. Це створює можливість поєднати в одному середовищі облік щоденних дій, планування та контроль періодичних процедур, фіксацію рекомендацій ветеринарного лікаря та накопичення історії догляду за твариною. Крім того, власна розробка дає змогу гнучко змінювати структуру та функціональні можливості додатку відповідно до реальних потреб користувачів.

Таким чином, доцільність розроблення мобільного додатку зумовлена характером процесу догляду за домашніми тваринами, його мобільністю, потребою в оперативному доступі до інформації та необхідністю підтримки регулярності процедур. Сукупність цих чинників обґрунтовує вибір мобільного додатку як оптимальної форми реалізації інформаційної системи та підтверджує доцільність його самостійної розробки.

Функціональна модель процесу догляду у статусі TO-BE відображає цільовий стан організації догляду за домашніми тваринами з використанням мобільного додатку як єдиного інструмента підтримки процесу. Модель побудована на основі виявлених проблем поточного стану (AS-IS) та передбачає впорядкування інформаційних потоків, зменшення залежності від пам'яті власника та забезпечення цілісного обліку процедур догляду.

Функція (A-0): Підтримувати та координувати процес догляду за домашньою твариною з використанням мобільного додатку як єдиного інформаційного середовища.

Input (входи):

- Дані про тварину (вид, вік, особливості, стан здоров'я).
- Події догляду (щоденні дії, планові процедури, ветеринарні призначення).
- Інформація від зовнішніх учасників (рекомендації лікаря, дані сервісів).

Control (керування):

- Налаштовані графіки процедур і нагадувань.
- Правила та рекомендації з догляду.
- Режим дня та доступність власника.

Output (виходи):

- Зафіксовані та виконані дії догляду.
- Актуальна історія догляду та стану тварини.
- Структурована інформація для взаємодії з ветеринаром і сервісами.

Mechanism (механізми):

- Власник тварини.
- Мобільний додаток як центральний інструмент.
- Ветеринарні клініки та сервісні організації.

Сутність змін порівняно з AS-IS: На відміну від поточного стану, де інформація розподілена між різними носіями, у TO-BE моделі мобільний додаток виступає єдиним центром збереження, планування та контролю процесу догляду.

Функція (A0): Організувати процес догляду як узгоджену систему взаємопов'язаних підпроцесів.

Підпроцеси TO-BE:

- A1. Планування та виконання щоденних дій догляду.
- A2. Управління гігієнічними процедурами.

- A3. Планування та контроль ветеринарних і профілактичних заходів.
- A4. Моніторинг стану тварини та фіксація змін.
- A5. Облік ресурсів і матеріалів для догляду.
- A6. Ведення централізованої історії догляду та комунікація.

Сутність змін: Кожен підпроцес має чітке місце в загальній схемі догляду і реалізується через функціональні можливості мобільного додатку, що усуває розрізненість дій і джерел інформації, зафіксовану в AS-IS.

Функція (A1): Планувати та фіксувати виконання щоденних дій догляду.

Input:

- Заплановані щоденні дії (годування, вода, активність, прибирання).
- Поточні спостереження за станом тварини.

Control:

- Налаштований режим дня.
- Індивідуальні рекомендації щодо догляду.

Output:

- Підтвердження виконання дій.
- Зафіксовані короткі спостереження.

Mechanism:

- Власник.
- Мобільний додаток (інтерфейс відміток і нотаток).

Усунення проблем AS-IS: Замість опори на пам'ять власника щоденні дії отримують просту фіксацію, що дозволяє бачити факт виконання та зберігати мінімальну історію подій без додаткового навантаження.

Функція (A3): Забезпечити планування, виконання та збереження результатів ветеринарних і профілактичних процедур.

Input:

- Дані про попередні процедури.
- Рекомендації ветеринара.

Control:

- Медичні графіки та інтервали.
- Налаштовані нагадування.

Output:

- Зафіксовані процедури.
- Наступні контрольні дати.
- Структуровані призначення.

Mechanism:

- Власник.
- Ветеринар.
- Мобільний додаток.

Усунення проблем AS-IS: Події, які раніше існували у вигляді розрізнених записів, формуються в єдину хронологічну послідовність, доступну для перегляду та передачі інформації.

Функція (A4/A6): Фіксувати зміни стану тварини та забезпечувати цілісне збереження історії догляду.

Input:

- Спостереження власника.
- Результати консультацій і лікування.

Control:

- Призначені схеми догляду або лікування.
- Налаштовані контрольні події.

Output:

- Хронологічна історія стану тварини.
- Актуальні дані для взаємодії з ветеринаром або сервісами.

Mechanism:

- Власник.
- Мобільний додаток як єдине сховище даних.

Усунення проблем AS-IS: Інформація про симптоми, лікування та рекомендації більше не зберігається у різних каналах, а формується як цілісний

журнал догляду, що зменшує втрату даних і спрощує повторне використання інформації.

Функціональна модель ТО-ВЕ демонструє перехід від фрагментарного, неформалізованого процесу догляду до впорядкованої системи, де мобільний додаток виконує роль центрального координатора. Така модель дозволяє зменшити залежність від людського фактора, забезпечити прозорість виконання процедур та створити основу для подальшої реалізації інформаційної системи.

Економічний ефект від впровадження мобільного додатку для підтримки догляду за домашніми тваринами доцільно розглядати не лише з позиції прямого прибутку, а передусім з точки зору зменшення непрямих витрат, пов'язаних з організацією догляду, та економії часу власника. У процесі аналізу AS-IS було встановлено, що значна частина втрат виникає через несвоєчасне виконання процедур, повторні звернення до ветеринарних спеціалістів, дублювання дій і витрати часу на пошук інформації.

Для оцінювання економічного ефекту розглянемо такі складові:

- економія часу власника;
- зниження витрат, пов'язаних із пропущеними або несвоєчасними процедурами;
- оптимізація витрат на ветеринарні послуги та догляд.

У поточному стані (AS-IS) власник витрачає час на:

- пошук інформації про попередні процедури;
- перевірку дат вакцинацій і обробок;
- перегляд нотаток, повідомлень, фото документів;
- координацію дій із сервісами та ветеринарними клініками.

За результатами аналізу можна припустити, що в середньому власник витрачає 10–15 хвилин на день на дії, пов'язані з організацією та перевіркою догляду (без урахування самих процедур).

Прийmemo:

- середній час на організаційні дії — 12 хв/день;
- кількість днів у місяці — 30.

Тоді місячні витрати часу становлять:

$$T_{AS-IS} = 12 \cdot 30 = 360 \text{ хв} = 6 \text{ год}$$

У ТО-ВЕ моделі більшість дій (нагадування, історія, фіксація процедур) централізовані в одному додатку, що дозволяє скоротити ці витрати приблизно до 4–5 хвилин на день.

$$T_{TO-VE} = 5 \cdot 30 = 150 \text{ хв} = 2,5 \text{ год}$$

Отже, **економія часу** становить:

$$\Delta T = 6 - 2,5 = 3,5 \text{ год/міс}$$

Якщо умовно оцінити одну годину часу власника у 150 грн/год, отримаємо:

$$E_{\text{time}} = 3,5 \cdot 150 = 525 \text{ грн/міс}$$

У AS-IS процесі несвоєчасне виконання профілактичних процедур (вакцинацій, обробок, контрольних оглядів) може призводити до:

- ускладнення захворювань;
- необхідності додаткових консультацій;
- повторних або позапланових візитів до ветеринара.

За консервативною оцінкою можна вважати, що 1–2 рази на рік виникають додаткові витрати на суму близько 800–1200 грн, яких можна уникнути за рахунок кращого контролю та регулярності.

Прийmemo середнє значення:

$$E_{\text{vet}} = 1000/12 \approx 83 \text{ грн/міс}$$

Централізований облік процедур та історії догляду дозволяє:

- уникати дублювання процедур;
- зменшувати кількість повторних консультацій через втрату інформації;
- точніше планувати витрати на корми та препарати.

Навіть мінімальне зменшення таких витрат на 100–150 грн на місяць є реалістичним у порівнянні з AS-IS станом.

Прийmemo:

$$E_{\text{opt}} = 120 \text{ грн/міс}$$

Сумарний щомісячний економічний ефект від впровадження мобільного додатку становить:

$$E_{\text{total}} = E_{\text{time}} + E_{\text{vet}} + E_{\text{opt}}$$

$$E_{\text{total}} = 525 + 83 + 120 = 728 \text{ грн/міс}$$

Річний економічний ефект:

$$E_{\text{year}} = 728 \cdot 12 = 8736 \text{ грн/рік}$$

Отримані результати свідчать, що впровадження мобільного додатку для підтримки догляду за домашніми тваринами забезпечує відчутний економічний ефект за рахунок економії часу власника, зниження непрямих витрат та оптимізації організації догляду. Навіть за консервативних оцінок сумарний ефект є позитивним, що підтверджує доцільність реалізації запропонованої інформаційної системи.

1.7 Постановка задачі

На основі проведеного системного аналізу предметної області догляду за домашніми тваринами, результатів функціонального моделювання поточного стану процесу (AS-IS), а також виявлених проблем і сформованої цільової моделі процесу (TO-BE), формується задача розроблення мобільного додатку для підтримки догляду за домашніми тваринами.

Об'єктом дослідження є процес догляду за домашніми тваринами, який включає щоденні рутинні дії, планові профілактичні та ветеринарні процедури, спостереження за станом здоров'я тварини, а також взаємодію власника з ветеринарними та сервісними організаціями.

Предметом дослідження є інформаційна підтримка та організація процесу догляду за домашніми тваринами з використанням мобільного додатку, зокрема способи збирання, збереження, структурування та використання інформації про догляд, а також механізми планування і контролю виконання процедур.

Метою розробки є створення мобільного додатку, який забезпечує централізовану підтримку процесу догляду за домашніми тваринами, зменшує

залежність від пам'яті власника, усуває фрагментарність обліку та забезпечує цілісне ведення історії догляду відповідно до цільової моделі процесу (ТО-ВЕ).

Враховуючи виявлені проблеми поточного стану (п. 1.4.2), у межах розроблюваного мобільного додатку доцільно автоматизувати такі задачі:

- автоматизацію планування та нагадувань про щоденні, періодичні та планові процедури догляду;
- автоматизацію обліку виконаних дій догляду з фіксацією часу та типу процедури;
- автоматизацію ведення хронологічної історії догляду за твариною;
- автоматизацію збереження та впорядкування інформації про ветеринарні призначення, документи та рекомендації;
- автоматизацію контролю регулярності процедур та повторюваних подій;
- автоматизацію інформаційної підтримки власника під час догляду за кількома тваринами або за участю кількох відповідальних осіб.

Основні функціональні задачі розробки

Для досягнення поставленої мети мобільний додаток повинен забезпечувати виконання таких функціональних задач:

1. Зберігання базової інформації про тварину (вид, вік, індивідуальні особливості).
2. Формування та редагування переліку процедур догляду з урахуванням їх періодичності.
3. Планування подій догляду та генерація нагадувань відповідно до встановлених графіків.
4. Фіксацію виконання щоденних і планових дій догляду.
5. Ведення централізованої історії догляду, доступної для перегляду та аналізу.
6. Збереження ветеринарних документів, призначень і результатів обстежень.
7. Підтримку обліку догляду за кількома тваринами в межах одного користувача.

Розроблюваний мобільний додаток орієнтований на підтримку та організацію процесу догляду, а не на заміну професійних ветеринарних рішень. Додаток не здійснює медичної діагностики та не формує лікувальні рекомендації, а забезпечує інформаційну підтримку, планування та контроль виконання процедур відповідно до рішень, прийнятих власником та ветеринарним спеціалістом.

Очікуваним результатом є мобільний додаток, який реалізує цільову модель процесу догляду (TO-BE) та забезпечує впорядковану, зручну й доступну підтримку догляду за домашніми тваринами в умовах повсякденного використання.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ МОБІЛЬНОГО ДОДАТКУ

2.1 Аналіз і специфікація вимог до інформаційної системи

Розглянемо вимоги до мобільного додатку.

Вимога 1. Продукт повинен мати інтуїтивно зрозумілий мобільний інтерфейс для власників тварин, адаптований під різні діагоналі смартфонів.

Вимога 2. Користувач повинен мати змогу отримувати миттєві пуш-сповіщення через ПЗ про критичні події (час щеплення, прийому ліків).

Вимога 3. Продукт повинен забезпечити можливість створення детального цифрового профілю для кожного вихованця з можливістю редагування даних.

Вимога 4. Продукт має бути швидким та стабільним, забезпечуючи синхронізацію з хмарною БД Firebase у реальному часі.

Вимога 5. ПЗ має надавати інструменти для структурування медичної історії, включаючи завантаження фото документів або результатів аналізів.

Вимога 6. Продукт повинен підтримувати офлайн-режим для перегляду базової інформації про тварину за відсутності інтернет-з'єднання.

Опис інформаційних взаємозв'язків задач

1. Користувач:
 - Задачі: Створення профілю улюбленця, перегляд графіку догляду, внесення медичних даних, налаштування сповіщень.
2. Система моніторингу та сповіщень:
 - Взаємодія з користувачем для ініціації нагадувань на основі внесеного у БД графіку.
3. Модуль медичної карти та історії:
 - Отримання та обробка даних від користувача щодо стану здоров'я тварини.
 - Надання структурованої інформації за запитом.
4. Система керування профілем та хмарна БД:

- Зберігання, оновлення та надійна автентифікація особистої інформації користувачів.
 - Відображення індивідуалізованої стрічки подій для кожної тварини окремо.
5. Система Контактної Інформації та Підтримки:
- Прийняття та обробка запитань та коментарів користувачів щодо роботи МД.
 - Надання довідкової інформації та зв'язок із технічною підтримкою у разі виникнення помилок синхронізації з БД.
6. Система Адміністрування:
- Редагування, видалення та додавання нових категорій догляду.
 - Керування зверненнями користувачів та модерація загальної бази знань.

Аналіз функціонала. Узгодження з вимогами:

Таблиця 2.1 – Таблиця функціональних вимог

№	User story	Result
1	Користувач бажає створити детальний профіль для свого вихованця.	Система надає форму для введення імені, породи, віку та завантаження фото тварини у БД.
2	Користувач бажає встановити нагадування про наступне щеплення.	ПЗ дозволяє обрати дату в календарі та автоматично генерує пуш-сповіщення у визначений час.
3	Користувач хоче бачити історію медичних показників (ваги) у динаміці.	У системі реалізовано модуль збереження антропометричних даних із відображенням попередніх записів.
4	Користувач має запитання щодо функціоналу додатка або термінової допомоги.	Присутня підсистема швидкого зв'язку (чат-бот або форма зворотного зв'язку) та розділ FAQ.
5	Користувач бажає мати доступ до даних навіть без активного інтернету.	Реалізовано механізм локального кешування даних, що синхронізуються з Firebase при появі мережі.
6	Адміністратор має змогу додавати нові типи процедур або поради щодо догляду.	Присутня панель керування контентом, що дозволяє оновлювати інформаційні довідники для всіх користувачів.

Інформаційна система повинна виконувати певні операції та функції відповідно до функціональних вимог. Основні групи користувачів взаємодіють із системою за допомогою діаграми варіантів використання, яка описує їхні дії та реакцію на ПЗ.

Вимоги було розділено на три групи: керування профілями, керування доступом та моніторинг стану здоров'я, щоб полегшити їх розуміння та впорядкування. Дані вимоги визначено як для звичайного користувача (власника тварини), так і для адміністратора системи.

Таблиця 2.1 - Матриця відношень ч1

Вимога користувача (Source)	Модуль профілю тварини	Модуль медичного обліку	Модуль сповіщень
Створення картки вихованця	Aggregation		
Редагування біометричних даних	Aggregation		
Внесення даних про щеплення		Aggregation	
Запис результатів аналізів		Aggregation	
Налаштування часу нагадувань			Aggregation
Вибір типу сповіщення (пуш/звук)			Aggregation

Таблиця 2.2 – Матриця відношень ч2

Назва поля (Field Name)	Тип даних (Data Type)	Опис (Function)
pet_name	String	Імя тварини (напр. Rex)
breed_type	String	Порода для фільтрації в списку
weight_val	Double	Актуальна вага (24.5 kg)
temp_val	Double	Показник температури (38.6 °C)
pulse_status	String	Статус пульсу (Normal)
image_url	Image Path	Посилання на фото з хмари

Таблиця 2.3 – Менеджер специфікацій

Назва вимоги (Item)	Тип (Stereotype)	Статус (Status)	Складність (Difficulty)	Пріоритет (Priority)
Авторизація користувача	Functional	Approved	Medium	High
Реєстрація вихованця	Functional	Approved	Medium	High
Редагування медкартки	Functional	Approved	Low	High
Завантаження фото (Storage)	Functional	Approved	Medium	Medium

Продовження таблиці 2.3 – Менеджер специфікацій

Налаштування сповіщень	Functional	Approved	High	High
Перегляд календаря подій	Functional	Approved	Medium	Medium
Модерація контенту (Адмін)	Functional	Approved	Medium	Low
Модерація контенту (Адмін)	Functional	Approved	Low	Medium

2.2 Постановка та алгоритм розв'язання задачі

Оскільки мобільний додаток, що планується розробити на платформі FlutterFlow, працює в режимі реального часу, було обрано підхід до проектування, що базується на подіях, або об'єктно-орієнтований дизайн. Процедури моделюються за допомогою технології BPMN. Власник тварини може переглядати у мобільному додатку нагадування, вносити інформацію про стан здоров'я та отримувати поради. Довідники та загальна база знань можуть бути оновлені адміністратором системи.

1. Діаграма зв'язків.

Розглянемо наведену на рисунку 2.2 матрицю відношень функціональних вимог. Вона демонструє архітектуру цілісності системи, де кожна атомарна операція (Джерело) збирається у відповідний високорівневий модуль керування (Мета). Таким чином можна організувати розробку інтерфейсу FlutterFlow та налаштування колекцій у Firebase.

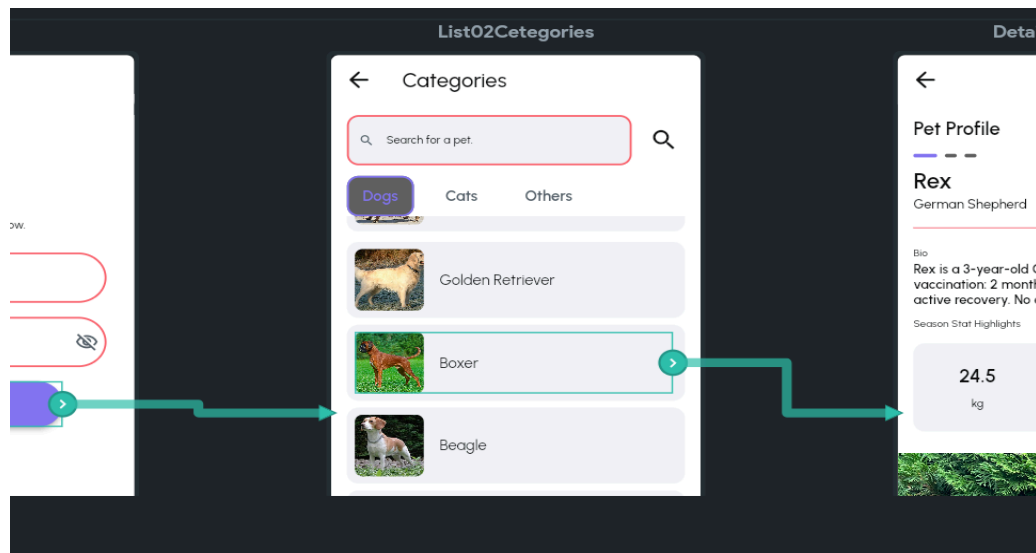


Рисунок 2.1 - Діаграма зв'язків

2. Діаграми процесів.

Включають в себе такі ключові процеси як:

- **Авторизація/Реєстрація** Відбуваються різні процеси в залежності від наявності даних у системі. Якщо у системі створено та збережено обліковий запис власника тварини, користувач автоматично входить до інтерфейсу моніторингу. Система видає помилку, якщо при вході обліковий запис не знайдено. Система видає помилку, якщо при реєстрації вказано дані, які вже є у системі. Хмарний сервіс Firebase Authentication виконує пошук та верифікацію користувачів.

На рисунку 2.1 наведено алгоритм, які складають основи функціональної частини додатку. Вони забезпечують безперервний цикл роботи, який починається з початку сесії користувача та закінчується виведенням динамічних даних про стан вихованців.

Характеристика задачі

Власники тварин можуть легко відстежувати поточні показники здоров'я своїх тварин за допомогою мобільного додатку, який використовує інформаційну систему ветеринарного контролю. Інформаційна система націлена на те, щоб зробити профілі вихованців простими для перегляду та перегляду історій стану (ваги та температури), щоб швидко реагувати на важливі зміни показників.

Вихідна інформація використовується для зручного та структурованого подання медичних даних власнику тварини, забезпечуючи контроль за здоров'ям вихованця у реальному часі.

Вхідна та вихідна інформація наведені у таблицях 2.3 та 2.4.

Таблиця 2.3 – Вхідна інформація в системі

Назва вхідного повідомлення	Ідентифікатор	Форма подання	Термін і частота надходження	Джерело
Дані про тварину	<i>PetData</i>	екранна форма	При реєстрації вихованця	Власник (введення)
Температура тіла	<i>Temperature</i>	екранна форма	По мірі оновлення датчиків	Firebase Firestore
Вага тварини	<i>Weight</i>	екранна форма	При кожному зважуванні	Firebase Firestore

Таблиця 2.4 – Вихідна інформація в системі.

Назва вихідного повідомлення	Ідентифікатор	Форма подання	Періодичність видання	Користувач інформації
Картка вихованця	<i>PetProfile</i>	екранна форма	Під час вибору об'єкта	Власник тварини
Список тварин	<i>PetList</i>	екранна форма	По мірі завантаження з БД	Власник тварини
Показники здоров'я	<i>VitalSigns</i>	екранна форма	В реальному часі	Власник тварини
Статус авторизації	<i>AuthStatus</i>	екранна форма	При вході в додаток	Власник тварини

2.3. Моделювання інформаційної системи

Сценарій взаємодії, наведений на рисунку 2.4, показує можливості користувача при керуванні функціями мобільного додатка для моніторингу стану тварин. Наступні діаграми послідовності та прецеденти показують порядок обміну даними між інтерфейсом додатка, хмарною інфраструктурою Firebase та власником тварини.

Розроблена модель поведінки охоплює такі ключові аспекти:

- Сценарії реєстрації та авторизації: забезпечують безпечний доступ до персональних карток вихованців через Firebase Authentication.
- Процеси моніторингу: автоматизоване отримання даних з бази Firestore та їх візуалізація на екрані профілю тварини.
- Управління профілями: логіка створення та редагування інформації про вихованця (кличка, порода, вік).

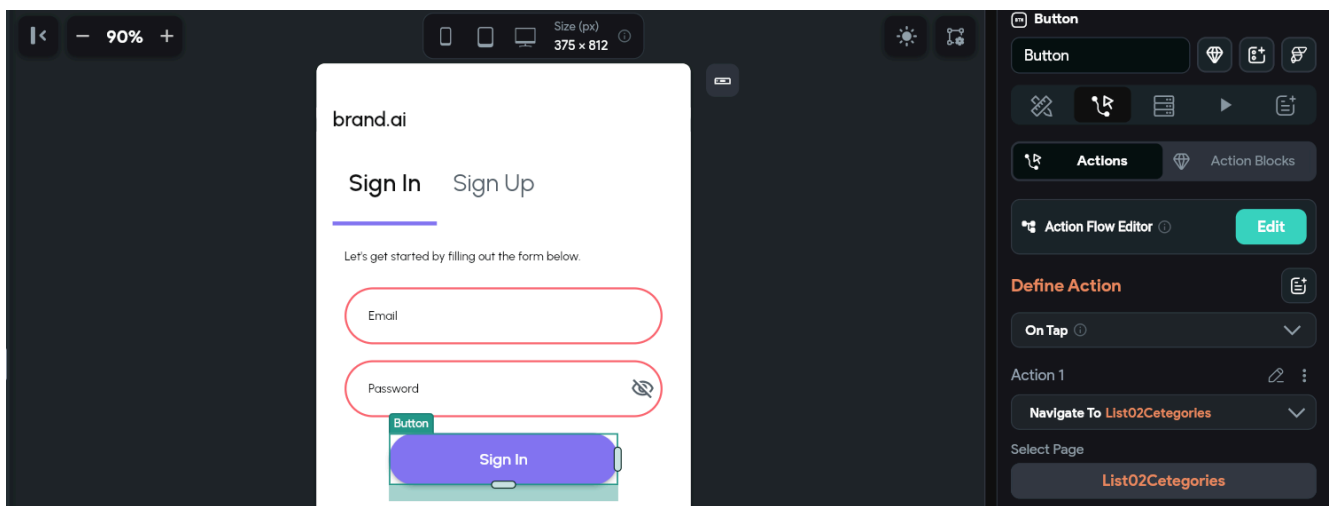


Рисунок 2.2 – Логіка обробки подій

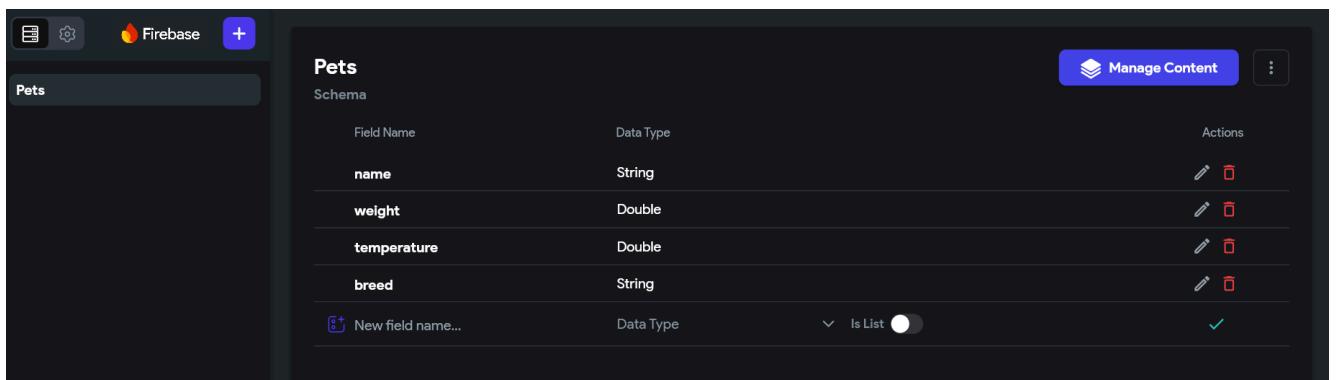


Рисунок 2.3 – Структура полів Firestore

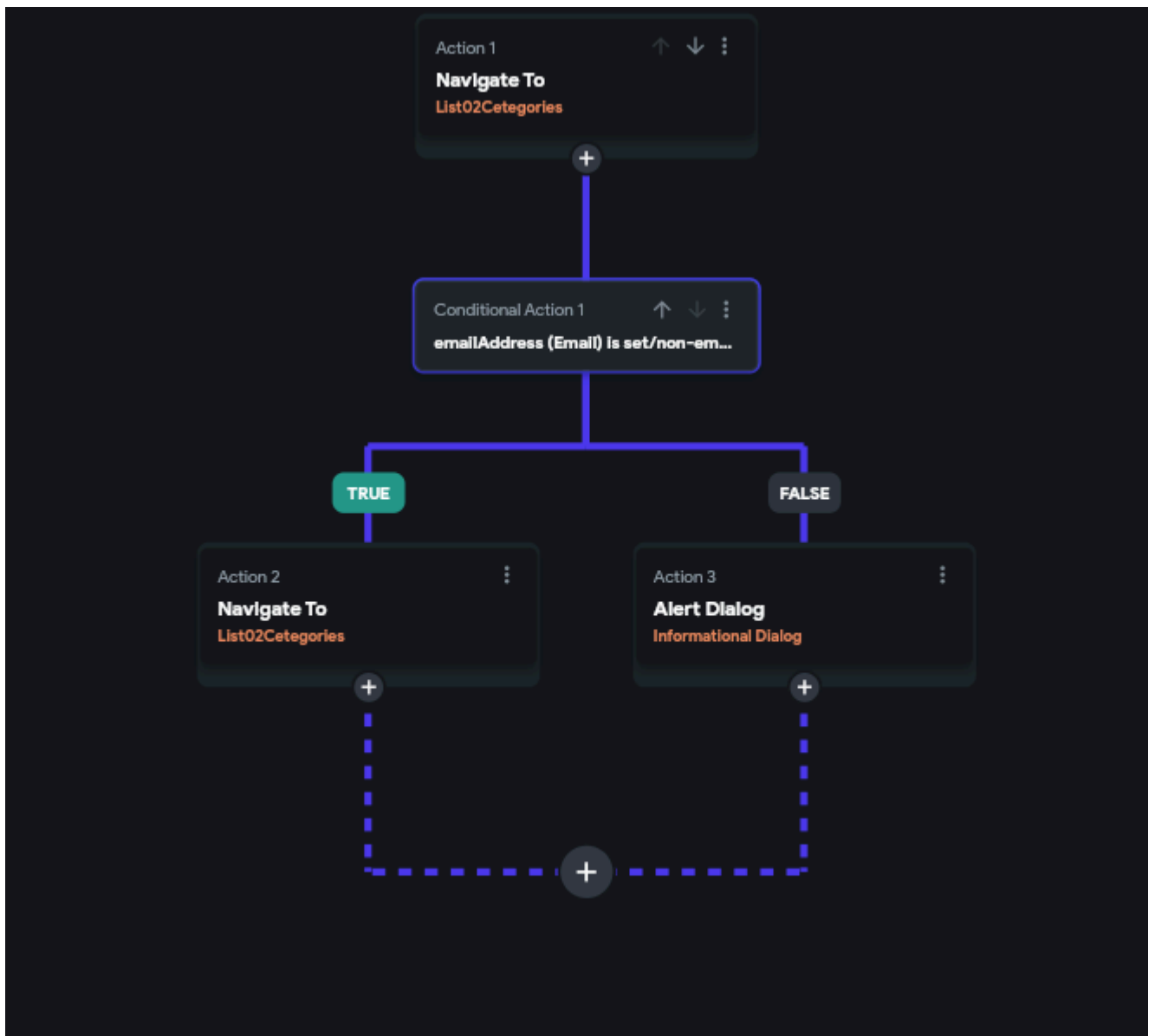


Рисунок 2.4 – Логічна схема алгоритму перевірки умов доступу користувача

Цей розділ описує внутрішню організацію даних і архітектуру інформаційних блоків системи моніторингу вихованців. Структура, яка базується на використанні нереляційної бази даних Firebase Firestore, дозволяє організувати дані у вигляді колекцій і документів у будь-який спосіб, який вам подобається.

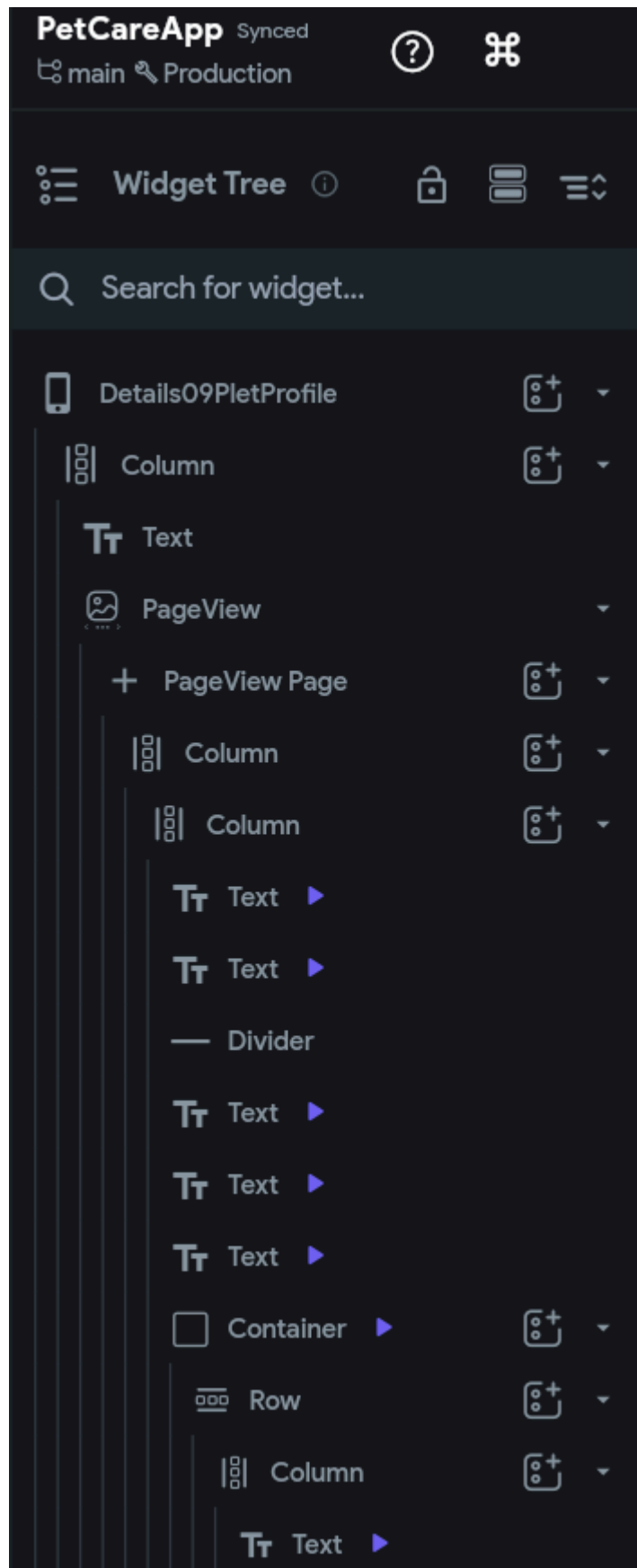


Рисунок 2.5 – Ієрархічна модель компонентів інтерфейсу

На рисунку. 2.5 представляє ієрархічну структуру побудови користувацького інтерфейсу в формі дерева віджетів. Така модель відображає вкладеність візуальних блоків (Container, Column, Row), що дозволяє системі чітко розмежувати зони відповідальності кожного елемента. Це також дозволяє забезпечити високу швидкість рендерингу сторінок додатка.

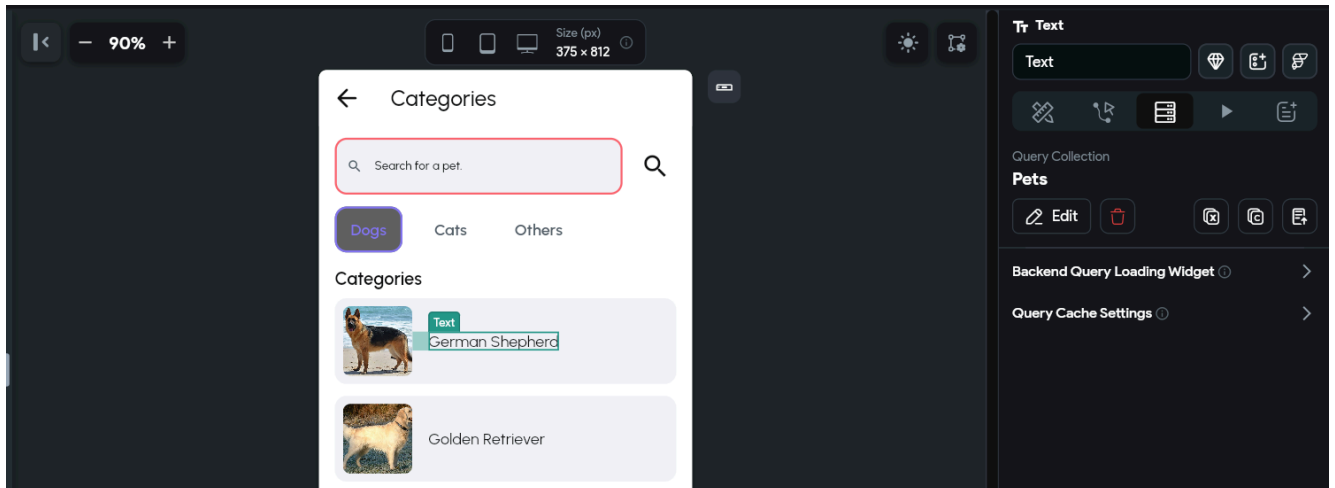


Рисунок 2.6 – Модель зв'язку з хмарною базою даних (Backend Query)

Архітектурне рішення для отримання даних із хмарного середовища показано на рисунку 2.6 через механізм Backend Query. Цей блок структури відповідає за автоматичне з'єднання інтерфейсу з колекцією Pets у Firebase Firestore і запускає запити, щоб отримувати поточні показники стану вихованця.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1 Інформаційне забезпечення мобільного додатку

Інформаційне забезпечення мобільного додатку для підтримки догляду за домашніми тваринами визначає склад, структуру та способи обробки даних, що використовуються в процесі функціонування системи. Основним завданням інформаційного забезпечення є формування єдиного, цілісного інформаційного простору для зберігання, оновлення та використання даних про тварин, їхній стан здоров'я, процедури догляду та дії користувачів.

Розроблювана система орієнтована на використання хмарної бази даних Firebase Firestore, що забезпечує зберігання інформації у вигляді колекцій і документів та підтримує синхронізацію даних у реальному часі. Такий підхід відповідає вимогам мобільного додатку, що використовується у повсякденних умовах і потребує швидкого доступу до актуальної інформації.

Основним користувачем мобільного додатку є власник домашньої тварини, який здійснює щоденний догляд, контролює стан здоров'я вихованця та взаємодіє з ветеринарними та сервісними організаціями. Аналіз предметної області та функціональних вимог показав, що користувач потребує доступу до таких основних груп інформації:

- загальні відомості про тварину (кличка, вид, порода, вік, фото);
- дані про стан здоров'я (вага, температура, інші фізіологічні показники);
- інформація про ветеринарні процедури (вакцинації, огляди, лікування);
- графіки та календар подій догляду;
- історія виконаних процедур і змін стану тварини;
- налаштування нагадувань і сповіщень;
- дані облікового запису користувача.

Інформаційні потреби користувача мають динамічний характер, оскільки більшість даних оновлюються регулярно. Тому система повинна забезпечувати можливість швидкого внесення змін, перегляду історичних даних та синхронізації інформації між пристроями користувача. Логічна модель даних відображає основні сутності предметної області та зв'язки між ними без прив'язки до конкретної реалізації бази даних. В основі логічної моделі мобільного додатку лежать такі ключові сутності:

- **Користувач (User)** — обліковий запис власника тварини;
- **Тварина (Pet)** — цифровий профіль вихованця;
- **Медичний запис (MedicalRecord)** — інформація про процедури, вакцинації, огляди;
- **Показники здоров'я (VitalSigns)** — значення фізіологічних параметрів;
- **Подія догляду (CareEvent)** — заплановані або виконані дії;
- **Нагадування (Reminder)** — об'єкти керування сповіщеннями.

Кожен користувач може мати один або кілька профілів тварин. Для кожної тварини зберігається набір медичних записів, показників здоров'я та подій догляду. Логічна модель забезпечує ієрархічну структуру даних, що відповідає реальному процесу догляду та дозволяє уникнути дублювання інформації.

Фізична модель бази даних реалізується на основі нереляційної СУБД Firebase Cloud Firestore. Дані зберігаються у вигляді колекцій і документів, що дозволяє гнучко масштабувати систему та ефективно працювати з великими обсягами інформації.

Основними колекціями бази даних є:

- `users` - зберігання даних користувачів;
- `pets` - профілі тварин;
- `medical_records` - записи про медичні процедури;
- `vital_signs` - фізіологічні показники;
- `care_events` - події догляду та календар;
- `reminders` - налаштування сповіщень.

Кожен документ має унікальний ідентифікатор та набір атрибутів, що відповідають логічній моделі. Зв'язки між сутностями реалізуються за допомогою ідентифікаторів користувача та тварини.

Колекція `pets` є центральною у структурі бази даних і містить такі основні поля:

- `pet_name` - ім'я тварини;
- `breed_type` - порода;
- `weight_val` - поточна вага;
- `temp_val` - температура тіла;
- `pulse_status` - статус пульсу;
- `image_url` - посилання на фото тварини.

Колекція `medical_records` зберігає інформацію про медичні процедури та включає:

- тип процедури;
- дату проведення;
- опис або примітки;
- ідентифікатор тварини.

Колекція `care_events` використовується для формування календаря догляду та містить дані про заплановані та виконані дії. Колекція `reminders` зберігає параметри сповіщень і час їх активації.

Цілісність даних у системі забезпечується за рахунок логічної структури зв'язків між колекціями та використання механізмів автентифікації і правил доступу Firebase. Кожен запис у базі даних прив'язаний до конкретного користувача, що унеможливорює несанкціонований доступ до інформації інших власників.

Використання ідентифікаторів тварин дозволяє пов'язувати медичні записи, показники здоров'я та події догляду з відповідним профілем. Такий

підхід забезпечує узгодженість інформації, спрощує її обробку та дозволяє формувати повну історію догляду за кожною твариною.

3.2 Алгоритмізація та реалізація комплексу задач автоматизації

Архітектура мобільного додатку для підтримки догляду за домашніми тваринами побудована за принципом клієнт–серверної взаємодії з використанням хмарної інфраструктури. Клієнтською частиною є мобільний додаток, який забезпечує користувацький інтерфейс, обробку введених даних та взаємодію з серверними сервісами. Серверна частина представлена хмарними сервісами Firebase, що відповідають за автентифікацію користувачів та збереження даних.

Основними компонентами архітектури є:

- мобільний клієнт (інтерфейс користувача, логіка роботи додатку);
- Firebase Authentication (реєстрація та автентифікація);
- Firebase Cloud Firestore (зберігання структурованих даних);
- Firebase Storage (зберігання зображень тварин та документів).

Взаємодія між компонентами здійснюється через захищені канали зв'язку з використанням протоколу HTTPS. Дані передаються у форматі JSON, що забезпечує зручність обробки та масштабування системи.

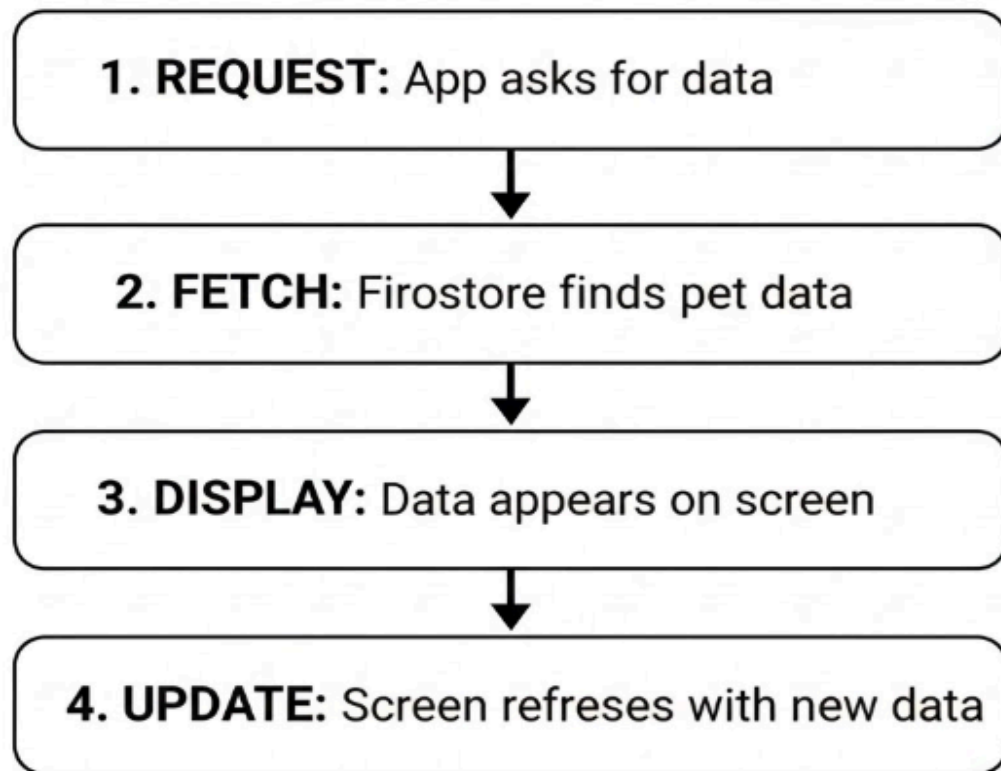


Рисунок 3.1 – Архітектура мобільного додатку

Архітектурне рішення дозволяє мінімізувати навантаження на клієнтський пристрій, забезпечити збереження даних у хмарі та доступ до них з різних пристроїв користувача після автентифікації.

Реєстрація та автентифікація користувача є початковим етапом роботи з мобільним додатком та забезпечує захист персональних даних і обмеження доступу до інформації. Алгоритм автентифікації реалізовано з використанням сервісу Firebase Authentication, який підтримує створення облікових записів на основі електронної пошти та пароля.

1. Користувач відкриває форму реєстрації.
2. Вводить адресу електронної пошти та пароль.
3. Дані передаються до сервісу Firebase Authentication.
4. У разі успішної реєстрації створюється унікальний ідентифікатор користувача (UID).
5. Додаткові дані користувача зберігаються у колекції users бази даних.
6. Користувач перенаправляється на головний екран додатку.

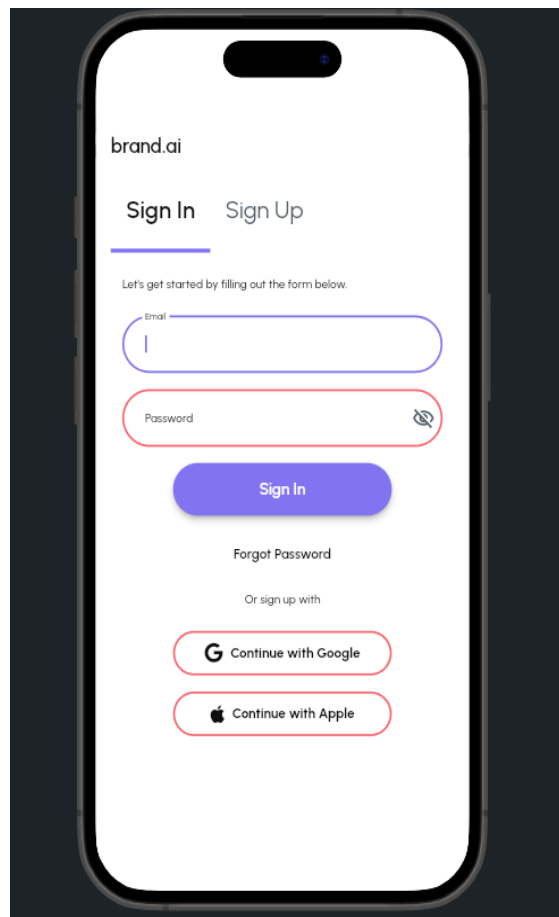


Рисунок 3.2 – Вікно реєстрації користувача

Розглянемо алгоритм функціонування мобільного застосунку при відображенні показників здоров'я тварини, наведений на рисунку 3.3:

1. Користувач вводить логін та пароль.
2. Дані перевіряються сервісом Firebase Authentication.
3. У разі успішної перевірки система надає доступ до функціоналу додатку.
4. Завантажуються персональні дані користувача та пов'язані з ним профілі тварин.

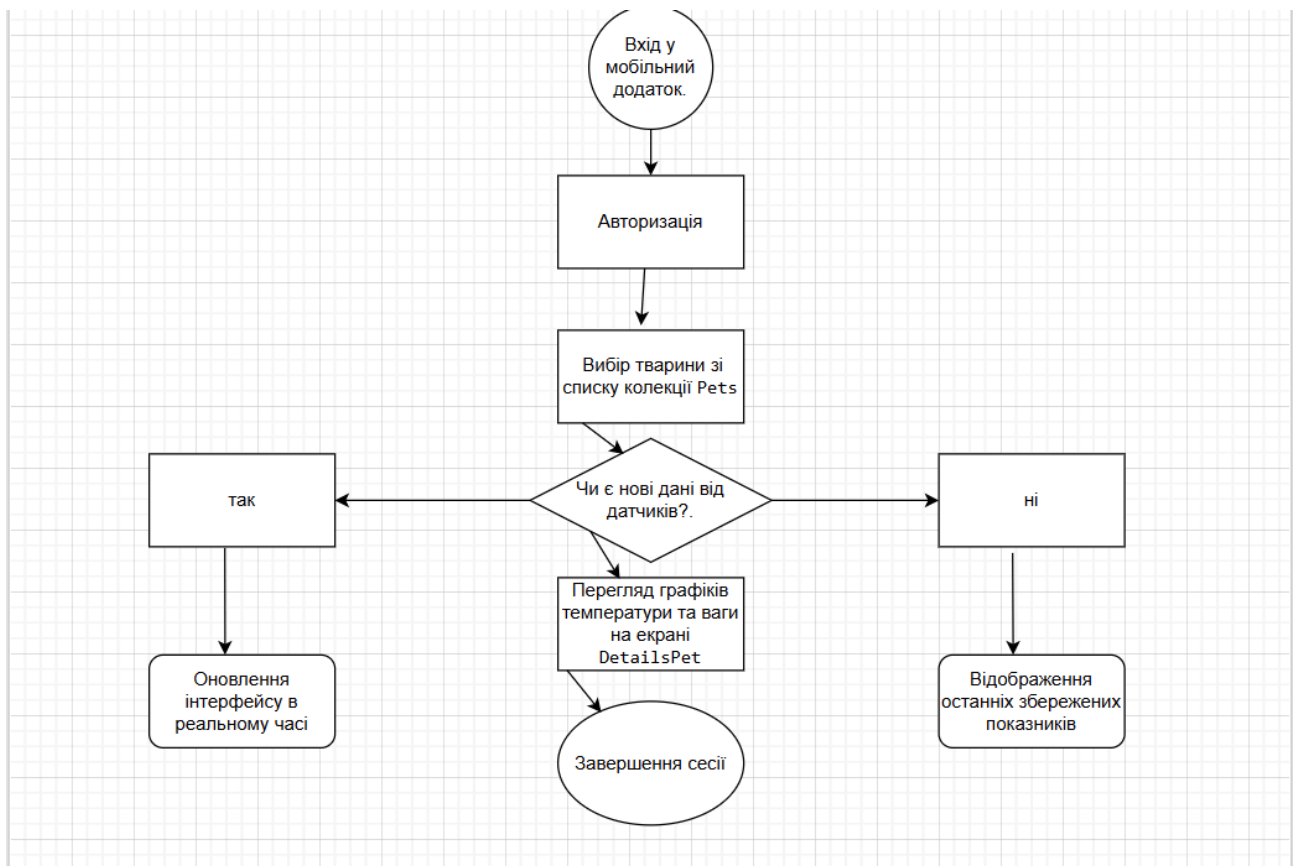


Рисунок 3.3 - Алгоритм функціонування мобільного застосунку при відображенні показників здоров'я тварини

Використання Firebase Authentication дозволяє не зберігати паролі у базі даних у відкритому вигляді та забезпечує базовий рівень безпеки системи.

Після успішної автентифікації користувач отримує можливість додавати та редагувати профілі домашніх тварин. Профіль тварини є центральною сутністю системи та використовується для пов'язування всіх подій догляду, медичних записів і нагадувань.

1. Користувач обирає функцію додавання нової тварини.
2. Заповнює форму з основними даними (ім'я, вид, порода, вага тощо).
3. За потреби додає фотографію тварини.
4. Дані передаються до бази даних Firestore.
5. Профіль тварини відображається у списку доступних тварин користувача.

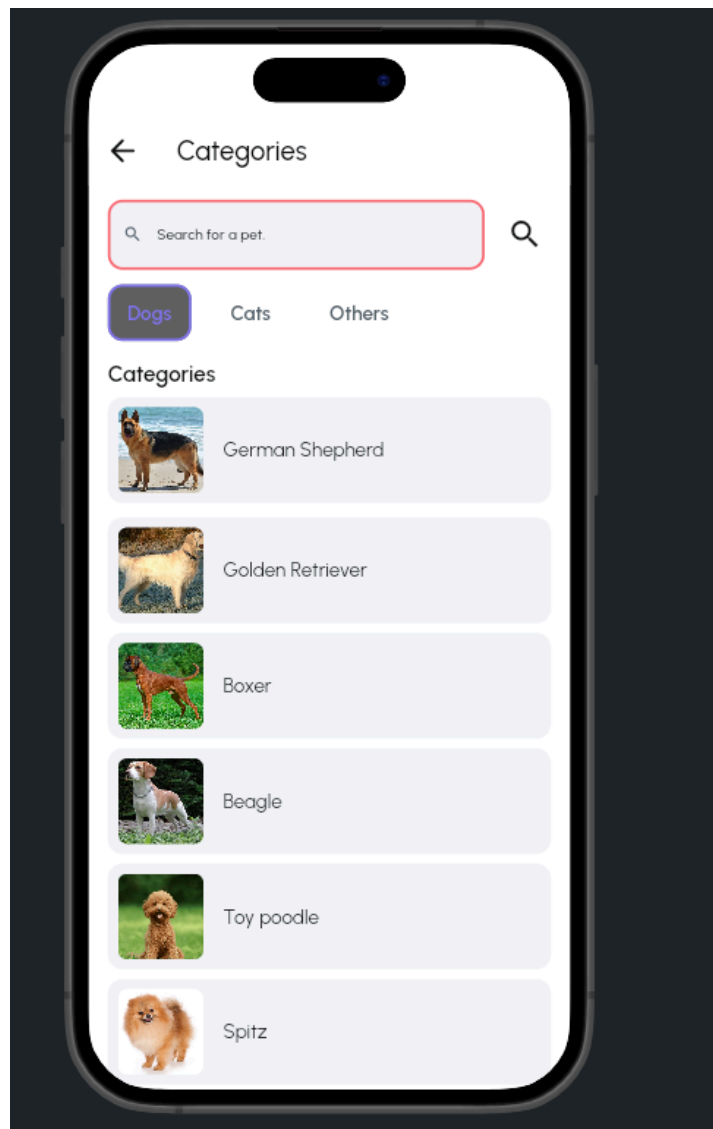


Рисунок 3.4 – Форма створення профілю тварини

Алгоритм редагування профілю тварини:

1. Користувач відкриває існуючий профіль тварини.
2. Вносить зміни до необхідних полів.
3. Оновлені дані зберігаються у відповідному документі Firestore.
4. Інтерфейс додатку оновлюється відповідно до внесених змін.

Реалізація даного алгоритму забезпечує гнучке керування профілями тварин та створює основу для подальшого планування процедур догляду й ведення історії стану здоров'я.

Процес планування процедур догляду є одним з ключових у функціонуванні мобільного додатку, оскільки забезпечує регулярність виконання дій і зменшує залежність від пам'яті користувача. Алгоритм

планування реалізується на основі подій, що мають задану періодичність або фіксовану дату виконання.

Алгоритм планування процедур догляду включає такі етапи:

1. Користувач обирає профіль тварини.
2. Вибирає тип процедури (годування, обробка, вакцинація, прогулянка тощо).
3. Задає дату та час виконання або періодичність процедури.
4. Визначає параметри нагадування (час сповіщення, повтор).
5. Дані зберігаються у базі даних як подія догляду.
6. Система автоматично формує розклад процедур.

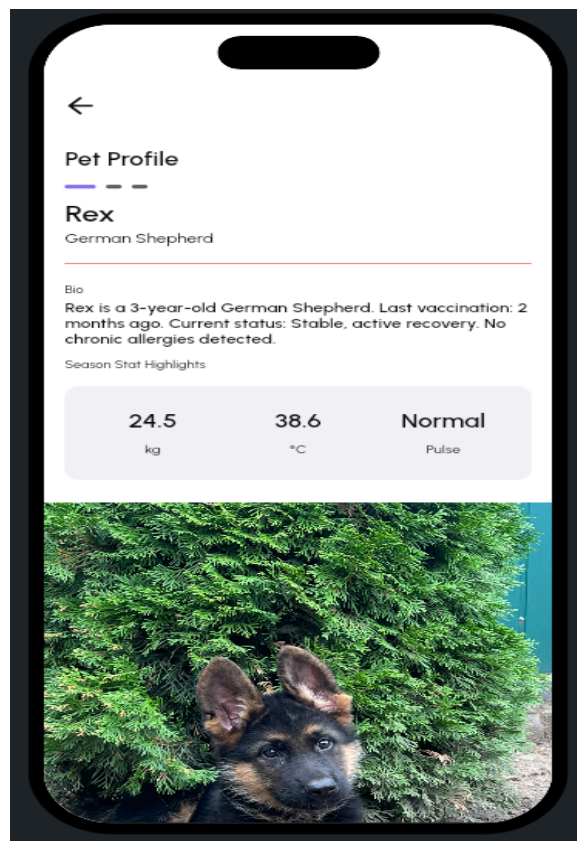


Рисунок 3.5 – Вікно створення події догляду

Алгоритм формування нагадувань:

1. Система аналізує заплановані події.
2. За заданий проміжок часу до події генерується сповіщення.
3. Користувач отримує нагадування на мобільному пристрої.
4. Подія позначається як очікувана або виконана.

Такий підхід дозволяє підтримувати регулярність процедур і мінімізувати ризик їх пропуску.

Фіксація виконаних дій є необхідною для формування хронологічної історії догляду та подальшого аналізу стану тварини. Алгоритм реалізує простий механізм підтвердження виконання процедур без перевантаження користувача додатковими діями.

Алгоритм фіксації виконання дій:

1. Користувач відкриває список запланованих процедур.
2. Обирає виконану процедуру.
3. Підтверджує факт виконання.
4. За необхідності додає короткий коментар або спостереження.
5. Дані зберігаються у базі даних з відміткою часу.

Формування історії догляду:

1. Система групує виконані дії за датами.
2. Відображає їх у хронологічному порядку.
3. Забезпечує доступ до історії для кожної тварини окремо.

Ведення історії дозволяє користувачу відстежувати регулярність процедур і аналізувати зміни у догляді за твариною.

Для повноцінної підтримки процесу догляду мобільний додаток забезпечує можливість збереження ветеринарних даних та супровідних документів, зокрема результатів оглядів, довідок, призначень і рекомендацій спеціалістів.

Алгоритм збереження ветеринарних даних:

1. Користувач відкриває розділ ветеринарної інформації.
2. Обирає тип запису (огляд, вакцинація, лікування).
3. Вводить опис або примітки.
4. Додає дату події.
5. Дані зберігаються у відповідній колекції бази даних.

Алгоритм збереження документів:

1. Користувач завантажує файл або фотографує документ.
2. Файл зберігається у Firebase Storage.

3. У базі даних зберігається посилання на файл.
4. Документ відображається у профілі тварини.

Такий алгоритм дозволяє зберігати всі ветеринарні дані в одному місці та забезпечує швидкий доступ до них у разі потреби.

3.3. Інструкція користувача

Процес авторизації та входу: робота з додатком починається з екрана автентифікації. Реалізована можливість входу через стандартну форму «Email/Password», а також інтегровані сервіси швидкої авторизації через Google та Apple ID. Це дозволяє користувачеві не витратити час на створення окремого профілю, а одразу перейти до функціонала. Якщо пароль було втрачено, передбачена функція «Forgot Password» для його оперативного відновлення.

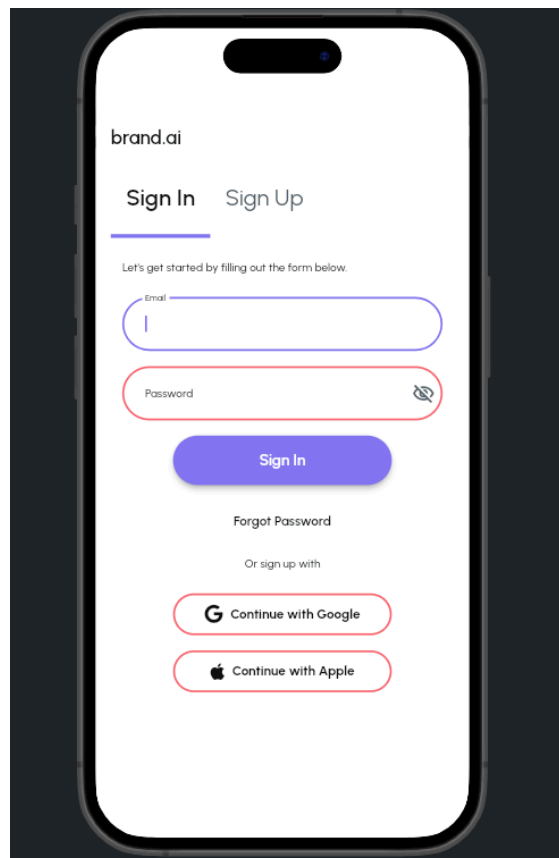


Рисунок 3.6 - Інтерфейс авторизації користувача в мобільному застосунку

Навігація та каталог вихованців: після успішного входу користувач потрапляє до головного каталогу. Розроблена система фільтрації за категоріями (Dogs, Cats, Others), що значно спрощує пошук у великій базі даних. Верхня частина екрана містить пошуковий рядок «Search for a pet», де за ключовими

словами (наприклад, порода «German Shepherd» або «Spitz») можна миттєво знайти потрібну картку. Кожна картка у списку містить мініатюру фото та назву породи для візуальної зручності.

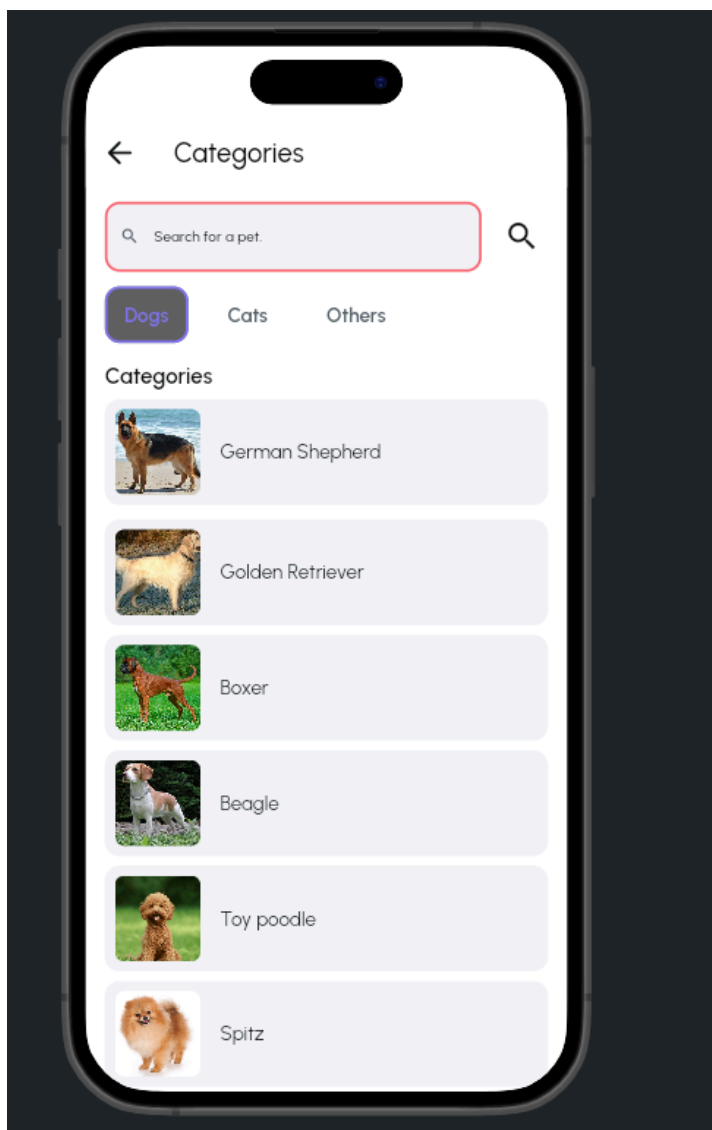


Рисунок 3.7 - Навігація та каталог вихованців

Робота з профілем улюбленця: Центральним елементом системи є детальна картка тварини (на прикладі профілю Рекса).

Найбільш критичні параметри для власника:

- **Біографічні дані:** порода, вік та коротке резюме («Віо»), де відображається інформація про останню вакцинацію та наявність хронічних алергій.

- **Моніторинг показників:** блок «Season Stat Highlights», де в реальному часі відображаються вага (24.5 kg), температура (38.6 °C) та стан пульсу (Normal). Ці дані дозволяють власнику швидко оцінити стан здоров'я тварини.

- **Візуалізація:** нижня частина екрана відведена під велике фото вихованця, що додає інтерфейсу персоналізації та емоційного відгуку

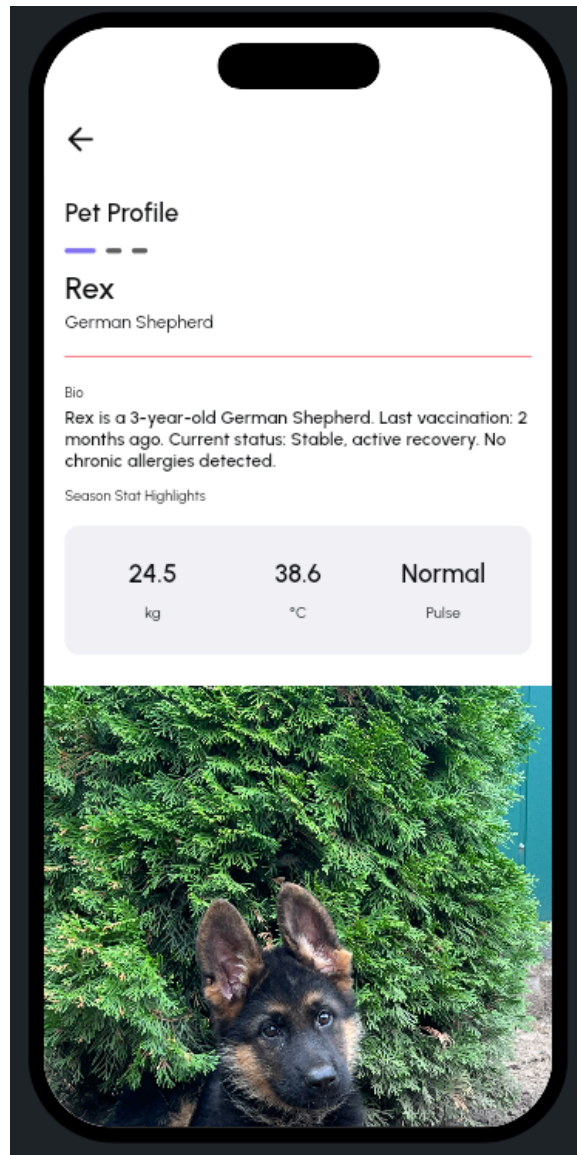


Рисунок 3.8 - Робота з профілем улюбленця

Управління медичними записами та формування звітності Для підтримки актуального стану здоров'я тварини реалізовано модуль документації, подібний до системи формування рахунків у логістичних модулях. Користувач може генерувати «Звіт про стан здоров'я» за певний період (наприклад, за останні 30 днів).

Для створення звіту, необхідно перейти у вкладку «Документація» та натиснути «Згенерувати історію процедур».

Система автоматично збирає всі дані про температуру, вагу та пульс (як на прикладі Рекса з вашою вагою 24.5 kg) і виводить їх у зручний PDF-файл для ветеринара.

Експорт та аналітика даних Враховуючи необхідність зберігання великої кількості записів про вихованців, інтегровано функцію експорту бази даних у зовнішні табличні формати.

Якщо власнику потрібно передати повний список проведених щеплень або історію хвороб у клініку, він може натиснути кнопку «Експорт у Excel». Сформована таблиця містить ідентифікатор спеціаліста

Моніторинг та статус виконання процедур Для оперативного контролю за процесом догляду впроваджено систему статусів у реальному часі, аналогічну моніторингу перевезень.

- Перейшовши у вкладку «Моніторинг», користувач бачить індикатори прогресу для кожної активної послуги (наприклад, «Грумінг» або «Вакцинація»)

- Статус кожного замовлення динамічно змінюється у вигляді шкали: від «В черзі» до «Виконано». Це дозволяє власнику точно знати, на якому етапі перебуває його улюбленець.

Розрахунок витрат на утримання вихованця. Для фінансового планування розроблено калькулятор витрат, де користувач може розрахувати бюджет на догляд.

- У вкладці «Вартість» необхідно ввести базові поля: кількість запланованих відвідувань, витрати на спеціалізоване харчування та ціну за одиницю послуги.

- Після натискання кнопки «Розрахувати» система видає загальну суму витрат, що допомагає власнику контролювати свій бюджет. Особлива увага при розробці була приділена створенню модульної архітектури. Це дозволяє відокремити візуальну складову від бізнес-логіки, що полегшує подальше

масштабування системи та інтеграцію нових датчиків моніторингу. Для забезпечення взаємодії з базою даних Cloud Firestore було реалізовано спеціалізовані сервіси, які в реальному часі відстежують зміни в стані тварини (температура, вага) та миттєво оновлюють дані на екрані користувача. Одним із ключових елементів користувацького інтерфейсу є програмний модуль навігації, який забезпечує безшовний перехід між основними функціональними екранами додатка. Реалізація цього модуля у формі окремого компонента (Component) дозволяє уніфікувати досвід користувача та зменшити навантаження на апаратні ресурси мобільного пристрою.

3.4 Технічне та системне забезпечення розробки

Мобільний пристрій користувача.

Основним технічним засобом взаємодії користувача з системою є мобільний пристрій (смартфон або планшет), що працює під управлінням сучасної мобільної операційної системи. Вибір мобільного пристрою як клієнтської платформи зумовлений характером процесу догляду за домашніми тваринами, який не має фіксованого місця виконання та потребує оперативного доступу до інформації.

Мобільний пристрій забезпечує:

- постійну доступність додатку для користувача;
- можливість швидкого введення та перегляду даних;
- отримання локальних і push-сповіщень;
- використання вбудованих апаратних засобів (камера для збереження фото документів або тварин).

Для коректної роботи додатку достатньо мобільного пристрою з базовими характеристиками, що відповідають сучасним вимогам мобільних операційних систем, що робить систему доступною для широкого кола користувачів.

Серверна частина.

Окремий фізичний сервер у межах розроблюваної системи не використовується. Серверні функції реалізуються за рахунок хмарної інфраструктури, що дозволяє уникнути витрат на придбання, налаштування та адміністрування власного серверного обладнання.

Використання хмарних сервісів дозволяє:

- забезпечити централізоване зберігання даних;
- автоматично масштабувати систему залежно від кількості користувачів;
- підтримувати доступ до даних з різних пристроїв після автентифікації.

Таким чином, серверна логіка системи реалізується у вигляді сервісів, доступних через мережу Інтернет, без необхідності розгортання власного серверного середовища.

Хмарні сервіси.

Для зберігання та обробки даних у мобільному додатку використовується хмарна платформа Firebase, яка надає набір інтегрованих сервісів для розробки мобільних застосунків. Основними використовуваними компонентами є:

- Firebase Authentication — для реєстрації та автентифікації користувачів;
- Firebase Cloud Firestore — для зберігання структурованих даних про тварин, процедури догляду та історію подій;
- Firebase Storage — для збереження фотографій тварин і ветеринарних документів.

Використання хмарних сервісів дозволяє забезпечити:

- високу доступність даних;
- автоматичне резервне копіювання;
- захист інформації за рахунок механізмів автентифікації та контролю доступу;
- зменшення навантаження на клієнтський пристрій.

Обраний підхід відповідає сучасним вимогам до мобільних інформаційних систем і дозволяє зосередитися на реалізації функціоналу додатку без необхідності підтримки складної серверної інфраструктури.

Таким чином, вибір мобільного пристрою як клієнтської платформи, використання хмарних сервісів замість окремого фізичного сервера та

застосування сучасної хмарної інфраструктури є обґрунтованими з точки зору функціональних вимог, зручності використання та технічної доцільності.

Топологія комп'ютерної мережі, що використовується для функціонування мобільного додатку підтримки догляду за домашніми тваринами, визначається архітектурою клієнт–серверної взаємодії та особливостями використання хмарних сервісів. Основним принципом побудови мережі є забезпечення надійного та безпечного обміну даними між мобільним клієнтом і серверною частиною системи.

Клієнт–серверна взаємодія.

У межах розроблюваної системи використовується клієнт–серверна топологія, у якій мобільний додаток виступає клієнтською частиною, а серверні функції реалізуються хмарними сервісами Firebase. Кожен мобільний пристрій користувача працює незалежно та встановлює з'єднання з серверною частиною через мережу Інтернет.

Особливістю такої топології є відсутність прямої взаємодії між клієнтськими пристроями. Усі операції з обробки та збереження даних здійснюються через централізовану серверну інфраструктуру, що забезпечує контроль доступу та цілісність інформації.

Обмін даними між мобільним додатком і базою даних.

Обмін даними між мобільним додатком і базою даних здійснюється через API хмарних сервісів Firebase з використанням захищених каналів зв'язку. Передача інформації відбувається у форматі структурованих об'єктів даних, що відповідають логічній моделі системи.

Основні типи передаваних даних включають:

- облікові дані користувача;
- профілі тварин;
- події догляду та нагадування;
- медичні записи та показники здоров'я;
- посилання на файли, що зберігаються у хмарному сховищі.

Дані передаються асинхронно, що дозволяє додатку коректно працювати навіть за нестабільного мережевого з'єднання та синхронізувати інформацію після його відновлення.

Схема взаємодії компонентів системи.

Загальна схема взаємодії компонентів включає такі елементи:

- мобільний пристрій користувача;
- сервіс автентифікації;
- хмарну базу даних;
- хмарне сховище файлів.

Мобільний додаток ініціює запити до сервісу автентифікації для перевірки прав доступу. Після успішної автентифікації додаток отримує доступ до відповідних даних у базі даних і сховищі файлів. Усі запити та відповіді передаються через мережу Інтернет із використанням захищеного протоколу.

Таким чином, обрана клієнт–серверна топологія комп'ютерної мережі забезпечує централізоване зберігання даних, безпечний обмін інформацією та масштабованість системи. Використання хмарної інфраструктури дозволяє спростити адміністрування та підвищити надійність роботи мобільного додатку.

Вибір операційної системи мобільного пристрою та протоколів обміну даними здійснювався з урахуванням вимог до доступності, безпеки, сумісності з хмарними сервісами та особливостей використання мобільного додатку у повсякденних умовах.

Операційна система мобільного пристрою.

Мобільний додаток орієнтований на роботу під управлінням сучасних мобільних операційних систем Android та iOS, які є найбільш поширеними платформами серед користувачів мобільних пристроїв. Обрані операційні системи забезпечують стабільну роботу додатку, підтримку сучасних програмних інтерфейсів та інтеграцію з хмарними сервісами.

Використання Android та iOS дозволяє:

- охопити більшість потенційних користувачів;
- забезпечити підтримку push-сповіщень;

- використовувати вбудовані можливості пристрою (камера, сховище, мережа);
- отримувати регулярні оновлення безпеки та системних компонентів.

Крім того, обидві операційні системи мають розвинену екосистему інструментів розробки, що спрощує підтримку та подальший розвиток мобільного додатку.

Протоколи обміну даними.

Для обміну даними між мобільним додатком і хмарними сервісами використовується протокол HTTPS, який є захищеним варіантом протоколу HTTP. Використання HTTPS забезпечує шифрування передаваних даних і захист від несанкціонованого доступу під час передавання інформації мережею Інтернет.

Протокол HTTPS дозволяє:

- забезпечити конфіденційність персональних даних користувачів;
- захистити інформацію про тварин і ветеринарні записи;
- гарантувати цілісність даних під час обміну.

Використання HTTP без шифрування в межах розроблюваної системи не розглядається, оскільки воно не відповідає сучасним вимогам інформаційної безпеки.

Формат обміну даними.

Для представлення та передавання даних між компонентами системи використовується формат JSON (JavaScript Object Notation). Даний формат є текстовим, легким для обробки та широко підтримується мобільними платформами і хмарними сервісами.

Використання JSON забезпечує:

- простоту серіалізації та десеріалізації даних;
- зручність роботи зі структурованою інформацією;
- сумісність з API хмарних сервісів;
- зменшення обсягу передаваних даних порівняно з альтернативними форматами.

Формат JSON добре підходить для реалізації клієнт–серверної взаємодії в мобіл

Таким чином, вибір операційних систем Android та iOS, використання захищеного протоколу HTTPS і формату обміну даними JSON є обґрунтованим з точки зору доступності, безпеки та технічної доцільності. Обрані технології забезпечують стабільну роботу мобільного додатку та створюють основу для його подальшого розвитку.

Стратегія адміністрування мобільного додатку для підтримки догляду за домашніми тваринами визначає порядок забезпечення стабільної роботи системи, збереження даних, актуальності інформації та підтримки користувачів. Обраний підхід до адміністрування орієнтований на використання хмарної інфраструктури, що дозволяє мінімізувати ручне втручання та знизити витрати на технічну підтримку.

Резервне копіювання даних.

Резервне копіювання даних є важливим елементом стратегії адміністрування, оскільки система зберігає персональні дані користувачів і інформацію про стан здоров'я тварин. У межах розроблюваної системи резервне копіювання реалізується за рахунок вбудованих механізмів хмарної платформи Firebase.

Хмарна база даних забезпечує автоматичне збереження даних та їх захист від втрати у разі збоїв. Це дозволяє уникнути необхідності ручного створення резервних копій і зменшує ризик втрати інформації. За потреби можливе налаштування додаткового резервного копіювання відповідно до політик зберігання даних.

Оновлення даних.

Оновлення даних у системі відбувається у режимі реального часу або при наступній синхронізації мобільного додатку з хмарною базою даних. Такий підхід забезпечує актуальність інформації незалежно від пристрою, з якого користувач працює з додатком.

Використання централізованого сховища даних дозволяє:

- автоматично застосовувати зміни без втручання адміністратора;
- уникати розбіжностей між локальними копіями даних;
- забезпечувати коректну роботу додатку після оновлення.

Підтримка користувачів.

Підтримка користувачів здійснюється за рахунок простого та зрозумілого користувацького інтерфейсу, а також можливості обробки звернень через стандартні канали зв'язку. У межах розроблюваної системи не передбачається складна адміністративна панель, оскільки більшість процесів керуються автоматично хмарною платформою.

У разі необхідності підтримка може включати:

- оновлення програмного забезпечення;
- усунення помилок;
- консультації щодо роботи з додатком.

Масштабування системи

Масштабування мобільного додатку забезпечується за рахунок використання хмарної інфраструктури, яка автоматично адаптується до зростання кількості користувачів та обсягу даних. Використання Firebase дозволяє масштабувати систему без зміни архітектури та без додаткових налаштувань серверної частини.

Такий підхід забезпечує:

- стабільну роботу додатку при збільшенні навантаження;
- можливість подальшого розвитку функціоналу;
- зменшення витрат на адміністрування системи.

Отже, обрана стратегія адміністрування забезпечує надійну роботу мобільного додатку, збереження та актуальність даних, підтримку користувачів і можливість масштабування системи без значних технічних витрат.

ВИСНОВКИ

Мобільний додаток, розроблений в процесі виконання кваліфікаційної роботи має практичне значення для власників, вихованців і ветеринарних фахівців. Мобільний додаток дозволяє швидко та ефективно відстежувати показники здоров'я тварин у режимі реального часу. Користувачі можуть отримати структуровані дані про фізичний стан тварини, такі як температура та вага, оперативно реагувати на будь-які відхилення показників від норми. У свою чергу хмарна архітектура дозволяє сенсорам і мобільним додатком негайно синхронізувати дані.

Створена у середовищі FlutterFlow модель мобільного додатка дозволяє переглянути вигляд проєкту та основні можливості після його реалізації. Вона виконує основні функції системи, такі як авторизація через Firebase, перегляд списку вихованців із колекції Firestore, сторінки з графіками та детальними медичними показниками, а також автоматичне оновлення інтерфейсу в реальному часі, коли змінюються дані в хмарному сховищі.

Використання сучасного стеку технологій (Flutter, Firebase) забезпечує високу продуктивність системи при мінімальних вимогах до технічного забезпечення кінцевого користувача, що підтверджує ефективність обраного підходу до проектування.

Робота над додатком включала в себе повний процес моделювання системи, починаючи з аналізу вимог до моніторингу фізіологічних показників тварин, опису основних бізнес-процесів взаємодії з хмарною базою даних та завершуючи тестуванням інтерфейсу.

У майбутньому розробка може включати інтеграцію зі сторонніми сервісами ветеринарної підтримки та аптечними мережами, щоб система могла автоматично замовляти необхідні препарати у разі погіршення стану вихованця. Крім того, рекомендується створити інтелектуальні модулі аналізу даних, щоб підвищити ефективність роботи з великою кількістю об'єктів моніторингу

одночасно. Це дозволить використовувати систему у великих притулках чи розплідниках тварин.

В подальшому варто приділяти особливу увагу впровадженню додаткових методів шифрування даних, щоб гарантувати повний захист медичної інформації та захистити особисті дані власників. Крім вищезазначених цілей, передбачається розширення можливості мобільного додатка доповненої реальності (AR), щоб забезпечити візуалізацію стану здоров'я тварини в реальному часі, що значно спростить процес для користувача отримання першого огляду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» денної та заочної форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.
2. Official Dart Language Guide :
<https://dart.dev/docs>
3. Напалков О. В., Кузнецова Н. В. Розробка кросплатформних мобільних додатків засобами фреймворку Flutter. *Системні технології*. 2022. № 2 (139).
4. Draw.io (Diagrams.net). Documentation for Flowcharts and Activity Diagrams:
<https://www.drawio.com/doc/>
5. Firebase Authentication. Security and identity management:
<https://firebase.google.com/docs/auth?hl=ru>
6. Google Cloud Platform NoSQL Database Services
<https://cloud.google.com/products/databases>
7. Flutter Documentation. <https://docs.flutter.dev/>
8. Firebase Documentation: <https://firebase.google.com/docs/firestore?hl=ru>
9. FlutterFlow Help Center: <https://docs.flutterflow.io/>
10. Береза А. М. Основи створення інформаційних систем. Київ: КНЕУ, 2020. 214 с.
11. Глибовець М. М. Штучний інтелект. Київ: Видавничий дім «Києво-Могилянська академія», 2022. 450 с.
12. Данильченко О. М. Проєктування інтерфейсів користувача мобільних застосунків. Харків: ХНУРЕ, 2021. 180 с.
13. Зайцев С. В. Мобільні операційні системи: архітектура та безпека. Одеса: Астропринт, 2021. 256 с.
14. Козак Л. І. Хмарні обчислення та сервіси Google Firebase. Вінниця: ВНТУ, 2022. 144 с.

15. Литвин В. В. Системи штучного інтелекту: навчальний посібник. Львів: Новий Світ-2000, 2023. 380 с.
16. Мельник Р. А. Технології програмування на мовах високого рівня. Львів: Видавництво Львівської політехніки, 2021. 288 с.
17. Нікольський Ю. В. Дискретна математика та основи алгоритмізації. Львів: Магнолія-2006, 2022. 430 с.
18. Орлюк П. В. Основи ветеринарної діагностики та клінічного огляду. Київ: Ліра-К, 2020. 312 с.
19. Пархоменко О. В. Проектування баз даних NoSQL. Київ: КПІ ім. Ігоря Сікорського, 2021. 192 с.
20. Романюк О. Н. Кросплатформна розробка застосунків. Вінниця: Універсум-Вінниця, 2023. 210 с.
21. Савченко Є. М. Ветеринарна інформатика та цифрова медицина. Харків: ХДВА, 2022. 200 с.
22. Ткаченко О. М. Управління ІТ-проектами: від ідеї до реалізації. Київ: Фенікс, 2021. 340 с.
23. Устенко С. В. Моделювання складних систем в ІТ. Миколаїв: ЧНУ ім. Петра Могили, 2020. 260 с.
24. Шеремет М. В. Клінічна біохімія тварин. Київ: Аграрна освіта, 2021. 370 с.
25. Глибовець М. М. Побудова інтелектуальних систем на основі мобільних технологій. Київ: Видавничий дім «Києво-Могилянська академія», 2023. 280 с.
26. Зайцев С. В. Безпека мобільних застосунків та хмарних сховищ. Одеса: Астропринт, 2024. 210 с.
27. Кунгурцев О. Б. Архітектура кросплатформного програмного забезпечення. Одеса: ОНПУ, 2022. 195 с.
28. Степаненко О. С. Моніторинг фізіологічного стану домашніх тварин: цифрові рішення. Харків: Золоті сторінки, 2023. 240 с.
29. Терейковський І. А. Проектування нейромережевих модулів для мобільних систем моніторингу. Київ: Політехніка, 2022. 320 с.

30. Прокопенко Т. О. Якість та тестування програмного забезпечення мобільних систем. Черкаси: ЧДТУ, 2023. 190 с.

ДОДАТКИ

ДОДАТОК А. Інтерфейс мобільного додатку

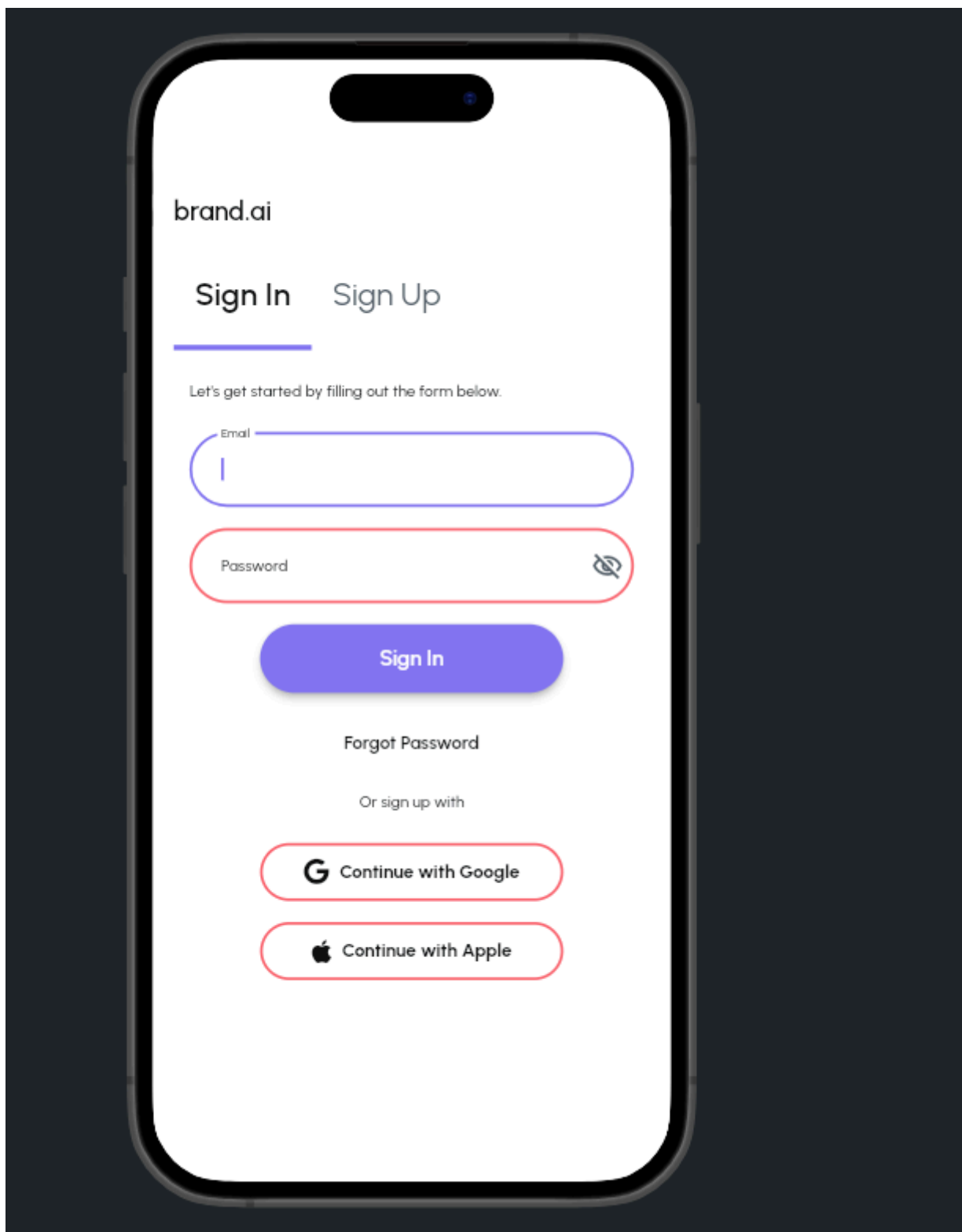


Рисунок А.1 – Головна сторінка

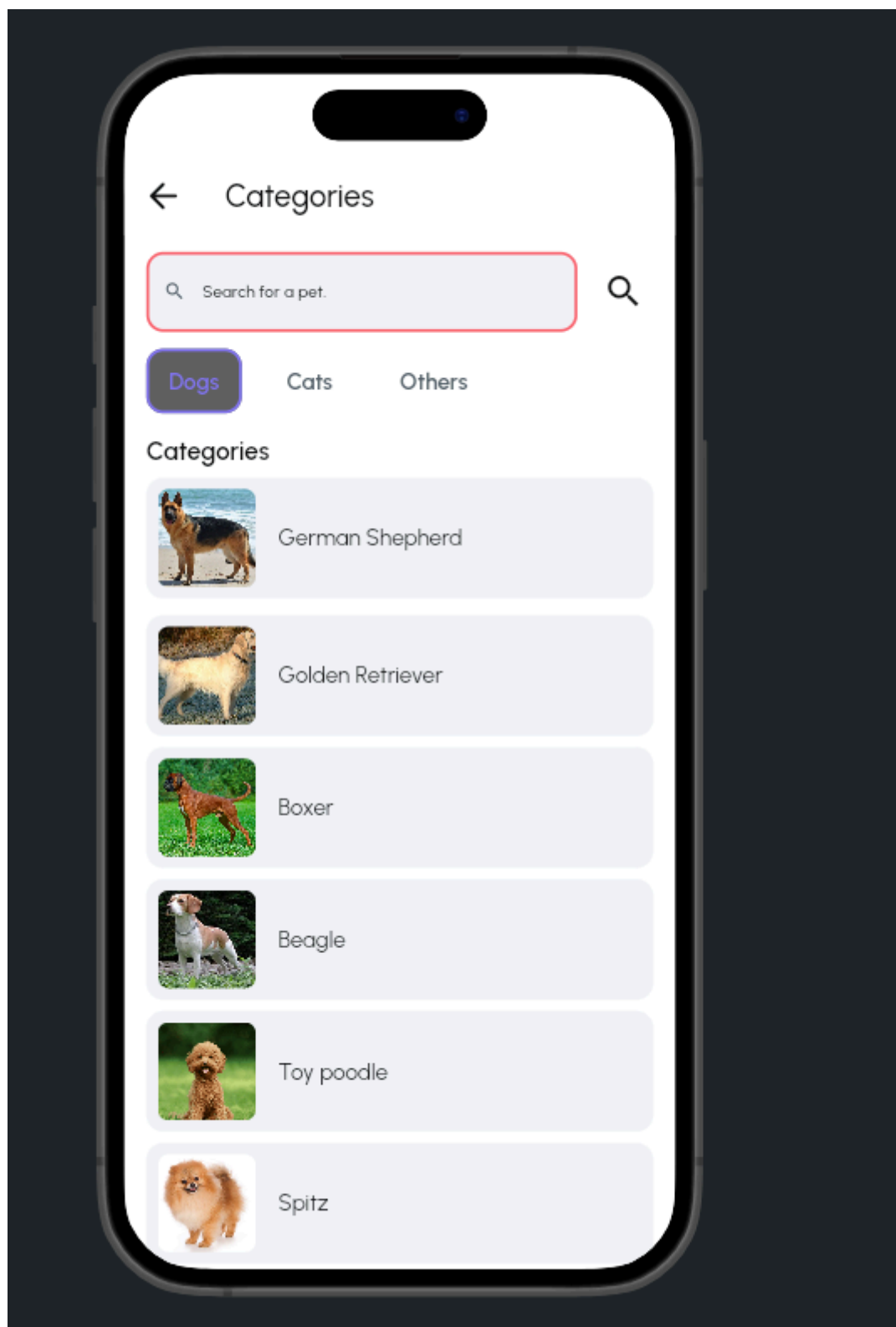


Рисунок А.2 – Каталог вихованців

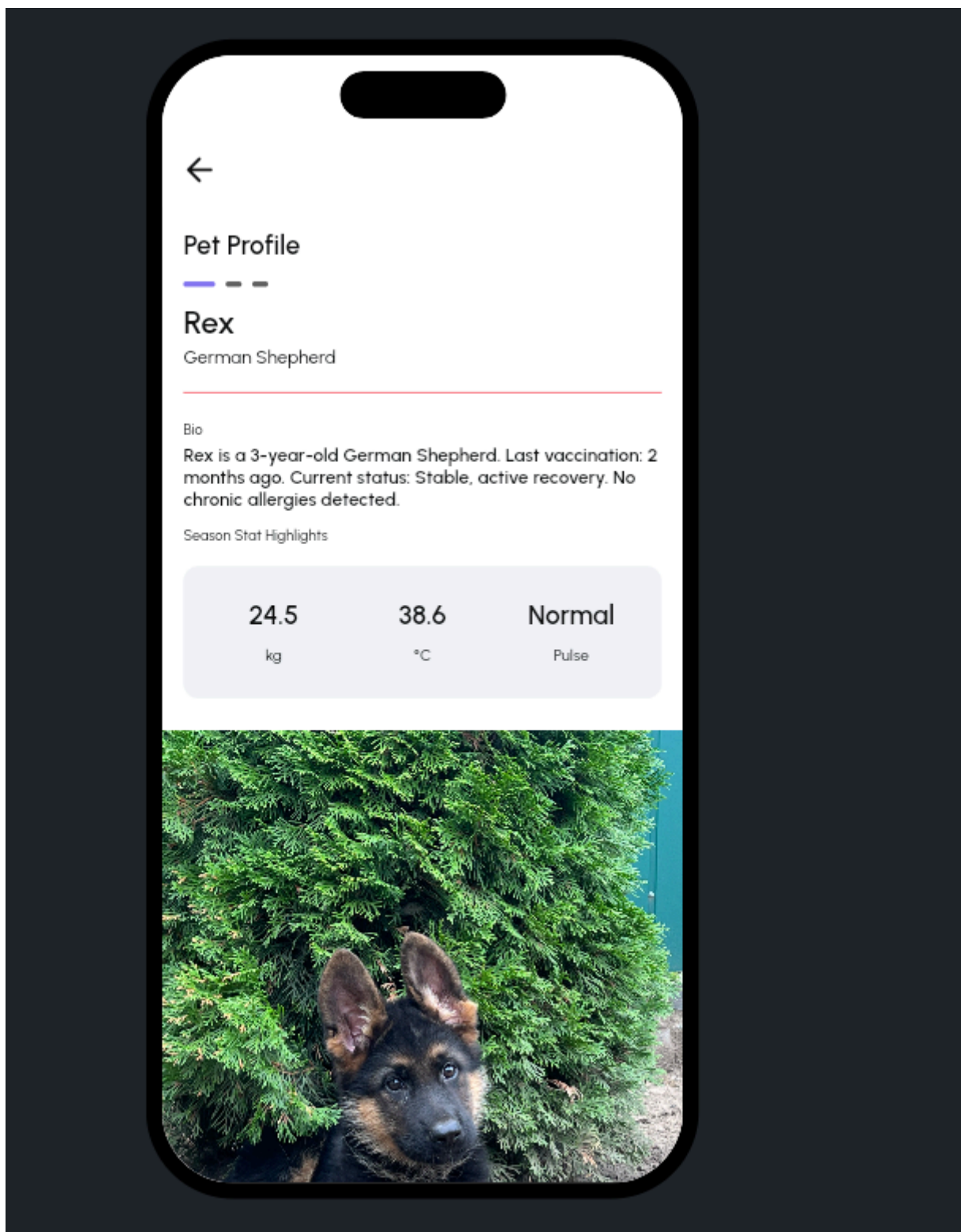


Рисунок А.3 – Профіль вихованця з візуалізацією

ДОДАТОК Б. Лістинг програмного модуля навігації додатка мовою dart

```
import '/flutter_flow/flutter_flow_theme.dart';
import '/flutter_flow/flutter_flow_util.dart';

import '/flutter_flow/flutter_flow_widgets.dart';

import 'dart:ui';

import 'package:flutter/material.dart';

import 'package:google_fonts/google_fonts.dart';

import 'package:provider/provider.dart';

import 'mobile_nav_model.dart';

export 'mobile_nav_model.dart';

class MobileNavWidget extends StatefulWidget {

  const MobileNavWidget({super.key});

  @override

  State<MobileNavWidget> createState() => _MobileNavWidgetState();

}

class _MobileNavWidgetState extends State<MobileNavWidget> {

  late MobileNavModel _model;
```

@override

```
void setState(VoidCallback callback) {  
    super.setState(callback);  
    _model.onUpdate();  
}
```

@override

```
void initState() {  
    super.initState();  
    _model = createModel(context, () => MobileNavModel());
```

```
    WidgetsBinding.instance.addPostFrameCallback((_) => safeSetState({}));  
}
```

@override

```
void dispose() {  
    _model.maybeDispose();  
  
    super.dispose();  
}
```

@override

```
Widget build(BuildContext context) {
```

```
    return Visibility(  
visible: responsiveVisibility(  
    context: context,  
    tabletLandscape: false,  
    desktop: false,  
),  
child: Container(  
    width: double.infinity,  
    height: 100,  
    decoration: BoxDecoration(  
color: FlutterFlowTheme.of(context).secondaryBackground,  
boxShadow: [  
    BoxShadow(  
        blurRadius: 0,  
        color: FlutterFlowTheme.of(context).lineColor,  
        offset: Offset(  
0,  
-1,  
        ),  
    )  
],  
    ),  
child: Padding(  

```

```
padding: EdgeInsetsDirectional.fromSTEB(0, 0, 0, 34),
child: Row(
  mainAxisAlignment: MainAxisAlignment.max,
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Expanded(
child: Column(
  mainAxisAlignment: MainAxisAlignment.max,
  mainAxisAlignment: MainAxisAlignment.center,
  children: [],
),
),
Expanded(
child: Padding(
  padding: EdgeInsetsDirectional.fromSTEB(0, 8, 0, 8),
  child: Column(
mainAxisSize: MainAxisSize.max,
mainAxisAlignment: MainAxisAlignment.center,
children: [],
),
),
),
Expanded(
```

```

child: Padding(
  padding: EdgeInsetsDirectional.fromSTEB(0, 8, 0, 8),
  child: Column(
mainAxisSize: MainAxisSize.max,
mainAxisAlignment: MainAxisAlignment.center,
children: [],
  ),
),
),
Expanded(
child: Padding(
  padding: EdgeInsetsDirectional.fromSTEB(0, 8, 0, 8),
  child: Column(
mainAxisSize: MainAxisSize.max,
mainAxisAlignment: MainAxisAlignment.center,
children: [],
  ),
),
),
Expanded(
child: Padding(
  padding: EdgeInsetsDirectional.fromSTEB(0, 8, 0, 8),
  child: Column(

```

```
mainAxisSize: MainAxisSize.max,  
mainAxisAlignment: MainAxisAlignment.center,  
children: [],  
  
),  
  
),  
  
),  
  
],  
  
),  
  
),  
  
),  
  
),  
  
);  
  
}  
  
}
```