

PBL 1

„INTELIGENTNY SYSTEM USPOKAJANIA ZWIERZĘCIA”



Opracowano przez:

Kinga Konieczna,
Jan Czechowski,
Mikołaj Bednarczyk,
Michał Kołodziejczyk,
Bartłomiej Gromulski

Warszawa, 2025
Warszawa, 01.11.2024 r. – 30.01.2025 r.

SPIS TREŚCI

1.	IDENTYFIKACJA PROBLEMU I BADANIA.....	8
1.1.	STATYSTYKI WŁAŚCICIELI ZWIERZĄT	8
1.2.	CHARAKTERYSTYKA WŁAŚCICIELI	10
1.3.	CHARAKTERYSTYKA ZWIERZĄT	11
1.3.1.	CHARAKTERYSTYKA PSÓW	11
1.3.2.	CHARAKTERYSTYKA KOTÓW	12
1.3.3.	CHARAKTERYSTYKA RYBEK	12
1.3.4.	CHARAKTERYSTYKA PTAKÓW	13
2.	ANALIZA RYNKU.....	14
2.1.	PRZEGLĄD RYNKU.....	14
2.2.	INFORMACJE REGIONALNE	14
2.3.	KLUCZOWE TRENDY NAPĘDZAJĄCE WZROST	14
3.	KOTY I PSY	16
3.1.	PROBLEMY ZDROWOTNE ZWIĄZANE Z POSIADANIEM KOTA.....	16
3.1.1.	CHOROBY UKŁADU POKARMOWEGO	16
3.1.2.	PROBLEMY Z UKŁADEM MOCZOWYM	16
3.1.3.	ZABURZENIA NEUROLOGICZNE	17
3.2.	PROBLEMY BEHAVIORALNE	17
3.2.1.	AGRESJA.....	17
3.2.2.	ZACHOWANIA ZWIĄZANE Z LĘKIEM	17
3.3.	LĘK SEPARACYJNY U KOTÓW I PSÓW.....	18
3.3.1.	PRZYCZYNY LĘKU SEPARACYJNEGO U KOTÓW I PSÓW.....	18
3.3.2.	OBJAWY LĘKU SEPARACYJNEGO U PSÓW I KOTÓW	19
3.3.3.	STRATEGIE RADZENIA SOBIE Z LĘKIEM SEPARACYJNYM	20
3.4.	WAŻNE ASPEKTY W OPIECZE NAD KOTEM.....	20
3.4.1.	ZDROWIE ZWIERZĘCIE	20
3.4.2.	DOBROSTAN PSYCHICZNY.....	21
3.4.3.	OPIEKA PRZY PEŁNOETATOWEJ PRACY.....	21
3.4.4.	ZROZUMIENIE JĘZYKA CIAŁA KOTA PRZEZ WŁAŚCICIELA.....	21
3.4.5.	SOCJALIZACJA I INTERAKCJE SPOŁECZNE.....	21
3.4.6.	PRZYGOTOWANIE NA SYTUACJE AWARYJNE.....	21
3.5.	BADANIE RAS	21

3.5.1.	RASY KOTÓW	21
3.5.2.	RASY PSÓW.....	22
3.6.	RÓŻNICA ZACHOWAŃ W ZALEŻNOŚCI OD PŁCI.....	22
3.6.1.	RÓŻNICE W REAKCJI NA STRES	22
3.6.2.	WPŁYW HORMONU TESTOSTERONU	23
3.6.3.	RÓŻNICE SPOŁECZNE.....	23
3.6.4.	REAKCJE NA INTERAKCJE Z LUDŹMI.....	23
4.	ANKIETY	24
4.1.	Podstawowe pytania	24
4.2.	Pytania dotyczące używania wcześniej technologii IoT	25
4.3.	Pytania dotyczące problemów ze zwierzętami	26
5.	SYNTEZA.....	28
6.	PRZYKŁADOWE PERSONY	29
6.1.	JAKUB KOWALSKI	29
6.2.	EWA DĄBROWSKA	30
6.3.	MAREK FAJK	30
6.4.	JULIA NOWAK	31
6.5.	KRZYSZTOF ZIELIŃSKI.....	31
7.	MAPY EMPATII	32
7.1.	MAREK FAJK	32
7.2.	JULIA NOWAK	32
8.	INTERESARIUSZE	33
8.1.	KTO ZYSKUJE?	33
8.2.	KTO TRACI?	33
9.	PYTANIA HMW	34
9.1.	Pytanie nr 1	34
9.2.	Pytanie nr 2	34
9.3.	Pytanie nr 3	34
9.4.	Pytanie nr 4	35
9.5.	Pytanie nr 5	35
9.6.	Pytanie nr 6	35
10.	IDEACJA I IDENTYFIKACJA	36
10.1.	BURZA MÓZGÓW	36

10.1.1.	INTERAKTYWNE ZABAWKI	36
10.1.2.	MONITOROWANIE AKTYWNOŚCI I ZDROWIA	36
10.1.3.	AUTOMATYCZNE KARMIENIE I NAWADNIANIE	36
10.1.4.	WYKRYWANIE DŹWIĘKÓW ZWIERZĘCIA.....	36
10.2.	IDENTYFIKACJA	36
10.2.1.	INTERAKTYWNA ZABAWKA	37
10.2.2.	OBROŻA MONITORUJĄCA AKTYWNOŚĆ.....	37
11.	PROTOTYPOWANIE URZĄDZENIA GŁÓWNEGO	38
11.1.	OKREŚLENIE WYMAGAŃ I FUNKCJONALNOŚCI URZĄDZENIA.....	38
11.2.	WYBÓR KOMPONENTÓW.....	38
11.2.1.	ESP32-C6 – Mikrokontroler	38
11.2.2.	AKY0393 LiPo 3.7V 2200mAh – Akumulator	39
11.2.3.	BMI160 – Czujnik ruchu IMU (6-osiowy).....	39
11.3.	PIERWSZE URUCHOMIENIE I WSTĘPNE TESTY	40
11.3.1.	PIERWSZE URUCHOMIENIE – PROBLEMY Z ZARZĄDZANIEM ENERGIĄ	40
11.3.2.	PROCES I DIAGNOZOWANIE OPTYMALIZACJI.....	40
11.3.3.	REZULTATY OPTYMALIZACJI	40
11.4.	POŁĄCZENIE Z BAZĄ DANYCH I API.....	40
11.5.	BUDOWA CASE'A	41
11.6.	WYKORZYSTANE METODY.....	41
11.6.1.	PRZETWARZANIE SYGNAŁU DŹWIĘKOWEGO I WYKORZYSTANIE FFT.....	41
11.6.2.	WYKRYWANIE KRZTAŁTU FALI DŹWIĘKOWYCH	41
11.6.3.	AUTOMATYCZNA REKALIBRACJA IMU	42
11.6.4.	AKUMULOWANIE INFORMACJI O RUCHU	42
11.6.5.	INTEGRACJA DANYCH I WYSYŁANIE DO API	42
11.7.	NAJWAŻNIEJSZE FRAGMENTY KODU.....	43
11.7.1.	Połączenie z Wi-Fi.....	43
11.7.2.	Inicjalizacja mikrofonu (przetwarzanie dźwięku)	43
11.7.3.	Inicjalizacja i obsługa IMU (akcelerometr i żyroskop)	43
11.7.4.	Kalibracja żyroskopu	43
11.7.5.	Przetwarzanie danych dźwiękowych z mikrofonu.....	44
11.7.6.	Obliczanie częstotliwości szczytowej.....	44
11.7.7.	Wykrywanie ruchu na podstawie danych z IMU	44

11.7.8.	Wysyłanie danych do serwera (API)	45
11.7.9.	Tryb uśpienia (oszczędność energii)	45
11.8.	OSTATECZNA WERSJA URZĄDZENIA	45
12.	ARCHITEKTURA API.....	49
12.1.	WPROWADZENIE	49
12.1.1.	KLUCZOWE FUNKCJE API	49
12.2.	ARCHITEKTURA SERWERA.....	49
12.2.1.	INFRASTRUKTURA SERWERA.....	50
12.2.2.	BEZPIECZEŃSTWO I MONITORING	50
12.3.	ARCHITEKTURA SYSTEMU.....	51
12.3.1.	WYMAGANE BIBLIOTEKI DO FASTAPI	51
12.3.2.	OPIS API	52
12.4.	ARCHITEKTURA BAZY DANYCH	53
12.5.	ARCHITEKTURA ENDPOINTÓW	54
12.5.1.	KONTROLA ZABAWKI(/toy/*)	54
12.5.2.	ZARZĄDZANIE DANYMI (/data, /events).....	55
12.5.3.	ŚLEDZENIE DYSTANSU (/distance/*).....	55
12.5.4.	ANALITYKA I RAPORTY (/analytics/*, /battery)	56
12.5.5.	KONSERWACJA I DEBUGOWANIE (/maintenance/, /debug/)	56
12.5.6.	KLUCZOWE CECHY ARCHITEKTONICZNE.....	57
12.5.7.	PPRZYKŁAD PRZEPŁYWU DANYCH.....	57
12.6.	NAJWAŻNIEJSZE FRAGMENTY KODU.....	58
12.6.1.	Inicjalizacja Bazy Danych	58
12.6.2.	Modele Pydantic z Walidacjami	59
12.6.3.	Obsługa Stanu Zabawki	59
12.6.4.	Główne Endpointy (Status Zabawki, Włączanie, Sprawdzanie Aktywności)	60
12.6.5.	Endpointy do Zapisania Danych i Sprawdzania Dystansu.....	61
12.6.6.	6. Analytics: Średnia Dzienna, Tygodniowy Dystans, i Analiza Godziny Szczytu	
12.6.6.1.	61	
12.6.7.	7. Generowanie Wykresu z Danych	62
12.6.8.	8. Czyszczenie Bazy Danych (Usuwanie Starych Rekordów).....	63
13.	PROTOTYPOWANIE ZABAWKI	64
13.1.	OKREŚLENIE WYMAGAŃ I FUNKCJONALNOŚCI URZĄDZENIA	64
13.2.	WYBÓR KOMPONENTÓW.....	64

13.3.	PIERWSZE URUCHOMIENIE I TESTY.....	65
13.3.1.	UDOSKONALANIE I DALSZE PRACE NAD PROTOTYPEM	65
13.3.2.	OPRACOWANIE KODU DO ESP 32.....	65
13.4.	OSTATECZNY MODEL URZĄDZENIA	65
13.4.1.	SCHEMAT POŁĄCZEŃ ELEKTRYCZNYCH	66
13.4.2.	BEZPIECZEŃSTWO I UWAGI	67
13.4.3.	INSTRUKCJA UŻYTKOWNIKA.....	67
14.	APLIKACJA MOBILNA	69
14.1.	OKREŚLENIE WYMAGAŃ I FUNKCJONALNOŚCI APLIKACJI	69
14.2.	WYBÓR PLATFORMY	69
14.3.	TWORZENIE INTERFEJSU	69
14.3.1.	PIERWSZE WŁĄCZENIE APLIKACJI W TRYBIE DEBUGOWANIA.....	69
14.3.2.	DOPRACOWYWANIE WYGLĄDU	70
14.3.3.	FINALNA WERSJA INTERFEJSU PetApp	75
14.3.4.	IKONKA APLIKACJI PetApp	77
14.4.	INTEGRACJA Z BACKENDEM	78
14.4.1.	FUNKCJE DO ŁĄCZENIA Z API.....	78
14.4.2.	FUNKCJE DO ZABAWKI Z POZIOMU APLIKACJI	80
14.5.	TESTOWANIE APLIKACJI.....	81
15.	TEST	83
15.1.	ZAŁOŻENIE OBROŻY NA PSA	83
15.2.	WNIOSKI	84
16.	BIBLIOGRAFIA	86
17.	SPIS RYSUNKÓW	88

WSTĘP

Problem: Opieka nad zwierzętami.

Współczesny styl życia wielu właścicieli zwierząt domowych wymaga pozostawiania swoich pupili samych w domu przez znaczną część dnia. Dla niektórych zwierząt, takich jak psy czy koty, długotrwała samotność może powodować stres, lęk separacyjny, a nawet destrukcyjne zachowania, takie jak niszczenie mebli, drapanie ścian czy gryzienie przedmiotów. Problem ten nie tylko generuje straty materialne, ale również negatywnie wpływa na dobrostan psychiczny zwierząt.

W odpowiedzi na ten problem opracowaliśmy prototyp inteligentnego systemu IoT, który pozwala monitorować stan emocjonalny zwierzęcia w czasie rzeczywistym oraz aktywnie angażować je w zabawę podczas nieobecności właściciela. Nasze rozwiązanie opiera się na algorytmach wykrywania niepokoju za pomocą analizy dźwięków, ruchu oraz zachowań zwierzęcia, a następnie automatycznym uruchamianiu interaktywnej zabawki, która pomaga zredukować stres i zapobiec niszczeniu otoczenia.

Projekt ten ma na celu połączenie technologii Internetu Rzeczy z potrzebami opieki nad zwierzętami, dostarczając rozwiązania, które mogą znaczco poprawić jakość życia zarówno zwierząt, jak i ich właścicieli.

Kto był za co odpowiedzialny?

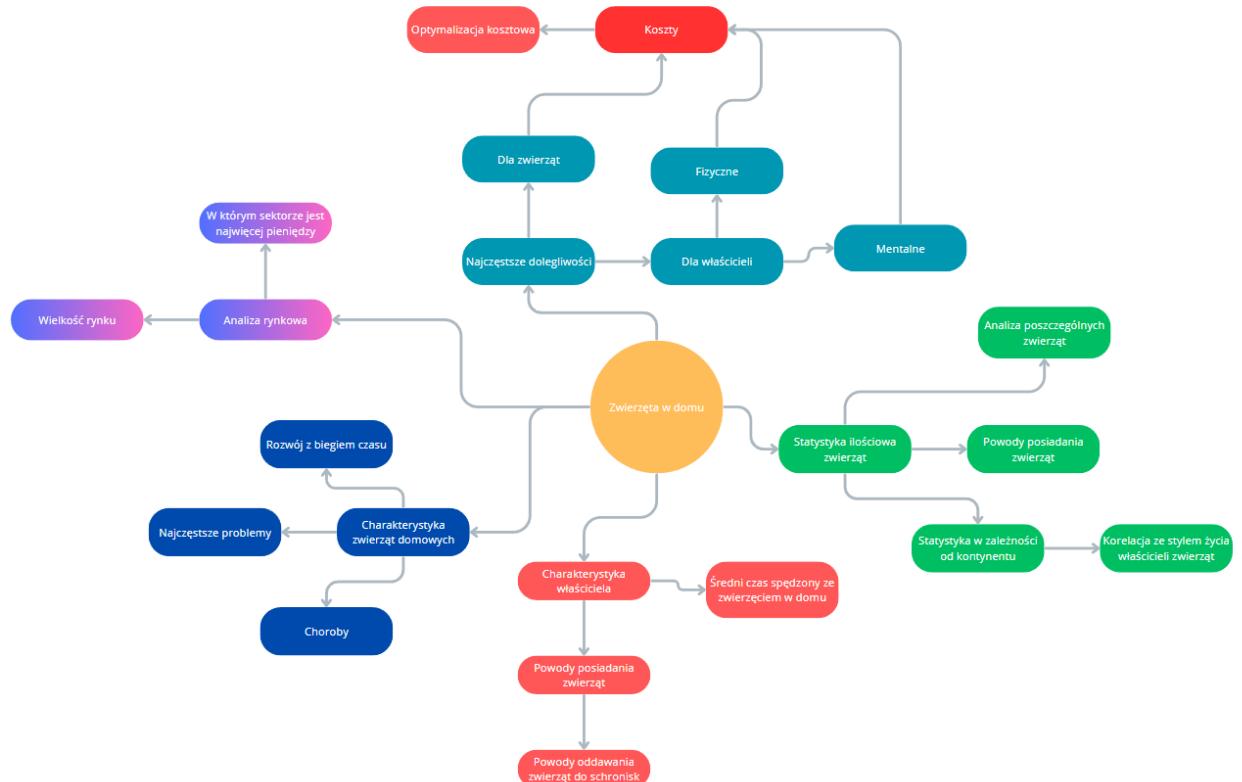
Budowa obroży, stworzenie serwera, zaprojektowanie API, idea: Mikołaj Bednarczyk

Budowa zabawki: Michał Kołodziejczyk oraz Bartłomiej Gromulski

Aplikacja mobilna: Kinga Konieczna oraz Jan Czechowski

1. IDentyfikacja problemu i badania

Podczas planowania research-u w zakresie zwierząt domowych stworzyliśmy mapę skojarzeń przedstawioną na Rys. 1.1 w kontekście problemów/obszarów dotyczących zwierząt zostających samych w domu. Na postawie mapy przenalizowaliśmy zakres problemu, najważniejsze elementy oraz na czym koncentrują się dotychczasowe większe koncerny.

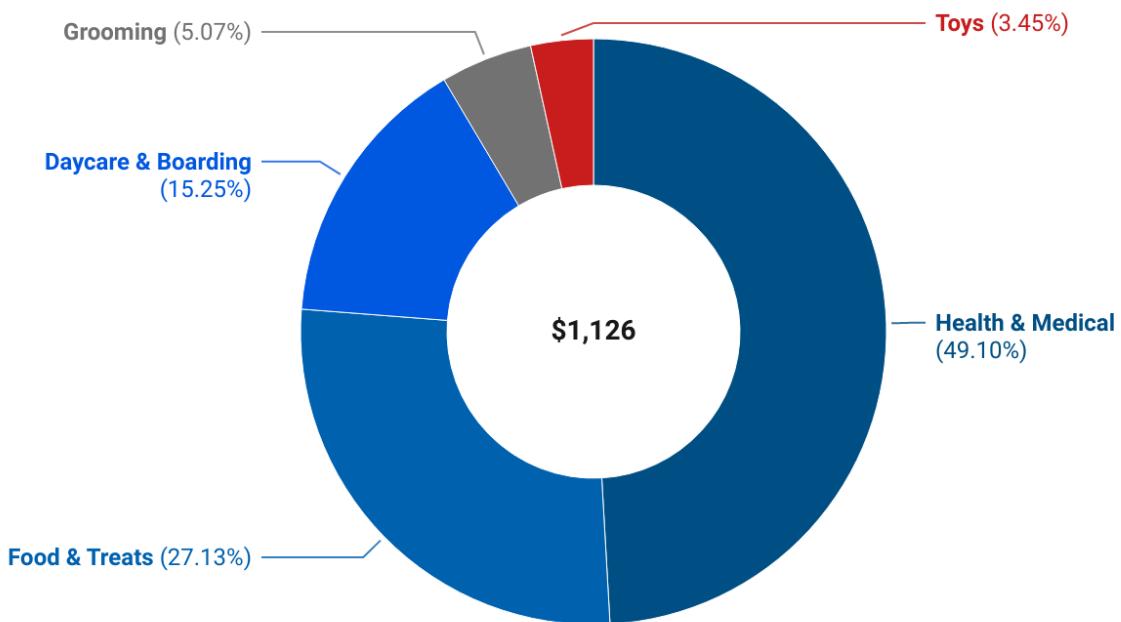


Rys. 1.1. Mapa skojarzeń. [opracowanie własne]

1.1. STATYSTYKI WŁAŚCICIELI ZWIERZĄT

Posiadanie zwierząt domowych to fenomen występujący na całym świecie. Na całym świecie aż **57% ludzi** jest posiadaczami zwierząt domowych. Na Rys. 1.2 dane przedstawiają średnią roczną kwotę wydawaną przez właścicieli zwierząt oraz rozkład wydatków na różne kategorie.

Distribution of annual pet expenses per household



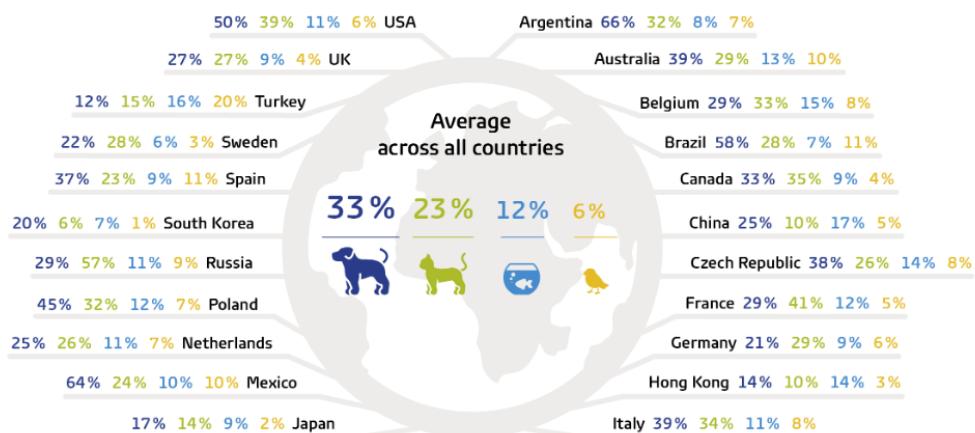
Source: Enterprise Apps Today

Rys. 1.2. Wydatki właścicieli zwierząt. [źródło: Enterprise Apps Today]

Na Rys. 1.3 widać, że zdecydowanie największą grupą zwierząt, które zostają przygarnięte przez właścicieli są psy, na drugim miejscu plasują się koty, a na trzecim rybki.

PET OWNERSHIP INTERNATIONALLY

Percentage of people living with different pets in 22 countries

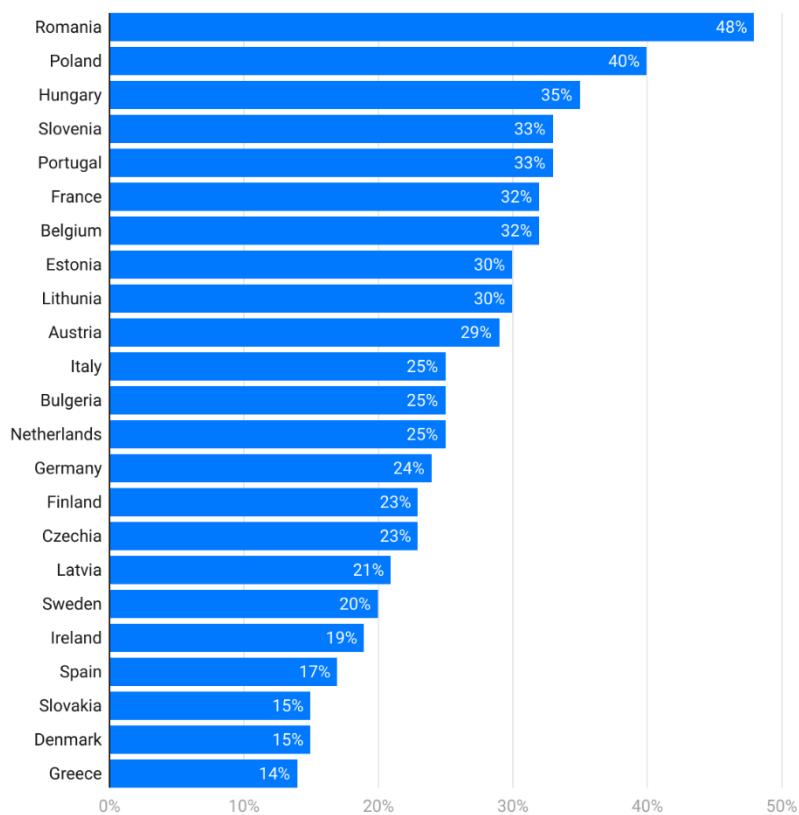


Source: GfK survey among 27,000+ Internet users (ages 15+) in 22 countries – multiple answers possible – rounded

Rys. 1.3. Statystyka przygarniętych zwierząt. [źródło: GfK]

Z Rys. 1.4 wynika, że patrząc na samą Unię Europejską w Rumunii aż 48% rodzin posiada przynajmniej jednego kota, Polska znajduje się na drugim miejscu (40%).

Share of households owning at least one cat in the European Union in 2022, by country



Source: Enterprise Apps Today

Rys. 1.4. Statystyka posiadania min. 1 kota. [źródło: Enterprise Apps Today]

1.2. CHARAKTERYSTYKA WŁAŚCICIELI

Poniżej przedstawiamy charakterystykę właścicieli zwierząt.

1. Priorytetowe traktowanie potrzeb zwierząt: Odpowiedzialni właściciele zwierząt stawiają potrzeby swoich pupili ponad swoimi własnymi, zapewniając im bezpieczne środowisko i właściwą opiekę, w tym regularne wizyty weterynaryjne i profilaktykę zdrowotną.
2. Zarządzanie czasem: Właściciele wygospodarowują codziennie czas na karmienie, ćwiczenia, trening i nawiązywanie więzi, co pomaga stworzyć rutynę, z której zwierzęta korzystają.
3. Zaangażowanie w szkolenie: Odpowiedzialni właściciele inwestują czas w szkolenie swoich pupili, wspierając ich dobre zachowanie i zwiększając ich pewność siebie.
4. Świadomość opieki profilaktycznej: Rozumieją znaczenie szczepień, opieki dentystycznej i kontroli pasożytów dla utrzymania zdrowia swojego zwierzaka.

5. Gotowość finansowa: Właściciele ci planują wydatki weterynaryjne, oszczędzając lub kupując ubezpieczenie dla zwierząt, wykazując się przezornością w zakresie odpowiedzialności finansowej.
6. Urozmaicenie otoczenia: Zapewniają zabawki i zajęcia, które stymulują ich zwierzęta psychicznie i fizycznie, przyczyniając się do ogólnego szczęścia.
7. Dbałość o dietę: Odpowiedzialni właściciele dbają o dietę swoich zwierząt, wybierając karmę dostosowaną do wieku i ograniczając smakołyki.
8. Bezpieczne przestrzenie do odpoczynku: Zapewniają swoim pupilom wygodne i ciche miejsca do odpoczynku, szanując ich potrzebę przestrzeni osobistej.
9. Wrażliwość na stres: Unikają zmuszania zwierząt do stresujących sytuacji, pozwalając im na stopniową aklimatyzację w nowym środowisku.
10. Codzienna interakcja: Angażowanie się w codzienną pielęgnację, zabawę i przywiązywanie ma kluczowe znaczenie dla budowania silnych więzi ze zwierzętami.

Stworzyliśmy również kilka typów właścicieli psów. Każdy typ właściciela posiada drastycznie inne priorytety oraz światopogląd.

- Młodzi i entuzjastyczni właściciele, którzy aktywnie uczą się, jak skutecznie dbać o swoje zwierzęta.
- Właściciele, którzy postrzegają swoje zwierzęta jako akcesoria do stylu życia, często przedkładając wygląd nad zdrowie.
- Doświadczeni właściciele zwierząt domowych, którzy traktują swoje zwierzęta jak członków rodziny i utrzymują ustrukturyzowane rutyny skoncentrowane na zdrowiu i dobrym samopoczuciu.
- Pragmatyczni właściciele, którzy rozumieją podstawowe potrzeby swoich zwierząt, ale mogą nie angażować się głęboko w emocjonalne lub estetyczne aspekty ich posiadania.

Rozdzieliśmy również właścicieli na różne podkategorie w zależności od posiadanego przez nich zwierzęcia.

- **Właściciele psów:** Ogólnie bardziej ekstrawertyczni, chętni do współpracy, empatyczni i odpowiedzialni. Mają tendencję do bycia odpornymi jednostkami, które lubią interakcje społeczne. Posiadanie psa często wiąże się ze zmniejszonym poczuciem samotności.
- **Właściciele kotów:** Zazwyczaj bardziej introwertyczni i wrażliwi, właściciele kotów mogą wykazywać wyższy poziom neurotyczności, ale także kreatywności i niezależności. Często preferują spokojniejsze otoczenie i mniej wymagające zajęcia.
- **Właściciele innych zwierząt:** Właściciele ryb są uważani za najszczęśliwszych wśród właścicieli zwierząt domowych; właściciele gadów są bardziej niezależni; podczas gdy właściciele małych ssaków często wykazują cechy opiekuńcze.

1.3. CHARAKTERYSTYKA ZWIERZĄT

Na początek badań stworzyliśmy charakterystykę najbardziej popularne zwierzęta.

1.3.1. CHARAKTERYSTYKA PSÓW

Cechy:

Towarzyskie i lojalne: Psy są zwierzętami stadnymi i rozwijają się dzięki interakcjom społecznym z ludźmi i innymi psami.

Różnorodność ras: Istnieją setki ras, z których każda ma odmienny temperament, rozmiary i potrzeby ruchowe.

Możliwość szkolenia: Większość psów dobrze reaguje na szkolenie, dzięki czemu nadają się do różnych ról, od towarzystwa po zwierzęta służbowe.

Problemy:

Problemy behawioralne: Problemy takie jak lęk separacyjny, agresja lub nadmierne szczekanie mogą pojawić się, jeśli nie są odpowiednio kontrolowane.

Problemy zdrowotne: Typowe problemy zdrowotne obejmują otyłość, dysplazję stawów biodrowych i problemy stomatologiczne. Regularne kontrole weterynaryjne są niezbędne.

Potrzeby ruchowe: Wiele ras wymaga codziennych ćwiczeń; brak aktywności fizycznej może prowadzić do otyłości i problemów behawioralnych.

1.3.2. CHARAKTERYSTYKA KOTÓW

Cechy charakterystyczne:

Niezależność: Koty są generalnie bardziej samowystarczalne niż psy, często wymagają mniej uwagi.

Terytorialne: Mogą być wrażliwe na zmiany w otoczeniu i stresować się nowymi zwierzętami lub przeprowadzką.

Zabawny: Koty lubią się bawić, zwłaszcza interaktywnymi zabawkami, które stymulują ich instynkt łowiecki.

Problemy:

Problemy z kuwetą: Problemy z korzystaniem z kuwety mogą wskazywać na problemy zdrowotne lub stres.

Agresja lub niepokój: Niektóre koty mogą wykazywać agresję wobec ludzi lub innych zwierząt domowych z powodu strachu lub sporów terytorialnych.

Zagrożenia dla zdrowia: Typowe problemy zdrowotne obejmują otyłość, choroby zębów i problemy z układem moczowym.

1.3.3. CHARAKTERYSTYKA RYBEK

Cechy charakterystyczne:

Niskie koszty utrzymania: Generalnie wymagają mniej codziennej opieki w porównaniu z innymi zwierzętami domowymi; jednak nadal potrzebują odpowiednich warunków w zbiorniku.

Różnorodność gatunków: Różne gatunki mają różne wymagania dotyczące temperatury wody, poziomu pH i partnerów w zbiorniku.

Atrakcyjność wizualna: Zbiorniki z rybami mogą być estetyczne i uspokajające.

Zagadnienia i problemy:

Zarządzanie jakością wody: Zła jakość wody może prowadzić do problemów zdrowotnych ryb; regularne testy są niezbędne.

Ograniczona interakcja: Ryby nie zapewniają takiego samego poziomu towarzystwa jak inne zwierzęta domowe, co może prowadzić do tego, że niektórzy właściciele czują się odłączeni.

Zarządzanie chorobami: Ryby są podatne na choroby, takie jak ich (choroba białych plam), które wymagają szybkiego leczenia.

1.3.4. CHARAKTERYSTYKA PTAKÓW

Cechy charakterystyczne:

Wokalizacja: Ptaki mogą być bardzo wokalne; gatunki takie jak papugi mogą naśladować dźwięki i mowę.

Inteligencja: Ptaki takie jak papugi są znane ze swojej inteligencji i umiejętności rozwiązywania problemów.

Problemy:

Problemy behawioralne: Ptaki mogą rozwinąć problemy behawioralne, takie jak wyrywanie piór, jeśli są znudzone lub zestresowane.

Potrzeby żywieniowe: Wymagają zbilansowanej diety, która obejmuje nasiona, owoce, warzywa i specjalistyczne granulki; zaniedbanie tego może prowadzić do problemów zdrowotnych.

Wymagania przestrzenne: Ptaki potrzebują dużo miejsca do latania; niewystarczająca przestrzeń może prowadzić do stresu i problemów zdrowotnych.

2. ANALIZA RYNKU

Rynek zwierząt w ostatnich latach gwałtownie rośnie, może być to związane z post-pandemiczną izolacją lub generalną chęcią posiadania przyjaciela.

2.1. PRZEGŁĄD RYNKU

Wartość globalna: Oczekuje się, że globalny przemysł zoologiczny wzrośnie z około 320 miliardów dolarów w 2023 roku do prawie 500 miliardów dolarów do 2030 roku, odzwierciedlając silną ekspansję napędzaną rosnącą liczbą zwierząt domowych i zwiększymi wydatkami na opiekę nad nimi.

Segment opieki nad zwierzętami domowymi: Rynek opieki nad zwierzętami domowymi, który obejmuje produkty i usługi dla zwierząt domowych, został wyceniony na około 304,4 miliarda dolarów w 2023 roku i przewiduje się, że będzie rósł przy CAGR na poziomie 6,8%, osiągając około 545,7 miliarda dolarów do 2032 roku. Segment ten obejmuje różne kategorie, takie jak żywność, produkty zdrowotne, pielęgnacja i usługi szkoleniowe.

Rynek karmy dla zwierząt domowych: Sektor karmy dla zwierząt domowych jest szczególnie godny uwagi, a prognozy wskazują, że może przekroczyć 135 miliardów dolarów do 2030 roku, napędzany rosnącą preferencją dla produktów premium i produktów zdrowotnych. Segment ten charakteryzuje się wysoką lojalnością konsumentów i stałymi wzorcami wydatków.

2.2. INFORMACJE REGIONALNE

Stany Zjednoczone: Stany Zjednoczone pozostają największym rynkiem dla zwierząt domowych, z oczekiwaniemi osiągnięcia około 200 miliardów dolarów do 2030 roku. Przewiduje się, że w 2024 r. wartość amerykańskiego rynku zwierząt domowych wyniesie 80,69 mld USD i wzrośnie do około 103,27 mld USD do 2029 r. przy CAGR na poziomie 5,06%. Czynniki przyczyniające się do tego wzrostu obejmują wysokie wskaźniki posiadania zwierząt domowych (66% gospodarstw domowych) i znaczący trend w kierunku traktowania zwierząt domowych jako członków rodziny.

Azja i Pacyfik: Kraje takie jak Chiny, Japonia i Indie są rynkami wschodząącymi o znacznym potencjale wzrostu ze względu na rosnącą urbanizację i dochody rozporządzalne. Region Azji i Pacyfiku jest świadkiem gwałtownego wzrostu posiadania zwierząt domowych, szczególnie wśród młodszych grup demograficznych.

2.3. KLUCZOWE TRENDY NAPĘDZAJĄCE WZROST

Humanizacja zwierząt domowych: Właściciele zwierząt domowych coraz częściej postrzegają swoje zwierzęta jako członków rodziny, co prowadzi do wyższych wydatków na produkty i usługi premium, które poprawiają ich samopoczucie. Trend ten jest widoczny w rosnącym popycie na karmy dla smakoszy, luksusowe akcesoria i specjalistyczną opiekę zdrowotną.

Ekspansja handlu elektronicznego: Rozwój internetowych platform zakupowych zmienił sposób, w jaki produkty dla zwierząt domowych są wprowadzane na rynek i sprzedawane. Oczekuje się, że sprzedaż e-commerce w USA podwoi się do 2030 roku, odzwierciedlając zmieniające się preferencje konsumentów w kierunku wygody.

Wydatki na opiekę zdrowotną: Obserwuje się znaczny wzrost wydatków na opiekę weterynaryjną i produkty związane ze zdrowiem, ponieważ zwierzęta domowe żyją dłużej dzięki postępowi w opiece zdrowotnej. Obejmuje to inwestycje w diagnostykę profilaktyczną i farmaceutykę.

Zrównoważony rozwój i produkty etyczne: Konsumenti są coraz bardziej zainteresowani etycznymi i zrównoważonymi produktami dla swoich zwierząt domowych, co kształtuje ofertę produktów w całej branży.

3. KOTY I PSY

Zarówno koty, jak i psy – jedne z najczęściej wybieranych zwierząt domowych – mogą cierpieć na różnorodne dolegliwości zdrowotne i zachowania problematyczne. Choroby skóry, alergie, zaburzenia układu pokarmowego czy trudności w zachowaniu, takie jak nadmierna lękliwość lub agresja, wymagają odpowiedniej uwagi i postępowania. Podobieństwa między gatunkami wskazują, że zarówno opiekunowie kotów, jak i psów powinni być przygotowani na właściwe zarządzanie zdrowiem i zachowaniem swoich podopiecznych.

3.1. PROBLEMY ZDROWOTNE ZWIĄZANE Z POSIADANIEM KOTA

3.1.1. CHOROBY UKŁADU POKARMOWEGO

Choroby układu pokarmowego to jedne z najczęstszych problemów zdrowotnych, które mogą występować zarówno u kotów, jak i u psów. W przypadku kotów często spotykane są wymioty, biegunki oraz zaparcia, których przyczyną bywa nieodpowiednia dieta, a także infekcje bakteryjne czy pasożytnicze. Podobnie u psów, zaburzenia żołądkowo-jelitowe, takie jak wymioty czy biegunki, mogą wynikać z niekorzystnego składu karmy, skażenia bakteryjnego lub obecności pasożytów w przewodzie pokarmowym.

Właściciele zarówno kotów, jak i psów powinni zwracać szczególną uwagę na jakość i dopasowanie diety do potrzeb danego zwierzęcia, zapewniać stały dostęp do świeżej wody i regularnie kontrolować stan zdrowia pupila. W razie pojawienia się niepokojących objawów konieczna jest konsultacja z weterynarzem, który zleci odpowiednie badania (np. badanie kału, krwi) i w razie potrzeby wdroży właściwe leczenie, obejmujące zmianę diety, stosowanie leków lub suplementów. Odpowiednia profilaktyka i szybka reakcja na ewentualne problemy mogą znacząco poprawić jakość i długość życia zwierzęcia.

3.1.2. PROBLEMY Z UKŁADEM MOCZOWYM

Problemy z układem moczowym to kolejna powszechna grupa schorzeń występujących zarówno u kotów, jak i u psów. Szczególnie narażone mogą być zwierzęta wykastrowane, u których częściej dochodzi do rozwoju chorób nerek, zapalenia pęcherza moczowego czy powstawania kamieni nerkowych. Objawy u obu gatunków mogą obejmować częste oddawanie moczu, trudności w jego oddawaniu, krwiomocz, a także zmiany w zachowaniach związanych z załatwianiem potrzeb fizjologicznych (np. unikanie kuwety u kotów czy niepokój przed oddawaniem moczu u psów).

Właściciele kotów i psów powinni zwracać szczególną uwagę na dietę swoich pupili oraz zapewnić im stały dostęp do świeżej wody, aby zapobiegać rozwojowi chorób układu moczowego. W przypadku podejrzenia problemów u któregoś z tych zwierząt, niezbędna jest konsultacja z weterynarzem. Diagnostyka obejmuje badania laboratoryjne (analiza moczu) oraz badania obrazowe. Leczenie zazwyczaj polega na stosowaniu odpowiednich antybiotyków, środków przeciwzapalnych, a czasem także specjalistycznej diety. Istotne jest również zapewnienie psu oraz kotu właściwych warunków higienicznych i możliwość regularnego opróżniania pęcherza, co sprzyja profilaktyce i poprawie jakości życia zwierzęcia.

3.1.3. ZABURZENIA NEUROLOGICZNE

Zaburzenia neurologiczne mogą występować zarówno u kotów, jak i u psów, obejmując szerokie spektrum chorób wpływających na funkcjonowanie układu nerwowego. Wśród najczęstszych objawów można wymienić drgawki, zaburzenia koordynacji ruchów, zmiany w zachowaniu oraz dezorientację. U kotów często spotyka się schorzenia takie jak toksoplazmoza czy zapalenie mózgu, natomiast u psów mogą pojawiać się m.in. padaczka idiopatyczna, wirusowe choroby neurologiczne (np. nosówka) oraz problemy wynikające z degeneracyjnych zmian kręgosłupa czy rdzenia kręgowego (np. zespół sztywnego karku – tzw. wobbler syndrome).

W przypadku podejrzenia zaburzeń neurologicznych u kota lub psa, konieczna jest niezwłoczna konsultacja z weterynarzem. Specjalista przeprowadzi odpowiednią diagnostykę (badania krwi, obrazowe, czasem testy płynu mózgowo-rdzeniowego), a następnie wdroży właściwe leczenie. Szybka reakcja opiekuna jest kluczowa, ponieważ wiele chorób neurologicznych może postępować i prowadzić do trwałych uszkodzeń układu nerwowego lub poważnego pogorszenia jakości życia zwierzęcia.

3.2. PROBLEMY BEHAVIORALNE

3.2.1. AGRESJA

Agresja zarówno u psów, jak i u kotów może występować w różnych formach, w tym jako agresja terytorialna, agresja lękowa, czy agresja wynikająca z frustracji.

Koty mogą wykazywać agresywne zachowanie w stosunku do innych zwierząt lub ludzi, co może prowadzić do problemów w domu. Właściciele kotów powinni zwrócić uwagę na te zachowania i, jeśli to konieczne, skonsultować się z weterynarzem lub behawiorystą zwierząt, aby zidentyfikować przyczyny agresji i wdrożyć odpowiednie metody zarządzania.

W przypadku wystąpienia agresywnych zachowań u psa, należy zwrócić uwagę na kontekst, w którym się pojawiają, oraz ich częstotliwość i intensywność. Podobnie jak u kotów, warto skonsultować się ze specjalistą – lekarzem weterynarii lub behawiorystą – który pomoże ustalić przyczyny agresji i opracować strategię postępowania, uwzględniającą modyfikację zachowania, szkolenie, a czasem również terapię farmakologiczną.

3.2.2. ZACHOWANIA ZWIĄZANE Z LĘKIEM

Koty mogą przejawać różne formy lęku, które objawiają się niepokojem, drapaniem lub nadmiernym miauczeniem w czasie nieobecności właściciela. Lęk może również prowadzić do agresji lub unikania kontaktu z ludźmi i innymi zwierzętami. W takich przypadkach ważne jest zrozumienie przyczyn lęku i wdrożenie odpowiednich technik zarządzania, takich jak stopniowe przyzwyczajanie kota do pozostawania samodzielnie w domu.

Psy także mogą cierpieć z powodu lęku związanego z rozłąką, co często objawia się niepokojem, niszczeniem przedmiotów, uporczywym szczekaniem czy wyciem podczas nieobecności opiekuna. U niektórych psów lęk może przerodzić się w zachowania agresywne lub powodować unikanie kontaktu z innymi zwierzętami i ludźmi. Istotne jest, aby właściciele psów

zrozumieli źródło lęku i skorzystali z właściwych metod, takich jak systematyczna nauka samodzielności, stopniowe wydłużanie czasu pozostawania zwierzęcia samego czy wprowadzenie dodatkowych bodźców środowiskowych (zabawek interaktywnych, mat węchowych). W razie potrzeby warto skonsultować się z behawiorystyą zwierzęcym, który pomoże dobrać odpowiednie strategie i ewentualne wsparcie farmakologiczne.

3.3. LĘK SEPARACYJNY U KOTÓW I PSÓW

Lęk separacyjny to problem, który dotyczy zarówno kotów, jak i psów, i ma istotny wpływ na ich zachowanie oraz dobrostan psychiczny. Choć koty są często postrzegane jako zwierzęta niezależne, wiele badań wykazuje, że potrafią one tworzyć silne więzi emocjonalne z właścicielami i źle znoszą ich nieobecność. Podobnie jak u psów, lęk separacyjny u kotów może prowadzić do problematycznych zachowań, pogorszenia zdrowia i napiętych relacji z opiekunem.

U psów lęk separacyjny objawia się najczęściej uporczywym szczeakiem, skomleniem, niszczeniem przedmiotów czy załatwianiem potrzeb fizjologicznych w domu podczas nieobecności opiekuna. Jako zwierzęta stadne, psy silnie przywiązuje się do właścicieli i mają trudności z rozłąką. Zbyt gwałtowne zmiany w harmonogramie dnia, brak stopniowego przyzwyczajania do samotności oraz wcześniejsze złe doświadczenia (np. z schroniska, porzucenie) mogą nasilać lęk.

Aby złagodzić lęk separacyjny u kotów i psów, skuteczne są strategie takie jak stopniowe treningi pozostawania samemu, odpowiednia stymulacja środowiskowa (np. zabawki interaktywne, maty węchowe) oraz, w razie potrzeby, konsultacja z behawiorystyą lub weterynarzem.

3.3.1. PRZYCZYNY LĘKU SEPARACYJNEGO U KOTÓW I PSÓW

3.3.1.1. ZMIANY W ŚRODOWISKU I RUTYNIE

Zarówno u kotów, jak i psów, nawet drobne zmiany w otoczeniu, takie jak przestawienie mebli, zmiana harmonogramu karmienia czy brak stałej obecności opiekuna, mogą wywoływać stres. Szczególnie wrażliwe na takie zmiany są zwierzęta, które nie doświadczyły wcześniej samotności.

3.3.1.2. DOŚWIADCZENIA TRAUMATYCZNE

Koty i psy z trudną przeszłością, taką jak porzucenie, przemoc czy pobyt w schronisku, mają wyższe ryzyko rozwoju lęku separacyjnego. Nieobecność opiekuna może przypominać im wcześniejsze negatywne doświadczenia, co prowadzi do silnego niepokoju.

3.3.1.3. BRAK SOCJALIZACJI W MŁODYM WIEKU

Zwierzęta, które w okresie socjalizacji nie miały kontaktu z różnymi ludźmi czy innymi zwierzętami, mogą być bardziej przywiązane do jednej osoby i odczuwać silny stres podczas jej nieobecności. Dotyczy to zarówno kotów, jak i psów.

3.3.1.4. STRATA TOWARZYSZA

Zarówno koty, jak i psy tworzą silne więzi z innymi zwierzętami lub ludźmi w gospodarstwie domowym. Utrata towarzysza może zwiększać ich zależność od opiekuna, potęgując lęk separacyjny i pogłębiając niepokój, szczególnie jeśli były silnie przywiązane.

3.3.2. OBJAWY LĘKU SEPARACYJNEGO U PSÓW I KOTÓW

Lęk separacyjny u kotów przejawia się zarówno w zmianach behawioralnych, jak i fizycznych.

3.3.2.1. PROBLEMY BEHAWIORALNE

- Nadmierne wokalizowanie – Zarówno u kotów, jak i psów, lęk separacyjny może objawiać się intensywnym miauczeniem, szczeżaniem, płaczem czy skomleniem podczas nieobecności opiekuna. Wokalizacja zazwyczaj ma charakter intensywny i występuje tylko w takich sytuacjach, gdy zwierzę czuje się samotne.
- Destrukcyjne zachowania – W odpowiedzi na stres, zarówno koty, jak i psy mogą niszczyć różne przedmioty w domu, np. drapać meble, gryźć kable, przewracać przedmioty czy niszczyć zabawki. Takie zachowania wynikają z prób radzenia sobie z frustracją lub z potrzeby „ucieczki” z zamkniętej przestrzeni.
- Załatwianie się poza kuwetą (u kotów) lub w nietypowych miejscach (u psów) – U kotów jednym z najbardziej charakterystycznych objawów lęku jest załatwianie się poza kuwetą, np. na łóżku właściciela lub ubraniach. U psów może to objawiać się załatwianiem się w domu, mimo że są nauczone załatwiać na zewnątrz. Jest to wyraz niepokoju lub chęci przywrócenia znanego zapachu właściciela.
- Nadmierne domaganie się uwagi - po powrocie właściciela, zarówno koty, jak i psy mogą wykazywać nadmierną potrzebę kontaktu. Koty mogą podążać za opiekunem po domu, ocierać się lub wspinać na kolana, a psy często skaczą, liżą twarz lub biegą do drzwi. Te zachowania są wynikiem niepewności i potrzeby bliskości.
- Ukrywanie się lub unikanie kontaktu - Niektóre koty i psy, zamiast szukać bliskości, mogą reagować odwrotnie. Ukrywają się lub unikają kontaktu po powrocie właściciela. Może to wynikać z przedłużającego się stanu stresu i potrzeby „odizolowania się” w celu uspokojenia.

3.3.2.2. FIZYCZNE SKUTKI STRESU

- Zarówno u kotów, jak i psów, stres separacyjny może prowadzić do problemów trawiennych, takich jak wymioty, biegunka, zaparcia czy utrata apetytu. Te dolegliwości nie mają często wyraźnej przyczyny medycznej, a wynikają z reakcji psychosomatycznych związanych z lękiem i stresem.
- Zmniejszenie apetytu lub nadmierne łaknienie - U psów i kotów stres separacyjny może zaburzać ich łaknienie. Niektóre zwierzęta przestają jeść w czasie nieobecności opiekuna, inne mogą reagować kompulsywnym jedzeniem, próbując w ten sposób złagodzić stres.
- Utrata sierści (u kotów) i nadmierne lizanie (u psów) – Koty cierpiące na lęk separacyjny często kompulsywnie wylijają swoją sierść, co może prowadzić do łysienia lub powstawania ran na skórze. U psów, stres może objawiać się nadmiernym lizaniem łap, ciała lub innych części ciała, co może prowadzić do podrażnień skóry i wyłysienia.

3.3.3. STRATEGIE RADZENIA SOBIE Z LĘKIEM SEPARACYJNYM

3.3.3.1. TWORZENIE BEZPIECZNEGO MIEJSCA

U kotów stworzenie przytulnego, osobistego kącika z legowiskiem i przedmiotami pachnącymi opiekunem (np. koszulką) może pomóc w obniżeniu stresu. U psów, które również potrzebują bezpiecznej strefy, świetnie sprawdzają się klatki kennelowe (odpowiednio wprowadzone i kojarzone z pozytywnymi bodźcami) lub wydzielone, ciche miejsca w domu, gdzie pies czuje się bezpiecznie.

3.3.3.2. STOPNIOWE PRZYZWYCZAJANIE KOTA DO SAMOTNOŚCI

U kotów zaleca się krótkie rozstania i stopniowe ich wydłużanie, nagradzając za spokojne zachowanie. Podobną technikę stosuje się u psów – właściciel zaczyna od wyjścia na chwilę za drzwi, by stopniowo zwiększać czas nieobecności. Po powrocie warto unikać zbyt emocjonalnych powitań, aby nie wzmacniać nadmiernego pobudzenia związanego z powrotem opiekuna.

3.3.3.3. UŻYCIE FEROMONÓW

Feromony (dla kotów np. Feliway, dla psów – Adaptil) pomagają stworzyć spokojną atmosferę. U psów stosowanie feromonów w formie obroży czy dyfuzorów może znacznie obniżyć poziom lęku i poprawić samopoczucie zwierzęcia w trakcie nieobecności opiekuna.

3.3.3.4. INTERAKTYWNE ZABAWKI I STYMULACJA

Koty można zająć zabawkami logicznymi, drapakami czy półkami do wspinaczki. U psów dobrym rozwiązaniem są interaktywne zabawki, maty węchowe, a także gryzaki, które zapewniają zajęcie i pozwalają na rozładowanie stresu. Zarówno kotom, jak i psom warto zostawić coś do robótki podczas nieobecności opiekuna, aby zapobiec znudzeniu i związanym z nim zachowaniom destrukcyjnym.

3.3.3.5. KONSULTACJA WETERYNARYJNA

Jeśli objawy lęku separacyjnego są nasilone, warto skonsultować się z weterynarzem lub behawiorystą. U kotów i psów specjalista może zalecić terapie behawioralne, trening habituacji, a w niektórych przypadkach leki uspokajające. Behawiorysta podpowie również, jak wdrażać strategie zaradcze i oceni poziom stresu zwierzęcia.

3.4. WAŻNE ASPEKTY W OPIECE NAD KOTEM

Opieka nad kotem lub psem wymaga uwzględnienia kilku kluczowych aspektów, takich jak zdrowie fizyczne, komfort psychiczny oraz organizacja życia codziennego w sposób, który sprzyja dobrostanowi naszego zwierzęcia. Na podstawie dostępnych materiałów omówimy najważniejsze praktyki w tym zakresie.

3.4.1. ZDROWIE ZWIERZĘCIE

Regularne wizyty u weterynarza, szczepienia, badania kontrolne i profilaktyka pasożytów są kluczowe zarówno u kotów, jak i u psów. Odpowiednia dieta (dla kotów bogata w wysokiej jakości białko, dla psów zbilansowana zgodnie z potrzebami rasy, wieku i stanu zdrowia) oraz właściwe nawodnienie odgrywają istotną rolę w utrzymaniu zdrowia fizycznego.

3.4.2. DOBROSTAN PSYCHICZNY

Koty, jako zwierzęta terytorialne, potrzebują stabilności i poczucia bezpieczeństwa. Psy, istoty społeczne, wymagają interakcji z właścicielem i odpowiedniej stymulacji umysłowej. W obu przypadkach stymulacja intelektualna oraz odpowiednie warunki środowiskowe zapobiegają nudzie i destrukcyjnym zachowaniom.

3.4.3. OPIEKA PRZY PEŁNOETATOWEJ PRACY

Przy długiej nieobecności opiekuna ważne jest zapewnienie zwierzęciu możliwości spędzania czasu samodzielnie w komfortowych warunkach. Automatyczne karmniki, fontanny z wodą i odpowiednia liczba kuwet lub wyjść na spacer z zaufaną osobą (dla psów) minimalizują stres. Po powrocie do domu warto poświęcić zwierzęciu czas na zabawę i głaskanie, aby utrzymać więź emocjonalną.

3.4.4. ZROZUMIENIE JĘZYKA CIAŁA KOTA PRZEZ WŁAŚCICIELA

Zarówno u kotów, jak i u psów, zrozumienie sygnałów wysyłanych przez zwierzę jest kluczowe do uniknięcia niepotrzebnego stresu. Pozycja uszu, ogona, postawa ciała oraz wokalizacja informują o nastroju i potrzebach zwierzęcia.

3.4.5. SOCJALIZACJA I INTERAKCJE SPOŁECZNE

3.4.5.1. RELACJE Z INNYMI ZWIERZĘTAMI

Wprowadzanie nowego kota czy psa do domu wymaga cierpliwości i stopniowego przyzwyczajania. Zarówno koty, jak i psy potrzebują czasu, aby zaakceptować nowych członków stada. Psy nierzadko łatwiej adaptują się do nowych towarzyszy, koty mogą wymagać dłuższego procesu socjalizacji.

3.4.5.2. KONTAKT Z CZŁOWIEKIEM

Koty, choć niezależne, cenią sobie interakcję z opiekunem. Psy zazwyczaj jeszcze mocniej potrzebują kontaktu z człowiekiem, co oznacza, że regularna zabawa, wspólne spacery czy treningi są nieodzowne do utrzymania ich dobrostanu psychicznego.

3.4.6. PRZYGOTOWANIE NA SYTUACJE AWARYJNE

Właściciele kotów i psów powinni być przygotowani na nagłe sytuacje, takie jak choroba zwierzęcia czy nagła zmiana środowiska. Podstawowe leki, kontakt do weterynarza i znajomość objawów wskazujących na poważne problemy zdrowotne są niezbędne w zapewnieniu bezpieczeństwa i dobrostanu zwierzęcia.

3.5. BADANIE RAS

Wielu badaczy i hodowców zauważało, że temperament zwierząt różni się w zależności od rasy, co może wpływać na to, jak reagują one na sytuacje takie jak zostanie same w domu.

3.5.1. RASY KOTÓW

Oto kilka przykładów ras, które różnią się pod względem poziomu spokoju i aktywności:

1. **Bengalskie koty** są znane ze swojej nadmiernej aktywności. Mają energię porównywalną do dzikich kotów, co sprawia, że wymagają dużo uwagi i interakcji, by zaspokoić swoje potrzeby. Są również bardzo ciekawskie, co może prowadzić do frustracji, gdy zostaną same w domu.
2. **Persy i koty egzotyczne krótkowłose** są natomiast znane ze swojej łagodnej, spokojnej natury. Są bardziej skłonne do relaksu i cieszenia się chwilami ciszy, co czyni je mniej podatnymi na objawy lęku separacyjnego.
3. **Maine Coon** to rasa kotów dużych, które mimo swojej wielkości charakteryzują się bardzo spokojnym i łagodnym temperamentem. Są to koty towarzyskie, które dobrze radzą sobie w domu z innymi zwierzętami, a ich odporność na stres sprawia, że nie mają problemów z byciem samotnym przez dłuższy czas.
4. **Siamese i Balinese** to koty bardziej towarzyskie, które mogą czuć się niekomfortowo, gdy są same przez dłuższy czas. Często angażują się w nadmierne działania, takie jak nadmierne czyszczanie, gdy są zestresowane.

Rasy takie jak **Chartreux** i **British Shorthair** są zazwyczaj spokojniejsze, preferując życie w komfortowym, mniej stresującym środowisku. Koty te nie przejawiają tendencji do nadmiernej aktywności i mogą lepiej tolerować czas spędzony samodzielnie w domu. Rasa ma zatem duży wpływ na temperament kota i jego reakcje na sytuacje takie jak lęk separacyjny. Istnieją różnice w poziomie aktywności i potrzeby interakcji, które mogą wpływać na to, jak koty radzą sobie z pozostawaniem samodzielnie w domu.

3.5.2. RASY PSÓW

Badania nad rasami psów wykazują, że różne rasy mają tendencję do wykazywania specyficznych zachowań, które wynikają z ich historii hodowlanej i przeznaczenia. Na przykład, rasy pasterskie, jak border collie, są zazwyczaj bardzo energiczne i łatwe do szkolenia, podczas gdy rasy myśliwskie, takie jak labrador retriever, mogą mieć silniejszy instynkt tropienia. Te różnice w zachowaniach mogą wpływać na sposób interakcji psów z ludźmi, innymi zwierzętami oraz ich zdolność do przystosowywania się do różnych środowisk.

3.6. RÓŻNICA ZACHOWAŃ W ZALEŻNOŚCI OD PŁCI

3.6.1. RÓŻNICE W REAKCJI NA STRES

Samce kotów, zwłaszcza niezadowolone z sytuacji (np. odizolowanie ich na długi czas), mogą przejawiać większą agresywność w odpowiedzi na stres, w tym również w sytuacjach, które wywołują lęk separacyjny. Przykładem może być agresja wobec innych zwierząt domowych lub ludzi w domu. Samce mogą także wykazywać destrukcyjne zachowania, takie jak drapanie, gryzienie przedmiotów czy załatwianie się poza kuwetą, co jest typowym objawem reakcji na lęk lub frustrację. Samice, z drugiej strony, mogą wykazywać objawy bardziej introspektywne, takie jak apatia, ukrywanie się w ustronnym miejscu, a także tendencję do częstszego wycofywania się w momentach stresu. Choć mogą również odczuwać lęk separacyjny, objawy te mogą być subtelniejsze niż u samców.

Podobne różnice można zauważać u psów, chociaż wpływ płci często jest modyfikowany przez kastrowanie lub sterylizację. U niekastrowanych samców psów wyższy poziom testosteronu

może nasilać zachowania dominacyjne i niepokój w sytuacji rozłęki. Samice psów bywają bardziej wrażliwe i subtelnie sygnalizują dyskomfort poprzez wycofanie czy zachowania lękowe, zamiast bezpośredniej agresji.

3.6.2. WPŁYW HORMONU TESTOSTERONU

Testosteron, hormon dominujący u samców, może wpływać na poziom energii i agresji. U niezoperowanych samców testosteron jest również odpowiedzialny za bardziej dominujące zachowania, co może przejawiać się w próbach zdominowania innych zwierząt w domu lub w wyrażaniu swojego terytorializmu, co może potęgować stres i lęk separacyjny. Samce mogą mieć także tendencję do oznaczania terytorium (np. przez markowanie moczem). U samic, zwłaszcza tych niekastrowanych, również mogą występować objawy dominacji, jednak ze względu na inny poziom hormonów, takie zachowania są mniej nasiłone, a stres związany z separacją może przybierać inne formy, np. zmiany w zachowaniach związanych z załatwianiem się.

3.6.3. RÓŻNICE SPOŁECZNE

Samce kotów mogą być bardziej skłonne do konfrontacji z innymi kotami, zwłaszcza w przypadku rywalizacji o zasoby, podczas gdy samice często wykazują wyższy poziom tolerancji na obecność innych zwierząt, co może ułatwiać życie w domu z wieloma kotami.

Samce psów często wykazują bardziej dominujące i terytorialne zachowania, zwłaszcza jeśli nie są kastrowane. Samice zazwyczaj są bardziej niezależne i mogą przejawiać silniejsze instynkty opiekuńcze, szczególnie w okresie cieczki.

3.6.4. REAKCJE NA INTERAKCJE Z LUDŹMI

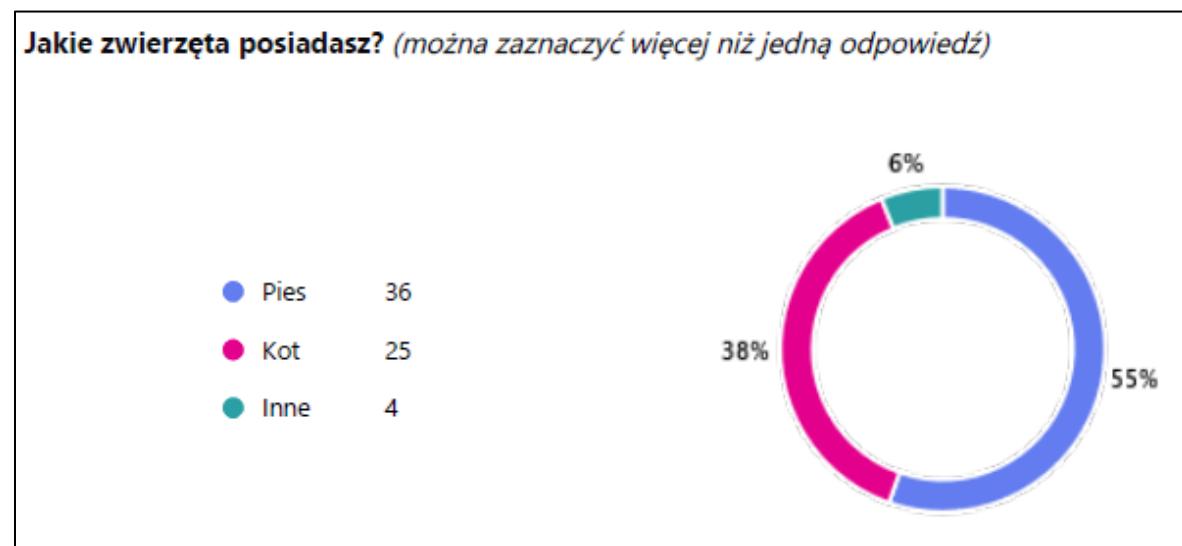
Samice kotów i psów częściej wykazują większą gotowość do interakcji z ludźmi w subtelny sposób, np. przez przytulanie się czy czułe gesty. Samce, zarówno kotów, jak i psów, mogą być bardziej samodzielne i skłonne do szukania interakcji w bardziej aktywny sposób, np. przez zabawę. Różnice te mogą również wpływać na to, jak zwierzęta reagują na samotność — samce psów i kotów mogą bardziej szukać interakcji i być bardziej zależne od obecności opiekuna, podczas gdy samice mogą wykazywać większą niezależność.

4. ANKIETY

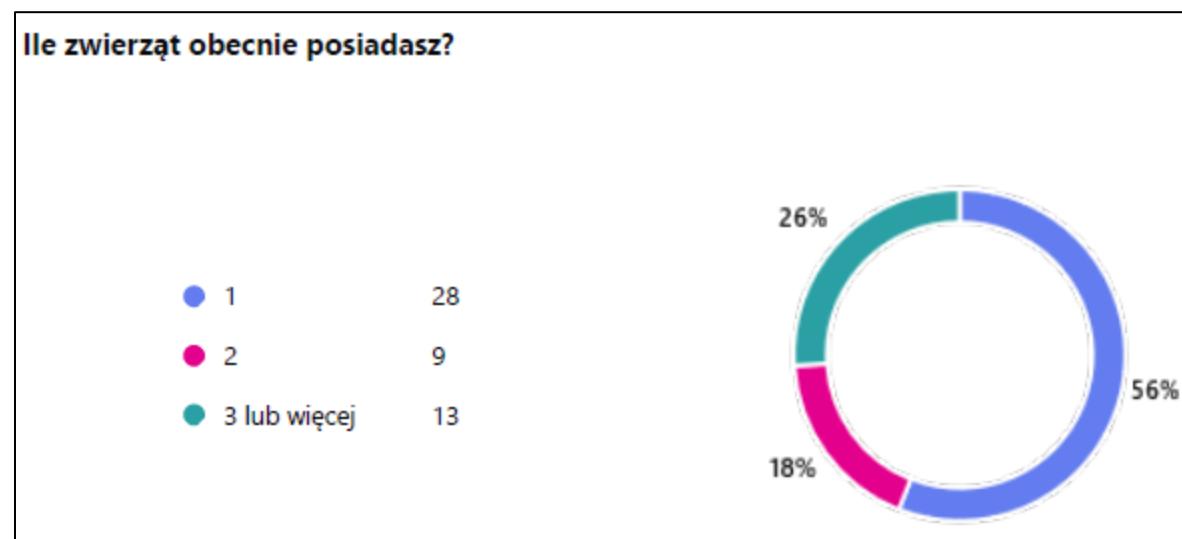
Przeprowadziliśmy ankietę dotyczącą opieki nad zwierzętami. Zebrałyśmy odpowiedzi od 50 osób, co pozwoliło nam na lepsze zrozumienie z jakimi problemami borykają się ludzie posiadający zwierzęta domowe oraz innych ważnych dla nas szczećłów. Pozwoliło nam to na określenie grupy docelowej naszego projektu oraz czego tak naprawdę właściciele potrzebują.

4.1. Podstawowe pytania

Żeby dowiedzieć się czy ankietowany mógłby być naszą grupą docelową, zadaliśmy najpierw podstawowe pytania, na które odpowiedzi widoczne są na Rys. 7.1, Rys 7.2 i Rys. 7.3.



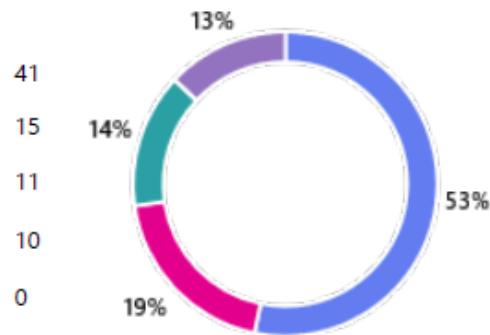
Rys. 4.1. Ankieta – pytanie 1. [opracowanie własne]



Rys. 4.2. Ankieta – pytanie 2. [opracowanie własne]

Gdzie Twoje zwierzę spędza większość czasu? (można zaznaczyć więcej niż jedną odpowiedź)

- W domu
- Na zewnątrz (np. ogród, podwórko)
- Spaceruje z opiekunem (dotyczy głównie psów)
- Wychodzi samodzielnie i wraca (dotyczy głównie kotów)
- Inne



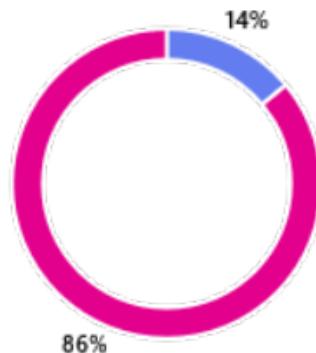
Rys. 4.3. Ankieta – pytanie 3. [opracowanie własne]

4.2. Pytania dotyczące używania wcześniej technologii IoT

Następnie zadaliśmy pytania dotyczące używania technologii IoT przez naszych ankietowanych. Wyniki przedstawiliśmy na Rys. 7.4 oraz Rys 7.5.

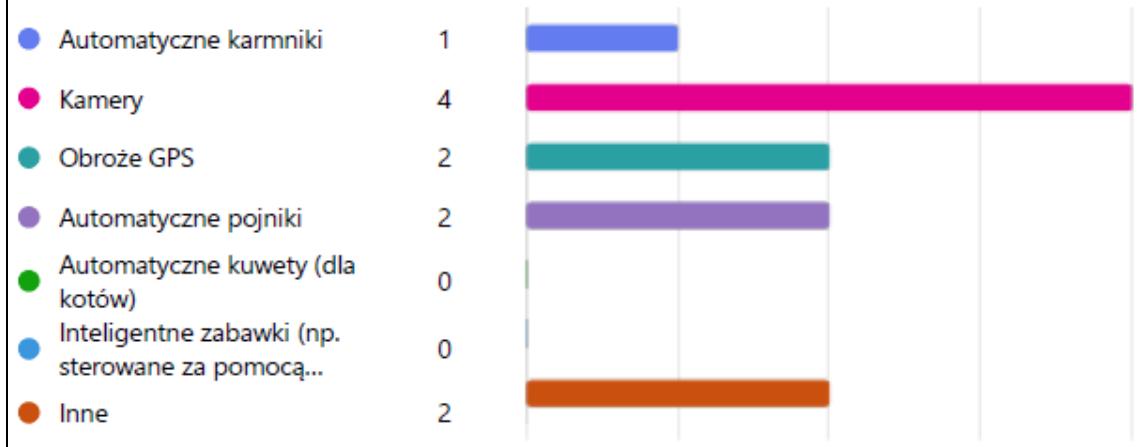
Czy korzystasz z rozwiązań technologicznych w opiece nad swoim zwierzęciem (np. automatyczne karmniki, kamery, obroże GPS)?

- Tak 7
- Nie 43



Rys. 4.4. Ankieta, pytanie 1. o IoT. [opracowanie własne]

Jeśli w poprzednim pytaniu odpowiedziałeś/łaś "tak", to z jakich? (można zaznaczyć więcej niż jedną odpowiedź)

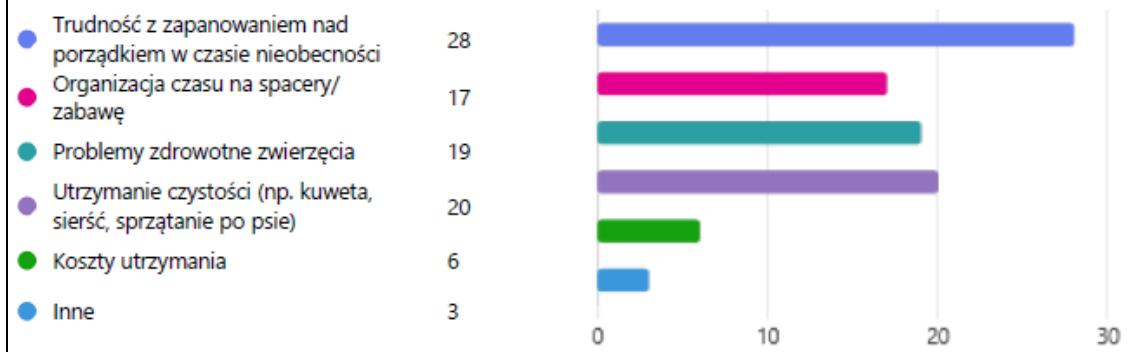


Rys. 4.5. Ankieta, pytanie 2. o IoT. [opracowanie własne]

4.3. Pytania dotyczące problemów ze zwierzętami

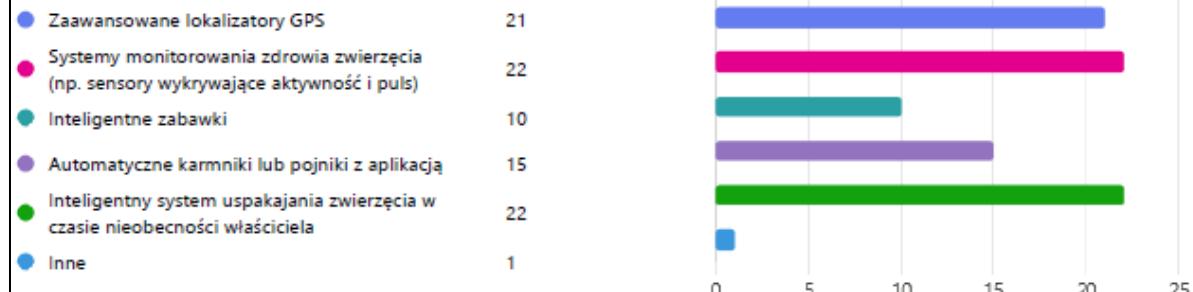
Sformułowaliśmy kilka pytań, dzięki którym dowiedzieliśmy się z jakimi głównie problemami borykają się właściciele zwierząt. Odpowiedzi ankietowanych przedstawiliśmy na Rys. 7.6 oraz na Rys. 7.7.

Jakie aspekty opieki nad zwierzęciem uważasz za najtrudniejsze? (można zaznaczyć więcej niż jedną odpowiedź)



Rys. 4.6. Ankieta – problemy właścicieli. [opracowanie własne]

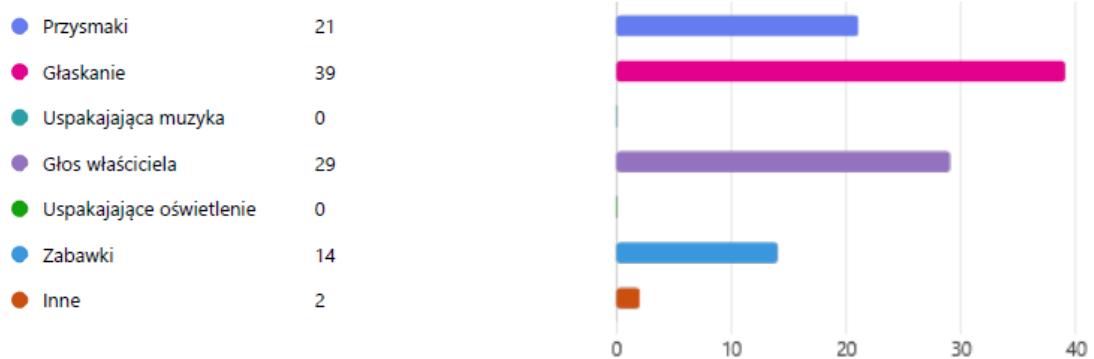
Jakie technologie byłyby dla Ciebie najbardziej interesujące? (można zaznaczyć więcej niż jedną odpowiedź)



Rys. 4.7. Ankieta – zainteresowanie technologią. [opracowanie własne]

Kluczowe dla nas było, co możemy zrobić, aby zwierzę stało się spokojne. Odpowiedzi widoczne są na Rys. 7.8.

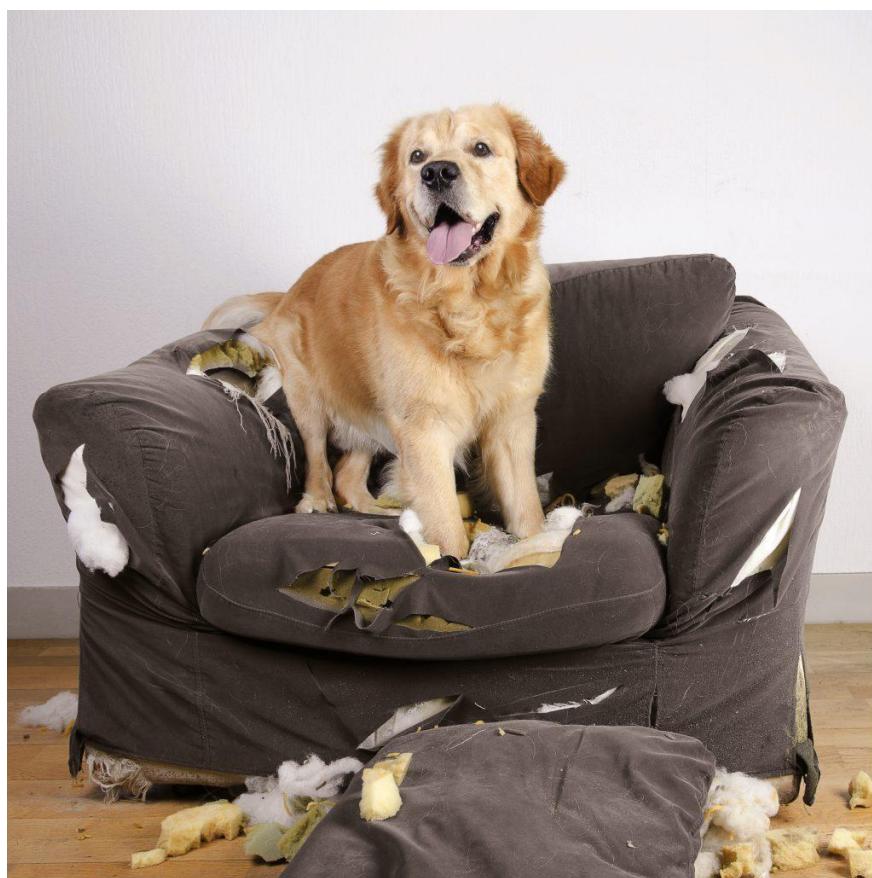
Co działa na Twoje zwierzę, aby było spokojniejsze? (można zaznaczyć więcej niż jedną odpowiedź)



Rys. 4.8. Ankieta – klucz do spokojnego zwierzęcia. [opracowanie własne]

5. SYNTEZA

Po przeanalizowaniu dostępnych danych i przeprowadzeniu wywiadów z właścicielami zwierząt, doszliśmy do wniosku, że największym problemem, z którym borykają się osoby posiadające psy i koty, jest lęk separacyjny. Zwierzęta, pozostawione same w domu przez dłuższy czas, często doświadczają silnego stresu, który może prowadzić do niepożądanych zachowań, takich jak niszczenie przedmiotów, szczekanie, drapanie czy inne formy agresji. Problem ten nie tylko wpływa na komfort zwierzęcia, ale również generuje straty materialne i emocjonalne dla właścicieli. Nasza synteza problemu wskazuje na potrzebę opracowania rozwiązania, które pomoże zwierzętom w radzeniu sobie z samotnością i stresem związанныm z nieobecnością ich opiekunów.



Rys. 5.1. Pies niszczący meble. [<https://petsworld.com.pl/lek-separacyjny-u-psow-2/>]

6. PRZYKŁADOWE PERSONY

Przykładowe persony w naszym projekcie reprezentują różne typy właścicieli zwierząt, którzy borykają się z problemem lęku separacyjnego swoich pupili. Każda persona została zaprojektowana w taki sposób, aby odzwierciedlała unikalne potrzeby i oczekiwania, co pozwala na lepsze dopasowanie proponowanego rozwiązania do różnych sytuacji życiowych i stylów opieki nad zwierzętami.

6.1. JAKUB KOWALSKI



Jakub Kowalski

Specjalista ds. sprzedaży

Edukacja: Absolwent zarządzania na Uniwersytecie Ekonomicznym w Warszawie, specjalista w zarządzaniu projektami.

Kariera: 25-letni specjalista ds. projektów IT w korporacji, ceniony za zaangażowanie i skuteczność, często pracuje po godzinach.

Status: Single, mieszka w Warszawie z kotem Felixem, którego uwielbia, choć często zostawia go samego z powodu pracy.

Biografia: Jakub to 25-latek, który od najmłodszych lat wykazywał ambicje i chęć zdobywania nowych umiejętności. Jako dziecko interesował się technologią, a w wieku 12 lat zbudował swój pierwszy komputer. Choć praca pochłania większość jego czasu, Jakub marzy o tym, by w przyszłości znaleźć lepszy balans między życiem zawodowym a osobistym. W wolnych chwilach uwielbia oglądać seriale, grać na konsoli i czytać książki o samorozwoju.

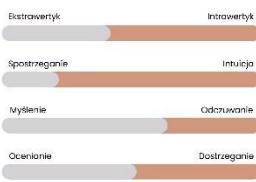
OPIS

- Uwiera, że ciekąca pracę oraz jego poświęceniami związanymi z życiem prywatnym osiągnie sukces
- Często wraca późno z pracy, ponieważ pracuje nadgodzinny
- Lubi po powrocie do domu w pełni odciąć się od pracy oraz odpocząć
- Kocha zwierzęta, lubi wracać do swojego kota, lecz często ma z nim problemy w związku z całodzienną nieobecnością

CELE

- Marzy o osiągnięciu stanowiska dyrektorskiego w dużej korporacji
- Chce znaleźć więcej czasu w momencie kiedy osiągnie wyznaczony przez siebie cel
- Chcieliby znaleźć również czas na życie towarzyskie oraz znaleźć czas na relacje romantyczne
- Zależy mu na rozwijaniu pasji związanej z elektroniką oraz tworzeniem urządzeń

OSOBOWOŚĆ



ŚRODOWISKO

- Pracuje w zdrowym lecz bardzo zaangażowanym i wymagającym środowisku
- Najbardziej frustruje się niespełnieniem oczekiwani nałożonych przez przełożonego
- Mieszka w niewielkim mieszkaniu na nowoczesnym osiedlu razem z kotem
- Otoczają go młodzi ludzie z dużymi ambicjami

Rys. 6.1. Persona 1 – Jakub Kowalski. [opracowanie własne]

6.2. EWA DĄBROWSKA



Ewa Dąbrowska

Dziennikarka śledcza, aktywistka

Edukacja: Ewa ukończyła dziennikarstwo na Uniwersytecie Warszawskim, specjalizując się w reportażu. Już w czasie studiów napisała artykuły o ochronie środowiska.

Kariera: Jest dziennikarką w portalu informacyjnym, zajmując się tematyką społeczną i klimatyczną. Często podtrudnia się służbowo, prowadzi też podcast o ekologii.

Status: Mieszka z trzema kotami (Tosią, Felekiem, Melą) i dwoma psami (Borysem, Luncą). Zwierzęta są jej największą radością.

Biografia: Pochodzi z Gdańska, od dziecka angażowała się w ochronę przyrody. Jest współzałożycielką fundacji klimatycznej i działa jako wolontariuszka w schroniskach. W wolnych chwilach spaceruje z psami lub bierze udział w protestach klimatycznych.

OSOBOWOŚĆ



OPIS

- Uważa, że praca na rzecz klimatu oraz dziennikarstwo śledcze to jej sposób na realne zmienianie świata.
- Często wyjeżdża służbowo, co powoduje, że rzadko bywa w domu.
- Po powrocie do domu stara się spędzać czas z psami i kotami, co daje jej odpoczynek po intensywnym dniu.
- Kocha zwierzęta i angażuje się w ich ochronę, ale czasem czuje się winna, że nie zawsze ma dla nich tyle czasu, ile by chciała.

CELE

- Chce, by jej praca jako dziennikarki miała realny wpływ na walkę ze zmianami klimatycznymi.
- Marzy o zbudowaniu większej platformy (np. fundacji lub serii publikacji), która edukowałaby społeczeństwo w kwestii ekologii.
- Pragnie znaleźć więcej czasu na życie prywatne i pasje, mimo wymagającego trybu pracy.
- Chce rozwijać swoje umiejętności w reportażu i dziennikarstwie śledczym.

ŚRODOWISKO

- Pracuje w dynamicznym, ale wymagającym środowisku dziennikarskim, gdzie liczy się tempo i rzetelność.
- Frustruje ją brak wystarczającego wsparcia społecznego dla działań na rzecz klimatu.
- Mieszka w przytulnym mieszkaniu pełnym roślin i ekologicznych rozwiązań razem z trzema kotami i dwoma psami.
- Otacza ją ludzie zaangażowani w aktywizm i działania proekologiczne.

Rys. 6.2. Persona 2 – Ewa Dąbrowska. [opracowanie własne]

6.3. MAREK FĄK



Marek Fąk

Specjalista ds. IoT

Edukacja: Marek ukończył studia inżynierskie z zakresu automatyki i robotyki na Politechnice Gdańskiej. Zawsze interesował się nowymi technologiami oraz ich zastosowaniem w codziennym życiu.

Kariera: Pracuje jako specjalista ds. technologii IoT w międzynarodowej firmie IT, gdzie zajmuje się projektowaniem rozwiązań inteligentnego domu. W wolnym czasie rozwija projekty DIY związanego z elektroniką i urządzeniami smart.

Status: Mieszka z żoną, synem i dwoma psami (Reks i Max) w domu na przedmieściach. Psy są częścią rodziny, a Marek dba o ich zdrowie i komfort, szczególnie gdy zostają same w domu.

Biografia: Marek Kowalski to specjalista ds. IoT, pasjonat nowych technologii i opieki nad zwierzętami. Pracuje w firmie zajmującej się inteligentnymi domami, mieszka z rodziną i dwoma psami, których opiekuje się przy użyciu nowoczesnych urządzeń monitorujących.

OSOBOWOŚĆ



OPIS

- Zależy mu na łączeniu nowych technologii z opieką nad zwierzętami. Widzi w IoT potencjał do poprawy ich dobrostanu.
- Często wyjeżdża służbowo, a psy zostają pod opieką automatycznych urządzeń i systemów monitorujących.
- Stara się minimalizować stres zwierząt w sytuacjach takich jak burze czy fajerwerki.
- Wierzy, że technologia powinna ułatwiać życie zarówno ludziom, jak i zwierzętom.

CELE

- Chce mieć pełną kontrolę nad stanem zdrowia i samopoczuściem swoich psów, nawet będąc poza domem.
- Szuka urządzeń IoT, które potrafią monitorować parametry fizjologiczne zwierząt i reagować w razie wykrycia stresu (np. przez emisję uspokajających dźwięków lub aromaterapię).
- Marzy o rozwijaniu własnych projektów smart dla zwierząt, które byłyby dostępne na rynku.

ŚRODOWISKO

- Pracuje w dynamicznej branży technologicznej, gdzie regularnie testuje nowoczesne urządzenia IoT.
- Mieszka w inteligentnym domu wyposażonym w systemy automatyzacji (świetlenie, ogrzewanie, monitoring).
- Otacza się ludźmi, którzy również pasjonują się nowymi technologiami i są otwarci na ich wdrożenie w codziennym życiu.

Rys. 6.3. Persona 3 – Marek Fąk. [opracowanie własne]

6.4. JULIA NOWAK



Julia Nowak

Teraapeutka zwierząt

Edukacja: Julia ukończyła psychologię na Uniwersytecie Jagiellońskim, specjalizując się w terapii zwierząt. Przez wiele lat pracowała jako behawiorysta zwierząt, pomagając właścicielowi radzić sobie z trudnymi zachowaniami swoich pupili.

Kariera: Julia jest konsultantką ds. zachowań zwierząt. Współpracuje z organizacjami pomagającymi zwierzętom, zwłaszcza adoptowanym lub z schronisk, a także jest autorką książek o psychologii zwierząt.

Status: Mieszka w Krakowie z trzema kotami, które znalazły się na ulicy. Jest zaangażowana w dobrostan zwierząt, pełniąc rolę "adoptującej matki" dla trudnych przypadków.

Biografia: Julia pochodzi z małego miasteczka, gdzie od dzieciństwa miała bliski kontakt z zwierzętami. Po ukończeniu psychologii, została behawiorystką zwierząt, pomagając im w adaptacji do nowych warunków.

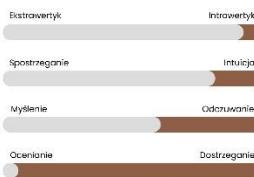
OPIS

- Julia jest przekonana, że każde zwierzę zasługuje na zrozumienie i odpowiednią opiekę, dlatego stawia na indywidualne podejście do każdego przypadku.
- Dba o zdrowie i emocjonalny komfort swoich kotów, zawsze szuka nowych metod i narzędzi, które mogłyby poprawić ich codzienne życie.
- W związku z tym, że często nie jest w stanie być w domu, szuka technologicznych rozwiązań, które umożliwią jej monitorowanie stanu emocjonalnego kotów i pomaganie im w radzeniu sobie z lękami czy stresem.

CELE

- Julia chciałaby mieć możliwość monitorowania samopoczucia swoich kotów w czasie rzeczywistym, aby reagować na ich potrzeby, zwłaszcza gdy jest poza domem.
- Szuka urządzeń, które pozwolą jej na zdalne kontrolowanie warunków otoczenia kotów (np. temperatura, dźwięki) oraz umożliwią im poczucie bezpieczeństwa, np. poprzez odtwarzanie uspokajających dźwięków.
- Jej celem jest również rozwinięcie oferty usług behawioralnych o technologii, która wspiera jej pracę z klientami.

OSOBOWOŚĆ



ŚRODOWISKO

- Julia pracuje w spokojnym, ale wymagającym środowisku, w którym często łączy pracę z pasją do zwierząt.
- Mieszka w nowoczesnym mieszkaniu, które zostało przystosowane do potrzeb jej kotów, z licznymi "kotami ścieżkami" i miejscami do zabawy.
- Otoczają ją osoby, które również interesują się dobrostaniem zwierząt i wykorzystaniem technologii w terapii behawioralnej.

Rys. 6.4. Persona 4 – Julia Nowak. [opracowanie własne]

6.5. KRZYSZTOF ZIELIŃSKI



Krzysztof Zieliński

Menedżer fundacji

Edukacja: Krzysztof ukończył studia z zakresu biologii na Uniwersytecie Warszawskim, specjalizując się w ekologii i ochronie środowiska. Zawsze interesował się ekosystemami i relacjami między ludźmi a zwierzętami.

Kariera: Pracuje jako menedżer w fundacji zajmującej się ochroną dzikich zwierząt, a także jako konsultant ekologiczny dla firm wdrażających rozwiązania przyjazne środowisku.

Status: Mieszka z partnerką na wsi, w domu otoczymy lasem, gdzie opiekuje się kilkoma adoptowanymi zwierzętami – psami i kotami.

Biografia: Krzysztof Zieliński wychował się na wsi, gdzie rozwinał pasję do ekologii i ochrony dzikiej fauny. Obecnie pracuje w fundacji zajmującej się ochroną fauny, jednocześnie dbając o dobrostan swoich adoptowanych zwierząt i poszukując technologii wspierających ich monitorowanie.

OPIS

- Krzysztof ma głęboką więź z naturą i wierzy, że technologia może wspierać ochronę środowiska, nie zastępując jednak tradycyjnych metod.
- Jego życie zawodowe i prywatne w dużej mierze koncentruje się na ochronie dzikiej fauny, ale również na dbaniu o dobrostan swoich adoptowanych zwierząt.
- Jest zaangażowany w rozwój technologii, które pomagają w monitorowaniu zdrowia zwierząt i dostosowywaniu warunków ich życia do naturalnych potrzeb.

CELE

- Krzysztof szuka urządzeń IoT, które pomogą monitorować zdrowie i zachowanie zwierząt w czasie rzeczywistym, szczególnie w kontekście dzikich gatunków.
- Pragnie wykorzystać nowoczesne technologie do poprawy warunków życia swoich pupili, szczególnie w kontekście wypadków, stresów czy zmian środowiskowych.
- Marzy o wprowadzeniu takich rozwiązań w rezerwatach przyrody, gdzie monitoring zdrowia dzikich zwierząt mógłby wspierać działania ochronne.

OSOBOWOŚĆ



ŚRODOWISKO

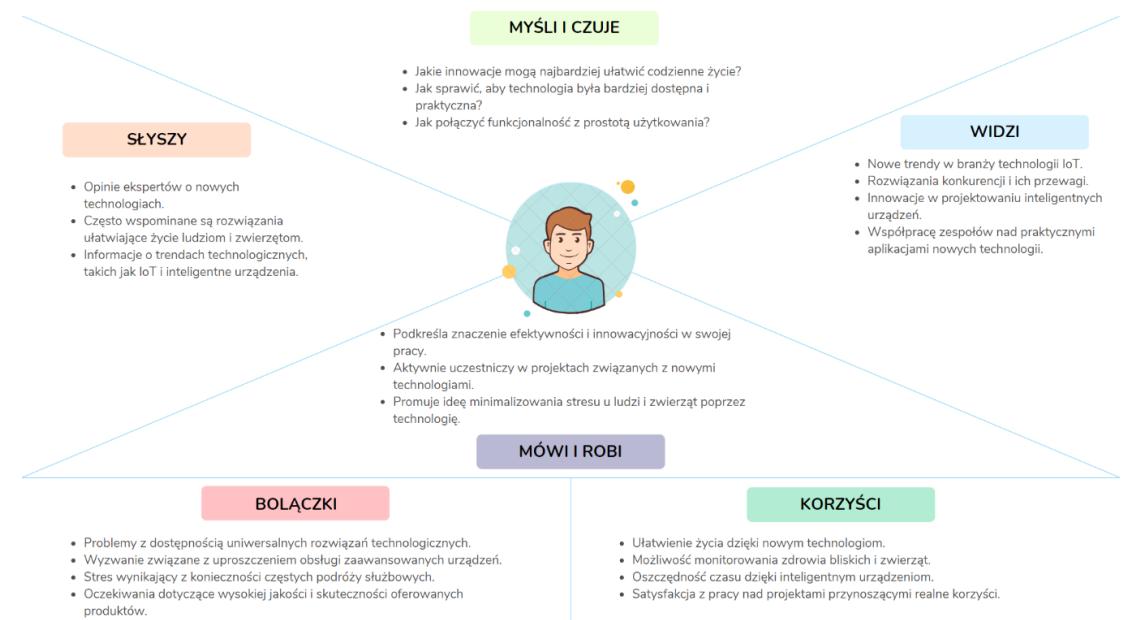
- Krzysztof pracuje w dynamicznym, ale wymagającym środowisku, gdzie technologia wspomaga działania ochrony środowiska.
- Mieszka na wsi w zgodzie z naturą, żyjąc w otoczeniu lasów i parków, z dużym naciskiem na ekologiczną uprawę roślin i dbanie o środowisko.
- Otoczają go osoby związane z ochroną przyrody, biologią i technologią, które wspólnie realizują projekty proekologiczne.

Rys. 6.5. Persona 5 – Krzysztof Zieliński. [opracowanie własne]

7. MAPY EMPATII

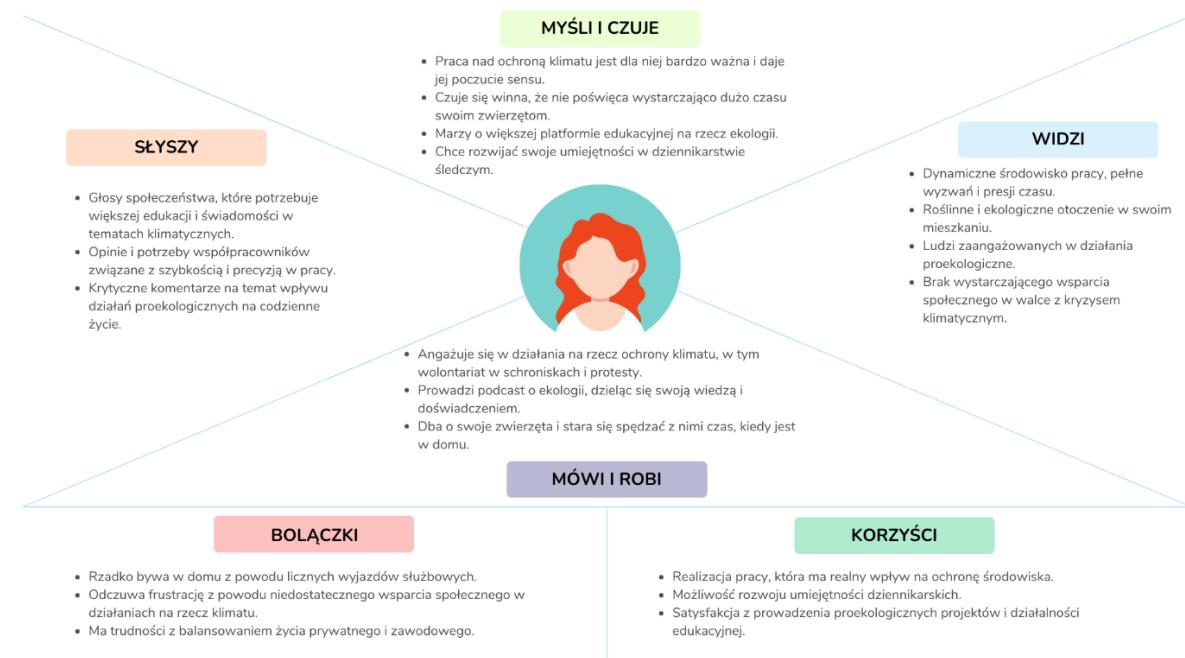
Mapy empatii w naszym projekcie służą jako narzędzie do głębszego zrozumienia potrzeb oraz emocji właścicieli zwierząt, a także ich pupili. Dzięki tym mapom mogliśmy lepiej określić, jakie są kluczowe problemy związane z lękiem separacyjnym i jak nasze rozwiązanie może pozytywnie wpływać na komfort zwierząt oraz ich opiekunów.

7.1. MAREK FĄK



Rys. 7.1. Mapa empatii Marka Fąka. [opracowanie własne]

7.2. JULIA NOWAK



Rys. 7.2. Mapa empatii Julii Nowak. [opracowanie własne]

8. INTERESARIUSZE

8.1. KTO ZYSKUJE?

- Właściciele zwierząt:

Korzyści: Większe poczucie bezpieczeństwa, dostęp do danych o zdrowiu pupila, łatwiejsze lokalizowanie zgubionych zwierząt.

- Weterynarze:

Korzyści: Dane zbierane przez urządzenia ułatwiają diagnozę, monitorowanie chorób przewlekłych i planowanie leczenia.

- Firmy technologiczne (produkenci IoT):

Korzyści: Możliwość sprzedaży nowych produktów i usług, np. abonamentów na analizę danych czy aktualizacje oprogramowania.

- Schroniska dla zwierząt:

Korzyści: Szybsze odnajdywanie zaginionych zwierząt, możliwość lepszego monitorowania zdrowia podopiecznych.

- Behawioryści i trenerzy zwierząt:

Korzyści: Dane o zachowaniu zwierzęcia pomagają w opracowaniu skuteczniejszych metod szkoleniowych.

8.2. KTO TRACI?

- Tradycyjne sklepy zoologiczne:

Straty: Mogą nie być przygotowane na sprzedaż zaawansowanych technologicznie urządzeń IoT, tracąc klientów na rzecz e-commerce.

- Osoby niechętne technologii:

Straty: Trudności w adaptacji technologii przez osoby starsze lub mniej obeznane z IoT.

- Zwierzaki (w pewnych sytuacjach):

Straty: Potencjalny dyskomfort związany z urządzeniem (np. ciężka obroża), ryzyko nieprzemyślanego wykorzystania technologii.

- Firmy oferujące tradycyjne akcesoria dla zwierząt:

Straty: Spadek sprzedaży produktów takich jak zwykłe obroże, gdy konsumenti wybierają wersje intelligentne.

9. PYTANIA HMW

Aby skutecznie rozwiązać problem lęku separacyjnego u zwierząt i poprawić ich dobrostan, kluczowe stało się wypracowanie odpowiedzi na pytania, które obejmują różne aspekty życia kota lub psa podczas nieobecności ich właścicieli. Pytania HMW pozwalają na odkrycie nowych możliwości i potencjalnych rozwiązań w takich obszarach jak monitorowanie zdrowia, zapewnienie odpowiedniej interakcji, kontrola aktywności fizycznej i umysłowej, a także zapewnienie bezpieczeństwa zwierząt. Dzięki tym pytaniom mogliśmy skupić się na najważniejszych potrzebach zwierząt, tworząc technologie, które będą wspierały ich codzienne życie, minimalizując stres oraz ryzyko niepokojów związanych z samotnością.

Każde z poniższych pytań zostało zaprezentowane w celu rozpoznania obszarów, które wymagają innowacyjnych rozwiązań technologicznych i które mogą pomóc właścicielom w dbaniu o ich pupili, nawet w czasie ich nieobecności.

9.1. Pytanie nr 1

Jak możemy monitorować zdrowie kota, używając inteligentnych czujników, aby właściciel miał pewność, że jego pupil jest w dobrym stanie?

Odpowiedź: Wykorzystanie inteligentnych czujników do monitorowania zdrowia kota może obejmować urządzenia śledzące aktywność fizyczną, temperaturę ciała, tętno, a także analizujące wzorce snu zwierzęcia. Dzięki integracji z aplikacją mobilną, właściciele mogliby na bieżąco otrzymywać dane dotyczące kondycji zdrowotnej swojego pupila. Przykładem takich czujników są opaski na szyję lub kamizelki, które zbierają dane o aktywności fizycznej, a także detektory drgań i temperatury, które mogą wskazywać na ewentualne problemy zdrowotne, takie jak gorączka czy nadmierny stres.

9.2. Pytanie nr 2

Jak możemy zautomatyzować karmienie kota, używając inteligentnych dozowników, aby upewnić się, że kot zawsze ma dostęp do jedzenia i wody?

Odpowiedź: Inteligentne dozowniki karmy mogą automatycznie podawać jedzenie i wodę zwierzęciu w zaplanowanych porcjach. Systemy te mogą być połączone z aplikacją, która pozwala właścicielowi ustawić harmonogram karmienia oraz kontrolować ilość podawanego jedzenia. Dodatkowo, inteligentne dozowniki mogą mieć czujniki poziomu wody i karmy, które informują właściciela, kiedy zasoby są na wyczerpaniu. W ten sposób właściciele mogą mieć pewność, że ich pupil ma zawsze dostęp do świeżego jedzenia i wody, nawet gdy są poza domem.

9.3. Pytanie nr 3

Jak możemy umożliwić interakcję z kotem, używając kamer i aplikacji, aby zredukować stres zwierzęcia podczas nieobecności właściciela?

Odpowiedź: Zainstalowanie kamery z funkcją dwustronnej komunikacji (audio i video) może pozwolić właścicielowi na rozmowę z kotem lub psem. Tego typu systemy umożliwiają

właścicielowi nawiązywanie kontaktu z pupilem za pomocą aplikacji, dzięki czemu zwierzęta mogą poczuć się mniej samotne. Dodatkowo, możliwe jest wykorzystanie kamer do monitorowania zachowań zwierzęcia, co pozwala na szybsze wykrywanie oznak lęku separacyjnego i podjęcie odpowiednich działań.

9.4. Pytanie nr 4

Jak możemy wykrywać nietypowe zachowania kota, używając danych z czujników, aby szybko reagować na potencjalne problemy?

Odpowiedź: Czujniki ruchu, akcelerometry oraz kamery mogą zostać wykorzystane do monitorowania codziennych aktywności kota, takich jak bieganie, chowanie się lub nadmierne drapanie. Algorytmy sztucznej inteligencji mogą analizować te dane i wykrywać nietypowe zachowania wskazujące na stres, lęk lub problemy zdrowotne. W przypadku wykrycia niepokojących sygnałów, system może wysłać powiadomienie do właściciela, który będzie mógł zareagować, na przykład uruchamiając interaktywną zabawkę lub skontaktować się ze specjalistą.

9.5. Pytanie nr 5

Jak możemy zwiększyć bezpieczeństwo kota w domu, używając systemów monitoringu, aby zapobiegać wypadkom lub ucieczkom?

Odpowiedź: Aby zwiększyć bezpieczeństwo kota, można zainstalować systemy monitoringu wyposażone w czujniki ruchu, kamery z funkcją rozpoznawania twarzy, a także czujniki otwarcia drzwi i okien. Dzięki tym urządzeniom, właściciele mogą monitorować, czy ich pupil nie zbliża się do potencjalnie niebezpiecznych miejsc, takich jak otwarte okna lub drzwi. Ponadto, systemy te mogą natychmiast powiadomić właściciela o nieoczekiwanych wydarzeniach, takich jak otwarcie drzwi lub ucieczka zwierzęcia, co pozwoli na szybszą reakcję i zapobieganie wypadkom.

9.6. Pytanie nr 6

Jak możemy zapewnić kotu odpowiedni poziom aktywności fizycznej i umysłowej, używając inteligentnych urządzeń, aby zadbać o jego dobrostan podczas nieobecności właściciela?

Odpowiedź: Inteligentne zabawki, takie jak interaktywne piłki, lasery lub roboty, które reagują na ruchy zwierzęcia, mogą zapewnić kotu aktywność fizyczną i umysłową. Takie urządzenia mogą być zaprogramowane tak, aby działały w określonych porach dnia, angażując zwierzę do zabawy i pomagając w walce z nudą. Dodatkowo, urządzenia te mogą być połączone z aplikacją, która pozwala właścicielowi na zdalne sterowanie zabawką lub ustawianie harmonogramu aktywności, co pozwala na regularne utrzymanie odpowiedniego poziomu ruchu i stymulacji umysłowej u kota.

10. IDEACJA I IDENTYFIKACJA

Etap ideacji w naszym projekcie polegał na generowaniu różnorodnych pomysłów i rozwiązań, które mogłyby skutecznie rozwiązać problem lęku separacyjnego u zwierząt. Wykorzystaliśmy podejście kreatywne, które pozwoliło na opracowanie wielu koncepcji technologicznych, uwzględniających potrzeby zarówno zwierząt, jak i ich właścicieli. Proces identyfikacji skupił się na wybraniu najlepszych pomysłów, które odpowiadałyby na kluczowe wyzwania zidentyfikowane w poprzednich etapach, przy jednoczesnym zapewnieniu wykonalności i efektywności zaproponowanych rozwiązań.

10.1. BURZA MÓZGÓW

Podczas burzy mózgów zespół wspólnie generował pomysły na rozwiązywanie lęku separacyjnego u zwierząt. Wykorzystując techniki kreatywne, takie jak mapa myśli i „brainwriting”, zebraliśmy szeroką gamę koncepcji, które następnie oceniliśmy pod kątem wykonalności i efektywności, wybierając najlepsze rozwiązania do dalszego rozwoju. Najlepsze z pomysłów zgromadziliśmy poniżej.

10.1.1. INTERAKTYWNE ZABAWKI

Zabawki, które mogą być sterowane zdalnie za pomocą aplikacji, pozwalają właścicielom angażować zwierzęta w zabawę, nawet gdy nie są w domu. Tego typu rozwiązanie pomaga w redukcji lęku separacyjnego, zapewniając zwierzęciu rozrywkę i stymulację fizyczną.

10.1.2. MONITOROWANIE AKTYWNOŚCI I ZDROWIA

Czujniki śledzące ruch zwierzęcia, jego temperaturę, tętno oraz inne parametry zdrowotne pozwalają właścicielom na bieżąco monitorować stan zdrowia pupila. W przypadku wykrycia nieprawidłowości, system automatycznie wysyła powiadomienie, umożliwiając szybszą reakcję w razie potrzeby.

10.1.3. AUTOMATYCZNE KARMIENIE I NAWADNIANIE

Inteligentne dozowniki karmy i wody, które automatycznie podają jedzenie i wodę w odpowiednich ilościach oraz czasie. Dzięki połączeniu z aplikacją, właściciele mogą na bieżąco kontrolować poziom karmy i wody oraz ustawać harmonogramy, zapewniając zwierzęciu stały dostęp do pożywienia, nawet pod ich nieobecność.

10.1.4. WYKRYWANIE DŹWIĘKÓW ZWIERZĘCIA

System oparty na analizie dźwięków może wykrywać specyficzne dźwięki, takie jak miauczenie kota lub szczekanie psa, oraz rozróżniać, czy są one wynikiem niepokoju, stresu, czy normalnej aktywności zwierzęcia. Dzięki połączeniu z aplikacją, system może wysłać powiadomienie do właściciela, informując go o nietypowych dźwiękach, co pozwala na szybsze podjęcie odpowiednich działań.

10.2. IDENTYFIKACJA

W etapie identyfikacji, po dokładnej analizie wykonalności pomysłów, zdecydowaliśmy się na skoncentrowanie się na dwóch kluczowych rozwiązańach, które pozwalają na skuteczne monitorowanie zachowań zwierząt w czasie nieobecności ich właścicieli: interaktywnej zabawce oraz obroży monitorującej lokalizację i aktywność. Ponadto, wykorzystaliśmy mikrofon do analizy dźwięków emitowanych przez zwierzęta, takich jak miauczenie kota czy szczerkanie psa, w celu wykrycia oznak stresu. Każde z tych rozwiązań zostało szczegółowo ocenione pod kątem wykonalności oraz zasobów niezbędnych do realizacji.

10.2.1. INTERAKTYWNA ZABAWKA

Analiza wykonalności: Interaktywna zabawka, która polega na przesuwającym się sznurku po bloczkach, jest rozwiązaniem, które angażuje zwierzę w zabawę i pomaga w zmniejszeniu lęku separacyjnego. Zabawka jest zaprojektowana w taki sposób, aby mogła przyciągnąć uwagę zwierzęcia i zapewnić mu rozrywkę. Mechanizm przesuwania sznurka po bloczkach jest prosty, ale efektywny, pozwalając na zabawę bez nadmiernego obciążenia technologią.

Co będzie potrzebne:

- **Mechanizm przesuwania sznurka:** prosty mechanizm oparty na bloczkach, który umożliwia płynne przesuwanie sznurka,
- **Napęd:** Silnik DC 540
- **Zasilanie:** 12V/1A (lub 24V/1,5A dla wyższych obrotów)
- **Mikrokontroler:** ESP32-Wersja z obsługą PWM (ESP32 - DevKit V1)
- **Sterownik:** L298N Max 46V, 2A na kanał

10.2.2. OBROŻA MONITORUJĄCA AKTYWNOŚĆ

Analiza wykonalności: Obroża monitorująca aktywność zwierzęcia wykorzystuje technologię akcelerometru i żyroskopu, co pozwala na śledzenie ruchów zwierzęcia oraz analizowanie poziomu aktywności. Zmiana poziomu aktywności może być wskaźnikiem niepokoju zwierzęcia, co pozwala na szybszą reakcję właściciela. Zastosowanie akcelerometru i żyroskopu pozwala na precyzyjne wykrywanie intensywności ruchu, co jest pomocne w ocenie zachowań zwierzęcia w czasie nieobecności właściciela.

Co będzie potrzebne:

- **Akcelerometr i żyroskop:** czujniki umożliwiające monitorowanie poziomu aktywności i zmiany w ruchu zwierzęcia,
- **Bateria:** wydajna bateria, która zapewni długotrwałe działanie obroży, szczególnie podczas długotrwałej nieobecności właściciela,
- **Mikrofon:** urządzenie, które skutecznie rejestruje dźwięki w różnych warunkach domowych,
- **Połączenie Wi-Fi:** zdalna synchronizacja danych i możliwość oglądania statystyk oraz sterowania zabawką.

11. PROTOTYPOWANIE URZĄDZENIA GŁÓWNEGO

11.1. OKREŚLENIE WYMAGAŃ I FUNKCJONALNOŚCI URZĄDZENIA

Wymagania dla naszej obroży obejmowały zarówno aspekty funkcjonalne, jak i techniczne:

1. Wymagania funkcjonalne:

- **Detekcja aktywności zwierzęcia** – obroża powinna rejestrować ruch (np. chodzenie, bieganie, leżenie) oraz wykrywać dźwięki wydawane przez zwierzę.
- **Wykrywanie niepokoju** – na podstawie danych z czujników system powinien identyfikować oznaki lęku separacyjnego, np. nadmierne poruszanie się, skomlenie czy miauczenie/szczekanie.
- **Łączność z aplikacją** – obroża musi stabilnie komunikować się z aplikacją mobilną PetApp w czasie rzeczywistym.
- **Brak fałszywych alarmów** - system musi rozróżniać dźwięki kota od dźwięków otoczenia, aby unikać niepotrzebnych reakcji.

2. Wymagania techniczne:

- **Niskie zużycie energii** – obroża powinna działać co najmniej przez czas przeciętnego dnia w pracy bez konieczności częstego ładowania.
- **Stabilne połączenie bezprzewodowe** – urządzenie musi utrzymywać łączność z aplikacją mobilną.
- **Lekka i komfortowa konstrukcja** – obroża powinna być wygodna dla zwierzęcia i nie ograniczać jego ruchów.
- **Czuły mikrofon** – system powinien dokładnie rejestrować dźwięki wydawane przez zwierzę, minimalizując zakłócenia z otoczenia.

11.2. WYBÓR KOMPONENTÓW

11.2.1. ESP32-C6 – Mikrokontroler

11.2.1.1. Opis

ESP32-C6 to zaawansowany mikrokontroler firmy Espressif oparty na architekturze RISC-V. Wyposażony jest w obsługę Wi-Fi 6 (802.11ax), Bluetooth 5 LE oraz Zigbee/Thread. Moduł oferuje wysoką wydajność obliczeniową przy jednoczesnym niskim poborze mocy, co czyni go odpowiednim do zastosowań IoT.

11.2.1.2. Specyfikacja techniczna

- **Procesor:** RISC-V 32-bit, taktowanie do 160 MHz
- **Pamięć:** 320 KB RAM + obsługa zewnętrznej pamięci Flash
- **Łączność:** Wi-Fi 6, Bluetooth 5 LE, Zigbee, Thread
- **Pobór mocy:** ~5 µA w trybie głębokiego uśpienia, ~130 mA w trybie aktywnym
- **Napięcie zasilania:** 3,3V

11.2.1.3. Uzasadnienie wyboru

- **Energooszczędność:** Możliwość pracy w trybach niskiego poboru mocy, co wydłuża czas pracy na baterii.
- **Kompaktowe wymiary:** Ułatwia integrację w niewielkim urządzeniu.
- **Obsługa protokołów bezprzewodowych:** Zapewnia możliwość komunikacji z chmurą, aplikacją mobilną lub innymi urządzeniami IoT.

11.2.2. AKY0393 LiPo 3.7V 2200mAh – Akumulator

11.2.2.1. Opis

Akumulator litowo-polimerowy o napięciu 3,7V i pojemności 2200mAh. Stosowany w urządzeniach przenośnych wymagających wysokiej gęstości energii i niskiej masy.

11.2.2.2. Specyfikacja techniczna

- **Pojemność:** 2200mAh
- **Napięcie nominalne:** 3,7V
- **Maksymalny prąd rozładowania:** ~2A
- **Masa:** ~40-50g
- **Wymiary:** ~50 mm x 30 mm x 10 mm

11.2.2.3. Uzasadnienie wyboru

- **Długi czas pracy:** Pozwala na wielogodzinne użytkowanie urządzenia bez konieczności częstego ładowania.
- **Niska waga i kompaktowe wymiary:** Minimalizuje obciążenie obroży zwierzęcia.
- **Bezpieczeństwo:** Ochrona przed przeładowaniem i głębokim rozładowaniem.

11.2.3. BMI160 – Czujnik ruchu IMU (6-osiowy)

11.2.3.1. Opis

BMI160 to zintegrowany czujnik ruchu 6DOF (akcelerometr + żyroskop) o bardzo niskim poborze mocy. Stosowany w aplikacjach wymagających precyzyjnego śledzenia ruchu.

11.2.3.2. Specyfikacja techniczna

- **Zakres pomiarowy akcelerometru:** ±2g, ±4g, ±8g, ±16g
- **Zakres pomiarowy żyroskopu:** ±125°/s, ±250°/s, ±500°/s, ±1000°/s, ±2000°/s
- **Pobór mocy:** ~950 µA w trybie aktywnym
- **Interfejs:** I²C, SPI
- **Wymiary:** 2,5 mm x 3,0 mm x 0,8 mm

11.2.3.3. Uzasadnienie wyboru

- **Śledzenie aktywności zwierzęcia:** Wykrywanie wzmożonego ruchu może sygnalizować stres lub zdenerwowanie.
- **Bardzo niski pobór mocy:** Minimalizuje zużycie energii.
- **Miniaturowy rozmiar:** Umożliwia łatwą integrację w urządzeniu noszonym.

11.3. PIERWSZE URUCHOMIENIE I WSTĘPNE TESTY

11.3.1. PIERWSZE URUCHOMIENIE – PROBLEMY Z ZARZĄDZANIEM ENERGIĄ

Podczas pierwszych testów napotkaliśmy wiele problemów związanych z zarządzaniem energią oraz zapewnieniem odpowiedniej mocy dla wszystkich komponentów urządzenia, zwłaszcza w momencie wysyłania zapytań do API. Kluczowe problemy obejmowały:

- Niestabilność napięcia:** Nagłe spadki napięcia podczas działania ESP32-C6 w trybie aktywnej transmisji Wi-Fi, które prowadziły do resetowania urządzenia.
- Niedostateczna wydajność prądowa:** Niektóre komponenty, takie jak mikrokontroler i czujniki, wymagały szczytowych wartości prądu, które nie były dostarczane w sposób stabilny.

11.3.2. PROCES I DIAGNOZOWANIE OPTYMALIZACJI

Aby rozwiązać te problemy, skupiliśmy się na testach i dobieraniu odpowiednich elementów zasilania:

- Testy kondensatorów:** Próbowaliśmy różnych wartości pojemności kondensatorów elektrolitycznych i ceramicznych, które mogłyby skutecznie wygładzić napięcie i zapewnić wystarczającą rezerwę energii w momentach szczytowego poboru prądu. Ostatecznie zastosowanie kondensatora o pojemności 470 μF w połączeniu z mniejszymi kondensatorami ceramicznymi ($\sim 10 \mu\text{F}$) przyniosło pozytywne rezultaty.
- Zmiana przetworników:** Pierwszy zastosowany przetwornik nie był wystarczająco wydajny, co powodowało straty energii. Po testach zdecydowaliśmy się na przetwornicę Pololu S9V11F3S5, która zapewnia stabilne napięcie 3,3V i wysoką sprawność ($\sim 90\%$).
- Stabilizatory napięcia:** Dodatkowo testowaliśmy różne stabilizatory liniowe oraz układy o niskim dropout (LDO), które minimalizują straty energii i poprawiają stabilność systemu.

11.3.3. REZULTATY OPTYMALIZACJI

Dzięki powyższym testom udało się osiągnąć:

- Stabilne zasilanie ESP32-C6 nawet podczas intensywnej transmisji Wi-Fi (np. podczas wysyłania zapytań do API).
- Zminimalizowanie ryzyka resetowania urządzenia w wyniku niestabilności napięcia.
- Efektywne zarządzanie energią, co pozwoliło wydłużyć czas pracy na baterii.

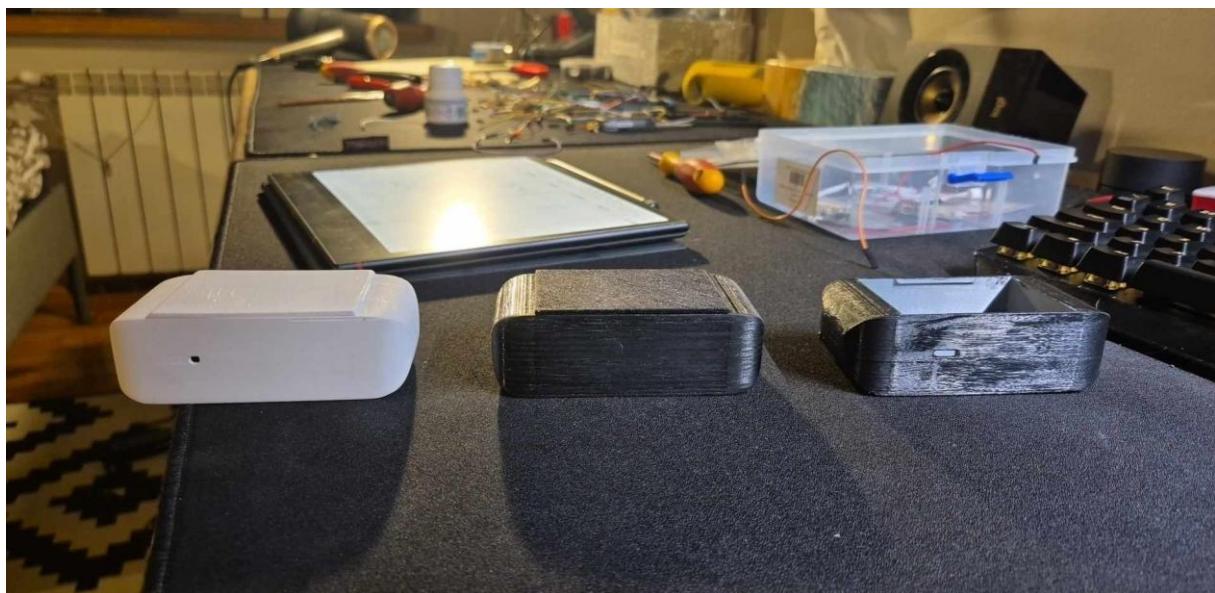
11.4. POŁĄCZENIE Z BAZĄ DANYCH I API

Urządzenie komunikuje się z API poprzez żądanie HTTP POST wysyłane na endpoint <http://74.63.9.24/data>. Żądanie zawiera nagłówek Content-Type: application/json i wysyła dane w formacie JSON, które opisują aktualny stan urządzenia.

Przesyłane informacje obejmują wykryty ruch (movement), poziom naładowania baterii (battery_percentage), zarejestrowany dźwięk (sound) oraz obliczoną odległość przemieszczenia (travel_distance). Urządzenie oczekuje odpowiedzi od serwera, a w przypadku niepowodzenia loguje kod błędu. Dodatkowo, ustawiony został 5-sekundowy timeout, aby zapewnić stabilność połączenia. Po wysłaniu danych licznik czasu ruchu zostaje zresetowany, co umożliwia kolejne pomiary.

11.5. BUDOWA CASE'A

Stworzyliśmy projekt w programie Fusion 360, a następnie wydrukowaliśmy go na drukarce 3D. Powstały 3 prototypy obudowy przedstawione na Rys. 11.1.



Rys. 11.1. Case'y do obroży. [opracowanie własne]

11.6. WYKORZYSTANE METODY

11.6.1. PRZETWARZANIE SYGNAŁU DŹWIĘKOWEGO I WYKORZYSTANIE FFT

Kod wykorzystuje mikrofon I2S do zbierania danych audio w czasie rzeczywistym. Do analizy dźwięków zastosowano **transformację Fouriera (FFT)**, która przekształca sygnał czasowy na widmo częstotliwościowe.

- **FFT umożliwia identyfikację dominujących częstotliwości** w sygnale dźwiękowym.
- Dźwięki zwierząt są identyfikowane na podstawie zdefiniowanych zakresów częstotliwości (np. miauczenie kota: **600–1600 Hz**, szczekanie psa: **100–500 Hz**).
- Wykryta dominująca częstotliwość jest porównywana z tymi zakresami, co pozwala na **rozpoznanie źródła dźwięku**.

11.6.2. WYKRYWANIE KRZTAŁTU FALI DŹWIĘKOWYCH

System analizuje amplitudy sygnału dźwiękowego w dziedzinie czasu, co pozwala na ocenę poziomu energii sygnału oraz jego kształtu.

- W celu **wyeliminowania przypadkowych szumów** stosowana jest funkcja okna Hamming'a, która **poprawia rozdzielcość częstotliwościową**.
- Jeśli poziom energii przekracza określony próg, sygnał uznawany jest za potencjalny dźwięk zwierzęcia.
- Po wykryciu odpowiedniej charakterystyki widma oraz poziomu amplitudy, system klasyfikuje dźwięk jako szczekanie lub miauczenie.

11.6.3. AUTOMATYCZNA REKALIBRACJA IMU

Aby zapewnić **dokładność pomiarów** czujnika ruchu BMI160, implementowana jest **automatyczna rekalibracja żyroskopu**.

- System **cyklicznie sprawdza poziom ruchu** – jeśli ruch nie jest wykrywany przez długi czas, żyroskop jest kalibrowany ponownie.
- **Kalibracja polega na uśrednieniu wartości odczytów żyroskopu**, co pozwala na eliminację błędów dryfu.
- Proces rekalibracji wykonywany jest **co godzinę**, o ile zwierzę nie jest w ruchu.

11.6.4. AKUMULOWANIE INFORMACJI O RUCHU

System nie tylko wykrywa ruch, ale także **mierzy czas jego trwania**, co pozwala na oszacowanie dystansu pokonanego przez zwierzę.

- Wartości przyspieszenia z **akcelerometru** są przechowywane w **buforze** i analizowane pod kątem wariancji.
- Jeśli wartości znacząco się zmieniają, oznacza to, że zwierzę jest w ruchu (**dynamiczna zmiana przyspieszenia**).
- **Czas ruchu jest akumulowany**, a na jego podstawie oraz średniej prędkości zwierzęcia obliczana jest **przybliżona odległość, jaką pokonało**.
- Po wysłaniu danych do serwera wartość akumulowanego ruchu zostaje **zresetowana**.

11.6.5. INTEGRACJA DANYCH I WYSYŁANIE DO API

Po przetworzeniu informacji o ruchu i dźwięku, dane są wysyłane do zewnętrznego serwera.

- Dane są **serializowane w formacie JSON**, zawierając informacje o:
 - Wykrytym ruchu,
 - Dźwięku i jego typie,
 - Pokonanym dystansie,
 - Poziomie naładowania baterii.
- Komunikacja odbywa się za pomocą **żądania HTTP POST**, co pozwala na łatwą integrację systemu z aplikacją monitorującą.

11.7. NAJWAŻNIEJSZE FRAGMENTY KODU

Poniżej znajduje się kod, który został wykorzystany w trakcie trwania testów. W kodzie w ważnych miejscach znajdują się szczegółowe komentarze umożliwiające dokładniejsze prześledzenie sensu wykorzystywanych metod.

11.7.1. Połączenie z Wi-Fi

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
```

W tej części kodu następuje próba połączenia z siecią Wi-Fi za pomocą podanych danych logowania (ssid i password). Dopóki urządzenie nie połączy się z siecią, będzie wyświetlać kropki w konsoli.

11.7.2. Inicjalizacja mikrofonu (przetwarzanie dźwięku)

```
DFRobot_Microphone microphone(I2S_SCK_IO, I2S_WS_IO, I2S_DI_IO);
```

Inicjalizuje mikrofon do odbierania danych dźwiękowych przez interfejs I2S. Mikrofon używa odpowiednich pinów GPIO, aby komunikować się z mikrokontrolerem.

11.7.3. Inicjalizacja i obsługa IMU (akcelerometr i żyroskop)

```
DFRobot_BMI160 bmi160;
if (bmi160.softReset() != BMI160_OK) {
    Serial.println("Błąd resetowania IMU");
    while (1);
}
```

Inicjalizuje sensor IMU (akcelerometr i żyroskop), który mierzy przyspieszenie i rotację. Jeśli urządzenie nie zostanie poprawnie zresetowane, kod zatrzyma dalsze wykonywanie.

11.7.4. Kalibracja żyroskopu

```
void calibrateGyro() {
    gyroBiasZ = 0;
    for (int i = 0; i < 500; i++) {
        int16_t accelGyro[6] = {0};
        if (bmi160.getAccelGyroData(accelGyro) == 0) {
            gyroBiasZ += accelGyro[2]; // Sumowanie wartości osi Z
        }
        delay(5);
    }
}
```

```

        }
        gyroBiasZ           /=      500;
        gyroBiasZ           /=     16.4;
    }
}

```

Funkcja kalibruje żyroskop przez zbieranie danych przez pewien czas (500 próbek), a następnie oblicza wartość przesunięcia (bias) na osi Z. Dzięki temu można uzyskać dokładniejsze odczyty ruchu.

11.7.5. Przetwarzanie danych dźwiękowych z mikrofonu

```

int samplesRead = microphone.read(i2sReadBuff,
sizeof(i2sReadBuff));

```

Mikrofon zbiera próbki dźwiękowe w postaci surowych danych i zapisuje je w buforze i2sReadBuff. Te próbki są następnie analizowane przez funkcję FFT (Fast Fourier Transform), aby uzyskać częstotliwości i rozpoznać dźwięki, takie jak miauczenie kota lub szczekanie psa.

11.7.6. Obliczanie częstotliwości szczytowej

```

double getPeakFrequency(double* vReal) {
    int peakIndex = 1;
    double maxMagnitude = 0;
    for (int i = 2; i < FFT_SIZE / 2; i++) {
        if (vReal[i] > maxMagnitude) {
            maxMagnitude = vReal[i];
            peakIndex = i;
        }
    }
    return (peakIndex * (SAMPLE_RATE / 2.0)) / (FFT_SIZE / 4.0);
}

```

Funkcja getPeakFrequency analizuje dane FFT, aby znaleźć częstotliwość o największej amplitudzie, co pozwala na rozpoznanie, czy dźwięk to miauczenie kota, szczekanie psa lub inny dźwięk.

11.7.7. Wykrywanie ruchu na podstawie danych z IMU

```

if (isDynamicMovement) {
    accumulatedMovementTime += 0.1; // Akumulacja czasu ruchu w sekundach
    movementDetected = true;
}

```

System monitoruje przyspieszenia z akcelerometru, aby wykryć dynamiczny ruch. Jeśli ruch jest wykryty, akumuluje czas ruchu (w sekundach) i ustawia flagę movementDetected na true.

11.7.8. Wysyłanie danych do serwera (API)

```
void sendApiCall() {  
    HTTPClient http;  
    String url = "http://74.63.9.24/data";  
    http.begin(url);  
    http.addHeader("Content-Type", "application/json");  
  
    String jsonPayload = "{\"events\": [{";  
    jsonPayload += "\"movement\": " + String(movementEvent ? "true"  
: "false") + ", ";  
    jsonPayload += "\"sound\": " + String(soundEvent ? "true" :  
"false") + ", ";  
    jsonPayload += "\"soundType\": \"\" + soundType + "\", ";  
    jsonPayload += "\"distance\": " + String(distance, 4);  
    jsonPayload += "}]}}";  
  
    int httpResponseCode = http.POST(jsonPayload);  
}
```

Funkcja sendApiCall zbiera dane o wykrytych zdarzeniach (ruch, dźwięk) i wysyła je na serwer w formacie JSON. Serwer odbiera te dane, co umożliwia późniejsze analizowanie zdarzeń.

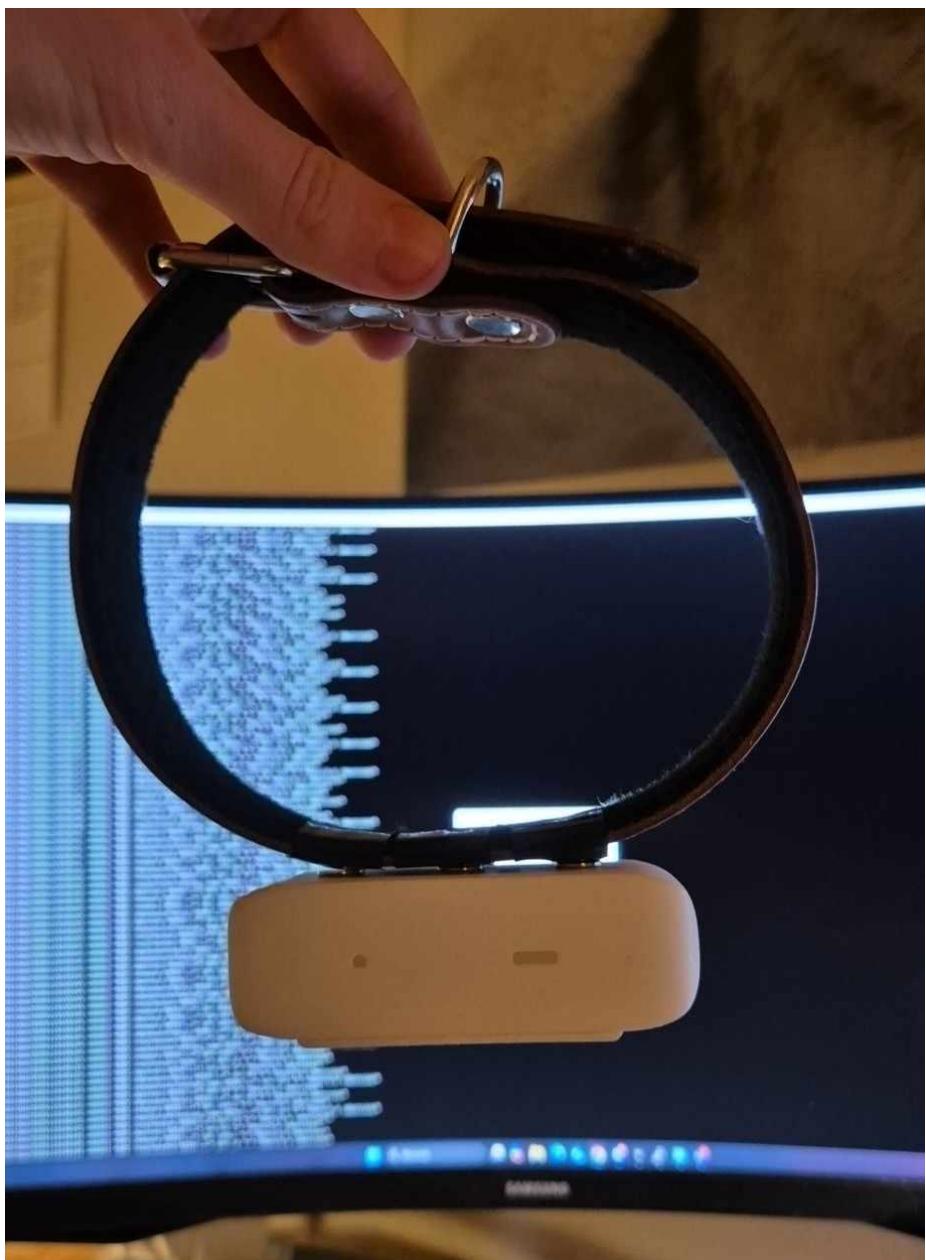
11.7.9. Tryb uśpienia (oszczędność energii)

```
esp_sleep_enable_timer_wakeup(1000); // 1ms  
esp_light_sleep_start();
```

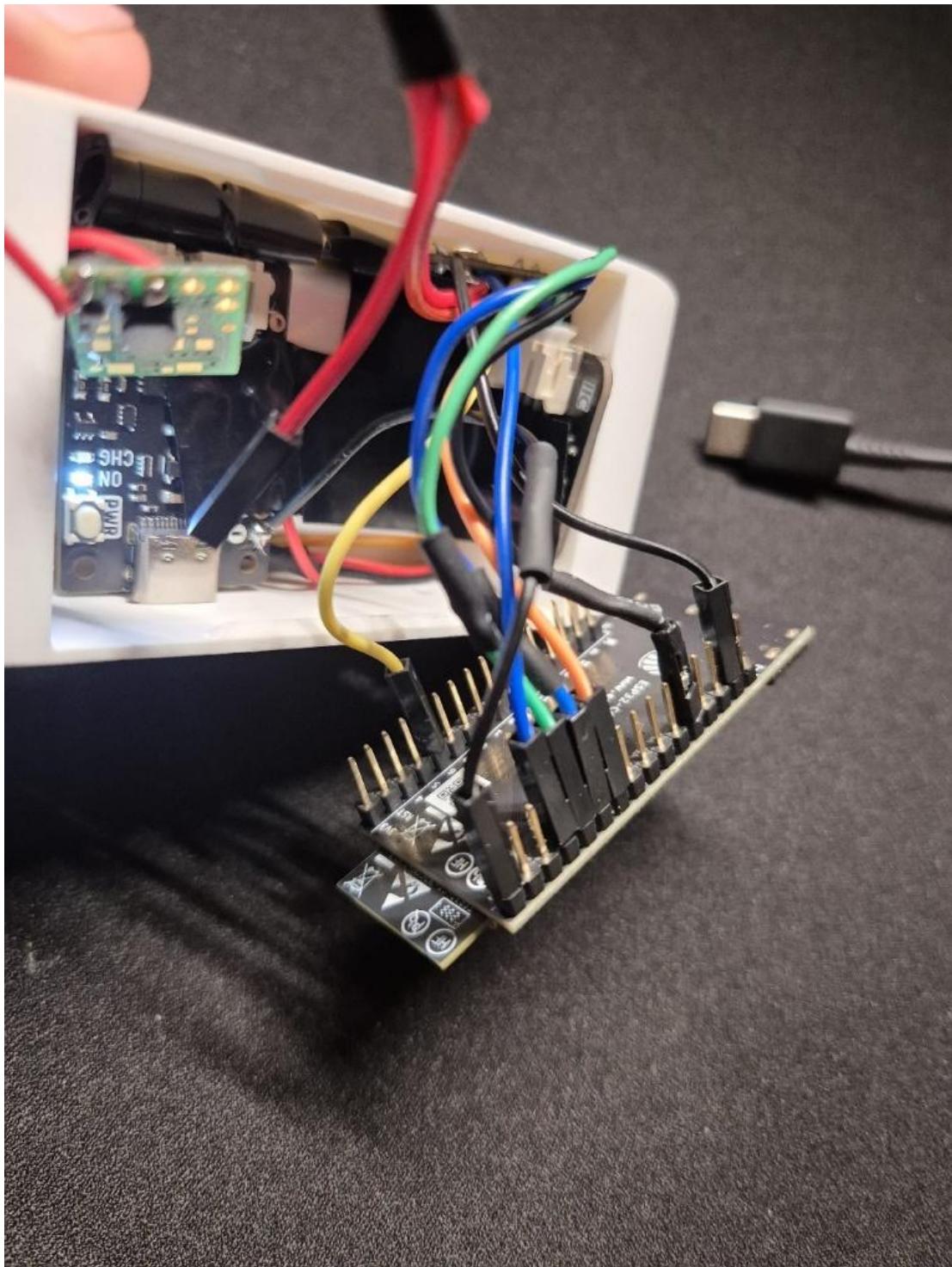
Aby oszczędzać energię, system wchodzi w tryb lekkiego uśpienia, budząc się co 1 milisekundę, aby wykonać dalsze czynności w pętli głównej.

11.8. OSTATECZNA WERSJA URZĄDZENIA

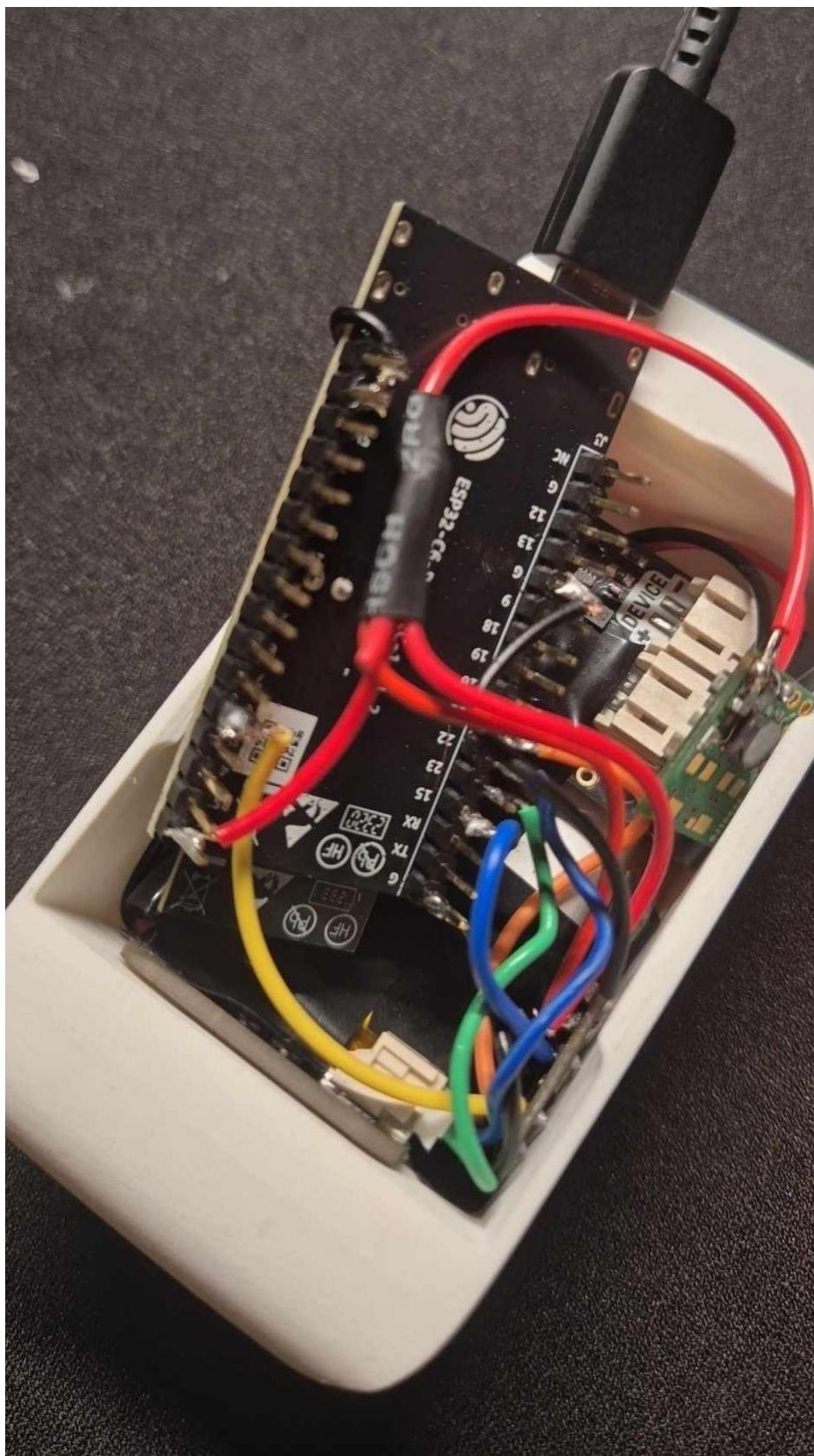
Na poniższych fotografiach (**Rys. 11.2**, **Rys. 11.3**, **Rys. 11.4**) przedstawiony jest gotowy produkt, który służył do przeprowadzenia testów. Wszystkie komponenty wewnętrzne zostały zlutowane i zabezpieczone taśmą izolacyjną. Sam Case idealnie pomieścił wszystkie elementy urządzenia w taki sposób, aby nic w środku nie przemieszczało się w trakcie ruchu zwierzęcia.



Rys. 11.2. Ostateczna wersja produktu bez wewnętrznej elektroniki.



Rys. 11.3. Połączenia w ostatecznej wersji urządzenia. [opracowanie własne]



Rys. 11.4. Połączenia w ostatecznej wersji urządzenia. [opracowanie własne]

12. ARCHITEKTURA API

12.1. WPROWADZENIE

API do monitorowania aktywności zwierząt domowych to kompleksowe rozwiązanie IoT umożliwiające śledzenie i analizę zachowania zwierząt w czasie rzeczywistym. System zbiera dane z czujników ruchu, detektorów dźwięku, oferując zaawansowane funkcje analityczne i interaktywne mechanizmy reagowania.

12.1.1. KLUCZOWE FUNKCJE API

◆ Śledzenie aktywności w czasie rzeczywistym

- Rejestrowanie ruchu i dźwięków
- Pomiar przebytego dystansu
- Monitorowanie poziomu baterii urządzenia

◆ Zaawansowane analizy statystyczne

- Tworzenie raportów dziennych i tygodniowych
- Identyfikacja szczytowych godzin aktywności
- Obliczanie czasu odpoczynku zwierzęcia

◆ Integracja z interaktywną zabawką

- Automatyczna aktywacja zabawki na podstawie aktywności zwierzęcia
- Możliwość ręcznego sterowania przez API
- Bezpieczne mechanizmy deaktywacji

◆ Narzędzia diagnostyczne i optymalizacyjne

- Wizualizacja danych aktywności
- Monitorowanie zużycia baterii
- Mechanizmy czyszczenia bazy danych

Dzięki połączeniu technologii IoT, analizy danych i interaktywnych funkcji, API stanowi nowoczesne rozwiązanie wspierające właścicieli zwierząt w dbaniu o ich zdrowie i samopoczucie.

12.2. ARCHITEKTURA SERWERA

Serwer API oparty jest na nowoczesnej architekturze, zapewniającej wydajność, niezawodność i łatwość w utrzymaniu. Jego komponenty są zaprojektowane w taki sposób, aby obsługiwały dużą liczbę zapytań, zapewniały ciągłą dostępność oraz umożliwiały prostą aktualizację kodu.

12.2.1. INFRASTRUKTURA SERWERA

1. Serwer fizyczny:

- a. Serwer znajduje się w centrum danych w **Frankfurcie**, w hostingowej infrastrukturze **Horizon CloudFerro**. Dzięki temu zapewniona jest wysoka dostępność oraz niskie opóźnienia w dostępie do API, szczególnie dla użytkowników w Europie.

2. Nginx (Proxy):

- a. **Nginx** działa jako **reverse proxy**, zapewniając wysoką wydajność i optymalizację ruchu przychodzącego do serwera. Nginx odpowiada za obsługę połączeń HTTP, kierując je do aplikacji backendowej uruchomionej na **Uvicornie**.
- b. Dodatkowo, Nginx pełni funkcję obsługi statycznych zasobów oraz zarządzania bezpieczeństwem, w tym SSL/TLS dla bezpiecznych połączeń HTTPS.

3. Uvicorn (ASGI Server):

- a. **Uvicorn** jest serwerem aplikacji działającym w standardzie **ASGI** (Asynchronous Server Gateway Interface), który umożliwia asynchroniczne przetwarzanie zapytań. Uvicorn obsługuje aplikację **FastAPI**, która jest fundamentem API do monitorowania aktywności zwierząt domowych.
- b. Uvicorn, w połączeniu z FastAPI, umożliwia wyjątkową wydajność w przetwarzaniu dużej liczby jednoczesnych zapytań, co jest kluczowe w środowiskach IoT.

4. Integracja z GitHubem (CI/CD):

- a. Kod serwera jest zarządzany przez system kontroli wersji **GitHub**. Zmiany w kodzie są automatycznie aktualizowane na serwerze dzięki **procesowi Continuous Integration / Continuous Deployment (CI/CD)**.
- b. Po zatwierdzeniu zmian w repozytorium na GitHubie, system automatycznie uruchamia proces aktualizacji na serwerze. Dzięki temu nowe funkcje i poprawki są wdrażane bez potrzeby manualnego interweniowania, co skraca czas wprowadzania poprawek oraz zapewnia aktualność środowiska produkcyjnego.

12.2.2. BEZPIECZEŃSTWO I MONITORING

1. Monitoring i logowanie:

- a. Serwer jest wyposażony w mechanizmy monitorujące jego wydajność i dostępność. Dane dotyczące stanu serwera, błędów aplikacji oraz wykrytych incydentów są rejestrowane i przechowywane w logach.
- b. Dzięki temu możliwe jest śledzenie i diagnozowanie problemów w czasie rzeczywistym oraz zapewnienie wysokiej dostępności API.

12.3. ARCHITEKTURA SYSTEMU

12.3.1. WYMAGANE BIBLIOTEKI DO FASTAPI

Aby uruchomić system oparty na **FastAPI**, konieczne jest zainstalowanie kilku podstawowych bibliotek, które wspierają działanie aplikacji oraz jej integrację z bazą danych, API, i innymi komponentami.

1. **FastAPI:**

FastAPI to główny framework do tworzenia API, który jest wykorzystywany w tym projekcie. Dzięki jego asynchronicznemu podejściu, zapewnia on bardzo wysoką wydajność w obsłudze zapytań.

2. **Uvicorn:**

Uvicorn to serwer ASGI, który uruchamia aplikację FastAPI. Jest lekki i szybki, idealny do obsługi asynchronicznych zapytań.

3. **SQLite3:**

SQLite3 jest bazą danych, wykorzystywaną do przechowywania danych w tym projekcie. Jest to lokalna baza danych, która przechowuje zdarzenia i inne dane związane z monitorowaniem zwierząt.

4. **Pydantic:**

Pydantic jest biblioteką do walidacji danych wejściowych i modeli. Jest integralną częścią FastAPI, umożliwiającą łatwą walidację danych przychodzących w zapytaniach API.

5. **Matplotlib:**

Matplotlib to biblioteka do generowania wykresów. Jest używana do tworzenia wizualizacji danych dotyczących aktywności zwierząt, takich jak dystans pokonany w określonym czasie.

6. **pytz:**

pytz to biblioteka do obsługi stref czasowych w Pythonie. Jest używana w systemie do konwersji czasów w bazie danych na odpowiednią strefę czasową (np. Warszawa).

12.3.2. OPIS API

API do monitorowania aktywności zwierząt domowych opiera się na framework'u **FastAPI** i zapewnia szereg funkcji, które umożliwiają zbieranie, analizowanie oraz prezentowanie danych związanych z aktywnością zwierząt. Poniżej znajduje się przegląd głównych komponentów API:

1. Rejestracja Zdarzeń:

- a. Użytkownicy mogą przesyłać dane związane z ruchami i dźwiękami zwierząt (np. detekcja dźwięku, ruchu). Dane są zapisywane w bazie danych SQLite3, a każda nowa rejestracja jest odpowiednio walidowana.
- b. Endpointy:
 - i. POST /data — rejestracja zdarzeń (ruchu, dźwięku, dystansu, poziomu baterii).

2. Śledzenie Aktywności:

- a. API umożliwia użytkownikom monitorowanie aktywności zwierząt w czasie rzeczywistym. Użytkownicy mogą sprawdzać, czy dane zdarzenia (ruch i dźwięk) miały miejsce w ciągu ostatnich kilku sekund.
- b. Endpointy:
 - i. GET /events — lista wydarzeń.
 - ii. GET /analytics/distance — analiza dystansu pokonanego przez zwierzę.

3. Integracja z Zabawą:

- a. API obsługuje interaktywne zabawki, które mogą być automatycznie włączane lub wyłączane w zależności od wykrytych zdarzeń. Istnieje możliwość ręcznego sterowania zabawką przez użytkownika.
- b. Endpointy:
 - i. GET /toy/status — sprawdzenie statusu zabawki.
 - ii. POST /toy/toggle — włączenie lub wyłączenie zabawki.

4. Analizy i Raportowanie:

- a. API oferuje zaawansowane mechanizmy analityczne, takie jak obliczanie średnich dziennych dystansów, szczytowych godzin aktywności, czy całkowitego dystansu pokonanego przez zwierzę.
- b. Endpointy:
 - i. GET /analytics/graph — generowanie wykresów aktywności.
 - ii. GET /analytics/resting_time — obliczanie czasu odpoczynku.

5. Wizualizacja Danych:

- a. Dzięki **Matplotlib**, API może generować wykresy z danymi, które są prezentowane użytkownikom w formie wizualnej, np. wykresy przedstawiające aktywność w danym czasie.
- b. Endpointy:
 - i. GET /analytics/graph — generowanie wykresu aktywności.

6. Zarządzanie Danymi:

- a. API oferuje funkcje zarządzania danymi, takie jak czyszczenie bazy danych w celu usunięcia starych rekordów, które już nie są potrzebne.

- b. Endpointy:
 - i. DELETE /maintenance/cleanup — usuwanie starych danych.

7. Monitoring Baterii:

- a. API umożliwia monitorowanie poziomu baterii urządzeń, które są wykorzystywane do zbierania danych (np. czujników ruchu).
- b. Endpointy:
 - i. GET /battery — informacje o poziomie baterii i jej zużyciu.

12.4. ARCHITEKTURA BAZY DANYCH

Baza danych systemu wykorzystuje strukturę SQLite i przechowuje informacje w tabeli **events**, która ma zdefiniowaną następującą strukturę:

- **id**: Unikalny identyfikator zdarzenia (INTEGER, AUTOINCREMENT).
- **timestamp**: Znacznik czasu rejestracji zdarzenia (DATETIME).
- **movement**: Flaga wykrycia ruchu (BOOLEAN: 1 – ruch wykryty, 0 – brak).
- **sound**: Flaga wykrycia dźwięku (BOOLEAN: 1 – dźwięk wykryty, 0 – brak).
- **sound_type**: Typ dźwięku (TEXT, np. "dog_bark"), wymagany tylko, gdy **sound** = 1.
- **battery_percentage**: Poziom baterii urządzenia (REAL, zakres 0–100).
- **travel_distance**: Przebyty dystans w metrach (REAL, wartość nieujemna).

Przykładowe dane pokazują typowe scenariusze:

- **Rejestracja ciągłego ruchu**: np. rekordy 1–10, gdzie **movement** = 1, **sound** = 0.
- **Pojedyncze zdarzenie z dźwiękiem**: np. rekord 17, gdzie **sound** = 1, **sound_type** = "dog_bark".
- **Zmiana dystansu**: np. rekord 14, gdzie następuje skok z 0 do 2,2 m.

W przykładowych danych na Rys. 12.1 poziom baterii jest stały (95%), co potwierdza poprawność działania walidacji. W rzeczywistych warunkach jednak wartość baterii będzie stopniowo malała. Tego typu dane służą do analizy zachowań zwierzęcia, generowania statystyk oraz automatycznej aktywacji zabawki w odpowiedzi na jednoczesny wykryty ruch i dźwięk. Struktura bazy zapewnia efektywne zarządzanie danymi historycznymi oraz integrację z modułem analitycznym API.

	id	timestamp	movement	sound	sound_ty...	battery_p...	travel_dis...
	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
1	1	2025-01-26 12:19:14	1	0	NULL	95	0
2	2	2025-01-26 12:19:19	1	0	NULL	95	0
3	3	2025-01-26 12:19:24	1	0	NULL	95	0
4	4	2025-01-26 12:19:29	1	0	NULL	95	0
5	5	2025-01-26 12:19:34	1	0	NULL	95	0
6	6	2025-01-26 12:19:39	1	0	NULL	95	0
7	7	2025-01-26 12:19:44	1	0	NULL	95	0
8	8	2025-01-26 12:19:49	1	0	NULL	95	0
9	9	2025-01-26 12:19:54	1	0	NULL	95	0
10	10	2025-01-26 12:19:59	1	0	NULL	95	0
11	11	2025-01-26 12:20:04	1	0	NULL	95	0
12	12	2025-01-26 12:26:53	1	0	NULL	95	1.05
13	13	2025-01-26 12:26:58	1	0	NULL	95	1.05
14	14	2025-01-26 12:27:03	1	0	NULL	95	2.2
15	15	2025-01-26 12:27:08	1	0	NULL	95	1.2
16	16	2025-01-26 12:27:13	1	0	NULL	95	2.1
17	17	2025-01-26 12:27:18	1	1	dog_bark	95	2.25
18	18	2025-01-26 12:27:23	1	1	dog_bark	95	2.2
19	19	2025-01-26 12:27:28	1	0	NULL	95	1.55
20	20	2025-01-26 13:27:53	0	1	dog_bark	95	0
21	21	2025-01-26 14:03:09	1	0	NULL	95	0.75
22	22	2025-01-26 14:03:14	1	1	dog_bark	95	2.05
23	23	2025-01-26 14:03:19	1	0	NULL	95	2
24	24	2025-01-26 14:03:23	1	0	NULL	95	2.2
25	25	2025-01-26 14:03:29	1	1	dog_bark	95	2.25
26	26	2025-01-26 14:03:33	1	0	NULL	95	1.7
27	27	2025-01-26 14:03:39	1	1	dog_bark	95	0.3
28	28	2025-01-26 14:03:44	1	0	NULL	95	0.75
+	157	29	2025-01-26 14:03:49	1	0	NULL	95
							2.1

Rys. 12.1. Wizualna reprezentacja części danych [opracowanie własne]

12.5. ARCHITEKTURA ENDPOINTÓW

Endpointy w systemie są zorganizowane w pięć głównych grup funkcjonalnych, odpowiadających różnym aspektom monitorowania aktywności zwierzęcia oraz zarządzania systemem. Poniżej znajduje się szczegółowy opis poszczególnych grup oraz ich funkcji.

12.5.1. KONTROLA ZABAWKI(/toy/*)

Grupa endpointów przeznaczona do sterowania interaktywną zabawką oraz monitorowania jej stanu.

- **GET /toy/status**
- Pobiera aktualny status zabawki (włączona/wyłączona).
- **POST /toy/toggle**

Przełącza stan zabawki (np. z "włączona" na "wyłączona").

- **POST /toy/set**

Ustawia jawnego stan zabawki (np. wymusza włączenie).

- **GET /toy/run**

Sprawdza, czy zabawka powinna się uruchomić, bazując na ostatnich zarejestrowanych zdarzeniach.

Integracja:

Dane o stanie zabawki są przechowywane w osobnej bazie danych **toy_state.db**, a logika decyzyjna opiera się na analizie zdarzeń zapisanych w tabeli **events**.

12.5.2. ZARZĄDZANIE DANYMI (/data, /events)

Grupa endpointów odpowiedzialna za zapis i odczyt danych pochodzących z czujników.

- **POST /data**

Zapisuje listę zdarzeń (np. ruch, dźwięk) do bazy danych SQLite.

Wymaga walidacji przez model **Pydantic** (EventList).

- **GET /events**

Zwraca listę ostatnich zdarzeń z paginacją (limit, offset).

Struktura danych:

Każde zdarzenie zawiera informacje takie jak: znacznik czasu, poziom baterii, dystans, flagi dotyczące ruchu i dźwięku.

12.5.3. ŚLEDZENIE DYSTANSU (/distance/*)

Grupa endpointów odpowiedzialna za analizowanie przebytego dystansu w różnych przedziałach czasowych.

- **GET /distance/current**

Zwraca dystans pokonany w bieżącej godzinie.

- **GET /distance/daily**

Zwraca całkowity dystans pokonany w ciągu dnia.

- **GET /distance/total**

Zwraca łączny dystans pokonany od początku działania systemu.

Mechanizm:

Dane są agregowane z kolumny **travel_distance** w tabeli **events** za pomocą zapytań SQL, takich jak **SUM** i **GROUP BY**.

12.5.4. ANALITYKA I RAPORTY (/analytics/*, /battery)

Grupa endpointów przeznaczona do generowania statystyk i wizualizacji aktywności zwierzęcia oraz monitorowania stanu baterii.

- **GET /analytics/resting_time**

Oblicza czas odpoczynku zwierzęcia (brak wykrywanego ruchu).

- **GET /analytics/distance**

Zwraca zestawienie średniego dziennego dystansu, tygodniowego dystansu oraz godzin szczytowej aktywności.

- **GET /analytics/graph**

Generuje dane do wykresów aktywności (JSON z danymi godzinowymi).

- **GET /battery**

Pokazuje aktualny poziom baterii oraz dzienne zużycie.

Optymalizacja:

Do analizy danych wykorzystane są zaawansowane zapytania SQL, w tym **strftime** do grupowania po czasie oraz konwersja stref czasowych (np. Warszawa).

12.5.5. KONSERWACJA I DEBUGOWANIE (/maintenance/, /debug/)

Grupa endpointów oferująca narzędzia dla administratorów systemu, umożliwiające konserwację danych oraz diagnostykę.

- **DELETE /maintenance/cleanup**

Usuwa stare rekordy z bazy danych (domyślnie starsze niż 30 dni).

- **GET /debug/distance**

Zwraca ostatnie 10 wpisów dotyczących dystansu (przeznaczone do diagnostyki).

Bezpieczeństwo:

Operacje czyszczenia danych wymagają ręcznej aktywacji, aby zapobiec przypadkowemu usunięciu istotnych danych.

12.5.6. KLUCZEOWE CECHY ARCHITEKTONICZNE

1. Modułowość:

Każda grupa endpointów jest odpowiedzialna za odrębny obszar funkcjonalny, co umożliwia łatwe zarządzanie i rozbudowę systemu.

2. Weryfikacja Danych:

Wszystkie dane wejściowe są walidowane przy użyciu modeli **Pydantic**, co zapewnia integralność danych (np. zakres baterii 0–100%).

3. Wydajność:

Zapytania SQL wykorzystują odpowiednie indeksowanie oraz optymalizowane funkcje agregujące, co pozwala na szybkie przetwarzanie danych.

4. Rozszerzalność:

Struktura systemu umożliwia łatwe dodawanie nowych endpointów, np. w celu obsługi dodatkowych czujników, urządzeń lub funkcji.

12.5.7. PPRZYKŁAD PRZEPŁYWU DANYCH

1. **POST /data** → Zapisanie zdarzeń związanych z aktywnością zwierzęcia w bazie danych.
2. **POST /toy/run** → Na podstawie zarejestrowanych zdarzeń system automatycznie uruchamia interaktywną zabawkę.
3. **GET /analytics/distance** → Zaktualizowanie statystyk dotyczących pokonanego dystansu przez zwierzę.

Na poniżej grafice na **Rys. 12.2** znajduje się podsumowanie wszystkich dostępnych endpointów z ich wytłumaczeniami. Takie podsumowanie endpointów ułatwiło nam integrację API z pozostałymi częściami systemu.

The screenshot shows the FastAPI documentation interface. At the top, it says "FastAPI 0.1.0 OAS 3.1" and has a link to "/openapi.json". Below this, there's a section titled "default" which lists several API endpoints:

- GET** /toy/status Get Toy Status
- POST** /toy/toggle Toggle Toy
- POST** /toy/set Set Toy State
- GET** /toy/run Check Toy Run
- POST** /data Save Events
- GET** /distance/current Get Current Hour Distance
- GET** /distance/daily Get Daily Distance
- GET** /distance/total Get Total Distance
- GET** /analytics/resting_time Get Resting Time
- GET** /analytics/distances Get Distance Analytics
- GET** /debug/distance Debug Distance
- GET** /events List Events
- GET** /analytics/graph Get Analytics Graph
- GET** /battery Get Battery Stats
- DELETE** /maintenance/cleanup Cleanup Database

Rys. 12.2. Wizualna reprezentacja części danych. [opracowanie własne]

12.6. NAJWAŻNIEJSZE FRAGMENTY KODU

Poniżej przedstawiliśmy kod API wraz z komentarzami. FAST API umożliwia proste tworzenie endpointów udostępniając możliwość szybkiego zaprojektowania koniecznych danych wejściowych oraz logiki umożliwiającej zwrócić konkretne informacje.

12.6.1. Inicjalizacja Bazy Danych

```
# Database configuration
DB_NAME = "pet_tracker.db"

def initialize_db():
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("""CREATE TABLE IF NOT EXISTS events (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
        movement BOOLEAN NOT NULL CHECK (movement IN (0, 1)),
        sound BOOLEAN NOT NULL CHECK (sound IN (0, 1)),
        sound_type TEXT,
        battery_percentage REAL NOT NULL,
        travel_distance REAL NOT NULL
    )""")
    conn.commit()
```

```
conn.close()

initialize_db()
```

12.6.2. Modele Pydantic z Walidacjami

```
from pydantic import BaseModel, field_validator
from typing import Optional

class Event(BaseModel):
    movement: bool
    sound: bool
    sound_type: Optional[str] = None
    battery_percentage: float
    travel_distance: float

    @field_validator('travel_distance')
    def validate_distance(cls, value):
        if value < 0:
            raise ValueError("Distance cannot be negative")
        return round(value, 2)

    @field_validator('battery_percentage')
    def validate_battery(cls, value):
        if not 0 <= value <= 100:
            raise ValueError("Battery percentage must be between 0 and 100")
        return round(value, 2)
```

12.6.3. Obsługa Stanu Zabawki

```
# Helper function to get the current state
def get_toy_state():
    conn = sqlite3.connect('toy_state.db')
    cursor = conn.cursor()
    cursor.execute('SELECT runToy FROM toy_state WHERE id = 1')
    result = cursor.fetchone()
    conn.close()
    return result[0] if result else False

# Helper function to update the state
```

```

def update_toy_state(new_state):
    conn = sqlite3.connect('toy_state.db')
    cursor = conn.cursor()
    cursor.execute('UPDATE toy_state SET runToy = ? WHERE id = 1',
    (new_state,))
    conn.commit()
    conn.close()

```

12.6.4. Główne Endpointy (Status Zabawki, Włączanie, Sprawdzanie Aktywności)

```

@app.get("/toy/status")
async def get_toy_status():
    runToy = get_toy_state()
    return {"runToy": runToy}

@app.post("/toy/toggle")
async def toggle_toy():
    current_state = get_toy_state()
    new_state = not current_state
    update_toy_state(new_state)
    return {"message": f"Toy state toggled to {'on' if new_state else 'off'}", "runToy": new_state}

@app.get("/toy/run")
async def check_toy_run():
    runToy = get_toy_state()
    if not runToy:
        return {
            "should_run": False,
            "message": "Toy is currently turned off"
        }
    conn = get_db_connection()
    cursor = conn.cursor()
    cursor.execute("""SELECT movement, sound FROM events WHERE
    timestamp >= datetime('now', '-5 seconds')""")
    events = cursor.fetchall()
    if not events:
        return {"should_run": False, "message": "No recent events
found"}
        should_run = any(event["movement"] and event["sound"] for event
in events)
        return {"should_run": should_run, "message": "Toy should run" if

```

```
should_run else "No recent movement and sound events"}
```

12.6.5. Endpointy do Zapisania Danych i Sprawdzania Dystansu

```
@app.post("/data")
async def save_events(data: EventList):
    conn = get_db_connection()
    cursor = conn.cursor()
    for event in data.events:
        if event.sound and not event.sound_type:
            raise HTTPException(400, "Sound type required when sound
is true")
        cursor.execute("""INSERT INTO events (movement, sound,
sound_type, battery_percentage, travel_distance) VALUES (?, ?, ?, ?, ?)""",
                      (int(event.movement), int(event.sound),
event.sound_type, event.battery_percentage, event.travel_distance))
    conn.commit()
    return {"message": f"Added {len(data.events)} events",
"valid_entries": len(data.events)}

@app.get("/distance/current")
async def get_current_hour_distance():
    conn = get_db_connection()
    cursor = conn.cursor()
    cursor.execute("""SELECT SUM(travel_distance) as distance FROM
events WHERE strftime('%Y-%m-%d %H', timestamp) = strftime('%Y-%m-%d
%H', 'now')""")
    result = cursor.fetchone()
    conn.close()
    return {
        "current_hour": datetime.now().strftime("%Y-%m-%d %H:00"),
        "distance": round(result["distance"] or 0.0, 2),
        "unit": "meters"
    }
```

12.6.6. Analytics: Średnia Dzienna, Tygodniowy Dystans, i Analiza Godziny Szczytu

```
@app.get("/analytics/distance")
async def get_distance_analytics():
```

```

conn = get_db_connection()
cursor = conn.cursor()

stats = {}

    cursor.execute("""SELECT AVG(daily_distance) as avg_daily FROM
(SELECT SUM(travel_distance) as daily_distance FROM events GROUP BY
date(timestamp))""")
    stats["daily_average"] = round(cursor.fetchone()["avg_daily"] or
0.0, 2)

    cursor.execute("SELECT SUM(travel_distance) FROM events WHERE
timestamp >= date('now', '-7 days')")
    stats["weekly_total"] = round(cursor.fetchone()[0] or 0.0, 2)

    cursor.execute("""SELECT strftime('%H:00', timestamp) as hour,
SUM(travel_distance) as distance FROM events GROUP BY hour ORDER BY
distance DESC LIMIT 1""")
    peak = cursor.fetchone()
    stats["peak_hour"] = {
        "hour": peak["hour"] if peak else None,
        "distance": round(peak["distance"], 2) if peak else 0.0
    }

conn.close()
return stats

```

12.6.7.7. Generowanie Wykresu z Danych

```

@app.get("/analytics/graph")
async def get_analytics_graph(hours: int = 24):
    if hours <= 0:
        raise HTTPException(status_code=400, detail="Time range must
be greater than 0 hours")

    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("""SELECT strftime('%Y-%m-%d %H:00', timestamp)
as hour, SUM(movement) as movement_count, SUM(sound) as sound_count,
SUM(travel_distance) as distance FROM events WHERE timestamp >=
datetime('now', ?) GROUP BY hour ORDER BY hour""", (f"-{hours}
hours",))

```

```

data = cursor.fetchall()

if not data:
    return {"message": f"No data found for the last {hours} hours", "data": None}

result = {"hours": [], "movement": [], "sound": [], "distance": []}
warsaw_tz = pytz.timezone('Europe/Warsaw')

for row in data:
    utc_time = datetime.strptime(row["hour"], '%Y-%m-%d %H:00')
    warsaw_time =
utc_time.replace(tzinfo=pytz.utc).astimezone(warsaw_tz)
    formatted_time = warsaw_time.strftime('%Y-%m-%d %H:00')

    result["hours"].append(formatted_time)
    result["movement"].append(row["movement_count"] or 0)
    result["sound"].append(row["sound_count"] or 0)
    result["distance"].append(row["distance"] or 0)

return {"message": f"Data for the last {hours} hours (Warsaw time)", "data": result}

```

12.6.8.8. Czyszczenie Bazy Danych (Usuwanie Starych Rekordów)

```

@app.delete("/maintenance/cleanup")
async def cleanup_database(older_than_days: int = 30):
    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("DELETE FROM events WHERE timestamp <
datetime('now', ?)", (f"-{older_than_days} days",))

    deleted_count = cursor.rowcount
    conn.commit()
    conn.close()

    return {"message": f"Removed {deleted_count} records",
"retention_period": f"{older_than_days} days"}

```

13. PROTOTYPOWANIE ZABAWKI

13.1. OKREŚLENIE WYMAGAŃ I FUNKCJONALNOŚCI URZĄDZENIA

Główym celem urządzenia jest zapewnienie zwierzętom (takim jak koty czy psy) interaktywnej rozrywki, która pobudza instynkt łowiecki oraz aktywizuje do ruchu. Kluczowe wymagania prototypu obejmują:

- Stabilną i bezpieczną konstrukcję mechanizmu (tak aby zwierzę nie zrobiło sobie krzywdy podczas zabawy),
- Odpowiednią szybkość obrotową silnika (przy 12 V ~1850 rpm, z możliwością zwiększenia do 24 V dzięki regulacji można dostosować do potrzeb zwierzęcia),
- Trójkatne rozmieszczenie bloczków, pozwalające na swobodne poruszanie się linki oraz zapewniające jej właściwe napięcie,
- Możliwość sterowania i zasilania silnika w sposób, który nie spowoduje przegrzania czy uszkodzenia podzespołów,
- Integrację z mikrokontrolerem ESP32, który umożliwia komunikację z serwerem a także kontrolę przez aplikację.

W praktyce, urządzenie powinno działać w sposób ciągły przez określony czas, nie powodując zagrożenia dla zwierząt ani użytkownika. Dlatego ważne jest zastosowanie silnika oraz drivera o odpowiednich parametrach i zachowanie bezpiecznych wartości napięcia.

13.2. WYBÓR KOMPONENTÓW

Wybór tych komponentów podyktowany był głównie dostępnością, łatwością montażu i konfiguracji.

- **Silnik DC 540:** charakteryzuje się wystarczającą mocą i momentem obrotowym (około 1850 rpm) do napędzania linki, zapewniając jednocześnie możliwość zwiększenia napięcia nawet do 24 V (przy zachowaniu odpowiedniego chłodzenia i nieprzekraczaniu limitów urządzenia).
- **Sterownik L298N:** pozwala na dwukierunkowe sterowanie silnikiem DC, posiada wbudowane zabezpieczenia, a jego parametry prądowe dobrze współgrają z silnikiem 540.
- **Mikrokontroler ESP32:** wybrany ze względu na dużą liczbę pinów, wbudowaną obsługę Wi-Fi, a także możliwość łatwego programowania w Arduino IDE. Kluczowym elementem przy wyborze mikrokontrolera była tu obsługa sieci Wi-Fi, dzięki czemu możliwe było odbieranie i wysyłanie komend na serwer, co przekłada się na zdalne sterowanie pracą silnika.
- **Linka wędkarska:** dzięki niskiej masie nie będzie dodatkowo obciążać silnika, a materiał (nylon) zapewnia wysoką odporność na przetarcia
- **Bloczki:** poliamidowe bloczki zapewniają niskie tarcie, pozwalające na swobodne ślizganie się linki

- **Zasilacz:** dobrany tak, aby dostarczał odpowiednie napięcie (12–24 V) i prąd umożliwiający stabilną pracę silnika oraz sterownika.

13.3. PIERWSZE URUCHOMIENIE I TESTY

Pierwsze uruchomienie układu odbyło się z pomocą Arduino, aby zweryfikować poprawność konstrukcji oraz działanie mechaniczne całego systemu. Pozwoliło to na wstępne sprawdzenie, czy wszystkie elementy są właściwie wyosiowane, a linka nie zsuwa się z bloczków, program co 5 sekund włączał silniczek na 10 sekund, a następnie zatrzymywał się.

13.3.1. UDOSKONALANIE I DALSZE PRACE NAD PROTOTYPEM

Po zakończeniu wstępnych testów dalsze ulepszenia prototypu polegało na zamianie Arduino na ESP 32 oraz połączenie tego z sterownikiem, dalszy rozwój prototypu obejmuje również przeprowadzenie długoterminowych testów użytkowania, które dostarczą cennych informacji zwrotnych. Dzięki temu było możliwe, zaprojektowanie finalnej wersji urządzenia, stanowiącej interaktywną i łatwą w obsłudze zabawkę dla zwierząt domowych

13.3.2. OPRACOWANIE KODU DO ESP 32.

Działanie kodu na ESP 32:

Co 5 sekund jest wysyłane żądanie (HTTP GET) przy użyciu obiektu `HTTPClient`, następnie kod odczytuje i przechowuje odpowiedź w zmiennej `payload`. Adresem końcowym w tym wypadku jest <http://74.63.9.24/toy/status>.

Program korzysta z układu L298N; piny są zdefiniowane jako `IN1 = 12`, `IN2 = 14` oraz `ENA = 13`.

W kodzie znajdują się dwie funkcje:

- `startMotor()` – ustawia `IN1` na stan wysoki, `IN2` na niski i `ENA` na 255 (pełna prędkość),
- `stopMotor()` – ustawia `IN1` i `IN2` na niski oraz `ENA` na 0 (silnik zatrzymany).

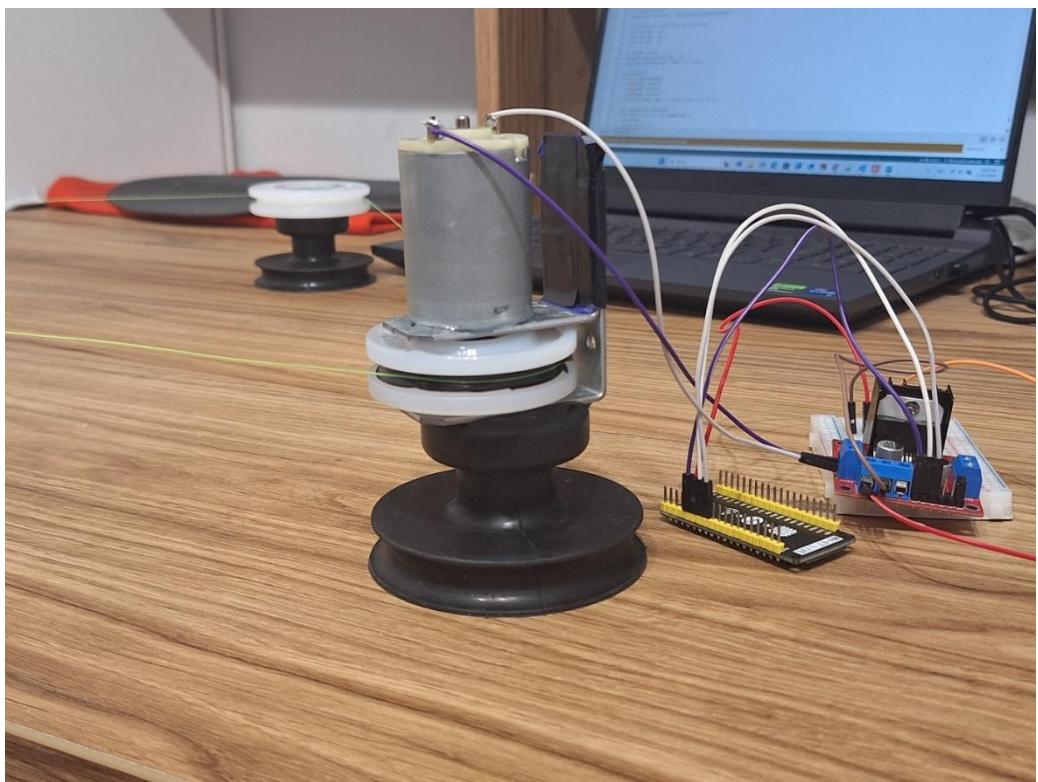
Gdy `runToy == 1`, następuje uruchomienie silnika na 15 sekund, co widać na komunikacie w **Tab. 13.1**, po czym silnik zostaje zatrzymany.

Tab. 13.1. Komunikat o uruchomieniu silnika zabawki. [opracowanie własne]

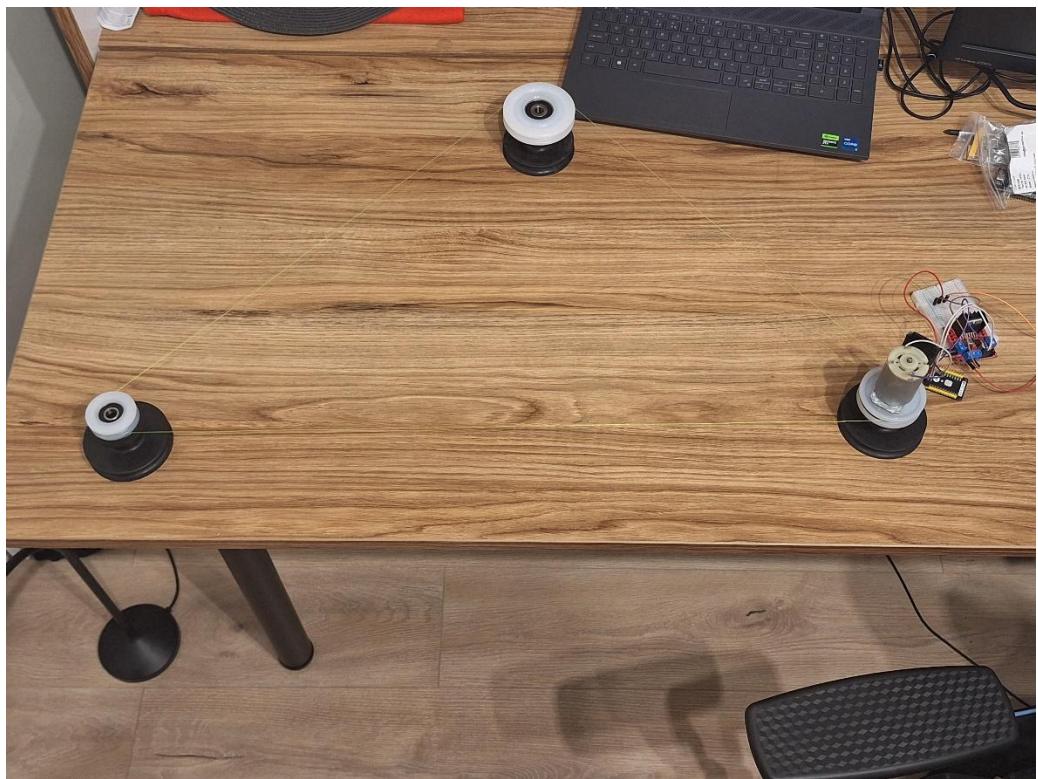
```
Otrzymano JSON: {"runToy":1}
runToy: 1
Silnik uruchomiony!
```

13.4. OSTATECZNY MODEL URZĄDZENIA

Przedstawiona na **Rys. 13.1** i **Rys. 13.2** ostateczna wersja interaktywnej zabawki dla zwierząt domowych stanowi rozwinięcie wszystkich założeń i rozwiązań testowanych podczas etapu prototypowania.



Rys. 13.1. Ostateczny model zabawki. [opracowanie własne]



Rys. 13.2. Widok z góry ostatecznego modelu zabawki. [opracowanie własne]

13.4.1. SCHEMAT POŁĄCZEŃ ELEKTRYCZNYCH

Połączenie IO12-ENA kontroluje prędkość silnika za pomocą sygnału PWM, ważne w tym połączeniu jest to, aby cały układ był podłączony do tego samego GND.

ESP32	L298N	Silnik	Zasilacz
IO12	ENA	-	-
IO13	IN1	-	-
IO14	IN2	-	-
-	OUT1	+Silnika	-
-	OUT2	-Silnika	-
-	VMS	-	+12V/24V
GND	GND	-	GND
5V	VCC	-	-

13.4.2. BEZPIECZEŃSTWO I UWAGI

Zasilanie:

- Silnik DC 540 może pracować między 12V a 24V, ale lepiej sprawdzić, czy nie przegrzewa się po 2–3 minutach.

L298N:

- Jeśli silnik pobiera powyżej 2A to i tak jest już zamontowany radiator, ale lepiej sprawdzić czy urządzenie się nie przegrzewa

Mechanika:

- Bloczki muszą mieć rowki zapobiegające ześlizgiwaniu się linki.
- Sprawdź napięcie linki – zbyt duże może przeciążyć silnik, za to zbyt małe może sprawić, iż silniczek nie będzie obracał linki z wystarczająco dużą prędkością.

13.4.3. INSTRUKCJA UŻYTKOWNIKA

Przeznaczenie i funkcje:

- Interaktywna zabawka została zaprojektowana, aby zapewnić Twojemu pupilowi rozrywkę i aktywność fizyczną podczas Twojej nieobecności. Głównym elementem zabawki jest linka poruszająca się po bloczkach, do której możesz zamocować coś co przyciągnie uwagę zwierzęcia, jak np. kolorowa wstążka. Urządzenie jest sterowane zdalnie za pomocą aplikacji mobilnej PetApp.

Elementy urządzenia:

- Moduł główny: Zawiera silnik DC, sterownik L298N, mikrokontroler ESP32, układ zasilania oraz mechanizm z bloczkami.
- Linka: Porusza się po bloczkach.

Przygotowanie do użycia:

- Sprawdzenie urządzenia: Przed pierwszym użyciem upewnij się, że wszystkie elementy urządzenia są poprawnie zamocowane.
- Podłączenie do zasilania: Podłącz zasilacz 12V/1A lub 24V/1,5A do urządzenia.
- Synchronizacja z aplikacją: Upewnij się, że aplikacja PetApp jest zainstalowana na Twoim smartfonie.

Uruchomienie urządzenia:

- Uruchomienie zabawki: Otwórz aplikację PetApp na smartfonie.
- Sterowanie: W aplikacji naciśnij przycisk „Turn On Toy”, aby włączyć linkę i rozpocząć zabawę. Aby wyłączyć zabawkę, naciśnij przycisk „Turn Off Toy”.

Podczas użytkowania:

- Obserwuj zwierzę: Nadzoruj pierwsze interakcje zwierzęcia z zabawką, aby upewnić się, że zabawa przebiega bezpiecznie.
- Sprawdzaj napięcie linki: Upewnij się, że linka nie jest zbyt mocno ani zbyt słabo napięta.
- Zapewnij bezpieczeństwo: Upewnij się, że zabawka znajduje się na stabilnym podłożu i nie ma możliwości, aby zwierzę ją przewróciło lub uszkodziło.

Konserwacja i pielęgnacja:

- Przegląd: Okresowo sprawdzaj stan linki, bloczków i połączeń elektrycznych, aby upewnić się, że nie są uszkodzone.
- Przechowywanie: Gdy urządzenie nie jest używane, przechowuj je w suchym i bezpiecznym miejscu.

14. APLIKACJA MOBILNA

Aplikacja mobilna PetApp została stworzona z myślą o właścicielach zwierząt domowych, którzy chcą monitorować aktywność swoich pupili. Dzięki połączeniu z inteligentną obrożą, aplikacja umożliwia śledzenie dziennego dystansu przebytego przez zwierzę, czasu spoczynku oraz ostatniego wydanego dźwięku. System ten wspiera właścicieli w lepszym zrozumieniu zachowania swoich podopiecznych i może pomóc w wykrywaniu niepokojących zmian w ich aktywności. Link do pliku apk oraz pliku dart: <https://gitlab-stud.elka.pw.edu.pl/kkoniecz/pbl1-petapp>.

14.1. OKREŚLENIE WYMAGAŃ I FUNKCJONALNOŚCI APLIKACJI

Aplikacja mobilna PetApp została zaprojektowana w celu monitorowania aktywności zwierząt domowych, takich jak psy i koty. Jej głównym zadaniem jest dostarczanie użytkownikowi informacji o dziennym dystansie przebytym przez zwierzę, czasie spoczynku oraz ostatnio wydanym dźwięku przez zwierzę. Kluczowe wymagania aplikacji obejmują:

- połączenie z obrożą wyposażoną w akcelerometr, żyroskop i mikrofon,
- pobieranie i przetwarzanie danych w czasie rzeczywistym,
- prezentowanie informacji w przejrzysty i intuicyjny sposób.

14.2. WYBÓR PLATFORMY

Do realizacji projektu wybrano technologię Flutter, co pozwala na jednocienne tworzenie aplikacji dla systemów Android i iOS. Wybór ten został dokonany ze względu na:

- Wydajność i szybkość działania aplikacji,
- Łatwość integracji z backendem oraz obsługę czujników urządzenia,
- Jednolity kod źródłowy, co ułatwia dalszy rozwój i utrzymanie aplikacji.

Mimo że Flutter umożliwia tworzenie aplikacji na wiele platform, zdecydowaliśmy się na wersję tylko na Androida. Wynikało to głównie z dostępności urządzeń testowych. Skupienie się na jednym systemie pozwoliło nam szybciej dopracować aplikację i przeprowadzić więcej testów.

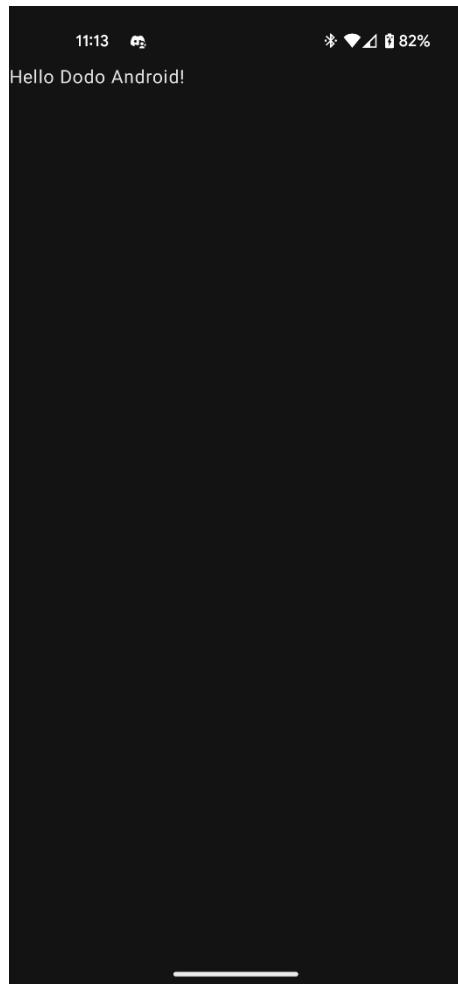
14.3. TWORZENIE INTERFEJSU

Tworzenie interfejsu w naszej aplikacji było kluczowym etapem, który miał na celu zapewnienie użytkownikom łatwego dostępu do funkcji i przyjemności z korzystania z aplikacji. Skupiliśmy się na intuicyjności, estetyce oraz funkcjonalności, dbając o to, by wszystkie elementy były łatwe w obsłudze. Projektując interfejs, uwzględniliśmy także specyficzne potrzeby użytkowników, o których napisaliśmy w dalszej części.

14.3.1. PIERWSZE WŁĄCZENIE APLIKACJI W TRYBIE DEBUGOWANIA

Pierwsze włączenie strony testowej naszej aplikacji, stworzonej w Flutterze, na telefonie to moment, w którym użytkownik widzi komunikat "Hello Dodo Android", co widać na ekranie

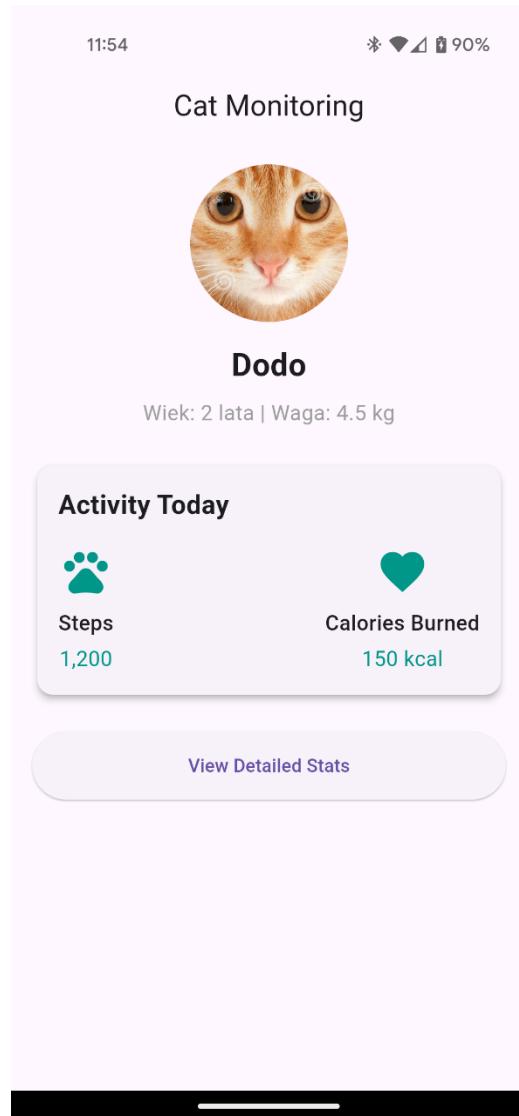
na Rys. 14.1. Jest to prosty, ale ważny etap, który potwierdza, że aplikacja została poprawnie zbudowana i uruchomiona na urządzeniu. Taki test pozwala również upewnić się, że połaczenie z Flutterem działa poprawnie, a strona jest wyświetlana zgodnie z oczekiwaniami na platformie Android. To pierwszy krok w dalszym rozwoju aplikacji, w którym możemy przejść do implementacji kolejnych funkcji.



Rys. 14.1 Pierwsze włączenie aplikacji. [opracowanie własne]

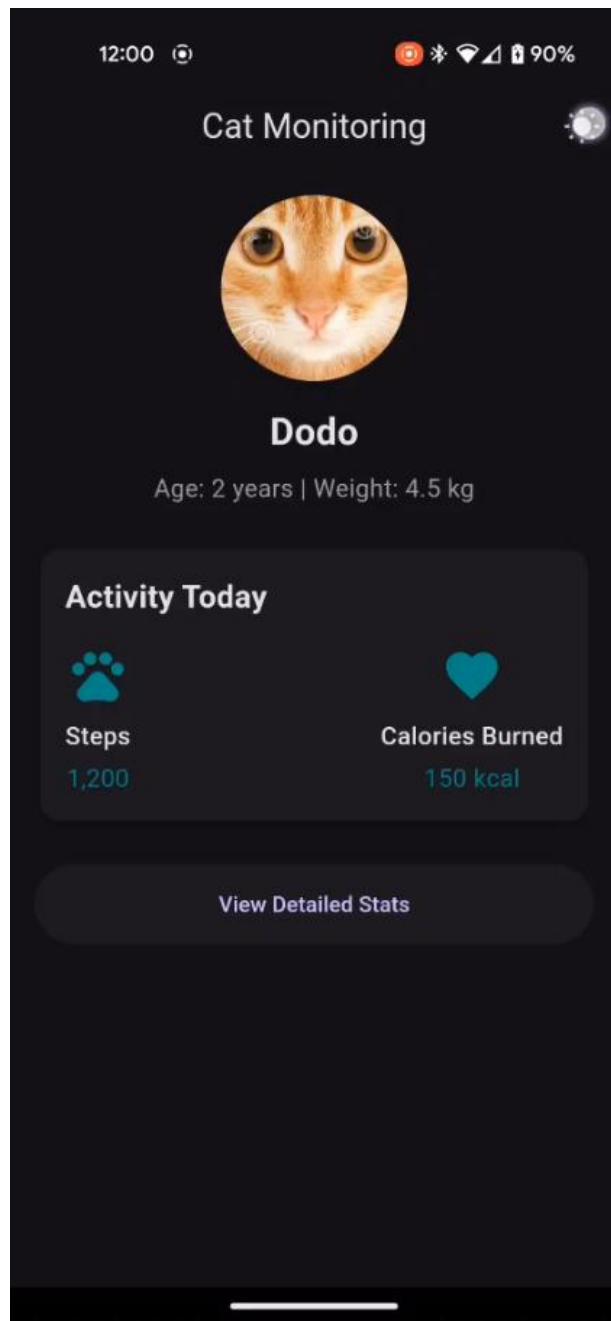
14.3.2. DOPRACOWYWANIE WYGLĄDU

Następnie przeszliśmy do tworzenia interfejsu (Rys. 14.2) i zakładek, które chcieliśmy mieć w naszej aplikacji. Początkowo były to dane zwierzęcia oraz względnie losowe zakładki, które stworzone zostały na razie w celu zagospodarowania miejsca na ekranie na poszczególne zakładki. Dodaliśmy także zdjęcie kota (początkowo mieliśmy robić projekt tylko dla kotów, jednakże potem zdecydowaliśmy się o rozszerzenie projektu także dla psów). Póki co dane są wpisane bezpośrednio w kodzie i nie mają odzwierciedlenia w rzeczywistości.



Rys. 14.2. Pierwszy interfejs aplikacji. [opracowanie własne]

Kolejnym krokiem było dodanie możliwości zmiany motywów – z jasnego na ciemny oraz odwrotnie, co jest widoczne na Rys. 14.3.

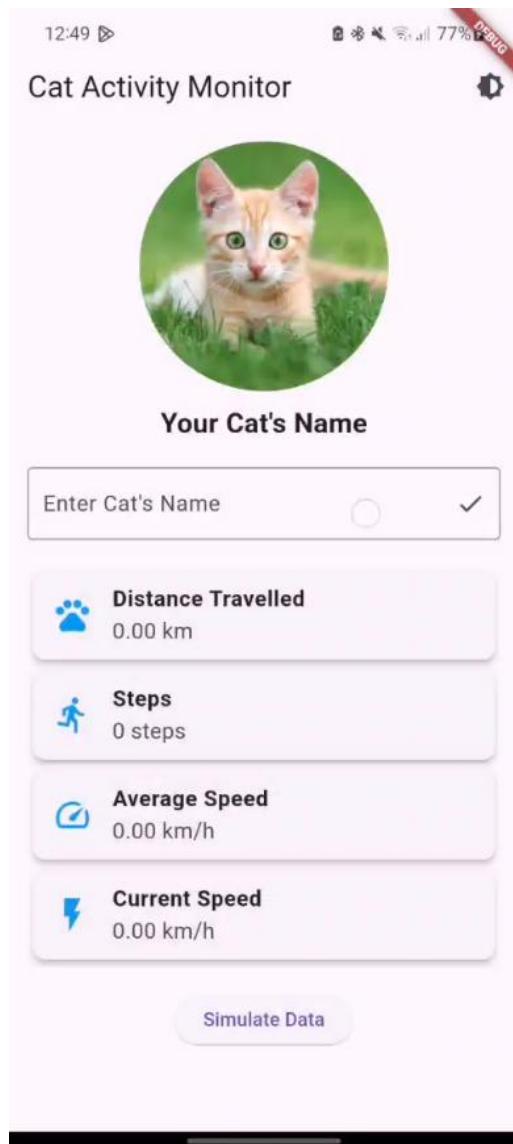


Rys. 14.3. Dodanie opcji zmiany motywu jasny/ciemny. [opracowanie własne]

W kolejnym etapie przedstawionym na Rys. 14.4. prac nad aplikacją skupiliśmy się na udoskonaleniu interfejsu oraz zakładek, które miałyby znaleźć się w finalnej wersji systemu. Początkowo interfejs zawierał podstawowe dane zwierzęcia i testowe sekcje, które miały na celu odpowiednie rozmieszczenie informacji na ekranie. Wraz z rozwojem projektu, zaczęliśmy dopracowywać szczegóły wizualne oraz układ poszczególnych elementów.

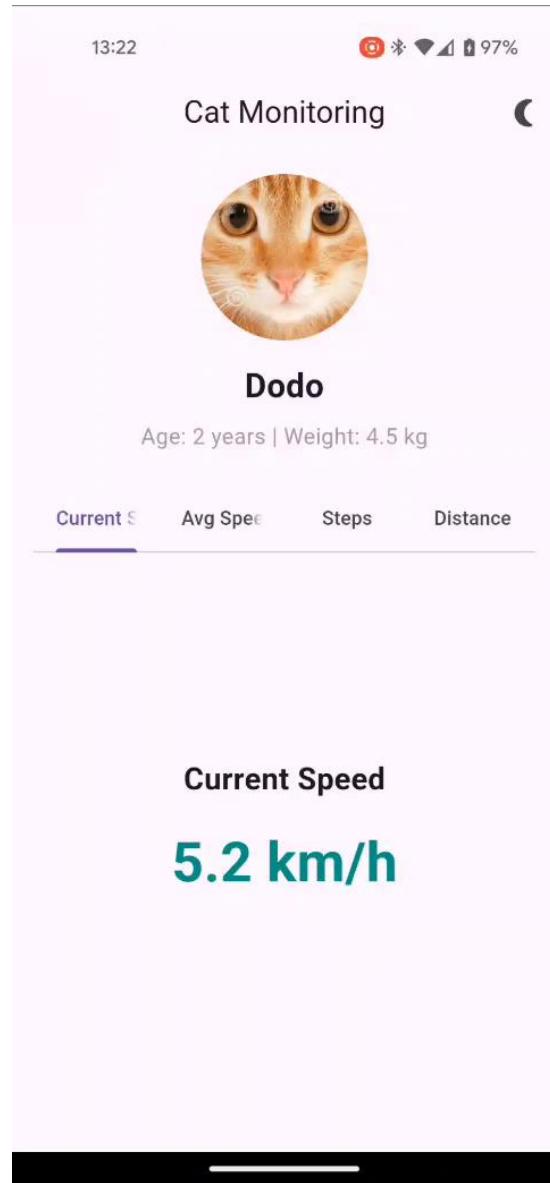
W nowej wersji interfejsu zmieniliśmy nieco układ informacji i styl prezentacji. Poprawiliśmy sposób wprowadzania imienia zwierzęcia, dodając dedykowane pole tekstowe, oraz zoptymalizowaliśmy wygląd kart z danymi dotyczącymi aktywności. Teraz użytkownik może w łatwy sposób zobaczyć m.in. dystans przebytej trasy.

Dodatkowo, wprowadziliśmy **przycisk do symulacji danych**, który pozwala na testowanie funkcjonalności aplikacji bez potrzeby realnego połączenia z urządzeniem. Dzięki temu mogliśmy zweryfikować poprawność działania aplikacji oraz ocenić, jak prezentują się dynamiczne zmiany wartości w interfejsie na kliknięcie przycisku. Aktualnie dane w aplikacji są jeszcze wprowadzane testowo i nie mają pełnego odzwierciedlenia w rzeczywistych pomiarach.



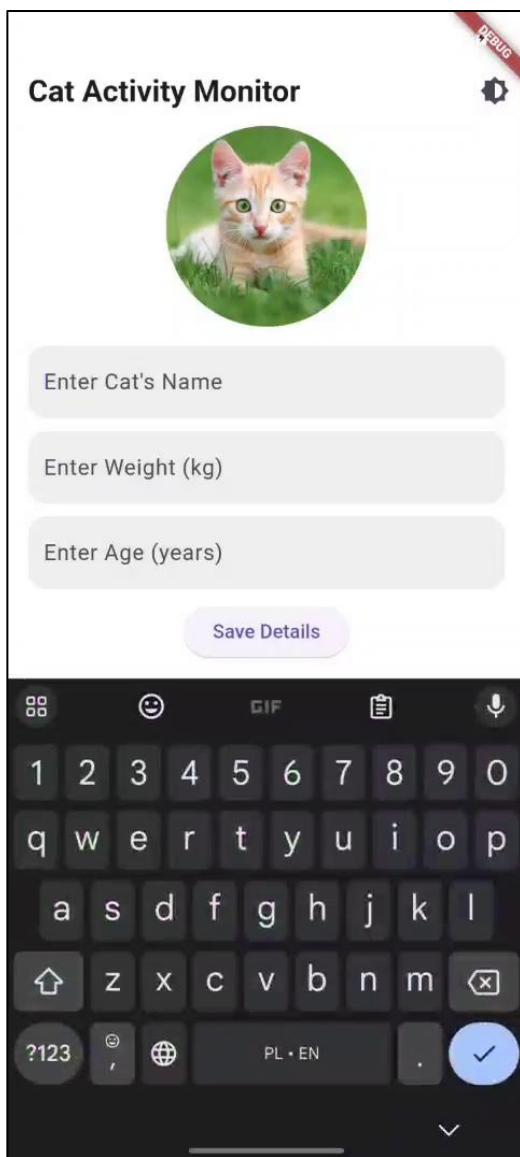
Rys. 14.4. Zmiana zakładek, dodanie przycisku symulacji danych oraz możliwość wprowadzenia imienia zwierzęcia.
[opracowane własne]

Testowaliśmy także różne inne interfejsy na ekranie poniżej na Rys. 14.5, żeby móc zadecydować który interfejs finalnie nam się najbardziej podoba i będzie spełniał nasze oczekiwania. Ten interfejs jednakże był gorszy, ponieważ po włączeniu aplikacji widoczna jest tylko jedna wartość i trzeba przechodzić do innych zakładek, żeby zobaczyć inne wartości.

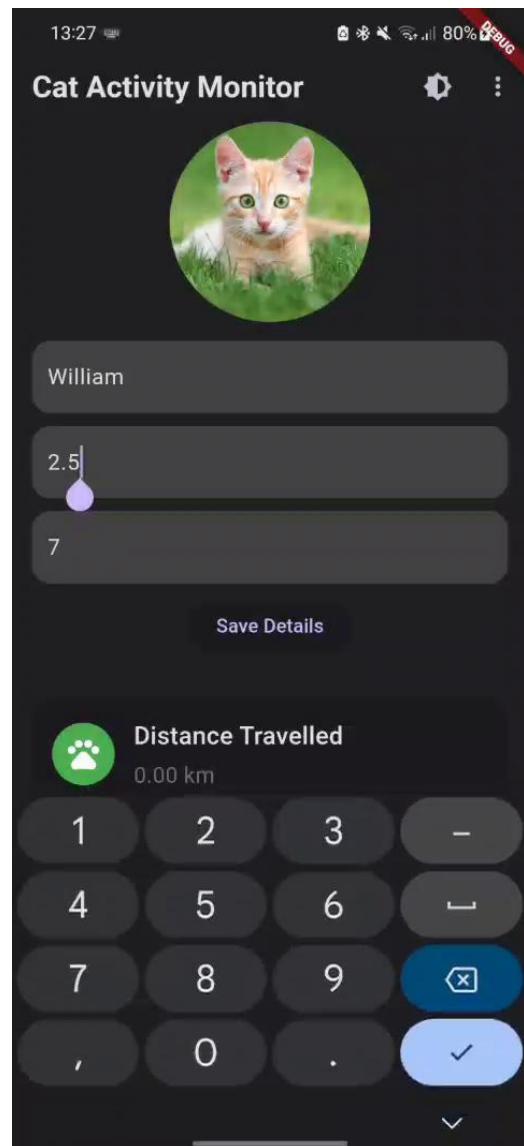


Rys. 14.5. Interfejs testowy. [opracowanie własne]

Teraz przy pierwszym włączaniu (**Rys. 14.6**) aplikacja pyta użytkownika o imię kota, jego wagę oraz wiek. Dodaliśmy także opcję zmieniania tych (dodanie zakładki do edycji w prawym górnym rogu) wartości po zapisaniu ich przy pierwszym włączaniu, czego test jest widoczny na **Rys. 14.7**. Dodatkowo, przy zmianie wartości liczbowych, czyli wagi i wieku, w telefonie uruchamia się od razu tryb klawiatury numerycznej.



Rys. 144.6. Interfejs przy włączaniu aplikacji.
[opracowanie własne]



Rys. 144.7. Możliwość edycji danych. [opracowanie własne]

14.3.3. FINALNA WERSJA INTERFEJSU PetApp

Ostateczna wersja aplikacji przedstawiona na Rys. 14.8 zawiera wszystkie docelowe zakładki, które będą realnie używane. Wdrożono moduł analizy aktywności, umożliwiający śledzenie przebytego dystansu, czasu spoczynku i ostatnich dźwięków.

Deklaracja początkowych wartości danych w naszych zakładkach została przedstawiona w Tab. 14.1.

Tab. 14.1. Deklaracja wartości w zakładkach. [opracowanie własne]

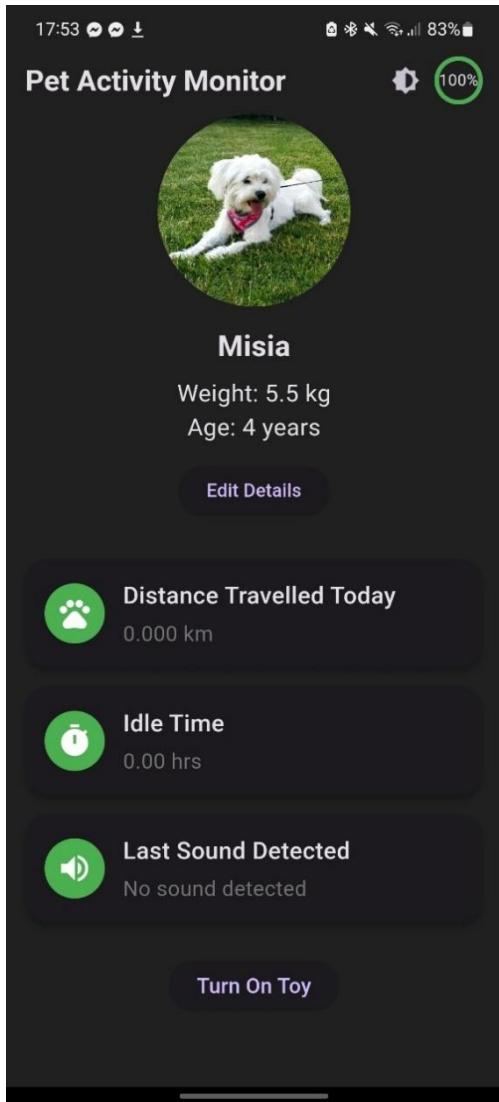
```
class _CatHomePageState extends State<CatHomePage> {
    String? _catName;
    String? _lastSoundTimestamp;
    double? _catWeight;
    int? _catAge;
    double _distanceTravelled = 0.0;
    double _idleTime = 0.0;
    bool _isToyOn = false;
    double _batteryLevel = 100.0;
```

Przycisk symulacji danych został zastąpiony funkcją zdalnego sterowania zabawką („Turn On Toy”), co pozwala użytkownikowi zablokować bądź odblokować włączanie się zabawki.

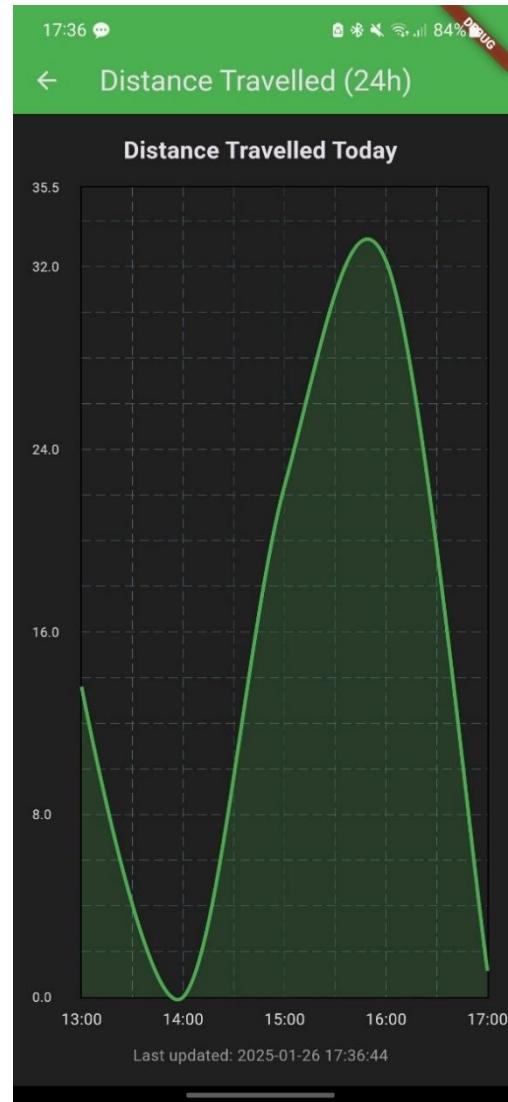
Dodano także **wykres dobowy dystansu (Rys. 14.9)**, który pozwala użytkownikowi monitorować aktywność zwierzęcia w ciągu dnia. Dodatkowo, wprowadzono **wskaźnik poziomu baterii**, informujący o stanie naładowania urządzenia.

Użytkownik może także ustawić **własne zdjęcie swojego zwierzęcia**, co pozwala na lepszą personalizację aplikacji.

Aplikacja jest w pełni funkcjonalna i gotowa do rzeczywistego użytku, oferując przejrzysty interfejs oraz dokładne dane o aktywności zwierzęcia.



Rys. 14.8. Finalny interfejs aplikacji. [opracowanie własne]



Rys. 14.9. Wykres w aplikacji [opracowanie własne]

14.3.4. IKONKA APLIKACJI PetApp

Ikonka aplikacji PetApp przedstawiona na Rys. 14.10 została wygenerowana za pomocą sztucznej inteligencji, co pozwoliło na uzyskanie unikalnego i nowoczesnego wyglądu. Przedstawia ona sylwetkę kota na błękitnym tle, co nawiązuje do głównego celu aplikacji – monitorowania aktywności zwierząt domowych, takich jak koty i psy.



Rys. 14.10. Logo PetApp. [wygenerowana przy użyciu modelu ChatGPT]

Ta ikona jest nie tylko estetyczna, ale również intuicyjna – błękitne tło symbolizuje spokój i opiekę, a postać kota wskazuje na głównych użytkowników aplikacji, czyli właścicieli zwierząt domowych.

14.4. INTEGRACJA Z BACKENDEM

Kluczowym aspektem działania aplikacji jest integracja z backendem, który odpowiada za gromadzenie, analizowanie i udostępnianie danych o aktywności zwierzęcia. Backend udostępnia szereg endpointów API, które umożliwiają komunikację między aplikacją a serwerem. W procesie integracji wykorzystaliśmy głównie metody **GET** oraz **POST** do zarządzania danymi.

Metody GET zostały użyte do pobierania informacji o aktywności zwierzęcia, takich jak przebyta odległość, czas odpoczynku czy analiza ruchu. Dzięki nim użytkownik może śledzić statystyki i historię aktywności swojego pupila.

Metoda POST pozwala na przesyłanie danych do backendu dotyczących włączenia bądź zablokowania uruchamiania zabawki.

14.4.1. FUNKCJE DO ŁĄCZENIA Z API

Funkcje w **Tab. 14.2** pełnią kluczową rolę w zapewnieniu, że aplikacja będzie zawsze wyświetlała aktualne informacje o zwierzęciu. Każda z nich wykonuje zapytanie API do odpowiednich endpointów na serwerze, aby pobrać konkretne dane, takie jak poziom baterii, ostatni dźwięk wydany przez zwierzę, dzienny przebyty dystans, status zabawki, czy czas bezczynności.

Aby zapewnić, że te dane są regularnie aktualizowane, funkcje są wywoływanie cyklicznie co 5 sekund za pomocą timera. Taki mechanizm odświeżania pozwala na płynne i dynamiczne wyświetlanie danych, bez konieczności ręcznego odświeżania aplikacji przez użytkownika.

Tab. 14.2. Funkcje do pobierania informacji z API. [opracowanie własne]

```
@override
void initState() {
    super.initState();
    _loadCatDetails();
    _loadImage();
    _fetchBatteryLevel();
    _fetchLastSoundTimestamp();
    _fetchDailyDistance();
    _fetchToyStatus();
    _fetchIdleTime();
    // Ustawienie timeru do odświeżania danych co 5 sekund
    _dataRefreshTimer = Timer.periodic(Duration(seconds: 5), (timer) {
        _fetchBatteryLevel();
        _fetchLastSoundTimestamp();
        _fetchDailyDistance();
        _fetchIdleTime();
        _fetchToyStatus();
    });
}
```

W Tab. 14.3 poniżej, przedstawiamy przykładową funkcję do pobierania danych z endpointów - `_fetchBatteryLevel`, która wykonuje zapytanie HTTP typu GET do API, aby pobrać dane o poziomie baterii z serwera. Jeśli odpowiedź z serwera ma kod statusu 200, funkcja dekoduje dane JSON i aktualizuje zmienną `_batteryLevel` w stanie aplikacji, aby wyświetlić poziom baterii.

Tab. 14.3. Przykładowa funkcja do wysyłania zapytania typu GET. [opracowanie własne]

```
Future<void> _fetchBatteryLevel() async {
    const String apiUrl = 'http://74.63.9.24/battery';

    try {
        final response = await http.get(Uri.parse(apiUrl));

        if (response.statusCode == 200) {
            final data = json.decode(response.body);
            print('Battery level from API: ${data['current_level']}');

            setState(() {
                _batteryLevel = data['current_level'] ?? _batteryLevel;
            });
        } else {
            print('Błąd podczas zapytania GET. Kod statusu: ${response.statusCode}');
        }
    } catch (e) {
        print('Błąd podczas połączenia z API: $e');
    }
}
```

W przypadku błędu zapytania lub problemu z połączeniem, funkcja wypisuje odpowiednie komunikaty o błędach lub o rzeczywistym stanie baterii w konsoli (**Tab. 14.4**), co ułatwiło nam szybkie odczytywanie danych i możliwość szybkiego testowania, czy wszystko zadziałało.

Tab. 14.4. Komunikaty o błędach lub o sukcesie. [opracowanie własne]

```
I/flutter (21688): Daily distance from API: 13.6
I/flutter (21688): Battery level from API: 95.0
I/flutter (21688): Toy state updated: ON
D/ProfileInstaller(21688): Installing profile for com.example.cat_app
I/flutter (21688): Daily distance from API: 13.6
I/flutter (21688): Battery level from API: 95.0
I/flutter (21688): Toy state updated: OFF
I/flutter (21688): Daily distance from API: 13.6
I/flutter (21688): Battery level from API: 95.0
I/flutter (21688): Battery level from API: 95.0
I/flutter (21688): Daily distance from API: 13.6
```

Pozostałe funkcje wysyłające zapytania typu GET i pobierające dane z endpointów zostały stworzone analogicznie.

14.4.2. FUNKCJE DO ZABAWKI Z POZIOMU APLIKACJI

Po włączeniu aplikacji funkcja w **Tab. 14.5** wysyłająca zapytanie GET pobiera dane o aktualnym stanie zabawki i dostosowuje napis na przycisku (Turn Off Toy / Turn On Toy) w zależności od aktualnego stanu zabawki. Ułatwia nam to synchronizację statusu zabawki w przypadku, gdy kilka osób ma aplikację do tego samego zwierzęcia.

Tab. 14.5. Funkcja pobierająca dane o stanie zabawki. [opracowanie własne]

```
Future<void> _fetchToyStatus() async {
  const String apiUrl = 'http://74.63.9.24/toy/status';

  try {
    final response = await http.get(Uri.parse(apiUrl));

    if (response.statusCode == 200) {
      final data = json.decode(response.body);
      print('Toy status from API: ${data['runToy']}');
      // Sprawdzamy stan zabawki
      setState(() {
        _isToyOn = data['runToy'] == 1; // Jeśli runToy to 1, zabawka jest włączona
      });
      print('Toy status: ${_isToyOn ? "ON" : "OFF"}');
    } else {
      print('Błąd podczas zapytania GET. Kod statusu: ${response.statusCode}');
    }
  } catch (e) {
    print('Błąd podczas połączenia z API: $e');
  }
}
```

Funkcja w **Tab. 14.6** `_toggleToy()` wysyła zapytanie HTTP typu POST do API, aby przełączyć stan zabawki, wysyłając w ciele zapytania informację o nowym stanie (włączonym lub wyłączonym). Zmienna `_isToyOn` kontroluje, czy zabawka jest włączona, i jej stan jest aktualizowany po pomyślnym otrzymaniu odpowiedzi o statusie 200. Jeśli zapytanie zakończy się błędem, w konsoli wyświetli się odpowiedni komunikat z kodem statusu lub błędem połączenia. Funkcja ta umożliwia dynamiczne sterowanie stanem zabawki z poziomu aplikacji.

Tab. 14.6. Przykładowa funkcja wysyłająca zapytania typu POST. [opracowanie własne]

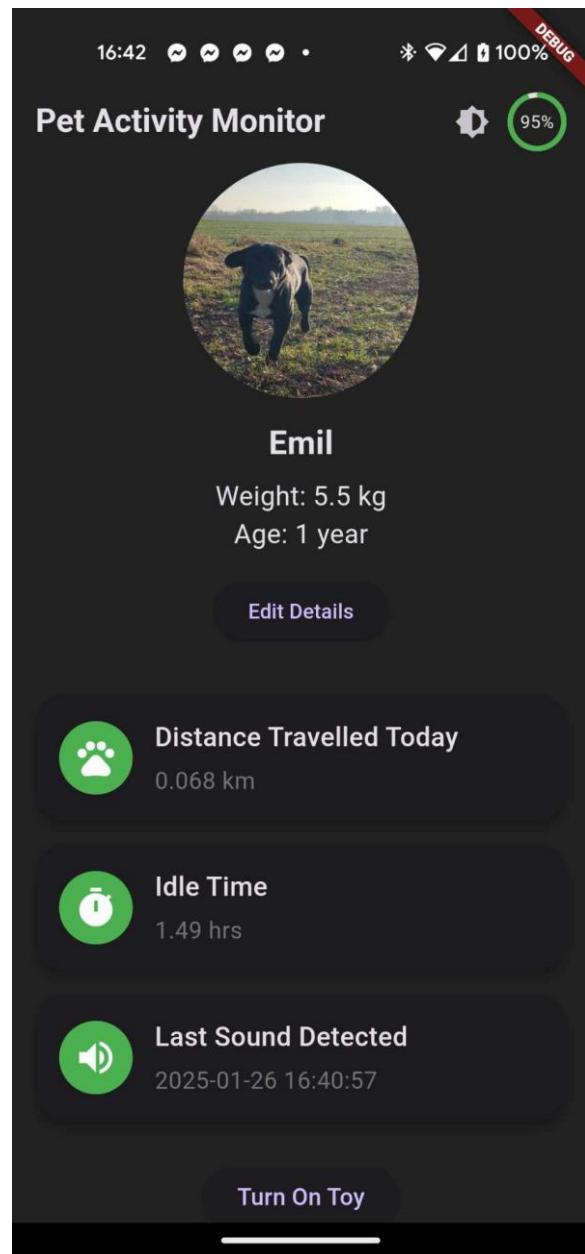
```
Future<void> _toggleToy() async {
  const String apiUrl = 'http://74.63.9.24/toy/toggle';

  try {
    final response = await http.post(
      Uri.parse(apiUrl),
      body: json.encode({'state': _isToyOn ? '0' : '1'}),
      headers: {
        'Content-Type': 'application/json',
      },
    );

    if (response.statusCode == 200) {
      setState(() {
        _isToyOn = !_isToyOn; // Przełączanie stanu zabawki
      });
      print('Toy state updated: ${_isToyOn ? "ON" : "OFF"}');
    } else {
      print('Błąd podczas zapytania POST. Kod statusu: ${response.statusCode}');
    }
  } catch (e) {
    print('Błąd podczas połączenia z API: $e');
  }
}
```

14.5. TESTOWANIE APLIKACJI

Widoczny interfejs przedstawia finalną wersję aplikacji PetApp, testowaną z rzeczywistymi danymi pobranymi z obroży podczas jej ruszania. Testowanie potwierdza poprawność zbierania informacji o aktywności zwierzęcia, poziomie baterii i wykrytych dźwiękach. Funkcja zdalnego sterowania zabawką działa zgodnie z założeniami, zapewniając pełną kontrolę nad urządzeniem.



Rys. 14.11. Widoczne dane w aplikacji ściągnięte z serwera. [opracowanie własne]

15. TEST

Złożyliśmy nasze urządzenie w całość – obrożę wyposażoną w czujniki ruchu oraz mikrofon, które połączyciśmy z aplikacją mobilną PetApp. Dzięki temu możemy monitorować aktywność zwierzęcia oraz analizować jego zachowanie w czasie rzeczywistym. Aplikacja pozwala również na zdalne włączanie i wyłączanie zabawki, co umożliwia interakcję ze zwierzęciem podczas nieobecności właściciela.

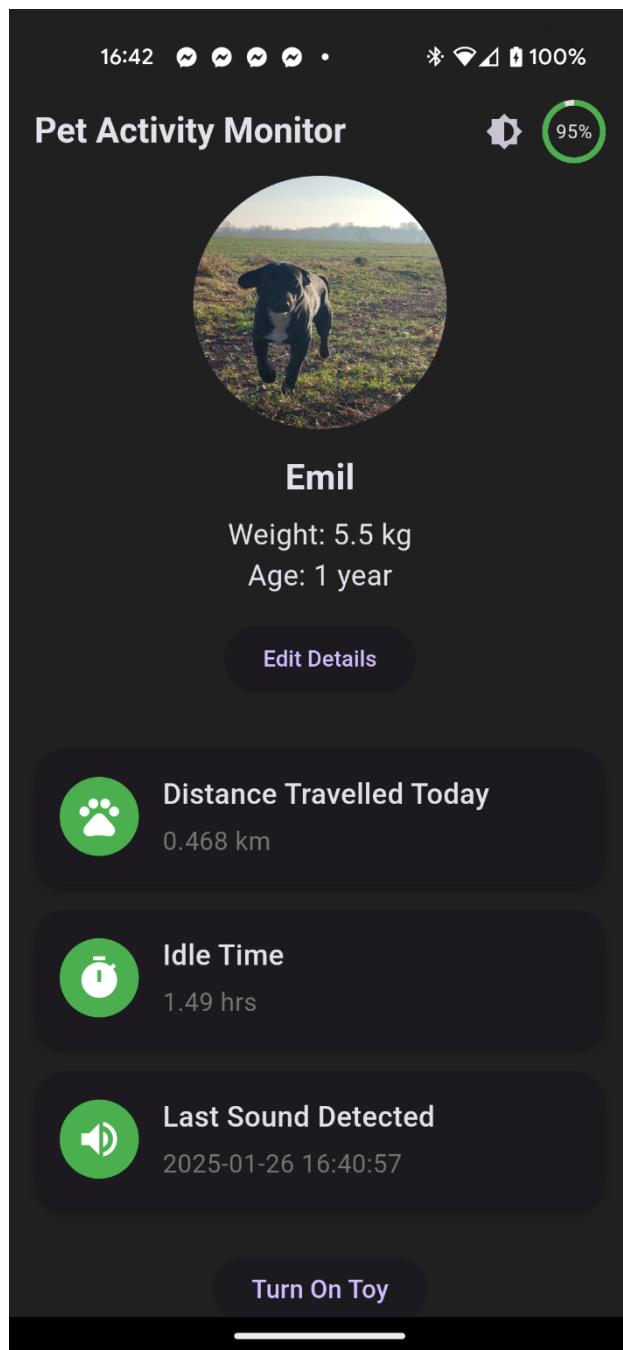
15.1. ZAŁOŻENIE OBROŻY NA PSA

Po założeniu obroży na psa, dane, które zostały przedstawione w **Tab. 14.1** zostały wysłane do bazy danych. Widać zarejestrowane szczenięcia psa, przebyty dystans przez psa oraz resztę danych.

Tab. 15.1. Baza danych. [opracowanie własne]

	id	timestamp	movement	sound	sound_ty...	battery_p...	travel_dis...
1	1	2025-01-26 12:19:14	1	0	NULL	95	0
2	2	2025-01-26 12:19:19	1	0	NULL	95	0
3	3	2025-01-26 12:19:24	1	0	NULL	95	0
4	4	2025-01-26 12:19:29	1	0	NULL	95	0
5	5	2025-01-26 12:19:34	1	0	NULL	95	0
6	6	2025-01-26 12:19:39	1	0	NULL	95	0
7	7	2025-01-26 12:19:44	1	0	NULL	95	0
8	8	2025-01-26 12:19:49	1	0	NULL	95	0
9	9	2025-01-26 12:19:54	1	0	NULL	95	0
10	10	2025-01-26 12:19:59	1	0	NULL	95	0
11	11	2025-01-26 12:20:04	1	0	NULL	95	0
12	12	2025-01-26 12:26:53	1	0	NULL	95	1.05
13	13	2025-01-26 12:26:58	1	0	NULL	95	1.05
14	14	2025-01-26 12:27:03	1	0	NULL	95	2.2
15	15	2025-01-26 12:27:08	1	0	NULL	95	1.2
16	16	2025-01-26 12:27:13	1	0	NULL	95	2.1
17	17	2025-01-26 12:27:18	1	1	dog_bark	95	2.25
18	18	2025-01-26 12:27:23	1	1	dog_bark	95	2.2
19	19	2025-01-26 12:27:28	1	0	NULL	95	1.55
20	20	2025-01-26 13:27:53	0	1	dog_bark	95	0
21	21	2025-01-26 14:03:09	1	0	NULL	95	0.75
22	22	2025-01-26 14:03:14	1	1	dog_bark	95	2.05
23	23	2025-01-26 14:03:19	1	0	NULL	95	2
24	24	2025-01-26 14:03:23	1	0	NULL	95	2.2
25	25	2025-01-26 14:03:29	1	1	dog_bark	95	2.25
26	26	2025-01-26 14:03:33	1	0	NULL	95	1.7
27	27	2025-01-26 14:03:39	1	1	dog_bark	95	0.3
28	28	2025-01-26 14:03:44	1	0	NULL	95	0.75
29	29	2025-01-26 14:03:49	1	0	NULL	95	2.1
30	30	2025-01-26 14:03:53	1	0	NULL	95	2.15
31	31	2025-01-26 14:03:59	1	0	NULL	95	2.2
32	32	2025-01-26 14:04:03	1	0	NULL	95	0.3
33	33	2025-01-26 14:05:44	1	0	NULL	95	2.45
34	34	2025-01-26 14:05:49	1	0	NULL	95	1.2
35	35	2025-01-26 14:44:19	0	1	dog_bark	95	0
36	36	2025-01-26 14:45:37	0	1	dog_bark	95	0
37	37	2025-01-26 14:47:57	0	1	dog_bark	95	0
38	38	2025-01-26 15:18:24	1	0	NULL	95	0.95
39	39	2025-01-26 15:18:28	1	0	NULL	95	2
40	40	2025-01-26 15:18:34	1	0	NULL	95	2.4

W aplikacji pojawiły się dane z określonego czasu, w którym zabawka działała, co zostało przedstawione na **Rys. 15.1**.



Rys. 15.1. Testy w ciągu dnia – widok w aplikacji. [opracowanie własne]

Po wcisnięciu przycisku „Turn On Toy” oraz zastaniu sytuacji, w której pies szczała i zaczynał biegać – zabawka uruchomiła się i dzięki temu pies mógł zająć się zabawką i uspokoił się.

15.2. WNIOSKI

Przeprowadzone testy wykazały, że wszystkie elementy systemu działają zgodnie z założeniami. Obroża poprawnie zbiera dane z czujników ruchu oraz mikrofonu, a aplikacja PetApp bezproblemowo łączy się z urządzeniem. Detekcja aktywności zwierzęcia działa precyzyjnie, a

funkcja zdalnego sterowania zabawką reaguje bez większych opóźnień. Testy potwierdziły, że system skutecznie wspomaga monitorowanie zwierzęcia i minimalizuje skutki lęku separacyjnego.

16. BIBLIOGRAFIA

- [1] <https://www.futuremarketinsights.com/reports/pet-market> [29.01.2025]
- [2] <https://www.bloomberg.com/company/press/global-pet-industry-to-grow-to-500-billion-by-2030-bloomberg-intelligence-finds/> [29.01.2025]
- [3] <https://www.mordorintelligence.com/industry-reports/united-states-pet-market> [29.01.2025]
- [4] <https://www.arizton.com/market-reports/pet-care-market> [29.01.2025]
- [5] <https://www.grandviewresearch.com/industry-analysis/pet-care-market> [29.01.2025]
- [6] <https://www.rocklinranchvet.com/blog/top-10-traits-of-a-responsible-pet-owner/> [29.01.2025]
- [7] <https://www.earth.com/news/personalities-animal-owners-pets/> [29.01.2025]
- [8] <https://www.petspyjamas.com/blog/there-are-four-types-of-pet-owner-which-one-are-you/> [29.01.2025]
- [9] <https://springnaturals.com/blogs/dog-cat-nutrition-blog/your-pet-and-your-personality> [29.01.2025]
- [10] <https://www.jcu.edu.au/news/releases/2024/august/personality-may-predict-pet-ownership> [29.01.2025]
- [11] <https://www.latrobe.edu.au/news/articles/2019/opinion/the-personalities-of-pets-and-owners> [29.01.2025]
- [12] <https://en.wikipedia.org/wiki/Pet> [29.01.2025]
- [13] <https://www.enterpriseappstoday.com/stats/pet-ownership-statistics.html> [29.01.2025]
- [14] <https://www.petfoodindustry.com/news-newsletters/pet-food-news/article/15462494/infographic-most-of-world-owns-pets-dogs-are-tops> [29.01.2025]
- [15] <https://www.avma.org/resources-tools/reports-statistics/us-pet-ownership-statistics> [29.01.2025]
- [16] <https://www.thezebra.com/resources/research/pet-ownership-statistics/> [29.01.2025]
- [17] <https://johndog.pl/blog/kot/kocie-problemy-zdrowotne-najczestsze-choroby/> [29.01.2025]
- [18] <https://www.sciencedirect.com/science/article/abs/pii/S0168159196011343> [27.11.2024]
- [19] <https://www.msdvetmanual.com/behavior/normal-social-behavior-and-behavioral-problems-of-domestic-animals/behavioral-problems-of-cats> [27.11.2024]

- [20] <https://www.vet.cornell.edu/departments-centers-and-institutes/cornell-feline-health-center/health-information/feline-health-topics/feline-behavior-problems-aggression>
[27.11.2024]
- [21] <https://www.sciencedirect.com/science/article/abs/pii/S0168159123001971>
[27.11.2024]
- [22] <https://onlinelibrary.wiley.com/doi/full/10.1155/2016/6296315> [27.11.2024]
- [23] <https://pmc.ncbi.nlm.nih.gov/articles/PMC6912669/> [27.11.2024]
- [24] <https://agriculture.vic.gov.au/livestock-and-animals/animal-welfare-victoria/cats/health/pet-care-practices-research-results> [27.11.2024]
- [25] <https://pangovet.com/pet-lifestyle/cats/how-to-care-for-a-cat-while-working-full-time/>
[27.11.2024]
- [26] <https://manypets.com/uk/articles/separation-anxiety-in-cats/> [27.11.2024]
- [27] <https://www.rover.com/blog/cat-separation-anxiety/> [27.11.2024]
- [28] **Nowak, K. (2017).** *Psychologia zwierząt domowych*. Wydawnictwo Naukowe PWN.
- [29] **Bąk, J. (2021).** Przewodnik po kotach: zachowanie i opieka. Wydawnictwo Znak
- [30] **Ciechanowicz, J. (2019).** *Zwierzęta w naszych domach: zachowanie kotów i psów*. Wydawnictwo Galaktyka.
- [31] <https://www.psy.pl/artykuly/zdrowie/choroby-neurologiczne-u-psow> [29.01.2025]
- [32] <https://www.josera.pl/poradnik/wiedza-o-psach/zapalenie-pecherza-moczowego-u-psa.html> [29.01.2025]
- [33] https://www.perfect-fit.pl/pies-wskazowki/ruch/agresja-u-psa-jak-sobie-z-nia-radzic-co-ja-powoduje#h_47178969611261633701545814 [29.01.2025]
- [34] <https://wamiz.pl/pies/rasy#letter-A> [29.01.2025]

17. SPIS RYSUNKÓW

Rys. 1.1. Mapa skojarzeń. [opracowanie własne]	8
Rys. 1.2. Wydatki właścicieli zwierząt. [źródło: Enterprise Apps Today].....	9
Rys. 1.3. Statystyka przygarniętych zwierząt. [źródło: GfK]	9
Rys. 1.4. Statystyka posiadania min. 1 kota. [źródło: Enterprise Apps Today]	10
Rys. 4.1. Ankieta – pytanie 1. [opracowanie własne]	24
Rys. 4.2. Ankieta – pytanie 2. [opracowanie własne]	24
Rys. 4.3. Ankieta – pytanie 3. [opracowanie własne]	25
Rys. 4.4. Ankieta, pytanie 1. o IoT. [opracowanie własne]	25
Rys. 4.5. Ankieta, pytanie 2. o IoT. [opracowanie własne]	26
Rys. 4.6. Ankieta – problemy właścicieli. [opracowanie własne]	26
Rys. 4.7. Ankieta – zainteresowanie technologią. [opracowanie własne]	27
Rys. 4.8. Ankieta – klucz do spokojnego zwierzęcia. [opracowanie własne]	27
Rys. 5.1. Pies niszczący meble. [https://petsworld.com.pl/lek-separacyjny-u-psow-2/].....	28
Rys. 6.1. Persona 1 – Jakub Kowalski. [opracowanie własne]	29
Rys. 6.2. Persona 2 – Ewa Dąbrowska. [opracowanie własne].....	30
Rys. 6.3. Persona 3 – Marek Fąk. [opracowanie własne]	30
Rys. 6.4. Persona 4 – Julia Nowak. [opracowanie własne].....	31
Rys. 6.5. Persona 5 – Krzysztof Zieliński. [opracowanie własne]	31
Rys. 7.1. Mapa empatii Marka Fąka. [opracowanie własne]	32
Rys. 7.2. Mapa empatii Julii Nowak. [opracowanie własne]	32
Rys. 11.1. Case'y do obroży. [opracowanie własne].....	41
Rys. 11.2. Ostateczna wersja produktu bez wewnętrznej elektroniki.	46
Rys. 11.3. Połączenia w ostatecznej wersji urządzenia. [opracowanie własne].....	47
Rys. 11.4. Połączenia w ostatecznej wersji urządzenia. [opracowanie własne].....	48
Rys. 12.1. Wizualna reprezentacja części danych [opracowanie własne]	54
Rys. 12.2. Wizualna reprezentacja części danych. [opracowanie własne]	58
Rys. 13.1. Ostateczny model zabawki. [opracowanie własne].....	66
Rys. 13.2. Widok z góry ostatecznego modelu zabawki. [opracowanie własne]	66
Rys. 14.1 Pierwsze włączenie aplikacji. [opracowanie własne].....	70
Rys. 14.2. Pierwszy interfejs aplikacji. [opracowanie własne]	71
Rys. 14.3. Dodanie opcji zmiany motywu jasny/ciemny. [opracowanie własne].....	72
Rys. 14.4. Zmiana zakładek, dodanie przycisku symulacji danych oraz możliwość wprowadzenia imienia zwierzęcia. [opracowane własne].....	73
Rys. 14.5. Interfejs testowy. [opracowanie własne]	74
Rys. 14.6. Interfejs przy włączaniu aplikacji. [opracowanie własne].....	75
Rys. 14.7. Możliwość edycji danych. [opracowanie własne]	75
Rys. 14.8. Finalny interfejs aplikacji. [opracowanie własne]	77
Rys. 14.9. Wykres w aplikacji [opracowanie własne]	77
Rys. 14.10. Logo PetApp. [wygenerowana przy użyciu modelu ChatGPT].....	78
Rys. 14.11. Widoczne dane w aplikacji ściągnięte z serwera. [opracowanie własne]	82
Rys. 15.1. Testy w ciągu dnia – widok w aplikacji. [opracowanie własne]	84

18. SPIS TABEL

Tab. 13.1. Komunikat o uruchomieniu silnika zabawki. [opracowanie własne].....	65
Tab. 14.1. Deklaracja wartości w zakładkach. [opracowanie własne].....	76
Tab. 14.2. Funkcje do pobierania informacji z API. [opracowanie własne].....	79
Tab. 14.3. Przykładowa funkcja do wysyłania zapytania typu GET. [opracowanie własne]	79
Tab. 14.4. Komunikaty o błędach lub o sukcesie. [opracowanie własne]	80
Tab. 14.5. Funkcja pobierająca dane o stanie zabawki. [opracowanie własne]	80
Tab. 14.6. Przykładowa funkcja wysyłająca zapytania typu POST. [opracowanie własne].....	81
Tab. 15.1. Baza danych. [opracowanie własne]	83